Manual

# HIQuad®

Safety-Related Controller

Operating System Functions

## Contact

HIMA Paul Hildebrandt GmbH

P.O. Box 1261

68777 Brühl, Germany

Phone: +49 6202 709-0

Fax: +49 6202 709-107

E-mail: info@hima.com

| Document designation | Description |
|---|---|
| HI 800 104 D, Rev. 3.01 (2019) | German original document |
| HI 800 105 E, Rev. 3.01.00 (2026) | English translation of the German original document |

# Table of Contents

# 1 Operating System Manual

This manual describes the functions of the operating system for the H41q/H51q system.

This manual is intended to assist in planning and engineering the hardware and software for safety-related automation systems.

## 1.1 Validity and Current Version

The most current version of the operating system manual is also valid for previous operating system versions. Special features of the individual versions are mentioned in this document.

The the latest version of this manual can be obtained upon request by sending an e-mail to: documentation@hima.com.

Extensive changes performed to the manual are identified by a revision status, less extensive changes by an issue status. The revision status is located on the front side behind the document HI number, the issue status is located on the rear side.

## 1.2 Target Audience

This manual is aimed at the planners, design engineers and programmers of automation devices concerned. Specialized knowledge of safety-related automation systems is required.

## 1.3 Writing Conventions

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

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

Safety notices and operating tips are specially marked.

### 1.3.1 Safety Notices

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

The safety notices are represented as described below.

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

The signal words have the following meanings:

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

**SIGNAL WORD**

**Type and source of risk!**

**Consequences arising from non-observance.**

**Risk prevention.**

---

**NOTICE**

**Type and source of damage!**

**Damage prevention.**

---

### 1.3.2 Operating Tips

Additional information is structured as presented in the following example:

i The text giving additional information is located here.

Useful tips and tricks appear as follows:

**TIP** The tip text is located here.

# 2 Safety

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

The product is operated with SELV or PELV. No imminent danger results from the PES itself. Use in the Ex zone is only permitted if additional measures are taken.

## 2.1 Intended Use

### 2.1.1 Application Area

The safety-related automation devices H41q, H41qc and H51q can be used in applications up to SIL 3 (IEC 61508), Cat. 4 or Pl e (ISO 13849-1).

All input and output modules can be used with a redundant as well as with a 1-channel central module structure.

When implementing safety-related communications between various systems, ensure that the overall response time does not exceed the process safety time. The calculation basis provided in the safety manual (HI 800 013 E) must be applied.

Only devices with electrically protective separation may be connected to the communication interfaces.

The H41q/H51q systems are certified for use in process controllers, protective systems, burner systems and machine controllers.

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

The automation devices are 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 power to perform its safety function.

Thus, if faults occur, the de-energized state is adopted as the safe state for inputs and outputs.

#### 2.1.1.2 Application in Accordance with the Energize to Trip Principle

The H41q/H51q controllers can also be used in applications that operate in accordance with the energize to trip principle.

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

To this end, the H41q/H51q controllers are tested and certified for use in fire alarm and fire-fighting systems in accordance with EN 54 and NFPA 72. To contain the hazard, these systems must be able to adopt an active state on demand.

#### 2.1.1.3 Explosion Protection

The safety-related automation devices H41q and H51q are suitable for mounting in zone 2. The declarations of conformity are annexed to the corresponding data sheets.

The operating conditions provided in the next sections must be observed!

#### 2.1.1.4 Use in Fire Alarm Systems

All H41q/H51q systems with analog inputs can be used in fire alarm systems in accordance with DIN EN 54-2 and NFPA 72.

The operating conditions provided in the next sections must be observed!

### 2.1.2 Non-Intended Use

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

With no safety-related fieldbus protocols, the fieldbus interfaces cannot ensure safety-related communication.

### 2.1.3 Test Requirements

#### 2.1.3.1 Environmental Requirements and Specifications

When using the safety-related H41q/H51q control systems, the following general conditions must be met:

| Condition type | Condition content |
|---|---|
| Protection class | Protection class II in accordance with EN 61131-2 |
| Ambient temperature | 0…+60 °C |
| Storage temperature | -40…+80 °C<br>(with battery: only -30…+75 °C) |
| Pollution | Pollution degree II |
| Installation height | < 2000 m |
| Enclosure | Standard: IP20<br>If required by the relevant application standards (e.g., EN 60204), the system must be installed in an enclosure with the specified degree of protection (e.g., IP54). |
| Power supply input voltage | 24 VDC |

Table 1: Environmental Requirements

Refer to the relevant data sheets for deviations.

The safety-related control systems H41q/H51q have been developed to meet the following standards for EMC, climatic and environmental requirements.

| Standard | Content |
|---|---|
| EN 61131-2:2007 | Programmable controllers, Part 2<br>Equipment requirements and tests |
| IEC/EN 61000-6-2:2005 | EMC<br>Generic standards, Part 6-2<br>Immunity for industrial environments |
| IEC/EN 61000-6-4:2007 | Electromagnetic compatibility (EMC)<br>Generic emission standard, industrial environments |

Table 2: Standards

### 2.1.3.2 Climatic Tests

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

| EN 61131-2 | Climatic tests |
|---|---|
| | Dry heat and cold; withstand tests:<br>+70 °C / -40 °C, 16 h, +85 °C, 1 h<br>Power supply not connected |
| | Temperature changes, withstand test:<br>Fast temperature changes: -40 °C / +70 °C, power supply not connected |
| | Immunity test<br>Slow temperature changes: -10 °C / +70 °C power supply connected |
| | Cyclic damp-heat; withstand tests:<br>+25 °C / +55 °C, 95 % relative humidity<br>Power supply not connected |
| EN 54-2 | Damp-heat<br>93 % relative humidity, 40 °C, 4 days in operation<br>93 % relative humidity, 40 °C, 21 days, power supply not connected |

Table 3: Climatic Requirements

### 2.1.3.3 Mechanical Tests

The following table lists the most important tests and limits for mechanical requirements:

| EN 61131-2 | Mechanical Tests |
|---|---|
| | Vibration immunity test:<br>5…9 Hz / 3.5 mm<br>9…150 Hz / 1 g, HIQuad in operation, 10 cycles per axis |
| | Shock immunity test:<br>15 g, 11 ms, HIQuad in operation, 3 shocks per axis and direction (18 shocks) |

Table 4: Mechanical Tests

### 2.1.3.4 EMC Tests

Refer to the EC declaration conformity to find out which test conditions are complied with.

All modules of the H41q and H51q systems meet the requirements of the EMC Directive of the European Union and are labeled with the CE mark.

The controllers respond safely to interferences exceeding the specified limits.

### 2.1.3.5 Power Supply

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

| EN 61131-2: | Verification of the DC supply characteristics |
|---|---|
| | Alternatively, the power supply unit must comply with the following standards:<br>▪ EN 61131-2 or<br>▪ SELV (Safety Extra Low Voltage, EN 60950) or<br>▪ PELV (Protective Extra Low Voltage, EN 60742) |
| | The H41q and H51q systems must be fuse-protected as specified in the data sheets. |
| | Voltage range test:<br>24 VDC, -20…+25 % (19.2…30.0 VDC). |
| | Momentary external current interruption immunity test: DC, PS 2: 10 ms |
| | Reversal of DC power supply polarity test:<br>Refer to the corresponding chapter of the catalog or data sheet of the power supply module. |
| | Backup battery, withstand test:<br>Test B, 1000 h, lithium battery as back-up battery |

Table 5: Verification of the DC Supply Characteristics

## 2.2 Tasks of Operators and Machine and System Manufacturers

Operators as well as machine and system manufacturers are responsible for ensuring that H41q/H51q systems are safely operated in automated systems and plants.

The machine and system manufacturers must validate that the H41q/H51q systems are properly programmed.

## 2.3 ESD Protective Measures

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

**NOTICE**

**Electrostatic discharge!**

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

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

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

## 2.4 Personnel Qualifications

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

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

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

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

## 2.5 Residual Risk

No imminent risk results from a HIMA system itself.

Residual risk may result from:

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

## 2.6 Safety Precautions

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

## 2.7 Emergency Information

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

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

# 3 Operating System Functions

The operating system includes all basic functions of the HIMA controller (programmable electronic system, PES). Which function the PES should perform is specified in the user program. ELOP II serves as planning tool for creating the user program. ELOP II translates the user program with a code generator into the machine code and transfers this machine code to the central module's flash EPROM via a serial interface or via TCP/IP.

The following table specifies all basic functions of the operating system and the requirements in the user program:

| Operating system functions | Requirements in the user program |
|---|---|
| Cyclic processing of the user program: | Function blocks, functions and variables |
| HIMA standard function blocks (included in the operating system) | Standard function blocks, variables |
| Configuration of the controller (1 or 2 I/O buses, number of the power supply units, etc.) | Definition in the resource type |
| Reload of the user program | Possible, if all restrictions are complied with, see the ELOP II manual (CD) |
| Tests in the central area and I/O bus | - - |
| Tests of I/O modules (depending on the type) | Type of I/O module |
| Response in the event of a fault:<br>Noise blanking | Preset or configurable<br>Configurable (resource properties) |
| Diagnostic indicators | - - -<br>see Chapter 12 |
| Diagnostic mode for testable I/O modules | HZ-DOS-3 function block, non-safety-related diagnostics |
| Communication to the PADT via serial interface or Ethernet (via F 8627X or F 8628X)<br>Actions permitted during operation | ELOP II<br><br>Definition in the resource type |
| PES master, not safety-related | Variable declaration, HIPRO-N communication |
| PES master, safety-related | Variable declaration, safety-related HIPRO-S communication |
| Third-party coupling to Modbus master system | Variable declaration, third-party coupling (communication with BUSCOM variables) |
| Third-party coupling to Modbus slave system | HK-MMT-3 function block, variable declaration, Third-party coupling (serial communication with BUSCOM variables) |
| Third-party coupling to master system with 3964R protocol | Variable declaration, third-party coupling |
| Logic-plan-controlled logging | Variable declaration, event-driven protocol texts<br>Protocol texts, see Chapter 6.7. |
| Self-education: User program adopted from redundant central module | - - - |

Table 6: Functions of the Operating System

## 3.1 User Program Size

The maximum user program size is:

**1020 kBytes program, 320 kBytes data**

To use this extended memory area, 3 requisites must be met:

1. Use of an E resource type, e.g., central modules such as the F 8650X.
2. Use of the operating system BS41q/51q V7.0-8 (07.30) described in this document.
3. Use of ELOP II as of version 3.5 with the corresponding compiler.

If one of these conditions is not met, the user program is limited to the following size:

**444 kBytes program, 96 kBytes data**

Size of the buffer space: 512 bytes (user data)

## 3.2 Cycle Processing

The operating system constantly executes the user program cyclically. The sequence order is represented in simplified form:

1. Reading the input signals.
2. Editing the logic functions in accordance with IEC 61131-3, Section 4.1.3.
3. Writing output signals.

The following basic functions are also executed:

- Comprehensive self-tests.
- Test of I/O modules during operation.
- Data transmission and comparison.

Redundant central modules are synchronized after each cycle phase. Communication and the self-test parts that are not executed in every cycle, do not depend on the phases.

For further information on additional test routines and fault responses reactions, refer to the safety manual (HI 800 013 E).

A cycle is processed in 7 phases.

| Cycle processing | PES with<br>2 central modules, 1 I/O bus | PES with<br>2 central modules, 2 I/O buses or<br>1 central module, 1 I/O bus |
|---|---|---|
| | H41q-H, -HS<br>H41qe-H, -HS<br>H51q-H, -HS<br>H51qe-H, -HS | H41q-M, -MS, -HR, -HRS<br>H41qe-M, -MS, -HR, -HRS<br>H51q-M, -MS, -HR, -HRS<br>H51qe-M, -MS, -HR, -HRS |
| Phase 1 | Cyclic self-tests.<br>Cyclic consistency check.<br>Master change of the central modules. | Cyclic self-tests.<br>Cyclic consistency check. |
| Phase 2 | Process all write transmissions (also via Ethernet module).<br>Read and test the inputs from master central module.<br>Receive data written to import variables. | Process all write transmissions (also via Ethernet module).<br>Read and test the inputs.<br><br>Receive data written to import variable.s |
| Phase 3 | Transfer the input values to the slave central module. | Transfer the input values to the other central module, if 2 central modules are used. |
| Phase 4 | Copy all import variables to internal variables.<br><br>**Process the user logic.**<br><br>Copy all the internal variables to export variables.<br>Write the export data to the Ethernet module(s).<br>Start importing data from the Ethernet module(s).<br>Compare memory cyclically. | Copy all import variables to internal variables.<br><br>**Process the user logic.**<br><br>Copy all the internal variables to export variables.<br>Write the export data to the Ethernet module(s).<br>Start importing data from the Ethernet module(s).<br>Compare memory cyclically, if 2 central modules are available. |
| Phase 5 | Exchange and compare output values. | Exchange and compare output values, if 2 central modules are available. |
| Phase 6 | Write the output signals via master central module. | Write the output signals. |
| Phase 7 | Read back the hardware outputs and compare them with the logic output signals via slave central module:<br>If the output signals are not identical, shut down the faulty output module.<br>Transition to the next cycle (Phase 1). | Read back the hardware outputs and compare them with the logic output signals via the central module:<br>If the output signals are not identical, shut down the faulty output module.<br>Transition to the next cycle (Phase 1). |

Table 7: Processing of a Cycle

## 3.3 Modes of Operation

The table specifies all possible modes of operation of the central module.

| Mode of operation | Indicators | Description |
|---|---|---|
| Run operation | RUN | Normal state<br>The central modules process the user program. |
| Mono operation | MONO | With a redundant system: only 1 central module operates properly, the other is in the stop or error stop state. |
| Error stop | STOP | The PES entered the safe state due to a problem. |
| Stop (safe state) | STOP | The user stopped the PES. All outputs are in the safe state. |
| Stop (freezing outputs) | STOP | The user stopped the PES. All outputs are in the state entered in the last user program cycle. This mode of operation is used to test the outputs and field conditions. |
| Break points | STOP | During the online test, it is possible to set the break points for types, instances of functions or function blocks. In each cycle, the user program either stops in every instance or in only one instance such that it is possible to verify the variable values. |
| Single-step | STOP | It is possible to execute each individual user program cycle, e.g., for testing purposes. |

Table 8: Modes of Operation of the Central Module

## 3.4 Standard Function Blocks

The following list specifies the HIMA standard function blocks that can be used irrespective of the input and output modules.

The standard function blocks that must be used with input and output modules are indicated in Chapter 5.1.

Refer to the safety manual (HI 800 013 E) or to the online help of the corresponding function block for a description of its function.

| Type | Function | TÜV test |
|---|---|---|
| H8-UHR-3 | Date and time | • |
| HA-PID-3 | PID controller | • |
| HK-AGM-3 | H51q PES master monitoring | • |
| HK-COM-3 | Monitoring of communication modules | • |
| HK-LGP-3 | LCL evaluation and configuration | • |
| HK-MMT-3 | Modbus master | • |
| HZ-DOS-3 | Non safety-related diagnostics | • |
| HZ-FAN-3 | Fault indicators for testable I/Os | • |

Table 9: Standard Function Blocks not Depending on the I/O Level

In the TÜV test column, the dot (•) indicates that the corresponding function block can be used in safety-related controllers and a corresponding TÜV safety case is available.

# 4 Operating System Versions

This chapter indicates the current version and describes the assignment of the versions to central module types and other firmware versions.

## 4.1 Current Version

The operating system program is loaded into a 1 MB flash EPROM. The operating system has the following designation:

**BS41q/51q V7.0-8 (09.12)**

Each new operating system is identified by a specific revision status. An additional identification option is the operating system signature, which can be displayed on the diagnostic panel while the PES is operating. For the signature, refer to the *Revision List of Devices and Firmware of H41q/H51q Systems of HIMA Paul Hildebrandt GmbH*.

Certificate number 01/205/5086.01/16.

## 4.2 Operating System Versions and Types for the Central Modules

| System family | H41q | | H51q | |
|---|---|---|---|---|
| System designation | H41q-M [1]<br>H41q-H [1]<br>H41q-HR [1] | H41q-MS<br>H41q-HS<br>H41q-HRS | H51q-M [1]<br>H51q-H [1]<br>H51q-HR [1] | H51q-MS<br>H51q-HS<br>H51q-HRS |
| Central module | F 8653 [1]<br>F 8653A [1]<br>F 8653E [1]<br>F 8653X [1] | F 8652 [1]<br>F 8652A [1]<br>F 8652E [1]<br>F 8652X | F 8651 [1]<br>F 8651A [1]<br>F 8651E [1]<br>F 8651X [1] | F 8650 [1]<br>F 8650A [1]<br>F 8650E [1]<br>F 8650X |
| Operating system | BS41q/51q V7.0-8 | | | |
| TÜV test | | • | | • |
| [1]) Discontinued. | | | | |

Table 10: Assignment of Operating Systems and Central Modules

The user programs of the E resources can only be run on the current central modules with the extension E or X.

The hardware revision status of the F 8650 (and F 8651) central modules must be AS02 or higher!

## 4.3 Operating System and Other Firmware Versions

| Operating system | CM operating system<br>BS51-CB V6.0-6<br>F 8621A | Communication module<br>F 8625 [1] / F 8626 [1]<br>F 8627(X) / F 8628(X) | Code generator<br>ELOP II RT H41/H51 |
|---|---|---|---|
| BS41q/51q V7.0-7 | (9808) | V1.0 (9835) | V2.0 (NT) |
| BS41q/51q V7.0-7 | (9808) | V1.0 (9835) | V2.0 (NT) |
| BS41q/51q V7.0-7 | (9808) | V1.6 (9918) | V2.0 (NT) |
| BS41q/51q V7.0-7 | (9808) | V1.6 (9918) | V3.0 (NT) |
| BS41q/51q V7.0-7 | (9808) | F 8626: V1.10 (0015)<br>F 8626: V1.10 (0015) | V3.0 (NT) |
| BS41q/51q V7.0-8 | (9808) | F 8625/F 8626: ≥V1.18<br>F 8627/F 8628: ≥ V3.12 | ≥V3.0 (NT) |
| BS41q/51q V7.0-8<br>≥(0410) | ≥(9808) | F 8625/F 8626: ≥V1.18<br>F 8627/F 8628: ≥ V3.12 | ≥V3.5 (NT)<br>Build 6812 IV5 |
| BS41q/51q V7.0-8<br>≥(0515) | ≥(9808) | F 8625/F 8626: ≥V1.18<br>F 8627/F 8628: ≥ V3.12 | ≥V4.1 |
| BS41q/51q V7.0-8<br>≥(05.34) | ≥(9808) | F 8625/F 8626: ≥V1.18<br>F 8627/F 8628: ≥ V3.12<br>F 8627X/F 8628X: ≥V4.x | ≥V4.1<br>Build 6118 |
| BS41q/51q V7.0-8<br>≥(07.14) | ≥(9808) | F 8625/F 8626: ≥V1.18<br>F 8627/F 8628: ≥ V3.12<br>F 8627X/F 8628X: ≥V4.14 | ≥V4.1<br>Build 6134 |
| BS41q/51q V7.0-8<br>≥(07.30) | ≥(9808) | F 8625/F 8626: ≥V1.18<br>F 8627/F 8628: ≥ V3.12<br>F 8627X/F 8628X: ≥V4.22 | ≥V5.1<br>Build 730.1646 IV5 |
| BS41q/51q V7.0-8<br>≥(08.17) | ≥(9808) | F 8625/F 8626: ≥V1.18<br>F 8627/F 8628: ≥ V3.12<br>F 8627X/F 8628X: ≥V4.22 | ≥V5.6<br>Build 1501.9810IV1 |
| **BS41q/51q V7.0-8**<br>**≥(09.12)** | **≥(9808)** | F 8625/F 8626: ≥V1.18<br>F 8627/F 8628: ≥ V3.12<br>**F 8627X/F 8628X: ≥V4.22** | **≥V5.8**<br>**Build 5003.7992IV2** |

1) Discontinued.

Table 11: Assignment of the Operating System and other Firmware

HIMA recommends using the release states marked in bold. The complete functionality is only ensured in this combination.

### NOTICE

**The current operating system only supports central modules with the extension X.**

**Starting with operating system BS41q/51q V7.0-8 ≥ (08.17), central modules of type F 8650X must be used for H51q and central modules of type F 8652X for H41q!**

The following tables specify the permitted combinations of operating systems (CU, CM, EN-M and PB-M), and ELOP II with H51-RT.

| CU operating system BS41q/51q V7.0-7 and V7.0-8 | CM operating system BS51-CM V6.0-6 | PB-M F 8628X | EN-M F 8627X | PB-M F 8628 | EN-M F 8627 | PB-M F 8626 | EN-M F 8625 | ELOP II Basis system and H51-RT |
|---|---|---|---|---|---|---|---|---|
| (9737) | From (9808) to * * | - | - | - | - | - | - | V1.31 (OS/2) |
| (9835)[1] | From (9808) to * * | - | - | - | - | V1.0 | V1.0 | From V2.0 (NT) to * * |
| From (9906)[1] to * *[2] | From (9808) to * * | From V4.14 to * * | From V4.14 to * * | From V2.16 to * * | From V2.8 to * * | From V1.5 to * * | From V1.5 to * * | From V2.0 (NT) to * * |
| Recommended combinations: (09.12)[1] | (9808) | V4.22 | V4.22 | V4.22 | V4.22 | V1.18 | V1.18 | V5.8 Build 5003.7992IV2 |

[1] This operating system requires AS02 for F 8650 and F 8651.

[2] «* *» means: including the current release.

Table 12: Assignment of the H41q/H51q Operating System

Abbreviations

| | |
|---|---|
| CM | Co-processor module |
| EN-M | Ethernet module |
| PB-M | PROFIBUS module |
| CU | Central module |

# 5 Input and Output Modules

Overview of the input and output modules for the H41q/H51q system

| Type | Channels | Description | SIL 3 |
|---|---|---|---|
| F 3221 | 16 | Digital input module for contacts | |
| F 3224A | 4 | Digital input module for proximity switches (Ex)i | |
| F 3236 | 16 | Digital input module for contacts | • |
| F 3237 | 8 | Digital input module for proximity switches | • |
| F 3238 | 8 | Digital input module for proximity switches (Ex)i | • |
| F 3240 | 16 | Digital input module for contacts | • |
| F 3248 | 16 | Digital input module for contacts | • |
| F 3322 | 16 | Digital output module | |
| F 3325 | 6 | Supply module for transmitter (Ex)i | |
| F 3330 | 8 | Digital output module | • |
| F 3331 | 8 | Digital output module | • |
| F 3333 | 4 | Digital output module | • |
| F 3334 | 4 | Digital output module | • |
| F 3335 | 4 | Digital output module (Ex)i | • |
| F 3349 | 8 | Digital output module | • |
| F 3422 | 8 | Digital relay output module | |
| F 3430 | 4 | Digital relay output module | • |
| F 5220 | 2 | Counter module | • |
| F 6215 | 8 | Analog input module | |
| F 6217 | 8 | Analog input module | • |
| F 6220 | 8 | Analog input module for thermocouples (Ex)i | • |
| F 6221 | 8 | Analog input module (Ex)i | • |
| F 6705 | 2 | Analog output module | • |
| F 6706 | 2 | Analog output module | |

Table 13: Overview of the Input and Output Modules

## 5.1 I/O Modules with Associated Standard Function Blocks

| I/O module | | Standard function block | | | |
|---|---|---|---|---|---|
| Type | TÜV 1) | Type | Function | TÜV 1) | OS 2) |
| F 3221 | - | | | | |
| F 3222 3) | - | | | | |
| F 3223 3) | - | | | | |
| F 3224 A | - | | | | |
| F 3225 3) | - | | | | |
| F 3227 3) | - | | | | |
| F 3228 3) | - | | | | |
| F 3235 3)4) | • | HB-RTE-3 | Monitoring of digital testable input modules | • | • |
| F 3236 4) | • | | | | |
| F 3237 4) | • | HB-RTE-3 | Monitoring of digital testable input modules | • | • |
| F 3238 4) | • | HB-RTE-3 | Monitoring of digital testable input modules | • | • |
| F 3240 4) | • | | | | |
| F 3248 4) | • | | | | |
| F 3311 3) | - | | | | |
| F 3312 3) | - | | | | |
| F 3313 3) | - | H8-STA-3 | Group shutdown | • | • |
| F 3314 3) | - | H8-STA-3 | Group shutdown | • | • |
| F 3321 3) | - | | | | |
| F 3322 | - | | | | |
| F 3323 3) | - | HB-BLD-3/4 | Module and line diagnostics | • | • |
| F 3325 | - | | | | |
| F 3330 4) | • | | | | |
| F 3331 4) | • | HB-BLD-3/4 | Module and line diagnostics | • | • |
| F 3332 3) | - | | | | |
| F 3333 4) | • | | | | |
| F 3334 4) | • | HB-BLD-3/4 | Module and line diagnostics | • | • |
| F 3335 | • | | | | |
| F 3348 3)4) | • | | | | |
| F 3349 | • | HB-BLD-3/4 | Module and line diagnostics | • | • |
| F 3412 3) | - | | | | |
| F 3413 3) | - | | | | |
| F 3422 | - | | | | |
| F 3430 4) | - | | | | |
| F 5202 3) | - | | | | |
| F 5203 3) | - | | | | |
| F 5220 5) | • | HF-CNT-3/4 | Function block for F 5220 (counter) | • | • |
| F 6103 3) | - | HA-LIN-3 | Evaluation of temperature measurement | • | • |
| F 6204 3) | - | HA-PMU-3 | Configurable transmitter | • | • |
| F 6207 3) | - | HA-LIN-3<br>HA-PMU-3 | Evaluation of temperature measurement for configurable transmitters | •<br>• | •<br>• |
| F 6208 3) | - | HA-PMU-3 | Configurable transmitter | • | • |
| F 6213 3)4) | • | HA-PMU-3 | Configurable transmitter | • | • |
| F 6214 3)4) | | HA-RTE-3 | Monitoring of analog testable input modules | • | • |
| F 6215 | - | HA-LIN-3<br>HA-PMU-3 | Evaluation of temperature measurement for configurable transmitters | •<br>• | •<br>• |
| F 6216A 3) | - | HA-LIN-3<br>HA-PMU-3 | Evaluation of temperature measurement for configurable transmitters | •<br>• | •<br>• |

| I/O module | | Standard function block | | | |
|---|---|---|---|---|---|
| Type | TÜV[1)] | Type | Function | TÜV[1)] | OS[2)] |
| F 6217 | • | HA-LIN-3<br>HA-PMU-3 | Evaluation of temperature measurement<br>Configurable transmitters | •<br>• | •<br>• |
| F 6220 [5)] | • | HF-TMP-3 | Function block for F 6220 (thermocouple) | • | • |
| F 6221 [5)] | • | HF-AIX-3 | Function block for F 6221 (analog (Ex)i) | • | • |
| F 6701 [3)] | - | HA-PMU-3 | Configurable transmitter | • | • |
| F 6705 [4)] | • | HZ-FAN-3<br>HA-PMU-3 | Fault indicators for testable I/O modules<br>Configurable transmitter | • | • |
| F 6706 | - | HA-PMU-3 | Configurable transmitter | • | • |

1) TÜV • (TÜV inspection) indicates that the corresponding I/O module or function block can be used for safety-related functions and was certified by the TÜV.

2) OS • indicates that the function block is delivered with the operating system.

3) Discontinued module - no longer deliverable.

4) Diagnostic mode with HZ-DOS-3 is possible.

5) Observe the settings for safety time and watchdog time, see Chapter 5.3.2.

Table 14: I/O Modules with Corresponding Standard Function Blocks

### 5.1.1 Scope of Input and Output Modules

The following table shows for which scope the input and output modules can be used without additional measures.

The scopes are in compliance with the following standards:

- Process technology in accordance with EN 61000-6-2, EN 61131-2 Zone B, EN 61326-1.
- Burners in accordance with EN 298: 2012.
- Alarm: Fire detection and fire alarm systems in accordance with EN 54-2 and alarm system in accordance with EN 50130-4.

| Module | Process technology | EN 61000-6-7 | EN 61326-3-1 | EN 61326-3-2 | NE21 | Burners | Alarm |
|---|---|---|---|---|---|---|---|
| F 3221 | • | | | | | | |
| F 3224A | • | | | | | | |
| F 3236 | • | • | • | • | • | • 1) | |
| F 3237 | • | • | • | • | • | • | |
| F 3238 | • | • | • | • | • | • | |
| F 3240 | • | • | • | | | | |
| F 3248 | • | • | • | | | | |
| F 3322 | • | | | | | | |
| F 3330 | • | • | • | • | • | • 1) | |
| F 3331 | • | • | • | • | • | • 1) | • |
| F 3333 | • | • | • | • | | • 1) | |
| F 3334 | • | • | • | • | • | • 1) | • |
| F 3335 | • | • | • | • | • | • | |
| F 3349 | • | • | • | | | • | |
| F 3422 | • | | | | | | |
| F 3430 | • | • | • | • | • | • | • |
| F 5220 | • | • | • | • | • | • | |
| F 6215 | • | | | | | | |
| F 6217 | • | • | • | • | • | • | • |
| F 6220 | • | • | • | | | | |
| F 6221 | • | • | • | • | • | • | • |
| F 6705 | • | • | • | • | • | • | |
| F 6706 | • | | | | | | |
| 1) Only applying to burners in accordance with EN 298:2012: Separated power and signal lines, even for short distances. | | | | | | | |

Table 15: Scope of HIQuad I/O Modules

## 5.2 Configuring the Inputs and Outputs

Proper function of the F 5220, F 6220 and F 6221 modules can only be ensured if the safety time and the watchdog time are set to ensure that:

Safety Time $\geq$ 3 * Watchdog Time

## 5.3 Noise Blanking

Disturbances are deviations from a module's normal operation. The extensive self-tests of all modules detect disturbances and cause the module or, if necessary, the entire controller to shut down. Transient interference is due to external influences and can also be detected during self-tests.

Noise blanking allows the operating system to tolerate such brief interference. Noise blanking is influenced by the *Number of Noise Blanking Cycles* parameter, the safety time and the watchdog time:

- If the number of noise blanking cycles is set to 1 and the ratio of safety time to watchdog time is 2:1, the integrated noise blanking is activated for a maximum of 1 cycle.
- If the parameters are set to other values, the user can define the number of cycles for which interference in the I/O area is tolerated.

i The system limits the number of noise blanking cycles to the value:

(Safety Time / Watchdog Time) - 2.

This ensures that the time between the moment in which the fault has occurred in the system and the moment in which the system responds to the fault does not exceed the safety time.

(Supposing that Cycle Time = Watchdog Time)

- If the number of noise blanking cycle is set to 0, noise blanking is deactivated.

| Example no. | 1 | 2 | 3 |
|---|---|---|---|
| Cycle time | 100 ms | 200 ms | 200 ms |
| Safety time | 1000 ms | 2000 ms | 1000 ms |
| Watchdog time | 300 ms | 500 ms | 500 ms |
| Number of noise blanking cycles limited to: | 1 | 2 | 0 [1)] |
| [1)] In this case, the integrated noise blanking is active, if the number of noise blanking cycles is set to 1. | | | |

Table 16: Examples of Noise Blanking

Error code 188 indicates that noise blanking is active for the F 7553 module, and error code 197 for I/O modules. The diagnostic panel on the central module can be used to determine on which module noise blanking was triggered.

### 5.3.1 Mode of Operation of the Integrated Noise Blanking

The integrated noise blanking tolerates faults in the cycles that can be followed by at least one other cycle (watchdog time) before the safety time expires.

Refer to the ELOP II online help for further details on how to configure noise blanking.

### 5.3.2 Noise Blanking in the F 5220, F 6220 and F 6221 Modules

In these modules, noise blanking operates irrespective of the *Number of Noise Blanking Cycles* parameter. It must be set in accordance with the selected safety time and the watchdog time. The following condition must be met:

Number of Noise Blanking Cycles = Safety Time / Watchdog Time - 2.

## 5.4 Behavior if an Output Module is Faulty

Depending on safety and/or availability, the controller's behavior in the event of a fault in a testable output channel can be defined by 1 out of 3 different parameter values.

- Normal operation:

  Faulty modules are shut down using the integrated safety shutdown. If a module cannot be shut down, the behavior depends on the system:

  - In the H51q system, the watchdog signal is switched off, thus causing the connection module to safely shut down the H51q system's subrack.
  - The H41q system enters the safe state, i.e., it is completely shut down.
- Display only (not for safety-related applications):

  Faulty modules are shut down using the integrated safety shutdown. If a module cannot be shut down, the connection module shuts down the subrack in a non-safety-related manner.
- Emergency stop:

  Immediate shutdown of the entire PES (emergency stop), if an output channel is defective or an I/O bus fault occurs. If the PES has a redundant I/O bus, only the central module with the faulty I/O bus is shut down.

## 5.5 Group Shutdown

The group shutdown feature allows an entire group of testable output modules to enter the safe state if a fault occurs in a testable output module.

The H8-STA-3 standard function block can be used to arrange up to 10 testable output modules into a group.

## 5.6 Field Terminals

The conditions specified in Table 15 must be observed when connecting the input and output modules to the field.

# 6 Communication

This chapter describes communication of an H41q/H51q controller with other HIMA controllers, the PADT and third-party systems.

| Communication partner connected | Address range[1)] |
|---|---|
| PADT (programming and debugging tool), serial<br>TCP / IP connection over F 8627X / F 8628X | SIO channel 1…8 or<br>IP address |
| LCL printer (logic-plan-controlled logging) | SIO channel 2 |
| Siemens protocol 3964R (Siemens PES is master) | SIO channel 1, 2 |
| Modbus master (HIMA PES is the slave) | SIO channel 1…8 |
| Modbus slave (telephone modem) with HK-MMT-3 function block (HIMA PES is the master) | SIO channel 1, 2 |
| Safety-related PES (HIPRO-S) | SIO channel 1…8 |
| PES (HIBUS with PES master) | SIO channel 3…8 |
| Ethernet via communication module F 8627(X):<br>▪ HIPRO-S/HIPRO-S V2<br>▪ Modbus TCP/IP (HIMA PES is the slave), F 8627X only<br>▪ A&E server (over Modbus TCP/IP)<br>▪ OPC (over Modbus TCP/IP or HIMA OPC Server) | In accordance with IP address |
| PROFIBUS-DP slave via communication module F 8628(X) | Station address 0…127 |
| 1) SIO channels 1 and 2 are in the central module, 3…8 on optional modules of type F 8621A. | |

Table 17: Overview of HIMA PES Communication Connections

Communication can only take place if the HIMA PES is in RUN. The only exception is communication with the PADT.

i Handling the requests of the communication partners requires time on the central module and increases its cycle. When planning the communication connections, ensure that the request frequency is set sufficiently low. Take the number of connections as well as their transfer rate and data volume into account.

This can be done by setting suitable waiting times between the requests in the master systems, for which the HIMA PES acts as a slave.

If a master system (e.g., PADT, DCS over Modbus) sends a request for data (read data), the controller immediately sends a response on the bus that received the read request.

## 6.1 Communication with Other HIMA PES

The operating systems of the controllers are designed to ensure data exchange between HIMA programmable electronic systems via the serial HIBUS bus system. To this end, at least one PES H51q must be equipped with a co-processor module (F 8621A) which acts as PES master. In the slave, the interfaces on the central module as well as those on the co-processor modules can be used.

Safety-related data transmission can also be ensured via Ethernet using the F 8627(X) communication module. In this case, each controller must be equipped with an F 8627(X). A PES master need not exist physically, but data traffic monitoring must be defined and configured in ELOP II.

Data to be sent and received by a PES is configured as variable with the HIPRO-S (for safety-related data transmission) or HIPRO-N (for non-safety-related data transmission) attribute.

Monitoring the safety-related communication for regular reception of data from the master system can be configured in the resource properties. For cases in which the master system does not write data within the defined time, the client system can be configured to set the imported data to FALSE (or 0). Regardless of that, communication errors are always reported via system variables.

### 6.1.1 Safety-Related Data Transmission

The variable values to be transferred via HIPRO-N are declared in the program instance, variable declaration, **HIPRO-N/S** tab.

During operation, the PES master reads all export data from the connected PES, assembles the transmissions for the import data of the connected PES and sends them.

### 6.1.2 Safety-Related Data Transmission via HIPRO-S/HIPRO-S V2

The variable values to be transferred are declared in the program instance, variable declaration, **HIPRO-N/-S** tab.

During operation, the PES master for HIPRO-S (serial only) organizes direct data transmission between the individual PES; the PES master itself does not save the data. Although data transmission runs via HIBUS, it should be considered as a point-to-point connection.

i **For safety-related communication, HIMA recommends using the HIPRO-S V2 protocol via Ethernet!**

#### 6.1.2.1 Determining the Monitoring Time

The transmission of safety-related data via HIPRO-S for serial interfaces is monitored to ensure that a given time limit is not exceeded. This time limit is entered as *Monitoring Time* in the *Edit Resource* dialog box of the corresponding target resource.

The monitoring time depends on the data rate.

The monitoring time $T_{Monitoring}$ is determined with the following calculation procedure:

1. Determining the token cycle time.
2. Determining the bus cycle time.
3. Determining the monitoring time for data rate.

### Determining the Token Cycle Time

The token cycle time, $T_{Token\ Cycle\ Time}$, i.e., the time required by the token to complete a cycle, is calculated as follows:

$$T_{Token\ Cycle} = n_{Masters} * T_{Master\ basis} + T_{All\ masters}$$

| | |
|---|---|
| $n_{Master}$ | Number of masters. |
| $T_{Master\ basis}$ | Master basis time for connection test, time broadcast and token passing, approx. 40 ms. |
| $T_{All\ masters}$ | The sum of the time values required by the masters for their task-specific actions. For each master, the times required in accordance with its function, must be added in accordance with the table: |

| Master | Action | Time required [ms] |
|---|---|---|
| PADT | Control Panel opened | 35 |
| | Online test | 135 [1)] |
| HIPRO-S | For each safety-related transmission | 50 |
| | For each byte data | 0.2 |
| 1) This applies to approx. 60 displayed variables; for each additional variable, the required time must be increased by approx. 1.5 ms! | | |

Table 18: Times Required for Safety-Related Communication

The time values, which are needed for each master and determined in accordance with the values specified in the table, must be added to $T_{All\ masters}$.

### Determining the Bus Cycle Time

The bus cycle time, $T_{\text{Bus cyle time}}$, is the time needed for transmitting the data over the bus. It results from the token time and is calculated with the following formula:

$T_{\text{Bus cycle}} = T_{\text{Token cycle}}$

or, if the only PES master is redundant:

$T_{\text{Bus cycle}} = 0.5 * T_{\text{Token cycle}}$

The calculated bus cycle time applies to a data rate of 57 600 baud.

For a data rate of 9600 baud, the bus cycle time can be assumed to be 5 times the bus cycle time applying at a rate of 57 600 baud. The following condition applies:

$T_{\text{Bus cycle\_9600 baud}} = 5 * T_{\text{Bus cycle\_57600 baud}}$

### Determining the Monitoring Time

The monitoring time $T_{\text{Monitoring}}$ is composed of the calculated bus cycle time plus a disturbance time. Fault detection, resends or bus switching may occur during the disturbance time. This disturbance time is assumed to be equal to four bus cycle times, such that the following equation applies:

$T_{\text{Monitoring}} = 5 * T_{\text{Bus cycle}}$

## 6.1.3 Safety-Related Communication via the F 8627(X) Module

The F 8627(X) communication module allows up to 64 HIMA PES of the H51q system family to safely communicate with one another if an operating system version as of 3.0 is used, or up to 31 HIMA PES if a previous operating system version is in use. This safety-related communication takes place via Ethernet and complies with IEEE 802.3 requirements. The requisite is that each controller operating on the bus is equipped with the communication module. The central module maps the safety-related data to be transferred from the communication module to the Ethernet network. The bus type is HIBUS. The PES master is defined in ELOP II as a dummy. The variables are defined as HIPRO-S variables in the variable declaration properties. It is also possible to start a compiler run for the (dummy) PES master to obtain a cross-reference list including all the communication variables.

Refer to the F 8627(X) data sheet for further configuration details.

**⚠WARNING**

**Physical injury due to corrupted safety-related data possible!**

**The following mixed operation is not allowed for the same HIPRO-S/HIPRO-S V2 connection (i.e., to the same communication partner):**

- **Serial via co-processor module F 8621A and**
- **In parallel via the Ethernet communication module F 8627(X)**

Mixed operation already exists if a serial connection is operating and an Ethernet module is additionally added to the PES.

Connection can be lost at any time.

Workarounds

- If an Ethernet module is required for other tasks, use the HK-COM-3 function block with channel-specific settings and its *Function* input to avoid HIPRO-S communication.
- If an Ethernet module is required for HIPRO-S V2 connections to other PES, use the HK-COM-3 function block with connection-specific settings and its *E5 – Connection Bitmask* input to avoid the special connection to the serial partner.

For details on HIPRO-S V2, refer to the HIPRO-S V2 manual (HI 800 723 E).

## 6.2 Communication with the Programming and Debugging Tool (PADT)

The programming and debugging tool (PADT) is used to program, load, monitor and document the HIMA controller's function. The PADT is a PC that is running the ELOP II programming tool. Two communication options are possible:

- Communication via HIBUS: The PADT acts as a master on the bus.
- Communication via Ethernet: A point-to-point connection is established between the PADT and the HIMA controller.

Refer to the F 8627X and F 8628X data sheets for further configuration details.

## 6.3 Communication with Third-Party Systems

The operating system of the H41q/H51q is designed to allow communication with third-party systems such as visualization or process control systems, or other automation systems.

The variables to be transferred are defined in the variable declaration as BUSCOM or 3964R variables.

A third-party system can read all the PES variables configured for export. Data sent from the third-party system must be defined as import variables.

Communication with third-party systems can only take place if the HIMA PES is in RUN.

### 6.3.1 Serial Connections

The H41q/H51q system can use the following protocols to ensure serial connections to third-party systems:

- Modbus protocol: the H41q/H51q system operates as slave or master system.
- 3964R Siemens protocol: the H41q/H51q system operates as slave system.

Interfaces 1 and 2 on the central module support the 3964R Siemens protocol and the HIMA controller operating as Modbus master. Only one protocol can be configured for each interface. The interface parameters can be defined in the Central Module Properties of the resource, if the parameters are different from the default settings (57 600 baud, 1 stop bit, parity bit even).

Serial Modbus connections to a Modbus master are also possible over the F 8621A module interface.

i

Interferences may occur if a baud rate of 19.2 kbit/s is used for serial Modbus connections via the F 8621A co-processor module.

For serial Modbus connections, HIMA recommends using baud rates of 9.6 kbit/s or 57.6 kbit/s for the F 8621A co-processor and not a baud rate of 19.2 kbit/s.

### 6.3.2 Connections via Communication Modules

For connection to third-party systems over a TCP/IP connection (Ethernet) using an F 8627(X) communication module, the following options are available:

- Communication with an HIMA OPC Server.
- Modbus TCP/IP protocol: the HIMA PES can only operate as a slave. Operation as a master is not possible.

The F 8628X communication module can be used to implement a slave connection via PROFIBUS DP (field bus). Refer to the F 8628X data sheet for further details.

## 6.4 Communication via Modbus Protocol

The Modbus protocol is designed as master-slave system and is usually used to connect HIMA controllers to a process control system, e.g., via a RS485 bus.

The following connection types can be used for the H41q and H51q controllers:

- Serial and TPC/IP connection, as slave systems, no standard function block is required.
- Serial connection, as master systems only, the HIMA HK-MMT-3 standard function block must be used.

  Refer to the ELOP II online help for a description of the function block's functionality.

The Modbus protocol was defined by Gould Modicon. HIMA recommends requesting the Modbus and Modbus TCP/IP documentation directly from the Modbus organization (www.modbus.org) and being informed about peculiarities of the third-party system acting as Modbus master.

i Only the function codes specified in the following chapters are available for H41q/H51q.

For better understanding, the following section describes the essential properties.

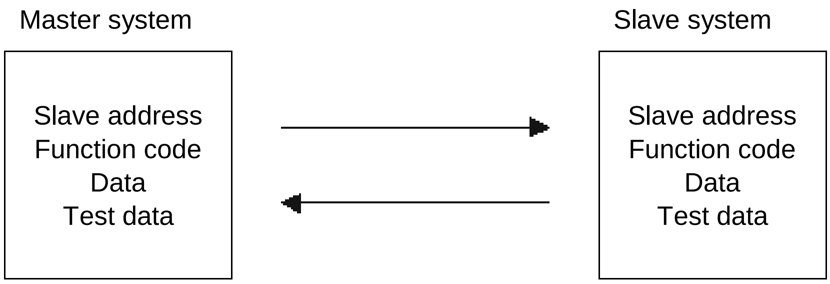



Figure 1: Principle for Data Traffic with Modbus Protocol

### 6.4.1 Transmission over Serial Connection (RS485)

The RTU (remote terminal unit) is the usual mode of transmission between computer systems and the only one implemented in HIMA PES. Data is transmitted asynchronously with 8 bits and test data.

In general, the RTU mode of transmission has the following data format:

| Start | Slave | Code | Data | Test data | Transmission end |
|---|---|---|---|---|---|
| T1 T2 T3 | 1 byte | 1 byte | 1) | 2 bytes | T1 T2 T3 |
| 1) The number of bytes depends on the function and number of addresses and data | | | | | |

Explanation of the fields:

| | |
|---|---|
| Start<br>Transmission end | Transmission start and end are marked by 3½ characters (bytes) pause (T1 T2 T3). |
| Slave | Address of the slave system.<br>(For HIMA: bus station number, setting on the central module). |
| Code | Function code: Writing or reading of variables or events. |
| Data | Data includes start address, number of addresses and data depending on the function. Refer to the specification in the Modbus protocol. |
| Test data | CRC code (cyclic redundancy check) automatically created by the sending system. |

### 6.4.2 Transmission over TCP/IP Connection

Similarly to a serial connection, the transmission data format includes the *Slave*, *Code* data fields and data. This data fields are packed into a TCP telegram.

With a TCP/IP connection, the HIMA PES may only act as a slave.

Peculiarities of the transmission over a TCP/IP connection:

- No broadcast transmission is possible. Transmissions that must be sent to multiple slaves, must be sent individually to each slave.
- If a controller cannot respond to a slave request, e.g., the central module is faulty the communication module sends error code 11 (dec) or 0B (hex) back to the master, which means *Gateway Target Device Failed to Respond*.
- Each TCP/IP connection increases the central module cycle. The master must pause between response and next request for at least:

  $T_{Pause}$ = Number of masters * 10 ms

Refer to the module manual (HI 800 265 E) for a description of the used network ports and the address mapping of the BUSCOM variables to the memory areas of the F 8627X module.

### 6.4.3 Functions of the Modbus Protocol

The Modbus protocol is used to implement 4 functions:

- Reading of variables.
- Writing to variables.
- Reading of events.
- Time synchronization.

The master system can read and write to the HIMA PES variables that are configured for BUSCOM import and export.

The value change of any Boolean variables can be defined in ELOP II as event. The status of the Boolean signal in the current cycle is compared to the signal in the previous cycle. If changes occur, the event number, the current state and the PES time at the cycle's beginning are stored into a buffer memory. Therefore, events recorded in the same cycle have the same timestamp.

Events are read from the buffer memory using user-defined or standard function codes (see Chapter 6.4.7 and Chapter 6.4.8).

---

i In the following section, the numbers have the addition hex if they are expressed as hexadecimal number.

Decimals have no addition or the addition dec.

---

### 6.4.4 Implemented Read Codes 1, 3 (1, 3 hex)

Function code 1 READ COIL STATUS is implemented for Boolean variables, and function code 3 READ HOLDING REGISTER for integer variables.

Refer to the resource-specific documentation *RES Docu (generated)* in ELOP II for further information on the corresponding Modbus addresses.

### 6.4.4.1 Example of Code 1: Read Boolean Variables

Slave number: 17 (dec = 11 hex)
Function code: 1 (01 hex)
Boolean variables: 20…56 = 37 variables (00 25 hex)
Refer to *RES Docu (generated)* in ELOP II for the start address.
Start address: 20 (14 hex)

Master system request:

| | Slave | Function | Start address | | Number | | CRC 2 bytes | |
|---|---|---|---|---|---|---|---|---|
| hex | 11 | 01 | 00 | 14 | 00 | 25 | xx | xx |

Slave response:

| | Slave | Function | Bytes | Data 27-20 | Data 35-28 | Data 43-36 | Data 51-44 | Data 56-52 | CRC 2 bytes | |
|---|---|---|---|---|---|---|---|---|---|---|
| hex | 11 | 01 | 05 | CD* | 6B* | B2* | 0E* | 1B* | xx | xx |
| * = possible values | | | | | | | | | | |

The content of the first data byte is CD (hex), which corresponds to the 11001101 bit pattern. This means that the variables numbered 27, 26, 23, 22 and 20 have the value 1, and the variables numbered 25, 24 and 21 have the value 0.

The controller sends the data to the master immediately after its request.

Refer to Chapter 6.4.8 for an example of how the event buffer memory is read.

i Modbus function code 1 provides duplicate responses if all the following conditions are met:
- Modbus is redundantly connected via the interfaces of the F 8621A module.
- The response length is identical to the request length.

Workaround: Set the number of values read with a request outside the range 17…24 to ensure that the response and request lengths differ.

### 6.4.4.2 Error Telegram and (Read) Error Codes

If the slave cannot process the master's request, e.g., because the requested address is invalid, the slave responds with an error telegram.

The error telegram repeats the function code with the most significant bit set, i.e., 01 (hex) becomes 81 (hex), 03 becomes 83.

The error telegram is structured as follows:

| | Slave | Function | Error | CRC 2 bytes | |
|---|---|---|---|---|---|
| hex | 11 | 81 | 02 | xx | xx |

| Error code | Description |
|---|---|
| 02 | Excessively high address value, variable not available.<br>Data >256 bytes (2048 Boolean values, 128 integer values). |

Table 19: Error Codes with Read Codes 1, 3

### 6.4.5 Implemented Write Codes 5, 15, 6, 16 (5, F, 6, 10 hex)

Implemented function codes

| | | |
|---|---|---|
| 5 | FORCE SINGLE COIL | Writing to individual Boolean variables. |
| 15 | FORCE MULTIPLE COILS | Writing to several Boolean variables. |
| 6 | PRESET SINGLE REGISTER | Writing to individual integer variables. |
| 16 | PRESET MULTIPLE REGISTERS | Writing to several integer variables. |

Refer to the documentation of the resource in the ELOP II structure tree, *RES Docu (generated)*, for further information on the corresponding Modbus addresses.

i HIMA recommends using Modbus codes 15 and 16 instead of code 5 and 6 for central module interfaces. If codes 15 and 16 are not possible, the BV 7046 Y cable must not be used and must be replaced with a cable for 1-channel operation, e.g., with BV 7040.

#### 6.4.5.1 Example of Code 5: Writing to a Boolean Variable

Slave number: 17 (dec = 11 hex)
Function code: 5 (writing to individual variables, 05 hex)
Refer to *RES Docu (generated)* for the value of the address.
Address: 37 (25 hex)

Master transmission:

| | Slave | Function | Start address | | Data | | CRC 2 bytes | |
|---|---|---|---|---|---|---|---|---|
| hex | 11 | 05 | 00 | 25 | FF | 00 | xx | xx |

Slave response (i.e., transmission repeated):

| | Slave | Function | Start address | | Data | | CRC 2 bytes | |
|---|---|---|---|---|---|---|---|---|
| hex | 11 | 05 | 00 | 25 | FF | 00 | xx | xx |

When the next cycle starts, the controller adopts the transmitted data into the variable. The longest response time is thus approximately the PES cycle time.

#### 6.4.5.2 Example of Code 15: Writing to Several Boolean Variables

Slave number: 17 (dec = 11 hex)
Function code: 15 (writing to several variables, 0F hex)
Number of Boolean variables: 10 (0A hex)
Values of the Boolean variables 1…16: 4D 03 (hex = 0100 1101 0000 0011 binary)
Refer to *RES Docu (generated)* for the value of the address.
Address: 37 (25 hex)

Master transmission:

| | Slave | Function | Start address | | Number of values | | Number of bytes | Value 1-8 | Value 9-16 | CRC 2 bytes | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| hex | 11 | 0F | 00 | 25 | 00 | 0A | 02 | 4D | 03 | xx | xx |

Slave response (i.e., transmission repeated):

| | Slave | Function | Start address | | Number of values | | CRC 2 bytes | |
|---|---|---|---|---|---|---|---|---|
| hex | 11 | 0F | 00 | 25 | 00 | 0A | xx | xx |

When the next cycle starts, the controller adopts the transmitted data into the variable. The longest response time is thus approximately the PES cycle time.

### 6.4.5.3 Write Error Codes

Chapter 6.4.4.2 describes the structure of the error telegram.

The error telegram repeats the function code with the most significant bit set, i.e., 0F (hex) becomes 8F (hex), 10 becomes 90.

| Error code | Description |
|---|---|
| 02 | Excessively high address value, variable not available.<br>Data >256 bytes (2048 Boolean values, 128 integer values). |
| 03 | Coded Boolean value not identical to FF00 hex or 0000 hex (Boolean values) - this only applies to code 5! |

Table 20: Error Codes with Write Codes 5, 15, 6, 16

i Applying to serial communication only: The write codes can send broadcast messages to all slaves, if 0 is used as slave address.

## 6.4.6 Loop Back Diagnostic Test, Function Code 8 (8 hex)

Diagnostic code 0 of function code 8 is used to request the slave system to repeat the master's request.

| Diagnostic code | Description |
|---|---|
| 0 | RETURN QUERY DATA |

Table 21: Diagnostic Code 0

It applies to all HIMA slaves.

The HIMA master knows all 21 diagnostic codes.

i HIMA recommends using Modbus codes 15 and 16 instead of code 8 for central module interfaces. If codes 15 and 16 are not possible, the BV 7046 Y cable must not be used and must be replaced with a cable for 1-channel operation, e.g., with BV 7040.

## 6.4.7 Function Codes for Events 65, 66, 67 (41, 42, 43 hex)

Any signal change of Boolean variable can be defined in ELOP II as event. The status of the Boolean signal in the current cycle is compared to the signal in the previous cycle. If changes occur, the event number, the current controller's state and time are stored into a buffer memory. Therefore, events recorded in the same cycle have the same timestamp.

In operating system with hardware revision status as of V7.0-8 (0214), the buffer contains 500 events of 8 bytes each. In previous revision statuses, the buffer contains 250 events.

A maximum of 8 events (= 64 bytes) are sent at once.

The buffer overflow is identified by FFFF (hex). If this overflow marker is also sent, the maximum length increases up to 66 bytes. The buffer is blocked for new events, until the overflow marker has been read out. Afterwards, the buffer can record new events.

The user-specific Modbus codes 65, 66, 67 are used to transmit events from the slave system to a master system:

| Code | Description | Function |
|---|---|---|
| 65 | Read event values<br>(event status) | Provides the state of all events without time specification. |
| 66 | Read new events<br>(address, status, time) | Provides events with time specification from the event buffer. |
| 67 | Events sent last | Request to repeat the last transmission. |

Table 22: Modbus Function Codes for Events

i These event function codes are HIMA-specific!

### 6.4.7.1 Function Code 65: Reading the Event Values (Event Status)

Request telegram:

| | Slave address | Function | Number of bytes | Start address | | Number of values | | CRC 2 bytes | |
|---|---|---|---|---|---|---|---|---|---|
| hex | 01 | 41 | 04 | 00 | 00 | 00 | 10 | F4 | DD |

The master uses this telegram to request slave 1 to send the states of all defined events. The timestamp is not transmitted.

This request is sent, for instance, when operation is started to check the events' functional state. The start address is always 0000 (hex), in the following example, 10 (hex) = 16 (dec) events are requested.

i The number of values to specify in the request telegram is always the number of defined events!

Response telegram:

| | Slave address | Function | Number of bytes | Event 1 to 8 | Event 9 to 16 | CRC 2 bytes | |
|---|---|---|---|---|---|---|---|
| hex | 01 | 41 | 02 | 00 | 00 | BC | 7C |

The slave responds by repeating the slave address and function, then specifying the number of bytes attached and including the required data.

The values of the requested event states are transmitted in a packed format. For further details, refer to the explanation concerning the response telegram provided in Chapter 6.4.4.

Error telegram:

If the slave cannot send the data (e.g., due to an invalid start address), the function with the most significant bit set (C1 hex instead of 41 hex) is repeated in the response telegram.

Example:

| | Slave address | Function | Error code | CRC 2 bytes | |
|---|---|---|---|---|---|
| hex | 01 | C1 | 02 | B1 | 30 |

### 6.4.7.2 Function Code 66 (42 hex): Reading New Events

Request telegram:

| | Slave address | Function | CRC 2 bytes | |
|---|---|---|---|---|
| hex | 01 | 42 | 80 | 11 |

The master uses this telegram to request slave 1 to send events with timestamp from the event buffer.

Response telegram (no event in the event buffer):

| | Slave address | Function | Number of bytes | CRC 2 bytes | |
|---|---|---|---|---|---|
| hex | 01 | 42 | 00 | 10 | A0 |

The slave responds by repeating the slave address and function. In doing so, it sets the number of bytes to 00, if no new events were recorded in the event buffer.

Response telegram (if new events occurred):

| | Slave address | Function | Number of bytes | Event number | | Event value | Timestamp | | | | | CRC 2 bytes | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | MSB | LSB | | ms | ds | s | min | h | | |
| hex | 01 | 42 | 08 | 0C | 00 | 01 | 14 | 09 | 3B | 15 | 0E | 0A | 44 |

The slave responds by repeating the slave address and function. Then specifying the number of bytes attached and including the required data. Every transmitted event with timestamp is composed of 8 bytes, consisting of relative address of the event, event value (00 or 01) and time stamp. A maximum of 8 events (= 64 bytes) can be sent at once.

The time in the previous example is 2 p.m. (0E hex), 21 minutes (15 hex) 59 seconds (3B hex) and 920 milliseconds (09 hex and 14 hex).

Event number:

| | |
|---|---|
| MSB | Most significant byte |
| LSB | Least significant byte |
| Event value: | 0 or 1 (1 byte) |

Timestamp:

| | |
|---|---|
| ms | 0…99 milliseconds |
| ds | 0…9 deciseconds |
| s | 0…59 seconds |
| m | 0…59 minutes |
| H | 0…23 hours |

The number of bytes in the response telegram indicates how full the buffer is:

| Number of bytes | Description |
|---|---|
| 0 | No new event occurred. |
| < 64 | The response includes all the current events. |
| ≥ 64 | The buffer could include further events. |

Table 23: Filling Level of the Event Buffer

Error telegram:

If the slave cannot provide the data, it repeats the function with the most significant bit set (C2 hex instead of 42 hex) in the response telegram. The example is similar to that described in Chapter 6.4.7.1.

### 6.4.7.3 Function Code 67 (43 hex): Events Sent Last

Request telegram:

| | Slave address | Function | CRC 2 bytes | |
|---|---|---|---|---|
| hex | 01 | 43 | 41 | D1 |

The master uses this telegram to request slave 1 to resend the events transmitted last (e.g., connection disturbed).

Response telegram:

See Chapter 6.4.7.2.

Error telegram:

For example, if code 67 is not allowed because code 66 was not used for the last request:

| | Slave address | Function | Error code | CRC 2 bytes | |
|---|---|---|---|---|---|
| hex | 01 | C3 | 01 | B1 | 30 |

i Code 67 is only possible after code 66, if the master system has not received the correct response to code 66. It causes the slave system to repeat its last response.

Code 65 should be sent after a restart or buffer overflow of the slave system. During normal operation, code 66 or code 67 must be cyclically sent by the master.

Error messages after event scanning

The error telegram repeats the function code with the most significant bit set, i.e., 43 (hex) becomes C3 (hex).

| Error code | Description |
|---|---|
| 1 | Code 67 was not preceded by code 66. |

Table 24: Error Code after Event Scanning

### 6.4.8 Event Scanning with Standard Codes 1, 3 (1, 3 hex)

The requests implemented with the special codes 65, 66, and 67 can also be submitted with standard codes 1 and 3. The following functions are possible:

- Querying event states using code 1.
- Reading events (number, status, time) using code 3.

If 2 master systems use different start addresses, code 3 allows them to simultaneously read events out of the same event buffer. The first master system uses start addresses 3072 and 3073, the second the start addresses 3584 and 3585.

The event variables must be defined in ELOP II (variable declaration, *Event-Driven* attribute). A maximum of 2048 events can be defined.

### 6.4.8.1 Status Query with Code 1

Starting with start address 2048, the status of the variables defined as events can be accessed via READ COIL STATUS.

Master request:

| | Slave | Function | Start address | | Number of events | | CRC 2 bytes | |
|---|---|---|---|---|---|---|---|---|
| | | | MSB | LSB | MSB | LSB | | |
| dec | xx | 1 | 2048…4095 | | Max. 2048 | | xx | xx |

The slave response is defined such as specified for code 1.

### 6.4.8.2 Event Query and Resend

Starting with operating system V7.0-8 (0214), the buffer memory contains a maximum of 500 events. If more events occur, the buffer overflow is recorded and identified by 8 bytes FF (hex). New events are only accepted in the buffer memory if the overflow marker has been read.

Each event is stored into the buffer memory with 8 bytes having the following meaning:

| Event number | | Value | Timestamp | | | | |
|---|---|---|---|---|---|---|---|
| MSB | LSB | | ms | ds | s | m | H |

Refer to *RES Docu (generated)* in ELOP II for the event number.

Event number:

| | |
|---|---|
| MSB | Most significant byte |
| LSB | Least significant byte |
| Value: | 00 or 01 (hex) |

Timestamp:

| | |
|---|---|
| ms | 0…99 milliseconds |
| ds | 0…9 deciseconds |
| s | 0…59 seconds |
| m | 0…59 minutes |
| H | 0…23 hours |

A buffer overflow is identified by the fact that all 8 bytes have the value FF(Hex).

If the slave's response includes all occurred events, i.e., the buffer does not contain any additional event, all bytes of the remaining send data have the value EE (hex).

The same applies if the buffer memory is empty.

If events are requested with code 3, the number of events read out from the event buffer (a maximum of 31 events * 4 integer values = max. 31 * 8 bytes) is identical to the number of events defined in the master system's request. Since 1 event consists of 8 bytes, 4 integer variables must always be read together.

To distinguish a request resent from a new request, the requests must be sent during normal operation with at least 2 alternating start addresses.

If the slave receives a request with the same start address as in the previous request, it assumes that the last master response was not received properly and the master is thus requesting the same events once again. In such a case, the slave resends the same events as in the previous request.

HIMA recommends reading the status of all events using code 1 when communication is started or after an event buffer's overflow.

Start addresses:

| | |
|---|---|
| Master 1: | 3072 and 3073 |
| Master 2: | 3584 and 3585 |

## Example

The master system requests the maximum number of events:

1. Request/transmission 1: start address: 3072

Number of integer variables: 124

2. Request/transmission 2: start address: 3073

Number of integer variables: 124

3. Request/transmission 3: start address: 3072

Number of integer variables: 124

The master system requests one event at a time:

1. Request/transmission 1: start address: 3072

Number of integer variables: 4

2. Request/transmission 2: start address: 3073

Number of integer variables: 4

3. Request/transmission 4: start address: 3072

Number of integer variables: 4

Error messages after event scanning

Chapter 6.4.4.2 describes the structure of the error telegram.

The error telegram repeats the function code with the most significant bit set, i.e., 03 (hex) becomes 83 (hex).

| Error code | Description |
|---|---|
| 02 | Start address or number of values does not correspond to the definition. |

Table 25: Error Code in Case of Invalid Start Address or Number of Values

### 6.4.9 Time Synchronization, Code 70 (46 hex)

A master can synchronize the time and date of the controller via Modbus. To do so, code 70 is used.

Request telegram:

The following telegram synchronizes all slaves on the bus:

| | Slave address | Function | Number of bytes | Time | | | | | Date | | | CRC 2 bytes | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | ms | ds | s | min | h | Day | Month | Year | | |
| hex | 00 | 46 | 08 | 28 | 00 | 00 | 16 | 0E | 09 | 02 | 5D | DB | I1 |

| | | |
|---|---|---|
| Time: | ms | 0…99 milliseconds |
| | ds | 0…9 deciseconds |
| | s | 0…59 seconds |
| | min | 0…59 minutes |
| | h | 0…23 hours |
| Date: | Day | 1…31 days |
| | Month | 1…12 months |
| | Year | 0…99 years |

The slave address is 00 since all slaves should be addressed (broadcast).

It is easy to recognize that the following date and time are used in the example to synchronize: 09.02.93 14:22:00.040 (MM.DD.YY hh:mm:ss.ms)

If only the time should be transmitted, d must be set to 0;
if only the data should be transmitted, ms must be set to 255.

The time is synchronized to the moment at which the first character of this transmission was received. The processing within the slave does not cause any delay.

Response telegram

- Serial connection: Since the transmission is broadcast, the master must only send a request with slave address 0, and no slave response follows.
- TCP/IP connection: The master must send a request to each individual slave and receives a response from each slave. It is therefore not possible to synchronize all slaves to the same time.

### 6.4.10 Time Synchronization, Code 6

The time in the PES can also be set with code 6. To do so, the telegram with code 6 must have the value 2048 as starting address and the number of milliseconds passed since the last full minute as data, i.e., the values are in the range 0…59999

- Serial connection: Since the transmission is broadcast, the master must only send a request with slave address 0, and no slave response follows.
- TCP connection: The master must send a request to each slave and receives a response from each slave. It is therefore not possible to synchronize all slaves to the same time.

The time is synchronized to the moment at which the first character of this transmission was received. The processing within the slave does not cause any delay.

With code 6, it is not possible to set the date and time to an absolute value.

### 6.4.11 Notes for Operating the System

The following paragraphs reference to some particular features when the system is connected to process control systems. HIMA recommends users to get more details on the Modbus connection of the process control system.

Communication only takes place if the PES is in RUN.

The values sent by the master system are processed in the user program at the beginning of the next cycle and are therefore treated as physical inputs.

The data requested by the master system is immediately sent to the master (during the cycle).

For a connection via RS485, the following default values are defined in the operating system:

| | |
|---|---|
| Mode of transmission: | RTU |
| Parity bit: | 1 (even) |
| Baud rate: | 57 600 Bd, 9600 Bd where required (DIP switch on the central module) |
| Number of stop bits: | 1 |

If necessary, baud rate, parity and stop bits can be changed in the resource's settings.

The slave number must be defined by setting the bus station number (coding switch on the central module).

In some process control systems, the address counting starts with 1; in the HIMA controller, it starts with 0 (in accordance with the definition in the Modbus manual). This point must be taken into account during engineering.

Example: Process control system address 1…100, corresponding H41q/H51q address 0…99.

### 6.4.12 Wiring the Modbus Cables for Serial Connections

In the following figures, Modbus 1 and Modbus 2 are redundant to one another.

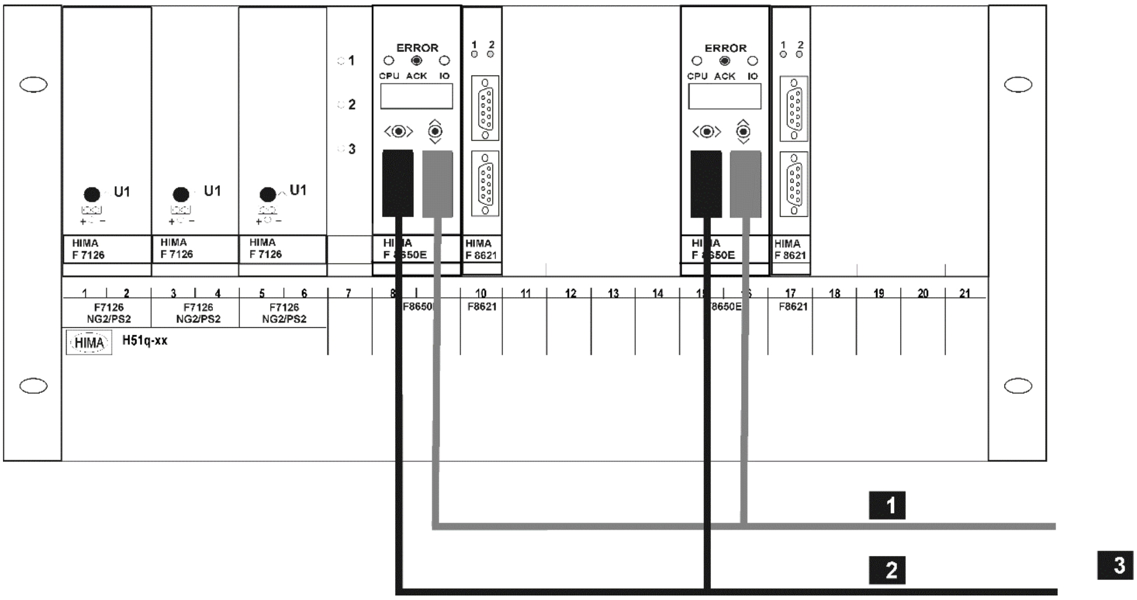
i If the H41q/H51q is operating as Modbus master, use the serial interfaces of the F 865x central module. Do not use the serial interfaces of the co-processor module.

#### 6.4.12.1 Standard Wiring Variants with BV 7046 Cable

To implement a redundancy monitoring for the connection using the BV 7046 cable (Y cable), the receive counter of the interface in question must be evaluated in the user program (see Chapter 6.4.12.3).

The redundant plugs of the BV 7046 cable (Y cable) must always be plugged in to the same interface sockets of the redundant modules.

### Variant 1: Redundant Connection via Central Modules (Redundant Bus)



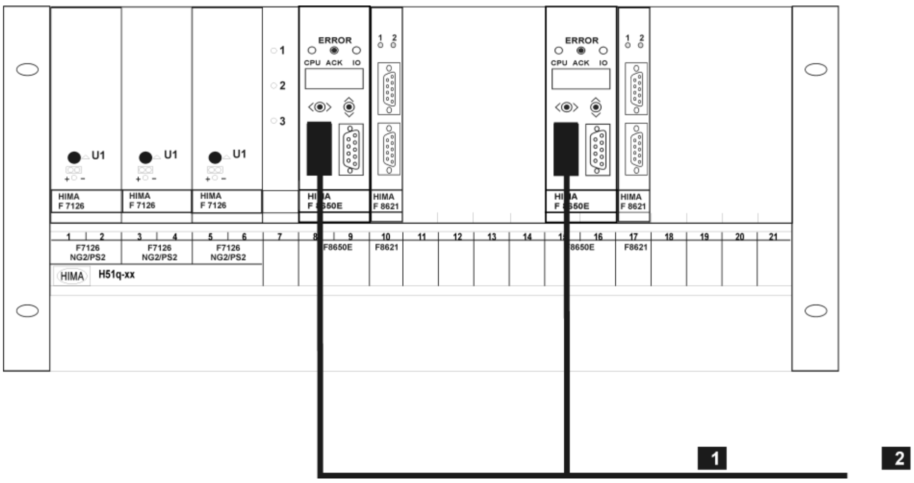

**1** Modbus 2
**2** Modbus 1
**3** To the third-party system

Figure 2: Variant 1, Redundant Connection via Central Modules

- Plug in the first BV 7046 cable (Y cable) for Modbus 1 to the left interface 1 (SIO1) of the redundant F 865x central modules.
- Plug in the second BV 7046 cable (Y cable) for Modbus 2 to the right interface 2 (SIO2) of the redundant F 865x central modules.

i This variant does not permit the use of logic-plan-controlled logging and/or of the 3964R Siemens protocol, since these functions need the interfaces of the central modules.

### Variant 2: Redundant Connection via Central Modules (Mono Bus)




**1** Modbus 1
**2** To the third-party system

Figure 3: Variant 2, Redundant Connection via Central Modules

Figure 3 shows the BV 7046 cable (Y cable) for Modbus 1 plugged in to the left interface 1 (SIO1) of the redundant F 865x central modules.

The BV 7046 cable (Y cable) for Modbus 2 can also be plugged in to the right interface 2 (SIO2) of the redundant F 865x central modules.

## 6.4.12.2 Standard Wiring Variants with BV 7040 Cable

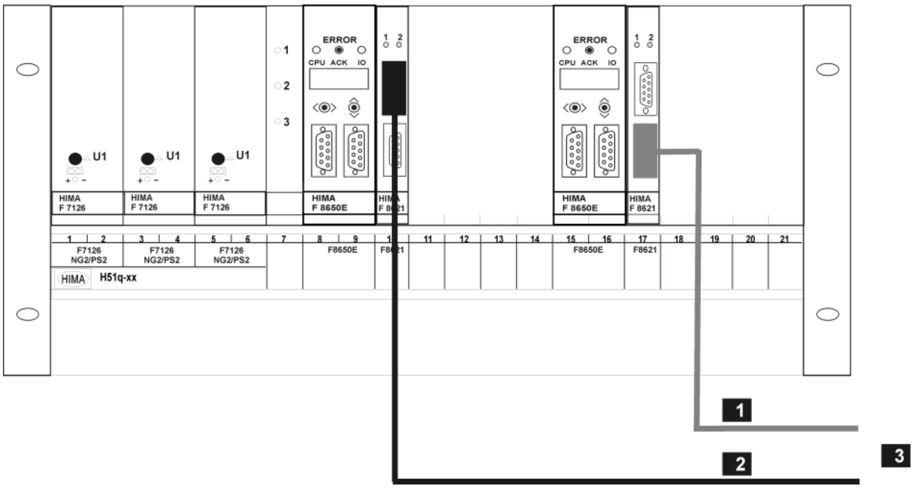
i The two BV 7040 cables for Modbus 1 and Modbus 2 must not be plugged in to the same interfaces of the redundant modules.

The following combinations, for instance, are not permitted:

CU1/SIO1 with CU2/SIO1, CU1/SIO2 with CU2/SIO2,

CM1/SIO1 with CM2/SIO1, CM1/SIO2 with CM2/SIO2 etc.

### Variant 3: Mono Connection via Co-Processor Modules (Redundant Bus)




**1** Modbus 2
**2** Modbus 1
**3** To the third-party system

Figure 4: Variant 3, Mono Connection via Co-Processor Modules (Redundant Bus)

Figure 4 shows:

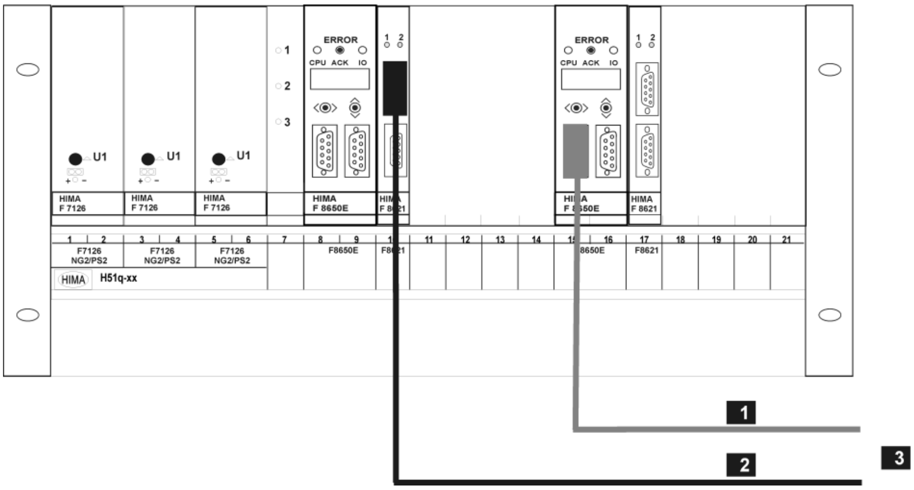
- The first BV 7040 cable for Modbus 1 plugged in to the upper interface 1 (SIO1) of the left co-processor module.
- The second BV 7040 cable for Modbus 2 plugged in to the lower interface 2 (SIO2) of the right co-processor module.

It is also possible:

- To plug in the first BV 7040 cable for Modbus 1 to the lower interface 2 (SIO2) of the left co-processor module.
- To plug in the second BV 7040 cable for Modbus 2 to the upper interface 1 (SIO1) of the right co-processor module.

i Since mono connection does not permit simultaneous use of the upper and lower interfaces, the connecting cable may only be connected in one of these ways.

### Variant 4: Mono Connection via Central Module and Co-Processor Module (Redundant Bus)




**1** Modbus 2

**2** Modbus 1

**3** To the third-party system

Figure 5: Variant 4, Mono Connection via Central Module and Co-Processor Module (Redundant Bus)

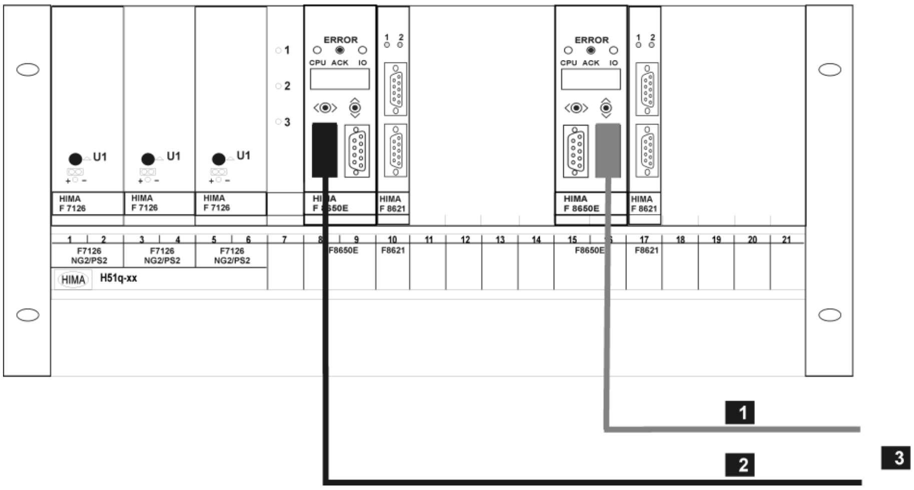
Figure 5 shows:

- The first BV 7040 cable for Modbus 1 plugged in to the upper or lower interface (SIO1 or SIO2) of the left F 8621A co-processor module.
- The second BV 7040 cable for Modbus 2 plugged in to the right or left interface (SIO1 or SIO2) of the right F 865x central module.

It is also possible:

- To plug in the first BV 7040 cable for Modbus 1 to the right or left interface (SIO1 or SIO2) of the left F 865x central module.
- To plug in the second BV 7040 cable for Modbus 2 to the upper or lower interface (SIO1 or SIO2) of the right F 8621A co-processor module.

Variant 5: Mono Connection via Central Modules (Redundant Bus)



1 Modbus 2
2 Modbus 1
3 To the third-party system

Figure 6: Variant 5, Mono Connection via Central Modules

Figure 6 shows:

- The first BV 7040 cable for Modbus 1 plugged in to the upper interface (SIO1) of the left F 865x central module.
- The second BV 7040 cable for Modbus 2 plugged in to the right interface (SIO2) of the right F 865x central module.

It is also possible:

- To plug in the first BV 7040 cable for Modbus 1 to the right interface (SIO2) of the left F 865x central module.
- To plug in the second BV 7040 cable for Modbus 2 to the left interface (SIO1) of the right F 865x central module.

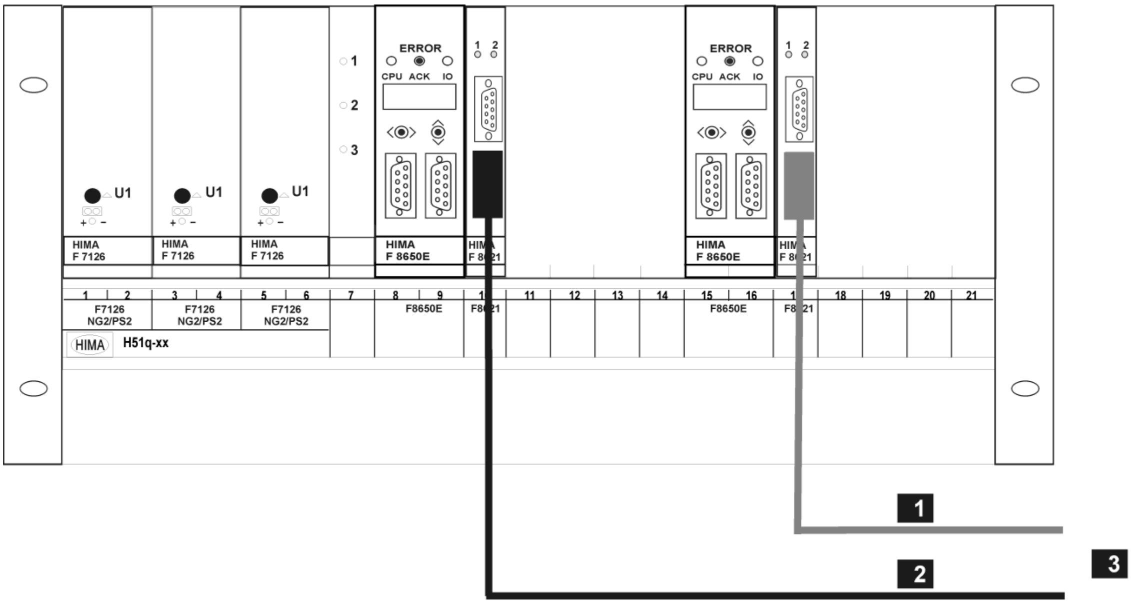
i This variant does not permit the use of logic-plan-controlled logging and/or of the 3964R Siemens protocol, since these functions need the interfaces of the central modules.

### Variant 6: Special Connection Variant with Co-processor Modules

This connection variant does not permit any event evaluation.

**For this reason, HIMA does not recommend using this connection variant!**

To implement a redundancy monitoring for the connection using the BV 7046 cable (Y cable), the receive counter of the interface in question must be evaluated in the user program (see Chapter 6.4.12.3).



1 Modbus 2
2 Modbus 1
3 To the third-party system

Figure 7: Variant 6, Redundant Connection via Co-Processor Modules

- Plug in the first BV 7046 cable (Y cable) for Modbus 1 to the upper interface 1 (SIO1) of the redundant F 8621A co-processor modules.
- Plug in the second BV 7046 cable (Y cable) for Modbus 2 to the lower interface 2 (SIO2) of the redundant F 8621A co-processor modules.

i Modbus variants for connecting to third-party systems other than those described in Chapter 6.4.12.1 and Chapter 6.4.12.2 are not approved and may lead to problems!

## 6.4.12.3 System Variables for Interface Receive Counters

Use the receive counters of the interfaces to perform the evaluations in the user program, and be able to detect if the relevant interface has failed. This can be done, e.g., by monitoring the receive counter for changes within a specific time period and considering no change as a failure of the corresponding interface.

**To create a system variable in ELOP II**

1. Create a new variable in the user program.
2. Double-click the new variable and open the *Variable Declaration* dialog box.
3. Enter UINT in the *Declaration* field.
4. Tick the **Tag Name** checkbox in the *Hardware Assignment* field.
5. It is now possible to perform the assignment:
   Example: *SIO.CU1/CM1.SIO1-Receive counter.*
   16 of these system variables exist.

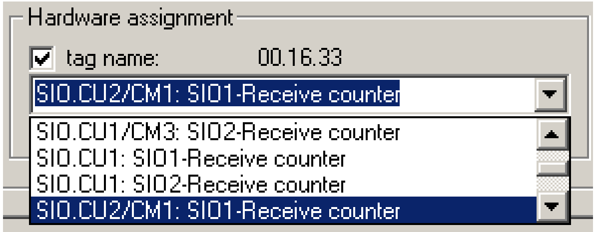



Figure 8: Assigning a Variable to a Receive Counter

### 6.4.13 Wiring the Modbus Cables for TCP/IP Connections

Refer to the data sheet for the F 8627X communication module for further details.

## 6.5 Modbus Function Codes of the Master

The Modbus function codes (request telegrams) allow the user to write and read variables in both directions. Individual variables or multiple consecutive variables can be read or written to.

Refer to the ELOP II online help for further details on how to configure the Modbus master.

### 6.5.1 Modbus Standard Function Codes

The HIMA Modbus master supports the following Modbus standard function codes:

| Element | Code | Type | Description |
|---|---|---|---|
| Read Coils | 01 | BOOL | Read multiple variables (BOOL) from the slave.<br>Read events from the H41q/H51q slaves. |
| READ DISCRETE INPUTS | 02 | BOOL | Read multiple variables (BOOL) from the slave. |
| Read Holding Registers | 03 | WORD | Read multiple variables of any type from the slave.<br>Read events from the H41q/H51q slaves. |
| Read Input Registers | 04 | WORD | Read multiple variables of any type from the slave. |
| WRITE SINGLE COIL | 05 | BOOL | Write one single signal (BOOL) in the slave. |
| WRITE SINGLE REGISTER | 06 | WORD | Write one single signal (WORD) in the slave.<br>Time synchronization of H41q/H51q slaves. |
| LOOP BACK DIAGNOSTIC TEST | 08 | | Diagnostics of the slave system.<br>All 21 diagnostic codes of function code 8 are possible. |
| Write Multiple Coils | 15 | BOOL | Write multiple variables (BOOL) in the slave. |
| Write Multiple Registers | 16 | WORD | Write multiple variables of any type in the slave. |

Table 26: Modbus Function Codes of the Master

i For further details on Modbus, refer to the *Modbus Application Protocol Specification* available at www.modbus.org.

## 6.6 Communication with the 3964R Protocol (Siemens Devices)

In contrast to the Modbus protocol, the 3964R Siemens protocol is not designed as bus system, but as a point-to-point connection.

HIMA recommends contacting Siemens for receiving the documentation on the 3964R protocol, and getting informed about peculiarities of the master in use.

The HIMA H41q and H51q PES can only be used as slave systems. The interfaces on the central module (1 or 2), but not the interfaces on the co-processor modules can be used for this data transmission type. Only the D data class (data function block) of the 3964R protocol is supported.
HIMA requires a byte for BCC (block check character) as part of the telegram.

HIMA does not support data transmission distributed over multiple transmissions, i.e., only telegrams having the coordination flag with the FFFF (hex) value may be transmitted (see telegram layout).

The variables to be read or written to, are defined in the resource (variable declaration) with the 3964R attribute.

Assignment of Boolean variables to data function blocks and data words:

A data word contains 16 Boolean variables.

Assignment of word variables (e.g., integer) to data words:

A data word contains a word variable.

Refer to *RES Docu (generated)* for the address that the master must provide to the slave to transfer a specific variable.

Abbreviations – The following abbreviations are used in the following chapter:

| | |
|---|---|
| DLE | Data link escape (if this byte is used in the user data, it must be repeated). |
| ETX | End of text (only the error check byte follows). |
| MSB | Most significant byte. |
| LSB | Least significant byte. |

### 6.6.1 Overview of the Functions of the 3964R Protocol

The Siemens 3964R protocol distinguishes between 2 functions:

- Writing to variables, SEND order, command A, data class D.
- Reading of variables, FETCH order, command E, data class D.

A maximum of 128 bytes can be read or written to at once.

### 6.6.2 Implemented Write Codes

Data is written in a SEND order to the slave (HIMA PES). The individual variables are addressed via data function blocks (DB) and data words (DW).

Send Telegram Structure (Siemens ⇒ HIMA):

| | ID | | Command type | Data class | Target address MSB | Target address LSB | Number of bytes MSB | Number of bytes LSB | Coordinating flag | | User data |
|---|---|---|---|---|---|---|---|---|---|---|---|
| hex | 00 | 00 | 41 | 44 | 04 | 00 | 00 | 20 | FF | FF | xx |
| ASCII | | | „A“ | „D“ | | | | | | | |

| User data (max. 128 bytes) | | DLE | ETX | Error check |
|---|---|---|---|---|
| xx | xx | 10 | 03 | 32 |

Structure of the response telegram - normal case

| | ID | | Error code MSB | Error code LSB | DLE | ETX | Error check |
|---|---|---|---|---|---|---|---|
| hex | 00 | 00 | 00 | 00 | 10 | 03 | 13 |

### 6.6.3 Implemented Read Codes

The data are read with a FETCH order. The individual variables are addressed via data function blocks (DB) and data words (DW).

Structure of the Receive Telegram (HIMA Æ Siemens):

| | ID | | Command type | Data class | Source address MSB | Source address LSB | Number of bytes MSB | Number of bytes LSB | Coordinating flag | | DLE | ETX | Error check |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| hex | 00 | 00 | 45 | 44 | 4B | 00 | 00 | 03 | FF | FF | 10 | 03 | 13 |
| ASCII | | | „E“ | „D“ | | | | | | | | | |

Structure of the response telegram - normal case:

| | ID | | Error code MSB | Error code LSB | User data (max. 128 bytes) | DLE | ETX | Error check |
|---|---|---|---|---|---|---|---|---|
| hex | 00 | 00 | 00 | 00 | | 10 | 03 | 42 |

### 6.6.4 Error Codes Sent to the Master

Structure of the response telegram in the event of a fault:

| | ID | | Error code MSB | Error code LSB | DLE | ETX | Error check |
|---|---|---|---|---|---|---|---|
| hex | 00 | 00 | xx | xx | 10 | 03 | |

Description of the error codes

| Error code | Description |
|---|---|
| 0 | No error |
| 1 | Format error: Faulty transmission structure, for instance:<br>▪ Incorrect checksum<br>▪ Incorrect coordinating flag<br>▪ Incorrect ID<br>▪ Incorrect type of command (neither A nor E)<br>▪ Incorrect data class (not D)<br>▪ Request with data (ED)<br>▪ Write command without data (AD)<br>▪ No double DLEs<br>▪ Telegram header > 10 bytes |
| 2 | Address error: The specified address is incorrect or invalid<br>(Variables not defined in the HIMA PES) |
| 3 | Errors connected to the number<br>▪ Number = 0<br>▪ Number > the number of defined variables<br>▪ Number > 128 bytes |

Table 27: Description of the Error codes

## 6.7 Logic-Plan-Controlled Logging

The logic-plan-controlled logging (LCL) is used to record events (signal changes with time) on the central module and print the events with configurable texts on a printer connected to the serial interfaces. Only interface 2 on the central module may be used for logic-plan-controlled logging. In the variable declaration, under **Events**, the *Logging LCL* property can be assigned to the individual variables. The events and texts are parts of the user program. The HK-LGP-3 software function block can be used to implement additional functions, see the description of the software function block provided in the ELOP II online help.

# 7 User Program Handling

Handling of the user program within the PES includes the processes of loading and deleting the user program and performing the online test.

The user program can be loaded in two different ways:

- Through a download
  The controller is in the STOP state.
- Through a reload
  The controller is in the RUN state.

i If code version number 0 is generated while compiling a program, the program must not be loaded into the controller!

Workaround: Modify and recompile the program to ensure that a new code version number not equal to 0 is generated. The changes must not modify the program function. For this reason, only objects that are independent from each other such as the inputs of an AND function block, may be graphically exchanged.

## 7.1 Download

This is the usual procedure for loading a new user program or one that was radically changed. Loading a new user program replaces the previous one. A command is used to cause the controller to enter the RUN state.

i Should an output module be deleted from the PADT's cabinet layout, but not removed from the subrack, the outputs retain the last value before the download.

To prevent this behavior, the outputs must be forced and thus reset, or the module must be removed, or power supply must be switched off.

## 7.2 Reload

Reload is intended for smaller modifications in the user program, e.g., for troubleshooting or enhancements. No restrictions exist for the extent of the changes.

After a more comprehensive change, the user program should be loaded by performing a download.

When a reload is planned and performed, the following points must be observed:

- A reload is only permitted after receiving consent from the test authority responsible for the acceptance test. When performing the reload, the person in charge must take further technical and organizational measures to ensure that the process is sufficiently monitored in terms of safety.
- Prior to performing the reload, use the C code comparator integrated in ELOP II to determine the changes in the user program compared to the user program still running.
- The changes in the user program and reload itself must be carefully tested on simulators prior to transferring them to the PES.
- If a logic part is deleted during a reload, e.g., a function controlling a physical output, the process image does not change. A variable, which is associated with that physical output during reload, receives the value available at that physical output as process image. For this reason, all outputs affected by the reload procedure must be switched off prior to performing a reload.

  To do so, these outputs must be set to FALSE during a first reload, and then they can be deleted during a second reload procedure (only forcing them is not sufficient!).

- A variable, which is removed from a physical output during a reload, is initialized.
  The necessary measures must be taken in the user program to ensure that such a variable is only used after being associated with a valid value.
- In a user program with a step sequence, deleting the active step and subsequently performing a reload causes the step enabling condition for the next step to be lost. The next step can thus no longer be reached and it is not possible to continue the step sequence!
- Changing names may impair the user program's proper functioning since the action has the same effect as deleting the previous object and inserting the new one:
  - Changing the name of a function block type causes the storing elements (such as flip-flops or SFC steps) of all instances of this function block type to be initialized and lose the current state.
  - Changing the name of a function block instance causes the storing elements (such as flip-flops or SFC steps) of the current instance to be initialized and lose the current state.
  - Changing the user program name has no effects.
- If the input variable (VAR_INPUT) of a function block is no longer written to after a reload, it retains its last value! The input variable is not automatically reset to FALSE / 0! This case occurs for instance, if the variable or assignment was deleted before the function block.

  This behavior concerns all the function blocks, but not functions.

  Workaround: Such an input must be connected to a new variable set to the required value.
- After a reload, all the variables with the *const* attribute are reset to their initial value, even if they have previously been set online to another value.
- During a reload, all the system parameters are reset to their configured value, even if they have previously been set online to another value. This may affect parameters such as the watchdog time, safety time or baud rate of the interfaces.
- Observe the following points during serial communication between PADT and PES:

  To ensure a successful reload, the baud rate setting in the project configuration must be consistent with the DIP switch setting on the central module!

If a reload may be performed to load the user program into the central module(s), the *Reloadable Code* message appears while the code generator is compiling the code.

Reloadability is lost in the following cases:

- If constants are added, e.g., variables with the CONST attribute.
- If modules are deleted or added in the control cabinet.
- If attributes of the following types are assigned more variables than deleted: HIPRO-N, HIPRO-S, BUSCOM, event, 3964R.
- If the basic addresses for BUSCOM are changed.
- If names of HIPRO-S variables are modified.
- If system variables' assignments are added or changed.

Exceptions

| System variables | Conditions for reload |
|---|---|
| IO.Acknowledge active<br>SYSTEM.Master Force inputs<br>SYSTEM.Master Force outputs<br>SYSTEM.Force switches outputs<br>SYSTEM.Illegal access<br>SYSTEM.Number of illegal accesses<br>SYSTEM.Run version<br>SYSTEM.Code version<br>SYSTEM.RAM/EPROM<br>HIBUS.*Resource name*.Receive counter<br>HIBUS.*Resource name*.Disturbed | Reload is always possible. |
| SIO.CU1.SIO1-Receive counter<br>SIO.CU1.SIO2-Receive counter<br>SIO.CU2.SIO1-Receive counter<br>SIO.CU2.SIO2-Receive counter<br>SIO.CU1.SIO1-Receive counter<br>SIO.CU1/CM1.SIO1-Receive counter<br>SIO.CU1/CM1.SIO2-Receive counter<br>SIO.CU1/CM2.SIO1-Receive counter<br>SIO.CU1/CM2.SIO2-Receive counter<br>SIO.CU1/CM3.SIO1-Receive counter<br>SIO.CU1/CM3.SIO2-Receive counter<br>SIO.CU2/CM1.SIO1-Receive counter<br>SIO.CU2/CM1.SIO2-Receive counter<br>SIO.CU2/CM2.SIO1-Receive counter<br>SIO.CU2/CM2.SIO2-Receive counter<br>SIO.CU2/CM3.SIO1-Receive counter<br>SIO.CU2/CM3.SIO2-Receive counter | Reload can be performed if one of the following conditions is met:<br>▪ The resource is an *E-Resource.*<br>▪ The *Always Reserve SIO Memory* parameter is activated.<br>▪ SIO parameters for SIO1 and/or SIO2 are not set to *Hardware Presetting*. |

Table 28: Influence of Certain System Variables on the Capability of Reload

## 7.2.1 Systems with 1 Central Module (Mono Systems)

While the user program is being reloaded, the I/O level may not be accessed, i.e., no I/O modules are read, written to or tested. During reload, the values previously valid are retained.

While a user program is being reloaded, the user program does not write to the export areas such that the read variables have the values written prior to performing the reload. During a reload, the interfaces can write to the import areas, but they are not processed.

i Reload must be completed within the safety time.
If necessary, this requirement must be verified on a simulator!

### 7.2.2 Systems with Redundant Central Modules

A reload may be performed in systems having redundant central modules without the restrictions mentioned for the systems equipped with one central module.

Reload operating sequence:

1. While the first central module is being reloaded, the second central module continues processing the user program in mono operation.
2. The newly loaded central module receives the current data from the operating central module and adopts mono operation with the new user program.
3. Once the second central module has been loaded, it receives the current data.
4. Both central modules start to operate redundantly.

---

i If a reload is performed for systems equipped with redundant central modules, the following actions might not be completed within the watchdog time:

- Variable values adopted from the redundant central module.
- Variable initialization.

This behavior must be precluded by verifying it on a simulator!

---

i The watchdog time for a H41qe/H51qe central module can be assessed with the following equation:

WDe = CT * 1.5 + D * 5.5 ms/kByte

| | |
|---|---|
| WDe: | Watchdog time (ms) for H41qe/H51qe |
| CT: | Maximum cycle time (ms) of the central module operating in RUN (displayed in the ELOP II Control Panel). |
| D: | Data size in kByte *Data Size (without SI Data)* (displayed in the ELOP II Compiler). |

The following applies to H41q/H51q central modules:

WD = CT * 1.7

| | |
|---|---|
| WD: | Watchdog time (ms) for H41q/H51q |
| CT: | Maximum cycle time (ms) of the central module operating in RUN mode (displayed in the ELOP II Control Panel). |

---

### 7.2.3 Generating a Code Version for Repeated Reload

Problem:

After changing the user program, new code is generated and loaded by performing a reload. In contrast to the user's expectations, a new code version (modified CRC) is created if no additional changes are performed and the code generator is **started again**.

This code version is usually not approved by the responsible test authority!

Explanation:

The reload procedure is based on a special shifting code. In other words, this shifting code is used to manage the differences between the loaded program code version and a new one. The code generator uses this table to decide if a program change can be loaded by performing a reload.

The shifting code is part of the created program code. It is created whenever a (reloadable) code is generated and describes the extent of the changes.

Therefore, a change to the shifting code also results in a change of the generated program code.

Procedure

| Action | Shifting code | PADT state, created version | PES state, loaded version |
|---|---|---|---|
| Initial state | - | Code version A | Code version A |
| Change, reloadable code generated | Changes result in a shifting code describing the changes. | Code version B, containing a new (modified) shifting code from code version A to code version B | Code version A |
| Load | - | | Code version B |
| No change, generate reloadable code | A shifting code is created although no changes were performed.<br><br>This shifting code is empty, but it results in a change of the shifting code content. (which had content and is now empty) | The code generator detects the shifting code change.<br>Result:<br>Code version C, containing a new (empty) shifting code from code version B to code version C | Code version B |
| Load | | | Code version C |
| No change, generate reloadable code | An empty shifting code is created, since no changes were performed.<br>The shifting code content remains unchanged (i.e., empty) | Code version C with empty shifting data code from code version C to code version C | Code version C |

Table 29: Procedure for Created Reloadable Code

Consequence:

To avoid unexpected system behaviors, HIMA recommends generating and loading the code twice whenever a change is performed to the program.

After this step, the code can be created as often as necessary (e.g., after restoring an archived application), but will no longer result in a new code version.

Under this condition, reload is always possible, even if no new code is generated.

This code version should also be used for a potential approval by the test authority.

A code version for repeated reload is also required to replace the operating system during operation, see Chapter 8.2.

## 7.3 Deleting the User Program

The user program can be deleted from within the front panel of the central module.

Premise: The controller is in RUN or STOP. The diagnostic panel displays RUN or STOP (default setting).

**To delete the user program**

1. Press the ⇓ button 8 times to enable the deletion of the user program, which is indicated on the central module through `-->`.
2. Press ⇒ to start the deletion, indicated as `ERASE APPLICATION`.
3. **Press** ⇒, ⇓ and ACK **simultaneously** to prepare the deletion of the user program.

   The central module displays STOP, and the CPU LED is lit.

   ☑ After pressing ⇓, the text ■■■■ is dislayed indicating that the user program is no longer available.
4. Pressing ACK results in the following actions:
   - The operating system is restarted.
   - The user program is actually deleted.

     `E001` indicates the deletion of the user program on central modules of type F 8651 and F 8653.

     In central modules of type F 8650 and F 8652, the count goes from `*103` to `*1F3` where `*` is represented by a rotary dash.

     At the end, the message `ERASE APPLICATION` is displayed.
   - Self-education runs if the corresponding requirements are met (see below).

► The user program is deleted.

## 7.4 Self-Education

*Self-education* refers to the capability of the operating system to learn the user program based on the user program loaded in the redundant central module. When a central module with no user program is started, it loads the user program from the central module redundant to it.

For this reason, if a faulty central module is being replaced in a redundant system, the user program need not be loaded into the new central module from the PADT. The new central module is inserted in the new subrack and automatically loads the user program from the redundant central module after the operating system is started for the first time.

The following requirements must be met for using the self-education feature:

1. The system structure must ensure high availability.
2. The redundant central module is operating in MONO mode.
3. The user program was generated for a high-availability resource.
4. The memory space of the new central module is not less than the space required by the user program resource type.
5. The flash memory of the new central module user program is empty, see Chapter 7.3.
6. The versions and CRCs of the operating systems are identical.
7. The position of the DIP switches on the new central module is the same as that on the redundant module.

If all these requirements are met, the central module loads the user program from the redundant module. After this step, the newly loaded central module is started up and enters the RUN state. This is signaled with error code 149 *Start-Up After Self-Education*.

During the self-education process, the redundant central module operates in MONO mode and enters the RUN state upon completion of self-education.

If the requirements are not met, the central module issues error code 164 *Flash Memory for User Program not Loaded*.

If errors have occurred during the loading procedure, e.g., communication problems between central modules, the central module issues error code 150 *Self-Education Aborted*.

If a central module with empty user program memory (e.g., a new one) is plugged in to a central module operating in MONO mode, and all other requirements are met, self-education is performed immediately!

## 7.5 Online Test (OLT)

It is possible to test the user program on the PES. The following options are available:

| Option | Description |
|---|---|
| Online test field | Online test fields are used to enable intermediary results of the arithmetic or logical connection to be displayed or forced. |
| Single-step mode | Enables the user program to be processed cycle per cycle |
| Break points on the function type | Stops the user program at each function type instance. |
| Break points on a function instance | Stops the user program at a specific function type instance. |

Table 30: Online Test Options the User Program

The online test fields can also be used during safety-related operation for monitoring the system.

For further information, refer to the ELOP II online help.

# 8 Loading the Operating System

The operating system is usually stored in a central module's flash EPROM and is delivered with the central module. In certain situations, it may be necessary to replace the existing operation system with an updated, technically enhanced version. Please contact HIMA to order updated versions.

If a HS/HRS system structure is used, this exchange can also occur while the PES is operating.

Loading a newer operating system version than currently loaded has no effect on the functionality of the application (user program). The application need not be checked.

However, if a previous version is exceptionally used to replace the current version, it might occur that functions used in the user program such as standard function blocks are not supported by the previous operating system version. If this happens, error handling becomes active and results in an error stop.

The operating system may only be loaded via **Ethernet** if an operating system as of V7.0-8 (05.34) is used in the central module(s).

i If an operating system prior to V7.0-8 (05.34) is loaded, communication with the PADT via TCP/IP is no longer possible!

**NOTICE**

**Notice! Unpredictable system behavior is possible if the system is operated neglecting the specified restrictions.**

**Prior to loading an operating system, observe the corresponding release notes.**

## 8.1 Load and User Program Stop (Interruption of Operation)

For systems without redundant central module, the user program must be interrupted for the duration of the loading procedure, since the central module must be stopped.

**To load the operating system**

1. Open the Control Panel.
2. Click the **OS Download** button.
   ☑ The Operating System Download dialog box appears.
3. Click the **Browse** button and select the file with the operating system to be loaded. Click **OK**.

► The operating system is loaded.

## 8.2 Load and no User Program Stop (no Interruption of Operation)

An HS/HRS system structure, in which an operating system version prior to (06.05) is loaded, allows the operating system of the central module to be replaced without stopping the user program.

If a version prior to (06.05) is loaded, one of the 3 versions (06.05), (07.14) or (07.30) must be loaded as interstage before loading the current operating system version without stopping the user program.

Alternatively: Stop the user program prior to performing the operating system download.

To this end, observe the described procedure.

i HIMA recommends loading the operating system offline, when possible, such as specified in Chapter 8.1.

Because of the many details to be taken into account, the operating system should only be replaced during operation by HIMA service personnel or experienced users!

i **Interruption of operation possible!**

If the operating system is loaded without stopping the user program, it is not 100% sure that no error stop occurs!

Claims for consequential damage and loss of profit of any kind regardless of their legal basis are excluded.

### 8.2.1 Requirements

- Prior to performing an operating system download without stopping the operating system, carefully check the effects that the changes made may have on the safe function of the entire plant!

  For instance, the download of a controller operating system load can result in delays or cause communication to be interrupted which, in turn, can affect other controllers.

  Observe the release notes of the operating system version that should be loaded!
- The following conditions apply to the computer used for replacing the operating system:
  - The PC must be operated on mains supply, battery operation must be avoided.
  - During replacement, only ELOP II must be running on the PC to avoid any interferences from other programs.
- The operating system should only be loaded directly using a data connection cable between the PC and the 2 central modules.

  Any attempt to load over some distance by using repeaters, telephone modems or the like, may result in problems!
- To prevent the PES from being interrupted during the operating system load over RS485, the ELOP II bus cable of 1 of the 2 central modules should be unplugged, such as described in the following section.

  The same effect can also be achieved for RS485 as well as Ethernet by marking the Transfer checkbox located in the Operating System Download dialog box only for the central module to be loaded.

  See also Table 31, notice 2 about the following scheme.

To determine whether the plant in consideration is suitable for loading the operating system during operation, check the plant in accordance with the following scheme. All relevant notices are located after the scheme.

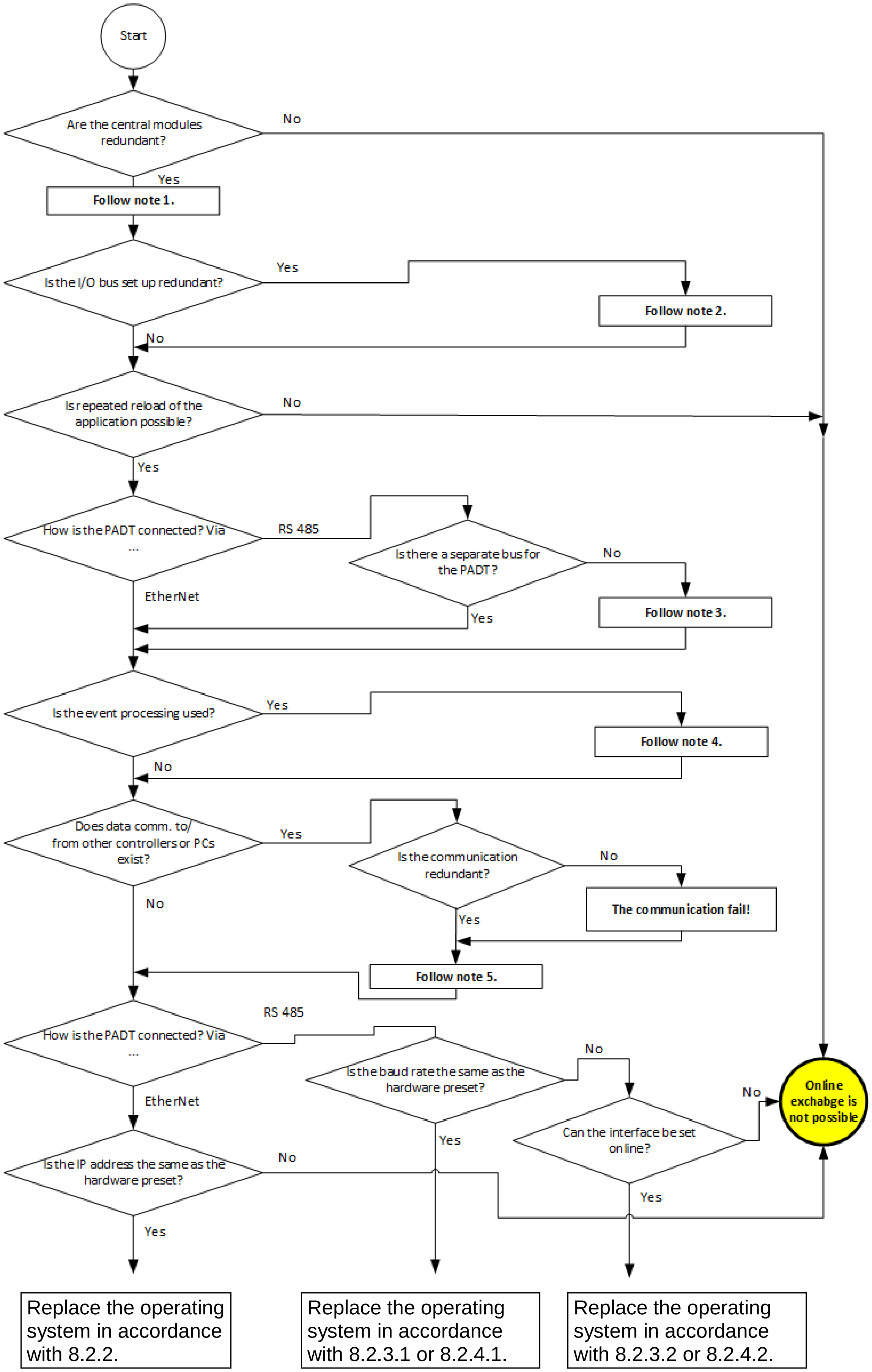



Figure 9: Determining if Operating System Download is Possible with no User Program Stop

Scheme notices:

| No. | Notice |
|---|---|
| 1. | A central module that enters the MONO state during the reload of redundant user programs (and thus during the online replacement of the operating systems), performs a reconfiguration process for each EN-M (F 8627X) assigned to it immediately after synchronization with the running central module in the following cases:<br>a) The EN-M (F 8627X) was de-energized since the last time it was in the RUN state (the green RUN LED is continuously on).<br>b) The EN-M (F8627X) itself has not been in the RUN state since the resource entered the RUN state.<br>c) The EN-M (F8627X) has not been in the RUN state since the last change of its Function configuration parameter (in the associated HK-COM-3 module, combined with Release Configuration = TRUE).<br>Point (a) applies to all OS versions, points (b) and (c) only apply as of (08.17).<br><br>A reconfiguration process may take several seconds and cause the EN-M to be able to transport messages with a delay (e.g., HIPRO-S).<br>However, since no communication has occurred in this time span via the redundant central module, communication connections (especially to HIPRO-S/ HIPRO-S V2) is possible due to the monitoring time being exceeded.<br>Recommendation: Prevent situations (a), (b) and (c) from occurring during the user program reload and online replacement of the operating system (which are events that do not occur in typical application scenarios) or configure sufficiently long monitoring times for communication connections that take into account the additional time required to reconfigure an EN-M (F 8627X). |
| 2. | In this case, an I/O bus is shut down during replacement!<br>Prior to shutting down the I/O bus, take all potential consequences of this action on the plant state into account! |
| 3. | The following condition only applies if the PADT is connected via RS485: The PADT needs an own bus (set even parity, 1 stop bit and the correct baud rate). If such a bus does not exist, it can be created as follows:<br>▪ An available interface is used.<br>▪ An occupied interface is temporarily reserved for connecting to the PADT, e.g., by:<br>  ▪ Directly connecting a PADT (PC) to the controller via an interface that is not used for the logic-plan controlled logging (LCL), the 3964R protocol or the Modbus master function.<br>  ▪ Using a Modbus. To do so, the controller must be disconnected from the communication partner and all potential consequences of this action on the plant state must be taken into account beforehand. |
| 4. | Notice that the event ring buffer in operating system versions as of (0213) is larger.<br>This causes the deletion of all events during the download of a previous operating system version to a version as of (0213). This is indicated by error code 199. For this reason, the operating system download with no stop of the user program should only be performed if the event ring buffer is empty. |
| 5. | For HIPRO-S: While the current data from a central module is being adopted into the central module with the newly loaded operating system, no transmissions are answered.<br>This effect must be taken into account in the communication partners when applying the communication monitoring function (monitoring time configured in the *HIPRO-S* tab of the resource's *Properties: Resource* dialog box or explicitly in the logic) |

Table 31: Notices for Determining Whether the Operating System Download is Possible with no User Program Stop

## 8.2.2 Connection via Ethernet (TCP/IP), Operating System Version as of (05.34)

If the PADT is connected to the controller via Ethernet using TCP/IP, proceed as follows:

i The PADT may only be connected via Ethernet, if the used resource ID is identical to the ID previously set (DIP switches on the central module, switches 1…5).

**To load the operating system into redundant central modules via Ethernet**

First central module:

1. Load the operating system. To do so:
   - Open the Control Panel.
   - In the *Operating System Download* dialog box, tick the *Transfer* field of the first central module and untick the *Transfer* field of the second central module.
   - Perform the **OS Download** for the first central module.

     Duration < 2 min.
2. Perform the user program reload:

   Select *Download/Reload* and load the user program in the first central module with **Reload**.

   ☑ The first central module enters the MONO operating state, the second the STOP operating state.

Second central module:

3. Load the operating system. To do so:
   - Open the Control Panel (if closed).
   - In the *Operating System Download* dialog box, untick the *Transfer* field of central module 1 and tick the *Transfer* field of central module 2.
   - In the lower part of the *Operating System Download* dialog box, select **First Load CU 2**.
   - Perform the **OS Download** for the second central module.

     Duration < 2 min.
4. If the user program loaded in Step 2 was changed with respect to the user program previously contained in the central module, proceed as follows:
   - Load the user program into the second central module.
   - Archive the project.
5. Verify the operating state of the controller:
   - Both central modules are operating redundantly (RUN).
   - Invoke the online test.

     If the online test can be accessed, the operating system and the user program are identical in both central modules of the controller.
   - Check the code versions displayed on the diagnostic panel: the two versions must be identical.

► The operating system is loaded into both central modules.

## 8.2.3 Connection via RS485, Operating System Version as of (0214)

The following cases may occur:

- The baud rate of the PADT bus and the baud rate set in the central module are identical.
- The baud rate of the PADT bus and the baud rate set in the central module differ.

### 8.2.3.1 Identical Baud Rates

If the baud rate of the PADT bus and the baud rate set in the central module are identical, proceed as follows:

**Load the operating system into both central modules via RS485 (identical baud rates).**

First central module:

1. Load the operating system. To do so:
   - Open the Control Panel.
   - Insert the PADT bus cable into the first central module and screw it tightly.
   - Remove the PADT bus cable from the second central module to prevent any unintentional load.
   - Perform the **OS Download** for the first central module

     Duration approx. 20 min at 57 600 baud.
2. Insert the PADT bus cable into the second central module and screw it tightly.
3. Perform the user program reload:
   - Select *Download/Reload* and use **Reload** to load the user program into the first central module.

   ☑ The first central module enters the MONO state, the second enters the STOP state.

Second central module:

4. To load the operating system:
   - Open the Control Panel (if closed)
   - Remove the PADT bus cable from the first central module to prevent any unintentional load.
   - Perform the OS Download for the second central module

     Duration approx. 20 min at 57 600 baud.
5. If the user program loaded in Step 3 was changed with respect to the user program previously contained in the central module, proceed as follows:
   - Load the user program into the second central module.
   - Archive the project.
6. Verify the operating state of the controller:
   - Both central modules are operating redundantly (RUN).
   - Invoke the online test.

     If the online test can be accessed, the operating system and the user program are identical in both central modules of the controller.
   - Check the code versions displayed on the diagnostic panel: the two versions must be identical.

► The operating system is loaded into both central modules.

### 8.2.3.2 Differing Baud Rates

If the baud rate of the PADT bus and the baud rate set in the central module (hardware presetting) are not identical, proceed as follows:

**To load the operating system into both central modules via RS485 (different baud rates)**

First central module:

1. Click the System Parameter button located in the Control Panel to open the dialog box for setting the baud rate for the interface. Enter the value defined in the central module using the hardware presetting.

i This setting may only be modified in the following cases:

- In the cabinet assignment, the *Manual* setting is enabled for the central module interface in use.
- The *Always Reserve SIO Memory* option is selected in the code generator.

   - Use **OK** to confirm. After approx. 15 seconds, ELOP II reports *No Communication*.

- Use **Cancel** to close the dialog box.
- Close the Control Panel

2. Change the baud rate for the PADT to the rate set in Step 1:
   - Select *Properties* from the context menu of configuration.
   - In the *Properties* dialog box, **Buses** tab, *s*elect the PADT bus and click **Edit**.
   - In the *Edit HIBUS* dialog box, **Parameter** tab, select the required baud rate.
3. Load the operating system:
   - Open the Control Panel.
   - Remove the PADT bus cable from the second central module to prevent any unintentional load.
   - Perform the **OS Download** for the first central module.

     Duration approx. 20 min at 57600 baud.
4. Close the Control Panel.
5. Reinsert the PADT bus cable in both central modules and screw it tightly.

6. Perform the user program reload:
   - Open the Control Panel.
   - Select *Download/Reload* and use **Reload** to load the user program into the first central module.
   - The first central module enters the MONO operating state, the second the STOP operating state.

   ☑ Communication with the PADT is lost!

Second central module:

7. Load the operating system:
   - Remove the PADT bus cable from the first central module to prevent any unintentional load.
   - Perform the **OS Download** for the second central module.

     Duration approx. 20 min at 57600 baud.
8. Close the Control Panel.
9. Reinsert the PADT bus cable on both central modules.
10. Reset the baud rate for the PADT to the rate originally set.
    - Select **Properties** from the context menu of configuration.
    - In the *Properties* dialog box, **Buses** tab, *s*elect the PADT bus and click **Edit**.
    - In the *Edit HIBUS* dialog box, **Parameter** tab, select the baud rate originally set.
11. If the user program loaded in Step 6 was changed with respect to the user program previously contained in the central module, proceed as follows:
    - Load the user program into the second central module.
    - Archive the project.
12. Verify the operating state of the controller:
    - Open the Control Panel.
    - Both central modules are operating redundantly (RUN).
    - Invoke the online test:

      If the online test can be accessed, the operating system and the user program are identical in both central modules of the controller.
    - Check the code version displayed on the diagnostic panel. They must be identical.

► The operating system is loaded into both central modules.

### 8.2.4 Connection via RS485, Operating System Version Prior to (0214)

If the loaded operating system version is prior to (0214), no F 8621(A)/25/26/27/28 communication modules may be inserted next to the central module being loaded during the loading process. For this reason, follow the steps described in the next sections.

#### 8.2.4.1 Identical Baud Rates

If the baud rate of the PADT bus and the baud rate set in the central module are identical, proceed as follows:

**To load the operating system into both central modules via RS485 (identical baud rates).**

First central module:

1. Load the operating system:
   - Remove the first central module.
   - **Remove** all F 8621(A)/25/26/27/28 next to the first central module.
   - Insert the first central module.
   - Open the Control Panel.
   - Insert the PADT bus cable into the first central module and screw it tightly.
   - Remove the PADT bus cable from the second central module to prevent any unintentional load.
   - Perform the **OS Download** for the first central module.

     Duration approx. 20 min at 57 600 baud.
   - Remove the first central module.
   - **Insert** all F8621(A)/25/26/27/28 next to the first central module.
   - Insert the first central module.
2. Insert the PADT bus cable into the second central module and screw it tightly.
3. Perform the user program reload.

Second central module:

4. Load the operating system:
   - Remove the second central module.
   - **Remove** all F 8621(A)/25/26/27/28 next to the second central module.
   - Insert the second central module.
   - Open the Control Panel.
   - Remove the PADT bus cable from the first central module to prevent any unintentional load.
   - Perform the **OS Download** for the second central module.

     Duration approx. 20 min at 57 600 baud.
   - Remove the second central module.
   - **Insert** all F 8621(A)/25/26/27/28 next to the second central module.
   - Insert the second central module.
5. If the user program loaded in Step 3 was changed with respect to the user program previously contained in the central module, proceed as follows:
   - Load the user program into the second central module.
   - Archive the project.
6. Verify the operating state of the controller:
   - Both central modules are operating redundantly (RUN).
   - Invoke the online test.

     If the online test can be accessed, the operating system and the user program are identical in both central modules of the controller.

- Check the code versions displayed on the diagnostic panel: the two versions must be identical.

► The operating system is loaded into both central modules.

#### 8.2.4.2 Differing Baud Rates

If the baud rate of the PADT bus and the baud rate set in the central module (hardware presetting) are not identical, proceed as follows:

**To load the operating system into both central modules via RS485 (different baud rates)**

First central module:

1. Preparation:
   - Remove the first central module.
   - Remove all F 8621(A)/25/26/27/28 next to the first central module.
   - Insert the first central module.
2. Click the **System Parameter** button located in the Control Panel to open the dialog box for setting the baud rate for the interface. Enter the value defined in the central module using the hardware presetting.

---

i This setting may only be modified in the following cases:
- In the cabinet assignment, the Manual parameter setting is enabled for the central module interface in use.
- The *Always Reserve SIO Memory* option is selected in the code generator.

---

   - Use **OK** to confirm; after approx. 15 seconds, ELOP II reports *No Communication*.
   - Click **Cancel** to close the dialog box.
   - Close the Control Panel
3. Change the baud rate for the PADT to the rate set in Step 2:
   - Select **Properties** from the context menu of configuration.
   - In the *Properties* dialog box, **Buses** tab, eselect the PADT bus and click **Edit**.
   - In the *Edit HIBUS* dialog box, **Parameter** tab, select the required baud rate.
4. Load the operating system:
   - Open the Control Panel.
   - **Remove** the PADT bus cable from the second central module to prevent any unintentional load.
   - Perform the **OS Download** for the first central module.

     Duration approx. 20 min at 57 600 baud.
   - Remove the first central module.
   - Insert all F8621(A)/25/26/27/28 next to the first central module.
   - Insert the first central module.
5. Close the Control Panel.
6. Reinsert the PADT bus cable on both central modules and screw it tightly.
7. Perform the user program reload:
   - Open the Control Panel once again.
   - Select **Download/Reload** and use **Reload** to load the user program into the first central module.
   - The first central module enters the MONO state, the second enters the STOP state.
   
   ☑ Communication with the PADT is lost!

Second central module:

8. Preparation:
   - Remove the second central module.

- **Remove** all F 8621(A)/25/26/27/28 next to the second central module.
- Insert the second central module.

9. To load the operating system:
   - Remove the PADT bus cable from the first central module to prevent any unintentional load.
   - Perform the **OS Download** for the second central module.
     Duration approx. 20 min at 57600 baud.
   - Remove the second central module.
   - **Insert** all F 8621(A)/25/26/27/28 next to the second central module.
   - Insert the second central module.
10. Close the Control Panel
11. Reinsert the PADT bus cable on both central modules and screw it tightly.
12. Reset the baud rate for the PADT to the rate originally set.
    - Right-clicht the **Configuration** and select **Properties** from the context menu.
    - In the *Properties* dialog box*,* **Buses** tab, *s*elect the PADT bus and click **Edit**.
    - In the *Edit HIBUS* dialog box, **Parameter** tab, select the baud rate originally set.
13. If the user program loaded in Step 7 was changed with respect to the user program previously contained in the central module, proceed as follows:
    - Load the user program into the second central module.
    - Archive the project.
14. Verify the operating state of the controller:
    - Open the Control Panel.
    - Both central modules are operating redundantly (RUN).
    - Invoke the online test:
      If the online test can be accessed, the operating system and the user program are identical in both central modules of the controller.
    - Check the code versions displayed on the diagnostic panel: the two versions must be identical.

► The operating system is loaded into both central modules.

# 9 Requirement Class and SIL

In ELOP II, one can set the requirement class RC (0…6) in the resource properties, but not the safety integrity level (SIL).

The following connection exists between the requirement classes and the safety integrity levels:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| NP: | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| SIL: | 0 | 1 | 1 | 1 | 2 | 3 | 3 |

If the requirement class is specified in ELOP II, the controller behaves such as described in the following section.

## 9.1 Requirement Class 0 Corresponds to SIL 0

If requirement class 0 is set, the H41q/H51q system behaves as follows:

- A test monitoring error (error code 7) is only report as diagnostic information, but it does not result in an error stop.
- The safe areas of pooled variables may be written using HIPRO-N and HIPRO-S/HIPRO-S V2.

## 9.2 Requirement Classes 1 to 3 Correspond to SIL 1

If requirement class 4, 5 or 6 is set, the H41q/H51q system behaves as follows:

- Errors in the communication path between central modules occurred while starting up one central module do not result in an error stop. Communication between central modules is checked again in the next cycle.
- A test monitoring error (error code 7) results in an error stop.
- The safe areas of pooled variables may only be written to using HIPRO-S/HIPRO-S V2.

## 9.3 Requirement Classes 4 to 6 Correspond to SIL 2 and SIL 3

If requirement class 4, 5 or 6 is set, the H41q/H51q system behaves as follows:

- Range overrun or underrun through an external write transmission results in the message *Unauthorized Access* in ELOP II.
- The detection of errors in the communication path between central modules while starting up a central module results in an error stop with error code 101.
- The limits of the event ring buffer are cyclically checked, if they are underrun or overrun, the event buffer is reinitialized and error code 199 is issued as diagnostic information.
- A test monitoring error (error code 7) results in an error stop.
- The safe areas of pooled variables may only be written to using HIPRO-S/HIPRO-S V2.

# 10 Operating System Limits

No statements are made regarding the system limits as imposed by ELOP II.

No difference exists between the system limits of H41qe and H51qe.

| No. | Designation | Limit value | | |
|---|---|---|---|---|
| 1 | **Maximum data and program memories** | | | |
| | Program memory: | Max. | 1020 KB | |
| | Data memory: | Max. | 320 KB | (Total) |
| | Safety-related data | Max. | 64 KB | |
| | Other data (variables, etc.) | Max. | 256 KB | |
| 2 | **Minimum cycle time of the systems** (without program logic processing and I/O tests) | Mono<br>Redundant | | 7 ms<br>27 ms |
| 3 | **Maximum number of variables in the system**<br>No restrictions exist, except for the memory limits specified in point 1. | | | |
| 4 | **Maximum number and mode of transmission of BUSCOM variables**<br>Transmission is possible using Modbus, PROFIBUS and OPC | | | |
| | Digital variables (BOOL) | Import<br>Export | | 6144<br>6144 |
| | Analog variables<br>(WORD, BYTE, UINT, INT, USINT, SINT, REAL) | Import<br>Export | | 6144<br>6144 |
| | REAL variables are mapped to two variables of Register Modbus type and must be considered as 2 when calculating the number.<br>Every two BYTE variables are packed to one variable of Register Modbus type. | | | |
| 5 | **Maximum number of variables** | | | |
| | Variables being able to trigger events | Max. 2048 | | |
| | Buffer size | 500 events + overflow marker | | |
| 6 | **Maximum size of LCL texts**<br>The sum of the logic-controlled logging texts must not exceed 64 KB. | | | |
| 7 | **Maximum number and mode of transmission of HIPRO-N variables** | | | |
| | Transmission performed by PES master: the PES master project is limited to | | | Max. 36 KB |
| | The transmission of general data types ANY_BIT and ANY_NUM is allowed, but not in the DOUBLE and LONG variants; the transmission of arrays, structures and user-defined types is not allowed (example: WORD, counterexample: LREAL).<br>Transmission of HIPRO-N variables via safe**ethernet** is not possible. | | | |
| 8 | **Maximum number and mode of transmission of HIPRO-N variables**<br>Transmission performed by the PES master or via safe**ethernet** (not simultaneously!)<br>Data types such as specified in point 7 (ANY_BIT and ANY_NUM, not in DOUBLE and LONG variants) | | | |
| | Total data volume | For each PES master project | | Max. 36 KB |
| | | safe**ethernet** | | Not limited |
| | Data volume for each transmission | 505 bytes | | |
| | The relationship described with the following formula exists for the maximum number of variables of the different data types: $n_1 + n_2 + n_3 \leq 505$, where:<br>$n_1$ = Number of BOOL variables/8, rounded up to the next integer. BOOL variables are packed when they are transferred.<br>$n_2$ = number of BYTE variables<br>$n_3$ = 2*number of 16-bit variables (WORD, INT, UINT) | | | |

| No. | Designation | Limit value | |
|---|---|---|---|
| **9** | **Transmission restriction for Modbus (type and number of variables)**<br>Use the HIMA standard function block HK-MMT-3. | | |
| | **Modbus master:** | As of BS41q/51q V7.0-8 (07.14): | Up to BS41q/51q V7.0-8 (07.14): |
| | UINT | 127 | 120 |
| | BOOL | 2040 | 1920 |
| | **Modbus slave:** | | |
| | UINT | 127 | |
| | BOOL | 2040 | |
| **10** | **Transmission restriction for OPC (type and number of variables)**<br>No restrictions other than those specified in point 4. | | |
| **11** | **Transmissions restriction for HIPRO-S variables via PES master or safe**ethernet<br>Simultaneous operation via PES master and safe**ethernet** not permitted.<br>Caution: If BUSCOM variables should be transmitted (e.g., via F 8627X/OPC or F 8628X/PROFIBUS DP) in addition to HIPRO-S data (via PES master or safe**ethernet**), the HK-COM-3 function block must be used!<br>The condition specified in point 8 applies to the PES master. | | |

Table 32: Operating System Limits

# 11 System Variables

System variables can be 16-bit or 1-bit values. UINT and WORD data types are allowed for 16-bit values. For 1-bit values, only the BOOL data type is allowed.

Through the system variables, the user program receives information from the system or provides information to the system. The following system variables are available:

| System variables (alphabetically) | Data type | Use | See |
|---|---|---|---|
| IO.Error | BOOL | READ | Page 74 |
| IO.Error code IO bus 1 | UINT | READ | Page 75 |
| IO.Error code IO bus 2 | UINT | READ | Page 76 |
| IO.Faulty position IO bus 1 | UINT | READ | Page 76 |
| IO.Faulty position IO bus 2 | UINT | READ | Page 76 |
| IO.Acknowledge | BOOL | WRITE | Page 74 |
| IO.Acknowledge active | BOOL | READ | Page 74 |
| HIBUS.Resource name.Receive counter | UNIT | READ | Page 75 |
| HIBUS.Resource name.Error | BOOL | READ | Page 74 |
| SIO.CU1: SIO1 receive counter | UINT | READ | Page 75 |
| SIO.CU1: SIO2 receive counter | UINT | READ | Page 75 |
| SIO.CU2: SIO1 receive counter | UINT | READ | Page 75 |
| SIO.CU2: SIO2 receive counter | UINT | READ | Page 75 |
| SIO.CU1/CM1.SIO1 receive counter | UINT | READ | Page 75 |
| SIO.CU1/CM1.SIO2 receive counter | UINT | READ | Page 75 |
| SIO.CU1/CM2.SIO1 receive counter | UINT | READ | Page 75 |
| SIO.CU1/CM2.SIO2 receive counter | UINT | READ | Page 75 |
| SIO.CU1/CM3.SIO1 receive counter | UINT | READ | Page 75 |
| SIO.CU1/CM3.SIO2 receive counter | UINT | READ | Page 75 |
| SIO.CU2/CM1.SIO1 receive counter | UINT | READ | Page 75 |
| SIO.CU2/CM1.SIO2 receive counter | UINT | READ | Page 75 |
| SIO.CU2/CM2.SIO1 receive counter | UINT | READ | Page 75 |
| SIO.CU2/CM2.SIO2 receive counter | UINT | READ | Page 75 |
| SIO.CU2/CM3.SIO1 receive counter | UINT | READ | Page 75 |
| SIO.CU2/CM3.SIO2 receive counter | UINT | READ | Page 75 |
| SYSTEM.Number of illegal accesses | UINT | READ | Page 76 |
| SYSTEM.Code version | UINT | READ | Page 76 |
| SYSTEM.Single channel operation | BOOL | READ | Page 74 |
| SYSTEM.Error code | UINT | READ | Page 76 |
| SYSTEM.Fault Mask 1 | UINT | READ | Page 76 |
| SYSTEM.Fault Mask 2 | UINT | READ | Page 76 |
| SYSTEM.Force switches outputs | BOOL | READ | Page 74 |
| SYSTEM.Force switches inputs | BOOL | READ | Page 74 |
| SYSTEM.Master force outputs | BOOL | READ | Page 74 |
| SYSTEM.Master force inputs | BOOL | READ | Page 74 |
| SYSTEM.Logic emergency off | BOOL | WRITE | Page 74 |
| SYSTEM.Normal | BOOL | READ | Page 74 |
| SYSTEM.RAM/EPROM | UINT | READ | Page 76 |
| SYSTEM.Run version | UINT | READ | Page 76 |
| SYSTEM.Illegal access | BOOL | READ | Page 74 |

Table 33: System Variables in Alphabetical Order

## 11.1 READ System Variables of BOOL Type

READ system variables provide information from the operating system to the user program.

| System variables | Description |
|---|---|
| IO.Error | TRUE, if the operating system detected faults in one or multiple testable I/O modules. This system variable shows an I/O fault. |
| IO.Acknowledge active | TRUE means that the error acknowledgment was activated, either by pressing the ACK key, or by setting the *IO.Acknowledge* system variable to TRUE.<br>The system variable is TRUE for one cycle. |
| HIBUS.*Resource name*.Error | This system variable is available for each resource with configured safety-related communication via HIPRO. *Resource Name* is replaced by the actual resource name.<br>The system variable is TRUE, if the specified resource did not receive any data within the configured monitoring time. |
| SYSTEM.Force switches outputs | This system variable is TRUE if at least output variable is forced. An output variable is a variable assigned with a tag name and allocated to an output channel. |
| SYSTEM.Force switches inputs | This system variable is TRUE if at least one input variable is forced. An input variable is a variable assigned with a tag name and allocated to an input channel. |
| SYSTEM.Master force outputs | This system variable is TRUE if the FORCE main switch for outputs is on. |
| SYSTEM.Master force inputs | This system variable is TRUE if the FORCE main switch for inputs is on. |
| SYSTEM.Illegal access | TRUE for the duration of a cycle if it was attempted to perform an illegal function.<br>The functions legal are configured in the **Safety** tab of the resource's properties. |
| SYSTEM.Single channel | TRUE, if a central module failed in a system with two central modules. |
| SYSTEM.Normal | TRUE if the system is free of faults. This system variable is used for displaying the general system status. |

Table 34: READ System Variables of BOOL Type

## 11.2 WRITE System Variables of BOOL Type

WRITE system variables are used by the user program to transmit information to the operating system.

| System variables | Description |
|---|---|
| IO.Acknowledge | If this system variable is set to TRUE, it indicates that a fault was acknowledged. The following functions are performed.<br>▪ Acknowledgment of a displayed I/O fault<br>The fault indicators are reset and the system is re-checked. If the system once again detects a fault, the position of the faulty module is displayed.<br>▪ The test routines of the testable I/O modules previously switched off are reconnected. |
| SYSTEM.Logic emergency off | If it is set to TRUE, the entire system is shut down. All outputs are de-energized. The system immediately enters the safe state.<br>The system variable can be connected to an external signal or a signal generated from the logic.<br>Press the ACK key on the central module to restart the system (system in RUN). |

Table 35: WRITE System Variables of BOOL Type

## 11.3 READ System Variables of UINT/WORD Type

READ system variables provide information from the operating system to the user program.

UINT or WORD data type may be used for the following system variables. The OLT field displays a decimal or hexadecimal value, depending on the data type in use.

| System variables | Description |
|---|---|
| HIBUS.*Resource name*.Receive counter | This system variable is available for each resource with configured safety-related communication via HIPRO. *Resource name* is replaced by the actual resource name.<br>Each safety-related transmission increases the value of the receive counter. The values are within the range 0…65535. Once the end value was achieved, the counter restarts by 0. |
| SIO receive counter | The following conditions apply to all SIO receive counters:<br>The receive counter increases by 1 whenever a transmission is received on this interface. The range of values is 0…65535 or 0000…FFFF. Once the maximum value has been achieved, the counter is reset to 0. |
| SIO.CU1: SIO1 receive counter | Receive counter for the first interface of the left central module. |
| SIO.CU1: SIO2 receive counter | Receive counter for the second interface of the left central module. |
| SIO.CU1/CM1.SIO1 receive counter | Receive counter for the first interface of the first co-processor allocated to the left central module. |
| SIO.CU1/CM1.SIO2 receive counter | Receive counter for the second interface of the first co-processor allocated to the left central module. |
| SIO.CU1/CM2.SIO1 receive counter | Receive counter for the first interface of the second co-processor allocated to the left central module. |
| SIO.CU1/CM2.SIO2 receive counter | Receive counter for the second interface of the second co-processor allocated to the left central module. |
| SIO.CU1/CM3.SIO1 receive counter | Receive counter for the first interface of the third co-processor allocated to the left central module. |
| SIO.CU1/CM3.SIO2 receive counter | Receive counter for the second interface of the third co-processor allocated to the left central module. |
| SIO.CU2: SIO1 receive counter | Receive counter for the first interface of the right central module. |
| SIO.CU2: SIO2 receive counter | Receive counter for the second interface of the right central module. |
| SIO.CU2/CM1.SIO1 receive counter | Receive counter for the first interface of the first co-processor allocated to the right central module. |
| SIO.CU2/CM1.SIO2 receive counter | Receive counter for the second interface of the first co-processor allocated to the right central module. |
| SIO.CU2/CM2.SIO1 receive counter | Receive counter for the first interface of the second co-processor allocated to the right central module. |
| SIO.CU2/CM2.SIO2 receive counter | Receive counter for the second interface of the second co-processor allocated to the right central module. |
| SIO.CU2/CM3.SIO1 receive counter | Receive counter for the first interface of the third co-processor allocated to the right central module. |
| SIO.CU2/CM3.SIO2 receive counter | Receive counter for the second interface of the third co-processor allocated to the right central module. |
| IO.Error code IO bus 1 | This system variable displays the faulty channels of the module specified in the *IO.Faulty position 1. IO bus* system variable. This function is only available if the module is provided with line diagnostics.<br>Values, see Table 37. |

| System variables | Description |
|---|---|
| IO.Faulty position IO bus 1 | This system variable contains the position of a faulty I/O module on the first I/O bus. The value corresponds to the bus number, the subrack and position of the module. If several modules are faulty, the module with the lowest position is displayed.<br>1405 means bus 1, subrack 4, position 05.<br>To ensure that the value is displayed in this format, use UINT as data type. |
| IO.Error code IO bus 2 | This system variable displays the faulty channels of the module specified in the *IO.Faulty position 2. IO bus* system variable. This function is only available if the module is provided with line diagnostics.<br>Values, see Table 38. |
| IO.Faulty position IO bus 2 | This system variable contains the position of a faulty I/O module on the second I/O bus. The value corresponds to the bus number, the subrack and position of the module. If several modules are faulty, the module with the lowest position is displayed.<br>1405 means bus 1, subrack 4, position 05.<br>To ensure that the value is displayed in this format, use UINT as data type. |
| SYSTEM.RAM/EPROM | This system variable shows whether the force information, constants and safety parameters were stored into the RAM and can thus be changed during operation. For details, refer to Table 39. |
| SYSTEM.Run version | This system variable provides the current RUN version of the resource.<br>Use the WORD data type for this variable to ensure that the image represented is identical to the image represented on the diagnostic panel of the central module. |
| SYSTEM.Code version | This system variable provides the current code version of the resource.<br>Use the WORD data type for this variable to ensure that the format is identical to that displayed on the diagnostic panel of the central module. |
| SYSTEM.Illegal accesses | This system shows how often it has been attempted to call an illegal action or function. |
| SYSTEM.Error code | The error code is used to allow a more detailed analysis of the fault occurred. Refer to Chapter 12.4 for details on the meaning of the error code.<br>This variable only reports an error code if a bit is set in *SYSTEM.Fault maske1*. The displayed fault code reports the **last** fault. However, this need not be a fault allocated to a bit of *SYSTEM.Fault mask1*.<br>For a more detailed analysis, refer to the error code history provided in the PADT. |
| SYSTEM.Fault mask 1 | Fault mask 1 displays faults detected in the central modules and in the I/O bus. Refer to Table 40 for more details on the error bits: |
| SYSTEM.Fault mask 2 | Fault mask 2 displays general power supply disturbances, fault in the co-processor modules, the active noise blanking and the allocation of faults to central modules. Refer to Table 41 for more details on the error bits: |

Table 36: READ System Variables of UINT/WORD Type

## 11.3.1 Values for the *IO.Error Code 1. I/O Bus* System Variable

| Bit no. 1…8 | dec | hex | Error |
|---|---|---|---|
| 00 000 000 | 0 | 0 | No error |
| 00 000 001 | 1 | 1 | Error within the circuit, channel 1 |
| 00 000 010 | 2 | 2 | Error within the circuit, channel 2 |
| 00 000 100 | 4 | 4 | Error within the circuit, channel 3 |
| 00 001 000 | 8 | 8 | Error within the circuit, channel 4 |
| 00 010 000 | 16 | 10 | Error within the circuit, channel 5 |
| 00 100 000 | 32 | 20 | Error within the circuit, channel 6 |
| 01 000 000 | 64 | 40 | Error within the circuit, channel 7 |
| 10 000 000 | 128 | 80 | Error within the circuit, channel 8 |
| 11 111 111 | 255 | FF | Module faulty |

Table 37: Values for the *IO.Error Code 1. I/O Bus* System Variable

If multiple external circuits are faulty, the sum of the corresponding values is displayed.

## 11.3.2 Values for the *IO.Error Code 2. I/O Bus* System Variable

| Bit no. 1…8 | dec. | hex | Error |
|---|---|---|---|
| 00 000 000 | 0 | 0 | No error |
| 00 000 001 | 1 | 1 | Error within the circuit, channel 1 |
| 00 000 010 | 2 | 2 | Error within the circuit, channel 2 |
| 00 000 100 | 4 | 4 | Error within the circuit, channel 3 |
| 00 001 000 | 8 | 8 | Error within the circuit, channel 4 |
| 00 010 000 | 16 | 10 | Error within the circuit, channel 5 |
| 00 100 000 | 32 | 20 | Error within the circuit, channel 6 |
| 01 000 000 | 64 | 40 | Error within the circuit, channel 7 |
| 10 000 000 | 128 | 80 | Error within the circuit, channel 8 |
| 11 111 111 | 255 | FF | Module faulty |

Table 38: Values for the *IO.Error Code 2. I/O Bus* System Variable

If multiple external circuits are faulty, the sum of the corresponding values is displayed.

## 11.3.3 Values for the *SYSTEM.RAM/EPROM* System Variable

| Value | | Operation of left central module | Operation of right central module | Parameters stored in/ Changeable online |
|---|---|---|---|---|
| hex | dec | | | |
| 1 | 1 | Mono | -[1] | RAM, Yes |
| 100 | 256 | - | Mono | |
| 101 | 257 | Redundant | | |
| 2 | 2 | Mono | - | EPROM, No |
| 200 | 512 | - | Mono | |
| 202 | 514 | Redundant | | |
| [1] - means STOP state or central module is not inserted. | | | | |

Table 39: Values for the *SYSTEM.RAM/EPROM* System Variable

## 11.3.4 Values for the *SYSTEM.Fault Mask 1* System Variable

| Error bit 1…16 | hex | dec | Type of error |
|---|---|---|---|
| 0000 0000 0000 0000 | 0 | 0 | No error |
| 0000 0000 0000 0001 | 1 | 1 | CPU |
| 0000 0000 0000 0010 | 2 | 2 | Hardware clock disturbed - during start-up |
| 0000 0000 0000 0100 | 4 | 4 | Hardware watchdog |
| 0000 0000 0000 1000 | 8 | 8 | Memory error |
| 0000 0000 0001 0000 | 10 | 16 | Program crash |
| 0000 0000 0010 0000 | 20 | 32 | Timeout |
| 0000 0000 0100 0000 | 40 | 64 | Deviation hardware clock not tolerable |
| 0000 0000 1000 0000 | 80 | 128 | Hardware clock disturbed - during operation |
| 0000 0001 0000 0000 | 100 | 256 | Connection to I/O level |
| 0000 0010 0000 0000 | 200 | 512 | Power supply monitoring |
| 0000 0100 0000 0000 | 400 | 1024 | Address test I/O subrack |
| 0000 1000 0000 0000 | 800 | 2048 | Time delay other CU |
| 0001 0000 0000 0000 | 1000 | 4096 | Output not set to 0 during start-up |
| 0010 0000 0000 0000 | 2000 | 8192 | Deviation hardware clock tolerable |
| 0100 0000 0000 0000 | 4000 | 16384 | Not used |
| 1000 0000 0000 0000 | 8000 | 32768 | Memory not identical |

Table 40: Values for the *SYSTEM.Fault Mask 1* System Variable

If multiple faults occur simultaneously, the output value displays the error bits at the corresponding positions. Multiple bits may thus be set at the same time.

Faults on the co-processor modules are only displayed if they are defined in the cabinet. No errors are displayed for the F 8627 (X) and F 8628 (X) communication modules.

## 11.3.5 Values for the *SYSTEM.Fault Mask 2* System Variable

| Error bit 1…16 | hex | dec | Type of error |
|---|---|---|---|
| 0000 0000 0000 0000 | 0 | 0 | No error |
| 0000 0000 0000 0001 | 1 | 1 | Voltage monitoring 3.3 V CU 1 [1)] |
| 0000 0000 0000 0010 | 2 | 2 | Voltage monitoring 3.3 V CU 2 [1)] |
| 0000 0000 0000 0100 | 4 | 4 | Power supply unit 1 |
| 0000 0000 0000 1000 | 8 | 8 | Power supply unit 2 |
| 0000 0000 0001 0000 | 10 | 16 | Power supply unit 3 |
| 0000 0000 0010 0000 | 20 | 32 | Noise blanking is active[2)] |
| 0000 0000 0100 0000 | 40 | 64 | Error in central module 1 |
| 0000 0000 1000 0000 | 80 | 128 | Error in central module 2 |
| 0000 0001 0000 0000 | 100 | 256 | Co-processor module 1 next to CU 1 |
| 0000 0010 0000 0000 | 200 | 512 | Co-processor module 2 next to CU 1 |
| 0000 0100 0000 0000 | 400 | 1024 | Co-processor module 3 next to CU 1 |
| 0000 1000 0000 0000 | 800 | 2048 | Co-processor module 1 next to CU 2 |
| 0001 0000 0000 0000 | 1000 | 4096 | Co-processor module 2 next to CU 2 |
| 0010 0000 0000 0000 | 2000 | 8192 | Co-processor module 3 next to CU 2 |
| 0100 0000 0000 0000 | 4000 | 16384 | Back-up battery on central module 1 |
| 1000 0000 0000 0000 | 8000 | 32768 | Back-up battery on central module 2 |

1) This bit only applies to central modules as of F 8650X, F 8651X, F 8652X, F 8653X, otherwise it has no meaning

2) If this bit is set, System.normal is not set to FALSE!

Table 41: Values for the *SYSTEM.Fault Mask 2* System Variable

If multiple faults occur simultaneously, the output value displays the error bits at the corresponding positions. Multiple bits may thus be set at the same time.

Faults on the co-processor modules are only issued if they are defined in the cabinet.

# 12 Diagnostic Panel

The diagnostic panel located on the front side of the central module is composed of the following parts:

- A 4-digit alphanumeric display.
- 2 LEDs labeled *IO* and *CPU*.

Additionally, 2 keys can be used to request information from the PES and are described below. The ⇓/⇑ key is used to select the next higher or lower level, the ⇐/⇒ key to select other information at the same level.

An overview of the diagnostic panel levels is provided at the end of the document.

The key sequence specified in the *To call the info* column always refers to the initial position in which the operating state is displayed, e.g., RUN, STOP

## 12.1 Information Accessible in RUN

(In RUN: the *CPU* and *IO* LEDs are not lit.)

| Indicators | | Explanation | To call the info |
|---|---|---|---|
| Text | Example | | |
| 3V3! | ---- | 3.3 V supply voltage out of the allowed limits | |
| BATI | ---- | Voltage of RAM buffer battery on central module too low | |
| BOOT-ID | | CRC of the boot sector | 4x⇓, 2x⇒ |
| BS41q/51q | | | |
| V7.0-8 | | | |
| BS41q/51q<br>V7.0-8 (09.10) | ---- | Operating system labeling<br>Version of the operating system<br>(Operating system edition) | 4x⇓ |
| BDSW | 5711 | Displayed for baud rate, parity, stop bits<br>Settings on central module<br>Refer to Table 43 for details. | 1x⇓, 3x⇒, 6x⇓ |
| BN | 5 | Bus station number (BSN),<br>Setting on the central module (switch 1…5) | 1x⇓, 3x⇒ |
| | - | Text displayed if the BSN was set out of the permitted range (0…31) | |
| BNSW | 0001 | Bus station number, setting on the central module, switch 1…5 | 1x⇓, 3x⇒, 2x⇓ |
| ID | 3 | ID, resource ID, 1…99 | 1x⇓, 3x⇒, 1x⇓ |
| | - | Text displayed if the ID was set out of the permitted range (1…99) | |
| IDSW | 0001 | ID, setting on the central module (switch 1…7) | 1x⇓, 3x⇒, 3x⇓ |
| EPROM-CRC | | Operating system CRC<br>Compare it to the value specified in the operating system safety case | 4x, 1x⇒ |
| CB1<br>CB2<br>CB3 | | For internal diagnostic purposes | 5x⇓<br>6x⇓<br>7x⇓ |
| CB11<br>CB12<br>CB21<br>CB22<br>CB31<br>CB32 | 5711<br>5711<br>5711<br>5711<br>5711<br>5711 | Baud rate, parity, stop bits displayed for co-processor modules 1…3, serial connectors 1…2.<br>Refer to Table 43 for details. | 1x⇓, 3x⇒, 7x⇓<br>1x⇓, 3x⇒, 8x⇓<br>1x⇓, 3x⇒, 9x⇓<br>1x⇓, 3x⇒, 10x⇓<br>1x⇓, 3x⇒, 11x⇓<br>1x⇓, 3x⇒, 12x⇓ |
| CODE-VERSION | AC34 | Code version number | 1x⇓, 1x⇒ |

| Indicators | | Explanation | To call the info |
|---|---|---|---|
| Text | Example | | |
| C.TIME | 0064 | Cycle time in ms | 1x⇓, 4x⇒ |
| DATE | 0212 | Date, day/month | 1x⇓, 8x⇒ |
| F | 47 | The new entry is displayed in the history.<br>Refer to the List of Error Codes in Chapter 12.4<br>(⇓ provides additional information, depending on the error code) | 1x⇓ |
| | | Indication of current fault<br>ZB/CU: CPU<br>ZB/CU: MEMORY<br>ZB/CU: REALTIME CLOCK<br>ZB/CU: COUPLING UNIT<br>ZB/CU: CLOCK<br>LOGIC OFF<br>DISTURBANCE BLANKING<br>FATAL ERROR<br>W-DOG<br>COUPLING UNIT / OTHER<br>Extended-FC<br>POWER SUPPLY MONITORING | 1x⇓, 1x⇒ |
| | | For internal diagnostic purposes:<br>Additional information | 1x⇒, 2 x⇓ |
| F197/188:POS | 1403 | Position of the lastly disturbed I/O module<br>(Noise blanking) | 1x⇐ |
| F3349:POS | 1101 | Position of F 3349 module | 3x⇐ |
| F3349:CODE | HHHH<br>HHHH | No normal operation F 3349<br>Displayed: 8-digit hex code | 3x⇐<br>1x⇓ |
| F3349:Info1 | HHHH<br>HHHH | F3349 additional information 1<br>Displayed: 8-digit hex code | 3x⇐<br>2x⇓ |
| F3349:Info2 | HHHH<br>HHHH | F3349 additional information 2<br>Displayed: 8-digit hex code | 3x⇐<br>3x⇓ |
| FX220:POS | 1403 | I/O module of type F 5220, F 6220, F 6221 | 2x⇐ |
| ■■■■ | | The central module is empty, it does not contain any user program | 1x⇓ |
| IO:CODE | HHHH<br>HHHH | No normal operation<br>F 5220 / F 6220 / F 6221<br>Displayed: 8-digit hex code | 2x⇐<br>1x⇓ |
| IO:LINE | HHHH<br>HHHH | No normal operation<br>F 5220 / F 6220 / F 6221<br>Displayed: 8-digit hex code | 2x⇐<br>2x⇓ |
| IO:EXCP-POS | 1403 | Communication problem<br>F 5220 / F 6220 / F 6221 | 2x⇐<br>2x⇓ |
| IO:EXCEPTIO<br>N | HHHHHH<br>HH | As before<br>Displayed: 8-digit hex code | 2x⇐<br>4x⇓ |
| K-IS | 0120 | Diagnostic codes for additional tests performed by the manufacturer | 2x⇒ |
| K-SO | 0034 | | 3x⇒ |
| KEY | 0022 | | 4x⇒ |
| Configuration | HIMA | Configuration name | 1x⇓, 2x⇐ |
| MAX170-ERR | 0013 | Diagnostic code for additional tests performed by the manufacturer | 5x⇒ |
| MONO | ---- | 1-channel operation with redundant CU | |
| RELOAD | ---- | Mono reload is being performed | |

| Indicators | | Explanation | To call the info |
|---|---|---|---|
| Text | Example | | |
| OSLD<br>*nxy 1) | *1A8 | OS loader is started<br>Text displayed during download<br>n = 0 or 1, x = 0…F, y = 5, 6, 7, 8 | |
| Program | PRO1 | Program name | 1x⇓, 1x⇐ |
| Resource | PES 4 | Resource name | 1x⇓ |
| RUN | ---- | PES in normal operation | |
| RUN-VERSION | 3402 | RUN version number, created during operation, depending on all values | 1x⇓, 2x⇒ |
| SC1…SC 64 | 0012 | Safe communication to the 1st partner (up to 64th partner).<br>No change in value: No data | 2x⇓,2…65x⇒,<br>2x⇓,2…65x⇒,<br>1x⇓ |
| SIO1 | 0012 | SS 1 on CU no change in value:<br>No communication via SS 1 | 2x⇓<br>3x⇓ |
| SIO2 | 0012 | SS 2 on CU no change in value:<br>No communication via SS 2 | 2x⇓, 1x⇒,<br>2x⇓, 1x⇒, 1x⇓ |
| STOP | | Stop performed by the PADT or by the operating system | |
| TIME | 1431<br>3132<br>32.3 | Time expressed in hours/minutes<br>Time expressed in minutes/seconds<br>Time expressed in seconds/deciseconds | 1x⇓, 5x⇒<br>1x⇓, 6x⇒<br>1x, 7x⇒ |
| --> | --> | User program deletion<br>Simultaneous engagement of ⇒ + ⇓ + ACK,<br>Use ACK to acknowledge the error stop | 8x⇓, 1x⇒<br>⇒ + ⇓+ACK<br>ACK |
| ZB:EXCP-POS | 1403 | Communication problem<br>F 5220 / F 6220 / F 6221<br>Central module view | 2x⇐<br>5x⇓ |
| ZB:EXCEPTIO N | HHHH HHHH | As before<br>Displayed: 8-digit hex code | 2x⇐<br>6x⇓ |
| ZBS1<br>ZBS2 | 5711<br>5711 | Baud rate, parity, stop bits displayed for central module, serial connectors 1 and 2. Refer to Table 43 for details. | 1x⇓, 3x⇒, 4x⇓<br>1x⇓, 3x⇒, 5x⇓ |
| 1) * is generated by fast rotating lines ( \| / – \ ) and is therefore a sign for a properly operating OS loader. If it stops, only one of the lines is displayed. | | | |

Table 42: Information Accessible in RUN

The values entered are fictitious.

If the Value column contains ---- , only the content of the Text column is displayed. If the Value column contains a number, the text and the value are displayed alternating during the PES operation. If the text has more letters than the 4 visible, it is displayed as a ticker.

The “wandering” dot in the display is used as an operating system's sign of life.

Baud rate, parity and number of stop bits displayed for serial connectors are indicated as *xxyz*. The two positions *xx* specify the baud rate, *y* the parity and *z* the number of stop bits.

| Field | Value | Description |
|---|---|---|
| xx | 01 | 1200 baud |
| | 02 | 2400 baud |
| | 03 | 300 baud |
| | 04 | 4800 baud |
| | 06 | 600 baud |
| | 09 | 9600 baud |
| | 19 | 19 200 baud |
| | 57 | 57 600 baud |
| y | 0 | No parity |
| | 1 | Even parity |
| | 2 | Odd parity |
| Z | 1 | 1 stop bit |
| | 2 | 2 stop bits |

Table 43: Baud Rate, Parity and Number of Stop Bits

Other key sequences can be used to call additional information details on the diagnostic panel. These details, not described in this manual, only have HIMA-internal diagnostic purposes and must be evaluated by HIMA personnel.

## 12.2 Faults in the Central Area (CPU LED is lit)

| Indicators | Explanation |
|---|---|
| DEAD<br>EXCP<br>NMIL | Fatal fault during start up: Send the module to HIMA.<br>Only switching off/on is possible.<br>If no communication: replace the module. |
| RAMT<br>CHCK<br>WAIT | Text displayed after powering up until the I/O level is connected. |
| STOP | Error stop<br>In case of error stop, the last fault can be called by using the ⇒ key on the central module once. |

Table 44: Faults in the Central Area (CPU LED is lit)

## 12.3 Faults in the I/O Area (IO LED is lit)

| Indicators | Explanation |
|---|---|
| 1204 | Position of a faulty I/O module<br>1: I/O bus number<br>2: I/O subrack number<br>04: Position on the I/O subrack |
| 1314/2,4 | Channel fault of I/O modules with line monitoring (first check the field wiring and field devices)<br>1: I/O bus number<br>3: I/O subrack number<br>14: Position on the I/O subrack<br>2,4: Number of the faulty channels |
| 14,*,* | Fault of more than 4 testable I/O modules or of the complete I/O subrack<br>1: I/O bus number<br>4: I/O subrack number<br>Impossible to address the entire I/O subrack (connecting cable, I/O bus, power supply, connection module) |

Table 45: Faults in the I/O Area (IO LED is lit)

If multiple I/O modules are faulty, all affected I/O positions, including the I/O channels, are displayed alternating. After replacement of the faulty I/O module or repair of the line fault, the ACK key on the central module can be used to reset the fault displayed and the channel is active again.

Additional information can be selected using the 2 keys, even if the *IO* LED is lit. The I/O positions appear again if no new information is requested within 20 s.

## 12.4 List of Error Codes

The following list contains all messages and error codes. The error codes important for the operator are explained in more detail. They are displayed in addition to the diagnostic texts previously described, if selected using the 2 keys located on the central module front side.

The remaining error codes and diagnostic codes are only of interest if a more specific examination performed by the manufacturer is required. If a fault occurs, its error code is saved. As soon as the next fault occurs, this error code is overwritten by a new error code such that only the latest fault is stored. Previous error codes can be accessed with ELOP II.

The error code is only deleted if a new project is loaded into the central module, or if the central module is deleted and loaded again.

| Number, error code | Explanation, cause of the error code |
|---|---|
| 0 | No error. |
| 1-4 | Error in central module. |
| 5 | Cycle time exceeded. |
| 6-12 | Error in central module. |
| 13 | Outputs are not de-energized when starting up the controller, e.g., input module is inserted into a slot defined for an output module. |
| 14 | Logic emergency off. |
| 15- | Error due to difference in memories which cannot be located. |
| 16 | Error in central module. |
| 17 | Difference in memories which cannot be located (diagnostics only). |
| 18 | Tolerable time basis deviation. |
| 19 | Error in central module. |
| 20-21 | Time delay of the other central module. |
| 22 | Loss of redundancy. |
| 23 | Redundancy recovered (again). |
| 24 | DIP switches S1…S8 in redundant partners are different when redundant operation stars. |
| 25 | DIP switches S1…S8 modified during operation (since the last power on). |
| 26-28 | Error in central module. |
| 29, 30 | Agreed I/O subrack does not exist, or faul in connection module. |
| 31-46 | Fault in central module. |
| 47 | Fault in power supply monitoring. |
| 48-52 | Fault in central module. |
| 53 | Unknown I/O module type (wrong entry in ELOP II). |
| 54-87 | Fault in central module. |
| 88 | The existing central module is not of type S (safety-related), but this is required by the user program. |
| 89-91 | Fault in central module. |
| 92 | Reload aborted due to violated HIPRO-S or I/O data structure. |
| 93-94 | Signature errors in the user program. |
| 95-99 | Fault in central module. |
| 100 | Complete initialization after memory error (including fault history reset). |
| 101 | Communication to other central module is not possible, reload was aborted or operating system and/or user program versions are different. |
| 102 | Time delay received from other central module: The waiting time for communication between central modules has expired. |
| 103-126 | Fault in central module. |
| 127 | Program run monitoring for HIMA function blocks. |

| Number, error code | Explanation, cause of the error code |
|---|---|
| 128 | Fault in central module. |
| 129 | Mono reload not performed due to wrong resource type. |
| 130 | Fault in central module. |
| 131 | Start-up triggered by PADT (programming and debugging tool). |
| 132 | Start-up after pressing the ACK key on the central module. |
| 133 | Start-up after self-test. As of revision (08.17): Fault in central module. |
| 134 | Start-up after power-up. |
| 135 | Fault in power supply unit. |
| 136, 137 | Fault in central module. |
| 138 | Time for mono reload expired. |
| 139 | After mono reload, differing memory area versions or the replaced new user program is inconsistent. |
| 140 | Fault in central module. |
| 141 | User program flash EPROM is being deleted. Preparing new user program load. |
| 142 | Start triggered by the PADT after download. |
| 143 | Central module that was loaded first, starts normal operation with the modified program (in case of reload with redundant central modules). |
| 144 | Normal operation restarted after mono reload. |
| 145 | Reload is starting for the 2nd central module. |
| 146 | Hot start triggered by the programming and debugging tool (PADT). |
| 147 | The user program requires extended memory, but the existing central module is not of type E (extended memory). |
| 148 | The hardware configuration does not match the resource type. |
| 149 | Start-up after self education. |
| 150 | Self-education aborted. |
| 151 | Cycle time exceeded, the value set for the watchdog time is too low. |
| 152 | The waiting time for communication between central modules has expired and the central modules cannot communicate with one another. One of the redundant central modules is not in RUN mode and initiates its own error stop. |
| 153 | Error and abort in connection with redundant reload. |
| 152-160 | Fault in central module. |
| 161 | Operation continued after break point. |
| 162, 163 | Fault in central module. |
| 164 | User program flash memory not loaded. |
| 165-175 | Fault in central module. |
| 176-179 | Agreed I/O subrack does not exist, or fault in connection module. |
| 180, 181 | Fault in I/O power supply. |
| 182 | Fault in connection module. |
| 183, 184 | Fault in I/O power supply. |
| 185 | Fault in connection module. |
| 186 | Fault in central module. |
| 187 | Agreed I/O subrack does not exist, or fault in connection module.<br>With H41q: Wrong resource type or fault in central module. |
| 188 | Noise blanking in connection module activated. |
| 189 | Fault in central module. |
| 190 | I/O subrack is shut down. |
| 191 | Maintenance switch of F 7553 is engaged. |
| 192 | Fault in connection module. |
| 193-196 | Fault in central module. |
| 197 | Noise blanking in I/O modules activated. |

| Number, error code | Explanation, cause of the error code |
|---|---|
| 198 | Fault in central module. |
| 199 | Event buffer initialization. |
| 200 | I/O error in F 3349 module. |
| 201-208 | Faulty F 6213/14/15/16 input module. |
| 209-212 | Fault in central module. |
| 213 | Faulty F 5220, F 6220, F 6221 input module. |
| 214 | Fault in central module. |
| 215-216 | Faulty F 3235 input module |
| 217-219 | Faulty F 3237 / F 3238 input module. |
| 220-222 | Faulty F 6705 output module. |
| 223-226 | Faulty F 3330/31/33/34/35/48 or F 3430 output module. |
| 227-228 | Faulty F 6217 input module. |
| 229 | Communication problems between central module and the Ethernet communication module F 8625/27 or the PROFIBUS communication module F 8626/28 |
| 230-239 | Fault in central module. |
| 240 | No digit set at the resource name position 7 and 8 (for Ethernet communication only). |
| 241-251 | Communication problems between central module and the Ethernet communication module F 8625/27 or the PROFIBUS communication module F 8626/28. |
| 252 | Fault in central module. |
| 253 | User program deletion via keys on the central module’s front side has been prepared (the error code is only displayed temporarily). |
| 254 | User program deletion via keys on the central module’s front side has been prepared (usually only displayed in the error history). |
| 255 | Fault in central module. |

Table 46: Error Codes

# Appendix

## Glossary

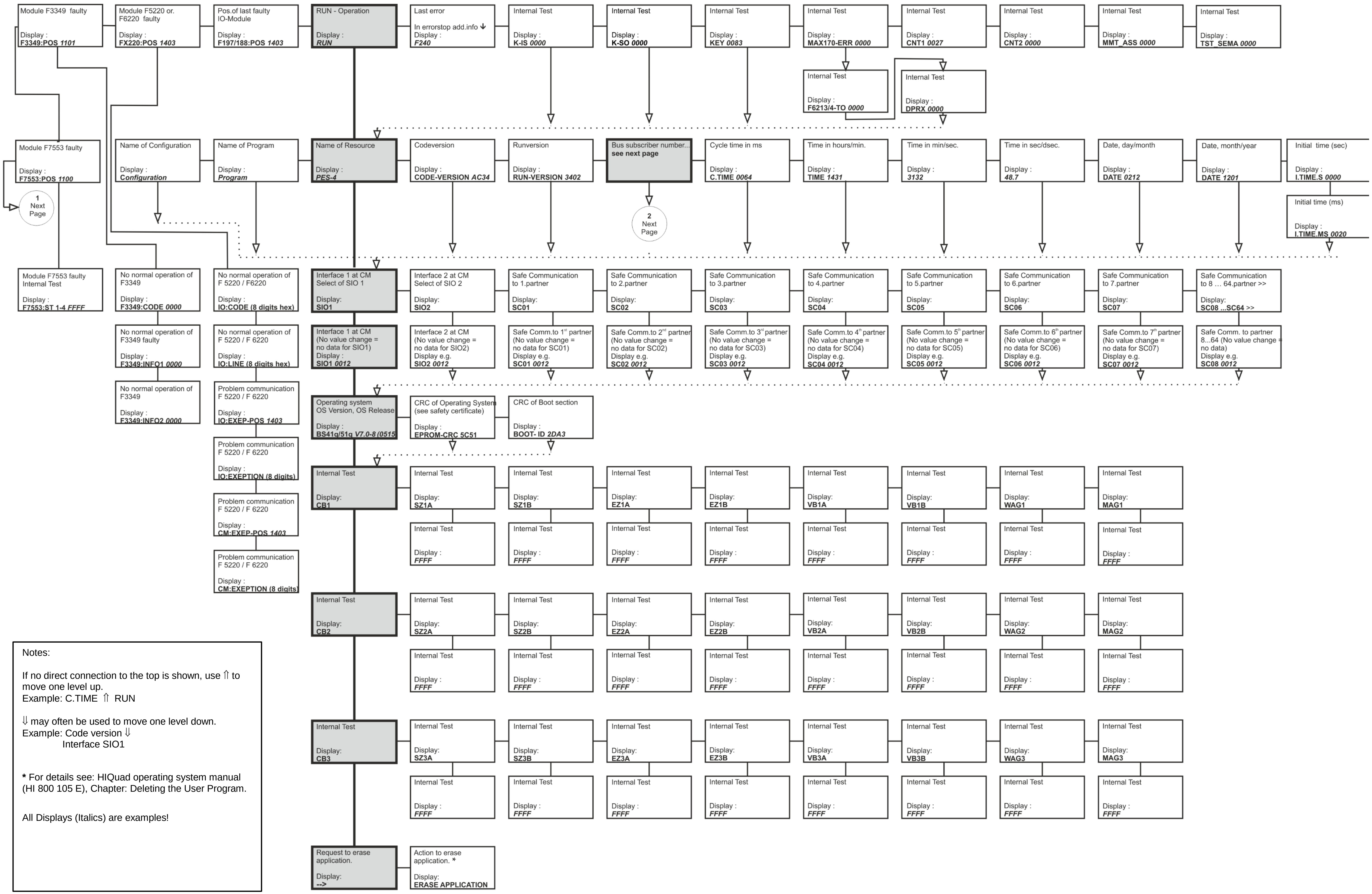
| Term | Description |
|---|---|
| AI | Analog input |
| AO | Analog output |
| ARP | Address resolution protocol, network protocol for assigning the network addresses to hardware addresses |
| DI | Digital input |
| DO | Digital output |
| EMC | Electromagnetic compatibility |
| EN | European standard |
| ESD | Electrostatic discharge |
| FB | Fieldbus |
| FBD | Function block diagrams |
| HW | Hardware |
| IEC | International electrotechnical commission |
| Interference-free | Inputs are designed for interference-free operation and can be used in circuits with safety functions |
| MAC | Media access control address, hardware address of one network connection |
| PADT | Programming and debugging tool (in accordance with IEC 61131-3), PC with SILworX |
| PELV | Protective extra low voltage |
| PES | Programmable electronic system |
| R | Read, the variable is read out |
| R/W | Read/Write, column title for system variable type |
| Rack ID | Base plate identification (number) |
| $r_P$ | Peak value of a total AC component |
| SC/OC | Short-circuit/open-circuit |
| SELV | Safety extra low voltage |
| SFF | Safe failure fraction, portion of faults that can be safely controlled |
| SIL | Safety integrity level (in accordance with IEC 61508) |
| SNTP | Simple network time protocol (RFC 1769) |
| SRS | System.Rack.Slot, addressing of a module |
| SW | Software |
| TMO | Timeout |
| W | Write, the variable receives a value, e.g., from the user program |
| WD | Watchdog, device for monitoring the system's correct operation Signal for fault-free process |
| WDT | Watchdog time |

## Index of Figures

## Index of Tables

# Index

# H41q/H51q Diagnostic Panel BS41q/51q V7.0-8 (09.12)



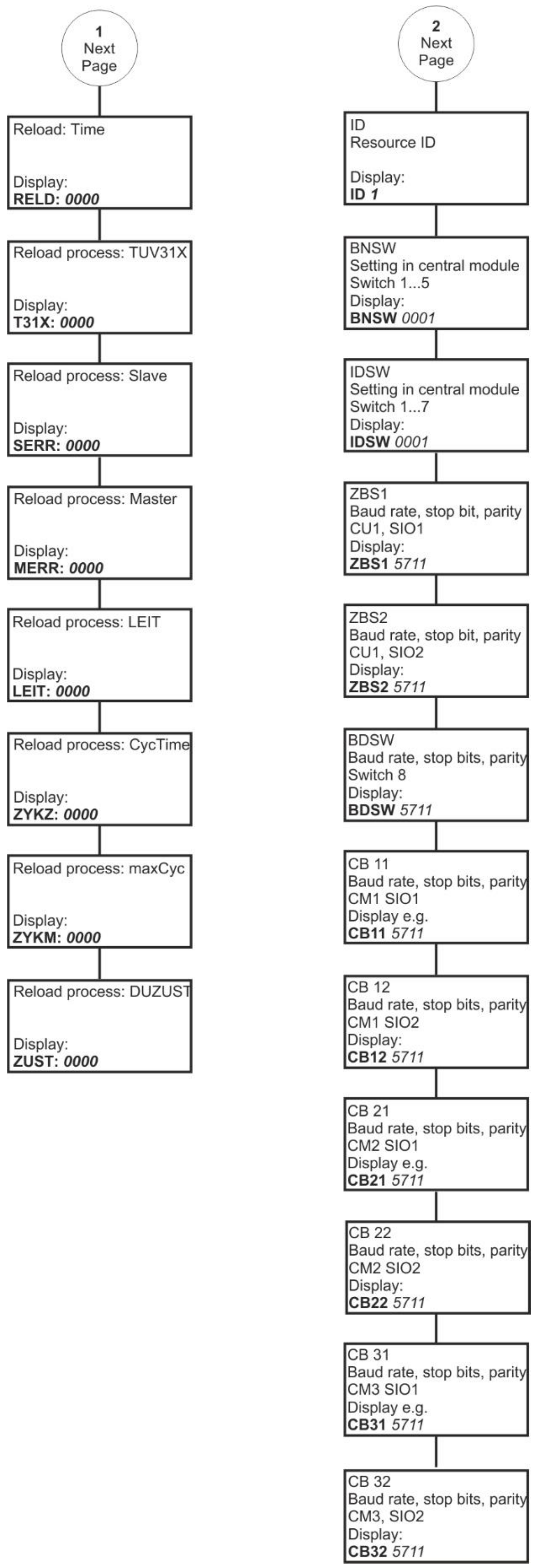

Notes:

If no direct connection to the top is shown, use ⇑ to move one level up.
Example: C.TIME ⇑ RUN

⇓ may often be used to move one level down.
Example: Code version ⇓
Interface SIO1

***** For details see: HIQuad operating system manual (HI 800 105 E), Chapter: Deleting the User Program.

All Displays (Italics) are examples!


1
Next
Page
Reload: Time
Display:
RELD: 0000
Reload process: TUV31X
Display:
T31X: 0000
Reload process: Slave
Display:
SERR: 0000
Reload process: Master
Display:
MERR: 0000
Reload process: LEIT
Display:
LEIT: 0000
Reload process: CycTime
Display:
ZYKZ: 0000
Reload process: maxCyc
Display:
ZYKM: 0000
Reload process: DUZUST
Display:
ZUST: 0000
2
Next
Page
ID
Resource ID
Display:
ID 1
BNSW
Setting in central module
Switch 1...5
Display:
BNSW 0001
IDSW
Setting in central module
Switch 1...7
Display:
IDSW 0001
ZBS1
Baud rate, stop bit, parity
CU1, SIO1
Display:
ZBS1 5711
ZBS2
Baud rate, stop bit, parity
CU1, SIO2
Display:
ZBS2 5711
BDSW
Baud rate, stop bits, parity
Switch 8
Display:
BDSW 5711
CB 11
Baud rate, stop bits, parity
CM1 SIO1
Display e.g.
CB11 5711
CB 12
Baud rate, stop bits, parity
CM1 SIO2
Display:
CB12 5711
CB 21
Baud rate, stop bits, parity
CM2 SIO1
Display e.g.
CB21 5711
CB 22
Baud rate, stop bits, parity
CM2 SIO2
Display:
CB22 5711
CB 31
Baud rate, stop bits, parity
CM3 SIO1
Display e.g.
CB31 5711
CB 32
Baud rate, stop bits, parity
CM3, SIO2
Display:
CB32 5711

MANUAL

**Operating System Functions**

**HI 800 105 E**

For further information, please contact:

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

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

Learn more about HIMA solutions online:

www.hima.com/en/

HIMA SMART SAFETY.

www.hima.com