HIMatrix

Safety-Related Controller

System Manual Compact Systems

HIMA Paul Hildebrandt GmbH

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 HIMA DVD and our website http://www.hima.de and http://www.hima.com.

© Copyright 2013, HIMA Paul Hildebrandt GmbH

All rights reserved

Contact

HIMA contact details:

HIMA Paul Hildebrandt GmbH

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 has been modified.	X	X
2.00	HIMatrix devices F*03, SILworX V4, HIMatrix CPU OS V8, COM OS V13 are taken into account	X	X
2.01	SILworX V5, adaptations from the engineering manual - *to replace the engineering manual*, parameters for remote I/Os, licensing are handled	X	X
2.02	Added: Notes on redundant safe**ethernet**, VLAN, CAN	X	

Table of Contents

1 Introduction

The safety-related compact systems described in this manual can be used for different 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 environment 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 operating the HIMatrix systems.
Product Description	Basic structure of the HIMatrix system.
Communication	Brief description of the HIMatrix compact 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 lifecycle.
Appendix	▪ Glossary ▪ Index of tables and index of figures ▪ Declaration of Conformity ▪ Index

i This document 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.

The following HIMatrix devices have additional functions:

- F60 CPU 03
- F35 03
- F31 03
- F30 03
- F10 PCI 03

All these devices are identified in this document with **F*03**. The additional features of these devices compared to standard devices are:

- Enhanced performance
- Sequence of events recording possible
- Multitasking possible
- Reload possible
- Two IP addresses

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

Programming tool	Hardware	Processor operating system	Communication operating system
SILworX	F*03	CPU OS V8 and higher	COM OS V13 and higher
SILworX	Default	CPU OS V7 and higher	COM OS V12 and higher
ELOP II Factory	Default	CPU OS up to V7	COM OS up to V12

Table 1: HIMatrix System Variants

The manual distinguishes among the different variants using:

- Separated chapters
- Tables differentiating among the versions, e.g., CPU OS V7 and higher, or CPU OS up to V7

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

Additionally, the following documents must be taken into account:

Name	Content	Document number
HIMatrix Safety Manual	Safety functions of the HIMatrix system	HI 800 023 E
SILworX Communication Manual	Description of the communication protocols, ComUserTask and their configuration in SILworX	HI 801 101 E
HIMatrix PROFIBUS DP Master/Slave Manual	Description of the PROFIBUS protocol and its configuration in ELOP II Factory	HI 800 009 E
HIMatrix Modbus Master/Slave Manual	Description of the Modbus protocol and its configuration in ELOP II Factory	HI 800 003 E
HIMatrix TCP S/R Manual	Description of the TCP S/R protocol and its configuration in ELOP II Factory	HI 800 117 E
HIMatrix ComUserTask (CUT) Manual	Description of the ComUserTask and its configuration in ELOP II Factory	HI 800 329 E
SILworX Online Help	Instructions on how to use SILworX	-
ELOP II Factory Online Help	Instructions on how to use ELOP II Factory, Ethernet IP protocol	-
SILworX First Steps	Introduction to SILworX	HI 801 103 E
ELOP II Factory First Steps	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 individual controllers and remote I/Os 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.
Italics	For parameters and system variables
`Courier`	Literal user inputs
RUN	Operating state are designated by capitals
Chapter 1.2.3	Cross-references are hyperlinks even if 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: warning, caution, notice
- Type and source of risk
- Consequences arising from non-observance
- Risk prevention

SIGNAL WORD

Type and source of risk!
Consequences arising from non-observance
Risk prevention

The signal words have the following meanings:

- 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.

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

2.1 Intended Use

This chapter describes the conditions for using HIMatrix systems.

2.1.1 Scope

The safety-related HIMatrix controllers can be used in applications up to SIL 3 in accordance with IEC 61508.

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

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

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

A system that operates in accordance with the de-energize to trip principle adopts the de-energized state if a fault occurs.

2.1.1.2 Application in Accordance with the Energize to Trip Principle

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 switches on, for instance, an actuator 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 or message reporting a triggered safety function may be required.

2.1.1.3 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 risks, 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.

2.2 Environmental Requirements

Requirement type	Range of values [1]
Protection class	Protection class III in accordance with IEC/EN 61131-2
Ambient temperature	0...+60 °C
Storage temperature	-40...+85 °C
Pollution	Pollution degree II in accordance with IEC/EN 61131-2
Altitude	< 2000 m
Housing	Standard: IP20
Supply voltage	24 VDC
[1] The values specified in the technical data apply and are essential for devices with extended environmental requirements.	

Table 3: Environmental Requirements

All the environmental requirements specified in this manual must be observed when operating the HIMatrix system.

2.2.1 Test Conditions

The devices have been tested to ensure compliance with the following standards for EMC, climatic and environmental requirements:

Standard	Content
IEC/EN 61131-2: 2007	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: 2007 + A1:2011	Electromagnetic Compatibility (EMC) Generic emission standard, industrial environments

Table 4: 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 III 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 13849), the HIMatrix system must be installed in an enclosure of the specified protection class (e.g., IP54).

Table 5: General Requirements

2.2.1.1 Climatic Requirements

The following table lists the most important tests and limits 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 und 0 °C / +55 °C, power supply not connected
	Cyclic damp-heat withstand tests: +25 °C / +55 °C, 95 % relative humidity, power supply not connected

Table 6: Climatic Requirements

Operating requirements other than those specified in this document are described in the device-specific or module-specific manuals.

2.2.1.2 Mechanical Requirements

The following table lists the most important tests and limits 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 7: Mechanical Tests

2.2.1.3 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, 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 8: Interference Immunity Tests

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

Table 9: Noise Emission Tests

2.2.1.4 Power Supply

The following table lists the most important tests and limits for the HIMatrix systems' power supply:

IEC/EN 61131-2	Verification 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...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 10: Verification of the DC Supply Characteristics

2.2.2 Noxious Gases

HIMatrix components may be operated without functional and safety restrictions in environments with noxious gas concentrations as described in the following standards:

- ANSI/ISA -S71.04:1985
  Corrosive gases, Class G3
- DIN EN 60068-2-60: 1996 (also IEC 68-2-60: 1995)

With noxious gas concentrations higher than those mentioned in the standards, a reduced component lifetime is to be expected. The user is responsible for demonstrating that the environment is sufficiently free from noxious gases.

2.3 Tasks and Responsibilities of Operators and Machine and System Manufacturers

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

Machine and system manufacturers must sufficiently validate that the HIMatrix systems were properly programmed.

2.3.1 Connection of Communication Partners

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

2.3.2 Use of Safety-Related Communication

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

2.4 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.**

2.5 Residual Risk

No imminent risk results from a HIMatrix compact system itself.

Residual risk may result from:

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

2.6 Safety Precautions

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

2.7 Emergency Information

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

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

3 Product Description

HIMatrix compact systems are compactly constructed, safety-related controllers including one safety-related processor system, a number of inputs and outputs and communication interfaces in its housing.

In addition to the controllers, HIMatrix compact system also comprise remote I/Os, which can be connected to the controllers via safe**ethernet** and expand the controllers by additional inputs and/or outputs.

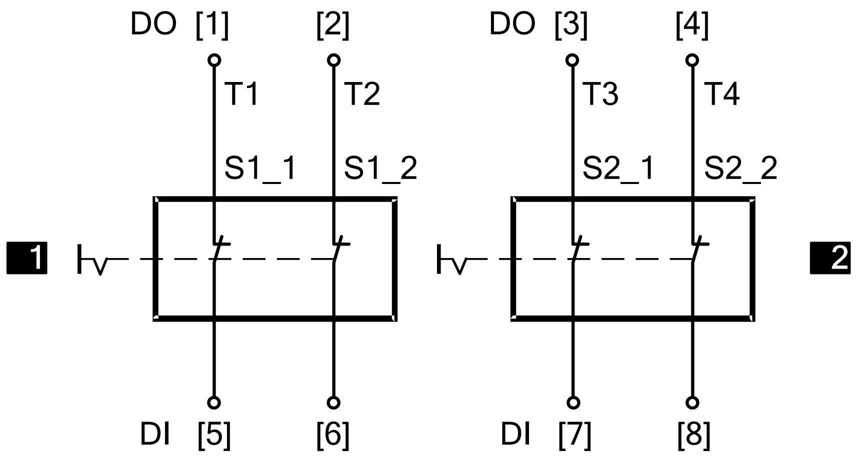
For a detailed description of the individual devices, refer to the corresponding manuals.

The compact systems may also be connected to modular systems F60, via safe**ethernet** as well.

3.1 Line Control

Line control is used to detect short-circuits or open-circuits and can be configured for the HIMatrix systems, e.g., on EMERGENCY STOP inputs complying with Cat. 4 and PL e in accordance with EN ISO 13849-1.

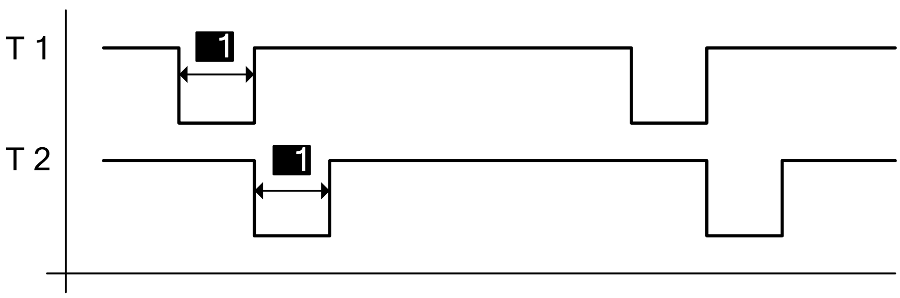
To this end, connect the digital outputs DO of the system to the digital inputs (DI) of the same system as follows:



1 EMERGENCY STOP 1
2 EMERGENCY STOP 2

EMERGENCY STOP devices in accordance with EN 60947-5-1 and EN 60947-5-5

Figure 1: Line Control



1 Configurable 5...2000 μs

Figure 2: Pulsed Signals T1 and T2

The digital outputs DO are pulsed (briefly set to low level), to monitor the wires connected to the digital inputs. The time base of the test pulse can be configured within 5...2000 µs (default value 400 µs).

i If line control is configured in a remote I/O, the remote I/O's watchdog time must be increased (default value 10 ms).

Line control detects the following faults:

- Cross-circuit between two parallel wires.
- Improper connection of two wires DO to DI, connection in contrast with the configuration specified in the software, e.g., DO 2 → DI 7 (configured), DO 2 → DI 6 (wired).
- Earth fault on one wire (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* LED blinks and the error code is created.

If such a fault occurs, the following reactions are triggered:

- The *FAULT* LED on the module's or device's front plate blinks.
- The inputs are set to low level.
- An (evaluable) error code is created.

If multiple faults occur simultaneously, the error code is the sum of all single fault error codes.

Line control can be configured if the following systems are used: F1 DI 16 01, F3 DIO 8/8 01, F3 DIO 16/8 01, F3 DIO 20/8, F20, F30 and F31.

3.2 Line Monitoring with HIMatrix F35

Refer to the HIMatrix F35 manual (HI 800 149 E) for details on how to implement open-circuit and short circuit monitoring for digital inputs.

3.3 Supply Voltage Monitoring

The HIMatrix system is a single voltage system. In accordance with IEC/EN 61131-2, the required supply voltage is defined as follows:

Supply voltage	
Nominal value	24 VDC, -15...+20 % 20.4...28.8 V
Max. permissible function limits in continuous operation	18.5...30.2 V (including ripple)
Maximum peak value	35 V for 0.1 s
Permissible ripple	r < 5 % as r.m.s. value r_{PP} < 15 % as value peak-to-peak
Ground	L- (negative pole) Earthing the ground is permitted, see Chapter 7.2.5.1.

Table 11: Supply Voltage

The power supply units of HIMatrix systems must comply with the SELV (Safety Extra Low Voltage) or PELV (Protective Extra Low Voltage) requirements.

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

The required SELV/PELV power supply units ensure safe operation.

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 12: 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.

3.4 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	*Temperature State* [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 13: Temperature Monitoring

If no or insufficient air circulates within a control cabinet and natural convection is not enough, the threshold associated with *High Temperature* in the HIMatrix 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 HIMatrix systems can be maintained.

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

3.4.1 Setting the Temperature Threshold for Messages in F*03 Devices

The temperature threshold that, if exceeded, causes a message to be issued can be defined for each base plate and each compact controller. In the SILworX Hardware Editor, use the detail view for the base plate or compact controller to configure this setting.

3.5 Short-Circuit Reaction of Output Channels

If a short-circuit occurs in an output channel, the HIMatrix automation systems switch off the affected channel. If multiple short-circuits occur, the channels are switched off individually in accordance with their current input.

If the maximum current permitted for all outputs is exceeded, all outputs are switched off and cyclically switched on again.

i The terminals for output circuits must not be plugged in while a load is connected. If short-circuits are present, the resulting high current may damage the terminals.

3.6 Alarms and Sequences of Events Recording - with F*03 Devices

The HIMatrix system is able to record alarms and sequences of events (SOE) .

3.6.1 Alarms&Events

Events are state changes of a variable that are performed by the plant or controller, and are provided with a timestamp.

Alarms are events that signalize increased 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 variable states.

Scalar events:

- Exceedance of the limit values defined for a scalar variable.
- Scalar variables have a numeric data type, e.g., INT, REAL.
- Two upper limits and two lower limits are possible.
- The following condition must be met for the limits:
  Highest limit (HH) ≥ high limit (H) ≥ normal range ≥ low limit (L) ≥ lowest limit (LL).
- A hysteresis can be effective in the following cases:
  - If the value falls below one of the upper limits.
  - If the value exceeds one of the lower limit.

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

HIMatrix can only create events if they are configured in SILworX, see Chapter 7.6. Up to 4 000 alarms and events can be defined.

3.6.2 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 3.6.3. 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.
- Low limit (L) exceeded.
- Lowest limit (LL) exceeded.
- High limit (H) exceeded.
- Highest limit (HH) exceeded.

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

3.6.3 Recording Events

The processor system records 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.

3.6.4 Transfer of Events

The X-OPC server readout events from buffer and transfers this to a third-party system for evaluation and representation. Four X-OPC servers can simultaneously read events out of a processor module.

3.7 Product Data

Designation	Value, range of values
Power supply	24 VDC, -15 %...+20 %, $r_{PP} \leq$ 15 %, externally fused
	Gold capacitor (for buffering date/time)
Operating temperature	0...+60 °C
Storage temperature	-40...+85 °C
Type of protection	IP20
Dimensions	Depending on the device
Weight	Depending on the device

Table 14: Specifications

The device specifications are described in the device manuals.

3.8 Licensing with F*03 Systems

The following features of the controllers must be activated using a common license:

- Multitasking
- Reload
- Sequence of events recording

The software activation code can be generated on the HIMA website using the system ID of the controller (value 1...65 535). To this end, the **SMR** license must be activated.

The software activation code is intrinsically tied to this system ID. One license can only be used one time for a specific system ID. For this reason, only activate the code when the system ID has been uniquely defined.

4 Communication

Communication runs via the following interfaces:

- Ethernet interfaces
- Fieldbus interfaces

4.1 HIMatrix Communication Protocols

Depending on the HIMatrix controller and its interfaces, different communication protocols can be activated:

1. safe**ethernet** and SNTP are activated by default in all HIMatrix systems.
2. Communication via serial interfaces requires the use of appropriate fieldbus submodules and, possibly, an additional license (software activation code).
   Refer to the communication manuals (HI 801 101 E, HI 800 009 E, HI 800 003 E and HI 800 329 E) for further information.
3. All Ethernet protocols can be tested without software activation code for 5000 operating hours.

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

Order the software activation code on time!

The software activation code can be generated on the HIMA website using the system ID of the controller (value 1...65 535).

The software activation code is intrinsically tied to this system ID. One license can only be used one time for a specific system ID. For this reason, only activate the code when the system ID has been uniquely defined.

HIMatrix systems support the following Ethernet interface communication protocols.

- safe**ethernet**, redundant operation possible for F*03 controllers
- Modbus TCP master
- Modbus TCP slave
- Send/Receive TCP
- SNTP
- EtherNet/IP
  Only up to CPU OS V6.x (ELOP II Factory)
- PROFINET IO controller
  Only F*03
- PROFINET IO device
  Only F*03

Each protocol can be used once per controller.

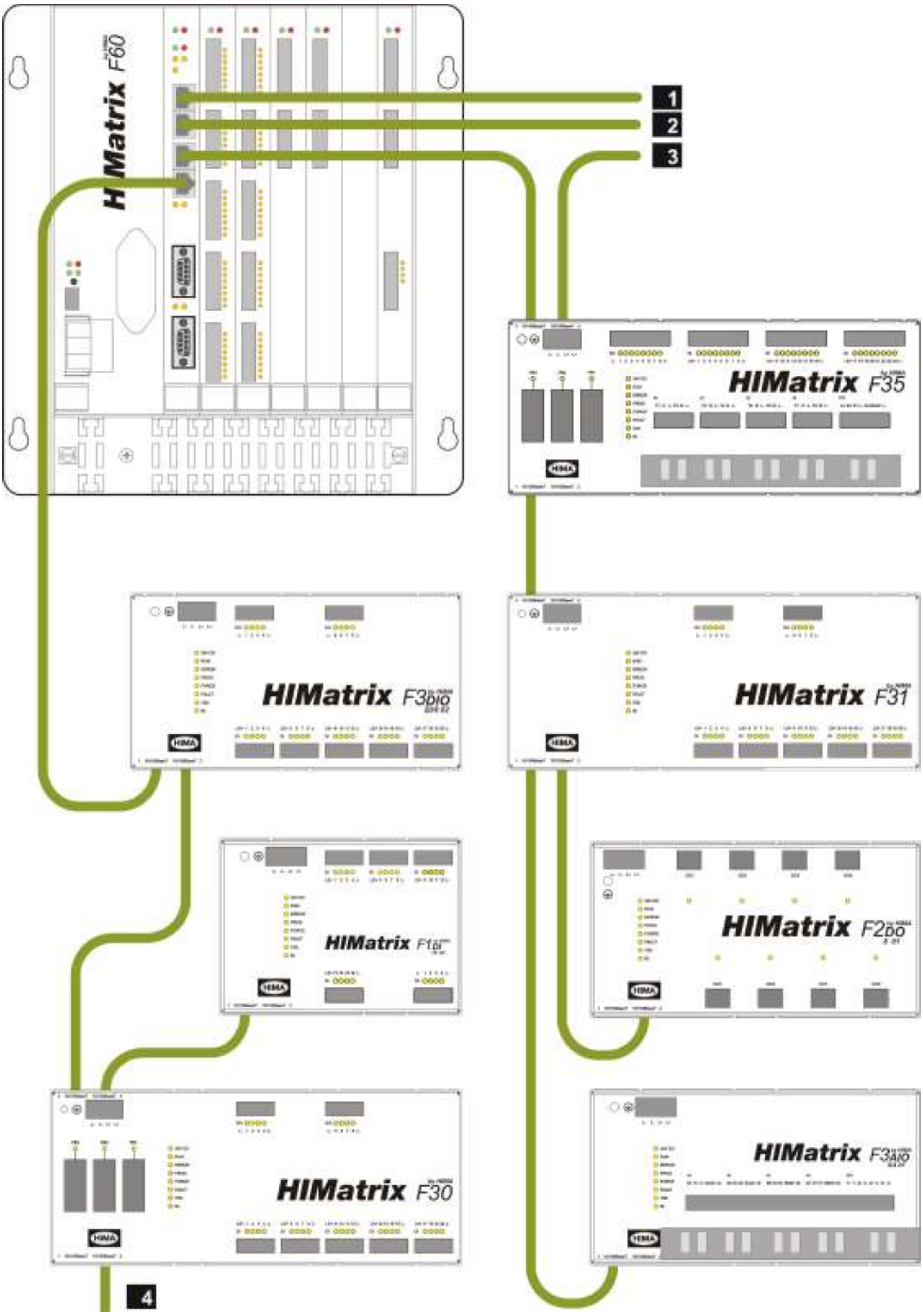
Communication options for serial interfaces are described in Chapter 4.3.

4.2 Ethernet Communication

4.2.1 safe**ethernet**

An overview of safe**ethernet** is available in the corresponding chapter of the SILworX communication manual (HI 801 101 E).

When configuring safety-related communication, observe the instructions specified in the safety manual (HI 800 023 E).



1 Example of other F60
2 PADT
3 Superior safe**ethernet**
4 PADT

Figure 3: safe**ethernet**/Ethernet Networking Example

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

NOTE

Ethernet operation may be disturbed!

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

If controllers and remote I/Os with different versions of operating systems are connected via safe**ethernet**, the following cases must be observed:

Operating system of controller	Operating system of remote I/O	safe**ethernet** connection possible?
CPU OS V7 and higher	CPU OS V7 and higher	Yes
CPU OS up to V7	CPU OS up to V7	Yes
CPU OS up to V7	CPU OS V7 and higher	Yes
CPU OS V7 and higher	CPU OS up to V7	No

Table 15: Connection of Controllers and Remote I/Os with Different Operating Systems

Controllers with different operating system versions (CPU OS V7 and higher and CPU OS up to V7) can be connected with cross-project communication, refer to the communication manual (HI 800 101 E)

4.2.2 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 the 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.

4.2.3 Connectors for safe**ethernet**/Ethernet

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

To interconnect the HIMatrix systems, only interference-free Ethernet cables may be used, e.g., shielded (STP)!

4.2.4 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 operating system version of the controller.

- CPU OS V7 and higher: SILworX
- CPU OS up to V7: 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.

4.2.5 Ethernet Communication Protocols

In addition to the safe**ethernet**, HIMatrix supports the following Ethernet communication protocols:

- SNTP
- Modbus TCP
- Send & Receive TCP
- PROFINET IO and PROFIsafe (with F*03 only)
- EtherNet/IP (up to CPU OS V7)

Refer to the corresponding communication manual for details on the various protocols.

4.2.5.1 SNTP

The SNTP protocol (simple network time protocol) is used to synchronize the time of the HIMA resources via Ethernet. The current time can be retrieved via Ethernet in predefined time intervals from a PC, or a HIMA resource configured as SNTP server.

HIMA resources with COM OS V6 and higher, can be configured and used as SNTP server and/or as SNTP client. The SNTP server communicates with the SNTP client via the non-safe UDP protocol on port 123.

For further details on the SNTP protocol, refer to the SILworX communication manual (HI 801 101 E) or the online help of the programming tool.

4.2.5.2 Modbus TCP

The HIMA-specific designation for the **non**-safety-related Modbus TCP is Modbus Master/Slave Eth.

The fieldbus protocols Modbus master/slave can communicate with the Modbus TCP via the Ethernet interfaces of the HIMatrix controllers.

In a standard Modbus communication, the slave address and a CRC checksum are transferred in addition to the instruction code and data, while in a Modbus TCP, this function is assumed by the subordinate TCP protocol.

For further details on the Modbus TCP, refer to the SILworX communication manual (HI 801 101 E) or HIMatrix Modbus master/slave manual (HI 800 003 E).

4.2.5.3 Send & Receive TCP

S&R TCP is a manufacturer-independent, **non**-safety-related protocol for cyclic and acyclic data exchange and does not use any specific protocols other than TCP/IP.

With S&R TCP, HIMatrix systems are able to support almost every third-party system as well as PCs with implemented socket interface to TCP/IP (e.g., Winsock.dll).

For further details on the S&R TCP, refer to the SILworX communication manual (HI 801 101 E) or HIMatrix TCP/SR manual (HI 800 117 E).

4.2.5.4 PROFINET IO and PROFIsafe (with F*03 only)

The non-safety-related protocol PROFINET IO and the safety-related protocol PROFIsafe are only available for F*03 controllers and must be configured using SILworX. Refer to the SILworX communication manual (HI 801 101 E) for more information about communication.

4.2.5.5 EtherNet/IP (up to CPU OS V7)

EtherNet/IP communication is only supported in ELOP II Factory. EtherNet/IP is not supported in SILworX.

EtherNet/IP (Ethernet Industrial Protocol) is an open industrial communication standard for exchanging process data via Ethernet.

For further information, see http://www.odva.org (ODVA = Open DeviceNet Vendor Association).

EtherNet/IP enables HIMatrix controllers to communicate with other Ethernet/IP devices (e.g., PLC, sensors, actuators and industrial robots).

The physical connection of EtherNet/IP runs over Ethernet interfaces with 10/100 Mbit/s.

For HIMatrix controllers, the EtherNet/IP protocol can be configured in the ELOP II Factory Hardware Management (with hardware revision 02).

A HIMatrix controller can be configured as EtherNet/IP scanner and/or as EtherNet/IP target.

Refer to the ELOP II Factory online help for further details on the EtherNet/IP communication.

4.3 Fieldbus Communication

The F20, F30, and F35 controllers are equipped with connectors for fieldbus communication (Modbus, PROFIBUS, and INTERBUS).

Prior to resetting a controller, take the consequences for other fieldbus subscribers into account! If required, appropriate measures must be taken, e.g., the separation of the fieldbus connection.

The F20, F30 and F35 controllers must be equipped with fieldbus submodules for fieldbus communication. The installation of the fieldbus submodules is optional and is carried out by the manufacturer. The fieldbus interfaces are not operational without fieldbus submodule.

4.3.1 Equipment of Fieldbus Interfaces with Fieldbus Submodules

The HIMatrix controllers can be equipped with fieldbus submodules in accordance with the following table:

Controller	FB1	FB2	FB3
F20	Freely equippable	Integrated RS485 [1)]	---
F30	Freely equippable	Freely equippable	Integrated RS485 [1)]
F35	Freely equippable	Freely equippable	Integrated RS485 [1)]
F60	Freely equippable	Freely equippable	---
[1)] RS485 fieldbus interfaces can be used for Modbus (master or slave) or ComUserTask.			

Table 16: Equipment of Fieldbus Interfaces with Fieldbus Submodules

i Only HIMA is authorized to mount the fieldbus submodules; failing which the warranty will be void.

Some fieldbus submodules are shown in Table 17. All available fieldbus submodules are listed in the SILworX communication manual (HI 801 101 E).

Fieldbus submodule	Protocols
PROFIBUS master	PROFIBUS DP master
PROFIBUS slave	PROFIBUS DP slave
RS485 module	RS485 for Modbus (master or slave) and ComUserTask
RS232 module	RS232 for ComUserTask
RS422 module	RS422 for ComUserTask
SSI module	SSI for ComUserTask
CAN module	CAN - for F*03 only

Table 17: Fieldbus Submodule

The fieldbus submodule is selected when ordering the controller using the part number.

Depending on the fieldbus submodule, the communication protocols must be activated. For further details on the procedures for registering and activating the protocols, refer to the communication manuals, see Table 2.

4.3.2 Restrictions for Operating Protocols Simultaneously

- PROFIBUS DP master or slave can only be operated on one fieldbus interface, i.e., two PROFIBUS masters or slaves may not be operated at a time within a resource and, therefore, they will not function.
- Modbus master/slave RS485 can only be operated on one fieldbus interface. However, simultaneous operation via RS485 and Ethernet is allowed.

i No safety-related communication can be ensured with the available fieldbus protocols.

The communication system with fieldbus interfaces is connected to the safety-related processor system. Only devices with safe electrical separation may be connected to the interfaces.

i The fieldbus submodules PROFIBUS master can be used on controllers F20, F30, F35 or F60 with hardware revision 02.

5 Operating System

The operating system includes all basic functions of the HIMatrix controller.

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:	Preset and not changeable 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	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 18: 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.

Additional features of one operating system version can only be used if a corresponding version of the programming tool is used.

5.2 Indication of the Operating System Versions

5.2.1 SILworX

The current COM and CPU operating system versions can be displayed using the module data overview, see the SILworX online help. The module data overview is activated from within the online view of the Hardware Editor, selecting the menu option **Online**.

The *OS* column contains the list of the current operating system versions.

5.2.2 ELOP II Factory

The current COM and CPU operating system versions can be displayed from within the Control Panel. The OS tab lists the current operating system versions loaded in the controller with the corresponding loader and CRC versions. Refer to the ELOP II Factory online help for more details.

5.3 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

5.3.1 Permanent Faults on Inputs or Outputs

A fault 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.

If the input channels are faulty, the operating system sends the safe value 0 or the initial value for further 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 a given output and even the second switch-off option is not effective, the controller enters the STOP state. The outputs are then switched off by the watchdog of the processor system.

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.

5.3.2 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.

5.3.3 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:

- Processor OS up to V6.44 for controllers, and up to V6.42 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 OS V6.44 and higher, and for remote I/Os V6.42 and higher:
  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.

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

- 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

5.4.1 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 parameters to TRUE using a program logic causes 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/VALID CONFIGURATIO N	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 CONFIGURATIO N	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 19: Modes of Operation for the Processor System

5.4.2 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 processor operating system V7 and higher.
- ELOP II Factory for HIMatrix systems with processor operating system up 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 during 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 20: User Program Modes of Operation

6.2 User Program Cycle Sequence, Multitasking with F*03 Devices

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

1. Process the input data.
2. Run the user program.
3. Supplying the output data.

These phases do not include special tasks, e.g., reload, that might be executed within a CPU cycle.

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

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

6.2.1 Multitasking

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 started and stopped independently from one another. SILworX displays the states of the individual user programs on the Control Panel and allows the user to operate them.

Using multitasking, the second phase changes so that a CPU cycle performs the following tasks:

1. Process the input data.
2. Processing of all the user programs.
3. Supplying the output data.

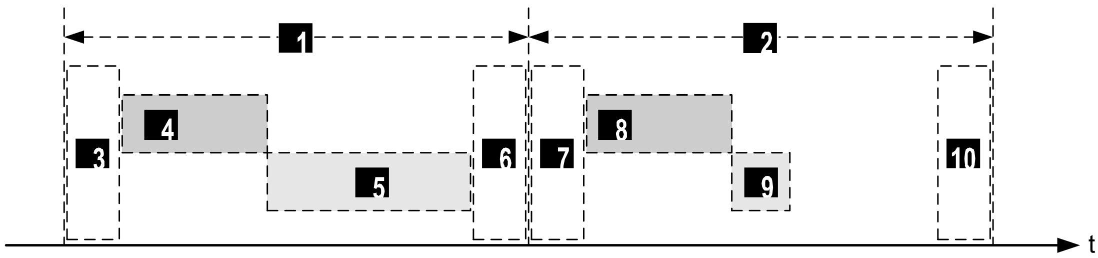
In the second phase, the HIMatrix can run up to 32 user programs. Two scenarios are possible for each user program:

- 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 is provided 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 cycle.

The example in Figure 4 shows both scenarios in a project containing two user programs: *Prg 1* and *Prg 2*.



1 First CPU cycle considered.
2 Second CPU cycle considered.
3 Input processing in the first CPU cycle
4 First *Prg 1* cycle considered
5 First portion of the considered *Prg 2* cycle
6 Output processing in the first CPU cycle
7 Input processing in the second CPU cycle
8 Second *Prg 1* cycle considered
9 Second portion of the considered *Prg 2* cycle
10 Output processing in the second CPU cycle

Figure 4: CPU Cycle Sequence with Multitasking

Each *Prg 1* cycle is completely processed during each CPU cycle. *Prg 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 *Prg 2* cycle requires two CPU cycles to be processed. *Prg 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 *Prg 2* is two times longer than that of *Prg 1*.

Upon completion of the first part 5 of the *Prg 2* cycle under consideration, *Prg 2* processing is completely aborted and only resumed when 9 starts. During its cycle, *Prg 2* processes the data provided by the system during 3. The results of *Prg 2* are available to the system during 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
Watchdog Time	Resource Watchdog Time	Resource, Multitasking Editor
Target Cycle Time [ms]	Required or maximum cycle time	Resource, Multitasking Editor
Multitasking Mode	Use of the execution duration unneeded by the user program, e. g., the difference between actual execution duration in one CPU cycle and the defined *Max. Duration for Each Cycle [µs]*.	Resource, Multitasking Editor
	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 During the execution time not needed by the user programs, the processor waits for the time to expire, thus increasing the cycle.	
Target Cycle Time Mode	Use of *Target Cycle Time [ms].*	Resource, Multitasking Editor
Program ID	ID for identifying the program when displayed in SILworX	User Program
Priority	Importance of a user program; highest priority: 0.	User Program
Program's Maximum Number of CPU Cycles	Maximum number of CPU cycles required to process one user program cycle.	User Program
Max. Duration for Each Cycle [µs]	Time permitted for executing the user program within a CPU cycle.	User Program

Table 21: Parameters Configurable for Multitasking

Observe the following rules when setting the parameters:

- If *Max. Duration for Each Cycle [µs]* is set to 0, the execution time of the user program is not limited, e.g., it is always processed completely. Therefore, the number of cycles may be set to 1 in this case.

- The sum of the *Max. Duration for Each Cycle [µs]* parameters in all user programs must not exceed the resource watchdog time. Make sure that sufficient reserve is planned for processing the remaining system tasks.
- The sum of the *Max. Duration for Each Cycle [µs]* parameters in all user programs must be large enough to ensure that sufficient reserve is available to maintain the target cycle time.
- The *Program ID*s of all user programs must be unique.

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

SILworX uses these parameters to calculate the user program watchdog time:
User program watchdog time = *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 operate interference-free and independently to one another. However, reciprocal influence can be caused by:

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

NOTE

Reciprocal influence of 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 in one location, either in a user program or from the hardware!**

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

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

The operating system defines in which order the user programs are executed in accordance with the following scheme:

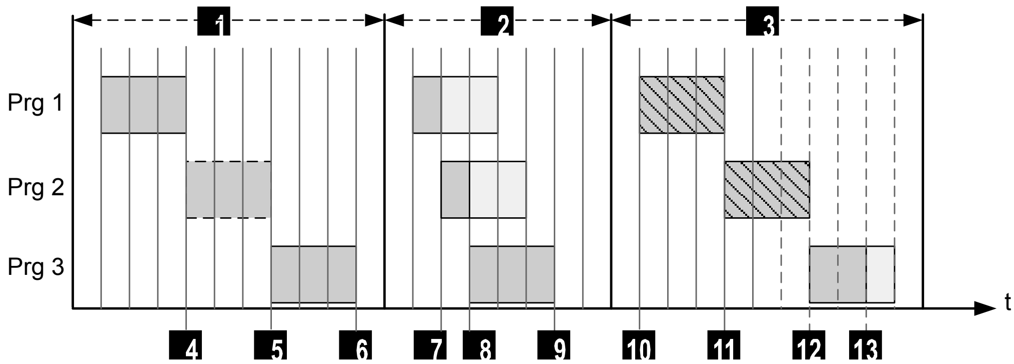
- User programs with lower priority are executed before user programs with higher priority.
- If the user programs have the same priority, the system executes them in ascending order of the *Program ID*s.

This order is also followed when starting and stopping the user program during the start and stop of the PES, respectively.

6.2.2 Multitasking Mode

Three operation modes exist for multitasking. These modes differ in how the time that is not needed for executing the CPU cycle of the user programs is used. One of these three modes can be selected for every resource.

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 (*Prg 1*, *Prg 2* and *Prg 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 *Max. Duration for Each Cycle [µs]* of *Prg 1* has expired, *Prg 2* starts.

5 *Max. Duration for Each Cycle [µs]* of *Prg 2* has expired, *Prg 3* starts.

6 *Max. Duration for Each Cycle [µs]* of *Prg 3* has expired, completion of the first CPU cycle.

7 Completion of the *Prg 1* cycle, *Prg 2* is resumed.

8 Completion of the *Prg 2* cycle, *Prg 3* is resumed.

9 *Max. Duration for Each Cycle [µs]* of *Prg 3* has expired, completion of the second CPU cycle.

10 The next user program cycle of *Prg 1* starts.

11 *Max. Duration for Each Cycle [µs]* of *Prg 1* has expired. The next user program cycle of *Prg 2* starts.

12 *Max. Duration for Each Cycle [µs]* of *Prg 2* has expired, *Prg 3* starts.

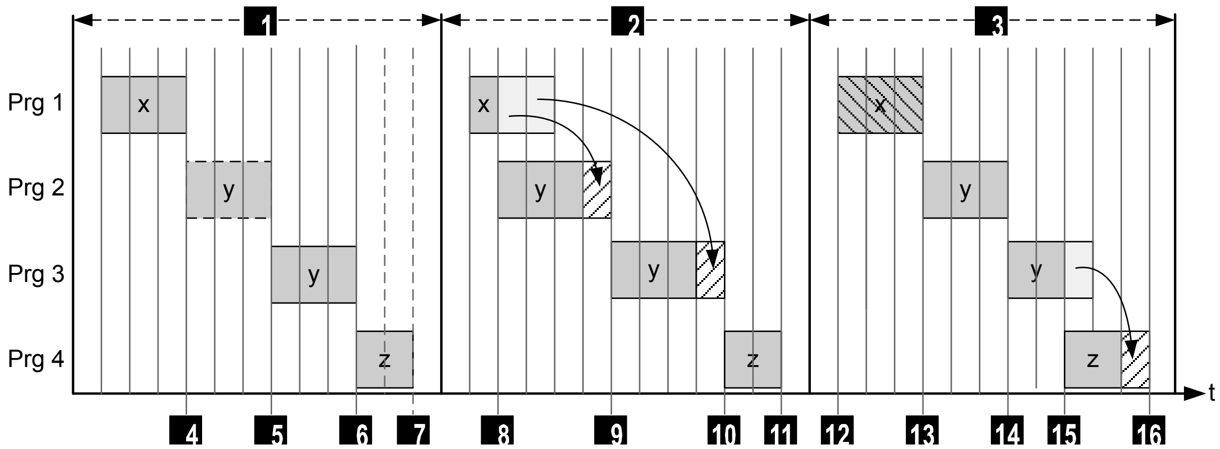
13 Completion of the *Prg 3* cycle.

Figure 5: 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.

   Four user programs are used in the example: *Prg 1…Prg 4*. The following priorities are allocated to the user programs:

   - *Prg 1* has the lowest priority, priority *x*
   - *Prg 2* and *Prg 3* have a medium priority, priority *y*
   - *Prg 4* has the highest priority, priority *z*



1 First CPU cycle considered.

2 Second CPU cycle considered.

3 Third CPU cycle considered.

4 *Max. Duration for Each Cycle [µs]* of *Prg 1* has expired, *Prg 2* starts.

5 *Max. Duration for Each Cycle [µs]* of *Prg 2* has expired, *Prg 3* starts.

6 *Max. Duration for Each Cycle [µs]* of *Prg 3* has expired, *Prg 4* starts.

7 *Max. Duration for Each Cycle [µs]* of *Prg 4* has expired, completion of the first CPU cycle.

8 Completion of the *Prg 1* cycle, *Prg 2* is resumed. The remaining duration is distributed to the *Max. Duration for Each Cycle [µs]* of *Prg 2* and *Prg 3* (medium priority y) (arrows).

9 *Prg 2 Max. Duration for Each Cycle [µs]* + proportional remaining duration of *Prg 1* have expired, *Prg 3* is resumed.

10 *Prg 3 Max. Duration for Each Cycle [µs]* + proportional remaining duration of *Prg 1* have expired, *Prg 4* starts.

11 *Max. Duration for Each Cycle [µs]* of *Prg 4* has expired, completion of the first CPU cycle.

12 The next user program cycle of *Prg 1* starts.

13 Completion of *Prg 1 Max. Duration for Each Cycle [µs]*, *Prg 2* resumes.

14 Completion of *Prg 2 Max. Duration for Each Cycle [µs]*, *Prg 3* is resumed.

15 Completion of the *Prg 3* cycle, *Prg 4* is resumed. The remaining duration is added to *Prg 4* (highest priority *z*).

16 *Max. Duration for Each Cycle [µs]* of *Prg 4*+ remaining duration of *Prg 3* have expired, completion of the third CPU cycle.

Figure 6: Multitasking Mode 2

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

- STOP
- ERROR
- TEST_MODE

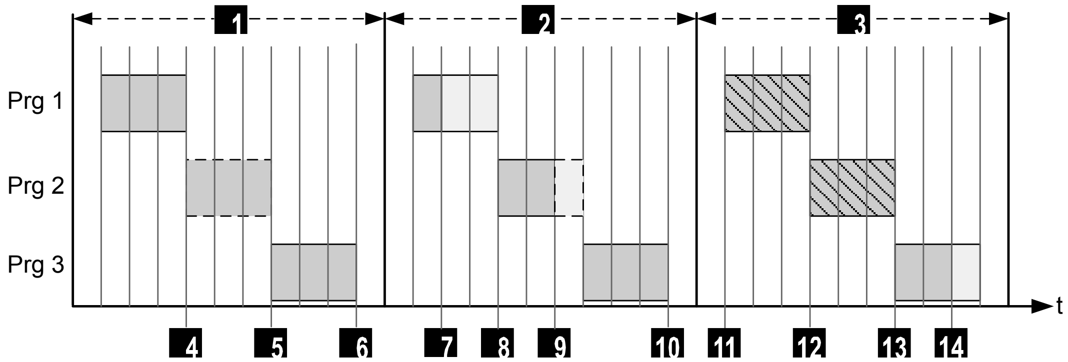
As a consequence, the number of CPU cycles required to process another user program cycle could increase.

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

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.

   The example examines user programs named *Prg 1*, *Prg 2* and *Prg 3*:



1 First CPU cycle considered.

2 Second CPU cycle considered.

3 Third CPU cycle considered.

4 *Max. Duration for Each Cycle [µs]* of *Prg 1* has expired, *Prg 2* starts.

5 *Max. Duration for Each Cycle [µs]* of *Prg 2* has expired, *Prg 3* starts.

6 *Max. Duration for Each Cycle [µs]* of *Prg 3* has expired, completion of the first CPU cycle. *Prg 1* is resumed.

7 Completion of the *Prg 1* cycle. Waiting for the remaining duration.

8 *Max. Duration for Each Cycle [µs]* of *Prg 1* has expired. *Prg 2* is resumed.

9 Completion of the *Prg 2* cycle. Waiting for the remaining duration.

10 *Max. Duration for Each Cycle [µs]* of *Prg 3* has expired. Completion of the second CPU cycle.

11 The next user program cycle of *Prg 1* starts.

12 *Max. Duration for Each Cycle [µs]* of *Prg 1* has expired. The next user program cycle of *Prg 2* starts.

13 *Max. Duration for Each Cycle [µs]* of *Prg 2* has expired. *Prg 3* is resumed.

14 Completion of the *Prg 3* cycle. Standby time until the *Prg 3 Max. Duration for Each Cycle [µs*] has expired. Completion of the third CPU cycle.

Figure 7: 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 F*03 Devices

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 points 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 points 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 function.

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

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

During a reload, the global and local variables should be 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 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 allows a reload in accordance with Table 22.

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	-	-	-	-
safe**ethernet**	-	-	•	-
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 22: Reloading after Changes

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

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

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

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

6.4 General Information

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.**

6.5 Forcing - CPU OS V7 and Higher

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 in Connection with F*03

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:

- If global forcing is used, the following settings can be selected:
  - The resource stops.
  - The resource continues to operate.
- If local forcing is used, 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.

Automatic Forcing Reset

The operating system resets forcing in the following cases:

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

In these cases, the user program changes the force settings as follows:

- Force values to 0 or FALSE
- Force switch to OFF
- Force main switch to OFF

During a reload, local and global force values as well as forcing times and force timeout reactions continue to be valid.

Global force values and force switches can be set when a resource is stopped. The configured values become valid after restarting the resource and forcing.

Local force values and force switches can be set when the user program is stopped. The configured values become valid after restarting the user program and forcing.

6.5.2 Forcing in Connection with Standard devices and Modules

Forcing in HIMatrix standard 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 Variables

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 variable 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.

Time Limits

A time limit can be defined for global 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:

- The resource stops.
- The resource continues to operate.

Local Variables

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 define the local process value.

6.5.3 Restriction to 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 linked to a digital input connected to a key switch.

  This system variable is not always enabled, see Table 23.

Devices	Effect description
F*03	*Force Deactivation* prevents global and local forcing from being started and stops an on-going forcing process.
Default	*Force Deactivation* prevents global forcing from being started and stops an on-going forcing process. *Force Deactivation* inhibits the **Edit Local Process Values** command, but it does not reset changed local variables to their previous process value.

Table 23: Effect of the *Force Deactivation* System Variable

6.6 Forcing - CPU OS up to V7

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 are only valid until overwritten by the user program!
However, if the user program does not overwrite the values, e.g., if an EN input is set to FALSE, the force value is used as a process value in the ensuing calculations.

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.

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, overrunning 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	OFF	OFF /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	OFF	OFF	ON
Remaining Force Time	Time-limit for the force value, time (in seconds)	0	0	Remaining force time or -1
[1] The *Force Allowed* and *Stop at Force Timeout* switches cannot be changed when a controller is operating and 'locked', i.e., define these settings prior to locking the controller.				

Table 24: Force Switches and Parameters up to CPU OS V7

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 retained, 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 HIMatrix compact systems comprises the following phases:

- Mounting the devices at suitable places

Take the dissipation of the generated heat into account.

- Electrical connection of power supply, earthing, sensors, and actuators
- Configuration
  - Writing the user program
  - Definition of safety, communication and other parameters

7.1 Considerations about Heat

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

During the system installation, it must be ensured that the approved environmental requirements have been adhered to. Lowering the operating temperature increases the lifetime and the reliability of the electronic components within the systems.

7.1.1 Heat Dissipation

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

Mounting type and position must be chosen such that heat dissipation is ensured.

The power dissipation of the installed equipment is decisive for determining the fan components. It is assumed that heat load and unhindered natural convection are uniformly distributed (see Chapter 7.1.1.3).

7.1.1.1 Definitions

P_V [W] Power dissipation (heat capacity) of the electronic components within the enclosure

A [m^2] Effective enclosure surface area (see Table 25)

k [W/m^2 K] Heat transfer coefficient of the enclosure, steel sheet: ~ 5.5 W/m^2 K

7.1.1.2 Installation Type

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

Enclosure installation type in accordance with VDE 0660, Part 5	Calculation of A [m^2]
Individual enclosure, free-standing on all sides	A = 1.8 x H x (W + D) + 1.4 x W x D
Individual enclosure for wall mounting	A = 1.4 x W x (H + D) + 1.8 x H x D
First or last enclosure in a suite, free-standing	A = 1.4 x B x (W + H) + 1.8 x W x H
First or last enclosure in a suite, for wall mounting	A = 1.4 x H x (W + D) + 1.4 x W x D
Central enclosure, free-standing	A = 1.8 x W x H+ 1.4 x W x D + H x D
Central enclosure for wall mounting	A = 1.4 x W x (H + D) + H x D
Enclosure within a suite, for wall mounting, covered roof surface	A = 1.4 x W + H+ 0.7 x W x D + H + D

Table 25: Installation Type

7.1.1.3 Natural Convection

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

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

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

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

Example: Calculation of power dissipation P_V for the F35 controller

- Idle current consumption of the controller: 0.75 A at 24 V.
- 8 digital outputs with current consumption of 1 A each at 2 V.
- The current consumption of the digital, analog and counter inputs can be neglected.

As a result, the maximum thermal power dissipation is approx. 34 W.

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

i All considerations about heat must take **every** component within a cabinet or enclosure into account!

7.2 Installation and Mounting

The safety-related HIMatrix controller systems can be installed on mounting surfaces, and also in closed enclosures such as control stations, terminal boxes and control cabinets. They have been developed in compliance with the relevant standards for EMC, climate and environmental requirements.

The list of the standards that must be observed is specified in Chapter 2.2 and in the manuals for the HIMatrix systems.

The protection class of the HIMatrix systems (IP20) can be increased by installing them in an appropriate enclosure in accordance with the requirements. In doing so, appropriate heat dissipation must be ensured (see Chapter 7.1).

The HIMatrix compact systems are mounted on a 35 mm DIN rail and not directly on a base.

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 device is protected from electrostatic discharge, e.g., by storing it in its packaging.**

7.2.1 Mounting

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

Horizontal mounting (with reference to the label on the front plate) is prescribed for all systems to ensure proper ventilation. Vertical mounting requires additional measures to ensure sufficient ventilation.

Refer to the relevant manuals, for more information on the dimensions of the various devices.

The minimum clearances of the HIMatrix systems among one another, to external devices and to the control cabinet enclosure are:

- Vertically, 100 mm.
- Horizontally, approx. 20 mm (with the F60, determined by the joint bars).

The mounting space (construction depth) must also be taken into account for attaching the connectors for inputs and outputs, and for communication (see Chapter 7.2.3).

Mount the devices on a DIN rail as follows:

To mount the device on the DIN rail

1. Shift the latch on the rear side of the device downwards, press it against the housing frame and snap it into position.
2. Attach the guiding rail located on the rear side of the device to the upper edge of the DIN rail.
3. Press the device against the rail and release the latch again to secure the device on the rail.

The device is attached to the DIN rail.

To remove the device from the DIN rail

1. Insert a flathead screwdriver into the gap between the housing and the latch, using it as a lever to move the latch downward and simultaneously lift the device from the rail

The device is removed from the DIN rail.

i

- To ensure efficient cooling, the device must be mounted on a horizontal DIN rail.
- A distance of at least 100 mm above and below the device must be maintained.
- The device must not be mounted above heating equipment or any heat source.

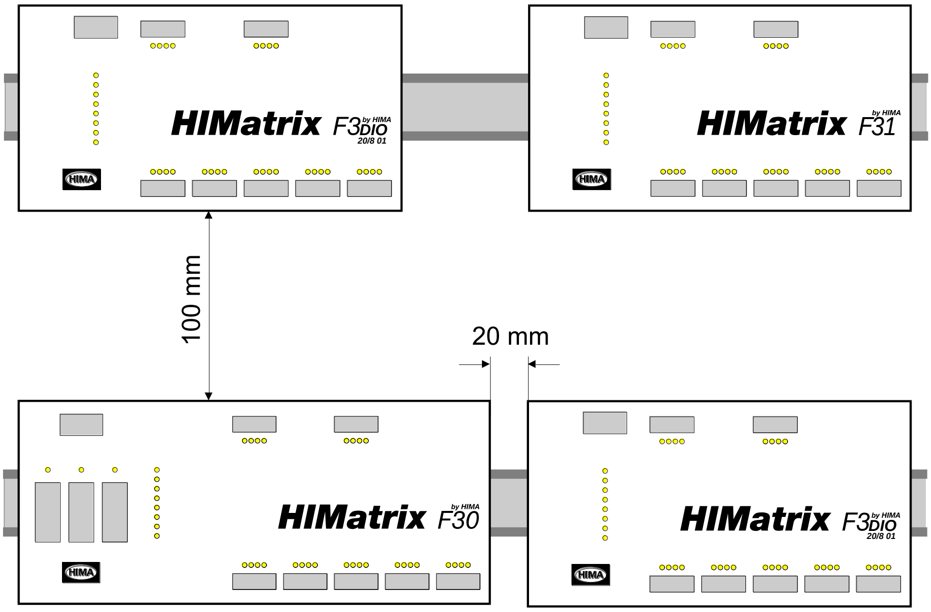


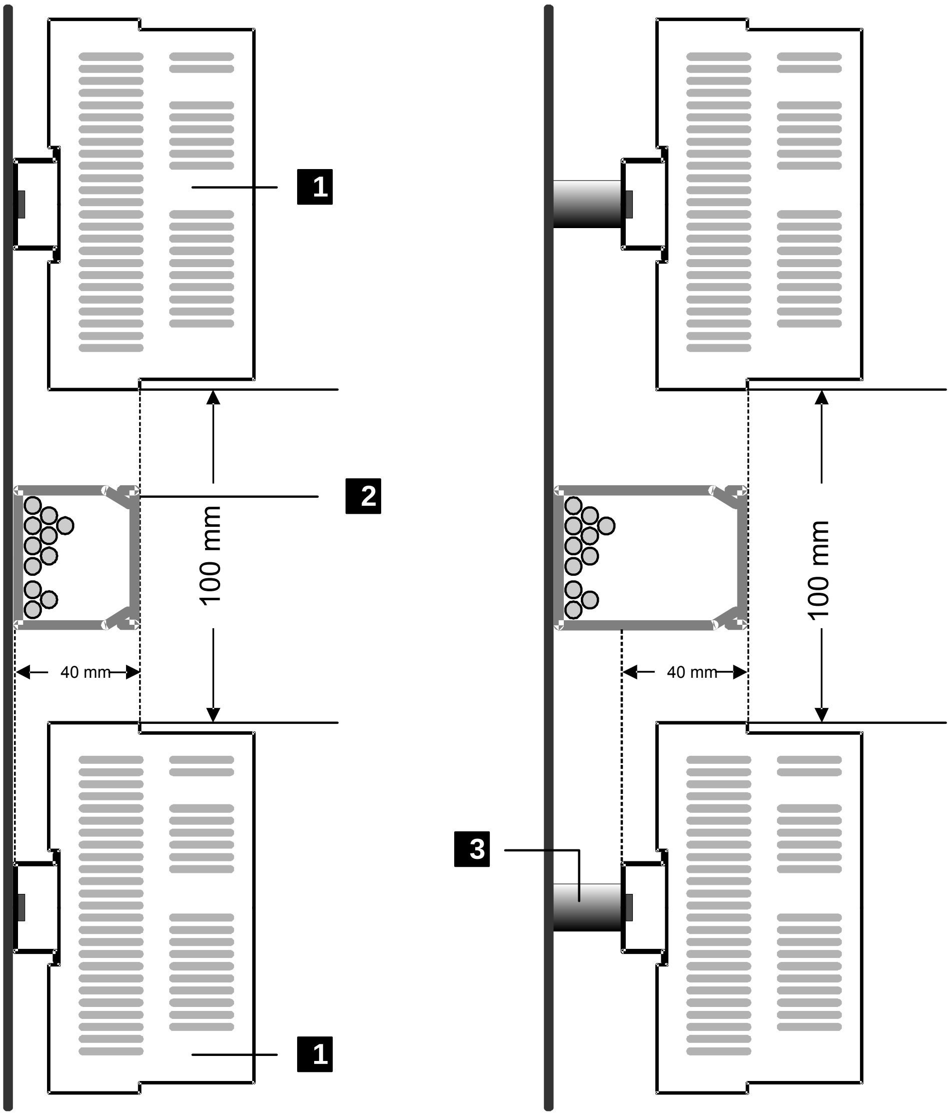
Figure 8: Minimum Clearances with HIMatrix Compact Systems

7.2.1.1 Routing Cables

Use the shortest possible route to connect the cable duct to the HIMatrix system. Avoid routing cables across the front plates of the system.

7.2.2 Air Circulation

The ventilation slots in the housing must not be obstructed. For this reason, when mounting compact systems and cable ducts at the same level, the cable ducts are limited to a height of 40 mm. If the cable ducts are higher, the mounting rails must be placed on spacers:

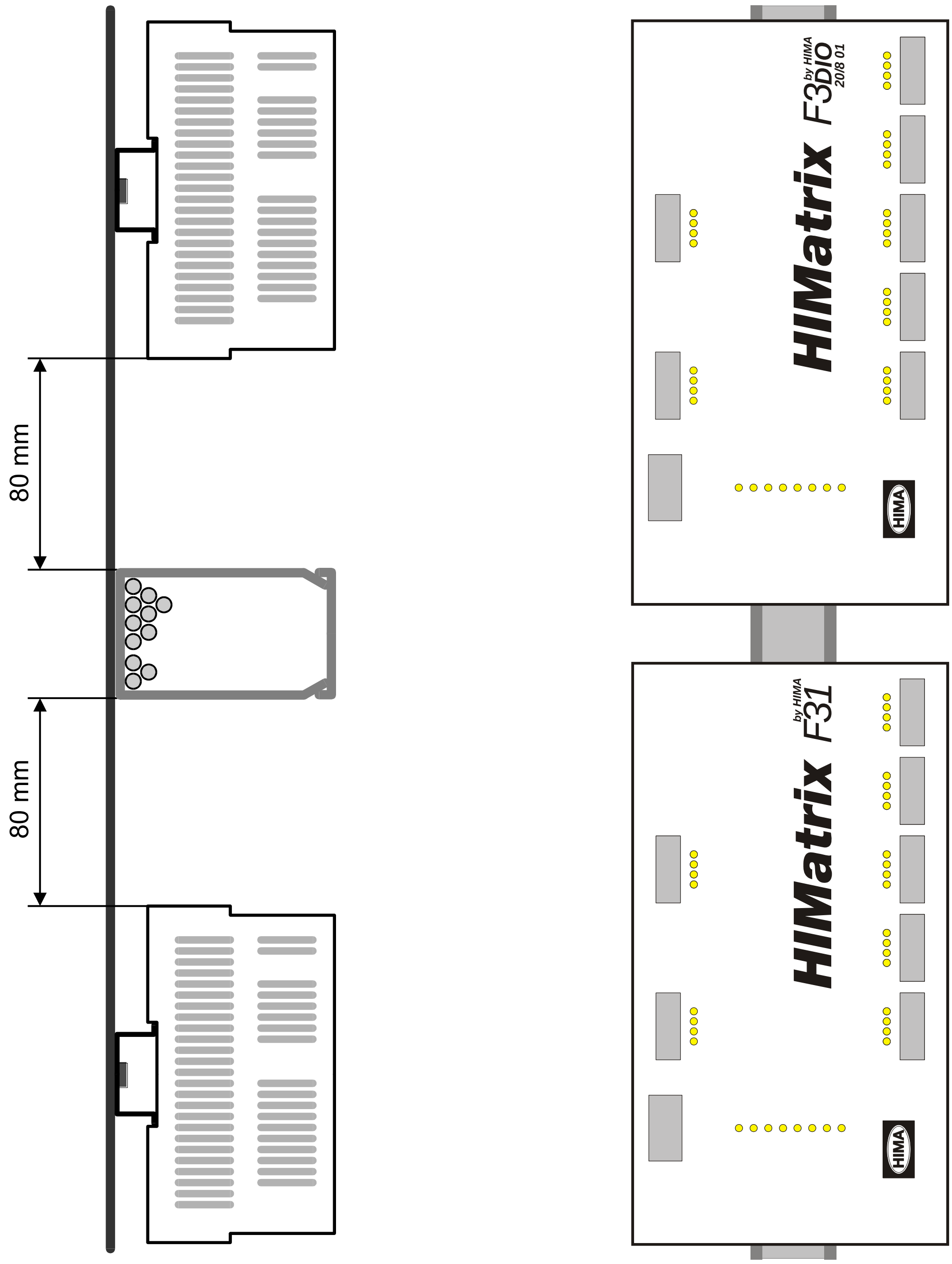


1 Compact system
2 Cable duct
3 Spacer

Figure 9: Use of Cable Ducts and Spacers

If more than two HIMatrix systems are mounted one upon the other (even if the minimum vertical clearance of 100 mm is maintained), additional ventilation measures are required to ensure uniform temperature distribution.

The illustration below left shows the minimum clearances if no spacers are used for the mounting rails:



Mounting clearances without spacers Vertical mounting of HIMatrix systems

Figure 10: Mounting without Spacers and Vertical Mounting

i HIMatrix devices may only be mounted vertically if sufficient ventilation is ensured!

With open mounting surfaces, it is not difficult to remain within the maximum operating temperature limits provided that the minimum clearances are maintained and air circulation is not obstructed.

7.2.3 Construction Depths

The construction depth specified in the table below must be used for the HIMatrix compact systems due to the communication connectors and the I/O level. The various construction depths are measured from the mounting rail.

HIMatrix System	Construction Depth
F1 DI 16 01	100 mm
F2 DO 4 01	100 mm
F2 DO 8 01	120 mm
F2 DO 16 01	100 mm
F2 DO 16 02	120 mm
F3 DIO 8/8 01	100 mm
F3 DIO 16/8 01	100 mm
F3 DIO 20/8 02	100 mm
F3 AIO 8/4 01	100 mm
F20 with PROFIBUS plug[1)] without PROFIBUS plug	----[1)] mm 100 mm
F30 with PROFIBUS plug[1)] without PROFIBUS plug	----[1)] mm 100 mm
F31	100 mm
F35 with PROFIBUS plug[1)] without PROFIBUS plug	----[1)] mm 100 mm
[1)] Construction depth = depth of HIMatrix + depth of PROFIBUS plug Right plug: 100 mm + 50 mm 45º plug: 100 mm + 40 mm 90º plug: 100 mm + 35 mm	

Table 26: Construction Depths

7.2.4 Connecting the Input and Output Circuits

Connect the input and output circuits using pluggable terminals located on the device's front plate.

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.

To attach the shielding to the shield contact plate with a clamp, it is useful to lead the shielded cables from below to controllers with analog inputs. 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 shield contact plate until it snaps into position.

7.2.5 Earthing and Shielding

7.2.5.1 Earthing the 24 VDC System Voltage

All devices of the HIMatrix family must be operated with power supply units that comply with the SELV (Safety Extra Low Voltage) or PELV (Protective Extra Low Voltage) requirements. A functional earth is prescribed to improve the electromagnetic compatibility (EMC).

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

Unearthed Operation

Unearthed operation offers some advantages in terms of EMC behavior.

Some applications have specific requirements for the unearthed operation of controllers, e.g., in accordance with the VDE 0116 standard, earth fault monitoring is required for unearthed operation.

Earthed Operation

The earth must be designed in accordance with the standard and must have a separate earth contact through which no power-dependent interference current may flow. Only earthing of negative pole (L-) is permitted. Earthing of positive pole (L+) is not permitted since a potential earth fault on the transmitter line would bypass the affected transmitter.

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

7.2.5.2 Earthing Connections

All HIMatrix devices have labeled screws for earthing. The wire cross-section for the connection to the screw is 2.5 mm^2. The earth wires must be as short as possible.

Provided that the DIN rail is earthed in accordance with the standard, mounting the HIMatrix compact systems on the DIN rail already ensures a sufficient earth connection.

These measures ensure a reliable earth ground and compliance with the current EMC requirements for HIMatrix systems.

7.2.5.3 Shielding

Sensor or actuator wires for analog inputs and outputs used in HIMatrix systems with shrouds (F3 AIO, F35 and F60) must be laid as shielded cables. The shielding must be connected to the HIMatrix system and the sensor or actuator housing and earthed on one end to the HIMatrix system side to form a Faraday cage.

To earth the cable shielding, the F3 AIO 8/4 01, F35 and F60 have rails on the front that are electrically connected to the housing potential. A clamp is used to connect the cable shielding to the rail.

In all other devices, the shielding must be positioned in the controller housing, terminal box, control cabinet, etc.

i The shield clamp must not be used as a strain relief for the connected cable.

7.2.5.4 EMC Protection

Windows in the enclosure in which the HIMatrix system is installed are permitted.

Increased EMC interferences outside the standard limit values require appropriate measures.

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.

7.2.6 Connecting the Supply Voltage

Protect the controller externally with a 10 A time-lag fuse.

The operating voltage is connected using a detachable four-pole connector located on the module's front plate. The connector can accept wires of up to a maximum of 2.5 mm².

Connec-tion	Function
L+	Power supply L+ (24 VDC)
L+	Power supply L+ (24 VDC)
L-	Power supply L- (24 VDC ground)
L-	Power supply L- (24 VDC ground)

Table 27: Power Supply Connectors

The two clamp terminals L+/L+ and L-/L- of the device are internally bypassed and intended for use in a two-wire supply. If the clamp terminals are connected to other devices, the maximum current of 10 A must not be exceeded.

Check proper polarity, voltage and ripple prior to connecting the operating voltage 24 VDC.

NOTE

Damage of the device possible!

Do not exchange the terminals L+ and L-, or connect them to other terminals of the device!

In case of a false connection, a pre-fuse blows to prevent the device from being damaged.

7.3 Configuration with SILworX - CPU OS V7 and Higher

This chapter describes how to configure resources using SILworX for CPU OS V7 **and higher**.

7.3.1 Configuring the Resource

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

7.3.1.1 Resource Properties

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

Parameter	Description	Default value	Setting for safe operation
Name	Resource name		Arbitrary
System ID [SRS]	System ID of the resource 1...65 535 The value assigned to the system ID must differ to the default value, otherwise the project is not able to run!	60 000	Unique value within the controller network. This network includes all controllers that can potentially be interconnected
Safety Time [ms]	Safety time in milliseconds 20...22 500 ms	600 ms/ 400 ms 1)	Application-specific
Watchdog Time [ms]	Watchdog time in milliseconds: 4...5000 ms for F*03 devices and modules; 8…5000 ms for standard devices and modules	200 ms/ 100 ms 1)	Application-specific
Target Cycle Time [ms]	Targeted or maximum cycle time, see *Target Cycle Time Mode*, 0...7500 ms. The maximum target cycle time value may not exceed the *watchdog time* - minimum watchdog time; otherwise it is rejected by the PES. If the default value 0 ms is set, the target cycle time is not taken into account.	0 ms	Application-specific
Target Cycle Time Mode	Use of *Target Cycle Time [ms]* see Table 29. With F*03 devices/modules, all the values can be used; with standard devices/modules, only *fixed* values!	Fixed-tolerant	Application-specific
Multitasking Mode	Only applicable with F*03 devices/modules! 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: During the execution time not needed by the user programs, the processor waits for the time to expire, thus increasing the cycle.	Mode 1	Application-specific

Parameter	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	60 ms	Application-specific
Max. Duration of Configuration Connections [ms]	Only applicable with F*03 devices/modules! It defines how much time within a CPU cycle is available for process data communication, 2...3500 ms	6 ms	Application-specific
Maximum System Bus Latency [µs]	Not applicable for HIMatrix controllers!	0 µs	-
Allow Online Settings	ON: All the switches/parameters listed below OFF can be changed online using the PADT. OFF: These parameters may **not** be changed online ▪ *System ID* ▪ *Autostart* ▪ *Global Forcing Allowed* ▪ *Global Force Timeout Reaction* ▪ *Load Allowed* ▪ *Reload Allowed* ▪ *Start Allowed* These parameters may be changed online if *Reload Allowed* is set to ON. ▪ *Watchdog Time* (for the resource) ▪ *Safety Time* ▪ *Target Cycle Time* ▪ *Target Cycle Time Mode* If *Reload Allowed* is set to OFF, they are not changeable online. i *Allow Online Settings* can only be set to ON if the PES is stopped.	ON	OFF is recommended
Autostart	ON: If the processor module is connected to the supply voltage, the user program starts automatically OFF: The user program does not start automatically after connecting the supply voltage.	OFF	Application-specific
Start Allowed	ON: A cold start or warm start permitted with the PADT in RUN or STOP OFF: Start not allowed	ON	Application-specific
Load Allowed	ON: Configuration download is allowed OFF: Configuration download is not allowed	ON	Application-specific
Reload Allowed	Only applicable with F*03 devices/modules! ON: Configuration reload is allowed OFF: Configuration reload is not allowed. A running reload process is not aborted when switching to OFF	ON	Application-specific
Global Forcing Allowed	ON: Global forcing permitted for this resource OFF: Global forcing not permitted for this resource	ON	Application-specific
Global Force Timeout Reaction	Specifies how the resource should behave when the global force timeout has expired: ▪ Stop Forcing ▪ Stop the Resource	Stop Forcing	Application-specific

Minimum Configuration Version	With this setting, code compatible to previous or newer CPU operating system versions in accordance with the project requirements may be generated.		SILworX V5 with new projects	Application-specific
	SILworX V2	The code is generated as in SILworX V2. With this setting, the use of the code on standard devices and modules is supported for CPU operating system V7.		
	SILworX V3	Not applicable for HIMatrix controllers!		
	SILworX V4	The generated code is compatible to the CPU operating system V8.		
	SILworX V5	Corresponds to *SILworX V4.* This setting ensures the compatibility with future versions.		
safeethernet CRC	SILworX V2.36.0	The CRC for safe**ethernet** is created as in SILworX V2.36.0. This setting is required for exchanging data with resources planned with SILworX V2.36 or previous versions.	Current Version	Application-specific
	Current Version	The CRC for safe**ethernet** is created with the current algorithm.		
1) First value applies to controllers, second value applies to remote I/Os.				

Table 28: System Parameters of the Resource - CPU OS V7 and Higher

The following table describes the effect of *Target Cycle Time Mode*.

Target Cycle Time Mode	Effect on user programs	Effect on reload of processor modules
Fixed	The PES maintains the target cycle time and extends the cycle if necessary. If the processing time of the user programs exceeds the target cycle time, the cycle duration is increased.	Reload is not processed if the target cycle time is not sufficient.
Fixed-tolerant	The same as *Fixed*.	At most, the duration of every fourth cycle is increased to allow reload.
Dynamic-tolerant	The same as *Dynamic*.	At most, the duration of every fourth cycle is increased to allow reload.
Dynamic	HIMatrix maintains the target cycle time as well as possible and also executes the cycle as quickly as possible.	Reload is not processed if the target cycle time is not sufficient.

Table 29: Effect of Target Cycle Time Mode

Notes on the *Minimum Configuration Version* Parameter:

- In a new project, the latest *Minimum Configuration Version* is selected. Verify that this setting is in accordance with the hardware in use, e.g., in HIMatrix standard devices, *Minimum Configuration Version* must be set to *SILworX V2.*
- In a project converted from a previous SILworX version, the value for *Minimum Configuration Version* remains the value set in the previous version. This ensures that the configuration CRC resulting from the code generation is the same as in the previous version and the generated configuration is compatible with the operating systems in the hardware.

  For this reason, the value of *Minimum Configuration Version* should not be changed for converted projects.
- If features only available in higher configuration versions are used in the project, SILworX automatically generates a higher configuration version than the preset *Minimum Configuration Version*. This is indicated by SILworX at the end of the code generation. The hardware denies loading a higher configuration version than that matching its operating system.

For help, compare the details provided by the version comparator with the module data overview.

- If a *Minimum Configuration Version* of *SILworX V4* or higher is set for a resource, the *Code Generation Compatibility* parameter must be set to *SILworX V4* in every user program (see below).

7.3.1.2 Parameters for the Remote I/Os

The following system parameters are available for remote I/Os:

Parameter	Description	Default value	Setting for safe operation
Name	Remote I/O Name		Arbitrary
Rack ID	Each individual remote I/O within a resource must have a unique rack ID. 200…1023	200	Unique value within the resource.
Safety Time [ms]	Safety time in milliseconds 20...22 500 ms	200 ms	Application-specific
Watchdog Time [ms]	Watchdog time in milliseconds 8...5000 ms	100 ms	Application-specific
Max. Com.Time Slice [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
Timeout [ms]	Monitoring time for command communication 600…60 000 ms *Timeout [ms]* >= 2 ∗ *resend time [ms]* A connection loss is detected upon expiration of the defined time. In the worst case scenario, the status indicators in the Control Panel of the parent resource are refreshed upon expiration of the defined timeout. *Timeout* applies for both the remote I/O and its parent resource.	20 000	Application-specific
Resend Time [ms]	Time interval after which a message is resent after the previous message was not acknowledged by the communication partner. 300…30 000 ms Resends increase availability and compensate disturbances in the network. HIMA recommends using values equal to or greater than the default value to ensure that the network is only	5000	Application-specific
Alive Interval [ms]	An acknowledgment for the received message is sent to the communication partner no later than upon expiration of the Alive Interval. 250…29 950 ms *Alive Interval [ms]* <= *Resend Time [ms]* - 50 ms. HIMA recommends using values equal to or greater than the default value to ensure that the network is only loaded if	2500	Application-specific

Table 30: System Parameters of the Remote I/O - CPU OS V7 and Higher

7.3.1.3 Hardware System Variables for Setting the Parameters

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.

Variable	Function	Default setting	Setting for safe operation
Force Deactivation	Used to prevent forcing and to stop it immediately	FALSE	Application-specific
Spare 2...Spare 16	No function	-	-
Emergency Stop 1... Emergency Stop 4	Emergency stop switch to shut down 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 F60! It controls the corresponding relay contacts, if existing.	FALSE	Application-specific
Reload Deactivation	Only applicable with F*03! It prevents the controller from being by performing a reload.	FALSE	Application-specific
User LED 1... User LED 2	Only applicable with F*03! It controls the corresponding LED, if existing.	FALSE	Application-specific

Table 31: Hardware System Variables - CPU OS V7 and Higher

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

7.3.1.4 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 Field Errors	Number of current I/O errors	UDINT
Number of Field Errors - Historic Count	Counted number of I/O errors (counter resettable)	UDINT
Number of Field Warnings	Number of current I/O warnings	UDINT
Number of Field Warnings - 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	Counted number of system warnings (counter resettable)	UDINT

Variable	Description	Data Type
Count		
Autostart	ON: When the processor system is connected to the supply voltage, it automatically starts the user program	BOOL
	OFF: When the processor system is connected to the supply voltage, it enters the STOP state	
OS Major	Operating system version contained in processor system	UINT
OS Minor		UINT
CRC	Resource configuration checksum	UDINT
Date/time [ms portion]	System date and time in s and ms since 1970-01-01	UDINT
Date/time [s portion]		UDINT
Force Deactivation	ON: Forcing is deactivated.	BOOL
	OFF: Forcing is possible.	
Forcing Active	ON: Global or local forcing is active.	BOOL
	OFF: Global and local forcing are not active.	
Force Switch State	State of the force switch	UDINT
	0xFFFFFFFE No force switch set	
	0xFFFFFFFF At least one force switch set	
Global Forcing Started	ON: Global forcing is active.	BOOL
	OFF: Global forcing is not active.	
Spare 0...Spare 16	Reserved	USINT
Spare 17		BOOL
Last Field Warning [ms]	Date and time of the last I/O warning in s and ms since 1970-01-01	UDINT
Last Field Warning [s]		UDINT
Last Communication Warning [ms]	Date and time of the last communication warning in s and ms since 1970-01-01	UDINT
Last Communication 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 Field Error [ms]	Date and time of the last I/O error in s and ms since 1970-01-01	UDINT
Last Field 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	BYTE
	0x01 Fan defective	
	0xFF Not available	
Allow Online Settings	Specifies if the online settings of the enable switch are allowed:	BOOL
	ON: The subordinate enable switches can be changed online.	

<table>
<tr><th>Variable</th><th colspan="2">Description</th><th>Data Type</th></tr>
<tr><td></td><td>OFF:</td><td>The subordinate enable switches cannot be changed online.</td><td></td></tr>
<tr><td rowspan="2">Read-only in RUN</td><td>ON:</td><td>The operator actions: Stop, Start, Download are locked.</td><td rowspan="2">BOOL</td></tr>
<tr><td>OFF:</td><td>The operator actions: Stop, Start, Download are not locked.</td></tr>
<tr><td rowspan="3">Reload Release</td><td colspan="2">With F*03 devices only!</td><td rowspan="3">BOOL</td></tr>
<tr><td>ON:</td><td>Controller can be loaded by performing a reload</td></tr>
<tr><td>OFF:</td><td>Controller cannot be loaded by performing a reload</td></tr>
<tr><td rowspan="3">Reload Deactivation</td><td colspan="2">With F*03 devices only!</td><td rowspan="3">BOOL</td></tr>
<tr><td>ON:</td><td>Loading by performing a reload is locked</td></tr>
<tr><td>OFF:</td><td>Loading by performing a reload is possible</td></tr>
<tr><td>Reload Cycle</td><td colspan="2">With F*03 devices only!
TRUE during the first cycle after a reload, otherwise FALSE</td><td>BOOL</td></tr>
<tr><td>CPU Safety Time [ms]</td><td colspan="2">Safety time set for the controller in ms</td><td>UDINT</td></tr>
<tr><td rowspan="2">Start Allowed</td><td>ON:</td><td>Start of processor system using PADT is allowed</td><td rowspan="2">BOOL</td></tr>
<tr><td>OFF:</td><td>Start of processor system using PADT is not allowed</td></tr>
<tr><td>Start Cycle</td><td colspan="2">ON during the first cycle after starting, otherwise OFF</td><td>BOOL</td></tr>
<tr><td>Power Supply State</td><td colspan="2">Bit-coded state of the power supply units.
Compact controllers and remote I/Os:
<table>
<tr><th>Value</th><th>State</th></tr>
<tr><td>0x00</td><td>Normal</td></tr>
<tr><td>0x01</td><td>Undervoltage with 24 V supply voltage</td></tr>
<tr><td>0x02</td><td>(Low voltage with battery) not used</td></tr>
<tr><td>0x04</td><td>Low voltage with internally generated 5 V voltage</td></tr>
<tr><td>0x08</td><td>Low voltage with internally generated 3.3 V voltage</td></tr>
<tr><td>0x10</td><td>Overvoltage with internally generated 3.3 V voltage</td></tr>
</table>
F60 modular controller:
<table>
<tr><th>Value</th><th>State</th></tr>
<tr><td>0x00</td><td>Normal</td></tr>
<tr><td>0x01</td><td>Error with 24 V supply voltage</td></tr>
<tr><td>0x02</td><td>Battery failure</td></tr>
<tr><td>0x04</td><td>Error with 5 V voltage of power supply</td></tr>
<tr><td>0x08</td><td>Error with 3.3 V voltage of power supply</td></tr>
<tr><td>0x10</td><td>Undervoltage with 5 V voltage</td></tr>
<tr><td>0x20</td><td>Overvoltage with 5 V voltage</td></tr>
<tr><td>0x40</td><td>Undervoltage with 3.3 V voltage</td></tr>
<tr><td>0x80</td><td>Overvoltage with 3.3 V voltage</td></tr>
</table></td><td>BYTE</td></tr>
<tr><td>System ID [SRS]</td><td colspan="2">System ID of the controller, 1...65 535</td><td>UINT</td></tr>
<tr><td>Systemtick HIGH</td><td colspan="2" rowspan="2">Revolving millisecond counter (64-bit)</td><td>UDINT</td></tr>
<tr><td>Systemtick LOW</td><td>UDINT</td></tr>
</table>

<table>
<tr><th>Variable</th><th>Description</th><th>Data Type</th></tr>
<tr><td>Temperature state</td><td>Bit-coded temperature state of the processor system<table><tr><th>Value</th><th>State</th></tr><tr><td>0x00</td><td>Normal temperature</td></tr><tr><td>0x01</td><td>Temperature threshold 1 exceeded</td></tr><tr><td>0x03</td><td>Temperature threshold 2 exceeded</td></tr><tr><td>0xFF</td><td>Not available</td></tr></table></td><td>BYTE</td></tr>
<tr><td>Remaining Global Force Duration [ms]</td><td>Time in ms until the time limit set for global forcing expires.</td><td>DINT</td></tr>
<tr><td>Watchdog Time [ms]</td><td>Maximum permissible duration of a RUN cycle in ms.</td><td>UDINT</td></tr>
<tr><td>Cycle Time, last [ms]</td><td>Current cycle time in ms</td><td>UDINT</td></tr>
<tr><td>Cycle Time, max [ms]</td><td>Maximum cycle time in ms</td><td>UDINT</td></tr>
<tr><td>Cycle Time, min [ms]</td><td>Minimum cycle time in ms</td><td>UDINT</td></tr>
<tr><td>Cycle Time, average [ms]</td><td>Average cycle time in ms</td><td>UDINT</td></tr>
</table>

Table 32: Hardware System Variables for Reading the Parameters

7.3.1.5 Rack System Parameter Settings

The rack system parameters can be set in the Rack's online view.

Parameter	Description	Default value
Type	Spare, not changeable	-
Name	Rack name for the controller, text	HIMatrix F...Rack
Rack ID	Not changeable	0
Temperature Monitoring	With F*03 devices only! Specifies the temperature threshold that, if exceeded, causes a warning message to be created: ▪ Warning at temperature thresholds 1 and 2 ▪ Warning at temperature threshold 2 only ▪ Warning at temperature threshold 1 only ▪ No warning at temperature thresholds 1 and 2	Warning at temperature thresholds 1 and 2

Table 33: System Parameters of the Rack

7.3.2 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 controllers and remote I/Os for more details.

7.3.3 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
Program ID	ID for identifying the program when displayed in SILworX, 0…4 294 967 295. If *Code Generation Compatibility* is set to *SILworX V2,* only the value 1 is permitted. (These settings are required for standard devices and modules).	0	Application-specific
Priority	Only applicable with F*03! Priority of the user program: 0...31, if multitasking is used	0	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 F*03!	1	Application-specific
Max. Duration for Each Cycle [µs]	Maximum time in each processor module cycle for executing the user program: 1...4 294 967 295 µs. Set to 0: No limitation. A value other than 0 µs is only allowed for HIMatrix controllers F*03!	0 µs	0 µs
Watchdog Time [ms] (calculated)	Monitoring time of the user program, calculated from the *Program's Maximum Number of Cycles* and the watchdog time of the resource Not changeable! For HIMatrix F*03 systems with counter inputs, ensure that the user program's watchdog time is less than or equal to 5000 ms.		-
Classification	Classification of the user program: Safety-related or Standard (for documentation only).	Safety-related	Application-specific
Allow Online Settings	It enables changes of other user program switches during operation. It only applies if the *Allowed Online Settings* switch for the resource is set to ON!	ON	-
Autostart	Enabled type of Autostart: Cold Start, Warm Start, Off	Warm start	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
Test Mode Allowed	ON The test mode is permitted for the user program. OFF The test mode is not permitted for the user program.	OFF	Application-specific
Reload Allowed	ON: User program reload is permitted OFF: User program reload is not permitted	ON	Application-specific
Local Forcing Allowed	ON: Forcing permitted at program level OFF: Forcing not permitted at program level	OFF	OFF is recommended
Local Force Timeout Reaction	Behavior of the user program after the forcing time has expired: ▪ Stop Forcing Only. ▪ Stop Program.	Stop Forcing Only.	Application-specific
Code Generation Compatibility	SILworX V4: Code generation is compatible with SILworX V4. SILworX V3: Not applicable for HIMatrix controllers! SILworX V2: Code generation is compatible with SILworX V2.	SILworX V4 with new projects	SILworX V2 with CPU OS V7 SILworX V4 with CPU OS V8

Table 34: System Parameters of the User Program - CPU OS V7 and Higher

Notes specific to the *Code Generation Compatibility* Parameter*:*

- In a new project, SILworX selects the latest value for the *Code Generation Compatibility* parameter. This ensures that the current, enhanced features are activated and the latest hardware and operating system versions are supported. Verify that this setting is in accordance with the hardware in use, e.g., in HIMatrix standard devices, *Minimum Configuration Version* must be set to *SILworX V2.*
- In a project converted from a previous SILworX version, the value for *Code Generation Compatibility* remains the value set in the previous version. This ensures that the configuration CRC resulting from the code generation is the same as in the previous version and the generated configuration is compatible with the operating systems in the hardware.

  For this reason, the value of *Code Generation Compatibility* should not be changed for converted projects.
- If a *Minimum Configuration Version* of *SILworX V4* or higher is set for a resource (see above), the *Code Generation Compatibility* parameter must be set to *SILworX V4* in every user program. If *SILworX V2* is required, the *Minimum Configuration Version* resource parameter must be set to *SILworX V2*.

7.3.4 Configuring the Inputs and Outputs

In the Hardware Editor, the inputs and outputs are configured by connecting 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 Input

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 provided to the user program. If analog inputs of type FS1000 are used, the range of values is 0...1000, with analog inputs of type FS2000, the range of values is 0...2000.

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.

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 Inputs

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

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.
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 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.

7.3.5 Configure Line Control

The pulse delay for line control is the time between setting the pulsed outputs to FALSE and the latest possible reading of the signal on the corresponding input.

The default value is set to 400 µs. This value might need be increased if longer wires are used. The maximum value is 2000 µs.

The minimum time for reading all inputs results in
pulse delay x number of pulses.

The pulsed outputs are usually set to TRUE and change to FALSE in succession for the duration of the pulse delay once per cycle.

7.3.5.1 Required Variables

The following parameters must be created as global variables in the SILworX Global Variable Editor:

Name	Type	Description	Initial Value	Remark
Sum_Pulse	USINT	Number of pulsed outputs	4	1...8, as required
Board_POS_Pulse	UDINT	Module slot with pulsed outputs	2	With compact devices, the DOs are used in slot 1, 2 or 3, see Table 37. With the F60, the slot (3...8) is given.
Pulse_delay	UINT	Pulse delay	400	Value in µs Maximum value: 2000 µs F20: Pulse delay must be ≥ 500 µs. Refer to the F20 manual.
T1 T2 ... T8	USINT USINT ... USINT	Pulse 1 Pulse 2 ... Pulse 8	1 2 ... 8	Pulse 1...8, as required, must match the number of pulsed outputs.
Pulse_ON	BOOL	Initialization value for pulsed outputs	TRUE	Activation of pulsed outputs

Table 35: Parameters for Line Control

The signal can be named freely; the names used in this manual are examples. All parameters have the *Const* attribute.

The following table specifies the switch variables used in the example:

Name	Type	Description	Remark
S1_1_pulsed S1_2_pulsed	BOOL BOOL	Value Value	First and second contact of switch 1
S2_1_pulsed S2_2_pulsed	BOOL BOOL	Value Value	First and second contact of switch 2
FC_S1_1_pulsed FC_S1_2_pulsed	BYTE BYTE	Error code Error code	Error codes for first and second contact of switch 1
FC_S2_1_pulsed FC_S2_2_pulsed	BYTE BYTE	Error code Error code	Error codes for first and second contact of switch 2

Table 36: Switch Variables for Line Control

The following table specifies the slot numbers of modules with pulsed outputs when compact devices are used.

Device	*DI Pulse Slot* system parameter
F1 DI 16 01	1
F3 DIO 8/8 01	3
F3 DIO 16/8 01	3
F3 DIO 20/8 02	2
F20	3
F30	3
F31	3

Table 37: Module Slot with Pulsed Outputs

If the modular F60 system is used, the number of the slot in which the module with pulsed outputs is inserted, must be used (3…8).

7.3.5.2 Configuring Pulsed Outputs

The pulsed outputs must begin in SILworX with channel 1 and reside in direct sequence, one after the other.

SILworX *Value [BOOL] ->*	Examples of permitted configurations ...				... of not permitted	
Channel no. 1	A1	Pulse_ON	Pulse_ON	Pulse_ON	A1	Pulse_ON
Channel no. 2	A2	Pulse_ON	Pulse_ON	Pulse_ON	Pulse_ON	Pulse_ON
Channel no. 3	A3	Pulse_ON	Pulse_ON	Pulse_ON	Pulse_ON	A3
Channel no. 4	A4	A4	Pulse_ON	Pulse_ON	Pulse_ON	Pulse_ON
Channel no. 5	A5	A5	A5	Pulse_ON	Pulse_ON	Pulse_ON
Channel no. 6	A6	A6	A6	Pulse_ON	A6	Pulse_ON
Channel no. 7	A7	A7	A7	A7	A7	A7
Channel no. 8	A8	A8	A8	A8	A8	A8

Table 38: Configuration of Pulsed Outputs

The corresponding inputs can be freely selected, i.e., two consecutive pulsed outputs need not be assigned to two adjacent inputs.

Restriction:
Two adjacent inputs may not be supplied from the same pulse to prevent crosstalk.

7.3.5.3 Configuration Example with SILworX

Fundamental Method for Assigning Variables

In SILworX, the global variables previously created in the Global Variable Editor are assigned to the individual hardware channels.

Assigning hardware channels global variables

1. Select *Hardware* in the structure tree of the project.
2. Right-click the input module and select *Detail View* from the context menu.
3. Change to the **DI XX: Channels** tab.
4. Drag the global variables onto the inputs to be used.
5. To assign the variables to the outputs, select the corresponding output module and proceed as described for the inputs.

The hardware channels are assigned global variables.

The following example is based on the list provided in Table 35 and the procedure described above.

Configuring Pulsed Outputs and Connecting them to the Inputs

The following table shows the connection of the system variables in the input module's Detail View to global variables:

Tab	System variable	Global Variable
Module	DI Number of Pulsed Outputs	Sum_Pulse
	DI Pulse Slot	Board_POS_Pulse
	DI Pulse Delay [µs]	Pulse_delay
DIxx: Channel	Pulsed output [USINT] -> consecutive channels from *Sum_Pulse* (4)	T1…T4

Table 39: Connection of the Global Variables to Output System Variables of the Input Module

Digital inputs (pulsed channels) may be arbitrarily connected to the pulsed outputs depending on the hardware configuration.

Connecting the Variables to the Inputs and Corresponding Error Codes

Each input channel value -> *Value [BOOL]* contained in the **DIxx: Channels** tab located in the input module's Detail View is allocated the corresponding error code -> *Error Code [BYTE]*. The error code must be evaluated in the user program.

The following table shows the connection of the system variables in the input module's Detail View to global variables:

System variable	Global Variable
-> *Value [BOOL]* of the corresponding channel	S1_1_Pulsed…S2_2_Pulsed (one variable for each channel)
-> *Error Code [BYTE]* of the corresponding channel	FC_S1_1_Pulsed…FC_S2_2_Pulsed (one variable for each channel)

Table 40: Connection of the Global Variables to Input System Variables of the Input Module

Activation of Pulsed Outputs

From the **DOxx: Channels** tab located in the output module's Detail View, connect the *Value [BOOL]* -> system variable of every fourth successive channels (=*Sum_Pulse*) with *Pulse_ON.*

The logical value of the *Pulse_ON* variable is TRUE. This results in pulsed outputs that are permanently activated and only set to FALSE for the duration of the pulse actuation.

7.3.6 Generating the Resource Configuration

With the following procedure, the code is generated twice and the resulting CRCs are compared with one another.

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** on the context menu.
   ☑ The *Code Generation <Resource Name>* dialog box appears.
3. Select **CRC Comparison** on the *Code Generation <Resource Name>* dialog box (default value).
4. In the Start Code Generation dialog box, click **OK**.
   ☑ An additional *Start Code Generation* appears, shows the progress of the two code generation processes and is closed again. The logbook contains one row informing about the code generation result and one row reporting the result of the CRC comparison.

A valid code is generated for the resource configuration.

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.

7.3.7 Configuring the System ID and the Connection Parameters

To 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** on the context menu.
   ☑ The *System Login* dialog box appears.
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 SRS, 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).

7.3.8 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 7.3.7.

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** on the context menu.
   ☑ The *System Login* dialog box appears.
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 cancel the password.
6. Select **Administrator** in the *Access Mode* field.
7. Click **Log-in.**

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

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

7.3.9 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 7.3.7.

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** on 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.

7.3.10 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 7.3.7.

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 to perform a download of the project.

To load 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.

7.3.11 Cleaning 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** on 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.

7.3.12 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** on 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.

7.4 User Management in SILworX - CPU OS V7 and Higher

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

7.4.1 User Management for SILworX Projects

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

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

Type	Description
Security Administrator (Sec Adm)	Security administrators can modify the user management scheme: setting up, deleting, changing the PADT user management scheme, and the 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 41: Authorization Types for the PADT User Management Scheme

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

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

7.4.2 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 user and its access rights are part of the project; they are defined with SILworX and loaded into the processor module.

The user management can be 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 after switching off the operating voltage.

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

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

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

Default User

The factory user settings apply if no user accounts were set up 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 Note that the default settings cannot be maintained if new user accounts are defined.

Parameters for User Accounts

To set up new user accounts, the following parameters must be set:

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 password 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: ▪ Read: Users may only read information but they cannot modify the controller. ▪ Read + Operator: Similar to *Read*, but users may also: Perform a download to load and start user programs Configure the processor modules as redundant Reset cycle time and fault or error statistics Set the system time, force, restart and reset modules Start system operation for processor modules. ▪ Read + Write: Similar to Read + Operator, but users may also: Create programs Translate programs Load programs into the controller Test programs ▪ Administrator: Similar to Read + Write, but users may also: 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 42: Parameters for User Accounts in the PES User Management Scheme

7.4.3 Setting up User Accounts

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

Observe the following points when setting up user accounts:

- Make sure that at least one user account is assigned with administrator rights. Define a password for the user account with administrator rights.
- After a user account was created in the user management, its password must be used to access and edit 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.

7.5 Configuration with SILworX - CPU OS V7 and Higher

This chapter describes how to configure communication using SILworX for processor operating systems V7 **and higher**.

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

- Ethernet/safeethernet.
- Standard protocols

  Refer to the SILworX communication manual (HI 801 101 E) for more information on how to configure the standard protocols.

7.5.1 Configuring the Ethernet Interfaces

Ethernet interfaces are configured in the Detail View of the communication module (COM). If the remote I/Os have no communication module, the Ethernet interfaces are configured in the Detail View of the processor module (CPU). 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.

In F*03 controllers, VLAN is available and allows one to configure the connection of the ports to CPU, COM and one another. VLAN is important for proper configuration of redundant safe**ethernet**.

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 *Speed* and *Flow Control* 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 43: Port Configuration Parameters - CPU OS and Higher

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 SILworX communication manual (HI 801 101 E) for more details about configuring safeethernet.

7.6 Configuring Alarms and Events in F*03 Devices

Event Definition

1. Define a global variable for each event. Generally use global variables that have already been defined for the program.
2. Below the resource, create a new Alarms&Events branch, if not existing.
3. Define events in the A&E Editor.

Drag global variables into the event window for Boolean or scalar events.

Define the details of the **events**, see the following 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:

<table>
<tr><th>Column</th><th colspan="2">Description</th><th>Range of values</th></tr>
<tr><td>Name</td><td colspan="2">Name for the event definition; it must be unique within the resource.</td><td>Text, max. 32 characters</td></tr>
<tr><td>Global Variable</td><td colspan="2">Name of the assigned global variable (added using a drag&drop operation)</td><td></td></tr>
<tr><td>Data Type</td><td colspan="2">Data type of the global variable; it cannot be modified.</td><td>BOOL</td></tr>
<tr><td rowspan="3">Event Source</td><td>CPU event</td><td>The processor module creates the timestamp. It creates all the events in each of its cycles.</td><td rowspan="3">CPU, Auto</td></tr>
<tr><td>Auto event</td><td>As CPU event</td></tr>
<tr><td colspan="2">Default value: Auto Event</td></tr>
<tr><td rowspan="3">Alarm when FALSE</td><td>Activated</td><td>If the global variable value changes from TRUE to FALSE, an event is triggered.</td><td rowspan="3">Checkbox activated, deactivated</td></tr>
<tr><td>Deactivated</td><td>If the global variable value changes from FALSE to TRUE, an event is triggered.</td></tr>
<tr><td colspan="2">Default value: Deactivated</td></tr>
<tr><td>Alarm Text</td><td colspan="2">Text specifying the alarm state</td><td>Text</td></tr>
<tr><td>Alarm Priority</td><td colspan="2">Priority of the alarm state
Default value: 500</td><td>0...1000</td></tr>
<tr><td rowspan="3">Alarm Acknowledgment Successful</td><td>Activated</td><td>The alarm state must be confirmed by the user (acknowledgement)</td><td rowspan="3">Checkbox activated, deactivated</td></tr>
<tr><td>Deactivated</td><td>The alarm state may not be confirmed by the user</td></tr>
<tr><td colspan="2">Default value: Deactivated</td></tr>
<tr><td>Return to Normal Text</td><td colspan="2">Text specifying the alarm state</td><td>Text</td></tr>
<tr><td>Return to Normal Severity</td><td colspan="2">Priority of the normal state
Default value: 500</td><td>0...1000</td></tr>
<tr><td>Return to Normal Ack Required</td><td colspan="2">The normal state must be confirmed by the user (acknowledgement)
Default value: Deactivated</td><td>Checkbox activated, deactivated</td></tr>
</table>

Table 44: Parameters for Boolean Events

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

<table>
<tr><th>Column</th><th colspan="2">Description</th><th>Range of values</th></tr>
<tr><td>Name</td><td colspan="2">Name for the event definition; it must be unique within the resource.</td><td>Text, max. 32 characters</td></tr>
<tr><td>Global Variable</td><td colspan="2">Name of the assigned global variable (added using a drag&drop operation)</td><td></td></tr>
<tr><td>Data Type</td><td colspan="2">Data type of the global variable; it cannot be modified.</td><td>Depending on the global variable type</td></tr>
<tr><td rowspan="3">Event Source</td><td>CPU event</td><td>The processor module creates the timestamp. It creates all the events in each of its cycles.</td><td rowspan="3">CPU, Auto</td></tr>
<tr><td>Auto event</td><td>As CPU event</td></tr>
<tr><td colspan="2">Default value: Auto Event</td></tr>
<tr><td>HH Alarm Text</td><td colspan="2">Text specifying the alarm state of the highest limit (HH).</td><td>Text</td></tr>
<tr><td>HH Alarm Value</td><td colspan="2">Highest limit (HH) triggering an event. Condition:
(HH Alarm Value - Hysteresis) > H Alarm Value or HH Alarm Value = H Alarm Value</td><td>Depending on the global variable type</td></tr>
<tr><td>HH Alarm Priority</td><td colspan="2">Priority of the highest limit (HH); default value: 500</td><td>0...1000</td></tr>
<tr><td rowspan="3">HH Alarm Acknowledgment Required</td><td>Activated</td><td>The user must confirm that the highest limit (HH) has been exceeded (acknowledgment).</td><td rowspan="3">Checkbox activated, deactivated</td></tr>
<tr><td>Deactivated</td><td>The user may not confirm that the highest limit (HH) has been exceeded.</td></tr>
<tr><td colspan="2">Default value: Deactivated</td></tr>
<tr><td>H Alarm Text</td><td colspan="2">Text specifying the alarm state of the high limit (H).</td><td>Text</td></tr>
<tr><td>H Alarm Value</td><td colspan="2">High limit (H) triggering an event. Condition:
(H Alarm Value - Hysteresis) > (L Alarm Value + Hysteresis) or H Alarm Value = L Alarm Value</td><td>Depending on the global variable type</td></tr>
<tr><td>H Alarm Priority</td><td colspan="2">Priority of the high limit (H); default value: 500</td><td>0...1000</td></tr>
<tr><td rowspan="3">H Alarm Acknowledgment Required</td><td>Activated</td><td>The user must confirm that the high limit (H) has been exceeded (acknowledgment).</td><td rowspan="3">Checkbox activated, deactivated</td></tr>
<tr><td>Deactivated</td><td>The user may not confirm that the high limit (H) has been exceeded.</td></tr>
<tr><td colspan="2">Default value: Deactivated</td></tr>
<tr><td>Return to Normal Text</td><td colspan="2">Text specifying the normal state</td><td>Text</td></tr>
<tr><td>Return to Normal Severity</td><td colspan="2">Priority of the normal state; default value: 500</td><td>0...1000</td></tr>
<tr><td>Return to Normal Ack Required</td><td colspan="2">The normal state must be confirmed by the user (acknowledgement); default value: Deactivated</td><td>Checkbox activated, deactivated</td></tr>
<tr><td>L Alarm Text</td><td colspan="2">Text specifying the alarm state of the low limit (L).</td><td>Text</td></tr>
<tr><td>L Alarm Value</td><td colspan="2">Low limit (L) triggering an event. Condition:
(L Alarm Value + Hysteresis) < (H Alarm Value - Hysteresis) or L Alarm Value = H Alarm Value</td><td>Depending on the global variable type</td></tr>
<tr><td>L Alarm Priority</td><td colspan="2">Priority of low limit (L); default value: 500</td><td>0...1000</td></tr>
<tr><td rowspan="3">L Alarm Acknowledgment Required</td><td>Activated</td><td>The user must confirm that the low limit (L) has been exceeded (acknowledgment).</td><td rowspan="3">Checkbox activated, deactivated</td></tr>
<tr><td>Deactivated</td><td>The user may not confirm that the low limit (L) has been exceeded.</td></tr>
<tr><td colspan="2">Default value: Deactivated</td></tr>
<tr><td>LL Alarm Text</td><td colspan="2">Text specifying the alarm state of the lowest limit (LL).</td><td>Text</td></tr>
<tr><td>LL Alarm Value</td><td colspan="2">Lowest limit (LL) triggering an event. Condition:
(LL Alarm Value + Hysteresis) < (L Alarm Value) or
LL Alarm Value = L Alarm Value</td><td>Depending on the global variable type</td></tr>
<tr><td>LL Alarm Priority</td><td colspan="2">Priority of the lowest limit (LL); default value: 500</td><td>0...1000</td></tr>
</table>

Column	Description		Range of values
LL Alarm Acknowledgment Required	Activated	The user must confirm that the lowest limit (LL) has been exceeded (acknowledgment).	Checkbox activated, deactivated
	Deactivated	The user may not confirm that the lowest limit (LL) has been exceeded.	
	Default value: 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 45: Parameters for Scalar Events

NOTE

Faulty event recording due to improper parameter settings possible!

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

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

7.7 Configuring a Resource Using ELOP II Factory - CPU OS up to V7

This chapter describes the various configuration options using ELOP II Factory for processor operating systems up to V7.

7.7.1 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	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 ≤ (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 *Safety Time* and *Watchdog Time* 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	On

Parameter	Range	Description	Default value
		Off: Load not allowed	
Test Mode Allowed	On/Off	Test mode On: Test mode Allowed Off: Test mode not allowed	Off
Changing the variables in the OLT allowed	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 46: Resource Configuration Parameters - CPU OS up to V7

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

7.7.2 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 V6.12, major version: 6 OS V6, 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 V6.12, minor version: 12 OS V6, 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 V6, 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[2)]	[BYTE] 0x00 0x01	R	 Normal (fan ON) Fan defective [not safe]

Signal	[Data type], Unit, value	R/W	Description
Power supply State	[BYTE] 0x00 0x01 0x02 0x04 0x08 0x10	R	 Normal Undervoltage 24 V [not safe] Battery undervoltage [not safe] Undervoltage 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 *[not safe]* characteristic may only be combined with signals defined as *[safe]* to trigger a safety shut-down. 2) Currently available with the F20 controller only, with all other systems: 0xFF = status not available			

Table 47: General System Signals and Parameters - CPU OS up to V7

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: Compatibility with previous controller versions is ensured. BIG: Compatibility with future controller versions.	SMALL

Table 48: User program Parameters - up to CPU OS V7

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**. 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.

7.7.3 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 module or 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.

7.7.4 Configure Line Control

The pulse delay for line control is the time between setting the pulsed outputs to FALSE and the latest possible reading of the signal on the corresponding input.

The default value is set to 400 µs. This value might need be increased if longer wires are used. The maximum value is 2000 µs.

The minimum time for reading all inputs results in
pulse delay x number of pulses.

The pulsed outputs are usually set to TRUE and change to FALSE in succession for the duration of the pulse delay once per cycle.

7.7.4.1 Required Signals

The following parameters must be created as signals in the Signal Editor of the ELOP II Factory Hardware Management:

Name	Type	Description	Initial Value	Remark
Sum_Pulse	USINT	Number of pulsed outputs	4	1...8, as required
Board_POS_-Pulse	UDINT	Module slot with pulsed outputs	2	With compact devices, the DOs are used in slot 1, 2 or 3, see Table 37. With the F60, the slot (3...8) is given.
Pulse_delay	UINT	Pulse delay	400	Value in µs Maximum value: 2000 µs F20: Pulse delay must be ≥ 500 µs. Refer to the F20 manual.
T1 T2 ... T8	USINT USINT ... USINT	Pulse 1 Pulse 2 ... Pulse 8	1 2 ... 8	Pulse 1...8, as required, must match the number of pulsed outputs.
Pulse_ON	BOOL	Initialization value for pulsed outputs	TRUE	Activation of pulsed outputs

Table 49: Signals for Line Control

The signal can be named freely; the names used in this manual are examples. All signals have the *Const* attribute.

The following table specifies the switch signals used in the example:

Name	Type	Description	Remark
S1_1_pulsed S1_2_pulsed	BOOL BOOL	Value Value	First and second contact of switch 1
S2_1_pulsed S2_2_pulsed	BOOL BOOL	Value Value	First and second contact of switch 2
FC_S1_1_pulsed FC_S1_2_pulsed	BYTE BYTE	Error code Error code	Error codes for first and second contact of switch 1
FC_S2_1_pulsed FC_S2_2_pulsed	BYTE BYTE	Error code Error code	Error codes for first and second contact of switch 2

Table 50: Switch Signals for Line Control

The following table specifies the slot numbers of modules with pulsed outputs when compact devices are used.

Device	System signal *DI Pulse Slot.*
F1 DI 16 01	1
F3 DIO 8/8 01	3
F3 DIO 16/8 01	3
F3 DIO 20/8 02	2
F20	2
F30	2
F31	2

Table 51: Module Slot with Pulsed Outputs

If the modular F60 system is used, the number of the slot in which the module with pulsed outputs is inserted, must be used (3…8).

7.7.4.2 Configuring Pulsed Outputs

The pulsed outputs must begin with *DO[01].Value* and reside in direct sequence, one after the other:

ELOP II Factory outputs	Examples of permitted configurations ...				... of not permitted	
DO[01].Value	A1	Pulse_ON	Pulse_ON	Pulse_ON	A1	Pulse_ON
DO[02].Value	A2	Pulse_ON	Pulse_ON	Pulse_ON	Pulse_ON	Pulse_ON
DO[03].Value	A3	Pulse_ON	Pulse_ON	Pulse_ON	Pulse_ON	A3
DO[04].Value	A4	A4	Pulse_ON	Pulse_ON	Pulse_ON	Pulse_ON
DO[05].Value	A5	A5	A5	Pulse_ON	Pulse_ON	Pulse_ON
DO[06].Value	A6	A6	A6	Pulse_ON	A6	Pulse_ON
DO[07].Value	A7	A7	A7	A7	A7	A7
DO[08].Value	A8	A8	A8	A8	A8	A8

Table 52: Configuration of Pulsed Outputs in ELOP II Factory

The corresponding inputs can be freely selected, i.e., two consecutive pulsed outputs need not be assigned to two adjacent inputs.

Restriction:
Two adjacent inputs may not be supplied from the same pulse to prevent crosstalk.

7.7.4.3 Configuration Example with ELOP II Factory

Basic Method for Assigning Signals

If ELOP II Factory is used, the signals previously defined in the Signal Editor (Hardware Management) are assigned to the individual hardware channels (inputs and outputs).

To connect signals to inputs and outputs

1. Select the Signals menu option to open Signal Editor of the ELOP II Factory's Hardware Management.
2. Right-click the HIMatrix I/O module and select **Connect Signals** from the context menu.
   ☑ A dialog box for connecting the signals in the Signal Editor to the available hardware channels appears and contains the tabs **Inputs** and **Outputs.**
3. If required, select the **Inputs** tab.
4. Position the two dialog boxes adjacently to get a better overview.
5. Drag the signals onto the inputs located in the *Signal Connections* pane.
6. To connect the signals to the outputs, select the **Outputs** tab and proceed as described for the inputs.

The signals are connected to the inputs and outputs.

The following example is based on the list provided in Table 35 and the procedure described above.

Configuring Pulsed Outputs and Connecting them to the Inputs

The following table shows the connection of the input module's output signals to signals:

System signal (output signal)	Signal
DI Number of Pulsed Outputs	Sum_Pulse
DI Slot Pulsed Outputs	Board_POS_Pulse
DI Pulse Delay [µs]	Pulse_delay
DI[xx].Pulse Output of 4 (=*Sum_Pulse)* consecutive output signals	T1…T4

Table 53: Connecting Signals to the Input Module's Output Signals

Digital inputs (pulsed channels) may be arbitrarily connected to the pulsed outputs depending on the hardware configuration.

Connecting the Signals to the Inputs and Corresponding Error Codes

For each useful signal *DI[xx].Value,* the relevant error code must also be evaluated

The following table shows the signals to be connected for each of the input channels to be monitored:

System signals	Signals
DI[xx].Value of the corresponding channel xx	*S1_1_pulsed…S2_2_pulsed* (one of the signals per channel)
DI[xx].Error Code of the corresponding channel xx	*FC_S1_1_pulsed…FC_S2_2_pulsed* (one of the signals per channel)

Table 54: Connecting Signals to the Input Module's Input Signals

Activation of Pulsed Outputs

For pulsed outputs of the output module, connect the *DO[xx].Value* output signals of the corresponding consecutive channels to the *Pulse_ON* signal.

The logical value of the signal *Pulse_ON* is TRUE. This results in pulsed outputs that are permanently activated and only set to FALSE for the duration of the pulse actuation.

7.7.5 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.

7.7.6 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).

7.7.7 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, 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 7.7.6.

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 V10.42 and Higher

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 V6.0 and Higher

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.

7.7.8 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 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/VALID 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 55: Sub-States Associated with STOP - up to CPU OS V7

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

7.7.9 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 is no longer accessible, the connection parameters for the project must be reset in the controller, see Chapter 7.7.6. 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 to perform 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** to open the Control Panel.
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.

7.7.10 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 from 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 systems.

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.

7.8 Configuring Communication with ELOP II Factory - up to CPU OS V7

This chapter describes how to configure communication using ELOP II Factory for processor operating systems **up to** V7.

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

- Ethernet/safe**ethernet**, 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)
  - EtherNet/IP in ELOP II Factory Online Help

7.8.1 Configuring the Ethernet Interfaces

COM OS up to V8.32:

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 Base T//100 Base Tx 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
Speed Mode	Autoneg	Autoneg	10 Mbit/s	100 Mbit/s
Flow Control Mode	Autoneg	Half Duplex	Half Duplex	Half Duplex

Table 56: Permissible Communication Settings for External Devices - CPU OS up to V7

The following network settings are not allowed:

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

Table 57: Impermissible Communication Settings for External Devices - CPU OS up to V7

COM OS V8.32 and Higher and ELOP II Hardware Management V7.56.10 and Higher:

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 11.

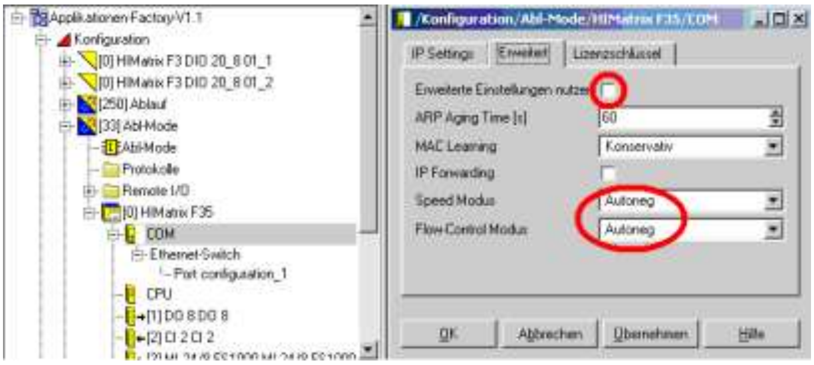


Figure 11: Communication System Properties - CPU OS up to V7

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.

- V8.32 of the communication operating system and
- V7.56.10 of ELOP II Hardware Management

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

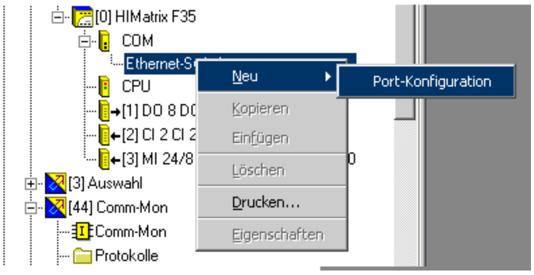


Figure 12: Creating a Port Configuration - CPU OS up to V7

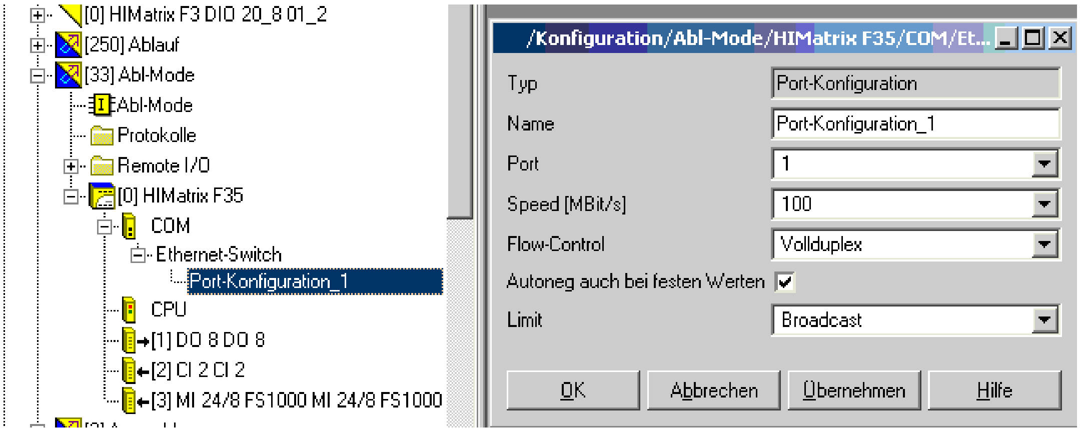


Figure 13: Parameters of a Port Configuration - CPU OS up to V7

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 *Speed* and *Flow Control* 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 58: Parameters of a Port Configuration - CPU OS up to V7

Click **Apply** to adopt the parameters in 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.

7.8.2 System Signals of safe**ethernet** Communication

The user program can use system signals to read the status of the safe**ethernet** 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 safe**ethernet** 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 to 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 59: System Signals of a safe**ethernet** Connection for Reading the Status - CPU OS up to V7

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 60: System Signal of a safe**ethernet** Connection for Setting the Connection Control - CPU OS up to V7

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 61: The *Connection Control* Parameter - CPU OS up to V7

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

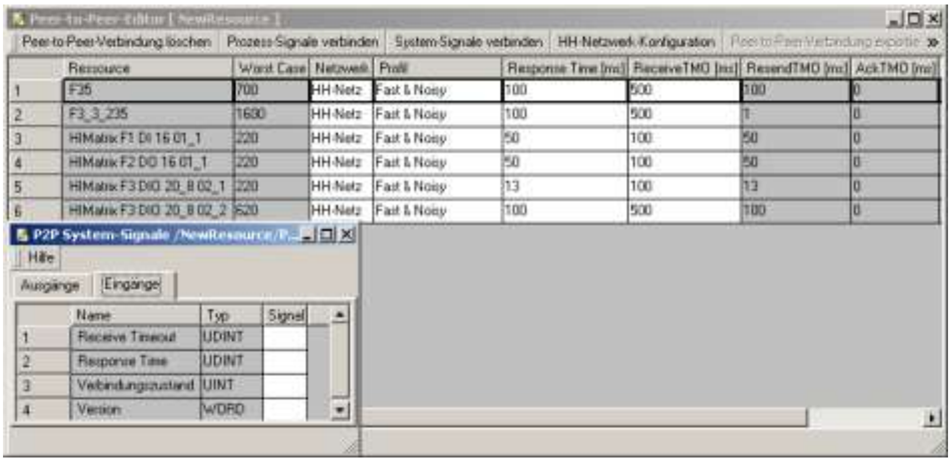


Figure 14: Peer-to-Peer Parameters in the **Inputs** Tab - CPU OS up to V7

4. 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

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 **Outputs** tab

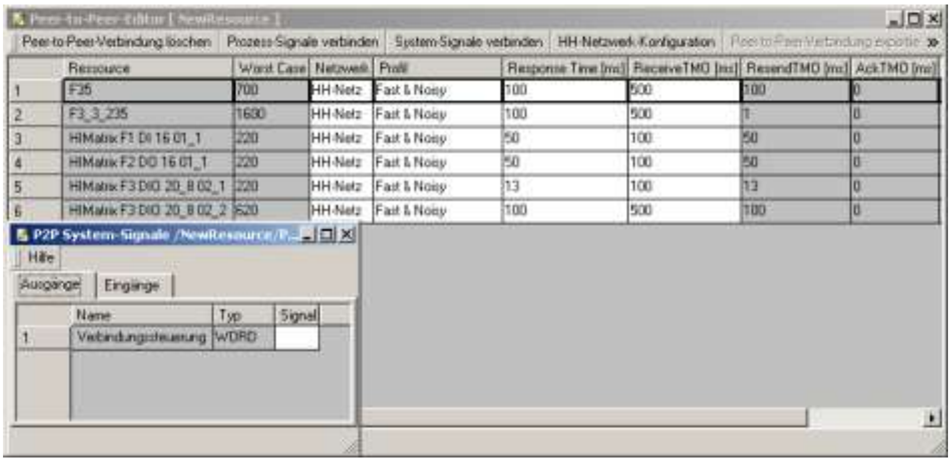


Figure 15: *Connection Control* System Signal in the **Outputs** Tab - CPU OS up to V7

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

7.8.3 Configuring the safe**ethernet** Connection

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

1. Profile - see below
2. Response Time

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

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

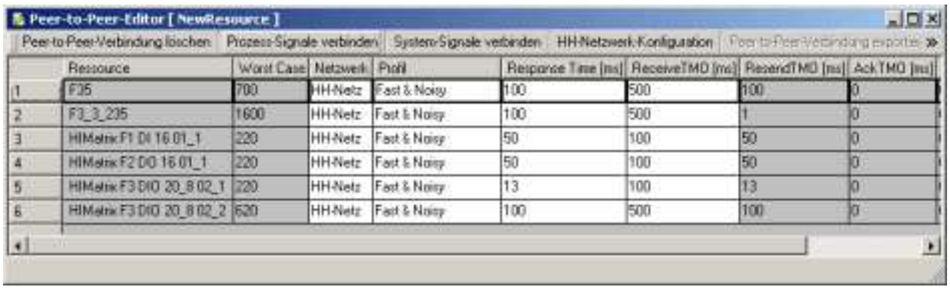


Figure 16: Setting the Parameters in the P2P Editor - up to CPU OS V7

The parameters specified above determine the data throughput and the fault and collision tolerance of the safe**ethernet** 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.

7.8.4 Configuring the Signals for safe**ethernet** Communication

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

To configure the signals for safe**ethernet 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.

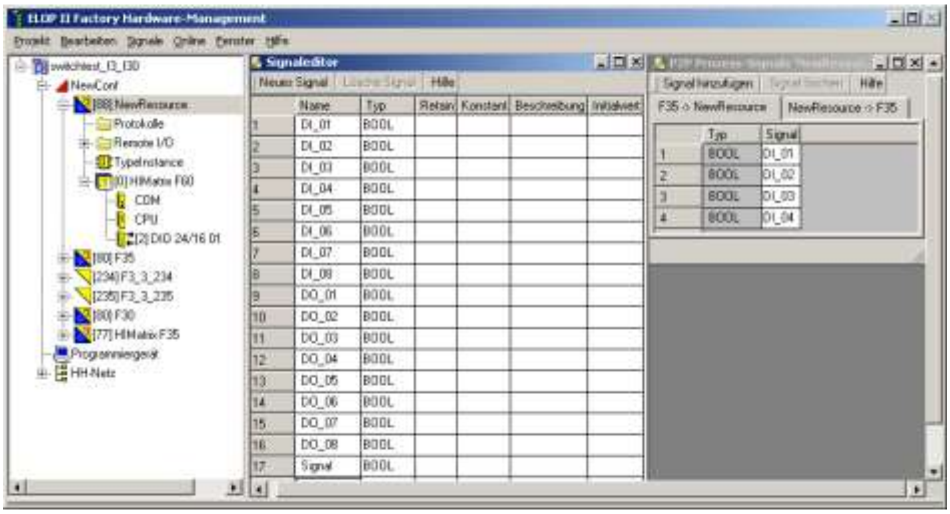


Figure 17: Assigning Process Signals per Drag&Drop - CPU OS up to V7

i By sending a signal value from a controller to another controller (PES_1 -> PES_2), the value is available in the second controller PES_2. To be able to use the value, use the same signals in the PES_1 and PES_2 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.

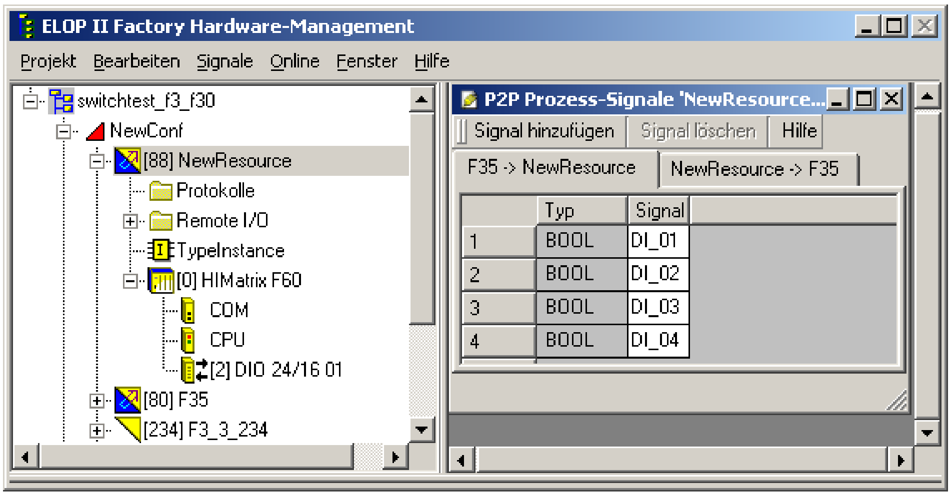


Figure 18: Example of Process Signals - CPU OS up to V7

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.

7.9 Handling the User Program

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

7.9.1 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.

7.9.2 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.

7.9.3 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.

7.9.4 Stopping the Program

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

7.9.5 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).

WARNING

Property damage or physical injury possible due to actuators in unsafe state!

Do not use the test mode function during safety-related operation!

7.9.6 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 PADT 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 operating state of the controller. Function and meaning of the LEDs depend on the processor system's operating system currently in use. Refer to the corresponding device-specific manuals for details.

The function and meaning of the fieldbus LEDs are described in the communication manual.

Version	Manual	Document number
CPU OS V7 and higher	SILworX Communication Manual	HI 801 101 E
CPU OS up to V7	HIMatrix PROFIBUS DP Master/Slave Manual HIMatrix Modbus Master/Slave Manual HIMatrix TCP S/R Manual HIMatrix ComUserTask (CUT) Manual	HI 800 009 E HI 800 003 E HI 800 117 E HI 800 329 E

Table 62: Manuals Describing the Communication LEDs

8.2.2 Diagnostic History

The diagnostic history records the various states of the processor 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 differs for hardware and the operating system versions:

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

Table 63: Maximum Number of Entries in the Diagnostic History for F*03

	CPU	COM
Number of entries in the long term diagnosis	300	230
Number of entries in the short term diagnosis	210	655

Table 64: Maximum Number of Entries in the Diagnostic History - up to CPU OS V7

	CPU	COM
Number of entries in the long term diagnosis	500	200/250 [1)]
Number of entries in the short term diagnosis	300	700/800 [1)]
[1)] Higher value for COM operating system version 4 and higher		

Table 65: Maximum Number of Entries in the Diagnostic History - up to CPU OS V7

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

- Reboot
- Changed mode of operation
  (INIT, RUN, STOP/VALID CONFIGURATION, STOP/INVALID CONFIGURATION),
- Changed program mode of operation
  (START, RUN, ERROR, TEST MODE),
- Configuration load or deletion
- Configuration and reset of switches
- Processor system failures
- Operating system download
- 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:

- Reboot of the communication system
- Changed mode of operation
  (INIT, RUN, STOP/VALID CONFIGURATION, STOP/INVALID CONFIGURATION),
- User log-in
- Operating system load

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:

- safe**ethernet**-related events
- Start / stop while writing to 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 V7 and Higher

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** from the context menu or the **Online** menu.

The diagnostic panel for the required module appears.

If a controller is operating, 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 - up to CPU OS V7

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. 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.

If a controller is operating, 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
- Loading Operating Systems

9.1 Interferences

Disturbances in the processor system (CPU) mostly result in the complete shut-down of the controller and are indicated via the *ERROR* LED.

Refer to the device-specific manual for the possible causes for activated *ERROR* indicators.

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 device's front plate.

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 an I/O module, check whether an external line disturbance exists and whether the corresponding sensor or actuator is ok.

9.2 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.

9.2.1 Loading the Operating System with SILworX

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

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 dialog box *Load Module Operating System,* select the type of the operating system to 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.

9.2.2 Loading the Operating System with ELOP II Factory

Use the ELOP II Factory programming tool if the operating system version loaded in the controller is **up 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. 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.

9.2.3 Switching between ELOP II Factory and SILworX - not with F*03

HIMatrix controllers (except for F*03 devices and modules) can either be programmed with ELOP II Factory or with SILworX, if the appropriate version for the 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	Up to V7	V7 and higher
Communication system	Up to V12	V12 and higher
OS loader	Up to V7	V7 and higher

Table 66: Operating System Versions and Programming Tools

9.2.3.1 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 (CPU-OS V7 and higher) into the controller.
2. Use ELOP II Factory to load the communication operating system into the controller, V12 and higher.
3. Use SILworX to load the OSL into the controller, V7 and higher.

The controller must be programmed with SILworX.

9.2.3.2 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, V7 and higher.
2. Use SILworX to load the processor operating system into the controller, V7 and higher.
3. Use SILworX to load the communication operating system into the controller, V12 and higher.

The controller must be programmed with ELOP II Factory.

i

F*03 controllers with CPU operating system V8 and higher cannot be changed to be programmed with ELOP II!

9.3 Repair of Devices and Modules

The operator is not authorized to repair devices and modules of HIMatrix systems. Defective HIMatrix systems must be returned to HIMA for repair after being tested by the operator with a brief description of the fault.

Equipment with a safety certificate is safety-relevant. The validity of the certificate expires if unauthorized repair is performed to safety-related devices of the HIMatrix system.

The warranty is void and no legal responsibility is taken for unauthorized repair.

10 Decommissioning

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

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 transport.

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
AO	Analog output
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
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
R	Read: The system variable or signal provides value, e.g., to the user program
Rack ID	Base plate identification (number)
Interference-free	Supposing that two input circuits are connected to the same source (e.g., a transmitter). An input circuit is termed *interference-free* if it does not distort the signals of the other input circuit.
R/W	Read/Write (column title for system variable/signal type)
SELV	Safety extra low voltage
SFF	Safe failure fraction, portion of faults that can be safely controlled
SIL	Safety integrity level (in accordance with IEC 61508)
SILworX	Programming tool for 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
r_{PP}	Peak-to-peak value of a total AC component
Watchdog (WD)	Time monitoring for modules or programs. If the watchdog time is exceeded, the module or program enters the ERROR STOP state.
WDT	Watchdog time

Index of Figures

Index of Tables

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

HIMA Paul Hildebrandt GmbH
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

(1336)

HI 800 141 E © by HIMA Paul Hildebrandt GmbH