HIMatrix

Safety-Related Controller

Safety manual

HIMA Paul Hildebrandt GmbH + Co KG
Industrial Automation

Rev. 1.01

HI 800 023 E

Contact

HIMA Address:

HIMA Paul Hildebrandt GmbH + Co KG

P.O. Box 1261

68777 Brühl, Germany

Tel: +49 6202 709-0

Fax: +49 6202 709-107

E-mail: info@hima.com

Revision index	Revisions	Type of Change	
		technical	editorial
1.00	The SILworX programming tool is taken into account The document layout was modified.	X	X
1.01	Range of values for watchdog time, description of safety-related analog inputs, checklist filenames	X	

Table of Contents

1 Safety manual

This manual contains information on how to operate the HIMatrix safety-related automation systems in the intended manner.

The following conditions must be met to safely install and start up the HIMatrix automation systems, 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.
- Failure to comply with the instructions detailed in this manual.

HIMA develops, manufactures and tests the HIMatrix automation devices in compliance with the pertinent safety standards and regulations. The use of the devices is only allowed if the following conditions are met:

- They are only used for the intended applications.
- They are only operated under the specified environmental conditions.
- They are only operated in connection with the approved external devices.

To provide a clearer exposition, this manual does not specify all details of all versions of the HIMatrix automation devices. For further details refer to the specific manuals.

1.1 Structure and Use of the Document

This safety manual examines the following topics:

- Intended use
- Safety concept
- Central functions
- Inputs
- Outputs
- Software
- Safety-related aspects of the user program
- Configuring communication
- Use in fire alarm systems
- Appendix:
  - Increasing the SIL of sensors and actuators
  - Glossary

Indexes

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

Programming tool	Processor operating system	Communication operating system
SILworX	Versions beyond 7	Version 12 and beyond
ELOP II Factory	Versions prior to 7	Versions prior to 12

Table 1: HIMatrix System Variants

The manual distinguishes among the different variants using:

- Separated chapters
- Tables differentiating among the versions, e.g., versions beyond 7, or prior to version 7

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

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

1.2 Target Audience

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

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 though they are not particularly marked. When the cursor hovers over a hyperlink, it changes its shape. Click the hyperlink to jump to the corresponding position.

Safety notes and operating tips are particularly marked.

1.3.1 Safety Notes

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

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

⚠ SIGNAL WORD

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

The signal words have the following meanings:

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

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.

2 Intended Use

2.1 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.

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

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

2.1.1 De-Energize to Trip Principle/ Energize to Trip Principle

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

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

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

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

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

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

2.1.2 Use in Fire Alarm Systems

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

The operating requirements must be observed!

2.2 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.

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

2.3 Operating Requirements

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

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

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

Table 2: Standards for EMC, Climatic and Environmental Requirements

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

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

Table 3: General requirements

2.3.1 Climatic Requirements

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

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

Table 4: Climatic Requirements

2.3.2 Mechanical Requirements

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

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

Table 5: Mechanical Tests

2.3.3 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 (DIS). See column "Criterion FS“ (Functional Safety).

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

Table 6: Interference Immunity Tests

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

Table 7: Interference Immunity Tests

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

Table 8: Noise Emission Tests

2.3.4 Power Supply

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

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

Table 9: Review of the DC Supply Characteristics

2.3.5 ESD Protective Measures

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

NOTE

Electrostatic discharge can damage the electronic components within the 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.**

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

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

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

2.5 Additional System Documentation

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

Name	Content	Document no. D = German E = English	Part no.
HIMatrix Engineering Manual	Description on how to plan and assemble the HIMatrix systems	HI 800 100 (D) HI 800 101 (E)	PDF file
HIMatrix System Manual Compact Systems	Description of the compact systems with the corresponding specifications	HI 800 140 (D) HI 800 141 (E)	PDF file
HIMatrix System Manual Modular System F60	Description of the modular F60 system with the corresponding specifications	HI 800 190 (D) HI 800 191 (E)	PDF file
Certified test report [1)]	Test principles, safety requirements, results	(D) (E)	96 9000104 96 9000105
SILworX Communication Manual	Description of the communication protocols, ComUserTask and their configuration in SILworX	HI 801 101 E	PDF file
HIMatrix PROFIBUS DP Master/Slave Manual	Description of the PROFIBUS protocol and its configuration in ELOP II Factory	HI 800 009 E	PDF file
HIMatrix Modbus Master/Slave Manual	Description of the Modbus protocol and its configuration in ELOP II Factory	HI 800 003 E	PDF file
HIMatrix TCP S/R Manual	Description of the TCP S/R protocol and its configuration in ELOP II Factory	HI 800 117 E	PDF file
HIMatrix ComUserTask (CUT) Manual	Description of the ComUserTask and its configuration in ELOP II Factory	HI 800 329 E	PDF file
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, INTERBUS protocol	-	-
HIMatrix First steps manual	Introduction to ELOP II Factory	HI 800 005 (D) HI 800 006 (E)	96 9000013 96 9000014 pdf file
First steps manual	Introduction to SILworX (using the HIMax system as an example)	HI 801 102 (D) HI 801 103 (E)	PDF file
[1)] Only supplied with the HIMatrix system			

Table 10: HIMatrix System Documentation

For more details on the devices and modules, refer to the corresponding manuals.

3 Safety Concept for Using the PES

This chapter contains important general items on the functional safety of HIMatrix systems.

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

3.1 Safety and availability

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

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

The HIMatrix systems have been tested and certified for use in fire alarm and fire-fighting systems in accordance with EN 54-2 and NFPA 72. To contain the hazard, these systems must be able to adopt an active state on demand.

No imminent danger results from the HIMatrix systems.

⚠ DANGER

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

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

NOTE

System damage!

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

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

3.1.1 Calculating the PFD and the PFH Values

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

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

For the controller (PES), 15 % of the limit value for PFD and PFH specified in the standard is assumed. The limit values for the controller portion is thus
PFD = 1.5 * 10^{-4} and PFH = 1.5 * 10^{-8} per hour.

A proof test interval of 10 years has been defined for the HIMatrix systems, with remote I/Os with modules and relay outputs the test interval is 3 years (offline proof test, see IEC 61508-4, paragraph 3.8.5).

3.1.2 Self-Test and Fault Diagnosis

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

- Processors
- Memory areas (RAM, non-volatile memory)
- Watchdog
- The individual I/O channels

If faults are detected during the tests, the operating systems switches off the defective module or remote I/O, or the faulty I/O channel.

In non-redundant systems, this means that sub-functions or even the entire PES will shut down.

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

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

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

For details on how to evaluate the diagnostic messages, refer to the Manual for Compact Systems (HI 800 141), or to the Manual for Modular Systems (HI 800 191), Chapter *Diagnosis*.

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

3.1.3 PADT

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

The PADT is a personal computer installed with the planning tool.

For the HIMatrix system, two planning tools are available depending on the operating system version loaded on the controller:

- SILworX must be used for operating system versions beyond 7.
- ELOP II Factory must be used for operating system versions prior to 7.

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

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

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

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

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

Ensuring the Safety Function

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

A redundant structure of the safety system components can be required to ensure the safety function:

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

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

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

Detection of Failed System Components

The safety systems recognizes that components are not functioning. This is done with:

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

The user program must be able to process the corresponding fault statuses and to activate redundant components.

3.2 Time Parameters Important for Safety

Single faults which may lead to a dangerous operating state are detected by the self-test facilities. Within the controller's safety time, the self-test facilities trigger predefined fault reactions which bring the faulty components into a safe state.

3.2.1 Fault Tolerance Time (FTT, see DIN VDE 0801, Appendix A1 2.5.3)

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

3.2.2 Safety Time (of PES)

The safety time is the time period after an internal fault occurred, during which the PES is in the RUN state and must provides a reaction.

From the process view point, the safety time is the maximum time within which the safety system must provide a reaction on the output after a change of the input signals (response time).

Operating system version	Safety time - from...to
Versions beyond 7	20...22 500 ms
Versions prior to 7	20...50 000 ms

Table 11: Range of Values for the Safety Time

3.2.3 Multiple Fault Occurrence Time (MOT)

The multiple fault occurrence time is the time span during which the probability that multiple, safety-critical faults will occur, is sufficiently low.

The multiple fault occurrence time is set in the operating system to 24 hours.

3.2.4 Response Time

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

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

- Reading the inputs
- Processing the user program
- Writing to the outputs
- Process data communication
- Performing test routines

Further, the switching times of the inputs and outputs must be taken into account when determining the worst case for the overall system.

3.2.5 Processor System Watchdog Time

The watchdog time is set in the menu for configuring the PES properties. This time is the maximum permissible duration of a RUN cycle (cycle time). If the cycle time exceeds the preset watchdog time, the CPU enters the STOP/INVALID CONFIGURATION state.

The Processor system watchdog time may be set to:
½ * PES safety time.

Operating system version	Range of values for the watchdog time	Default value for the controllers	Default value for the remote I/Os
Versions beyond 7	8...5 000 ms	200 ms	100 ms
Versions prior to 7	2...5 000 ms	50 ms	10 ms

Table 12: Range of Values for the Watchdog Time

3.3 Proof Test

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

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

With remote I/Os and modules with relay outputs, the proof test for the relay must be performed in the intervals defined for the plant.

3.3.1 Proof Test Execution

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

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

3.3.2 Frequency of Proof Tests

The HIMatrix controller can be proof tested by testing the entire safety loop.

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

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

If additional proof test requirements apply for special devices, the manual of the corresponding device must be observed.

3.4 Safety requirements

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

3.4.1 Hardware Configuration

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

Product-Independent Requirements

- To ensure safety-related operation, only approved fail-safe hardware and software may be used. The approved hardware and software are listed in the *Version List of Devices and Firmware of HIMatrix Systems of HIMA Paul Hildebrandt GmbH + Co KG, Certificate-No. 968/EZ 128.19/09.* The latest versions can be found in the version list maintained together with the test authority.
- The operating requirements specified in this safety manual (see Chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**) about EMC, mechanical, chemical, climatic influences must be observed.
- Non fail-safe, non-reactive hardware and software may be used for processing safety-relevant signals, but not for handling safety-related tasks.
- The de-energized to trip principle must be applied to all safety circuits externally connected to the system.

Product-Dependent Requirements

- Only connect devices to the system that are safely electrically isolated from the power supply.
- The safe electrical power supply isolation must be ensured within the 24 V system supply. Only power supply units of type PELV or SELV may be used.

3.4.2 Programming

Personnel developing user programs must observe the following safety requirements.

Product-Independent Requirements

- In safety-related applications, ensure that the safety-relevant system parameters are properly configured.
- In particular, this applies to the system configuration, maximum cycle time and safety time.

Product-Dependent Requirements

Requirements for using the programming tool

- The following tools must be used for programming:
  - Operating system versions beyond 7: **SILworX**.
  - Operating system versions prior to 7: **ELOP II Factory**.
- Once the application has been created, compile the program twice and compare the two resulting CRCs to ensure that the program was compiled properly.

- The correct implementation of the application specification must be validated and verified. A complete test of the logic must be performed by trial.
- Repeat this procedure every time the application is changed.
- The system response to faults in the fail-safe input modules, output modules and remote I/Os must be defined in the user program in accordance with the system-specific safety-related conditions.

3.4.3 Communication

- When implementing safety-related communications between the various devices, ensure that the system's overall response time does not exceed the fault tolerance time. All calculations must be performed in accordance with the rules given in 9.2.
- The transfer of safety-relevant data through public networks like the Internet is currently not permitted.
- If data is transferred through company-internal networks, administrative or technical measures must be implemented to ensure sufficient protection against manipulation (e.g., using a firewall to separate the safety-relevant components of the network from other networks).
- At this stage, the serial interfaces may only be used for non-safety-related purposes.
- All devices to be connected to the communication interfaces must be equipped with safe electrical isolation.

3.4.4 Maintenance Work

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

3.5 Certification

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

3.5.1 TÜV Certificate

TÜV Anlagentechnik GmbH
Automation, Software and Information Technology
Am Grauen Stein
51105 Köln

Certificate and Test Report n. 968/EZ 128.19/09
Safety-Related Automation Devices
HIMatrix F60, F35, F31, F30, F20, RIO-NC

International standards:		
EN / IEC 61508, Parts 1-7: 2000		SIL 3
EN / IEC 61511: 2004		SIL 3
EN / ISO 13849-1: 2006		Performance level e
EN / IEC 62061: 2005		
EN 50156-1: 2004		
EN 12067-2: 2004		
EN 298: 2003		
EN 230: 2005		
NFPA 85: 2007		
NFPA 86: 2007		
EN / IEC 61131-2: 2007		
EN / IEC 61000-6-2: 2005		
EN 61000-6-4: 2007		
EN 54-2: 1997 + A1:2007	F60 and F35	
EN 50130-4: 1989 + A1: 1989 +A2: 2003 + Corr. 2003		
NFPA 72: 2007	F60 and F35	

Chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** contains a detailed list of all environmental and EMC tests performed.

All devices have received the CE mark of conformity.

3.5.2 EU Type Examination

TÜV Anlagentechnik GmbH
Automation, Software and Information Technology
Am Grauen Stein
51105 Köln

EU Type Examination Certificate n. 01/205/0644/09
Safety PES System Family
HIMatrix F20, F30, F31, F35, F60, RIO-NC

International standards:		
EN / IEC 61508, Parts 1-7: 2001		SIL 3
EN / IEC 61511: 2004		
EN ISO 13849-1:2008		
EN 62061: 2005		
EN 50156-1: 2004		
EN 12067-2: 2004		
EN 298: 2003		
EN 230: 2005		
EN 61131-2: 2007		
EN 61000-6-2: 2005		
EN 61000-6-4: 2007		
NFPA 85: 2007		
NFPA 86: 2007		
EN 54-2:1997 /A1: 2007	F60 and F35	
NFPA 72: 2007	F60 and F35	

4 Central Functions

The controllers and remote I/Os of type F1.., F2.., F3.. are compact systems that cannot be modified.

The controllers of type F60 are modular systems that, when combined with a power supply module and a processor module, may be used with up to 6 I/O modules.

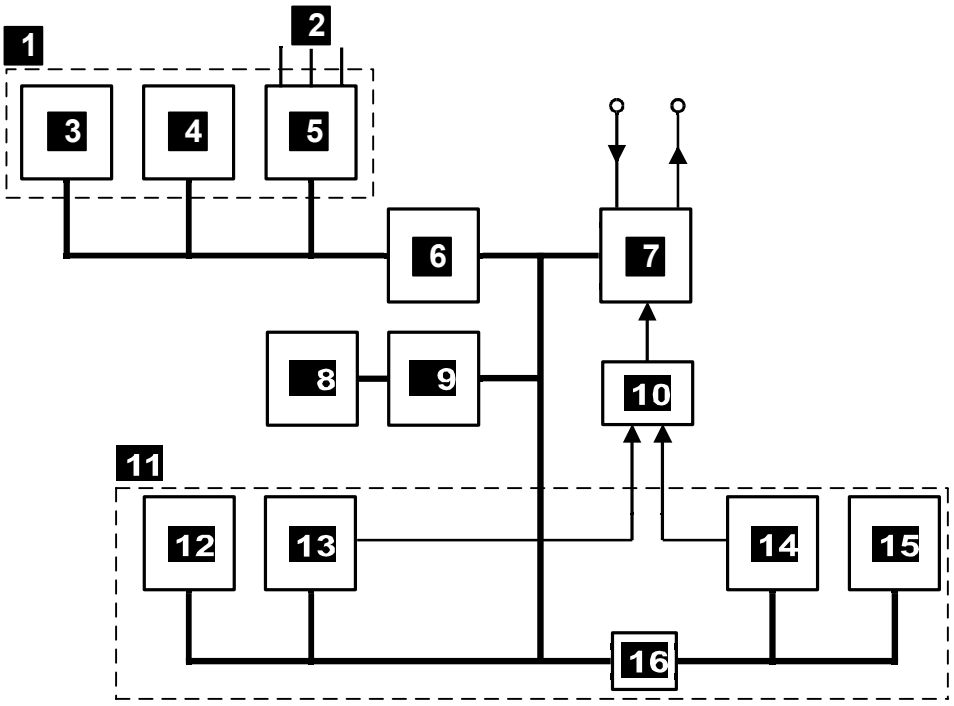
4.1 Power Supply Units

A power supply module is only available with the F60. With compact systems, this function is integrated in the device and cannot be considered as being modular.

The PS 01 power supply module (with F60) or the integrated function (with compact systems) converts the 24 VDC supply voltage to 3.3 VDC and 5 VDC (by using an internal I/O bus).

4.2 Functional Description of the Processor System

The processor system is composed of the following function blocks:



1 Communication System
2 Communication Interfaces: 2 Field Bus Interfaces, Ethernet Switch
3 Communication System ROM
4 Communication System RAM
5 Communication System Processor
6 Dual Ported RAM
7 Connection to I/O Modules
8 ROM
9 Non-Volatile Memory (NVRAM)
10 Watchdog
11 Safety-Related Processor System
12 RAM 1 for the Processor System
13 Processor 1 for the Processor System
14 Processor 2 for the Processor System
15 RAM 2 for the Processor System
16 Comparator

Figure 1: Function Blocks of the F60 CPU 01

Properties of the F60 CPU 01 processor module

- Two synchronous microprocessors (processor 1 and processor 2).
- Each microprocessor has its own RAM memory.
- Testable hardware comparators for all external accesses of both microprocessors.
- In the event of an error the watchdog is set to a safe state.
- Flash EPROM for operating system and user program, suitable for at least 100,000 memory cycles.
- Data memory in NVRAM.
- Multiplexer for connecting I/O bus, dual port RAM (DPR).
- Back-up battery or Goldcap for date/time.
- Communication processor for fieldbus and Ethernet connections.
- Interface for data transfer between F3 controllers, F60 and the PADT, based on Ethernet.
- Optional interface(s) for data exchange via fieldbus.
- LEDs for indicating the system statuses.
- I/O bus logic for connection to I/O modules.
- Safe watchdog (WD).
- Monitoring of power supply units, testable (3.3 V = / 5 V = system voltages).

4.3 Self-Tests

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

4.3.1 Microprocessor Test

The following is tested:

- All commands and addressing modes used.
- The writability of the flags and the commands generated by them.
- The writability and crosstalk of the registers.

4.3.2 Memory Areas Test

The operating system, user program, constants and parameters as well as the variable data are saved in memory areas of both processors and are tested by a hardware comparator.

4.3.3 Protected Memory Areas

The operating system, user program and parameter area are each stored in a memory. They are protected by write protection and a CRC test.

4.3.4 RAM Test

A write and read test is performed to check the modifiable RAM areas, in particular stuck-at and crosstalk.

4.3.5 Watchdog Test

The watchdog signal switches off if it is not triggered from both CPUs within a defined time window and also if the test of the hardware comparator fails. An additional test determines whether the watchdog signal is able to switch off.

4.3.6 Test of the I/O Bus Inside the Controller:

The connection between the CPU and the associated inputs and outputs (I/O modules) is tested.

4.3.7 Reactions to Processor System Failures

A hardware comparator within the central area permanently compares whether the commands and data in microprocessor system 1 and in microprocessor system 2 are identical. If they are different, or if the test routines detect failures in the processor module, the watchdog signal is automatically switched off. This means that the input signals are no longer processed by the controller and the outputs switch to the de-energized, switched-off state.

If such a fault occurs for the first time, the controller is restarted (reboot). If a further internal fault occurs within the first minute after start-up, the controller enters the STOP/INVALID CONFIGURATION state and will remain in this state.

4.4 Fault Diagnosis

Each F60 module has an own LED for reporting module malfunctions or faults in the external wiring. This allows the user to quickly diagnose faults in a faulty module.

In the compact systems F1.., F2.., F3.., these error messages are grouped together in one common error message.

Additionally, the user program can evaluate various system signals associated with the inputs, outputs or the controller.

Faults are only signaled if they do not hinder communication with the processor system, i.e., the processor system must be still able to evaluate the faults.

The user program logic can evaluate the error codes of the system signals and of all input and output signals.

An extensive diagnostic record of the system's behavior and detected faults are logged and stored in the diagnostic memory of the processor and the communication system. After a system fault, the recorded data can be read using the PADT.

For more details on how to evaluate the diagnostic messages, refer to the System Manual for Compact Systems (HI 800 141) or the System Manual for the Modular System F60, (HI 800 191), Chapter *Diagnosis*.

5 Inputs

Overview of the HIMatrix system inputs:

Device	Type	Number of inputs	Safety-related	Non-reactive	Electrically isolated
F20 controller	Digital	8	•	•	-
F30 controller	Digital	20	•	•	-
F31 controller	Digital	20	•	•	-
F35 controller	Digital	24	•	•	-
	24-bit counter	2	•	•	-
	Analog	8	•	•	-
F1 DI 16 01 remote I/O	Digital	16	•	•	-
F3 DIO 8/8 01 remote I/O	Digital	8	•	•	-
F3 DIO 16/8 01 remote I/O	Digital	16	•	•	-
F3 AIO 8/4 01 remote I/O	Analog	8	•	•	-
F3 DIO 20/8 02 remote I/O	Digital	20	•	•	-
F60 modular controller:					
DIO 24/16 01 module	Digital	24	•	•	•
DI 32 01 module (configurable for line control)	Digital	32	•	•	•
DI 24 01 module (110 V)	Digital	24	•	•	•
CIO 2/4 01 module	24-bit counter	2	•	•	•
AI 8 01 module	Analog	8	•	•	•
MI 24 01 module	Analog or digital	24	•	•	•

Table 13: Overview of the HIMatrix System Inputs

5.1 General

Safety-related inputs can be used for both safety-related signals and non-safety-related signals.

The controllers provide status and fault information as follows:

- Through the diagnostic LEDs on the devices and modules.
- Using system signals or system variables that the user program is able to evaluate.
- Storing messages in the diagnostic memory that can be read by the PADT.

Safety-related input modules automatically perform stringent, cyclic self-tests during operation. These test routines are TÜV tested and monitor the safe functioning of the corresponding module.

If a fault occurs, the controller sends a low level to the user program and, if possible, issues the fault information. The user program can read out the error code and evaluate this fault information.

For a few number of component failures that do not affect safety, no diagnostic information is provided.

5.2 Safety of Sensors, Encoders and Transmitters

In safety-related applications, the controller and its connected sensors, encoders and transmitters must all meet the safety requirements and achieve the specified SIL. For more on this, see the Annex Increasing the SIL of Sensors and Actuators.

5.3 Safety-Related Digital Inputs

The described properties apply to both digital input channels of F60 modules and digital input channels of all compact systems (unless stated otherwise).

5.3.1 General

The digital inputs are read once per cycle and saved internally; cyclic tests are performed to ensure their safe functioning.

Input signals that are present for a time shorter than the time between two samplings, i.e., shorter than a cycle time, may not be detected.

5.3.2 Test Routines

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

5.3.3 Reaction in the Event of a Fault

If the test routines for digital inputs detect a fault, the user program processes a low level for the defective channel in accordance with the de-energize to trip mode

In addition to the channel signal value, the user program must also consider the corresponding error code.

A compact system activates the *ERROR* LED*,* a F60 module the *ERR* LED.

The error code allows the user to monitor the external wiring and program additional fault reactions in the user program.

Version	Access to the error code	Error code name
Versions beyond 7	In the ...*Channels* tab located in the detail view of the module or device	->Error code [bytes] in the row with the channel number
Versions prior to 7	In the *Signal Connections...* window of the module or device	DI[*xx*].error code, *xx* = channel number

Table 14: Error Codes with Digital Inputs

5.3.4 Surges on Digital Inputs

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

The following measures ensure proper operation in environments where surges may occur:

1. Install shielded input wires
2. Activate noise blanking: a signal must be present for at least two cycles before it is evaluated.

i Activating noise blanking increases the response time of the HIMatrix system!

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

In particular, the design must include protective measures with respect to overvoltage, lightning, earth grounding and plant wiring in accordance with the relevant standards and the instructions specified in the System Manual (HI 800 141 or HI 800 191).

5.3.5 Configurable Digital Inputs

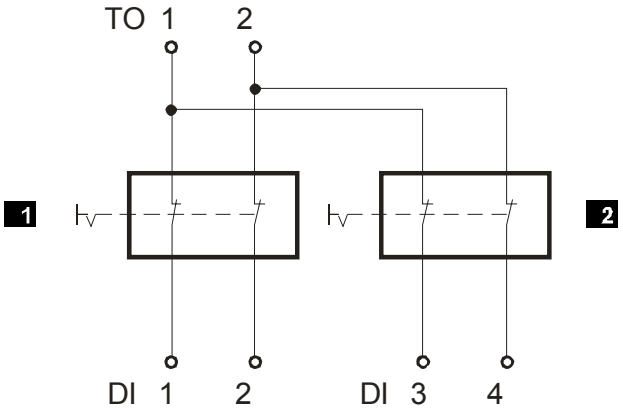
The digital inputs of the F35 controller and the MI 24 01 module operate as analog inputs, but return digital values due to the configuration of switching thresholds.

For configurable digital inputs, the test routines and safety-related functions for analog inputs apply as specified in Chapter 5.4.

5.3.6 Line Control

Line control is used to detect short-circuits or open-circuits and can be configured for the HIMatrix systems with digital inputs (and not with configurable digital inputs), e.g., on EMERGENCY STOP devices complying with Cat. 4 in accordance with EN 954-1.

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

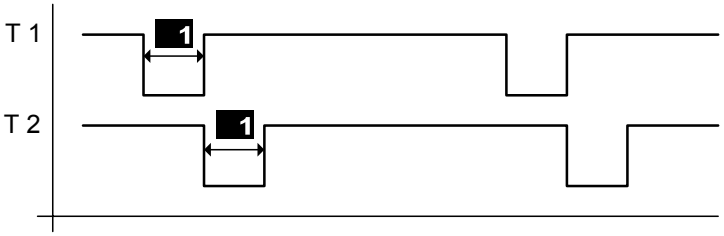


1 EMERGENCY STOP 1
2 EMERGENCY STOP 2

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

Figure 2: Line Control

The digital outputs TO 1 and TO 2 are pulsed (T1 and T2) to monitor the wire to the digital inputs. The signals for the pulsed outputs must begin with *DO[01].Value* and reside in direct sequence, one after the other (see system variables/signals in the manuals).



1 Configurable 5...2 000 µs

Figure 3: Pulsed Signal T1, T2

Line control detect the following faults:

- Cross-circuit between two parallel lines,
- Improper connections of two lines (e.g., TO 2 to DI 3),
- Earth fault of a line (with earthed ground only),
- Open-circuit or open contacts, i.e., including when one of the two EMERGENCY STOP switches mentioned above has been engaged, the 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 controller's front plate blinks.
- The inputs are set to low level.
- An (evaluable) error code is created.

5.4 Safety-Related Analog Inputs (F35, F3 AIO 8/4 01 and F60)

5.4.1 General

The analog input channels convert the input signals into a value of type INTEGER. The values are available to the user program as variables associated with the following system variables or system signals :

Operating system version	Value
Versions beyond 7	System variable -> *Value [INT]*
Versions prior to 7	System signal *AI[xx].Value* (xx = channel number).

Table 15: Value of Safety-Related Analog Inputs

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

The value range for the inputs depend on the device or module:

F35 controller

Input channels	Measurement procedure	Current, voltage	Range of values in the application		Safety-related accuracy
			FS1000[1)]	FS2000[1)]	
8	Unipolar	0...+10 V	0...1000	0...2000	2 %
8	Unipolar	0...20 mA	0...500[2)] 0...1000[3)]	0...1000[2)] 0...2000[3)]	2 %

[1)] can be configured by selecting the type in the PADT

[2)] with external 250 Ω shunt adapter, part number: 98 2220059

[3)] with external 500 Ω shunt adapter, part number.: 98 2220067

Table 16: Analog Inputs of the F35 Controller

F3 AIO 8/4 01 remote I/O

Input channels	Measurement procedure	Current, voltage	Range of values in the application	Safety-related accuracy
8	Unipolar	0...+10 V	0...2000	2 %
8	Unipolar	0...20 mA	0...1000[1)] 0...2000[2)]	2 %

[1)] with external 250 Ω shunt adapter, part number.: 98 2220059

[2)] with external 500 Ω shunt adapter, part number.: 98 2220067

Table 17: Analog Inputs of the F3 AIO 8/4 01 Remote I/O

F60 controller

Input channels **AI 8 01**	Measurement procedure	Current, voltage	Range of values in the application		Safety-related accuracy
			FS1000[1)]	FS2000[1)]	
8	Unipolar	-10...+10 V	-1000...1000	-2000...2000	1 %
8	Unipolar	0...20 mA	0...1000[3)]	0...2000[3]	1 %
8	Unipolar	0...20 mA	0...500[2)]	0...1000[2)]	4 %
4	bipolar	-10...+10 V	-1000...1000	-2000...2000	1 %
MI 24 01					
24	Unipolar	0...20 mA	0...2000[4)]		1 %

[1)] can be configured by selecting the type in the PADT (F60)

[2)] with 250 Ω external shunt, part number: 00 0710251

[3)] with 500 Ω external shunt, part number: 00 0603501 (accuracy 0.05%, P 1 W)

[4)] internal shunts

Table 18: Analog Inputs of the F60 Controller

The module AI 8 01 of the HIMatrix F60 can be configured in the user program for eight unipolar or four bipolar functions. However, the mixing of functions on a module is not permitted.

The analog inputs of the F35 controller, the F3 AIO 8/4 01 remote I/O and the AI 8 01 module operate with voltage measurement. With the analog inputs of the HIMatrix F35 and F3 AIO 8/4 01, digital outputs of the own system (F35) or of other HIMatrix controllers can be monitored to detect open-circuits. Further information is available in the manuals of the corresponding HIMatrix controllers.

If an open-circuit occurs (the line is not monitored by the system), any input signals are processed on the high-resistance inputs. The value resulting from this fluctuating input

voltage is not reliable; with voltage inputs, the channels must be terminated with a 10 kΩ resistor. The internal resistance of the source must be taken into account.

To measure currents, the shunt is connected in parallel to an input; in doing so the 10 kΩ resistor is not required.

The inputs of the MI 24 01 module are only current inputs, because of the internal shunts, and cannot be used as voltage inputs.

If input channels are not used, the measurement input must be connected to the reference potential. Thus negative influences (fluctuating input voltages) on other channels in case of an open-circuit are avoided.

Operating system version	Procedure
Versions beyond 7	It is sufficient not to assign unused inputs global variables.
Versions prior to 7	For the unused input channel, set the corresponding signal *AI[0x].Used* to the default value *FALSE* or *0* in ELOP II Hardware Management. In doing so, the channel is masked out in the user program, i.e., no signals of this channel are available within the logic.

Table 19: Configuration of Unused Inputs

5.4.2 Test Routines

The controller processes analog values in parallel via two multiplexers and two analog/digital converters with 12-bit resolution and compares the results. Additionally, the controller switches on test values via digital/analog converters, converts them back to digital values and compares them with the default values.

When an error is detected, the controller sends 0 to the user program as input value for further processing, and sets the error state.

5.4.3 Reaction in the Event of a Fault

If there are channel faults in the analog inputs, the error code of the corresponding channel is set to a value > 0. If the entire module is faulty the error code for the module is set to a value > 0.

In addition to the analog value, the user program must also evaluate the error code. For values less than 0, a safety-related reaction must be planned.

A compact system activates the *FAULT* LED, a F60 module the *ERR* LED.

The error code allows the user to monitor the external wiring and program additional fault reactions in the user program.

Version	Access to the error code	Error code name
Versions beyond 7	In the ...*Channels* tab located in the detail view of the module or device	*->Error code [bytes]* in the row with the channel number
Versions prior to 7	In the *Signal Connections...* window of the module or device	*AI[xx].error code*, *xx* = channel number

Table 20: Error Codes with Analog Inputs

5.5 Safety-Related Counters (F35 and F60)

Unless otherwise noted, the points previously mentioned apply for the CIO 2/4 01 counter module of the F60 as well as for the counters of the F35.

5.5.1 General

A counter channel can be configured for operation as a high-speed up or down counter with 24-bit resolution or as a decoder in Gray code.

If used as high-speed up or down counters, the pulse input and count direction input signals are required in the application. A reset only takes place in the user program.

The CIO 2/4 01 counter module of the F60 has 4-bit or 8-bit encoder resolution, whereas the F35 has a 3-bit or 6-bit encoder resolution. A reset is possible.

The interconnection of two independent 4-bit inputs to an 8-bit input (example of F60) can only be carried out via the user program. No switching option is planned for this purpose.

The encoder function monitors the change of the bit pattern on the input channels. The bit patterns on the inputs are transferred directly to the user program. They are represented in the PADT as decimal numbers corresponding to the bit pattern (*Counter[0x].Value*).

Depending on the application, this number (which corresponds to the Gray Code bit pattern) can be converted into, for example, the corresponding decimal value.

5.5.2 Reaction in the Event of a Fault

If the test facilities detect a fault in the counter section of the device or module, they set a status bit for evaluation in the user program. Additionally, the user program can also consider the corresponding error code.

A compact system activates the *ERROR* LED, a F60 module the *ERR* LED.

The error code allows the user to monitor the external wiring and program additional fault reactions in the user program.

Version	Access to the error code	Error code name
Versions beyond 7	In the ...*Channels* tab located in the detail view of the module or device	*->Error code [bytes]* in the row with the channel number
Versions prior to 7	In the *Signal Connections...* window of the module or device	*Counter[xx].error code*, *xx* = channel number

Table 21: Error Codes with Counter Inputs

5.6 Checklist for Safety-Related Inputs

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

When engineering or starting up the system, a checklist must be filled out for each of the safety-related input channels used in the system to verify the requirements to be met. This is the only way to ensure that all requirements were considered and clearly recorded. The checklist also documents the relationship between the external wiring and the user program.

The checklist *HIMatrix_Checklist_Inputs.doc* is available in Microsoft® Word® format. All the checklists are contained in the ZIP file *HIMatrix_Checklists.zip* that can be downloaded from the HIMA website www.hima.com.

6 Outputs

Overview of the HIMatrix system outputs:

Device	Type	Number of inputs	Safety-related	Electrically isolated
F20 controller	Digital	8	•	-
	Pulse	4	-	-
F30 controller (configurable for line control)	Digital	8	•	-
F31 controller (configurable for line control)	Digital	8	•	-
F35 controller		8	•	-
F1 DI 16 01 remote I/O	Pulse	4	-	-
F2 DO 4 01 remote I/O	Digital	4	•	-
F2 DO 8 01 remote I/O	Digital	8	•	•
F2 DO 16 01 remote I/O	Digital	16	•	-
F2 DO 16 01 remote I/O	Relay	16	•	•
F3 DIO 8/8 01 remote I/O	Digital one-pole	8	•	-
	Digital two-pole	2		
F3 DIO 16/8 01 remote I/O	Digital one-pole	16	•	-
	Digital two-pole	8		
F3 AIO 8/4 01 remote I/O	Analog	4	-	-
F3 DIO 20/8 01 and F3 DIO 20/8 02 remote I/Os (configurable for line control)	Digital	8	•	-
F60 modular controller:				
DIO 24/16 01 module (configurable for line control)	Digital	16	•	•
DI 32 01 module (configurable for line control)	Digital	32	•	•
DO 8 01 (110 V) module	Relay	8	•	•
CIO 2/4 01 module	Digital	4	•	•
AO 8 01 module	Analog	8	•	•

Table 22: Overview of the HIMatrix System Outputs

6.1 General

The controller writes to the safety-related outputs once per cycle, reads back the output signals and compares them with the specified output data.

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

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

The CPU watchdog signal is the second way to perform a safety shutdown: If the watchdog signal is lost, the safe state is immediately adopted.

This function is only effective for all digital outputs and relay outputs of the controller.

The error code allows the user to configure additional fault reactions in the user program.

6.2 Safety of Actuators

In safety-related applications, the controller and its connected actuators must all meet the safety requirements and achieve the specified SIL. For more details, see Increasing the SIL of Sensors and Actuators in the Annex.

6.3 Safety-Related Digital Outputs

The following points apply to the digital output channels of the F60 modules and to the digital output channels of the compact systems, but not to the relay outputs.

6.3.1 Test Routines for Digital Outputs

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

- Reading the output signals back from the switching amplifier. The switching threshold for a read-back signal is 2 V. The diodes used prevent a feed back of signals.
- Checking the integrated redundant safety shutdown.
- A shutdown test of the outputs is carried out within the MOT for a maximum of 200 µs. The minimum time between two tests is ≥ 20 seconds.

The system monitors its operating voltage and de-energizes all outputs at a low voltage of less than 13 V.

6.3.2 Reaction in the Event of a Fault

If the controller detects a faulty signal, it sets the affected device or module output to the safe, de-energized state using the safety switches. If a module fault occurs, all module outputs are switched off. A compact system additionally reports the two faults via the *ERROR* LED, a F60 module via the *ERR* LED.

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

If the output is short-circuited to L- or overloaded, the device or module is still testable. A safety shutdown is not required.

The controller monitors the device's or module's total current input and set all output channels to the safe state if the threshold is exceeded.

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

6.3.4 Line Control

The controller can pulse safety-related digital outputs and use them with the safety-related digital inputs of the same system (but not the configurable digital inputs) to detect open-circuits and short-circuits, e.g., on EMERGENCY STOP devices complying with Cat. 4 in accordance with EN 954-1 (see Chapter 5.3.6 Line Control).

NOTE

Malfunctions of the connected actuators are possible!

Pulsed outputs must not be used as safety-related outputs (e.g., for activating safety-related actuators)!

Relay outputs cannot be used as pulsed outputs.

6.4 Safety-Related Two-Pole Digital Outputs

The following points apply to two-pole digital outputs of the compact systems.

6.4.1 Test Routines for Two-Pole Digital Outputs

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

- Reading the output signals back from the switching amplifier. The switching threshold for a read-back signal is 2 V. The diodes used prevent a feed back of signals.
- Checking the integrated (redundant) safety shutdown.
- A shutdown test of the outputs is carried out within the MOT for a maximum of 200 µs. The minimum time between two tests is ≥ 20 seconds.
- Line diagnosis with two-pole connection
  F3 DIO 16/8 01:
  - Short-circuit to L+, L-.
  - Short-circuit between two-pole connections.
  - Open-circuit in one of the two-pole lines.
- Line diagnosis with two-pole connection
  F3 DIO 8/8 01: Short-circuit to L+, L-..
- Test of L- switch capability at two-pole connection with line diagnosis (F3 DIO 16/8 01).
- Monitoring of the output current

The system monitors its operating voltage and de-energizes all outputs at a low voltage of less than 13 V.

6.4.2 One-Pole/Two-Pole Connection (F3 DIO 8/8 01, F3 DIO 16/8 01):

The digital outputs can be configured as follows:

- Digital output with two-pole connection with line diagnosis
- Digital output with two-pole connection without line diagnosis
- Digital output with one-pole L+ switching DO+
- Digital output with one-pole L- switching DO+

Two-Pole Connection

NOTE

A relay or actuator connected to the output may accidentally be switched on!

With applications in accordance with EN 954-1 Cat. 4, the line diagnosis status signal must be used to switch off the outputs (DO+, DO-), if a failure occurs.

i

If the requirements previously described cannot be met, observe the following case:

If a short-circuit occurs between DO- and L-, a relay may be energized or some other actuator may be set to a different switching state.

Reason: During the monitoring time specified for line diagnosis, a 24 V voltage level (DO+ output) is present on the load (relay, switching actuator) allowing it to receive enough electrical power to potentially switch to another state.

NOTE

Detection of open-circuits may be disturbed!

In a 2-pole connection, no DI input must be connected to a DO output. This would inhibit the detection of open-circuits.

NOTE

The controller or contiguous electronic devices or systems may be disturbed!

Inductive loads must be connected with free-wheeling diode on the actuator.

6.4.3 Reaction in the Event of a Fault

DO- Outputs

If a faulty signal is detected, the device or module sets the affected output to the safe, de-energized state using the safety switches. A device or module fault causes all outputs to switch off. A compact system additionally reports the two fault types via the *ERROR* LED, a F60 module via the *ERR* LED.

DO+ Outputs

If a faulty signal is detected, the device or module sets the affected output to the safe, de-energized state using the safety switches. A device or module fault causes all output to switch off. A compact system additionally reports the two faults via the *ERROR* LED, a F60 module via the *ERR* LED.

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

If the output is short-circuited to L-, L+ or overloaded, the device or module is still testable. A safety shutdown is not required.

The total current input of the device or module is monitored. If the threshold is exceeded, the output device or the module sets all channels to the safe state.

In this state, the device or module checks the outputs every few seconds to determine whether the overload is still present. In a normal state, the device or module switches on the outputs once again.

6.5 Relay Outputs

The relay outputs correspond to functional digital outputs, but offer galvanic isolation and higher electrical strength.

6.5.1 Test Routines for Relay Outputs

The device or the module automatically tests its outputs during operation. The main test functions are:

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

The system monitors its operating voltage and de-energizes all outputs at a low voltage of less than 13 V.

With the DO 8 01 module and the F2 DO 8 01 and F2 DO 16 02 remote I/Os, the outputs are equipped with three safety relays:

- Two relays with forcibly guided contacts.
- One standard relay.

This enables the outputs to be used for safety shutdowns.

6.5.2 Reaction in the Event of a Fault

If a faulty signal is detected, the device or module sets the affected output to the safe, de-energized state using the safety switches. If a module fault occurs, all module outputs are switched off. Additionally, a compact system reports the two faults via the *ERROR* LED, a F60 module via the *ERR* LED.

6.6 Safety-Related Analog Outputs (F60)

The AO 8 01 module has an own safety-related 1oo2 A/D microprocessor system with safe communication. It writes to the analog outputs once per cycle and saves the values internally. The module itself tests the function..

The DIP switches on the safety-related analog output modules can be used to set the outputs to voltage or current outputs. In doing so, ensure that the setting for use in the system comply with the configuration in the user program. If this is neglected, faulty module behavior may result.

NOTE

Module malfunctions are possible!

Prior to inserting the module into the system, check the following:

- **Module's DIP switch settings.**
- **Module configuration in the user program.**

Depending on the device type selected (...FS1000, ...FS2000) during configuration, multiple values must be taken into account in the logic for the output signals to obtain identical output values (see the AO 8 01 Manual (HI 800 195), Chapter *Signals and Error Codes for the Outputs*).

Each group of two analog outputs are galvanically connected:

- Outputs 1 and 2.
- Outputs 3 and 4.
- Outputs 5 and 6.
- Outputs 7 and 8.

The analog output circuits have current or voltage monitoring, read back and test channels (even for parallel output circuits), as well as two additional safety switches for the safe disconnection of the output circuits in the event of a fault. This ensures that the safe state is achieved (current output: 0 mA, voltage output: 0 V).

6.6.1 Test Routines

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

- Duplicated read back of the output signal.
- Crosstalk test between the outputs.
- Checking the integrated safety shutdown.

6.6.2 Reaction in the Event of a Fault

The module reads back the output signals once every cycle and compares them with the internally saved output signals. If the module detects a discrepancy, it switches off the faulty output channel via the two safety switches and reports a module fault via the *ERR* LED.

The error code allows the user to configure additional fault reactions in the user program.

To determine the worst case reaction time of the analog outputs, add the double watchdog time of the AO CPU (2 * $WDT_{AO\ \mu C}$) to the double watchdog time (2 * WDT_{CPU}).

the worst case reaction time is specified in the corresponding manual.

6.7 Analog Outputs with Safety-Related Shutdown (F3 AIO 8/4 01)

The remote I/O writes to the analog outputs once per cycle and saves the values internally.

All the outputs are non-safety-related, but all together they can be shut down safely.

To achieve SIL 3, the output values must be read back via safety-related analog inputs and evaluated in the user program. Reactions to incorrect output values must also be specified in the user program.

6.7.1 Test Routines

The remote I/O automatically tests the two safety switches used to switch off all 4 module outputs during operation.

6.7.2 Reaction in the Event of a Fault

If an internal fault occurs, the remote I/O simultaneously switches off all 4 output channels via the two safety switches and reports the module fault via the *FAULT* LED on the front plate.

The error code allows the user to configure additional fault reactions in the user program.

6.8 Checklist for Safety-Related Outputs

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

When engineering or starting up the system, a checklist must be filled out for each of the safety-related output channels used in the system to verify the requirements to be met. This is the only way to ensure that all requirements were considered and clearly recorded. The checklist also documents the relationship between the external wiring and the user program.

The checklist *HIMatrix_Checklist_Outputs.doc* is available in Microsoft® Word® format. All the checklists are contained in the ZIP file *HIMatrix_Checklists.zip* that can be downloaded from the HIMA website www.hima.com.

7 Software for HIMatrix Systems

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

- Operating system
- User program
- Programming tool in accordance with IEC 61131-3.

The operating system is loaded into the controller's central part (CPU) and must be used in the current version certified by TÜV for safety-related applications.

The programming tool serves for creating the user program with the application-specific functions that should be performed by the automation device. The programming tool is also used to configure and operate the operating system functions.

The code generator integrated in the programming tool translates the user program into a machine code. The programming tool uses the Ethernet interface to transfer this machine code to the flash EPROM of the automation device.

7.1 Safety-Related Aspects of the Operating System

Each approved operating system is identified by a unique name. To help distinguish the systems from one another, the version number and the CRC signature are given. The valid versions of the operating system and corresponding signatures (CRCs) - approved by the TÜV for use in safety-related automation devices - are subject to a revision control and are documented in a list maintained together with the TÜV.

The current version of the operating system can be read using the programming tool. A control check performed by the user is required (see 7.6 Checklist for Creating a User Program).

7.2 Operation and Functions of the Operating System

The operating system executes the user program cyclically. In a simplified form, it performs the following functions:

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

The following basic functions are also executed:

- Comprehensive self-tests.
- Test of inputs and outputs during operation.
- Data transfer.
- Diagnosis.

7.3 Safety-Related Aspects of Programming

7.3.1 Programming Tool's Safety Concept

The safety concept on which the two programming tools, ELOP II Factory and SILworX, are based on is:

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

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

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

Function Test of the Controller

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

The controller and the application are sufficiently tested.

If a user program is modified, only the program components affected by the change must be tested.

Versions beyond 7

The safe revision comparator in SILworX can be used to determine and display all changes relative to the previous version.

7.3.2 Verifying the Configuration and the User Program

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

An independent test of each loop (consisting of input, the key interconnections in the application and output) is usually sufficient. The programming tool and the measures defined in this safety manual make it sufficiently improbable that a code generated properly from a semantic and syntactic view point can still contain undetected systematic faults resulting from the code generation process.

Suitable test cases must also be created for the numerical evaluation of formulas. Equivalence class tests are convenient which are tests within defined ranges of values, at the limits of or within invalid ranges of values. The test cases must be selected such that the program logic can be proven to be correct. The required number of test cases depends on the program logic used and must include critical value pairs.

An active simulation with data sources is the only way to prove that the sensors and actuators in the system (also those connected to the system via communication with remote I/Os) are properly wired. This is also the only way to verify the system configuration.

This procedure must be followed both when creating the user program for the first time and when modifying it.

7.3.3 Archiving a Project

HIMA recommends archiving the project every time the program is loaded into the controller by performing a download or a reload.

The procedure for archiving a project is radically different in ELOP II Factory and SILworX.

Archiving a Project - Versions Beyond 7

SILworX creates a project in a project file. This must be suitably stored, e.g., on a storage medium.

Archiving a Project - Versions Prior to 7

ELOP II Factory creates a project in a structure of sub-directories. To archive the project, ELOP II Factory can store the content of this structure to an archive file, the project archive. This project archives must be suitably stored, e.g., on a storage medium.

To create a project archive

1. Print the user project to compare the logic with the specifications.
2. Compile the user program for generating the CPU configuration CRC.
3. Verify the CRCs and note down the CPU configuration CRC version. To do so, right-click the controller in the Hardware Management and select **Configuration Information** to display the versions. The following information is required to determine a version:
   - rootcpu.config shows the safety-related CPU configuration, i.e., the CPU configuration CRC.
   - rootcom.config shows the non-safety-related COM configuration.
   - root.config shows the overall configuration, including the remote I/Os (CPU + COM).
4. Create a project archive with the user program name, the CPU configuration CRC CPUs and date and store it to a storage medium.
   This recommendation does not replace the user's internal documentation requirements.

The project archive is complete.

7.3.4 Options for Identifying the Program and the Configuration

The application programs are unambiguously identified with the configuration CRC of the project. This can be compared to the configuration CRC of the loaded projects.

Project Files - Versions Beyond 7

To ensure that the saved project file remained unchanged, compile the corresponding resource and compare the configuration CRC with the loaded configuration's CRC. This CRC can be displayed with SILworX.

Archive - Versions Prior to 7

The archive name should contain the configuration CRCs of the root.config.

To ensure that the the used archive did not changed, compile the resource after restoring the project from the archive and compare the configuration CRC of root.config with the CRC of the loaded configuration that can be displayed with ELOP II Factory.

To check them, open the **Resource → Consistency Check** in the resource's Control Panel.

7.4 Resource Parameters

⚠ DANGER

Physical injury possible due to incorrect configuration!

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

These parameters are:

- **System ID**
- **Rack ID, refer to the system manuals (HI 800 141 and HI 800 191).**
- **Safety Time**
- **Watchdog Time**
- **Main Enable**
- **Autostart**
- **Start Allowed**
- **Load allowed**
- **Reload Allowed**
- **Global Forcing Allowed**

The following parameters are defined in the programming tool for the operations permissible in the safety-related operation of the automation device and are referred to as safety-related parameters.

Parameters that may be defined for safety-related operation are not firmly bound to any specific requirement classes. Instead, each of these must be agreed upon together with the responsible test authority for each separate implementation of the controller.

7.4.1 Parameters - Versions Beyond 7

In versions beyond 7, the distinction between system parameter of the resource and system parameters of the hardware is made.

System Parameters of the Resource

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

Parameter / Switch	Function	Default setting	Setting for safe operation
Name	Resource name	-	Arbitrary
System ID [SRS]	System ID of the resource 1...65 535	60 000	Unique value within the controller network. This includes all controllers that may be potentially connected with one another.
Safety Time [ms]	Safety time in milliseconds 20...22 500	600	Application-specific
Watchdog Time [ms]	Watchdog time in milliseconds 8...5 000 ms	200	Application-specific
Main Enable	The following switches/parameters can be changed during operation (= RUN) using the PADT. When OFF, the parameters are "frozen“	ON	OFF is recommended
Autostart	Automatic start allowed after Power ON of CPU	OFF	Application-specific
Start Allowed	A cold start, warm start or hot start permitted with the PADT in RUN or STOP	ON	Application-specific
Load allowed	Load of the user program permitted	ON	Application-specific
Reload Allowed	Releases reloading a user program. The reload process currently running is not aborted when switching to OFF	ON	Application-specific
Global Forcing Allowed	Releases global forcing for this resource	OFF	Application-specific
Global Force Timeout Reaction	Determines how the resource should behave on expiry of the global force time-out: Stop Forcing Only, Stop Resource	Stop Forcing Only	Application-specific
Max.Com. Time Slice ASYNC [ms]	Highest value in ms for the time slice used for communication, 2...5000 ms	10	Application-specific
Period	Cycle time required for the resource	0	Application-specific

Table 23: System Parameters of the Resource

Hardware System Variables

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

Parameter / Switch	Function	Default setting	Setting for safe operation
Force Deactivation	Used to prevent forcing and to stop it immediately	OFF	Application-specific
Spare 0 ... 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	OFF	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.	OFF	Application-specific
Reload Deactivation	Prevents execution of reload	OFF	Application-specific

Table 24: Hardware System Variables

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

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

7.4.2 Parameters - Versions Prior to 7

Safety-Related Parameters	Safe settings
Safety time in ms	Process-dependent
Watchdog time in ms	max. 50 % of the safety time
Start/Restart [1)]	Reset / Off (in RUN only)
Force Enable	Reset / Off
Forcing (individual switch) [1)]	Reset / Off
Main Enable Switch (Change of the safety parameters) [1)]	Reset / Off
Test mode [1)]	Reset / Off (in RUN only)
[1)] It cannot be changed with remote I/Os (except for F3 DIO 20/8 01)	

Table 25: Resource Parameter - Versions Prior to 7

7.5 Protection against Manipulation

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

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

- Each change to the user program or configuration creates a new CRC.
- The operating options depend on the user login into the PES.
- The programming tool prompts the user to enter a password in order to connect to the PES.
- No connection is required between the PADT and PES in RUN.

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

NOTE

Only authorized personnel may be granted access to the HIMatrix controller!

Take the following measures to ensure protection against unauthorized changes to the controller:

- **Change the default settings for user name and password!**
- **Users must keep their passwords secret.**
- **Upon completion of the start-up phase, disconnect the PADT from the controller and only connect it again if changes are necessary.**

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

The connection between PADT and PES is only required for downloading the user program or reading the variables or signals. The PADT is not required during normal operation. Disconnecting the PADT and PES during normal operation protects against unauthorized access.

7.6 Checklist for Creating a User Program

To comply with all safety-related aspects during the programming phase, HIMA recommends using the checklist prior to and after loading a new or modified program.

The checklist *HIMatrix_Checklist_Program.doc* is available in Microsoft® Word® format. All the checklists are contained in the ZIP file *HIMatrix_Checklists.zip* that can be downloaded from the HIMA website www.hima.com.

8 Safety-related Aspects of the User Program

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

- Specify the controller functionality.
- Write the user program.
- Use the C-code generator to compile the user program.
- Compile the user program a second time and compare the resulting CRCs.
- The program generated is error-free and can run,
- Verify and validate the user program.

Finally, the PES can start the safety-related operation.

8.1 Scope for Safety-Related Use

(Refer to Chapter 3.4 for more details about specifications, rules and explications to safety requirements)

Enter the user program with the allowed programming tool:

- SILworX for operating system versions beyond 7.
- ELOP II Factory for operating system versions prior to 7.

Which operating systems for personal computer have been released is specified in the release notes of the programming tool.

Essentially, the programming tool includes:

- Input (Function Block Editor), monitoring and documentation.
- Variables with symbolic names and data types (BOOL, UINT, etc.).
- Assignment of HIMatrix controllers.
- Code generator (for translating the user program into a machine code).
- Hardware configuration.
- Communication configuration.

8.1.1 Programming Basics

The tasks to be performed by the controller should be defined in a specification or a requirements specification. This documentation serves as the basis for checking its proper implementation in the user program. The specification format depends on the tasks to be performed. These include:

- Combinational logic.
  - Cause/effect diagram.
  - Logic of the connection with functions and function blocks.
  - Function blocks with specified characteristics.
- Sequential controllers (sequence control system).
  - Written description of the steps and their enabling conditions and of the actuators to be controlled.
  - Flow charts.
  - Matrix or table form of the step enabling conditions and the actuators to be controlled.
  - Definition of constraints, e.g., operating modes, EMERGENCY STOP, etc.

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

- Sensors (digital or analog).
  - Signals during normal operation ('de-energize-to-trip' principle with digital sensors, 'life-zero' with analog sensors).
  - Signals if a fault occurs.
  - Definition of required safety-related redundancies (1oo2, 2oo3) (see Chapter Increasing the SIL of Sensors and Actuators)
  - Discrepancy monitoring and reaction.
- Actuators.
  - Positioning and activation during normal operation.
  - Safe reaction/positioning at shutdown or after power loss.

Programming goals for user program:

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

8.1.2 Functions of the User Program

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

- Only elements complying with IEC 61131-3 together with their functional requirements are permitted within the logic.
- The physical inputs and outputs usually operate in accordance with the 'de-energize-to-trip' principle, i.e. their safe state is *0*. This must be taken into account during programming.
- The user program includes meaningful logic and/or arithmetic functions irrespective of the 'de-energize-to-trip' principle of the physical inputs and outputs.
- The program logic should be clear and easy to understand and well documented to assist in debugging. This includes the use of functional diagrams.
- Any kind of negations are permitted.
- Fault signals from the inputs or outputs, or from logic blocks must be evaluated.

The "packaging" of functions in user-defined function blocks and functions consisting of standard functions is important. This ensures that a program can be clearly structured in modules (functions, function blocks). Each module can be considered individually; the user can create a comprehensive, complex function by grouping the individual modules to form a single larger module or a single program.

8.1.3 Declaration of Variables and Signals

A variable is a placeholder for a value within the program logic. The variable name is used to symbolically address the storage space containing the stored value. A variable is created in the variable declaration for the program or function block.

	Number of characters for the names of variables
Version beyond 7	31
Version prior to 7	256

Table 26: Length for the Name of the Variable

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

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

In versions beyond 7, variables are used instead of signals.

Signals - Versions Prior to 7

A signal is used for associating various areas of the overall controller. The signal is created in the Signal Editor and corresponds to the global level of a program's VAR_EXTERNAL, if the connection has been previously established.

8.1.4 Acceptance by Test Authority

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

8.2 Procedures

This chapter describes the procedures typically used for developing the user programs for safety-related HIMatrix controllers.

8.2.1 Assigning Variables to Inputs or Outputs

The required test routines for safety-related I/O devices, I/O modules or I/O channels are automatically executed by the operating system.

The procedure for assigning the variables used in the user program is different in ELOP II Factory and SILworX.

Operating System Versions Beyond 7

To assign a variable to an I/O channel

1 Define a global variable of an appropriate type.
2. Enter an appropriate initial value, when defining the global variable.
3 Assign the global variable the channel value of the I/O channel.
4. In the user program, evaluate the error code -> *Error Code [Byte]* and program a safety-related reaction.

The global variables is associated with an input/output channel

Operating System Versions Prior to 7

Proceed as follows to assign the value of a variable to an I/O channel:

To assign a variable to an I/O channel

1 Define a variable of an appropriate type.
2. In the Signal Editor located in the Hardware Management, define a signal and name it as the variable.
3. Drag the signal onto the program's variable declaration.
4. Drag the signal onto the channel list associated with the I/O module.
5. In the user program, evaluate the error code and program a safety-related reaction.

The variable is assigned to an I/O channel.

The system signal name for the error code depends on the I/O channel type.

8.2.2 System Parameters of the Resource

The parameters listed below define how the controller behaves during operation and are configured in the resource's properties.

Specific switches define which PADT actions are allowed during safety-related operation.

System Parameters of the Resource - Versions Beyond 7

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

Parameter / Switch	Description	Default value	Setting for safe operation
Name	Resource name		Arbitrary
System ID [SRS]	System ID of the resource 1...65 535 The system ID must have a value different from the default value, otherwise the project cannot be executed!	60 000	Unique value within the controller network.
Safety Time [ms]	Safety time in milliseconds 20...22 500 ms	600 ms	Application-specific
Watchdog Time [ms]	Watchdog time in milliseconds 8...5000 ms	200 ms/ 100 ms[1)]	Application-specific
Main Enable	ON: The following switches/parameters can be changed during operation (= RUN) using the PADT. OFF: The parameters cannot be changed during operation.	ON	OFF is recommended
Autostart	ON: If the processor system 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 using the PADT is permitted in the states RUN or STOP OFF: Start not allowed	ON	Application-specific
Load allowed	ON: Download of the user program permitted OFF: Download of the user program not permitted	ON	Application-specific
Reload Allowed	Not applicable for HIMatrix controllers!	ON	-
Global Forcing Allowed	ON: Global forcing permitted for this resource OFF: Global forcing not permitted for this resource	ON	Application-specific
Global Force Timeout Reaction	Specifies how the resource should behave when the global force time-out has expired: ▪ Stop Forcing ▪ Stop resource	Stop Forcing	Application-specific
Max.Com. Time Slice ASYNC [ms]	Highest value in ms for the time slice used for communication during a resource cycle, see the Communication Manual (HI 801 101 E), 2...5000 ms	10 ms	Application-specific
Target Cycle Time [ms]	Not applicable for HIMatrix controllers!	0 ms	-
safeethernet CRC	Irrespective of the setting configured in this box, the CRC for safe**ethernet** is always calculated for HIMatrix controllers with the procedure used in SILworX V.2.36. Take this into account when connecting to HIMax systems!	Current Version	-
Multitasking Mode	Not applicable for HIMatrix controllers!	Mode 1	-

Parameter / Switch	Description	Default value	Setting for safe operation
Sum of UP Max. Duration for Each Cycle [µs]	Sum of the values indicated for *Max. Duration for each Cycle [µs]* in all the user programs. Not changeable		-
Target Cycle Time Mode	Not applicable for HIMatrix controllers!	Fixed	-
Minimum configuration version	Not applicable for HIMatrix controllers! (it always corresponds to SILworX version 2, irrespective of the setting)	SILworX V2	-
[1] 200 ms with controllers, 100 ms with remote I/Os.			

Table 27: System Parameters of the Resource - Versions Beyond 7

System Parameters of the Resource - Versions Prior to 7

Switch	Function	Default value	Setting for safe operation
Main enable	The following switches/parameters can be changed during operation (= RUN) using the PADT.	ON	OFF [1]
Autostart	Automatic start after powering on the controller.	OFF	ON / OFF [2]
Start/Restart allowed	Cold start, warm start or hot start with PADT in RUN or STOP.	ON	OFF [1]
Load allowed	Load enable for a user program.	ON	ON
Test mode allowed	The test mode is permitted or not for the user program. During the test mode, the program processing is frozen. The outputs remain active and the program processing can be continued in single steps.	OFF	OFF
Changing the variables in the OLT allowed	The values of variables can be visualized and modified in the online test (OLT) fields of the logic.	OFF	OFF [3]
Forcing allowed	Entering and activating values for the PES variables/signals are allowed, irrespective of the current value of the process or logic signal.	OFF	Defined by the test institute.
Stop at Force Timeout	It stops the CPU upon expiration of the force time.	ON	Defined by the test institute.

[1] In the RUN state, it is only possible to switch to the OFF value.

[2] The setting ON or OFF depends on the application.

[3] In the RUN state, it is only possible to switch to ON.

Table 28: System Parameters of the CPU

For forcing, additional switches and parameters can be preset.

Hardware System Variables - Versions Beyond 7

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	OFF	Application-specific
Spare 0 ... 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	OFF	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.	OFF	Application-specific
Reload Deactivation	Prevents execution of reload	OFF	Application-specific

Table 29: Hardware System Variables

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

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

8.2.3 Locking and Unlocking the Controller

Locking the controller locks all functions and prevents users from accessing them during operation. This also protects against manipulations to the user program. The scope of the locking procedure should be viewed relative to the safety requirements for using the PES, but might also be coordinated with the test authority responsible for the final system acceptance test.

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

i The locking and unlocking functions are only available with controllers and the F3 DIO 20/8 01 remote I/O, but not with the remaining remote I/Os!

Versions beyond 7

Three system variables serve for locking:

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

Table 30: System Variables for Locking and Unlocking the PES

If all three system variables are ON: no access to the controller is possible. In this case, the controller can only adopt the STOP/VALID CONFIGURATION state after a restart. Then loading a new user program is possible.

Example for using these system variables:

To make a controller lockable

1 Define a global variable of type BOOL and set its initial value to OFF.
2 Assign global variables to the three system variables *Read only in Run*, *Reload Deactivation*, and *Force Deactivation*.
3. Assign the global variable to the channel value of a digital input.
4. Connect a key switch to the digital input.
5. Compile the program, load it on the controller, and start it.

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

Versions Prior to 7

Locking procedure - Proceed as follows to lock the PES:

To lock a controller

1. Set the following values in the controller prior to compiling (see also Chapter Code Generation):

Main Enable	set to	ON
Forcing allowed	set to	OFF (depending on the application)
Test mode allowed	set to	OFF
Start/Restart allowed	set to	ON
Load allowed	set to	ON
Autostart	set to	ON / OFF
Stop at Force Timeout	set to	ON (depending on the application)

2. After loading and starting, change the switches in the online controller following the specified order:

Start/Restart allowed	set to	OFF
Load allowed	set to	OFF
Main Enable	set to	OFF

i The following switches may only be set to different values after receiving consent from the test authority:

Forcing allowed	set to	ON
Stop at Force Timeout	set to	ON / OFF
Start/Restart allowed	set to	ON
Autostart	set to	ON

The controller is locked.

Unlocking procedure - To be able to unlock the controller (Main Enable set to ON), the controller must be in STOP. Main Enable cannot be activated while the controller is operating (RUN state), but it can be deactivated.

To allow a restart after the CPU initialization (e.g., after voltage drops), proceed as follows when unlocking the PES:

To unlock the controller

1. Set Main Enable to ON.
2. Set Start/Restart to ON.
3. Start the user program.

The controller is unlocked.

8.2.4 Code Generation

After entering the complete user program and the I/O assignments of the controller, generate the code. The code generator creates the configuration CRC. This is a signature for the entire configuration of CPU, inputs/outputs and communication, and is issued as a 32-bit, hexadecimal code. The signature includes all of the configurable or modifiable elements such as the logic, variable or switch parameter settings.

To ensure that the not safe PC has no influence on the process, generate the code a second time. The two resulting configuration CRCs must be identical.

To generate the code for safety-related operation

1. Start the code generator to create the code with the configuration CRC.
   ☑ Executable code 1 with CRC 1.
2. Start the code generator once again to create the code with the configuration CRC.

☑ Executable code 2 with CRC 2.

3. Compare CRC 1 with CRC 2.
   ☑ The two CRCs are identical.

The generated code may be used for safety-related operation and for the system's certification performed by the test authority.

8.2.5 Loading and Starting the User Program

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

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

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

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

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

8.2.6 Forcing

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

- a physical input
- communication
- a logic operation.

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

Forcing is used for the following purposes:

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

⚠ WARNING

Physical injury due to forced vaules is possible!

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

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

NOTE

Use of forced values can disrupt the safety integrity!

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

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

This document is available on the TÜV homepage:

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

8.2.7 Forcing - Versions Beyond 7

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.

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

The force values of global variables are only valid until the user program overwrites the values!
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 connected to forced global variables may therefore show the forced value even if a value created by the user program is already taken into account in the following calculations or is effective on an output.

8.2.8 Time Limits

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

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

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

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

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

8.2.9 Restricting the Use of Forcing

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

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

⚠ WARNING

Use of forced values can disrupt the safety integrity!

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

8.2.10 Force Editor

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

For each variable, the following can be set:

- Force Value
- Force Switch

The force switch enables the forcing of variables. The forcing of all variables with an active force switch can only begin when local or global forcing has been started.

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

It is possible to start forcing with or without a specified time limit. If none of the restrictions apply, all variables with an active force switch are set to their force values..

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

For more information about the Force Editor and forcing, refer to the SILworX online help.

8.2.11 Forcing Signals - Versions Prior to 7 (possible with controllers and the F3 DIO 20/8 01 remote I/O)

Forcing a signal means that any values can be entered for the signal. A signal can be associated with an input or output, or can be used for communication.

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 CPU 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) See also the warning message below: 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 31: Force Switches and Parameters

Enter the value -1 for forcing without time limit.

Force Allowed - CPU Switch

- Not set:
  - Forcing is not possible (default setting).

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

Forcing is terminated and the process value reactivated upon expiration of the force time or when forcing is intentionally stopped.

NOTE

Physical injury due to forced signals is possible!

Only set the *Forcing Allowed* switch after receiving consent from the test authority responsible for the final system acceptance test.

Provided that *Stop at Force Timeout* is activated 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.

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.

Forcing can be time-limited. If the Force Time is overrun, the user can define if the CPU should enter the STOP state or the force value should no longer be valid such that normal operation can be continued. In any case, overrunning the force time affects the user program and thus the process.

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

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.

NOTE

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!

8.2.12 Online Test

The Online Test function allows the user to display online test fields (OLT fields) in the logic and to set variables while the controller is operating.

Versions Beyond 7

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

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

Versions Prior to 7

The CPU switch *Changing the variables in the OLT allowed* defines the possibility to change variablen online.

State	Effect
OFF	Variables cannot be changed. It is possible to display the values of variables in the OLT fields, but not to change them.
ON	Variables can be changed. The values of variables can be entered manually and set in the corresponding OLT fields while the program is running. However, the set value only applies until the program logic replaces it with a new value.

Table 32: CPU Switch *Changing the variables in the OLT allowed* - Version Prior to 7

For more information on how to use OLT fields, enter *OLT field* in the ELOP II Factory online help.

8.2.13 Program Documentation for Safety-Related Applications

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

- Interface declaration
- Signal list
- Logic
- Description of data types
- Configurations for system, modules and system parameters
- Network configuration
- List of signal cross-references
- Code generator details

This documentation is required for the acceptance test of a system subjected to approval by a test authority (e.g., TÜV). This acceptance test only applies to the user functionality, but not to the safety-related modules and automation devices of the HIMatrix system that have already been approved.

9 Configuring Communication

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

This data exchange can occur in either read-only or read/write mode.

9.1 Standard Protocols

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

⚠ DANGER

Physical injury due to usage of unsafe import data

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

Depending on the controller variant, the following standard protocols are available:

- SNTP
- Send/Receive TCP
- Modbus (master/slave)
- PROFIBUS DP (master/slave)
- INTERBUS.

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

Safety-related communication via safe**ethernet** is certified up to SIL 3.

Use the safe**ethernet** Editor or P2P Editor to configure how safety-related communication is monitored.

For determining the *Receive Timeout* and *Response Time* safe**ethernet** parameters, the following condition applies:
The communication time slice must be sufficiently high to allow all the safe**ethernet** connections to be processed within one CPU cycle.

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

NOTE

Unintentional transition to the safe state possible!

***ReceiveTMO* is a safety-related parameter!**

If all values must be transferred, the value of a signal must either be present for longer than *ReceiveTMO* or it must be monitored using a loop back.

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

9.2.1 Receive Timeout

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

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

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

Since *ReceiveTMO* is a safety-relevant component of the Worst Case Reaction Time T_R (see Chapter 9.2.3 et seqq.), its value must be determined as described below and entered in the safe**ethernet** Editor.

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

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

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

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

NOTE

The maximum value permitted for *ReceiveTMO* depends on the application process and is configured in the safe**ethernet Editor, along with the expected maximum response time and the profile.**

9.2.2 Response Time

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

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

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

ResponseTime ≤ ReceiveTMO / n

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

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

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

min. Response Time ≤ ReceiveTMO / 2 ≥ 2*Delay + 2.5*max. Cycle Time

If this condition is met, the loss of at least one data packet can be intercepted without interrupting the safe**ethernet** connection.

i If this condition is **not met**, the availability of a safe**ethernet** connection can only be guaranteed in a collision and fault-free network. However, this is not a safety problem for the processor module!

i Make sure that the communication system complies with the configured response time!

If this conditions cannot always be ensured, a corresponding connection system variable for monitoring the response time is available. If the measured response time is not seldom exceeded for over the half P2P ReceiveTMO, the configured response time must be increased.

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

NOTE

In the following examples, the formulas for calculating the worst case reaction time only apply for a connection with HIMatrix controllers if the parameter

safety time = 2 * watchdog time

has been set in the systems.

9.2.3 Maximum Cycle Time of the HIMatrix Controller

To determine the maximum cycle time for a HIMatrix controller, HIMA recommends proceeding as follows:

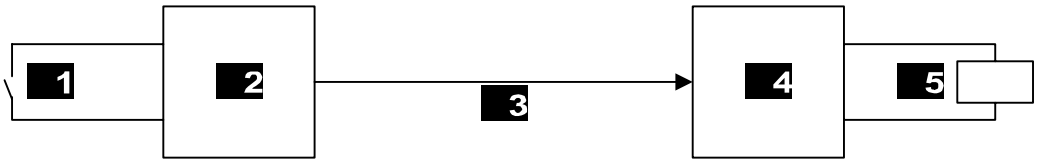
To determine the maximum cycle time for the HIMatrix controller

1. Use the system under the maximum load. In the process, all communication connections must be operating both via safeethernet and standard protocols. Frequently read the cycle time in the Control Panel and note the maximum cycle time.
2. Repeat step 1 for the next communication partner (i.e., the second HIMatrix controller).
3. The required maximum cycle time is the greater of the two time values ascertained.

The maximum cycle time was determined and is used in the following calculations.

9.2.4 Calculating the Worst Case Reaction Time

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



1 Input
2 HIMatrix Controller PES 1
3 Safety-Related Protocol
4 HIMatrix Controller PES 2
5 Output

Figure 4: Reaction Time with Interconnection of Two HIMatrix Controllers

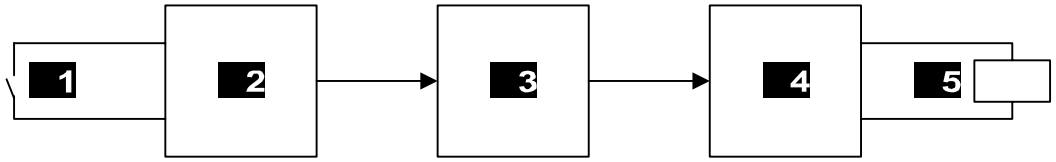
$T_R = t_1 + t_2 + t_3$

T_R Worst case reaction time
t_1 2 * watchdog time of the HIMatrix controller 1.
t_2 ReceiveTMO
t_3 2 * watchdog time of the HIMatrix controller 2.

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

9.2.5 Calculating the Worst Case Reaction Time with two Remote I/Os

The worst case reaction time TR is the time between a change on the sensor input signal of the first HIMatrix PES or Remote I/O (e.g., F3 DIO 20/8 01) and a reaction on the corresponding output of the second HIMatrix PES or Remote I/O. It is calculated as follows:



1 Input
2 Remote I/O 1
3 HIMatrix Controller
4 Remote I/O 2
5 Output

Figure 5: Reaction Time with Remote I/Os

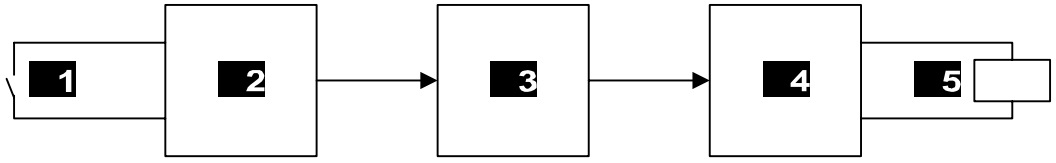
$T_R = t_1 + t_2 + t_3 + t_4 + t_5$

T_R Worst case reaction time
t_1 2 * watchdog time of remote I/O 1
t_2 $ReceiveTMO_1$
t_3 2 * watchdog time of the HIMatrix controller.
t_4 $ReceiveTMO_2$
t_5 2 * watchdog time of remote I/O 2

Note: Remote I/O 1 and remote I/O 2 can also be identical. The time values still apply if a HIMatrix controller is used instead of a remote I/O.

9.2.6 Calculating the Worst Case Reaction Time with two HIMatrix and one HIMax Controller

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



1 Input
2 HIMatrix Controller 1
3 HIMax Controller
4 HIMatrix Controller 2
5 Output

Figure 6: Reaction Time with Two HIMatrix Controllers and One HIMax Controller

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

T_R	Worst case reaction time
t_1	2 * watchdog time of the HIMatrix controller 1.
t_2	ReceiveTMO$_1$
t_3	2 * watchdog time of the HIMax controller.
t_4	ReceiveTMO$_2$
t_5	2 * watchdog time of the HIMatrix controller 2.

Remark: HIMatrix controller 1 and HIMatrix controller 3 can also be identical.

9.2.7 Terms

ReceiveTMO	Monitoring time of controller 1 within which a correct response from controller 2 must be received. Once the time has expired, safety-related communication is terminated.
Production Rate	Minimum interval between two data transmissions.
Watchdog Time	Maximum permissible duration of a controller's RUN cycle (cycle time).
Worst case reaction time	The worst case reaction time is the time between a change in a physical input signal of controller 1 and a reaction on the corresponding output of controller 2.

9.2.8 Assigning safe**ethernet** Addresses

Take the following points into account when assigning network addresses (IP addresses) for safe**ethernet**:

- The addresses must be unique within the network used.
- When connecting safe**ethernet** to another network (company-internal LAN, etc.), make sure that no disturbances can occur. Potential sources of disturbances include:
  - The data traffic.
  - Coupling with other networks (e.g., Internet).

In these cases, implement suitable measures to counteract against such disturbances using Ethernet switches, firewall and similar.

10 Use in Fire Alarm Systems

The HIMatrix systems may be used in fire alarm systems in accordance with DIN EN 54-2 and NFPA 72, if line monitoring is configured for the inputs and outputs.

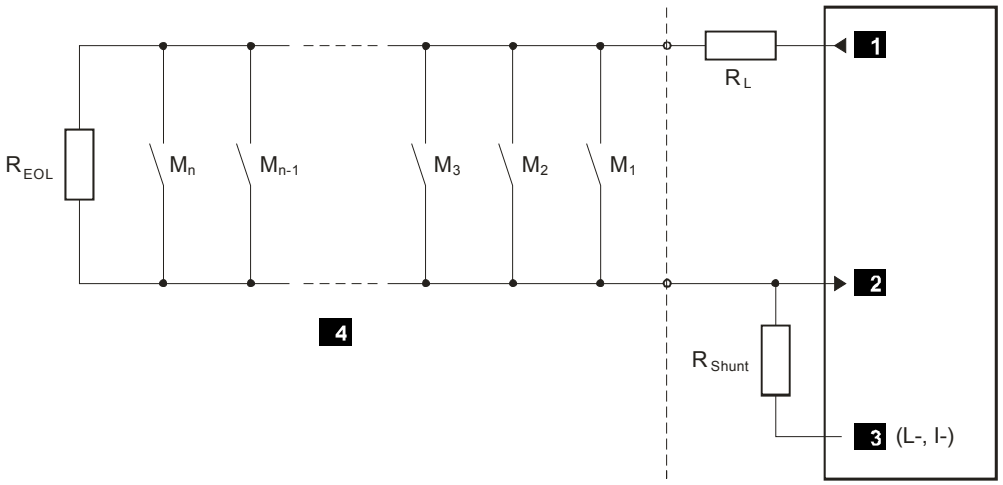
In this case, the user program must fulfill the requirements specified for fire alarm systems in accordance with the standards previously mentioned.

The maximum cycle time of 10 seconds required by DIN EN 54-2 for fire alarm systems and the safety time of 1 second (fault reaction time) required in certain cases, can be easily met since the cycle times for these systems reside in the millisecond range.

According to EN 54-2 the fire alarm system has to be in the fault report state within 100 seconds after the HIMatrix system has received the fault message.

The fire alarms are connected using the energize to trip principle with line monitoring for the detection of short-circuits and open-circuits. To this end, the following devices and modules may be used:

- The digital and analog inputs of the F35 controller.
- The analog inputs of the F3 AIO 8/4 01 remote I/O.
- The digital inputs and outputs of the F3 DIO 16/8 01 and F3 DIO 8/8 01 remote I/Os.
- The AI 8 01 and MI 24 01 input modules of the F60



1 Sensor Supply
2 Analog Input
3 Ground
4 Detection Loop

M Fire detector
R_{EOL} Terminating resistor on the last loop sensor
R_L Limit for the maximum loop current
R_{Shunt} Shunt

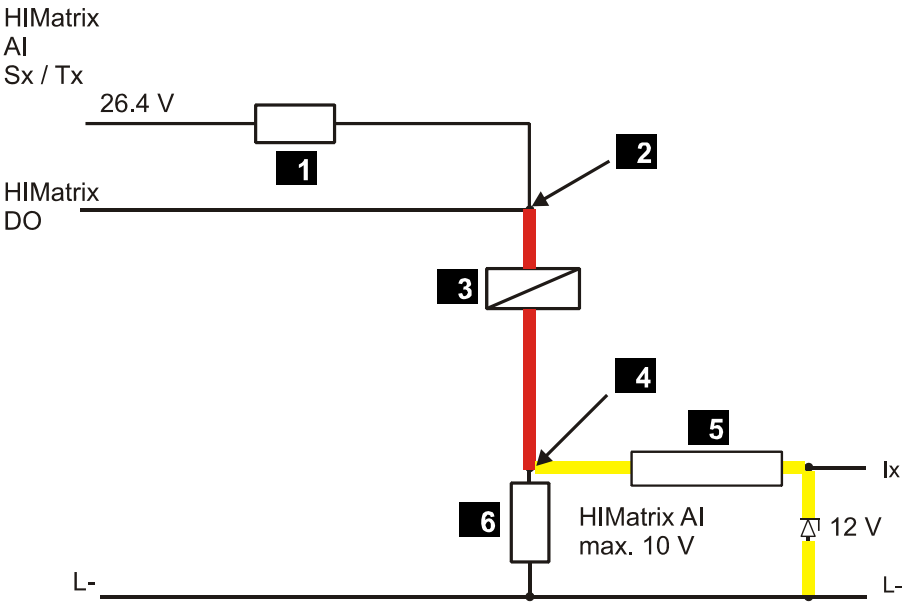
Figure 7: Wiring of Fire Alarms

For the application, the R_{EOL}, R_L and R_{Shunt} resistors must be calculated as dictated by the sensors in use and the number of sensors per detection loop. Refer to the data sheet from the sensor manufacturer for the necessary data.

The alarm outputs for controlling lamps, siren, horns etc. are operated in accordance with the energize to trip principle. These outputs must be monitored for short-circuits and open circuits. This can be done by returning the output signals directly from the actuator to the inputs.

The current in the actuator circuit can be monitored via an analog input using an appropriate shunt. Z-diodes and series resistor connected in series protect the input against overvoltage if a short-circuit occurs.

For an explicit detection of open-circuits (with de-energized DO outputs), a transmitter supply is required in addition to the analog inputs (see draft below):



1 Resistor
2 Field Clamp Terminal
3 Actuator
4 Field Clamp Terminal
5 Diode Resistor
6 Shunt
Line Monitoring Range (Open-Circuits and Short-Circuits)
Protective Circuit for Short-Circuit

Figure 8: Example of Open-Circuit and Short-Circuit Monitoring of Digital Outputs (Actuator Circuit)

For an example of how to configure the open-circuit and short-circuit monitoring of actors using analog inputs, refer to Chapter *Line Monitoring* in the HIMatrix F35 Manual (HI 800 149).

A suitable user program can be used to control visual display systems, indicator light panels, LED indicators, alphanumeric displays, audible alarms, etc.

The routing of fault signals via the input and output channels or to transmission equipment for fault signals must occur in accordance with the de-energize to trip principle.

Fire alarms can be transmitted from one HIMatrix system to a different system using the existing Ethernet communication standard (OPC). Any communication loss must be reported.

HIMatrix systems that are used as fire alarm systems must have a redundant power supply. Precautionary measures must also be taken against power supply drops, e.g., the use of a battery-powered horn. Uninterrupted operations must be ensured while switching from the main power supply to the back-up power supply. Voltage drops for a duration of up to 10 ms are permitted.

If system failures occur, the operating system writes to the system signal or system variable assigned in the user program. This allows the user to program fault signaling for faults detected by the system. If a fault occurs, the HIMatrix system switches off the safety-related inputs and outputs with the following effects:

- The low level is processed in all channels of the faulty inputs.
- All channels of the faulty outputs are switched off.

Appendix

Increasing the SIL of Sensors and Actuators

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

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

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

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

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

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

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

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

Glossary

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

Index of Figures

Index of Tables

Index

HIMA Paul Hildebrandt GmbH + Co KG
P.O. Box 1261
68777 Brühl, Germany
Tel: +49 6202 709-0
Fax: +49 6202 709-107
E-mail: info@hima.com Internet: www.hima.com

(1018)