PC-based Systems
ELOP II-NT

Resource type

HIMA Paul Hildebrandt GmbH + Co KG
Industrieautomatisierung

Contents

1 Creating a configuration 1

1.1 Properties; Buses.. 1

1.1.1 Editing HIBUS..2

1.1.2 PES Masters 7

2 Configuring the resource.................................9

2.1 Instancing a program type to the resource............. 11

2.2 Configuring a resource.. 12

2.2.1 Editing Cabinet layout..13

2.2.2 Resource properties 19

3 Resource-related functions in the Editor 29

3.1 Assigning system variables..................................... 30

3.2 Creating Events 31

3.2.1 Variables for transmission with HIPRO...........................31

3.2.2 Variables for transmission to external systems.............33

4 Communication between HIMA PESs............37

4.1 Protocols... 37

4.1.1 Safety-related transfer: HIPRO-S......................................37

4.1.2 Non-safety related transfer: HIPRO-N..............................37

4.2 HIBUS Hardware... 37

4.3 Redundancy... 38

4.4 Configuring data interchange 38

4.4.1 Safety-related data transfer: HIPRO-S..............................38

4.4.2 Non-safety related data transfer: HIPRO-N......................39

4.4.3 Variables declaration 39

4.5 Configuring data interchange to ELOP slaves........ 40

4.5.1 Configuration in ELOP...41

4.5.2 Bus configuration 41

4.5.3 Resource configuration...42

4.5.4 Variables declaration 42

5 Communicating with external systems 43

5.1 BUSCOM Communication 43

5.2 3964R Communication.. 45

6 Generating the program code (compiling) 49

6.1 Code generator 49

6.2 Compiling and loading PES masters....................... 49

6.3 Loading the program into the controller.................. 50

6.4 Download/Reload 52

6.4.1 OS Download 53

7 Online Test..53

7.1 Starting the online test... 53

7.2 Explanation of the OLT window 54

7.2.1 Operating buttons 55

7.2.2 Program structure tree 56

7.3 Functions in the ONLINE test................................... 57

7.3.1 Meanings..57

7.3.2 Forcing variables 58

8 Documentation in the configuration 63

8.1 Documentation functions of the configuration 63

8.1.1 Bus docu..64

8.1.2 PES Master Docu 65

8.1.3 CFG Docu...66

8.2 Documentation functions of the resource 66

8.2.1 CRF Docu...67

8.2.2 RES Docu...68

8.2.3 RES Docu (generated) 69

8.2.4 RES Docu (loaded)..70

9 Safety related functions 71

9.1 Building an application... 72

9.1.1 Create the logic diagrams 72

9.1.2 Code generation..72

9.1.3 Download 72

9.1.4 Function check of the PES 72

9.2 Modification of an application 73

9.2.1 Modification of logic diagrams 73

9.2.2 Code generation 73

9.2.3 Download 74

9.2.4 Function check of the PES 74

10 HIMA Standard Blocks 75

11 H8-STA-3 77

11.1 Function information 78

11.2 General block information 78

11.3 GENERAL 78

11.3.1 Block assignment 78

11.4 Notes on assigning inputs/outputs 79

11.4.1 Bus-No. Rack Pos. 79

11.4.2 Bus-No. Rack Pos. Group Amplifier 79

11.4.3 Bus-No. Rack Pos. red. group amp. 79

11.4.4 Control Group Relay 79

11.4.5 Cont. red. Group Rel. 79

12 H8-UHR-3 81

12.1 Function information 82

12.2 General block information 82

12.3 GENERAL 82

12.4 Notes on the assignment of inputs/outputs 83

12.4.1 Year...83

12.4.2 Month 83

12.4.3 Day..83

12.4.4 Enter date...83

12.4.5 Hour..83

12.4.6 Minute...83

12.4.7 Second 83

12.4.8 Enter time...83

12.4.9 Minute pulse 84

12.4.10 Hour pulse 84

12.4.11 Day pulse 84

12.4.12 Year...84

12.4.13 Month 84

12.4.14 Day...84

12.4.15 Hour...84

12.4.16 Minute..84

12.4.17 Second 84

12.4.18 Millisecond...85

12.4.19 Minute pulse 85

12.4.20 Hour pulse 85

12.4.21 Day pulse 85

13 HA-LIN-3 87

13.1 Function information 88

13.2 General block information...................................... 88

13.3 GENERAL.. 88

13.4 Notes on the assignment of inputs/outputs 89

13.4.1 Measurement 1...4 89

13.4.2 Faulty measurement 1...4 89

13.4.3 Lower range limit in deci ×C 89

13.4.4 Sign start/type of sensor 89

13.4.5 Range in deci ×C 90

13.4.6 Lower range limit digital (bool)..90

13.4.7 Upper range limit digital (bool) 91

13.4.8 Lowest permitted analog value (underflow) 91

13.4.9 Highest permitted analog value (overflow).................................91

13.4.10 Ref. temp./Line resistance...91

13.4.11 Measurement 1...4 (0...1000)...92

13.4.12 Measurement 1...4 in deci ×C..92

13.4.13 Sign measurement 1...4 92

13.4.14 Faulty measurement 1...4 92

13.5 Programming notes and example............................ 93

13.5.1 Programming with negative input values 93

13.5.2 Assignment of the block when using PT100 sensors with F 6215 94

14 HA-PID-3 95

14.1 Function information 96

14.2 General block information....................................... 96

14.3 GENERAL... 97

14.4 Notes on the assignment of inputs/outputs 97

14.4.1 Actual-value controlled variable.......................................97

14.4.2 Setpoint controlled variable..97

14.4.3 PID structure (PID=111;PD=101) 98

14.4.4 P gain in %..98

14.4.5 I reset time in ds (TN)..98

14.4.6 D rate time in ds (TV) 98

14.4.7 Scan time in ds (TA)...99

14.4.8 Enter k, TN, TV, TA 99

14.4.9 Processing pulse 99

14.4.10 Correcting parameter (0,1,2,3) 99

14.4.11 Manual setpoint correcting variable..............................100

14.4.12 TRUE=Manual (PID=0) 100

14.4.13 TRUE Manual compensating..100

14.4.14 Min. correcting deviation...100

14.4.15 Max. correcting variable 100

14.4.16 Max. correcting deviation...101

14.4.17 Correcting variable 101

14.4.18 Position display..101

14.4.19 Correcting deviation sign...101

14.4.20 Correcting deviation value 101

14.4.21 System deviation sign 101

14.4.22 System deviation value 101

14.4.23 P component sign 101

14.4.24 P component value 102

14.4.25 I component sign 102

14.4.26 I component value..102

14.4.27 D component sign..102

14.4.28 D component value 102

14.4.29 Max. correcting variable achieved..................................102

14.4.30 Max. correcting deviation achieved 102

15 HA-PMU-3 103

15.1 Function information 103

15.2 General block information...................................... 104

15.3 GENERAL.. 104

15.4 Notes on the assignment of inputs/outputs 106

15.4.1 Input value 1...8 106

15.4.2 1=4...20mA; 0=0...20mA 106

15.4.3 1=analog values; 0=per mil value...................................106

15.4.4 8=8 bit; 12=12 bit analog value 106

15.4.5 Damping (0=none; 1=damping) 107

15.4.6 Output values 1...8 107

15.4.7 Error 1...8 107

16 HA-RTE-3...109

16.1 Function information 110

16.2 General block information...................................... 111

16.3 GENERAL.. 111

16.4 Notes on the assignment of inputs/outputs 111

16.4.1 Bus No. Rack Pos. (e.g. 1305) 111

16.4.2 Bus No. BT Pos. red. module 112

16.4.3 Assigned channels 112

16.4.4 Permitted difference red. values in per mil 112

16.4.5 0=none; 1=attenuation 112

16.4.6 Permitted time difference red. input in ds 112

16.4.7 Module test switch 113

16.4.8 Test switch red. module 113

16.4.9 Maximum test time in min 113

16.4.10 Test 4 channels with 1 switch (1,0) 113

16.4.11 0=available, 1=safe reaction 113

16.4.12 Output in event of error 114

16.4.13 1=4...20 mA, 0=0...20 mA 115

16.4.14 0=conversion to per mil, 1=no conversion 115

16.4.15 Max. underflow in per mil (1000 = 20 mA) 115

16.4.16 Value 1...4 115

16.4.17 Error value 1...4 116

16.4.18 Change error code pulse 116

16.4.19 Error 116

16.4.20 Module error code 117

16.4.21 Error code red. module 117

17 HB-BLD-3 119

17.1 Function information 119

17.2 General block information 120

17.3 GENERAL 120

17.4 Notes on the assignment of inputs/outputs 120

17.4.1 Bus No. Rack Pos. (e.g.1305) 120

17.4.2 Mode channel 1...8 (0/1/2) 121

17.4.3 .... Max. time inrush current in ms121

17.4.4 .... Error channel 1...8122

17.4.5 Pulse on error 122

17.4.6 Error 122

17.4.7 .... Error code122

18 HB-BLD-4 125

18.1 Function information 126

18.2 General block information 126

18.3 GENERAL 126

18.4 Notes on the assignment of inputs/outputs 127

18.4.1 Bus No. Rack Pos. (e.g. 1305) 127

18.4.2 .... Bus No. Rack Pos. red. module127

18.4.3 Mode channel 1...8 (0/1/2) 127

18.4.4 Max. time inrush current in ms, module ; red. module.128

18.4.5 Error channel 1...8 128

18.4.6 Pulse on error 129

18.4.7 Error 129

18.4.8 Error code module or red. module 129

19 HB-RTE-3 131

19.1 Function information 132

19.2 General block information 132

19.3 GENERAL 132

19.4 Notes on the assignment of inputs/outputs 133

19.4.1 Bus No. Rack Pos. (e.g. 1305) 133

19.4.2 Bus No. BT Pos. red. module 133

19.4.3 Assigned channels 133

19.4.4 1=1of2; 2=2of2 shutdown 134

19.4.5 Permitted error time in min 134

19.4.6 Permitted time difference red. inputs in ds 134

19.4.7 Output 1...8 135

19.4.8 Channel error mask 135

19.4.9 Error code change, pulse 135

19.4.10 Error 135

19.4.11 Error code module 136

19.4.12 Error code red. module 136

20 HK-AGM-3 137

20.1 Function information 138

20.2 General block information 138

20.3 GENERAL 138

20.4 Notes on assigning the inputs/outputs 139

20.4.1 Central module (1,2)..139

20.4.2 Coprocessor module (1, 2, 3)..139

20.4.3 0=no func. 1=DC 2=safety-rel. 3=DC+safety-rel. 139

20.4.4 Current. funct. copro. mod. 139

20.4.5 Error in coprocessor module..139

20.4.6 Function counter..139

20.4.7 Bus cycle time in ms..140

20.4.8 Error mask bus 1, BTN 1 - 16 140

20.4.9 Bus 1, BTN 17 - 31..140

20.4.10 Error mask red. bus, BTN 1 - 16.....................................140

20.4.11 red. Bus, BTN 17 - 31 140

21 HK-LGP-3..141

21.1 Function information 142

21.2 General block information....................................... 142

21.3 GENERAL 142

21.4 Notes on the assignment of inputs/outputs 143

21.4.1 Stop event printout 143

21.4.2 FALSE=LGP, TRUE=MODBUS..143

21.4.3 Output emergency message texts (1,0) 143

21.4.4 Response to day change (0,1,2) 143

21.4.5 Printer error 144

21.4.6 Printer offline...144

21.4.7 Printer buffer overflow..144

21.4.8 No paper...144

21.4.9 Printer cover open 144

21.4.10 Printer conn. lost...144

21.4.11 Event in buffer 144

21.4.12 Buffer overflow...145

21.4.13 Printout active 145

21.4.14 Message counter...145

21.5 Program example 146

22 HK-MMT-3 149

22.1 Function information 150

22.2 General block information..................................... 150

22.3 GENERAL.. 151

22.3.1 Potential applications 151

22.4 Notes on the assignment of inputs/outputs 153

22.4.1 Central module interface (1,2)..153

22.4.2 Telephone number 153

22.4.3 Type of call establishment (0,1,2) 153

22.4.4 Enable, TRUE...154

22.4.5 Slave number (target bus stn.).......................................154

22.4.6 Modbus function code..154

22.4.7 Relative address 1st var. master 154

22.4.8 No. 1st event in master...155

22.4.9 Number var. (BOOL)..155

22.4.10 Relative address 1st var. slave 155

22.4.11 Modbus function code..155

22.4.12 Relative address 1st var. master 155

22.4.13 Number var. (UINT)..156

22.4.14 Relative address 1st var. slave 156

22.4.15 Max. slave answer time in ms 156

22.4.16 Establishment delay in s 157

22.4.17 Modem error code...157

22.4.18 Error code transmission..157

22.4.19 Modbus error code transmission 157

22.4.20 Error code transmission..158

22.4.21 Modbus error code transmission 158

22.4.22 No connection 158

22.4.23 Task active 158

22.4.24 Task counter 158

22.5 Applications.. 159

22.5.1 Input of the telephone number 159

22.5.2 Entering events 160

22.5.3 Time consideration 163

23 HZ-DOS-3 167

23.1 Function information 167

23.2 General block information..................................... 168

23.3 GENERAL... 168

23.4 Notes on the assignment of inputs/outputs 169

23.4.1 Bus No. Rack Pos. (e.g. 1306)..169

23.4.2 Error code 169

24 HZ-FAN-3 171

24.1 Function information 171

24.2 General block information..................................... 172

24.3 GENERAL... 172

24.4 Notes on the assignment of inputs/outputs 172

24.4.1 Bus No. Rack Pos. (e.g. 1306)..172

24.4.2 Error code 172

24.4.3 Error 173

24.5 Applications 174

25 Resource Types 175

25.1 Current Resource Types 175

25.2 Predecessor Resource Types 176

26 Subracks...176

26.1 Current subracks.. 177

26.2 Old subracks... 177

27 Modules 178

27.1 Current modules... 178

27.2 Old modules.. 179

28 Supported IEC Functions and Data Types ..180

28.1 Supported data types.. 180

28.2 Supported functions and function blocks 181

29 System Variables 184

29.1 READ System variables of the BOOL type 185

29.1.1 I/O Error...185

29.1.2 HIBUS.Resource-Name.error 185

29.1.3 SYSTEM.Force individual switch outputs.....................185

29.1.4 SYSTEM.Force individual switch inputs.......................186

29.1.5 SYSTEM.Force main switch outputs.............................186

29.1.6 SYSTEM.Force main switch inputs 186

29.1.7 SYSTEM.Prohibited access...186

29.1.8 SYSTEM.Single-channel..186

29.1.9 SYSTEM.normal 186

29.2 WRITE System variables of the BOOL type.......... 186

29.2.1 I/O Error acknowledgment 186

29.2.2 SYSTEM.Logic emergency stop 186

29.3 READ System variables of the UINT/WORD type.. 187

29.3.1 HIBUS.Resource-Name.Receive counter......................187

29.3.2 SIO Receive Counter..187

29.3.3 I/O.Error code 2nd I/O bus...188

29.3.4 I/O.Error position 2nd I/O bus 189

29.3.5 I/O.Error code 1st I/O bus 189

29.3.6 I/O.Error position 1st I/O bus 190

29.3.7 SYSTEM.RAM/EPROM 190

29.3.8 SYSTEM.Runversion..190

29.3.9 SYSTEM.Codeversion..191

29.3.10 SYSTEM.Number of prohibited accesses......................191

29.3.11 SYSTEM.Errormask2 191

29.3.12 SYSTEM.Errormask1 192

29.3.13 SYSTEM.Errorcode 192

Configuration

1 Creating a configuration

Configurations are the highest-level structuring tools, and group together the PESs (Programmable Electronic Systems) of a complete system or subsystem. For each of these PESs they contain a resource that complies with IEC 61131-3. A configuration can be visualized as a bus system to which the PESs are connected. Configurations contain:

- Resources
- Global variables
- Libraries
- Structuring folders
- Connections

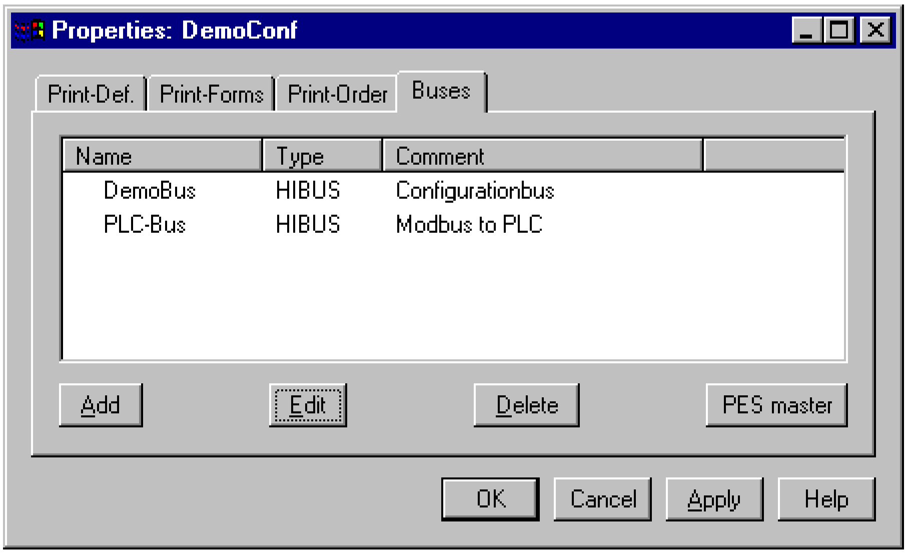
1.1 Properties; Buses

Once you have defined the resources of your configuration you can identify the main buses and their communication parameters. You can then access the bus parameters when configuring the serial interfaces of the resources.

Note:

You will need at least one bus for communicating between the PADT (Programming and Debugging Tool = programming unit) and the resources for loading and for online operations!

To define a bus, click on ´Properties´ in the context menu of the configuration.



Name	Type	Comment
DemoBus	HIBUS	Configurationbus
PLC-Bus	HIBUS	Modbus to PLC

Configuration properties window, ´Buses´ tab

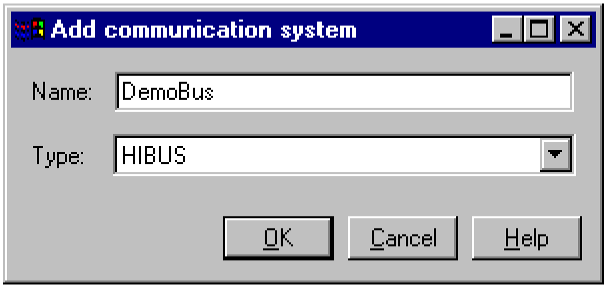
Previously defined buses are listed in this dialogue box. Columns in the index display the main bus data in table form:

- Name: Name of the bus
- Type: Type of communication system (currently HIBUS only)
- Commentary: any comments on the bus

Command buttons

You can add new buses, edit and delete existing buses and open a dialogue box to compile and download the PES masters of all buses.

- ´Add´: Add a communication system



Name: DemoBus

Type: HIBUS

Dialogue box ´Add communication system´

Give the bus a unique name and select the type of communication system (only the ´HIPRO´ type is currently available). Confirm your choice with ´OK´ to open a dialogue box as described below under ´Edit´.

- Edit: Edit a bus configuration
  Mark the bus you want to edit and click on ´Edit´. You will see a dialogue box in which you can enter comments about the bus, define the bus stations and set up the bus communication parameters (See “Editing HIBUS” on page 2.).
- Delete: Delete a bus configuration
  Mark the bus you want to delete and click on ´Delete´. The bus will be deleted without a confirm prompt.
- PES Masters: Compile and download PES masters
  you will see a dialogue box listing all the defined PES masters for compiling and downloading (See “PES Masters” on page 7.).

1.1.1 Editing HIBUS

HIBUS is used to transmit data between individual HIMA PESs and for communication between the PESs and the PADT (Programming and Debugging Tool: programming unit in IEC, the PC with ELOP II-NT) or with external systems.

The protocols used for communication between individual PESs are based on the master/slave principle, and several masters can be connected to a bus as well as several slaves. If more than one master is connected to a bus then they share control over the bus cyclically. Each station is identified by a user-defined bus station number (BSN). Up to 31 logic bus stations can work on a bus, whether as slaves or as masters.

Note:

The redundant central modules of a PES are addressed with a bus station number (BSN).

The following can be used as master systems:

- Programming units (PADT)

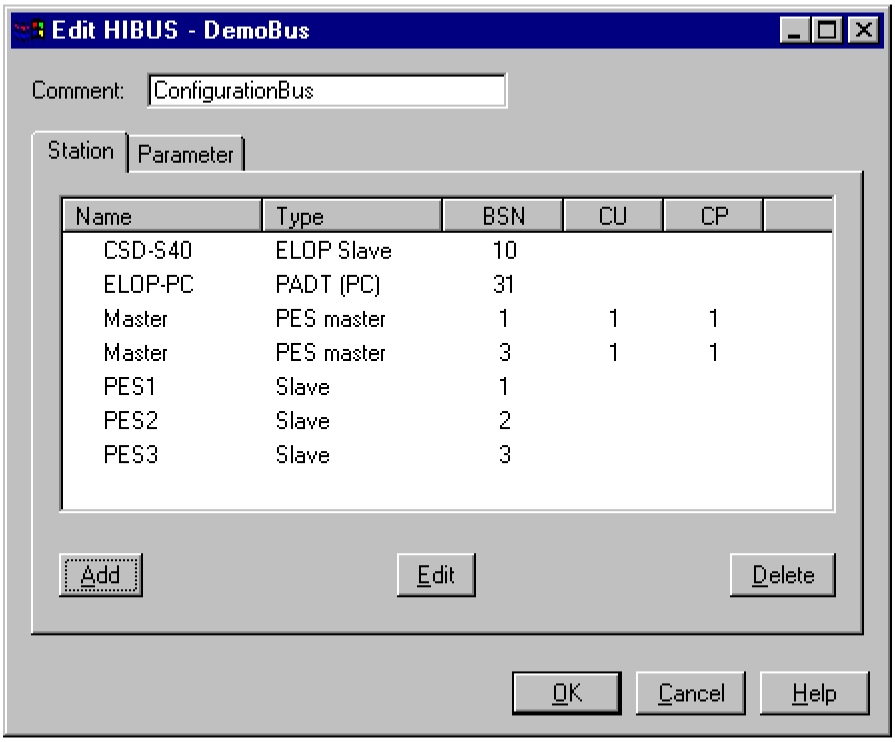
- Coprocessor modules (CM) in the H51 or H51q systems as PES masters

The following can be used as slave systems:

- HIMA-PES type A1, H11, H41 and H51 programmed with ELOP
- HIMA-PES type A1, H11, H41, H51, also H41q and H51q programmed with ELOP II-NT

When you add or edit a bus you will see a dialogue box in which you can enter a description of the HIBUS, the stations and the bus communication parameters. The dialogue box offers ´Stations´ and ´Parameters´ tabs for this purpose.

Stations tab



Comment: ConfigurationBus

Name	Type	BSN	CU	CP
CSD-S40	ELOP Slave	10		
ELOP-PC	PADT (PC)	31		
Master	PES master	1	1	1
Master	PES master	3	1	1
PES1	Slave	1		
PES2	Slave	2		
PES3	Slave	3		

Dialogue box ´Edit HIBUS´, ´Stations´ tab

The ´Stations´ index lists previously defined bus stations. Columns in the index display the main bus station data in table form:

- Name: Name of the bus station
- Type: Type of bus station
- BSN: Bus station number
- CU: Assigned central module of the PES master
  1 left CU, 2 = right CU
- CM: Coprocessor module in which the PES master runs
  1.3 = first to third CM according to the slot next to the CU

Command buttons

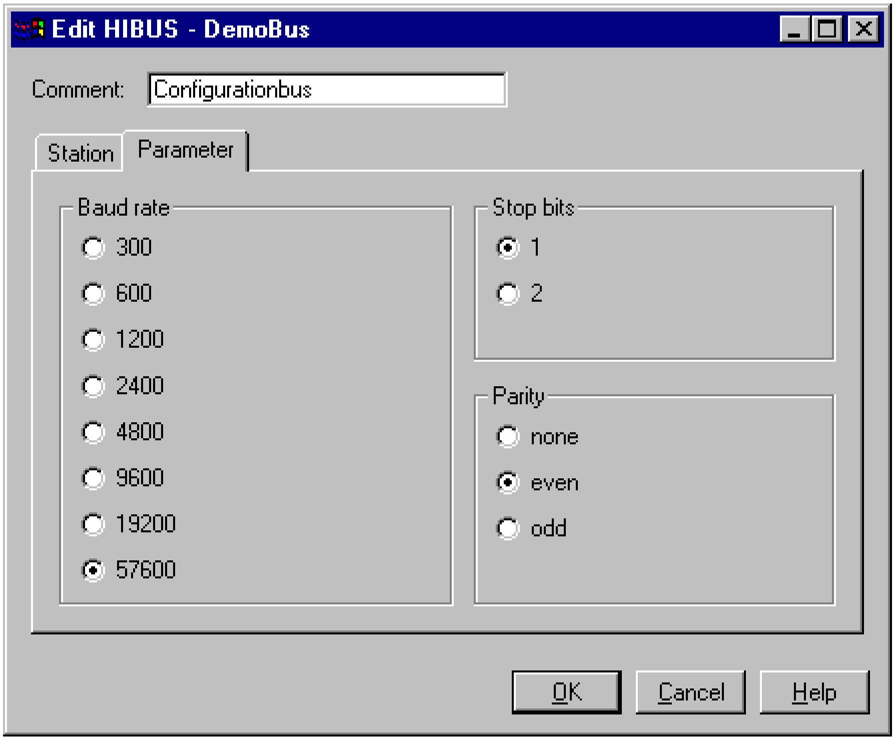
You can add new bus stations and edit or delete existing stations.

- Add: Add a bus station
  You will see a dialogue box as described under ´Edit´.
- Edit: Edit a bus station
  Mark the bus station you want to edit and click on ´Edit´. You will see a dialogue box in which you can enter the name and type of the bus station and any parameters. See “HIPRO Stations” on page 5.

- Delete: Delete a bus station
  Mark the bus station you want to delete and click on ´Delete´. The bus station will be deleted without a confirm prompt.

Parameters tab

The ´Parameters´ tab in the ´Edit HIBUS´ dialogue box is used to define the communication parameters of the bus.



Edit HIBUS - DemoBus

Comment: Configurationbus

Station | Parameter

Baud rate
○ 300
○ 600
○ 1200
○ 2400
○ 4800
○ 9600
○ 19200
● 57600

Stop bits
● 1
○ 2

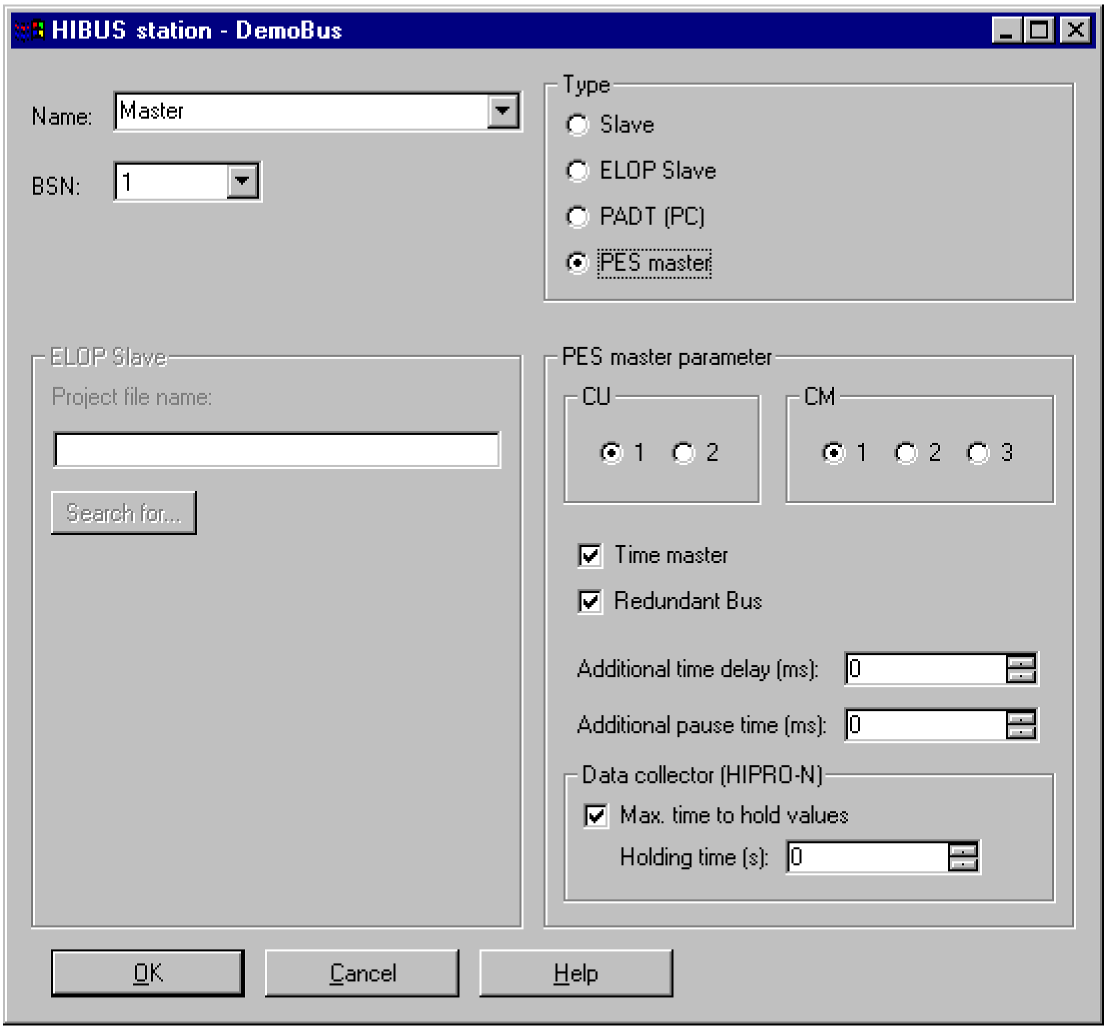
Parity
○ none
● even
○ odd

Dialogue box ´Edit HIPRO´, ´Parameters tab

You can set the baud rate, the number of stop bits and the parity of the serial bus communication here. These parameters can then be accessed when the serial interfaces of the bus stations are configured. Subsequent changes to bus parameters will be automatically entered and can be made centrally.

1.1.1.1 HIBUS stations

Clicking on the ´Add´ or ´Edit´ command button in the ´Edit HIBUS´ dialogue box opens another dialogue box in which you can enter the name, type and bus station number (BSN) of the station.



HIBUS station - DemoBus

Name: Master

BSN: 1

Type

○ Slave

○ ELOP Slave

○ PADT (PC)

● PES master

ELOP Slave

Project file name:

PES master parameter

CU: ● 1 ○ 2

CM: ● 1 ○ 2 ○ 3

☑ Time master

☑ Redundant Bus

Additional time delay (ms): 0

Additional pause time (ms): 0

Data collector (HIPRO-N)

☑ Max. time to hold values

Holding time (s): 0

´HIBUS stations´ dialogue box

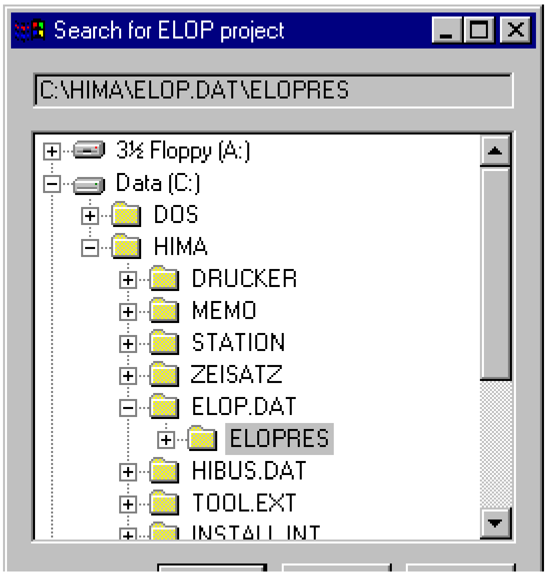
- Type: Type of the bus station
- Slave: PES programmed with ELOP II-NT
- ELOP Slave: PES programmed with the ELOP firmware (See “Configuring data interchange to ELOP slaves” on page 40.)
- PC master: PADT
- PES masters: Coprocessor modules (CM) in the H51 or H51q systems as PES masters
- Name: Name of the bus station
  The name of a previously defined resource can be selected from the list box for ´slave´ type stations. For bus stations of the ´PC master´ or ´PES master´ type you must enter the name yourself.

Note:

The name may not be more than 8 characters long and should begin with a letter. Many special characters such as ´umlauts´ cannot be mapped in the controller and are therefore replaced by question marks.

- BSN: Bus station numbers
  station number for PESs is set by hardware switches on the central module and must be the same as the number entered here. The BSN must be unique.
- ELOP Slave: Project file name of the ELOP Slave
  the path and file name of the ELOP project are entered here for

´ELOP Slave´ type bus stations. You can open a dialogue box to select the project folder by clicking on the ´Search...´ command button:



C:\HIMA\ELOP.DAT\ELOPRES

´Search for ELOP project´ dialogue box

Projects created with the ELOP firmware are usually stored in the [Drive]:\HIMA\ELOP.DAT folder.

- PES master parameters: Parameters for ´PES master´ type bus stations
- CU: Central module which the PES master is next to.
  1 left CU, 2 = right CU
- CM: Coprocessor module containing the PES masters.
  1.3 = first to third CM according to the slot next to the central module
- Time master: Activate time master function
  PES master can synchronize the time of all PESs connected to the bus by periodically transmitting the time of the central module assigned to it to all other bus stations. Here you can define whether the created PES master will be a time master
- Redundant bus:
  The PES master uses a reserve bus to handle data traffic if the normal bus fails
- Increase time delay (ms):
  For communications over modem or satellite links this setting can be increased to allow for the delay due to the carrier
- Increase time interval (ms):
  For communications over modem or satellite links this setting can be used to increase the interval between receiving the last transmission and starting the next one
- Restricted holding time of settings on call failure:
  Here you can define the response of the PES master if it cannot export data from this controller because a call to a PES has failed. This setting is only active when the HIPRO-N protocol is used (PES master as data centre).
  If this control box is highlighted, the values of the failed controller are held by the PES master for all the other PESs in the data interchange for the preset time (in seconds). When the holding time times out the values are written to FALSE or 0 if no new call to the failed controller has been made in that time.

If the control box is not highlighted then values are frozen until a new call is made.

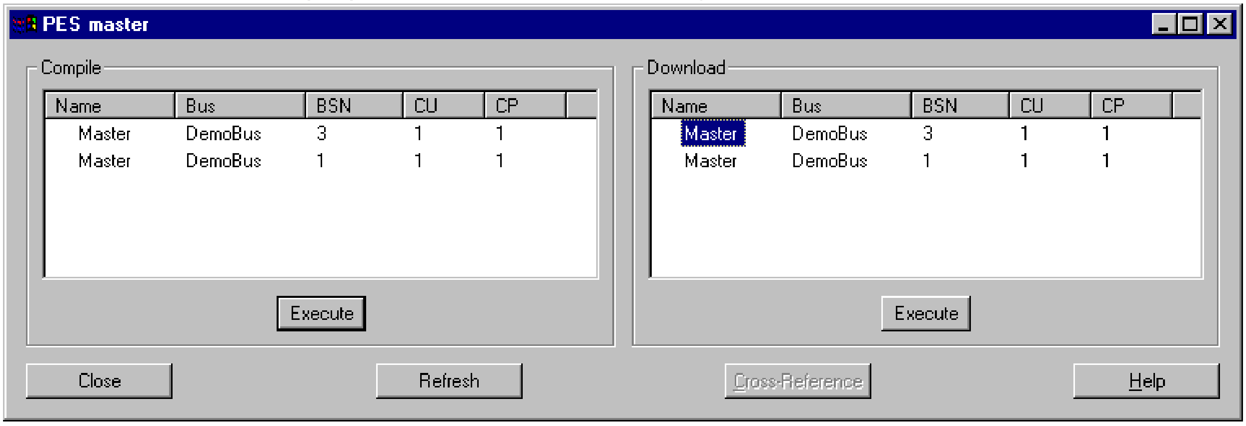
Note:

If you want the target controllers to detect when values are frozen then this must be included in the design, e.g. by transmitting a constantly changing signal.

1.1.2 PES Masters

PES masters are coprocessor modules of the H51 and H51q systems which have a program for controlling the transfer of data between the PESs (slaves). A PES master is always assigned to a PES and can service up to 31 slaves.

You can open a dialogue box for compiling the PES master programs and downloading the compiled programs to the coprocessor modules by using the ´PES masters´ option in the context menu of the configuration, or by clicking on the ´PES masters´ command button in the ´Buses´ tab in the properties window of the configuration.



Compile

Name	Bus	BSN	CU	CP
Master	DemoBus	3	1	1
Master	DemoBus	1	1	1

Download

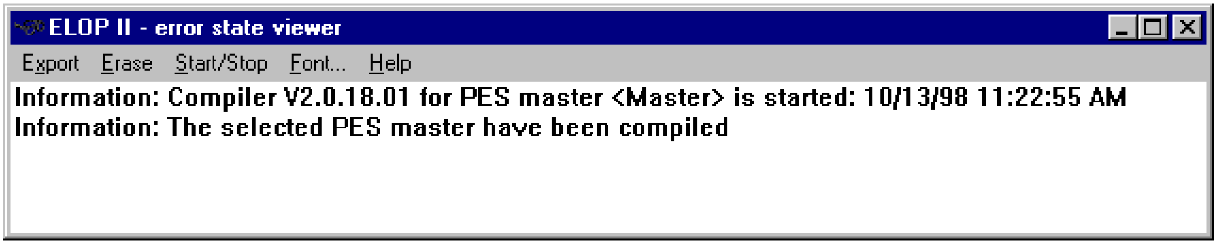
Name	Bus	BSN	CU	CP
Master	DemoBus	3	1	1
Master	DemoBus	1	1	1

PES-Master

Command buttons

You can compile and download PES masters, update the data in the list boxes and open a cross-reference window in which the variables transmitted by the PES masters are documented.

- ´Execute´ under the ´Compile´ list box: Compile the PES masters Highlight the PES master(s) you wish to compile and click on ´Execute´. The selected PES masters will be compiled. The error status display opens and shows you how the compile routine is progressing.

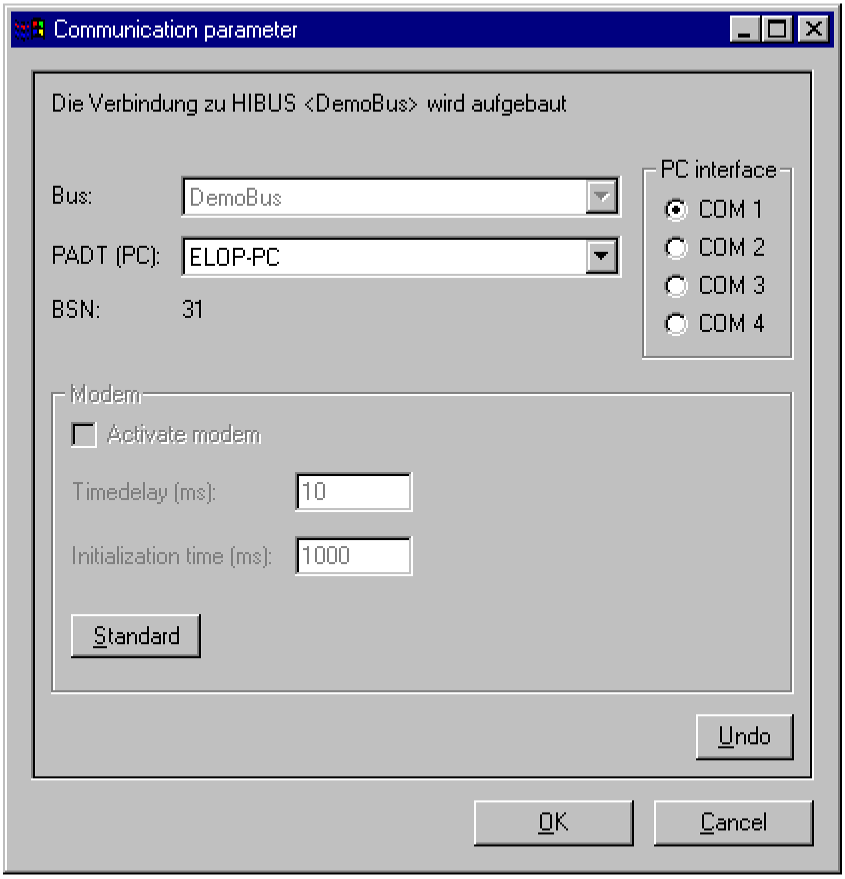


```
Information: Compiler V2.0.18.01 for PES master <Master> is started: 10/13/98 11:22:55 AM
Information: The selected PES master have been compiled
```

Error status display after running the PES master code generation

- ´Execute´ under the ´Download´ list box: Transfer the PES masters to the coprocessor modules
Highlight the PES master(s) you wish to load and click on ´Execute´. You will see a dialogue box in which you can enter the PADT

communication parameters.



Die Verbindung zu HIBUS <DemoBus> wird aufgebaut

Bus: DemoBus

PADT (PC): ELOP-PC

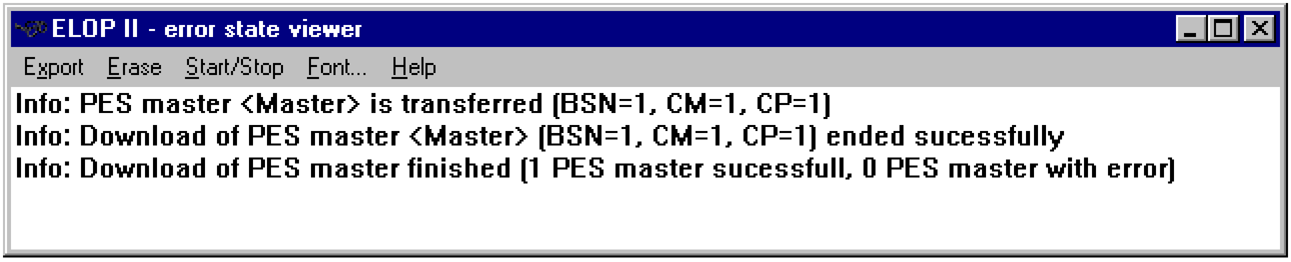
BSN: 31

PC interface
● COM 1
○ COM 2
○ COM 3
○ COM 4

Modem
□ Activate modem
Timedelay (ms): 10
Initialization time (ms): 1000

Dialogue box ´Communication parameters

Select the name of the required PC master from the list box and define the serial interface of the PC that will be used for transmitting data. Click on ´OK´ to transmit the selected PES masters. The error status display opens and shows you how the PES master download is progressing.

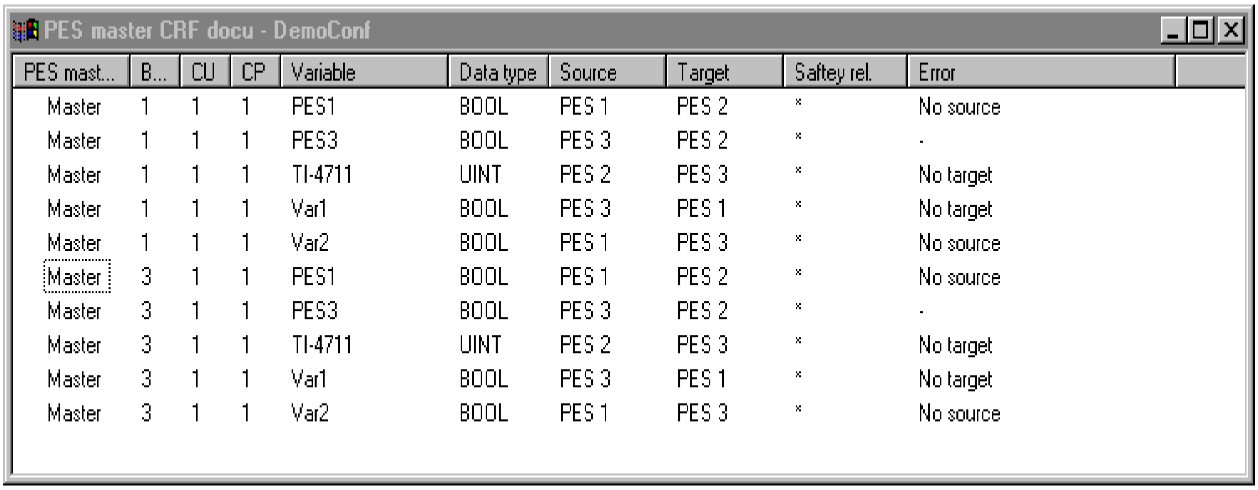

ELOP II - error state viewer
Export Erase Start/Stop Font... Help

```
Info: PES master <Master> is transferred (BSN=1, CM=1, CP=1)
Info: Download of PES master <Master> (BSN=1, CM=1, CP=1) ended sucessfully
Info: Download of PES master finished (1 PES master sucessfull, 0 PES master with error)
```

Error status display after running the PES master download

When the download is complete you will see a dialogue box containing information about the status of the transfer.

- ´Update´: Update data in both lists
  Action is possible when the code generator for a resource is running while the ´PES master´ dialogue box is open. Instead of quitting the dialogue box and opening it again when the code generator run is complete, you can call ´Update´.
- ´Cross-reference´: Output documentation of the data transmitted by the PES masters.



PES master CRF docu - DemoConf

PES mast...	B...	CU	CP	Variable	Data type	Source	Target	Saftey rel.	Error
Master	1	1	1	PES1	BOOL	PES 1	PES 2	*	No source
Master	1	1	1	PES3	BOOL	PES 3	PES 2	*	-
Master	1	1	1	TI-4711	UINT	PES 2	PES 3	*	No target
Master	1	1	1	Var1	BOOL	PES 3	PES 1	*	No target
Master	1	1	1	Var2	BOOL	PES 1	PES 3	*	No source
Master	3	1	1	PES1	BOOL	PES 1	PES 2	*	No source
Master	3	1	1	PES3	BOOL	PES 3	PES 2	*	-
Master	3	1	1	TI-4711	UINT	PES 2	PES 3	*	No target
Master	3	1	1	Var1	BOOL	PES 3	PES 1	*	No target
Master	3	1	1	Var2	BOOL	PES 1	PES 3	*	No source

CRF-Documentation

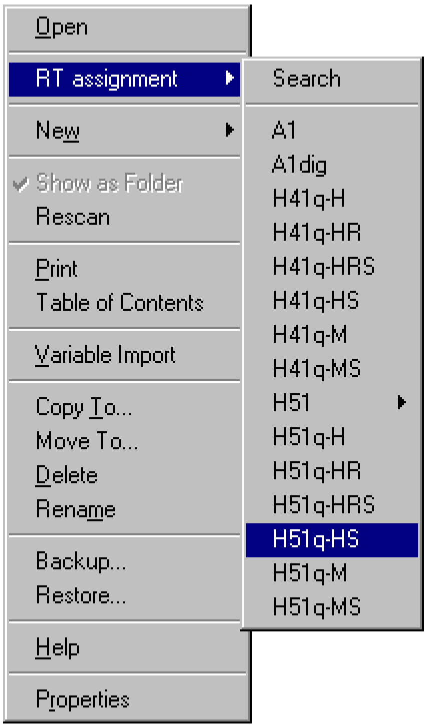
2 Configuring the resource

Resources structure the configuration of a system in the next stage. Each resource symbolizes a PES and contains one or more programs that are run under the controller by zero or more tasks. Resources are created with the context menu of the configuration (´New´ option in the ´Resource´ submenu) and they can contain the following objects:

- Program instances
- Type instances
- Tasks
- Global variables
- Libraries
- Structuring folders
- Connections

A newly generated resource is not yet standardized (“generic“) and does not yet possess any special, manufacturer-specific properties. You only define the hardware of the target PES when you assign a resource type. In this way the resource is given the properties and functions of a certain controller type which are associated with the resource type.

To assign a resource type, proceed as follows:
Click on the resource with the right-hand mouse button and point the mouse pointer to ´RT Assignment´ in the context menu which now appears. Now click on the desired resource type in the following submenu. Alternatively you can assign a resource type interactively by selecting the ´Search´ submenu option:

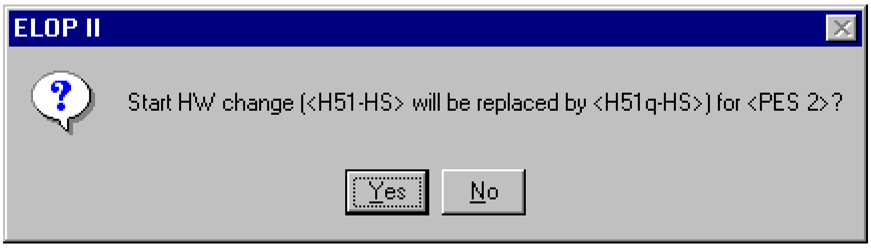


Dialogue box RT assignment

Once a resource type has been assigned the ´RT Assignment´ option in the context menu of the resource changes to the option.

You can change an assigned resource type at any time:

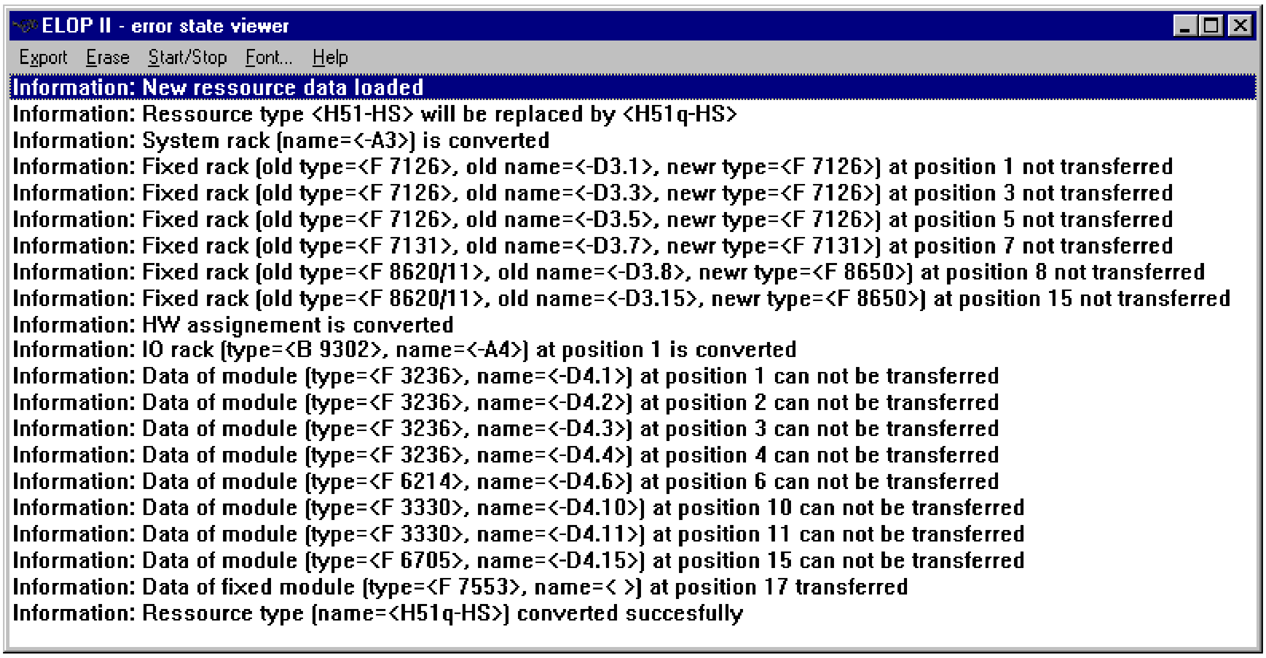
Click on the resource with the right-hand mouse button and point the mouse pointer to ´Hardware change <*Resource Type*>´ in the context menu which now appears. Now click on the desired new resource type in the following submenu, or start the ´RT assignment´ dialogue from the ´Search´ submenu option. You are prompted to confirm your choice before the action is executed:


ELOP II

Start HW change (<H51-HS> will be replaced by <H51q-HS>) for <PES 2>?

Dialogue box ´Start HW change ...´

When you confirm by clicking on ´Yes´ the error status display opens and shows you how the hardware change is progressing.



```
Information: New ressource data loaded
Information: Ressource type <H51-HS> will be replaced by <H51q-HS>
Information: System rack (name=<-A3>) is converted
Information: Fixed rack (old type=<F 7126>, old name=<-D3.1>, newr type=<F 7126>) at position 1 not transferred
Information: Fixed rack (old type=<F 7126>, old name=<-D3.3>, newr type=<F 7126>) at position 3 not transferred
Information: Fixed rack (old type=<F 7126>, old name=<-D3.5>, newr type=<F 7126>) at position 5 not transferred
Information: Fixed rack (old type=<F 7131>, old name=<-D3.7>, newr type=<F 7131>) at position 7 not transferred
Information: Fixed rack (old type=<F 8620/11>, old name=<-D3.8>, newr type=<F 8650>) at position 8 not transferred
Information: Fixed rack (old type=<F 8620/11>, old name=<-D3.15>, newr type=<F 8650>) at position 15 not transferred
Information: HW assignement is converted
Information: IO rack (type=<B 9302>, name=<-A4>) at position 1 is converted
Information: Data of module (type=<F 3236>, name=<-D4.1>) at position 1 can not be transferred
Information: Data of module (type=<F 3236>, name=<-D4.2>) at position 2 can not be transferred
Information: Data of module (type=<F 3236>, name=<-D4.3>) at position 3 can not be transferred
Information: Data of module (type=<F 3236>, name=<-D4.4>) at position 4 can not be transferred
Information: Data of module (type=<F 6214>, name=<-D4.6>) at position 6 can not be transferred
Information: Data of module (type=<F 3330>, name=<-D4.10>) at position 10 can not be transferred
Information: Data of module (type=<F 3330>, name=<-D4.11>) at position 11 can not be transferred
Information: Data of module (type=<F 6705>, name=<-D4.15>) at position 15 can not be transferred
Information: Data of fixed module (type=<F 7553>, name=< >) at position 17 transferred
Information: Ressource type (name=<H51q-HS>) converted succesfully
```

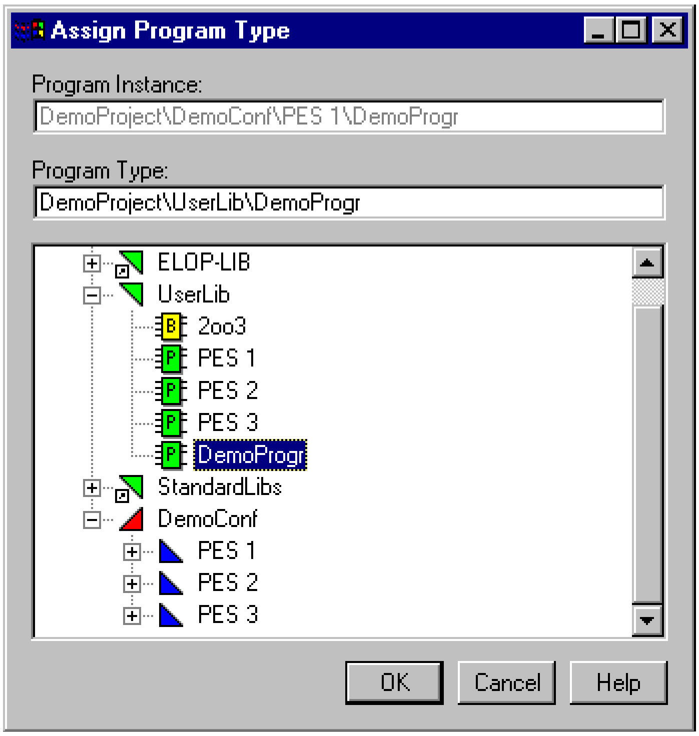
Error status display after running a hardware change

When the conversion is complete you will see a dialogue box containing information about the status of the resource type change.

2.1 Instancing a program type to the resource

Program instances execute the functionality that is defined in an associated program type declaration on the PES of your resource. They do not move or copy anything - they merely create an instance (an example) of the program type declaration. After instancing you can continue editing the program instance by double clicking. Whereas the program type declaration describes the functionality of a specific controller, in the program instance the properties of the target PES can be accessed as well.

Several instances of a program type can be created in different resources. To create a program instance select the ´New´ option from the context menu of the appropriate resource, then select the ´Program instance...´ submenu. A dialogue box now opens in which you can select the program type to which to assign the program instance.



Assign Program Type

Program Instance:
DemoProject\DemoConf\PES 1\DemoProgr

Program Type:
DemoProject\UserLib\DemoProgr

Dialogue box ´Assign program type´

To assign the program type, highlight the assignment target in the project structure window, and the name will be immediately matched under the ´Program type´ input box. The program type will be instanced to the resource when you confirm with ´OK´.

Note:

IEC 61131-3 allows a resource to contain a number of program instances that are run by zero or more tasks in the PES. This option is not available with the current PESs of HIMA, i.e. only one program instance can be contained.

You can assign another program instance to a resource at any time by deleting the program instance in the resource using its context menu, and then repeating the steps described above.

2.2 Configuring a resource

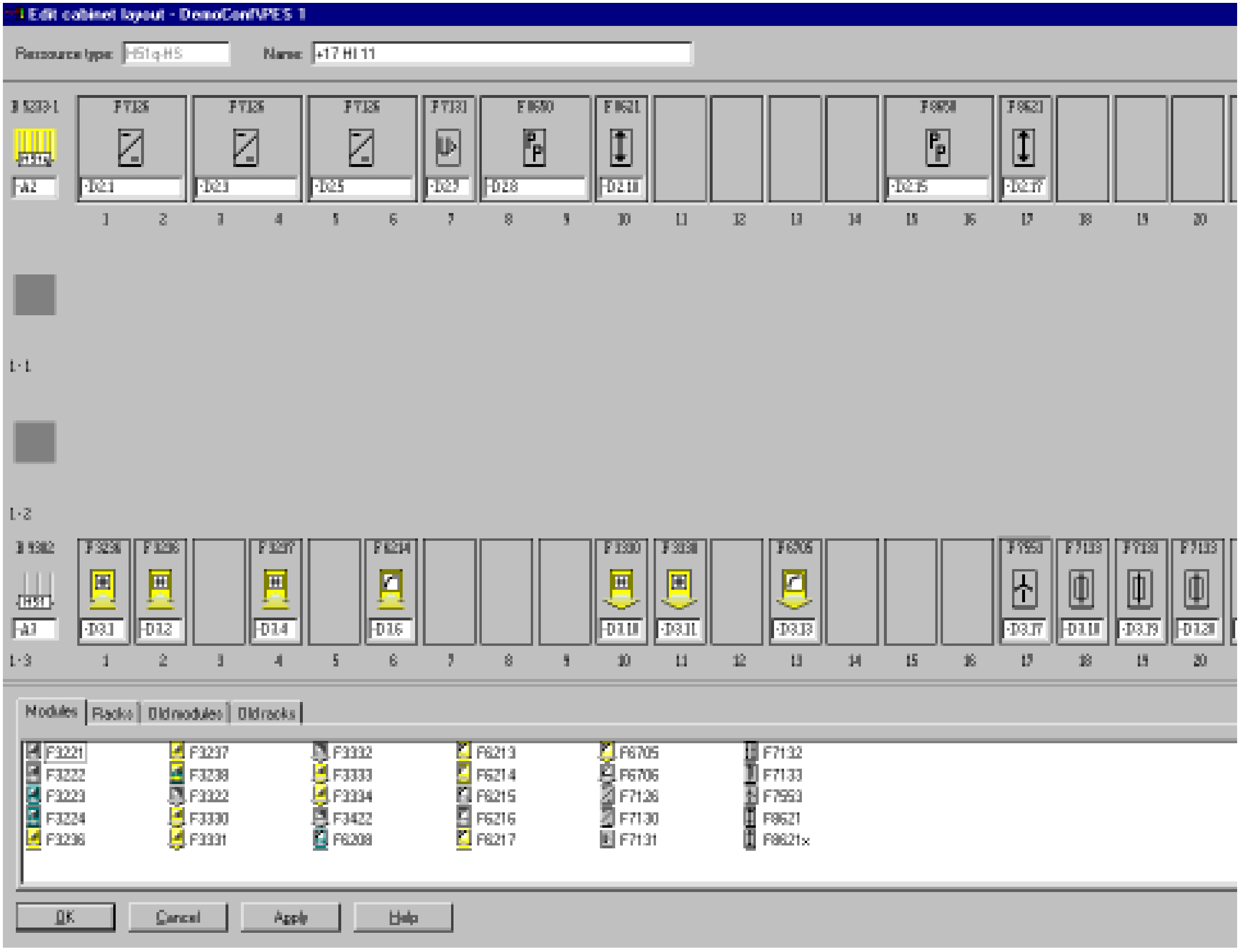
If the resource has been assigned a resource type, you can access the special properties and functions of the Programmable Electronic System (PES) which it defines.

You can use the ´Edit cabinet´ option in the context menu of the resource to define which modules are installed in your PES, and where. You can also set up parameters for the modules you use (See “Editing Cabinet layout” on page 13.).

You can configure your resource with the ´Properties´ option in the context menu of the resource. The tabs described below are displayed for the HIMA PESs as well as the default properties of all objects. See “Resource properties”.

2.2.1 Editing Cabinet layout

To define the modules of your PES click on the resource with the right-hand mouse button. Now click on the ´Edit cabinet´ option in the context menu which now appears. A dialogue box opens with a schematic of the cabinet and you can define and parameterize the modules to be used.



Edit cabinet layout - DemoConf\PES 1

Resource type: H51q-HS Name: +17 HI 11

Modules | Racks | Old modules | Old racks

F3221
F3222
F3223
F3224
F3236
F3237
F3238
F3322
F3330
F3331
F3332
F3333
F3334
F3422
F6208
F6213
F6214
F6215
F6216
F6217
F6705
F6706
F7126
F7130
F7131
F7132
F7133
F7553
F8621
F8621x

Dialogue box ´Edit cabinet´

The subracks are shown at the top of the dialogue box. A symbol on the left indicates the subrack type, and on the right are the subrack slots. All available modules and subracks are listed in the lower part of the dialogue box and can be selected by index pages.

You can enter comments about the PES in the ´Name´ input box on the first line next to the resource type, e.g. you can enter the name of the cabinet where the PES is installed. The subrack at the top symbolizes the central unit. Wildcard symbols for 16 I/O subracks are shown under the central unit for the H51 and H51q systems. The numbers beneath the wildcard describe the subrack addressing (e.g. 1-3: I/O bus/ cubicle 1, I/O subrack 3) that must be set using the coding switch on the connecting module. The numbers 1...21 under the module slots correspond to the physical arrangement of the slots in the subrack. To use an I/O subrack, select a subrack from the ´Subracks´ or ´Old subracks´ index page and drag it to a subrack wildcard symbol with the ´Drag&Drop´ feature. You can enter a commentary (e.g. a location code) about the selected subrack in the input box beneath it.

Note:

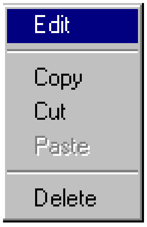
The subrack address you assign here will also be used to display errors in *the central module display. To assist fault-finding you should use addresses that correspond to the physical code number of the subrack in the cubicle.*

You can now select the modules you wish to use from the ´Modules´ or ´Old modules´ index page and use Drag&Drop to move them to a module slot in the subrack. Module information can also be entered in the commentary box.

If you drag a module to an assigned slot, the data of the original module will be accepted (if possible). In this way you can change the type of a previously placed module without having to enter new data.

Context menu

You can open a context menu for the corresponding object above a commentary box, a positioned subrack or a positioned module.



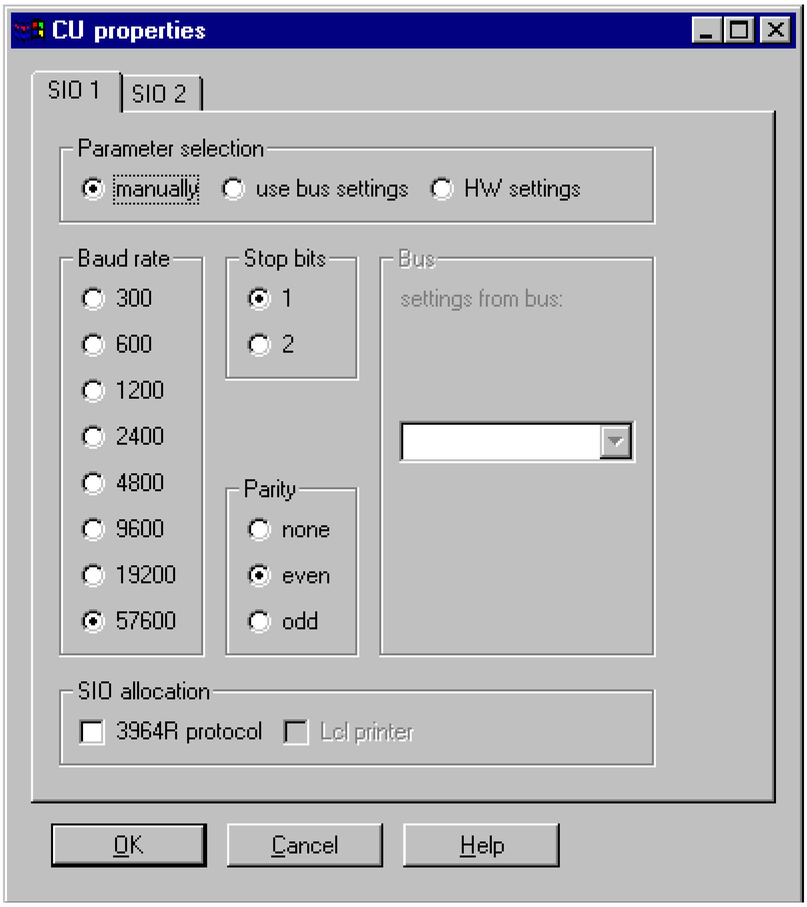
Context menu

Editing modules

Double clicking on a central module, a coprocessor module or an I/O module, or selecting the ´Edit´ option from the context menu of the module opens a module-specific dialogue box for you to configure the module´s parameters.

CU properties, CM properties

A dialogue box opens for central or coprocessor modules for you to configure the communication parameters of the module´s two serial interfaces.



CU properties

SIO 1 | SIO 2

Parameter selection: ● manually ○ use bus settings ○ HW settings

Baud rate: ○ 300 ○ 600 ○ 1200 ○ 2400 ○ 4800 ○ 9600 ○ 19200 ● 57600

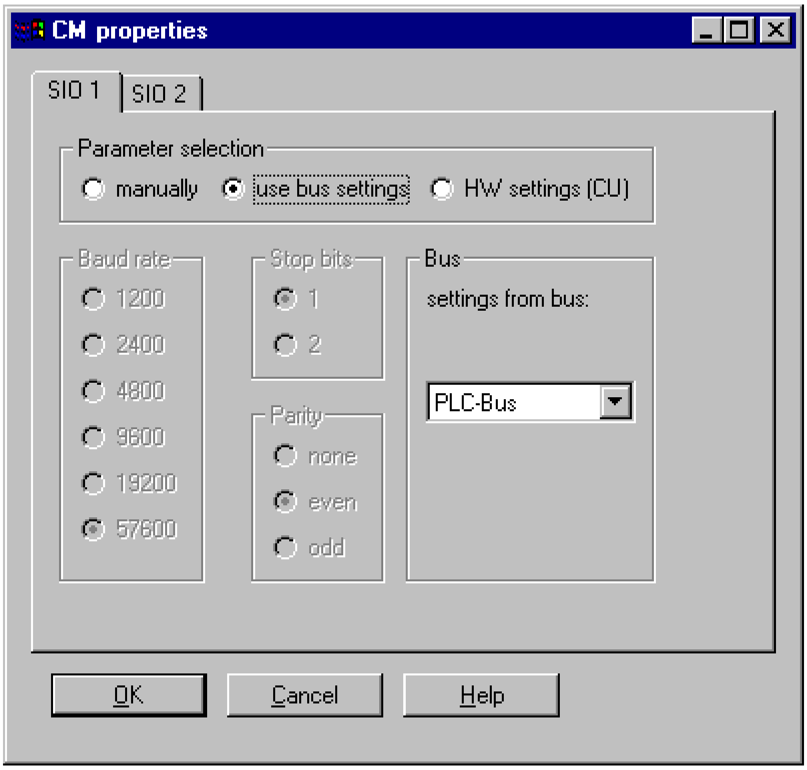
Stop bits: ● 1 ○ 2

Parity: ○ none ● even ○ odd

Bus: settings from bus:

SIO allocation: □ 3964R protocol □ Lcl printer

Dialogue box ´CU properties´



CM properties

SIO 1 | SIO 2

Parameter selection: ○ manually ● use bus settings ○ HW settings (CU)

Baud rate: ○ 1200 ○ 2400 ○ 4800 ○ 9600 ○ 19200 ● 57600

Stop bits: ● 1 ○ 2

Parity: ○ none ● even ○ odd

Bus: settings from bus: PLC-Bus

Dialogue box ´CM properties´

There are three modes of parameter selection for configuring a serial interface:

- Manual:
  Set up all the parameters yourself (baud rate, stop bits, parity)
- By bus configuration:
  ´Enter parameters from bus´ selection box is activated. When you select a bus from the list, the system will enter the parameters of that bus, and any subsequent changes to the bus configuration will then be entered automatically.
- Hardware presetting:
  Serial interface is not configured by ELOP II, and the parameters set by switch S1 on the central module are used instead.

A special communication protocol for the interfaces of the central module can also be selected using the small control box in the ´SIO reservation´ box:

- 3964R communication:
  Reserved interface communicates as a slave system with protocol 3964R.
- Lcl printer:
  Printer for logic plan controlled logging (Lcl) is connected to this interface. This is only available for central module interface 2.

Note:

In a redundant central unit the left and right hand halves of the unit can be regarded as a single logic system so they have the same parameters. It is therefore irrelevant whether the parameters on the left hand CU/CMs or right hand CU/CMs are configured.

Editing the tag name

Each I/O module consists of inputs and/or outputs to which you can assign symbolic names (tag names). As soon as you have assigned tag names to the I/O modules, variables of the program instance can be assigned to those tag names. In this way you can define I/O addresses for the variables

There are two basic methods for assigning addresses:

First define the I/O modules of the PES, then create the logic functionality

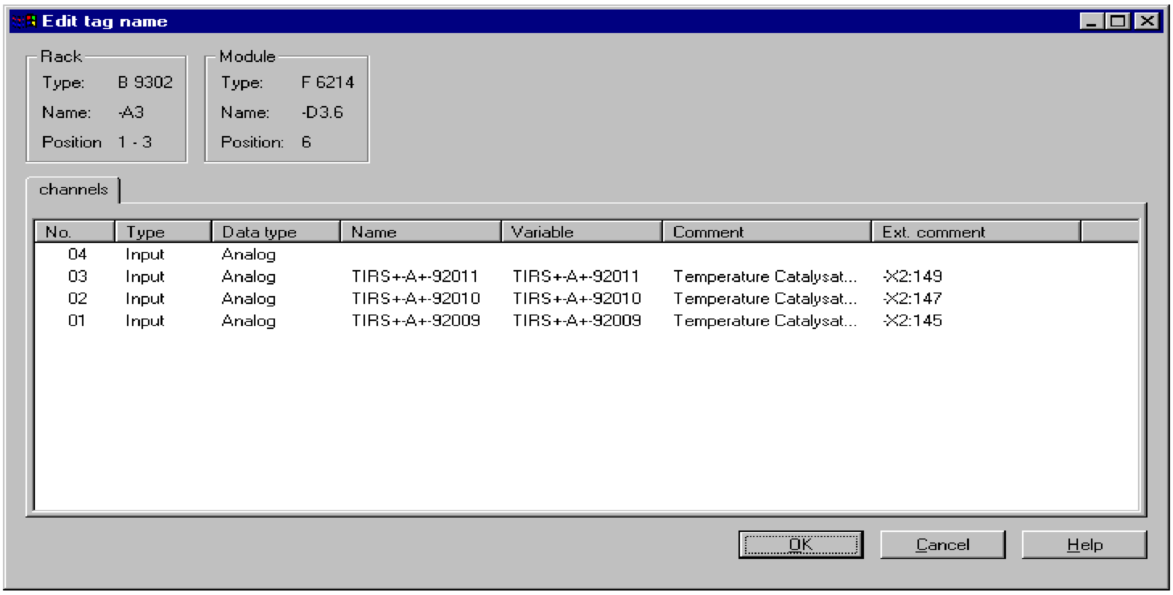
- Edit the cabinet
- Define the I/O modules to be used
- Create tag names for the I/O channels
- Create the program type
- Instance the program type to the resource
- Edit the program instance
- Assign the defined tag names to the variables of the program instance

First create the logic functionality of the PES, then define the I/O modules

- Create the program type
- Instance the program type to the resource
- Edit the cubicle
- Define the I/O modules to be used
- Define tag names for the I/O channels and assign the defined vari-

ables of the program instance to them.

Double clicking on an I/O module or selecting the ´Edit´ option from the context menu of the I/O module opens a module-specific dialogue box for you to assign tag names to the channels of the I/O module.



Rack
Type: B 9302
Name: -A3
Position 1 - 3

Module
Type: F 6214
Name: -D3.6
Position: 6

No.	Type	Data type	Name	Variable	Comment	Ext. comment
04	Input	Analog				
03	Input	Analog	TIRS+-A+-92011	TIRS+-A+-92011	Temperature Catalysat...	-X2:149
02	Input	Analog	TIRS+-A+-92010	TIRS+-A+-92010	Temperature Catalysat...	-X2:147
01	Input	Analog	TIRS+-A+-92009	TIRS+-A+-92009	Temperature Catalysat...	-X2:145

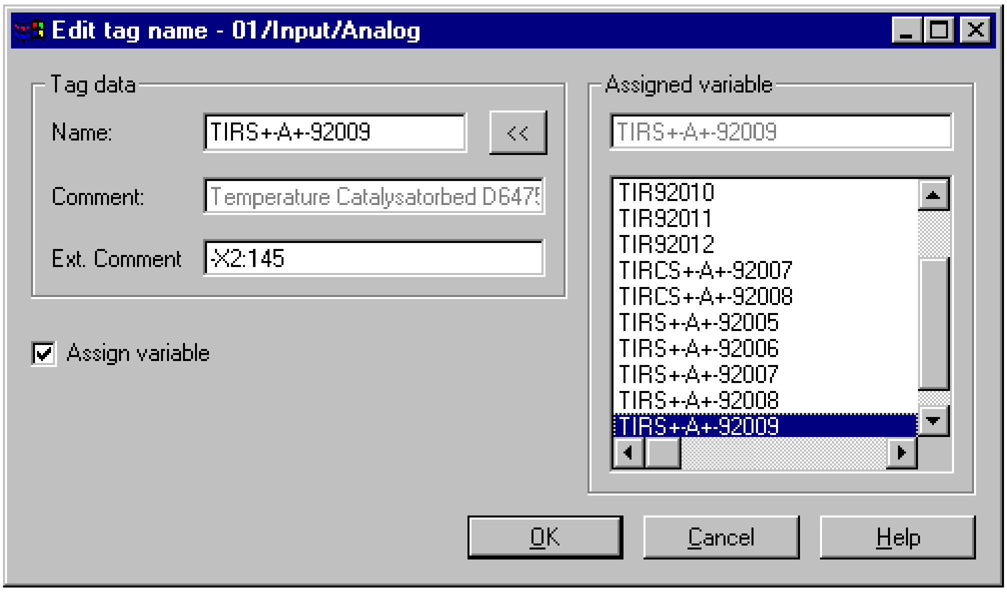
Dialogue box ´Edit tag name´

This dialogue box contains the individual I/O channels of the module and the tag names which have already been assigned. Details of the subrack and module are listed above the index. Columns in the index display the main data of the I/O channels of the module in table form:

- No.: Channel number of the I/O module
- Type: I/O type of the channel
  ´Input´ or ´Output´
- Data type: The data type of the channel
  ´Digital´ for assigning single-bit variables (variables of the BOOL data type), ´Analog´ for assigning multiple-bit variables (variables of the WORD, INT or UINT data type)
- Name: tag name of the channel
- Variable: assigned variable of the program instance of the resource
- Comment: A commentary on the tag name or long name of the assigned variable
- Ext. comment: any additional commentary

Editing tag names

Double clicking on a module channel opens a dialogue box in which you can edit the tag data:



Name: TIRS+-A+-92009

Comment: Temperature Catalysatorbed D647!

Ext. Comment -X2:145

☑ Assign variable

Assigned variable: TIRS+-A+-92009

Dialogue box ´Edit tag´

You can give a module channel an tag name, a commentary and an additional commentary, and assign the I/O channel a variable from the program instance of the resource.

- Name: Enter an tag name

Quick method:

- In the previous dialogue box ´Edit tag name´, highlight the channel number you wish to edit.
- Click in the ´Name´ column of the module channel.
- Enter the tag name in the text box.
- Press ENTER.

Note:

If you use identical symbolic identifiers for the tag names and for the variables of the program instance, then tag names and variables can be assigned automatically.

- Commentary: Your comments about the tag name
- Extra commentary: additional comments on the tag name

Quick method:

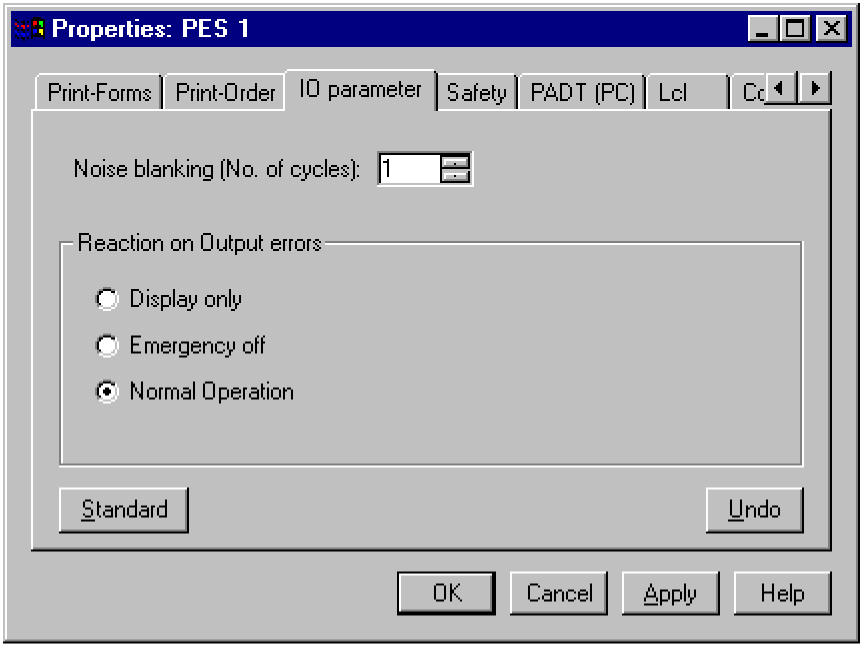
- In the previous dialogue box ´Edit tag name´, highlight the channel number you wish to edit.
- Click in the ´Extra comment´ column of the module channel.
- Enter your extra comments in the text box.
- Press ENTER.
- Assign variable:
  If a program type is instanced in the resource, then the variables of this program instance can be accessed. Assigning a variable to an tag name defines the I/O address of that variable.
- Click on the ´Assign variable´ control box.
- Double click on a variable name in the list box.
- The variable name will appear in the ´Assigned variable´ box, and the ´Comment´ box displays the long name of the variable.

- Click on command button ´<<´ if you want to use the variable name as the tag name.

2.2.2 Resource properties

You can configure your resource with the ´Properties´ option in the context menu of the resource. The tabs described below are displayed for the HIMA PESs as well as the default properties of all objects.

2.2.2.1 I/O parameters



Properties: PES 1

Noise blanking (No. of cycles): 1

Reaction on Output errors

○ Display only

○ Emergency off

● Normal Operation

Resource properties window, ´IO parameters´ tab

This tab is used to define the response of the PES to I/O faults.

- Noise blanking (cycles):
  The comprehensive tests of all components detect any deviation from the specification. The operating system tolerates transient faults with its integrated noise blanking feature. In addition, this input box defines the number of cycles for which I/O faults will be tolerated. The value 0 activates integrated noise blanking.

Note:

The system limits the number of noise blanking cycles to (safety time/ watchdog time)-2.

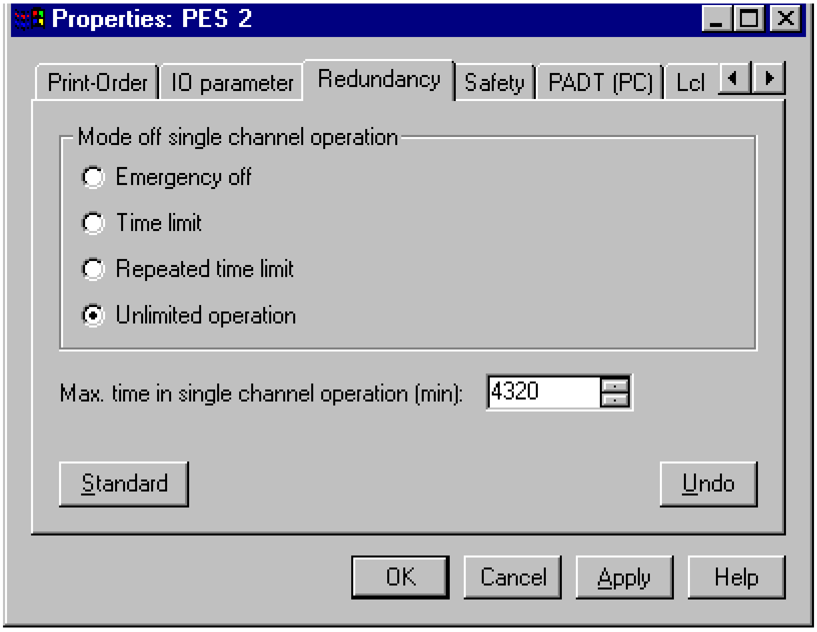
Example	1	2	3
Cycle time	100 ms	200 ms	200 ms
Watchdog time	300 ms	500 ms	500 ms
Safety time	1 s	2 s	1 s
Max. number of blanking cycles	1	2	0*

Table 1: Noise blanking
*Integrated noise blanking is active in this case.

The response of the controller to a fault in a testable output amplifier can be defined by one of 3 different parameters depending on safety and/or availability.

- Display only:
  Faulty modules are shut down by the integral safety shutdown feature. If a module does not permit shutdown, then the shutdown of the subrack by the connection module is not safety-related. When an H8-STA-3 function block is used (see also the description of the function block) all the modules entered on the block are shut down by the integrated safety shutdown if a module is faulty.
- Emergency stop:
  General shutdown (emergency stop) of the PES in the event of a fault in an output amplifier or an I/O bus fault. If the PES has a redundant I/O bus, only the central module which has the fault in its I/O bus shuts down.
- Normal operation:
  Equal ´Display only´, but the connecting module is shut down safety-related by the shutdown of the watchdog signal.

2.2.2.2 Redundancy (H41/H51 systems only)



Mode off single channel operation

○ Emergency off
○ Time limit
○ Repeated time limit
● Unlimited operation

Max. time in single channel operation (min): 4320

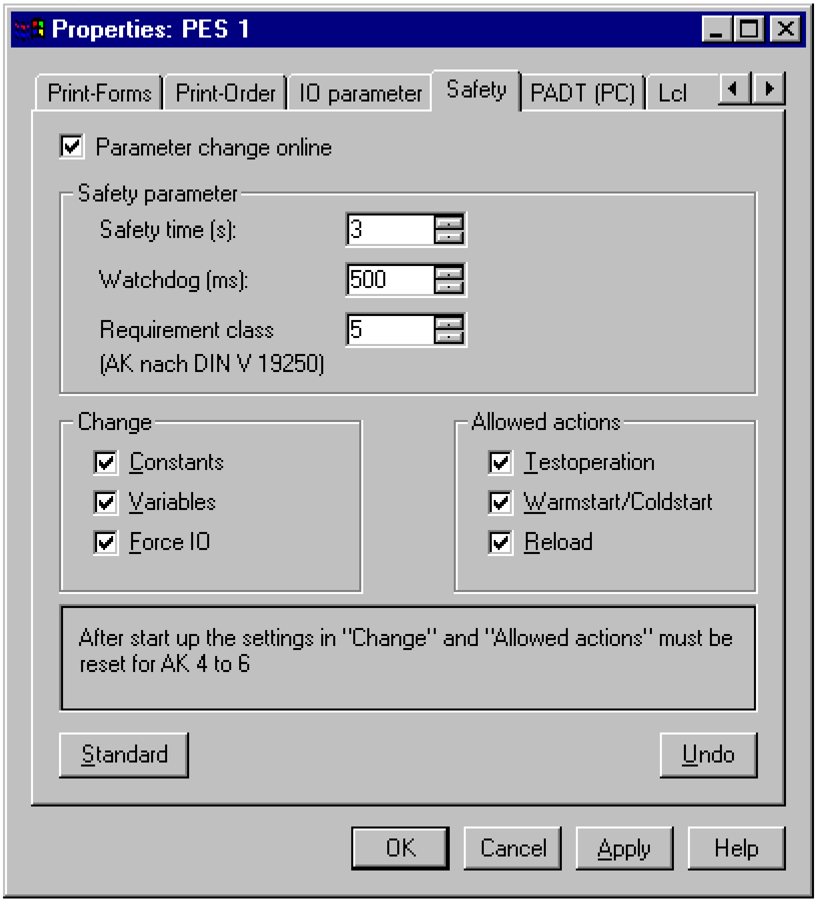
Resource properties window, ´Redundancy´ tab

On this tab you can define the response of a redundant PES in the event of failure by a central unit (not for H41q and H51q systems). Depending on the required level of safety and/or availability you can specify the redundancy loss mode by one of four different parameters:

- Emergency stop
  Emergency stop trips (general system shutdown) immediately when central unit fails.
- Timer limit
  One central unit fails, the other central unit runs on for the time in minutes specified in the ´Max. time in single channel operation´ box. When the timer times out a general PES shutdown is performed. If the faulty module is replaced by a functional module in the remaining time, then redundant mode is resumed and there is no further time restriction.

- Repeated time limit
  As for timer mode, but each time the ACK button is pressed on the central module the time in the ´Max. time 1-channel´ box is uploaded again and the shutdown is delayed.
- Unlimited mode:
  Unlimited 1-channel operation.
- Max. time in single channel operation (min):
  n minutes for single-channel central unit operation until general shutdown, valid for ´Time limit´ and ´Repeated time limit´. This time is process-dependent and must be agreed with the accepting authority.

2.2.2.3 Safety



☑ Parameter change online

Safety parameter

Safety time (s): 3

Watchdog (ms): 500

Requirement class (AK nach DIN V 19250): 5

Change

☑ Constants

☑ Variables

☑ Force IO

Allowed actions

☑ Testoperation

☑ Warmstart/Coldstart

☑ Reload

After start up the settings in "Change" and "Allowed actions" must be reset for AK 4 to 6

Resource properties window, ´Safety´ tab

This tab is used to define safety-critical parameters and to specify the permitted actions with the PADT during safety-related operation of the PES.

Note:

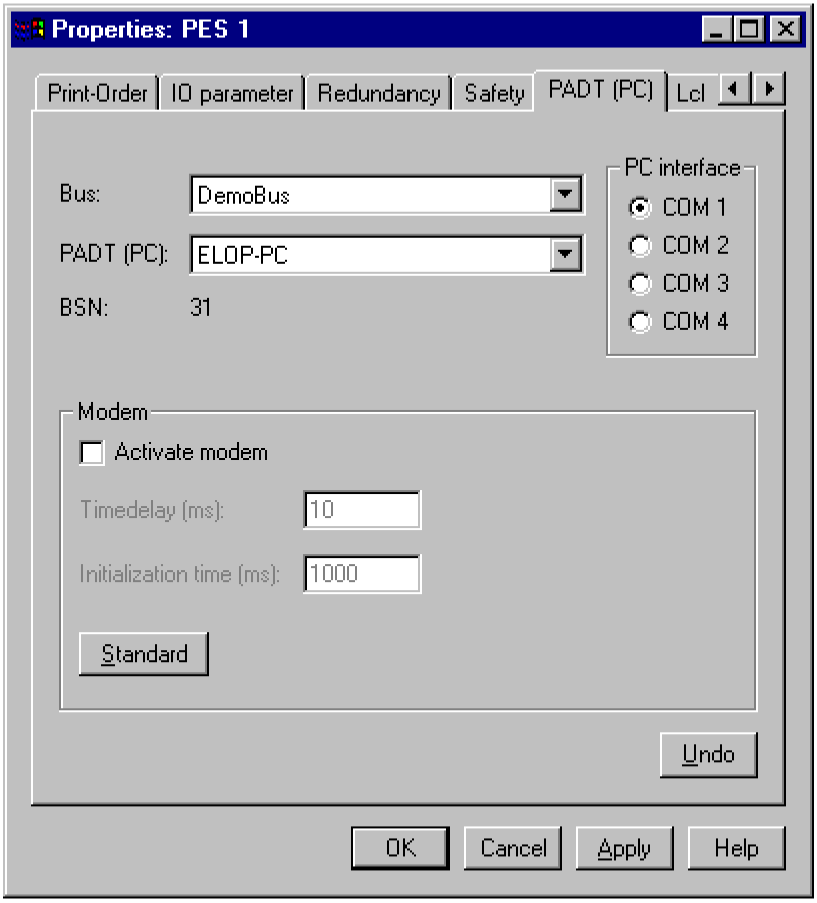
The assignments that are possible during safety-related operation are not rigidly bound by any particular class of requirements but they must be agreed with the approving authority responsible for each use of the PES.

- ´Parameters change online´:
  The changes to the safety parameters specified on this tab are disabled if the control box is deactivated.
- Safety time (s) (H41/H51only):
  Self-tests of a PES are divided into foreground and background

tests. Foreground tests are used to detect dangerous first faults and are performed within a cycle or within the safety time (in seconds) that is specified in this input box. Background tests are additional test routines used to detect faults that can have adverse safety-technical effects in conjunction with other faults. Background tests are performed in a longer time interval. Shortening the safety time therefore increases the cycle time. With long safety times some tests are spread over several cycles. The safety time must be matched to suit the controlled process.

- Watchdog (ms):
  Input box shows the cycle monitoring time for the system in milliseconds. It must not be more than half the safety time.
- Requirements class (to DIN V 19250):
  The safety-related use of the PES, this input box indicates the safety requirement (requirement classes 0 to 6 to DIN V 19250).
- Constants:
  Constants (VAR CONST) and system parameters cannot be changed online by the PADT unless the control box is activated.
- Variables:
  Variables cannot be changed (forced) online unless the control box is activated.
- Forcing IO:
  Force main switch and individual switches for the inputs and outputs of the PES cannot be set unless the control box is activated.
- Test mode
  Deactivating the control box disables execution of the following PADT commands:
  Start (if the PES is in RUN mode; the command is always executed in STOP)
  Stop
  Step
  Continue AP
- Hot start/Cold start:
  Deactivating the control box disables execution of all start commands of the PADT if the PES is in RUN mode; the commands are always executed in STOP.
- Reload:
  Reload cannot be executed in the Reload mode when the control box is deactivated.

2.2.2.4 PADT (PC)



Bus: DemoBus

PADT (PC): ELOP-PC

BSN: 31

PC interface
● COM 1
○ COM 2
○ COM 3
○ COM 4

Modem
□ Activate modem
Timedelay (ms): 10
Initialization time (ms): 1000

Resource properties window, ´Programming Unit´ tab

You can use this tab to define how you communicate with the PADT to the controllers. To do this, select the bus, the name of the PC master and the serial interface of the PC which you wish to use. If you use a telephone modem you can configure additional communication parameters for triggering the modem.

- Bus:
  Select a previously defined bus for your communication. See “Properties: Buses” of the configuration
- PADT (PC master):
  With which PC master you want to access the selected bus.
- PC interface:
  Select the communication interface by which the PC is connected to the bus

Note:

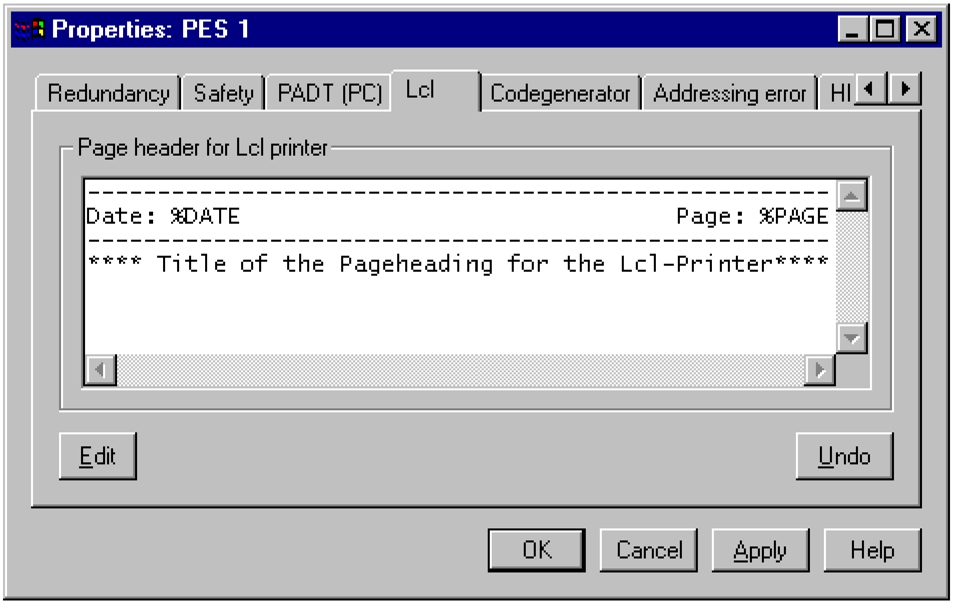
If using an interface converter e.g. H7505 please note that ELOP II-NT is not switching the DTR Signal for control of the conversion direction as a standard. An RS485 interface is recommended.

- Activate telephone modem:
  The PC master is connected to the selected bus via a modem, then highlight this control box. The DTR signal will then be switched first when operation starts.
- Time delay (ms):
  Communicating via a telephone modem this time can be increased to allow for the delay in response by the bus stations caused by the modem.

- Initializing time (ms):
  the time needed by the modem for dialing here.

2.2.2.5 Lcl Texts

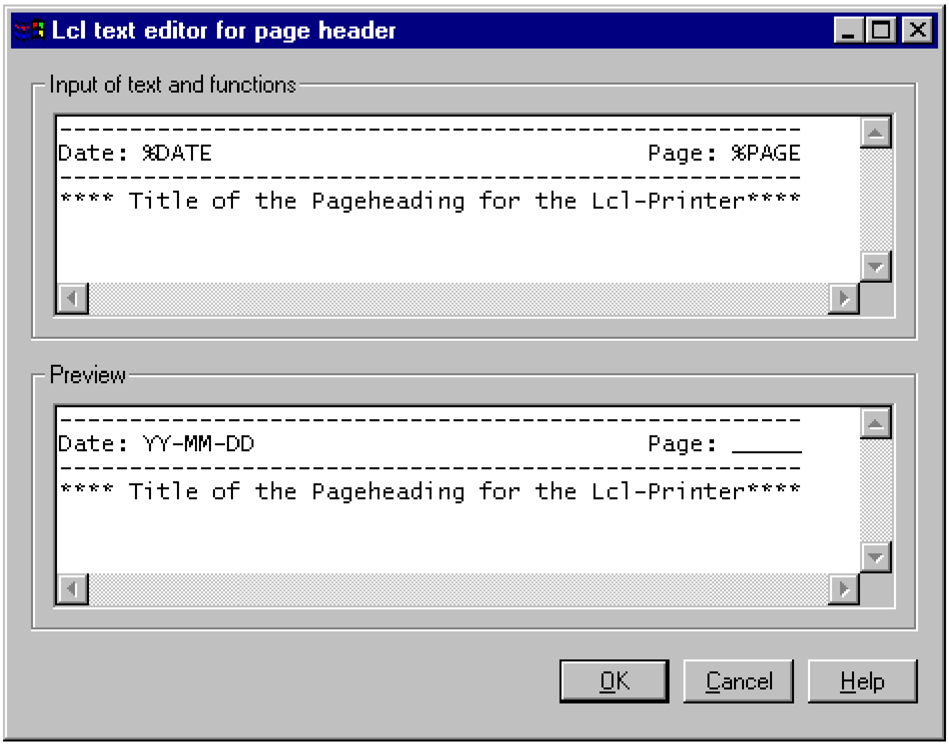
Logic controlled logging (Lcl) is used to record events (Boolean signal changes with time of day) on the central module and to print off the events - with interpretation - on a connected printer. You can define a page header for this report printout on this tab. A print preview of the Lcl page header is displayed in the ´Page header for Lcl output´ box.



```
-----------------------------------------------------
Date: %DATE                              Page: %PAGE
-----------------------------------------------------
**** Title of the Pageheading for the Lcl-Printer****
```

Resource properties window, ´Lcl´ tab

Click on the ´Edit´ command button to start interactive header creation.



```
-----------------------------------------------------
Date: %DATE                              Page: %PAGE
-----------------------------------------------------
**** Title of the Pageheading for the Lcl-Printer****
```

```
-----------------------------------------------------
Date: YY-MM-DD                           Page: _____
-----------------------------------------------------
**** Title of the Pageheading for the Lcl-Printer****
```

Dialogue box ´Lcl text editor for page header´

The dialogue is divided in two parts: you enter the texts and wildcards for the page header in the top half, and the bottom half displays a print preview. The following wildcards can be used:

- %CR: word wrapping (carriage return)
- %DATE: current date in format YY-MM-DD
- %PAGE: current page number of the printout

2.2.2.6 Code generator

The user program is translated into machine code (code generation) once you have entered the complete function, variables and resource declaration. The machine code required for the PES is generated depending on the selected resource type. After the code is generated the user program can be loaded into the PES memory

Here you are able to configure the codegenerator settings

- Acknowledge start of code generator
  Start of the code generator must be explicitly confirmed again if this option is selected
- Create reloadable code
  A reloadable code must be generated before online changes can be made.

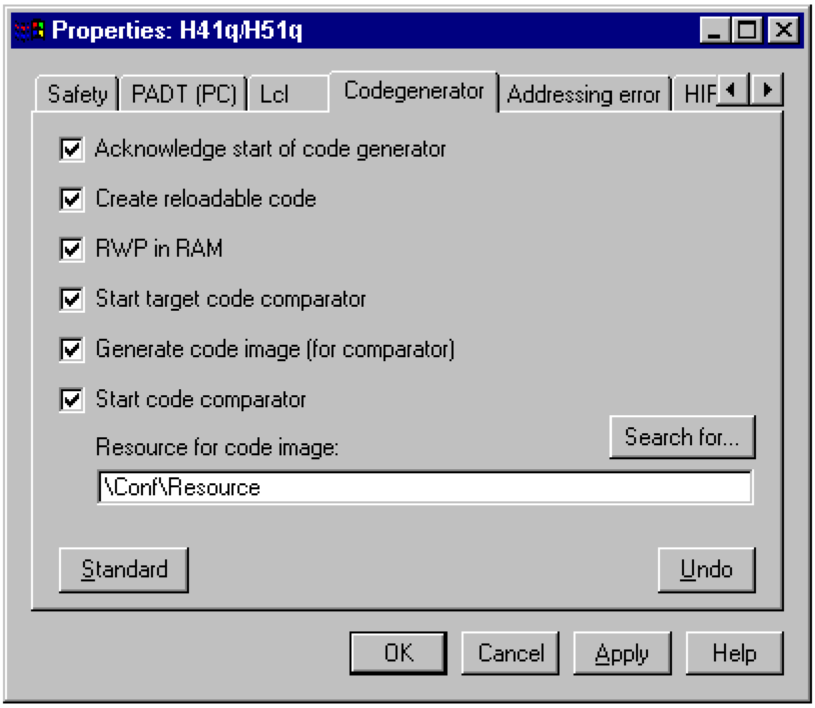
Note:

A reloadable code cannot always be generated. This is not possible when modules or communication variables are inserted or deleted.

- SLP in RAM
  The SLP area that is normally stored in the flash EPROM can be stored in the RAM if required. This will then enable variables to be changed during operation.

In resources H41q/H51q the following options for safety related revision control and code generation are available on top of these. These options are certified by TÜV Product Service (further Information in the chapter ´Procedure for safety related PES´).

- Start target code comparator
  By selecting this option the target code is generated in two separate tasks and compared afterwards. Random errors of the hardware platform would be detected this way.
- Generate code compare image
  The code compare image is needed for a later safety related revision check.
- Start code comparator
  By selecting this option changes within a resource will be detected. You need to select a resource to compare with. This could be exactly the same resource or a backup of an earlier revision.



☑ Acknowledge start of code generator

☑ Create reloadable code

☑ RWP in RAM

☑ Start target code comparator

☑ Generate code image (for comparator)

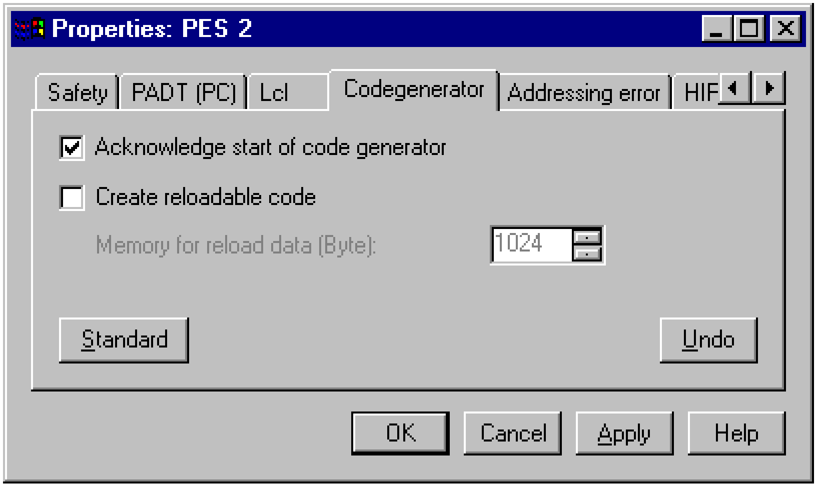
☑ Start code comparator

Resource for code image: \Conf\Resource

Resource properties window, ´Code generator´ tab for H41q/H51q family

Next you see the properties for the PES A1, A1dig, H11, H41, H51. The options for the safety related codegeneration and revision check are not available. The option ´RWP in RAM´ is deleted, since there is no other option in these systems.

Instead it is necessary to configure a memory area for reload data, since there is only a restricted area available. The default of 1024 bytes is a good compromise between program size and sufficient space for online changes.



☑ Acknowledge start of code generator

☐ Create reloadable code

Memory for reload data (Byte): 1024

Resource properties window, ´Code generator´ tab for systems A1, H11, H41/H51 family

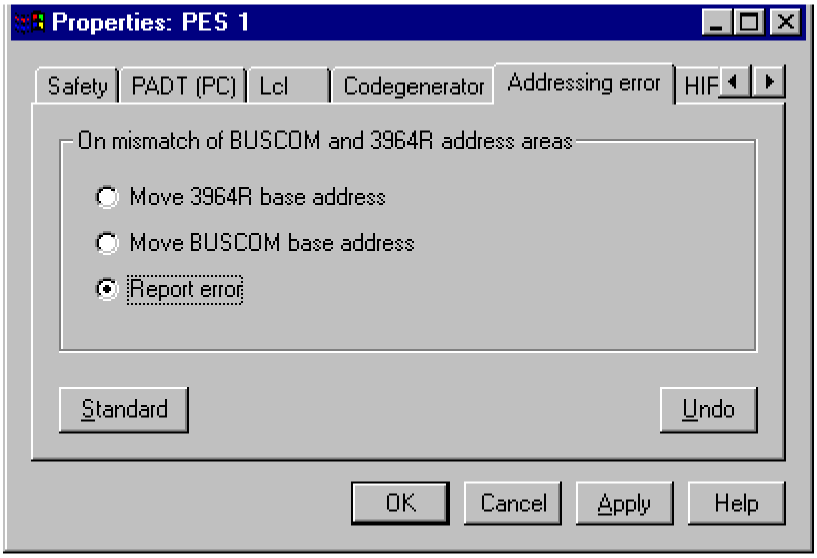
2.2.2.7 Address conflicts

If address conflicts occur during code generation, you can use this function to select a response. The same basic address (0) is specified as default for communication via 3964R and for BUSCOM, which is why there can be address conflicts:

- Move 3964R basic address

If this option is selected the 3964R basic address is automatically moved to a value that will avoid conflict with BUSCOM addresses used at the same time.

- Move BUSCOM basic address
  If this option is selected the BUSCOM basic address is automatically moved to a value that will avoid conflict with 3964R addresses used at the same time.
- Generate error
  If this option is selected, addresses will not be moved automatically and instead you will see an error message during compiling which draws your attention to the conflict. You must then make a manual correction.



On mismatch of BUSCOM and 3964R address areas

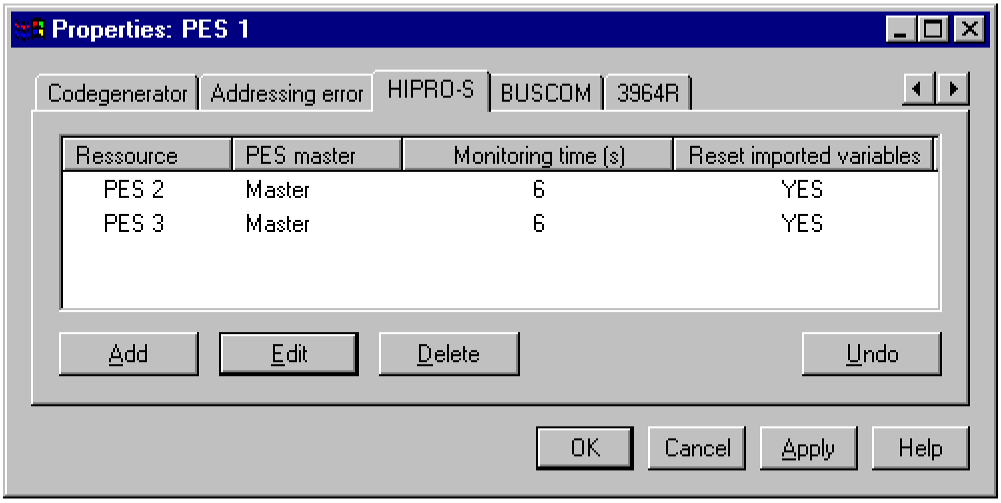
○ Move 3964R base address
○ Move BUSCOM base address
● Report error

Resource properties window, ´Addressing errors´ tab

2.2.2.8 HIPRO-S

The HIPRO-S protocol is used for safety-related communication between two or more PESs. The controllers communicate across the configured communication system.

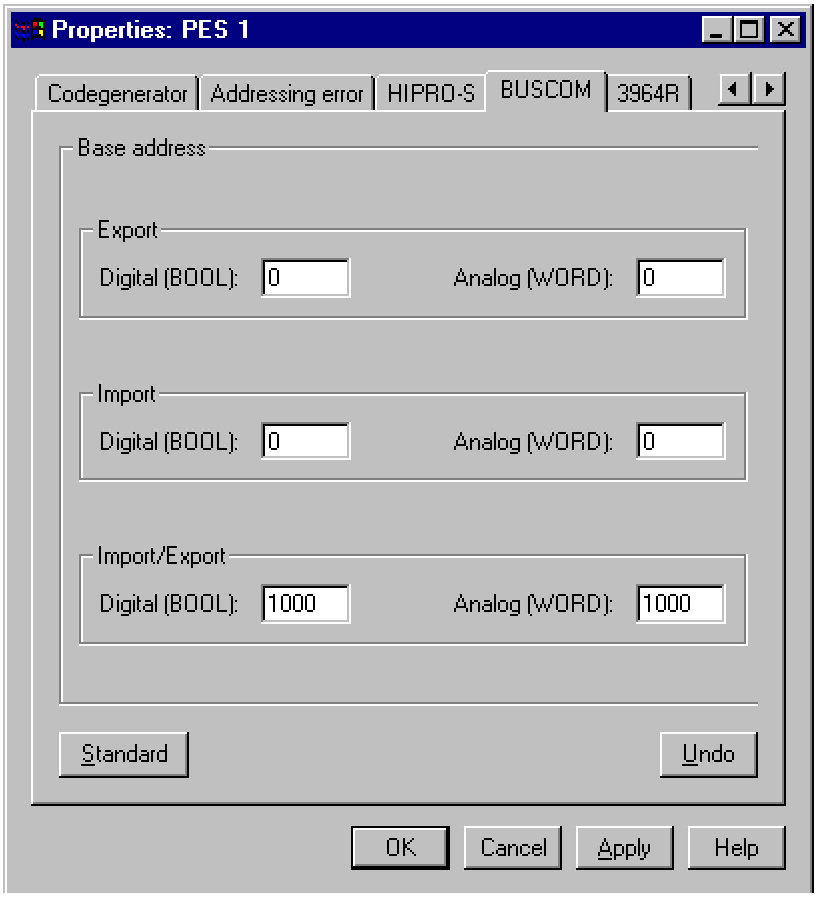
For each resource it must be configured with which other resources safety related data exchange is performed. The variables for the data exchange are configured in the variables declaration editor of the program instance (See “Safety-related data transfer: HIPRO-S” on page 38.)



Ressource	PES master	Monitoring time [s]	Reset imported variables
PES 2	Master	6	YES
PES 3	Master	6	YES

Resource properties window, ´HIPRO-S´ tab

2.2.2.9 BUSCOM protocol



Base address

Export
Digital (BOOL): 0
Analog (WORD): 0

Import
Digital (BOOL): 0
Analog (WORD): 0

Import/Export
Digital (BOOL): 1000
Analog (WORD): 1000

Resource properties window, ´BUSCOM´ tab

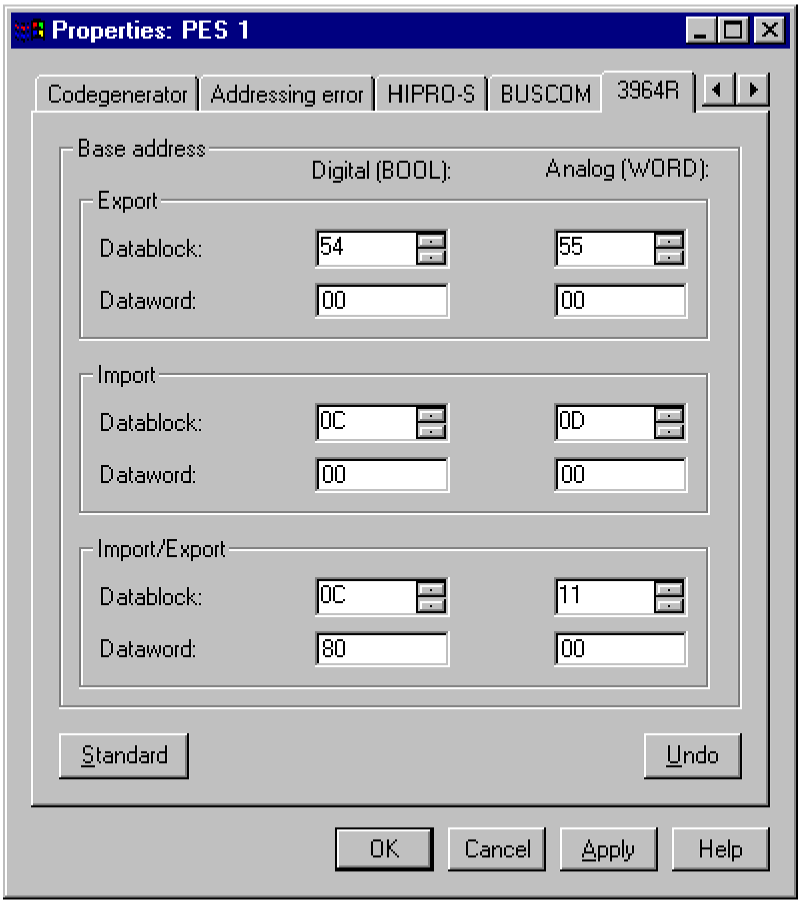
Here you can enter the basic addresses for BUSCOM, e.g. for READ and WRITE of Variables by a MODBUS master:

- Export
  Here are the basic addresses starting from which a MODBUS master can export variables of the PES with function code 1 (digital (bool)) or 3 (analog (word)).
- Import
  Here are the basic addresses starting from which a MODBUS master can import variables of the PES with function codes 5 and 15 (digital (bool)) or 6 and 16 (analog (word)).
- Export/Import
  Some control systems are unusual in that they immediately re-export the variables that they import. They only change the function code without changing the basic address, so a separate memory area must be reserved for write/read accesses of this type.

Note

For H41q/H51q systems the valid address range for BUSCOM is 0 to 2048 and 4096 to 8192.

2.2.2.10 3964R Protocol



Base address

	Digital (BOOL):	Analog (WORD):
Export		
Datablock:	54	55
Dataword:	00	00
Import		
Datablock:	0C	0D
Dataword:	00	00
Import/Export		
Datablock:	0C	11
Dataword:	80	00

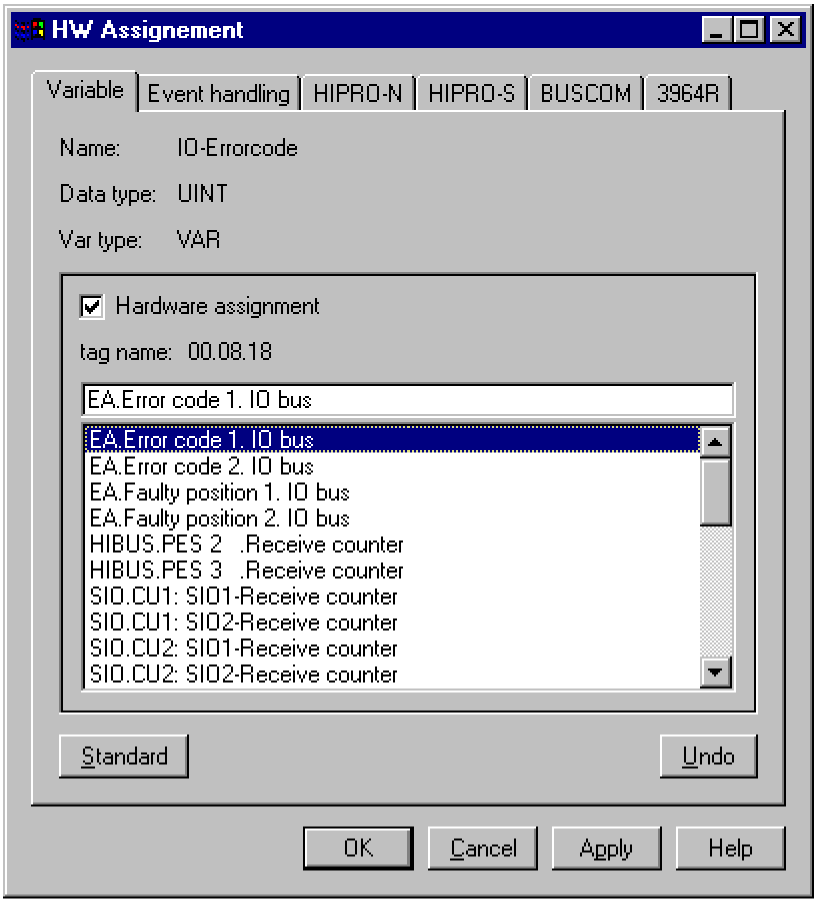
Resource properties window, ´3964R´ tab

The data blocks and data words required for communication via the 3964R protocol can be entered here.

3 Resource-related functions in the Editor

Once a program has been instanced to a resource, a number of resource-related items can be configured as you edit the program instance in the Editor.

3.1 Assigning system variables



Name: IO-Errorcode

Data type: UINT

Var type: VAR

☑ Hardware assignment

tag name: 00.08.18

EA.Error code 1. IO bus

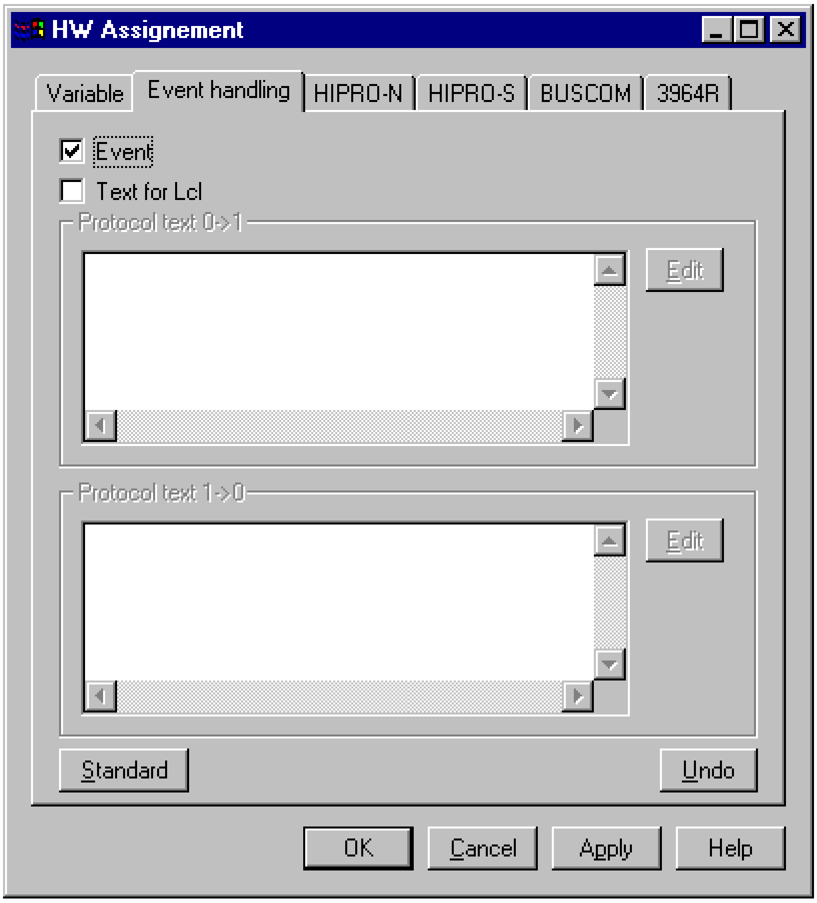
EA.Error code 1. IO bus
EA.Error code 2. IO bus
EA.Faulty position 1. IO bus
EA.Faulty position 2. IO bus
HIBUS.PES 2 .Receive counter
HIBUS.PES 3 .Receive counter
SIO.CU1: SIO1-Receive counter
SIO.CU1: SIO2-Receive counter
SIO.CU2: SIO1-Receive counter
SIO.CU2: SIO2-Receive counter

Variable properties window, ´Variable´ tab

To access the dialogue shown above, call the context menu of a variable used in the logic and then select the ´Properties´ option.

You will see different system variables in the selection window depending on the type of variable you have clicked on (BOOL or UINT). The system information is available after you click on ´Assign hardware´ and select a system variable.

3.2 Creating Events



☑ Event

□ Text for Lcl

Variable properties window, ´Events´ tab

All necessary data of the variables that are needed for the HIMA communication protocol (time stamp) is created for the variable when you select the ´Event-driven option.

3.2.1 Variables for transmission with HIPRO

HIPRO-N and HIPRO-S are the non-safety related and safety-related protocols used for data interchange between HIMA PESs. They work on the master/slave principle, and several masters can be connected to a bus as well as several slaves.

A maximum of 255 bus stations can be addressed on a HIPRO-N/S.

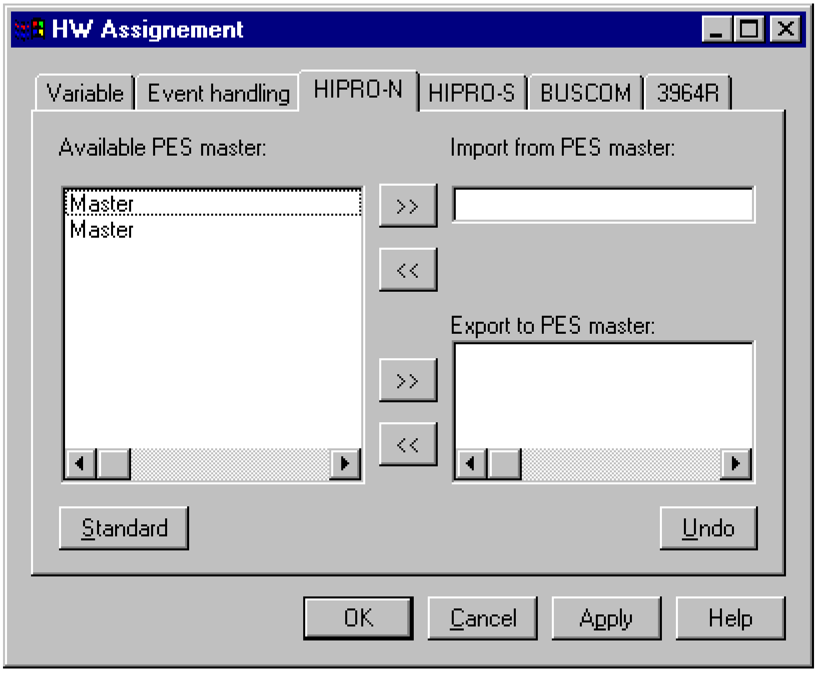
Because up to 8 bus stations per controller are possible in PESs that belong to the H41/H51 and H41/H51q family (2 CPUs and 6 coprocessor modules), this maximum number must be divided by 8 to arrive at the number of bus stations.

Up to 31 bus stations in the H41/H51 and H41q/ H51q family can therefore be connected to a HIPRO-N/S as master or slave.

Every bus station on the HIPRO-N/S is identified by a user-defined bus station number (BSN) that is set on the system´s central module by means of DIP switches. Since every master/slave in a bus station can be connected to a separate bus, each bus station can be connected to up to 8 buses. The number of controllers that can interchange safety-related data on the HIPRO-S is limited to 6 in the H41/H51 family and 64 bus stations in the H41q/H51q family.

Functions of the PES master

- PES masters as data centre for HIPRO-N
  With this type of data transfer the PES master acts as the data centre, reading the data of the connected slaves, compiling the transmission and then sending it to the appropriate PESs.
  If data cannot be exported from a PES because of a failed connection during PES master operation, the PES master writes the data from this PES to the other PES to FALSE or, if configured, holds the data at its last value prior to the connection failure for a specified time or until the connection is restored.
  If changes to the user program of a resource are made in the configuration, then only the user program of the affected PES and PES master need to be re-compiled and re-loaded.
- PES master for starting safety-related transmissions
  The PES Master does not act as the data centre for safety-related transmissions via the HIPRO-S between different PESs, it only starts the transmissions. The necessary configurations for the PES master are made in the settings of the resource.
  The variables of the sending controller are transmitted directly to the receiving controller. Notionally, this type of transfer is like a point-to-point connection, except that it runs via the HIPRO-S. Individual transmissions are protected by codes and signatures.
  All variables that are exported to another resource must be the same as the variables imported from that resource.



Available PES master:
Master
Master

Import from PES master:

Export to PES master:

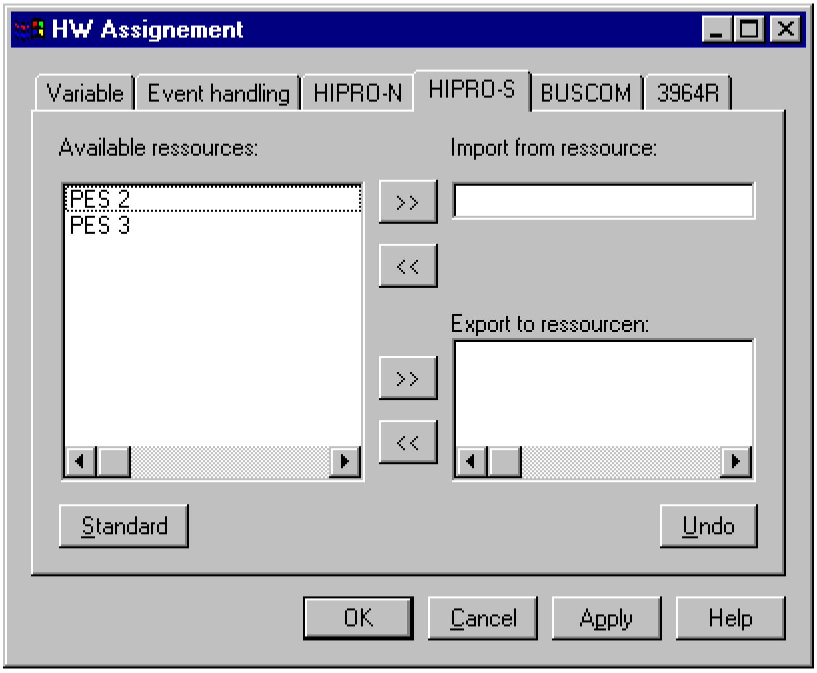
Variable properties window, ´HIPRO-N´ tab

In this tab you can define by which PES master the variables should be transferred non safety-related.

The data of the resource just edited can either be imported from a PES master (´Import from PES master´) or exported to several other PES master (´Export to PES master´).

Note:

You can only specify one PES master for ´Import from PES master´ because only a single master can access the variables for importing, whereas with ´Export to PES master´ several masters can be specified as *different masters can be permitted to access variables for export.*



PES 2
PES 3

Variable properties window, ´HIPRO-S´ tab

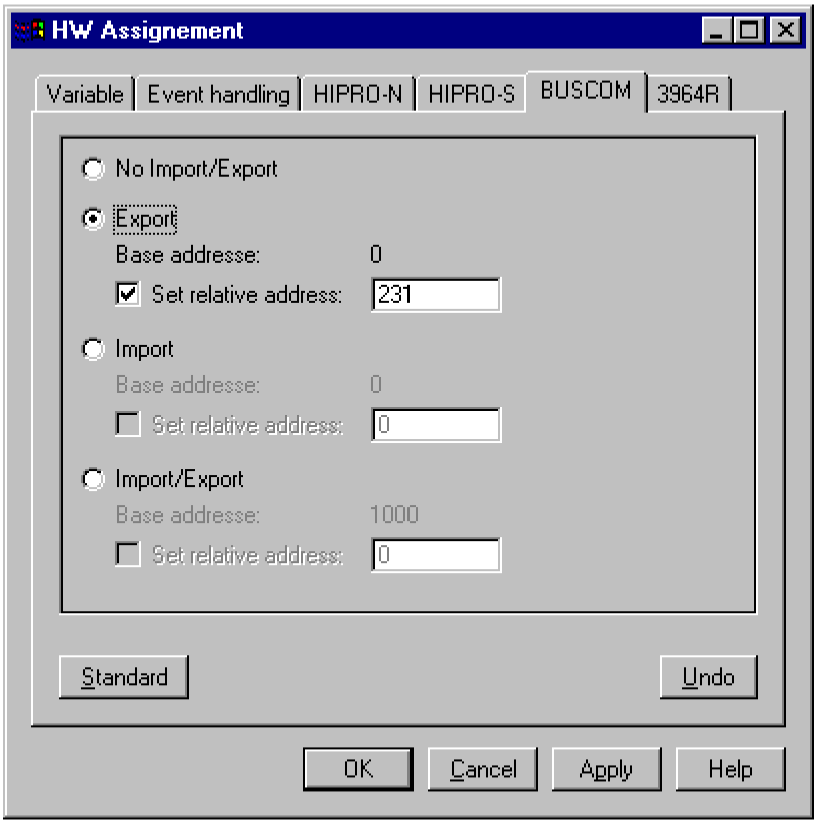
In this tab you can define two PESs between which data will be transferred safety-related. All resources on the HIPRO-S from which data can be imported or to which data can be exported are displayed under ´Available resources´.

Note:

You can only specify one resource for ´Import from resource´ because the currently edited resource may only take data directly from another (point-to-point connection), whereas several resources can be specified for ´Export to resources´ because the currently edited resource may send its data to several resources.

3.2.2 Variables for transmission to external systems

In the following tabs you can define options for access by external systems (MODBUS master, OPC Server, Profibus-DP master or 3964R master) to the HIMA PES variables.



○ No Import/Export

● Export

Base addresse: 0

☑ Set relative address: 231

○ Import

Base addresse: 0

□ Set relative address: 0

○ Import/Export

Base addresse: 1000

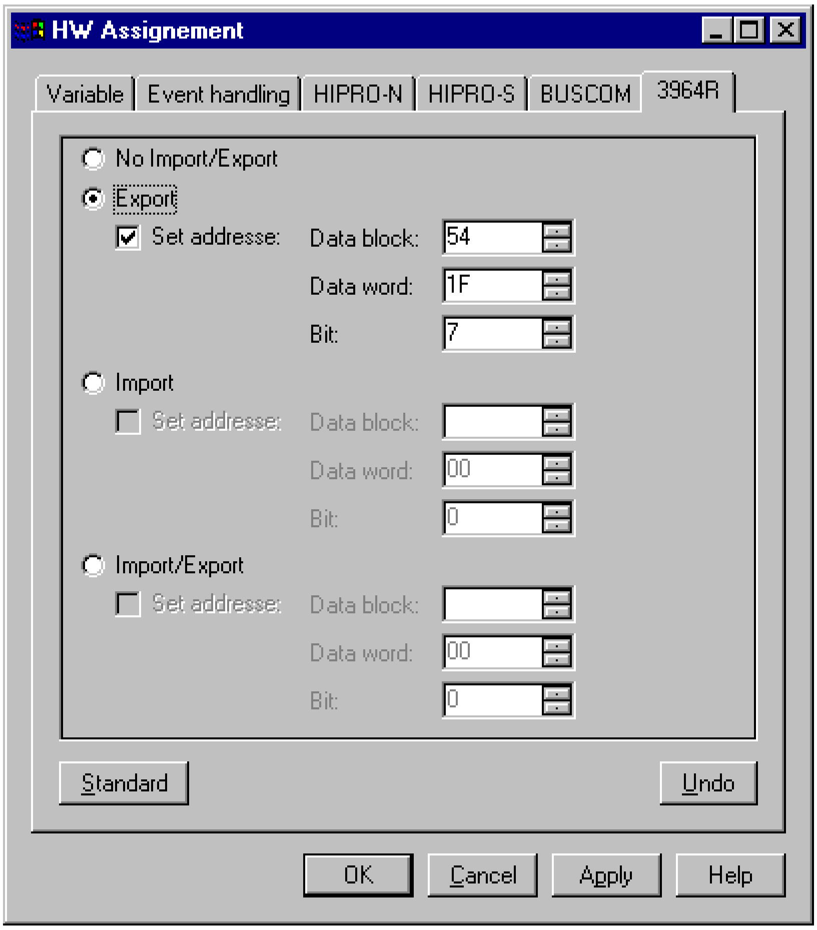
□ Set relative address: 0

Variable properties window, tab

In the BUSCOM communication tab you can define whether a variable will be exported, imported or im- and exported by the master. Selecting one of these points activates the option of defining a relative address.

Note:

If you do not use the option of defining relative addresses manually, then all BUSCOM variables will be sorted in alphanumeric order and addressed sequentially.



○ No Import/Export

● Export

☑ Set addresse: Data block: 54

Data word: 1F

Bit: 7

○ Import

□ Set addresse: Data block:

Data word: 00

Bit: 0

○ Import/Export

□ Set addresse: Data block:

Data word: 00

Bit: 0

Variable properties window, ´3964R´ tab

Manual settings for the 3964R protocol can be made here similar to the configuration of BUSCOM variables.

Communication

4 Communication between HIMA PESs

The communication system must be configured for communication with the programming unit (PDAT) or for communication between HIMA PESs and between a HIMA PES and the HIMA OPC server.

Only HIBUS is currently available as the communication system, and this requires a bus configuration (See “Creating a configuration” on page 1.)

The bus configuration contains all the PESs, PES masters and PADTs that are connected to the bus and which can therefore interchange data. The PES master in a coprocessor module controls the communication.

The BUSCOM configuration is used for communicating on the Ethernet.

4.1 Protocols

Data transfer between HIMA PESs can be both safety-related and non-safety related.

4.1.1 Safety-related transfer: HIPRO-S

The PES master only controls data traffic. Data is exchanged directly between two PESs. Safety-related communication between HIMA PESs is certified according to AK6.

4.1.2 Non-safety related transfer: HIPRO-N

The PES master acts as the data centre. A PES master reads and writes the interchanged data which is buffered in the PES master.

4.2 HIBUS Hardware

There is a range of system components available for setting up a HIBUS between HIMA PESs (terminals for DIN rail mounting, converters and standard cables). You will find data sheets and wiring options in the system catalogue for H41q/H51q.

Note:

The coprocessor module in which a PES master is loaded must be connected to the HIBUS. This will also connect the PES (slave) to the HIBUS. A separate connection is not permitted.

4.3 Redundancy

The HIBUS can be connected redundantly to the PESs and the PES master. Only one free interface is required for this in the PESs; the redundant bus connection of the PES master must be configured in the bus configuration accordingly.

The PES master performs a continuous connection check on both buses. If there is no connection to a PES on the main bus (the bus that is connected to the first interface of the PES master), then communication with that PES is routed via the reserve bus (the bus on the second interface of the PES master).

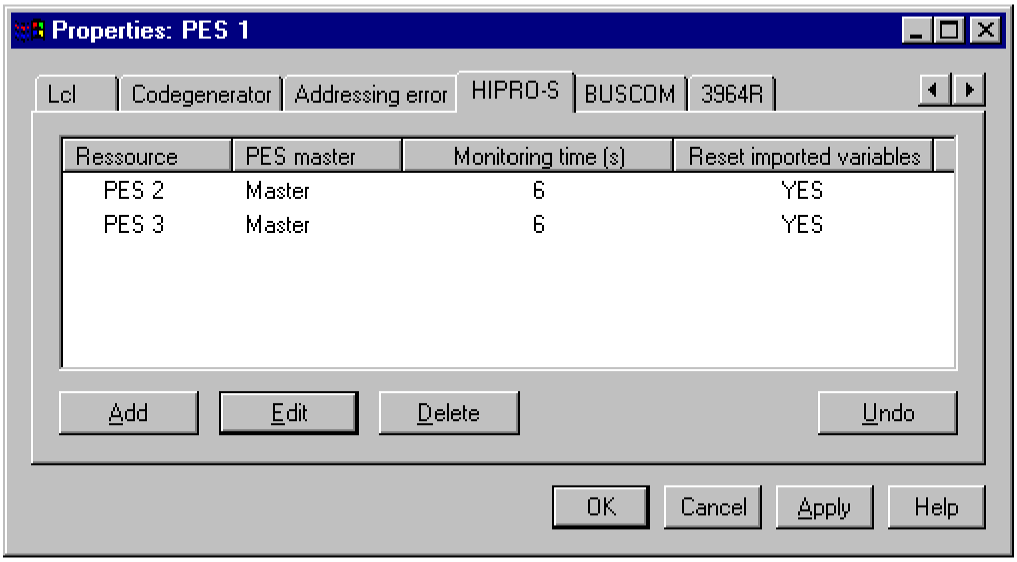
The same name must be used in the bus configuration for redundant PES masters. Full redundancy is achieved when two PES masters are used in different PESs.

4.4 Configuring data interchange

Different configurations must be made depending on the type of data transmission (safety-related or not).

4.4.1 Safety-related data transfer: HIPRO-S

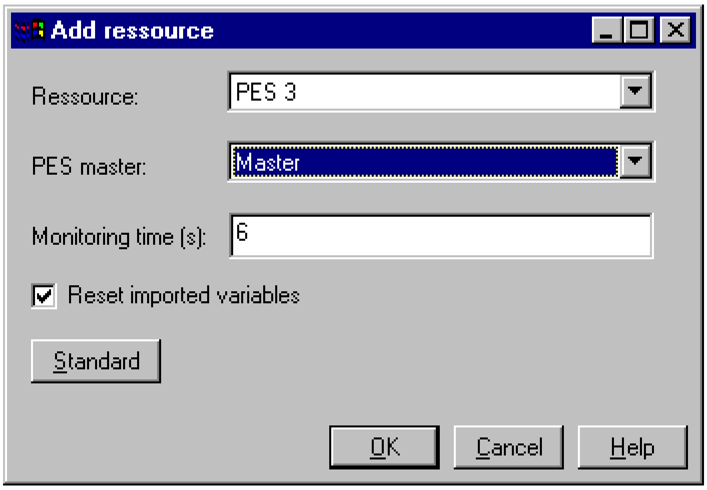
Safety-relate data transfer takes place between two PESs, so for each resource you must configure which of the PESs on the bus will be used to transfer data.



Ressource	PES master	Monitoring time (s)	Reset imported variables
PES 2	Master	6	YES
PES 3	Master	6	YES

Properties resource

In this tab you can add, edit or delete a resource. Up to 64 resources can be specified in PES H41q/H51q. This number is limited to 6 in PES H41/H51.



Ressource: PES 3

PES master: Master

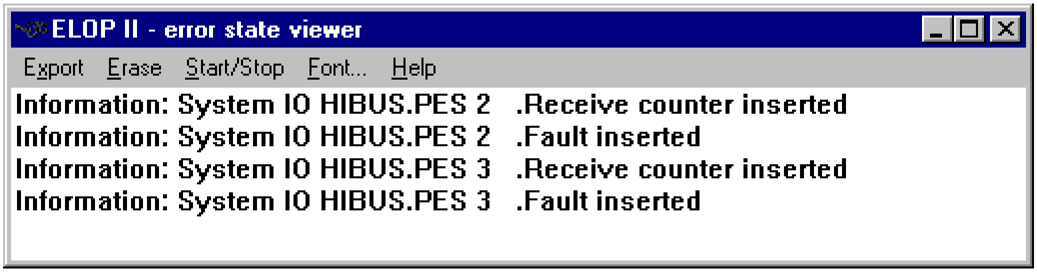
Monitoring time (s): 6

☑ Reset imported variables

Add resource

- ´Resource´
  The resources contained in the bus configuration are available here in a selection list.
- ´PES master´
  Here enter the PES master that will control data traffic to this resource. All the PES masters contained in the bus configuration are available in a selection list.
- ´Monitoring time´
  If no data are written from the resource inside this time, then they are reset.

Now activate the settings with the ´OK´ button. System variables will now be automatically generated for each resource.



```
Information: System IO HIBUS.PES 2   .Receive counter inserted
Information: System IO HIBUS.PES 2   .Fault inserted
Information: System IO HIBUS.PES 3   .Receive counter inserted
Information: System IO HIBUS.PES 3   .Fault inserted
```

Error-Status-Display after closing the properties dialog

For each resource, the data to be transmitted is configured within the program instance in the variables declaration.

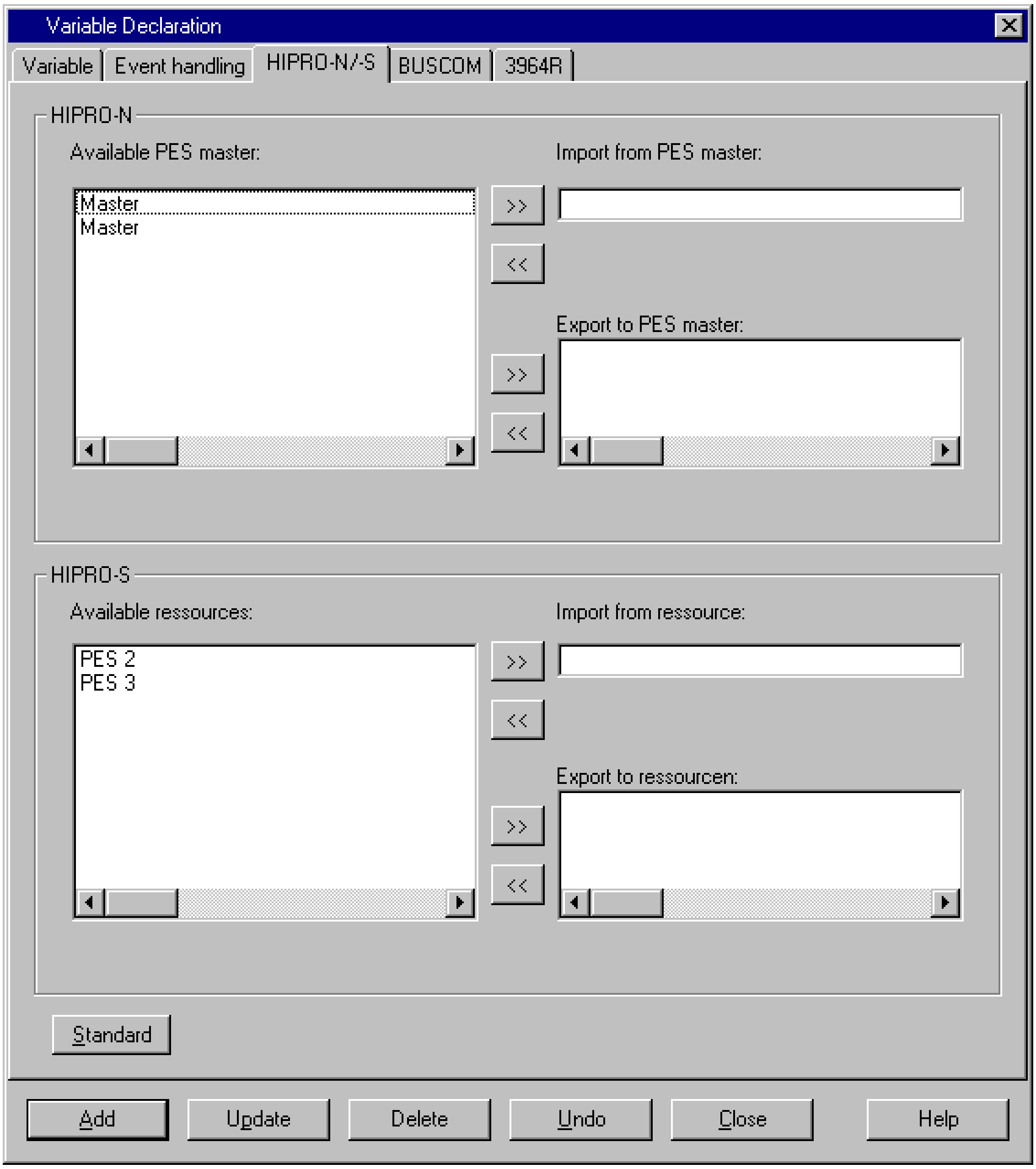
4.4.2 Non-safety related data transfer: HIPRO-N

For each resource, the data to be transmitted is configured within the program instance in the variables declaration.

4.4.3 Variables declaration

In the HIPRO tab you can specify where each variable will be imported from or exported to. The resource must be declared for safety-related data transfer and the PES master for non-safety related data transfer.

The resources and PES masters configured in the bus configuration are available for selection.



Variable declaration HIPRO

Note:

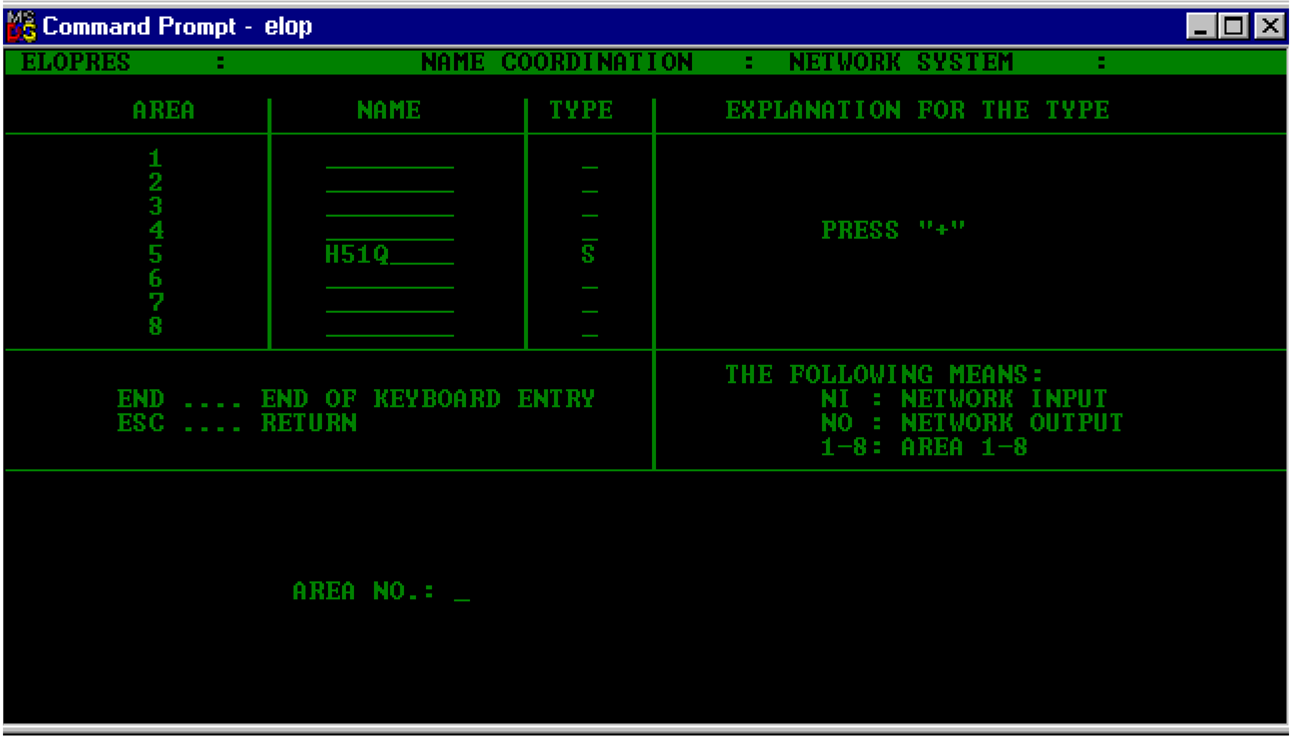
For each imported variable there must be precisely one variable that is exported. With safety-related data interchange this creates a 1:1 relationship, for non-safety related data interchange the relationship is 1:n, as an exported variable can be imported into several resources because the PES master acts as the data centre.

4.5 Configuring data interchange to ELOP slaves

It is not only possible to transfer data between PES programmed with ELOP II-NT, but also with PES programmed with ELOP (DOS based programming software from HIMA). The protocol HIPRO-S is also used for this data transfer. By this way there is no data transfer between PES programmed with ELOP.

4.5.1 Configuration in ELOP

First of all you need to add a network (area 3-8) in the project in ELOP and configure the variables (please refer to the ´ELOP manual: Programming, Monitoring, Documentation). All blocks need to be filled with 16 variables (no empty entry).

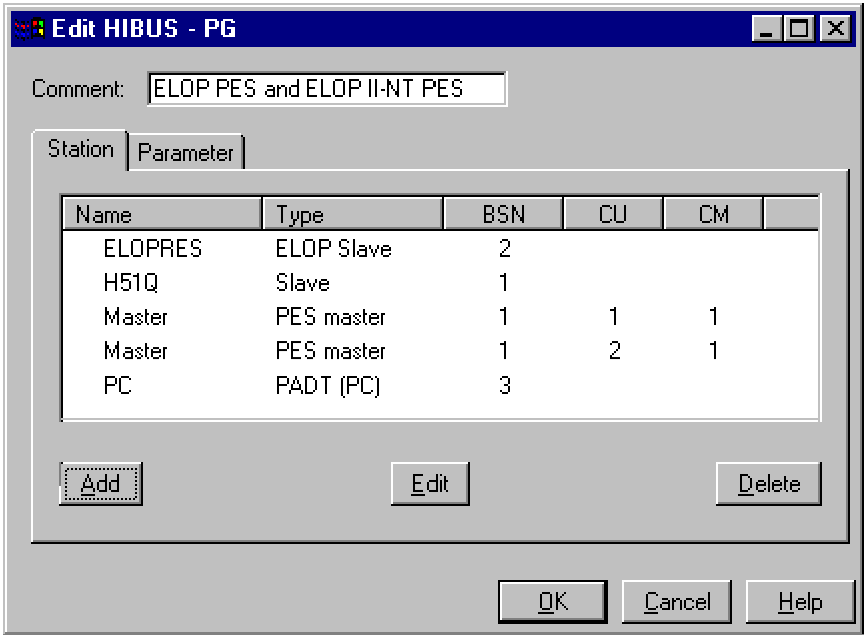


```
 ELOPRES      :            NAME COORDINATION   :  NETWORK SYSTEM     :

      AREA      |      NAME      |  TYPE  |     EXPLANATION FOR THE TYPE

       1        |   __________   |   _    |
       2        |   __________   |   _    |
       3        |   __________   |   _    |
       4        |   __________   |   _    |       PRESS "+"
       5        |   H51Q______   |   S    |
       6        |   __________   |   _    |
       7        |   __________   |   _    |
       8        |   __________   |   _    |

                                          |   THE FOLLOWING MEANS:
     END .... END OF KEYBOARD ENTRY       |         NI  : NETWORK INPUT
     ESC .... RETURN                      |         NO  : NETWORK OUTPUT
                                          |         1-8: AREA 1-8


                 AREA NO.: _
```

Name coordination: network system in ELOP

4.5.2 Bus configuration

In the bus configuration the PES programmed with ELOP need to be configured as ´ELOP Slave´ (See “HIBUS stations” on page 4.) The project name is automatically used for the resource name after configuration of the project file.



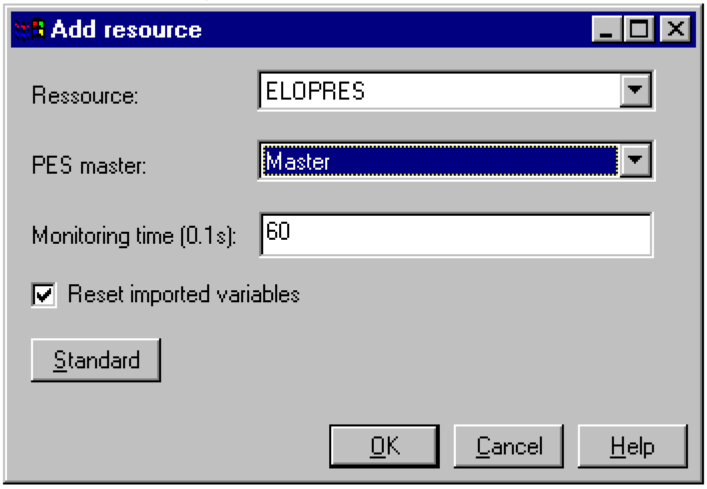
Comment: ELOP PES and ELOP II-NT PES

Name	Type	BSN	CU	CM
ELOPRES	ELOP Slave	2		
H51Q	Slave	1		
Master	PES master	1	1	1
Master	PES master	1	2	1
PC	PADT (PC)	3		

Bus configuration with ELOP slave

4.5.3 Resource configuration

The safety related data transfer is done between two PES. Therefore in the ELOP II-NT resource the PES programmed with ELOP needs to be added in the register ´HIPRO-S´.



Ressource: ELOPRES

PES master: Master

Monitoring time (0.1s): 60

☑ Reset imported variables

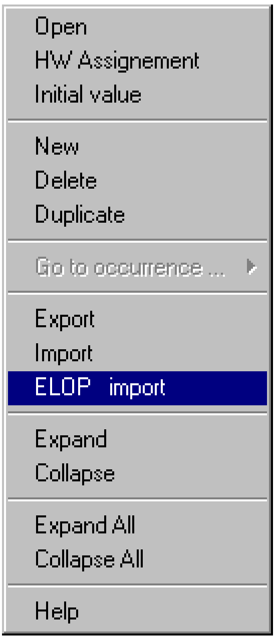
Properties resource

4.5.4 Variables declaration

The safety related transfer requires identical names in the source and target PES. Since the variable names in ELOP are restricted to 8 characters and ELOP has no import function, the variables should be configured first of all in ELOP (as described before).

The variables can then be imported in ELOP II-NT.

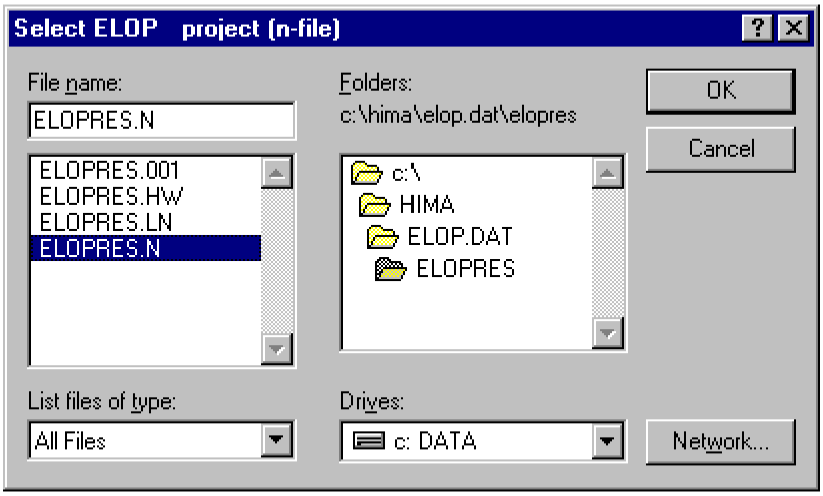
In the context menu of the variables declaration select ´ELOP Import´.



Context menu of the program instance

Next you have to select the n-file of the ELOP project. You find this file in the ELOP project directory:

[drive]:\HIMA\ELOP.DAT\[PROJECT NAME]\[PROJECT NAME].N



Select ELOP project (n-file)

File name: ELOPRES.N

Folders: c:\hima\elop.dat\elopres

List files of type: All Files

Drives: c: DATA

Select ELOP project (n-file)

The variables configured in ELOP will then be imported. The HIPRO configuration is added automatically. The binary variables in ELOP are configured as data type BOOL and the digital variables are configured as data type UINT.

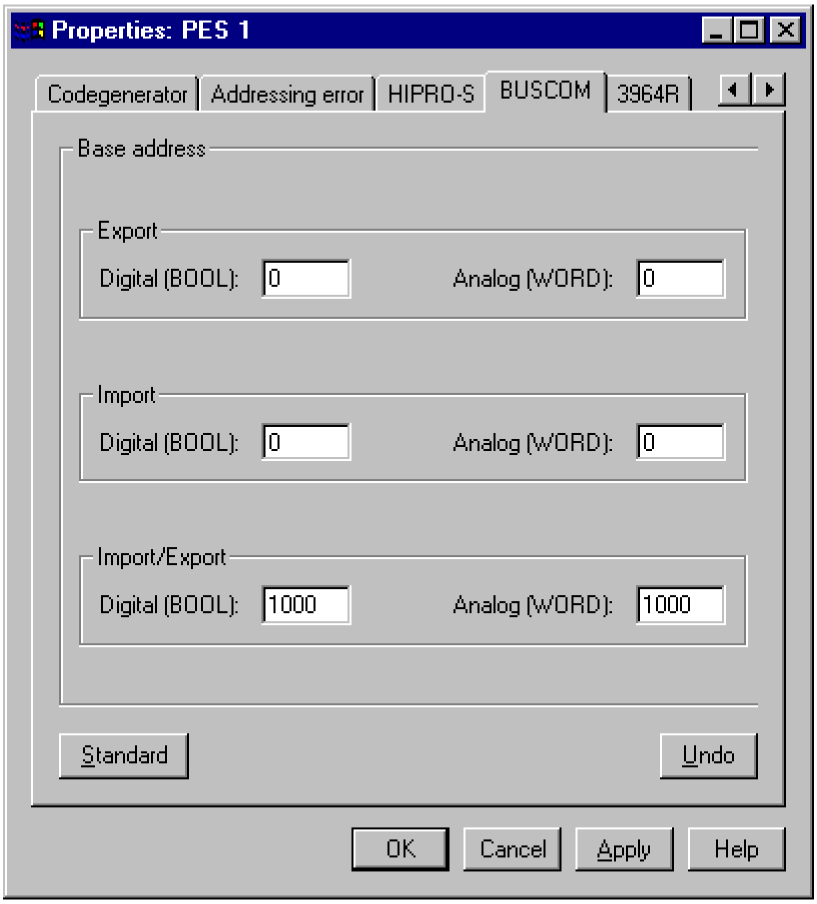
After loading the PES master the safety related data transfer is starting.

5 Communicating with external systems

Different protocols can be used to communicate with external systems. The BUSCOM definition is used for the configuration

5.1 BUSCOM Communication

The tab ´BUSCOM´ is used for the setting of base addresses for bus communication. The settings are used for MODBUS, OPC and fieldbuses.



Base address

Export
Digital (BOOL): 0
Analog (WORD): 0

Import
Digital (BOOL): 0
Analog (WORD): 0

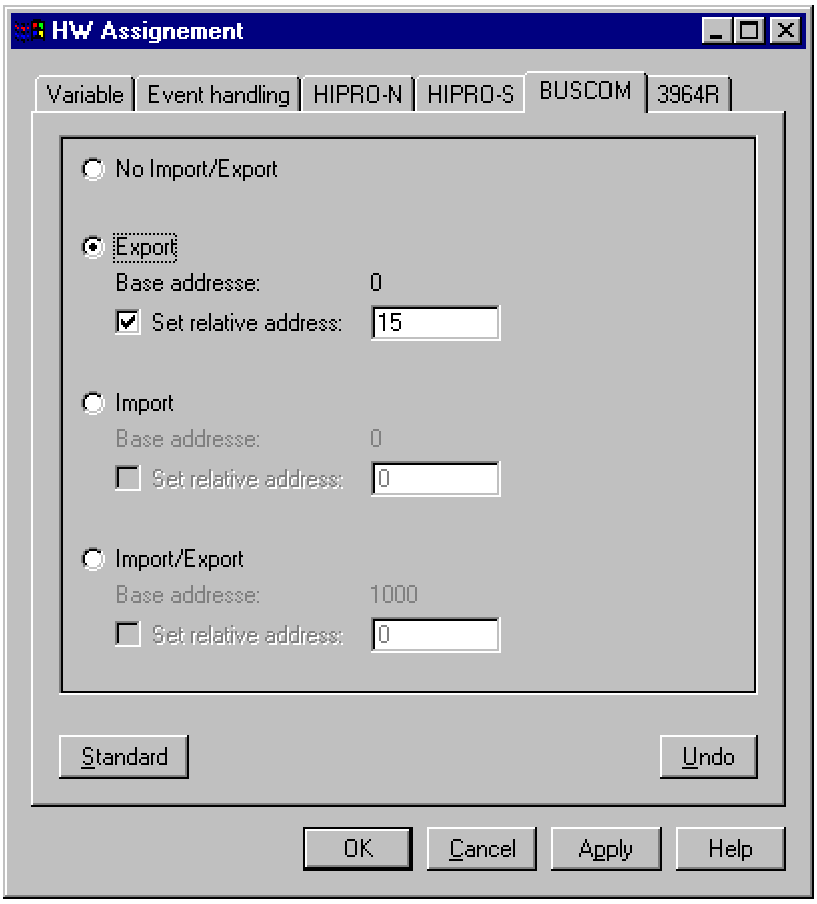
Import/Export
Digital (BOOL): 1000
Analog (WORD): 1000

Properties resource

The HIMA PES have implemented the MODBUS function codes 1, 3, 5, 6, 8, 15 and 16

- ´Export´: Digital
- Function code 1: Variable is exported to the external system
- ´Export´: Analog
  Function code 3: Variable is exported to the external system
- ´Import´: Digital
  Function code 5/15: Variable is imported from the external system
- ´Import´: Analog
  Function code 6/16: Variable is imported from the external system
- ´Export/Import´: Digital
  Function code 1 and 5/15: Variable is exported to and imported from the external system
- ´Export´: Analog
  Function code 1 and 6/16: Variable is exported to and imported from the external system

The variables that are available for data transfer are configured in the variables declaration. The relative addresses can be assigned either automatically (in alphanumeric order) or manually



○ No Import/Export

● Export

Base addresse: 0

☑ Set relative address: 15

○ Import

Base addresse: 0

☐ Set relative address: 0

○ Import/Export

Base addresse: 1000

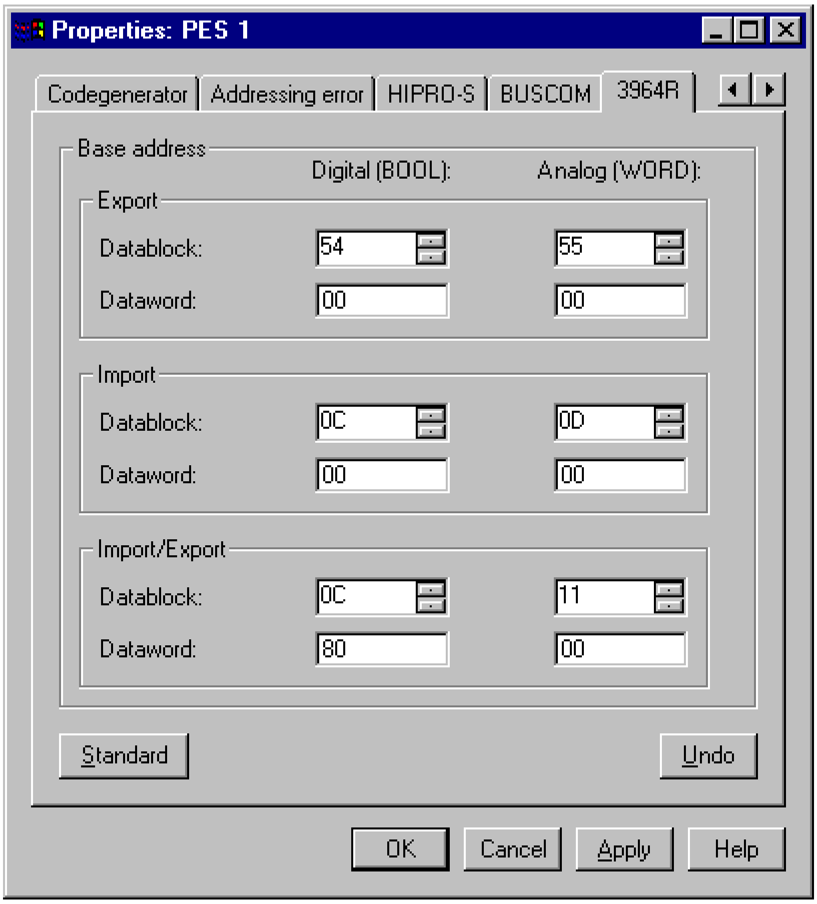
☐ Set relative address: 0

HW assignment, BUSCOM

5.2 3964R Communication

HIMA PESs support the 3964R protocol. The data blocks can be defined individually for exporting and importing

.



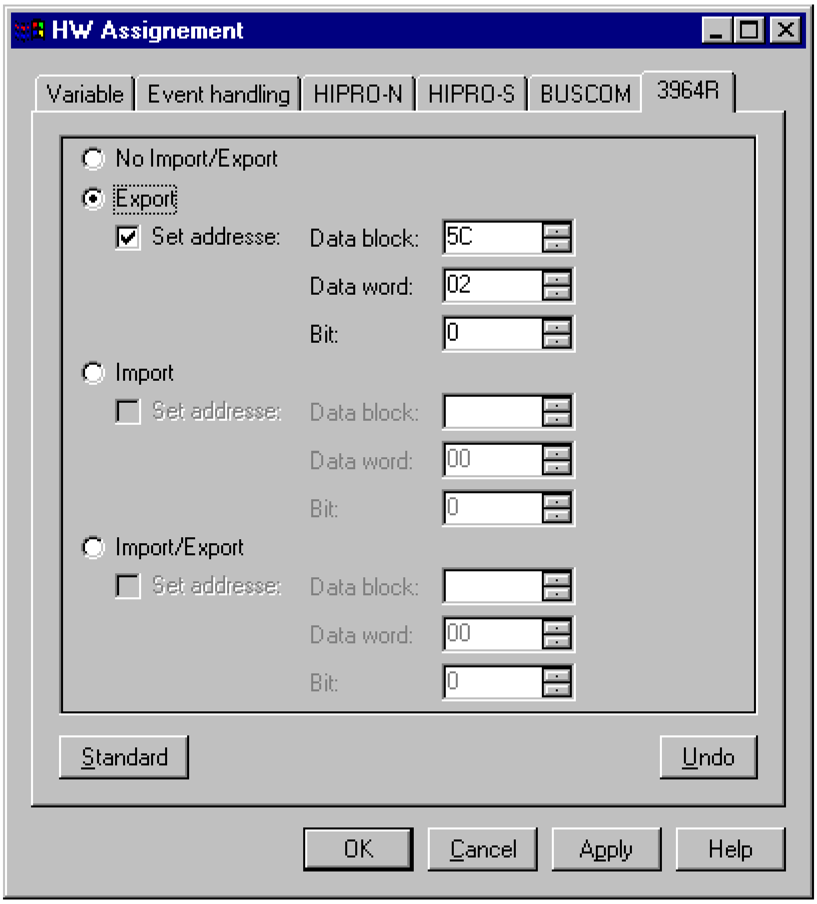
Properties: PES 1

Base address

	Digital (BOOL):	Analog (WORD):
Export		
Datablock:	54	55
Dataword:	00	00
Import		
Datablock:	0C	0D
Dataword:	00	00
Import/Export		
Datablock:	0C	11
Dataword:	80	00

Properties resource

The variables and their relative addresses are configured in the variables declaration.



○ No Import/Export

● Export

☑ Set addresse: Data block: 5C

Data word: 02

Bit: 0

○ Import

□ Set addresse: Data block:

Data word: 00

Bit: 0

○ Import/Export

□ Set addresse: Data block:

Data word: 00

Bit: 0

HW Assignment

Program Code

6 Generating the program code (compiling)

Once you have created the program and have function tested it with off-line simulation, you must generate the program code before you can load the program into the PES.

6.1 Code generator

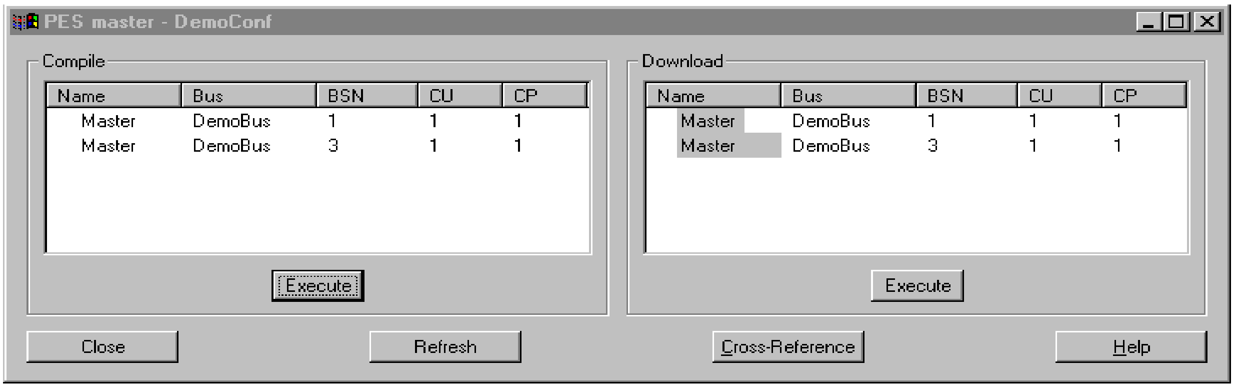
The code generator is started from the context menu of the resource in which the program will be loaded. All error and status messages generated during the compile run will be displayed in a separate window.

After the compile run, the version numbers of the project will be shown in this window.

- Code version
  Changes when the code is changed
- Program version
- Data version
  Changes only when variables are changed
- Area version
  Changes only when the I/O level is changed
- Run version
  Changes when the code and parameters are changed.

6.2 Compiling and loading PES masters

Having defined variables for data interchange and generated code for the resources, you must now generate the code for the PES masters. To do this, select ´PES masters´ [´PES-Master´] in the context menu of the configuration. You will see the defined PES masters:



Compile:

Name	Bus	BSN	CU	CP
Master	DemoBus	1	1	1
Master	DemoBus	3	1	1

Download:

Name	Bus	BSN	CU	CP
Master	DemoBus	1	1	1
Master	DemoBus	3	1	1

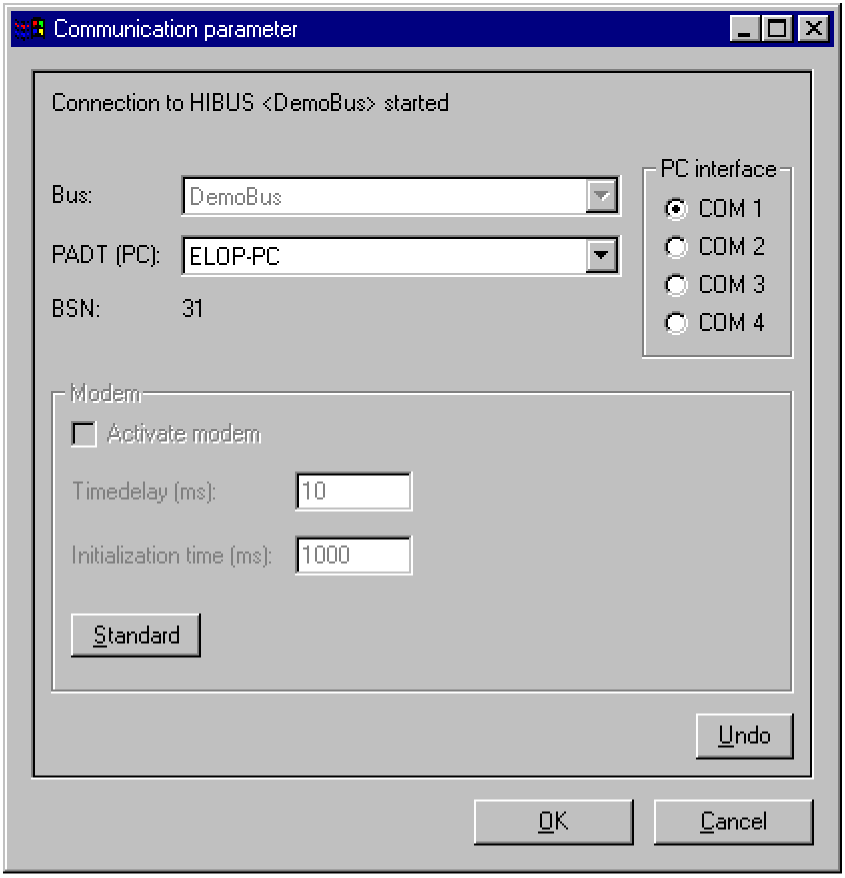
PES master Compile and Download

All the defined PES masters are listed in the ´Compile´ list box. The PES masters which still have to be compiled are highlighted in grey.

The ´Download´ list box contains all the PES masters that have already been compiled. The grey highlighted PES masters are the ones which are not currently loaded in the coprocessor module.

You can now compile and download the PES masters, update the data in the list boxes and open a cross-reference window showing the variables transferred by the PES masters.

- ´Execute´ under the Compile list box:
  Codes generated for the selected PES masters.
- ´Execute´ under the Download list box:
  Selected PES masters are loaded. A dialogue window opens to allow you to specify the communication parameters of the PADT.



Communication parameter

Connection to HIBUS <DemoBus> started

Bus: DemoBus

PADT (PC): ELOP-PC

BSN: 31

PC interface

● COM 1

○ COM 2

○ COM 3

○ COM 4

Modem

□ Activate modem

Timedelay (ms): 10

Initialization time (ms): 1000

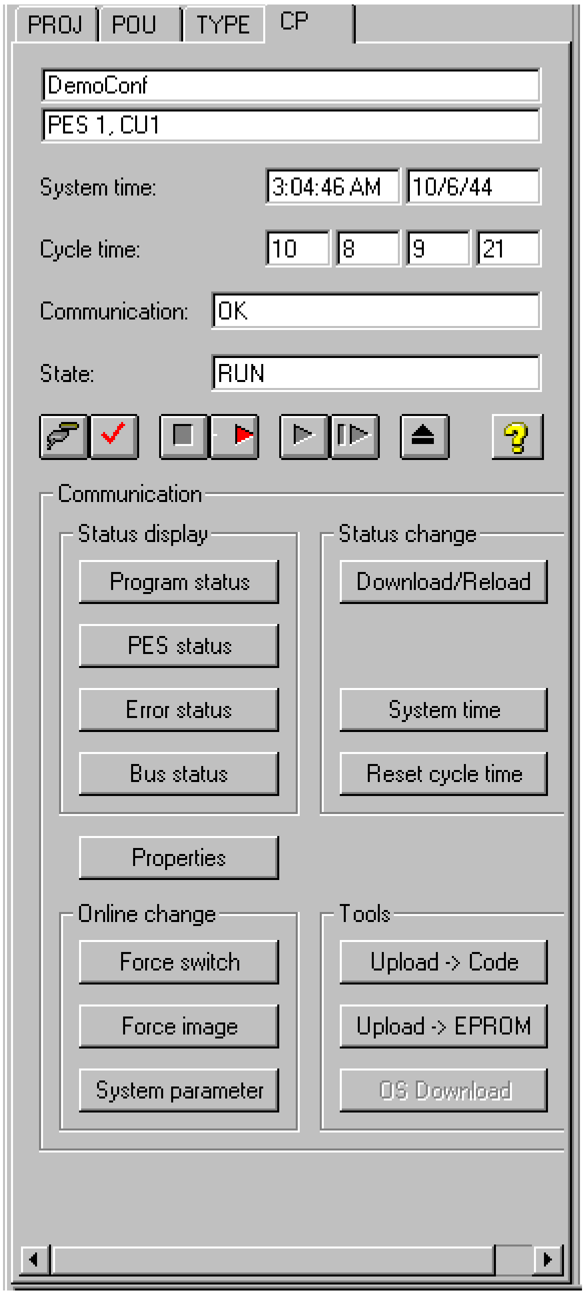
Communication parameter

Here you select the PES master you want from a list box and define the PC interface you wish to use for data transfer.

6.3 Loading the program into the controller

The Control Panel is used to load the program into the controller and to operate the PES. You start it from the context menu of the appropriate resource or you can find it in an additional project properties tab.

The name of the configuration and of the resource are shown as an overview. You will also see the system time, cycle time of the PES, the status of the communication and the status of the PES as will be displayed later in the online test.



DemoConf

PES 1, CU1

System time: 3:04:46 AM 10/6/44

Cycle time: 10 8 9 21

Communication: OK

State: RUN

Control panel

The ´Status display´ box summarizes functions for different information about the program, the PES and the bus.

The ´Status change´ box contains functions for downloading the generated program code, changing the system time of the PES and resetting the computed system time.

The ´Online change´ box shows functions for setting and resetting force switches, displaying and editing the force information for all variables and for displaying and editing the system parameters.

The ´Tools´ box contains functions for uploading the program code, creating a file for EPROM programming (for H41/H51 only) and for downloading the operating system (H41q/H51q only).

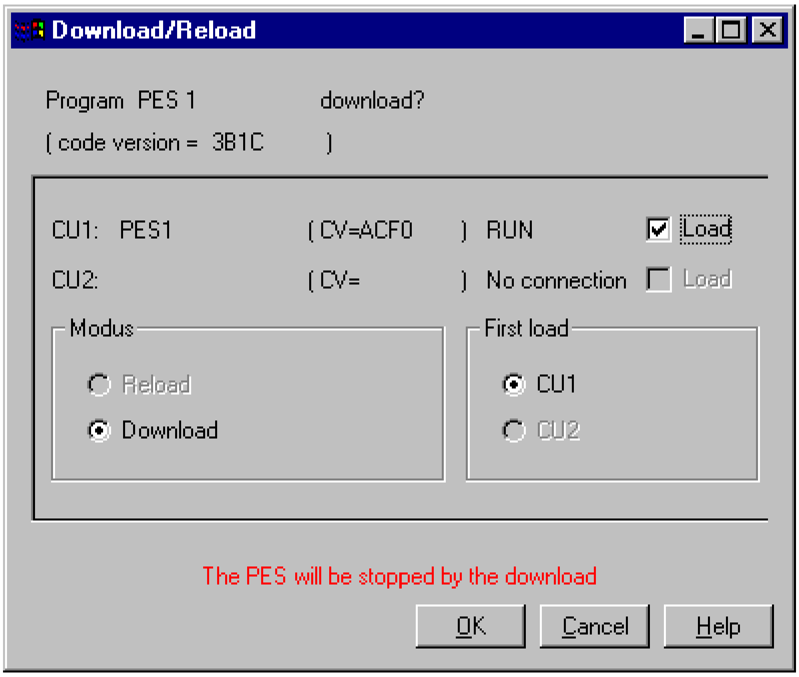
The operating buttons have the following meaning

Button	Meaning
🔌	Initializing the communication after a connection loss
✓	Acknowledge of denied actions
■	Stop the PES. Outputs can be held or reset.
▶	Start the PES. Coldstart, Warmstart or Hotstart possible
▷	Continue after a break point occurred in the PES
|▷	Cycle step of the PES (only in Stop mode). You can execute precisely one cycle for test purpose.
⏏	Quit the control panel
?	Online help for the control panel

Table 2: Buttons inside the control panel

6.4 Download/Reload

Click on the ´Download/Reload´ button to open a window in which you can select which central module will be loaded in which mode.



Download/Reload

Program PES 1 download?

(code version = 3B1C)

CU1: PES1 (CV=ACF0) RUN ☑ Load

CU2: (CV=) No connection □ Load

Modus

○ Reload

● Download

First load

● CU1

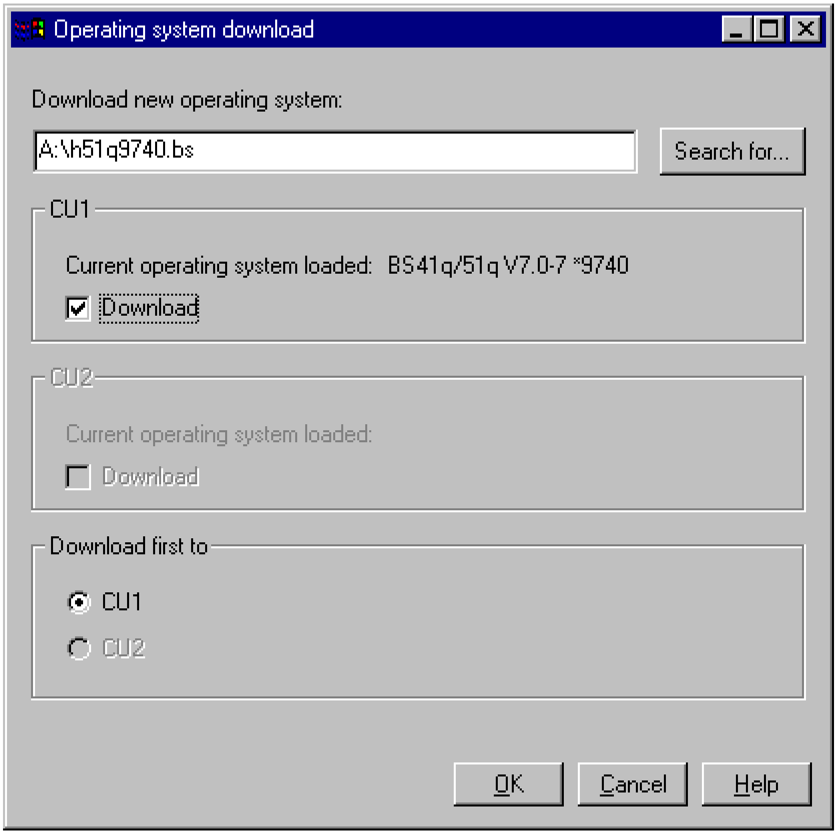
○ CU2

The PES will be stopped by the download

Download/Reload

6.4.1 OS Download

You can update the operating system in PES H41q/H51q with this menu option. You will need an updated operating system file to do this.



Operating system download

Download new operating system: A:\h51q9740.bs

CU1

Current operating system loaded: BS41q/51q V7.0-7 *9740

☑ Download

CU2

Current operating system loaded:

□ Download

Download first to

● CU1

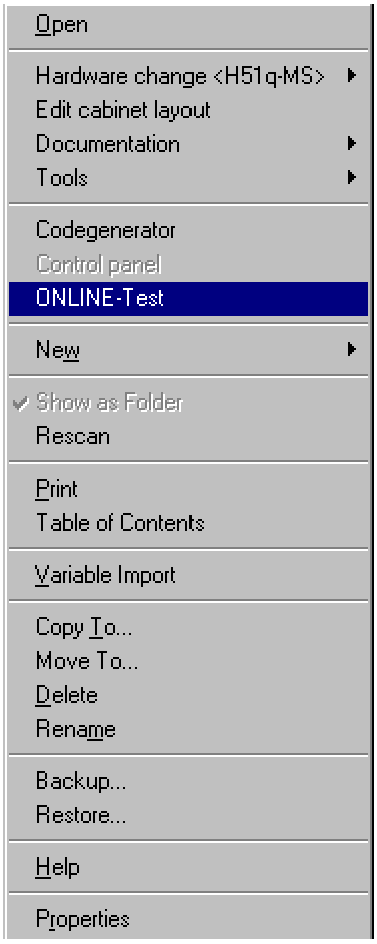
○ CU2

OS-Download, H41q and H51q

7 Online Test

7.1 Starting the online test

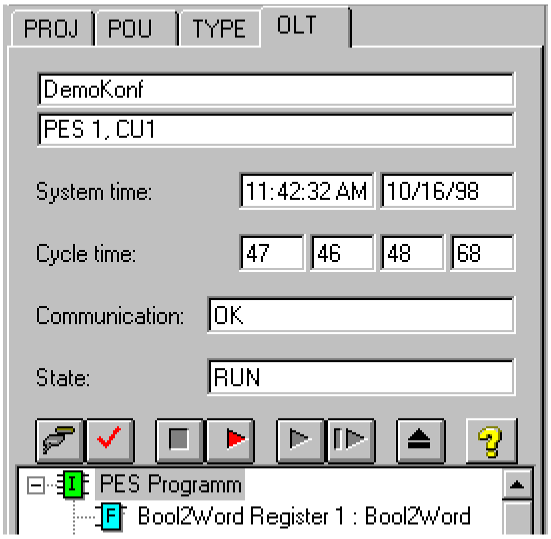
Start the online test by clicking the right-hand mouse button once on the resource in the configuration. You will now see the resource´s context menu. In the third section of the context menu, click the left-hand mouse button on ONLINE Test.



Context menu of the resource

You will now see the online test window.

7.2 Explanation of the OLT window



DemoKonf

PES 1, CU1

System time: 11:42:32 AM 10/16/98

Cycle time: 47 46 48 68

Communication: OK

State: RUN

Online test

The OLT window displays the following information

- name of the configuration

- name of the resource
- time and date of the PES
- cycle time of the PES (current, minimum, average, maximum)
- communication status (OK, no connection)
- status of the PES (see table 3)

Display	Meaning
No connection	No connection to the PES, data interchange is interrupted
Mono	PES is in RUN mode, but a central module has gone down in the redundant controller
RUN	PES is in RUN mode
PGSTOP, outputs on LOW	PES was stopped by the ELOP II-NT station and the outputs were reset
PGSTOP, outputs held	PES was stopped by the ELOP II-NT station, outputs not reset
Break point, outputs held	A break point was reached during program run, the system was stopped and the outputs not reset
ERROR STOP	PES was stopped by the operating system because of an error and all outputs have been reset

Table 3: Status

7.2.1 Operating buttons

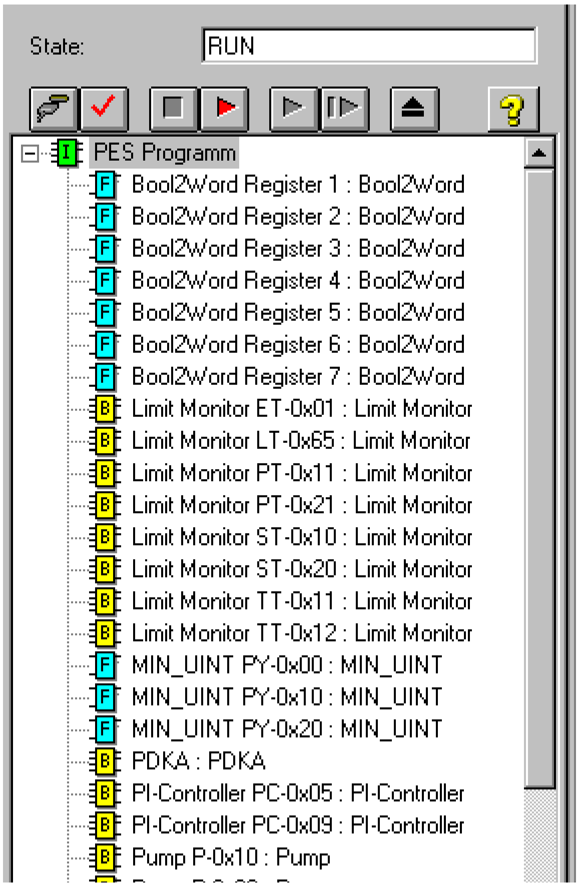
Explanation of the buttons from left to right.

Button	Meaning
🔌	Initializing the communication after a connection loss
✓	Acknowledge of denied actions
■	Stop the PES. Outputs can be held or reset.
▶	Start the PES. Coldstart, Warmstart or Hotstart possible
▷	Continue after a break point occurred in the PES
▮▷	Cycle step of the PES (only in Stop mode). You can execute precisely one cycle for test purpose.
⏏	Quit the online test
?	Online help for the online test

Table 4: Operating buttons

7.2.2 Program structure tree

The program structure tree is used to open the online test of the program or of individual blocks.



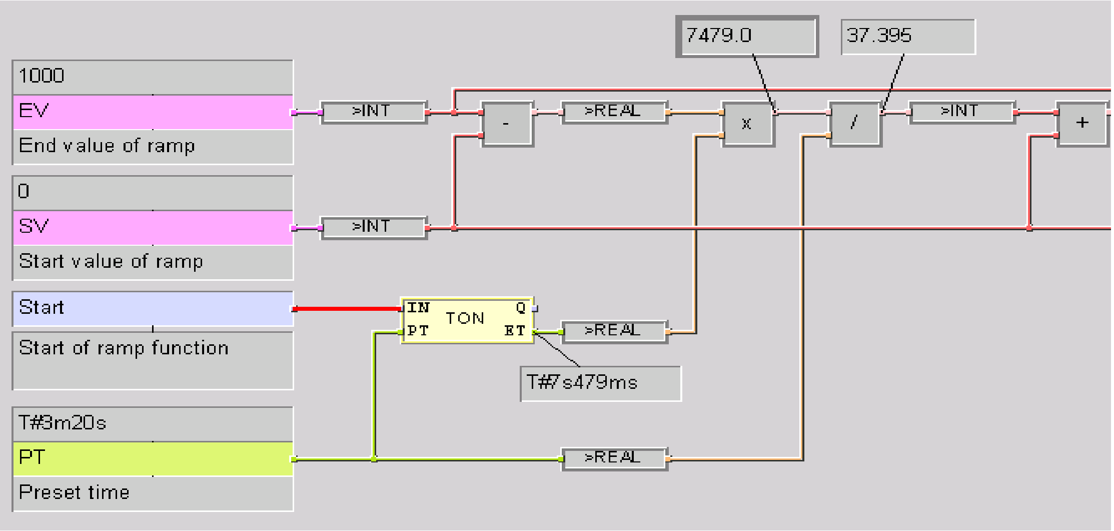
State: RUN

- I PES Programm
  - F Bool2Word Register 1 : Bool2Word
  - F Bool2Word Register 2 : Bool2Word
  - F Bool2Word Register 3 : Bool2Word
  - F Bool2Word Register 4 : Bool2Word
  - F Bool2Word Register 5 : Bool2Word
  - F Bool2Word Register 6 : Bool2Word
  - F Bool2Word Register 7 : Bool2Word
  - B Limit Monitor ET-0x01 : Limit Monitor
  - B Limit Monitor LT-0x65 : Limit Monitor
  - B Limit Monitor PT-0x11 : Limit Monitor
  - B Limit Monitor PT-0x21 : Limit Monitor
  - B Limit Monitor ST-0x10 : Limit Monitor
  - B Limit Monitor ST-0x20 : Limit Monitor
  - B Limit Monitor TT-0x11 : Limit Monitor
  - B Limit Monitor TT-0x12 : Limit Monitor
  - F MIN_UINT PY-0x00 : MIN_UINT
  - F MIN_UINT PY-0x10 : MIN_UINT
  - F MIN_UINT PY-0x20 : MIN_UINT
  - B PDKA : PDKA
  - B PI-Controller PC-0x05 : PI-Controller
  - B PI-Controller PC-0x09 : PI-Controller
  - B Pump P-0x10 : Pump

Online test

Select the program (green symbol) or the desired block (yellow symbol for a function block, blue symbol for a function) with the mouse, then double-click on the symbol.

7.3 Functions in the ONLINE test



Online test opened

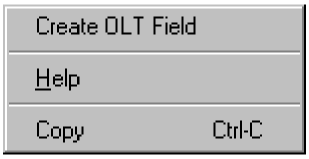
7.3.1 Meanings

The status of a BOOL type variable is indicated by the color of the interconnecting lines.

Blue means FALSE and red means TRUE.

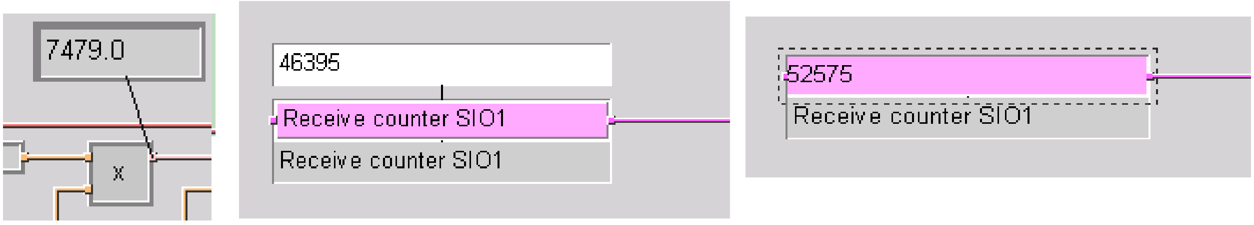
The status of a variable can also be displayed in an OLT box that can be generated. An OLT box is always needed to display values (all data types except BOOL).

You can generate OLT boxes at any node or directly on a variable. Position your mouse pointer on the desired node or variable and call the context menu by clicking the right-hand mouse button once.



Create OLT Field

Now select ´Generate OLT box´ with the left-hand mouse button. Drag the OLT box to the desired position and click the left-hand mouse button. The OLT box is now ready to display the value or status.



OLT Fields and value displays

The right display is generated by pressing ALT and left mouse button.

7.3.2 Forcing variables

In the online test you can force all variables that are defined as physical input or outputs, or local variables that are not overwritten by the program.

When an input variable is forced, then this forced value will be used throughout the logic.

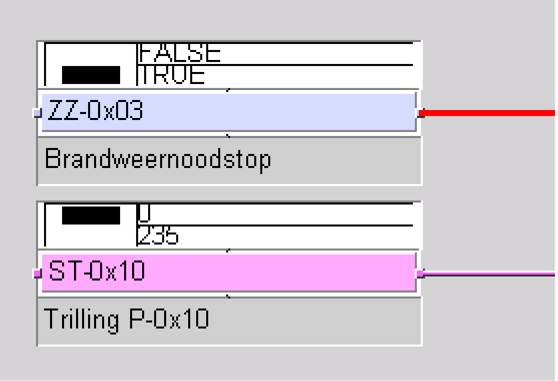
When an output variable is forced however, only the physical output is forced. If the output is interrogated in the logic, then the value defined by the logic will be used.

Forcing can be done directly in the logic with the OLT box, or it can also be done using the Force graphic in the Control Panel (CP).

7.3.2.1 Forcing IO in the OLT box

The OLT box for the IO is divided up into two columns and two lines. The left hand column contains a square. An empty square means that the force main switch is off. A filled square means that the force main switch is on. A square on the first line means the variable is not forced. A (filled) square on the second line means the variable is forced.

The value of the variable of the input or output is shown on the second column, first line.

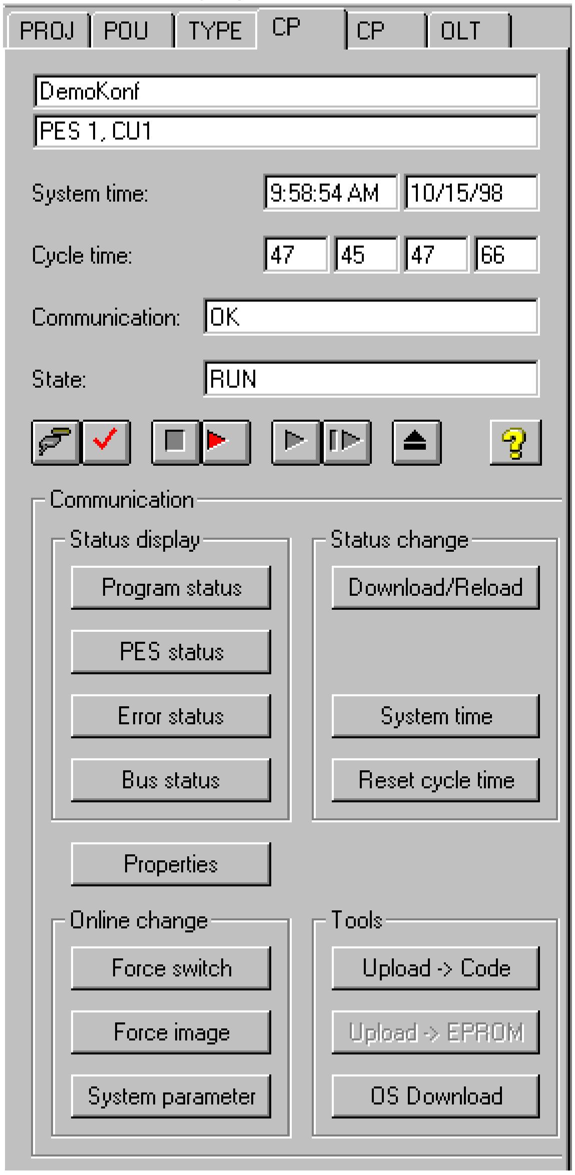


OLT field

The force value is shown on the second line.

7.3.2.2 Forcing IO in the Force graphic

Call the Force graphic with the Control Panel´s ´Force image´ button.



DemoKonf
PES 1, CU1

System time: 9:58:54 AM 10/15/98

Cycle time: 47 45 47 66

Communication: OK

State: RUN

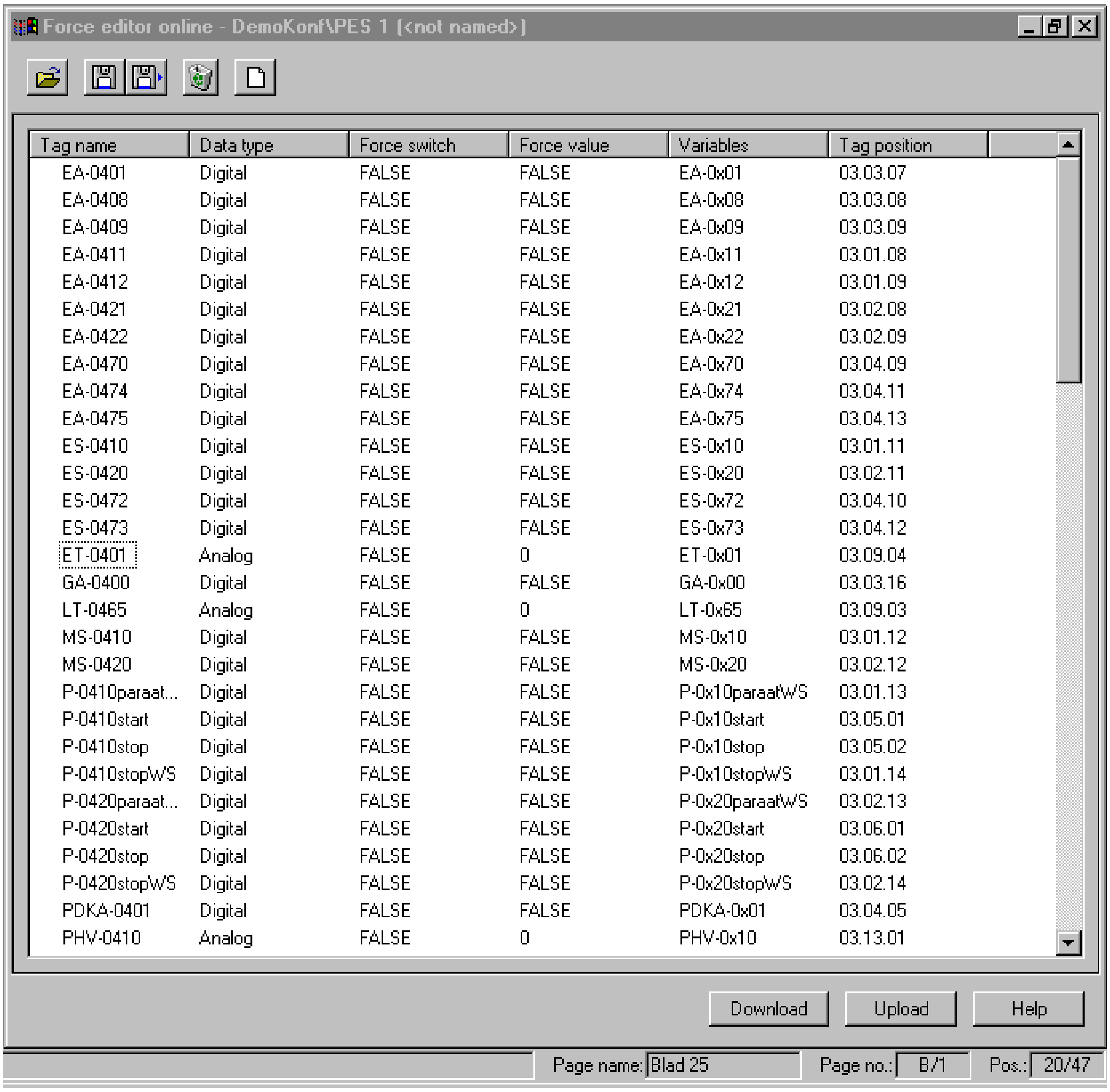
Control Panel

The Force graphic shows all variables that are combined with an I&C name. The Force graphic consists of the following columns: tag name, data type, Force switch, Force value, variables and I&C position.

You can toggle the Force switch between FALSE and TRUE simply by double clicking with the mouse.

The digital (Boolean) force values can be changed in the same way. Double click on an analog (word) force value to change it, then confirm with

Enter.



Tag name	Data type	Force switch	Force value	Variables	Tag position
EA-0401	Digital	FALSE	FALSE	EA-0x01	03.03.07
EA-0408	Digital	FALSE	FALSE	EA-0x08	03.03.08
EA-0409	Digital	FALSE	FALSE	EA-0x09	03.03.09
EA-0411	Digital	FALSE	FALSE	EA-0x11	03.01.08
EA-0412	Digital	FALSE	FALSE	EA-0x12	03.01.09
EA-0421	Digital	FALSE	FALSE	EA-0x21	03.02.08
EA-0422	Digital	FALSE	FALSE	EA-0x22	03.02.09
EA-0470	Digital	FALSE	FALSE	EA-0x70	03.04.09
EA-0474	Digital	FALSE	FALSE	EA-0x74	03.04.11
EA-0475	Digital	FALSE	FALSE	EA-0x75	03.04.13
ES-0410	Digital	FALSE	FALSE	ES-0x10	03.01.11
ES-0420	Digital	FALSE	FALSE	ES-0x20	03.02.11
ES-0472	Digital	FALSE	FALSE	ES-0x72	03.04.10
ES-0473	Digital	FALSE	FALSE	ES-0x73	03.04.12
ET-0401	Analog	FALSE	0	ET-0x01	03.09.04
GA-0400	Digital	FALSE	FALSE	GA-0x00	03.03.16
LT-0465	Analog	FALSE	0	LT-0x65	03.09.03
MS-0410	Digital	FALSE	FALSE	MS-0x10	03.01.12
MS-0420	Digital	FALSE	FALSE	MS-0x20	03.02.12
P-0410paraat...	Digital	FALSE	FALSE	P-0x10paraatWS	03.01.13
P-0410start	Digital	FALSE	FALSE	P-0x10start	03.05.01
P-0410stop	Digital	FALSE	FALSE	P-0x10stop	03.05.02
P-0410stopWS	Digital	FALSE	FALSE	P-0x10stopWS	03.01.14
P-0420paraat...	Digital	FALSE	FALSE	P-0x20paraatWS	03.02.13
P-0420start	Digital	FALSE	FALSE	P-0x20start	03.06.01
P-0420stop	Digital	FALSE	FALSE	P-0x20stop	03.06.02
P-0420stopWS	Digital	FALSE	FALSE	P-0x20stopWS	03.02.14
PDKA-0401	Digital	FALSE	FALSE	PDKA-0x01	03.04.05
PHV-0410	Analog	FALSE	0	PHV-0x10	03.13.01

Page name: Blad 25 Page no.: B/1 Pos.: 20/47

Force image

7.3.2.3 Function buttons and functions in the Force graphic

Explanations of the individual buttons are given in the table below:

Button	Meaning
📂	This button opens a Force graphic which has been previously saved.
💾	Save: when you are making changes to a Force graphic and want to save them, save the graphic with this button.
💾	Save As: use this button to save your Force graphic under a new name
🗑	Delete: this button deletes a previously saved Force graphic which is no longer needed

Button	Meaning
🗋	Reset Force values: this button is used to set all Force switches in the graphic to FALSE, the Force values of the digital (Boolean) variables to FALSE and of the analog (word) variables to 0.
Download	Download: use this button to transfer your Force graphic from the ELOP II station to the PES.
Upload	Upload: use this button to load the currently active Force graphic from the PES to the ELOP II station.
Help	Help: activate this button to call online help

Table 5: Buttons of the force image

Documentation

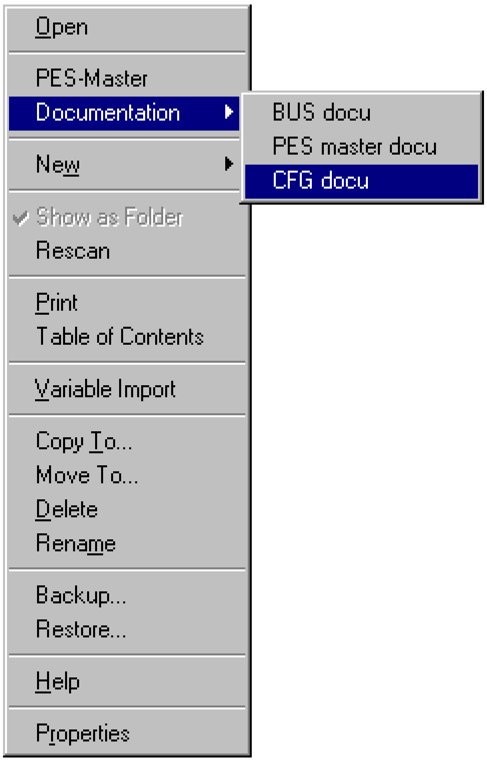
8 Documentation in the configuration

You will find in the context menu of the configuration and the resource apart from the base functions contents and print (refer to the base manual) an additional item documentation.

8.1 Documentation functions of the configuration

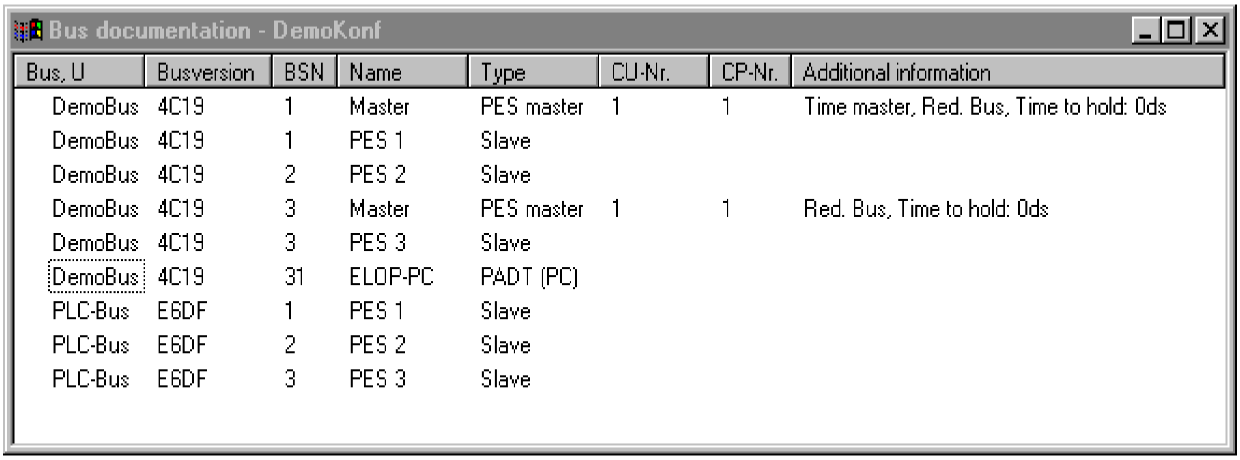
In the context menu of the configuration you will find in documentation the following items:

- BUS docu
- PES master docu
- CFG docu



Context menu of the configuration

8.1.1 Bus docu



Bus, U	Busversion	BSN	Name	Type	CU-Nr.	CP-Nr.	Additional information
DemoBus	4C19	1	Master	PES master	1	1	Time master, Red. Bus, Time to hold: 0ds
DemoBus	4C19	1	PES 1	Slave			
DemoBus	4C19	2	PES 2	Slave			
DemoBus	4C19	3	Master	PES master	1	1	Red. Bus, Time to hold: 0ds
DemoBus	4C19	3	PES 3	Slave			
DemoBus	4C19	31	ELOP-PC	PADT (PC)			
PLC-Bus	E6DF	1	PES 1	Slave			
PLC-Bus	E6DF	2	PES 2	Slave			
PLC-Bus	E6DF	3	PES 3	Slave			

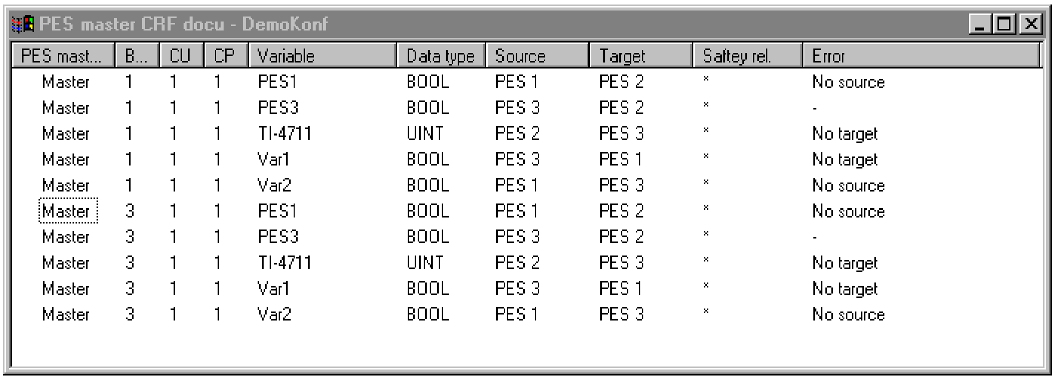
Bus documentation of the configuration

Bus Docu is an online documentation that provides information about your bus definitions. The table below gives explanations about the 7 columns that are displayed:

Column	Meaning
Bus	Name of the bus overview, the display can be in ascending or descending alphabetical order.
Bus version	The system creates a version number with your bus overview. This version number changes when you make changes to your bus overview.
BSN	Bus station number: each system that is connected to the bus is assigned a bus station number. This number can only occur once on the bus.
Name	Name of the connected system. The resource name must be used for the PES.
Type	System type. There are 3 types: Slave, PC master and PES master.
CU No.	Number (1 or 2) of the central module assigned to the PES master. Left central module = 1 and right central module = 2.
CM No.	Number of the coprocessor module used as the PES master. Slot 1, 2 or 3 next to the central module.
Additional information	Extra information.

Table 6: Key to column

8.1.2 PES Master Docu



PES master CRF docu - DemoKonf

PES mast...	B...	CU	CP	Variable	Data type	Source	Target	Saftey rel.	Error
Master	1	1	1	PES1	BOOL	PES 1	PES 2	*	No source
Master	1	1	1	PES3	BOOL	PES 3	PES 2	*	-
Master	1	1	1	TI-4711	UINT	PES 2	PES 3	*	No target
Master	1	1	1	Var1	BOOL	PES 3	PES 1	*	No target
Master	1	1	1	Var2	BOOL	PES 1	PES 3	*	No source
Master	3	1	1	PES1	BOOL	PES 1	PES 2	*	No source
Master	3	1	1	PES3	BOOL	PES 3	PES 2	*	-
Master	3	1	1	TI-4711	UINT	PES 2	PES 3	*	No target
Master	3	1	1	Var1	BOOL	PES 3	PES 1	*	No target
Master	3	1	1	Var2	BOOL	PES 1	PES 3	*	No source

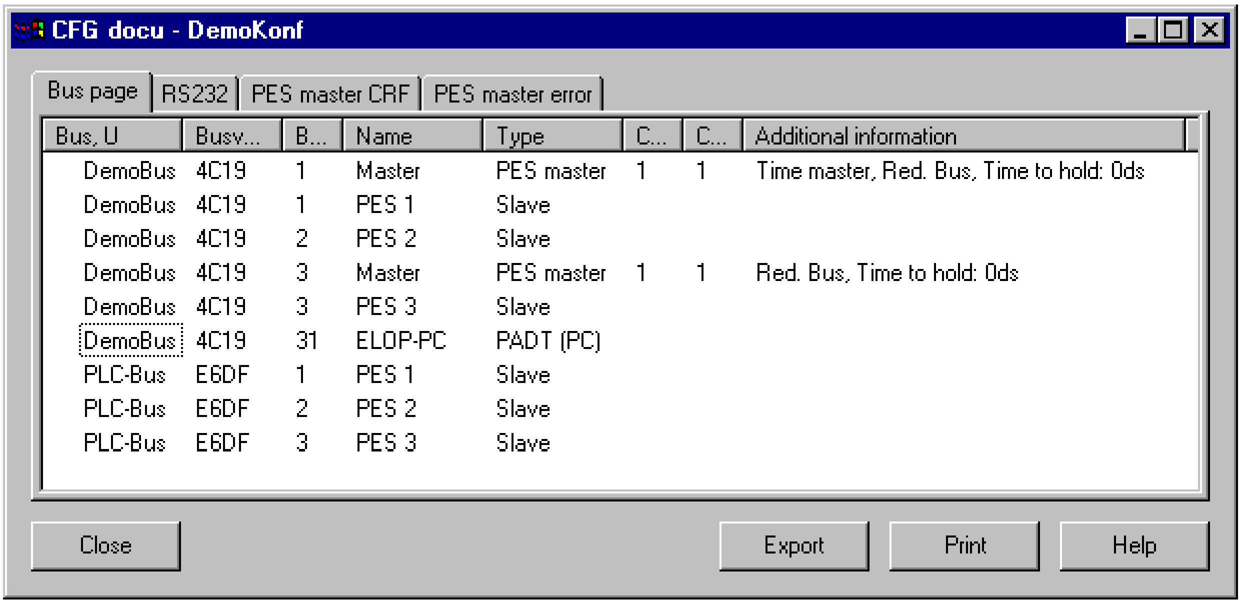
PES master documentation of the configuration

PES Master Docu is an online documentation that gives you information about your bus definitions. The table below gives explanations about the 10 columns that are displayed:

Column	Meaning
PES Master	Name of the PES master project. The display is in ascending or descending alphabetical order
BSN	Bus station number of the PES master
CU	Number of the central module assigned to the PES master. 1 = Left central module, 2 = right central module.
CM	Slot of the PES master (coprocessor module), 1, 2 or 3
Variable	Name of the variable transmitted from one HIMA PES to another HIMA PES.
Data type	Type of the variable (BOOL, WORD or UINT)
Source	Resource name of the source PES
Target	Resource name of the target PES
Safety	Flags whether variable transmission is safety related or non-safety related
Error	Error information: No source available Several sources available No target available

Table 7: Key to columns

8.1.3 CFG Docu



Bus, U	Busv...	B...	Name	Type	C...	C...	Additional information
DemoBus	4C19	1	Master	PES master	1	1	Time master, Red. Bus, Time to hold: 0ds
DemoBus	4C19	1	PES 1	Slave			
DemoBus	4C19	2	PES 2	Slave			
DemoBus	4C19	3	Master	PES master	1	1	Red. Bus, Time to hold: 0ds
DemoBus	4C19	3	PES 3	Slave			
DemoBus	4C19	31	ELOP-PC	PADT (PC)			
PLC-Bus	E6DF	1	PES 1	Slave			
PLC-Bus	E6DF	2	PES 2	Slave			
PLC-Bus	E6DF	3	PES 3	Slave			

CFG documentation of the configuration

CFG documentation consists of:

- Bus list, this bus list is the same as the bus documentation
- RS232, transfer parameters such as baud rate, stop bits, parity
- PES master CRF, same as PES Master Docu
- PES master error, error information of the PES Master Docu

In the CFG docu the following actions can be performed:

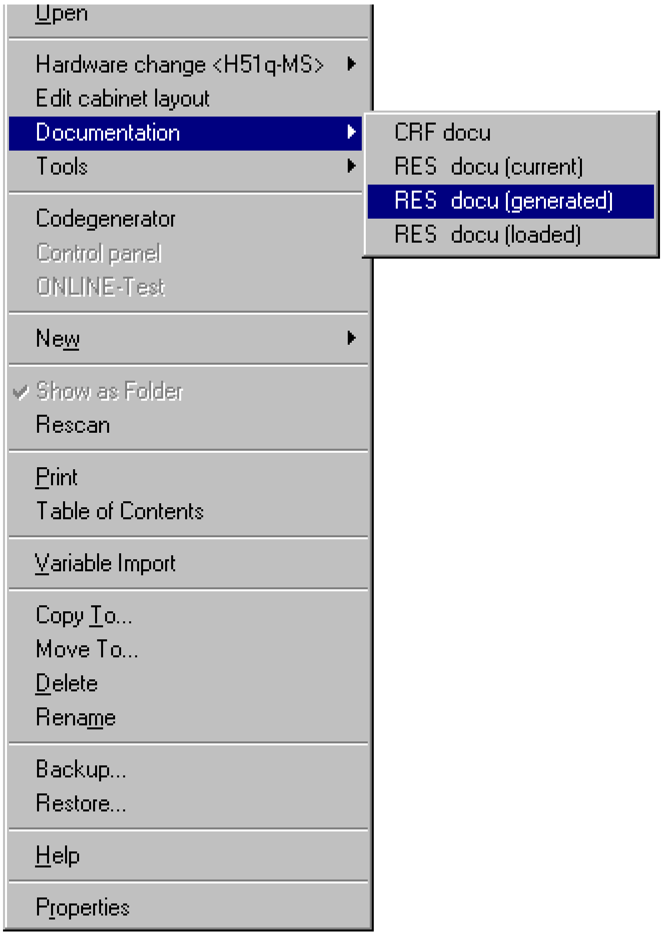
Button	Meaning
Close	Close the CFG docu
Export	Export the CFG docu in an ASCII file
Print	Print CFG docu
Help	Start help for CFG docu

Table 8: Function keys in the CFG docu

8.2 Documentation functions of the resource

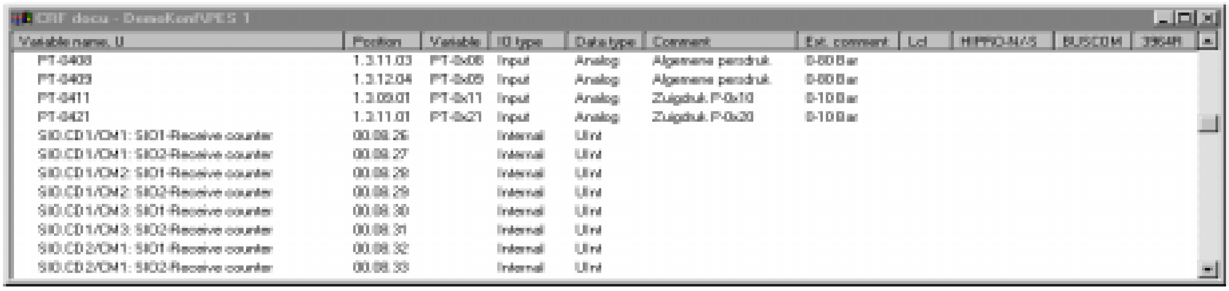
In the context menu of the resource you will find in documentation the following items:

- CRF docu
- RES docu (current)
- RES docu (generated)
- RES docu (loaded)



Context menu of the resource

8.2.1 CRF Docu


CRF docu - DemoKonf\PES 1

Variable name, U	Position	Variable	IO type	Data type	Comment	Ext. comment	Lcl	HIPRO-N/S	BUSCOM	3964R
PT-0408	1.3.11.03	PT-0x08	Input	Analog	Algemene persdruk.	0-80 Bar				
PT-0409	1.3.12.04	PT-0x09	Input	Analog	Algemene persdruk.	0-80 Bar				
PT-0411	1.3.09.01	PT-0x11	Input	Analog	Zuigdruk. P-0x10	0-10 Bar				
PT-0421	1.3.11.01	PT-0x21	Input	Analog	Zuigdruk. P-0x20	0-10 Bar				
SIO.CD 1/CM 1: SIO1-Receive counter	00.08.26		Internal	UInt						
SIO.CD 1/CM 1: SIO2-Receive counter	00.08.27		Internal	UInt						
SIO.CD 1/CM 2: SIO1-Receive counter	00.08.28		Internal	UInt						
SIO.CD 1/CM 2: SIO2-Receive counter	00.08.29		Internal	UInt						
SIO.CD 1/CM 3: SIO1-Receive counter	00.08.30		Internal	UInt						
SIO.CD 1/CM 3: SIO2-Receive counter	00.08.31		Internal	UInt						
SIO.CD 2/CM 1: SIO1-Receive counter	00.08.32		Internal	UInt						
SIO.CD 2/CM 1: SIO2-Receive counter	00.08.33		Internal	UInt						

CRF Docu

CRF Docu is an online documentation that gives you information about the inputs and outputs. It comprises 11 columns. The table below contains further information:

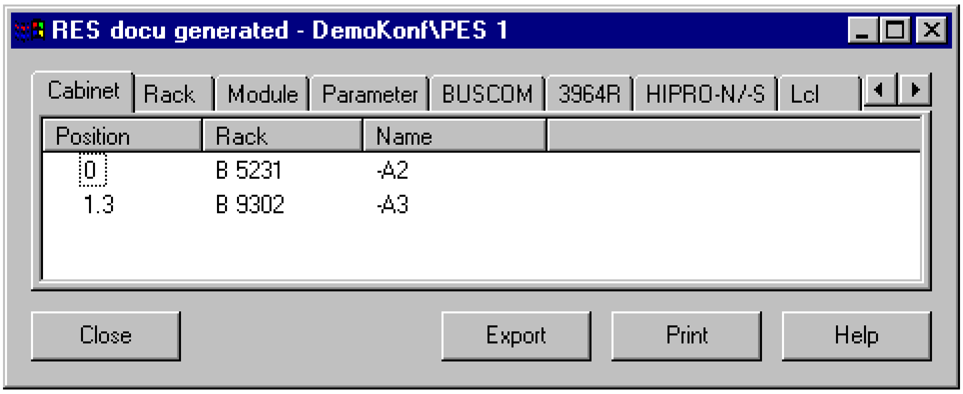
Column	Meaning
Tag name	Name of the input/output in the hardware assignment, or the name of the system variable

Table 9: Explanations of CRF Docu

Position	Position and channel of the input/output. The first digit is the cabinet and next digit is the subrack for H51 and H51q systems, followed by the slot, and finally the channel. 0 is always output for cabinet and subrack for H41, H41q, H11 and A1.
Variable	Name of the variable as defined in the program. May differ from the tag name, but is usually the same.
I/O type	This indicates whether it is an input, an output or a system variable (internal).
Data type	I/O data types can be digital (Boolean variables) and analog (UINT variables). Data types for system variables are BOOL and UINT
Commentary	Shows the long name from the variable declaration
Ext. Commentary	Shows the commentary from the variables declaration
Lcl	Flags whether I/O is used for Lcl E for Event P for Logging Lcl
HIPRO-N/-S	Flags that I/O is transmitted by HIPRO -/E HIPRO-N Export to PES master -/I HIPRO-N Import from PES master IS/- HIPRO-S Import from PES master ES/- HIPRO-S Export to PES master
BUSCOM	Flags that I/O is transmitted via BUSCOM R master reads variable W master writes variable
3964R	Flags that I/O is transmitted by 3964 R

Table 9: Explanations of CRF Docu

8.2.2 RES Docu



Position	Rack	Name
0	B 5231	-A2
1.3	B 9302	-A3

RES Docu

RES Docu consists of:

- Cabinet
  the documentation of the cabinet contains the cabinet number and the subrack number of the used subrack, the short name (column rack) and the name (input in edit cabinet)

- Rack
  the documentation of the rack contains the cabinet number, the subrack number, the position of the used subracks and modules, the short names and names.
- Module
  the documentation of the module contains all subrack, modules and channels.
- Parameters (system parameters)
  the documentation of parameters contains all settings of the system parameters
- BUSCOM
  the documentation of BUSCOM contains all variables which are transferred via a bus system, currently MODBUS and the addresses.
- 3964R
  the documentation of 3964R contains all variables which are transferred via a bus system with the protocol 3964 R and the addresses.
- HIPRO-N/-S
  the documentation of HIPRO contains all variables which are transferred via HIPRO with the concerning PES master
- Lcl
  the documentation of Lcl contains all variables which are printed via Lcl with the text for status TRUE and FALSE.
- CRF (tag name, position, variable name, I/O type, data type, commentary, extra commentary, Lcl, HIPRO, BUSCOM, 3964R)

You can perform the following actions in RES Docu:

Button	Meaning
Close	Close the RES Docu
Export	Export the RES Docu to an ASCII document
Print	Print off RES Docu
Help	Call online help

Table 10: Function buttons in RES Docu

RES Docu displays your current project status. This may well be different from RES Docu (generated) or RES Docu (loaded).

8.2.3 RES Docu (generated)

RES Docu (generated) is based on the resource data with which the code was generated.

CG Error is output in addition to RES Docu. CG Error is the same as the

“Program.ERR” file. This file contains the code generator messages.

All other points are the same as for RES Docu.

8.2.4 RES Docu (loaded)

RES Docu (loaded) is the same as RES Docu, but is based on the resource data that was loaded into the PES.

Safety aspects

9 Safety related functions

ELOP II-NT has safety related functions, which guarantee for the system family H41q/H51q that

- the programming system (PS) works correctly, i.e. no error occurs on the programming system
- the PS is used correctly, which means that user error are excluded

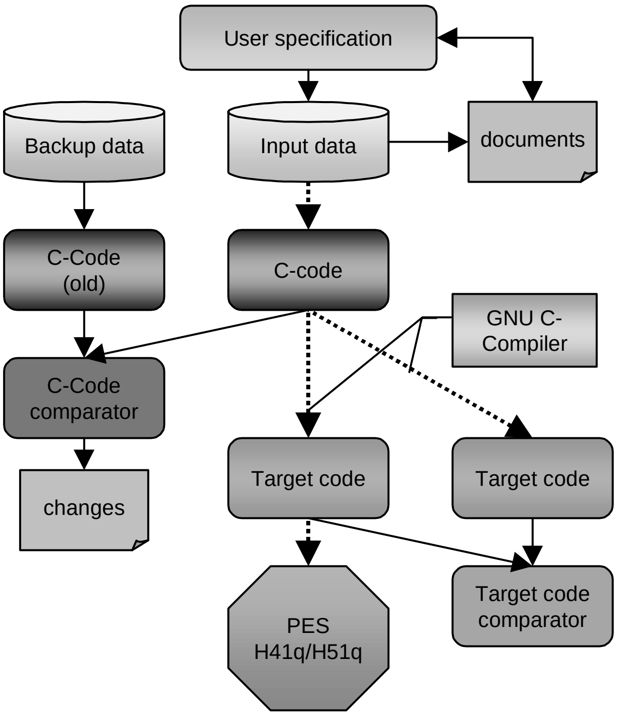
During the first start up of a safety related PES (programmable electronic system) a complete function test is necessary to proof the safety of the system

After modification of an application it is only necessary to perform a function test for the modifications. A complete function test is **not** required.

The important functions for this in ELOP II-NT are:

- C-code comparator
- target code comparator
- proven GNU-C-compiler

The C-code comparator detects changes in the application program. The target code comparator compares two target codes, which have been generated by the proven GNU-C-compiler in two different tasks. By this errors of the non safety related PC can be detected.



Principle scheme for the code generation.

9.1 Building an application

In this chapter you will find a basic overview on the procedure to build an application. Please also refer to the safety manual for the system family H41q/H51q.

9.1.1 Create the logic diagrams

The program is built based on a specification for the application, which should be available as specification or function requirements. First the basic functions are programmed in ELOP II-NT using POU’s function and function block. Using these basic function you then built more complex functions which are used to build up the program. Application parts which function do not depend on each other have to be separated in different functions or function blocks.

The main targets building your application should be:

- Easy to understand
- Easy to follow up
- Easy to modify

ELOP II-NT provides several non safety related functions such as version management, revision control in the documentation object and off-line simulation to assist in the design of logic diagrams. After a modification the user name (=user in Windows NT) and the date are stored automatically (see also ´revision service´).

9.1.2 Code generation

After the program is assigned to the resource, you should check the settings for the code generator (see “Code generator” on page 25.). To avoid random errors of the not safety related PC, please activate the target code comparator. To allow the safety related determination of future modifications please select to generate a code compare image.

Of course you should check also the other settings of the resource especially the safety parameter (see “Safety” on page 21.)

9.1.3 Download

For the download it is important that you select the bus configuration which matches the real bus connection. By this a download to the wrong PES is avoided, due to the bus station number.

9.1.4 Function check of the PES

After downloading the program to the resource a complete function test must be performed in order to check the correct design according to the original specification. Again it is helpful to check the base function first and afterwards the more complex functions.

The function test should be carried out by a person which was not involved in the program design. The check can be documented by selecting ´Is checked´ in the context menu of the program instance for each page. The user name (=user in Windows NT) and the date are stored automatically (see also ´revision service´).

Do not forget to backup the data.

9.2 Modification of an application

In this chapter you will find a basic overview on the procedure to modify an application. Please also refer to the safety manual for the system family H41q/H51q.

9.2.1 Modification of logic diagrams

The program is modified according to specification for the application, which should be available as specification or function requirements. You have to follow the basic rules on how to build a logic, especially the separation of application parts which function do not depend on each other in different functions or function blocks.

The check should be carried out by a person which was not involved in the program modification. The modification and the check rest. the release must be documented. ELOP II-NT provides the non safety related functions revision control in the documentation object and revision service for that purpose.

9.2.2 Code generation

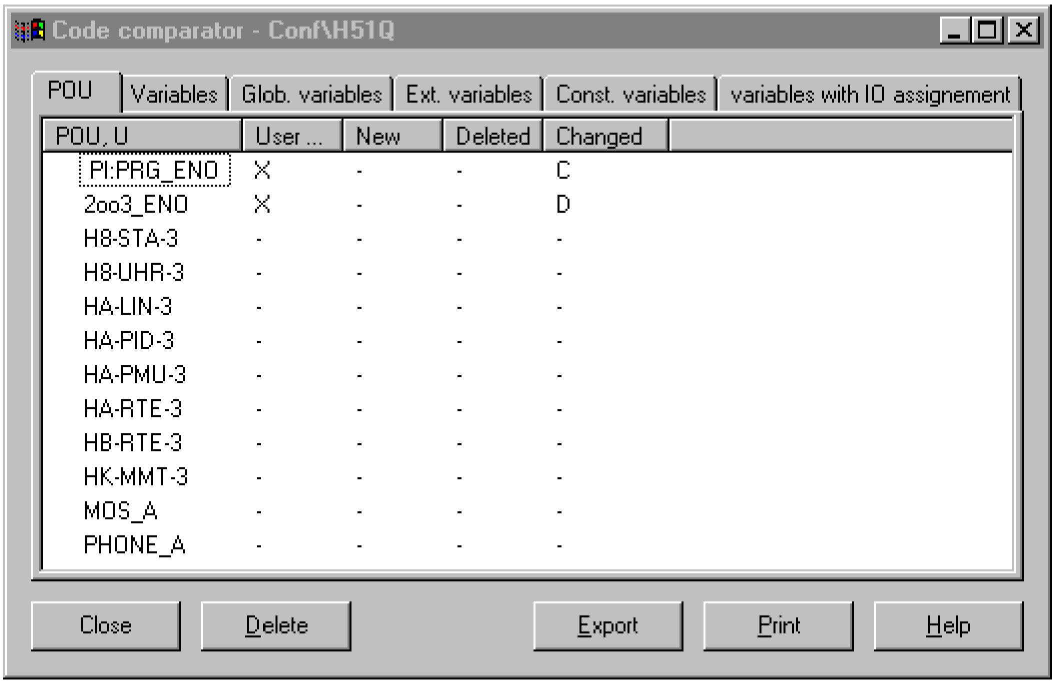
In the resource properties you need to select a resource for code image. This can be either a backup of the project on another drive or exactly the same resource. With this setting the system will determine safety related the changes between the current and the code image.

Note:

In order to be able to perform the safety related comparison, it is necessary to generate a code image before during code generation (see “Code generator” on page 25.).

The modifications are displayed per POU.

.



POU, U	User ...	New	Deleted	Changed
PI:PRG_ENO	X	-	-	C
2oo3_ENO	X	-	-	D
H8-STA-3	-	-	-	-
H8-UHR-3	-	-	-	-
HA-LIN-3	-	-	-	-
HA-PID-3	-	-	-	-
HA-PMU-3	-	-	-	-
HA-RTE-3	-	-	-	-
HB-RTE-3	-	-	-	-
HK-MMT-3	-	-	-	-
MOS_A	-	-	-	-
PHONE_A	-	-	-	-

Output of the code comparator

All POU’s are listed and the POE designed by the user are clearly marked. In ´changed´ you will find the modifications per POU (C: code has been modified; D: Data has been modified, e.g. modification of variables). Modifications are also shown for variables, global variables, external variables, constant variables and variables with IO assignment are listed in separate tabs. To display all variables please select ´Expand all´ in the context menu.

9.2.3 Download

For the download it is important that you select the bus configuration which matches the real bus connection. By this a download to the wrong PES is avoided, due to the bus station number.

9.2.4 Function check of the PES

After downloading the program to the resource the modifications need to be checked in order to prove a correct design.

The function test should be carried out by a person which was not involved in the program modification. The check can be documented by selecting ´Is checked´ in the context menu of the program instance for each page.

Do not forget to backup the data.

Blocks

10 HIMA Standard Blocks

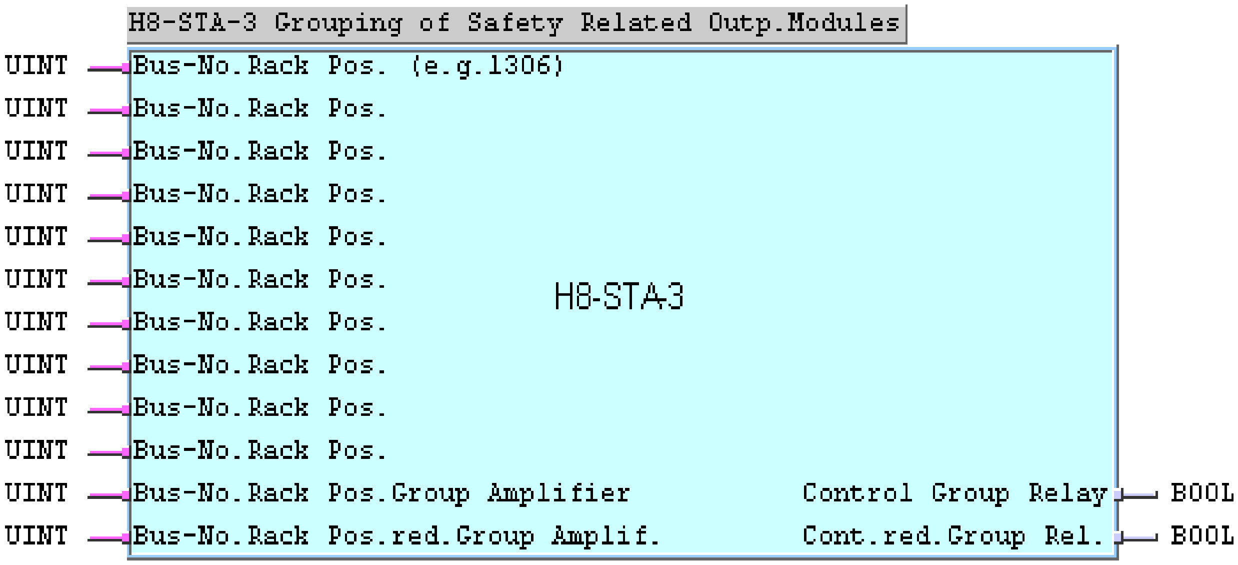
HIMA standard blocks are the blocks that are contained in ELOP-LIB. They are:

- H8-STA-3, grouping of safety-relevant testable outputs
- H8-UHR-3, date and time
- HA-LIN-3, temperature linearization
- HA-PID-3, PID controllers
- HA-PMU-3, configurable transducers
- HA-RTE-3, watchdog for analog (word) testable input modules F 6213 and F 6214
- HB-BLD-3, modules and line diagnosis of testable outputs
- HB-BLD-4, modules and line diagnosis of testable redundant outputs
- HB-RTE-3, watchdog for testable input modules F 3235, F 3237 and F 3238
- HK-AGM-3, H51, H51q PES master supervision
- HK-LGP-3, Lcl evaluation and configuring
- HK-MMT-3, MODBUS master with telephone modem
- HZ-DOS-3, diagnosis without safety
- HZ-FAN-3, error display of testable I/O modules

Please refer to the following block data sheets for further particulars.

11H8-STA-3

Grouping of safety relevant output modules



Input	Type
Bus No. BT Pos.	UINT
Bus No. BT Pos. group amp.	UINT
Bus No. BT Pos. group amp.	UINT
Outputs	
Trigger group rel.	BOOL
Trigger red. group rel.	BOOL
Function information	Short information
Block information	General block information
General	Description of block function

table 11: Input and output specification

11.1 Function information

Grouping of digital (boolean) testable output groups for group shutdown

11.2 General block information

Valid from operating system	BS41/51 V6.0-6 (9636) BS51-M, H, HR V6.0-6 (9636) BS41q/51q V7.0-7 (9737)
Special features	Block may only be used in the program type
Reload (load program), change	Not permitted
Use in safety controllers with TÜV test	Permitted
Assignment rule	None

table 12: General block information

11.3 GENERAL

The block is used to configure grouping within the safety-oriented / high-availability PES types H41, H51, H41q/ H51q.

One H8-STA-3 block is used per shutdown group.

11.3.1 Block assignment

11.3.1.1 Modules without integrated safety shutdown

The associated positions of the testable output and group amplifier modules and the names of the group amplifier channels are output for each group.

11.3.1.2 Modules with integrated safety shutdown

As well as specifying the positions of the testable output modules with integral safety shutdown, one of the **Bus No. BT Pos. group amp**. and **Bus No. BT Pos. red. group amp.** inputs must be assigned. One of the inputs must be assigned a dummy position of a group amplifier. This dummy position may not exist in the system, i.e. you must use a subrack and a module position which does not exist but which is possible.

11.3.1.3 Operation

In systems with group amplifiers, when the system is activated the group amplifier channels are triggered first and then the logic signals of the safety-relevant output amplifiers are turned off 0.3 s later (in the next cycle).
If a safety-relevant output amplifier fails, the associated group amplifiers are reset, and this resets all the modules in the group.
For modules with integrated safety shutdown, the system resets the asso-

ciated modules with the integrated safety shutdown.
A general shutdown (emergency stop) is initiated if no group amplifier is assigned. A general shutdown is also initiated if a group amplifier and an output amplifier fail together (systems without integrated safety shutdown only).
When the faulty output amplifier has been replaced the system is activated during operation by pressing the reset button.

Please refer to the Safety Manual for possible applications of the group shutdown and its implementation.

11.4 Notes on assigning inputs/outputs

11.4.1 Bus-No. Rack Pos.

Type: UINT
The positions of the testable output amplifiers with integrated safety shutdown are shown as 4-digit decimal numbers according to the position of the board in the CABINET program section.

Example:	Cabinet (1- 2):	1
	Subrack (1- 8):	3
	Board position (1-16):	6

Position of the unit: 1306

Up to 10 output modules can be combined to form a group.

11.4.2 Bus-No. Rack Pos. Group Amplifier

Type: UINT
The position of the group amplifiers is shown as a 4-digit decimal number according to the position of the board in the CABINET program section (see **Bus-No. Rack Pos.**).

11.4.3 Bus-No. Rack Pos. red. group amp.

Type: UINT
see input **Bus No. Rack Pos. group Amplifier**

11.4.4 Control Group Relay

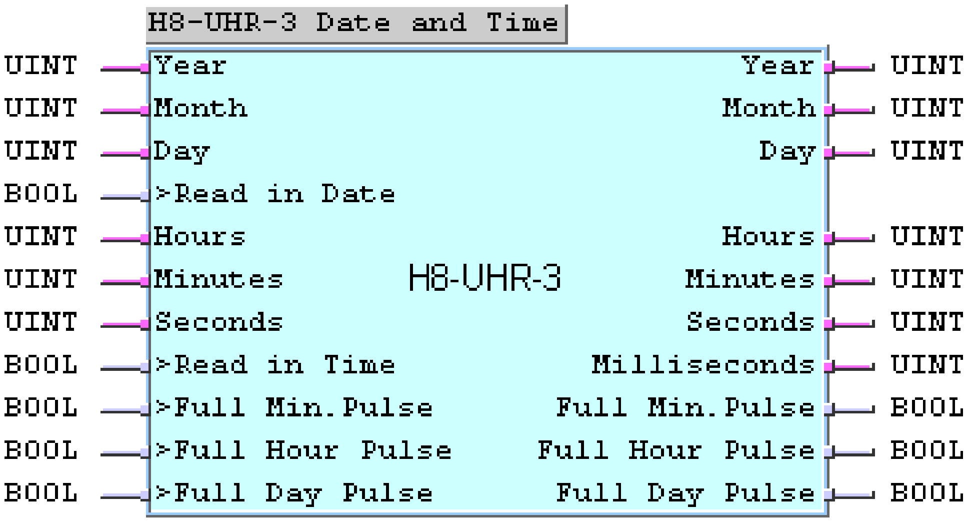
Type: BOOL
Names of the group amplifier channels. Further programming in the logic of the user program is not required.

11.4.5 Cont. red. Group Rel.

Type: BOOL
see output **Trigger group rel.**

12H8-UHR-3

Date, time



Inputs	Type
Year	UINT
Month	UINT
Day	UINT
Enter date	BOOL
Hour	UINT
Minute	UINT
Second	UINT
Enter time	BOOL
Minute pulse	BOOL
Hour pulse	BOOL
Day pulse	BOOL
Year	UINT
Month	UINT
Day	UINT
Hour	UINT
Minute	UINT
Second	UINT

table 13: Input and Ouput specification

Millisecond	UINT
Minute pulse	BOOL
Hour pulse	BOOL
Day pulse	BOOL
Function information	Short information about the block
Block information	General block information
General	Description of block function

table 13: Input and Ouput specification

12.1 Function information

Used to set and change the date and time of the PES

For synchronising the time with external systems

12.2 General block information

Valid from operating system	BS41/51 V6.0-6 (9636) BS51-M, H, HR V6.0-6 (9636) BS11 V6.0-6 (9636) BS51 A1 V6.0-6 (9636) BS41q/51q V7.0-7 (9737)
Special features	May only be programmed once in the user program. Block may only be used in the program type.
Reload (load program), change	Permitted
Use in safety controllers with TÜV test	Permitted
Assignment rule	If one of the first four inputs is assigned, then all four must be assigned. This also applies to the next four inputs. The date and time must always be complete.

table 14: General block information

12.3 GENERAL

The block is used to externally set or change the date and time of the PES. The date and time can therefore be set as boolean variables.

It is also possible to synchronise the time for event recording for external systems.

The block’s outputs carry the current date and time and the minute, hour and day pulses.

12.4 Notes on the assignment of inputs/outputs

12.4.1 Year

Type: UINT
This input must be assigned if a date input is assigned.
Value of the year number to be read in (2-digit). Range 0...99

12.4.2 Month

Type: UINT
This input must be assigned if a date input is assigned.
Value of the month number to be read in. Range: 1...12

12.4.3 Day

Type: UINT
This input must be assigned if a date input is assigned.
Value of the day number to be read in. Range: 1...31

12.4.4 Enter date

Type: BOOL
This input must be assigned if a date input is assigned.
The values present at the **Year, Month and Day** inputs are read in when a positive-going edge change occurs. The date is not entered if invalid values are present.

12.4.5 Hour

Type: UINT
This input must be assigned if a time input is assigned.
Hour value to be read in. Range: 0...23

12.4.6 Minute

Type: UINT
This input must be assigned if a time input is assigned.
Minute value to be read in. Range: 0...59

12.4.7 Second

Type: UINT
This input must be assigned if a time input is assigned.
Second value to be read in. Range: 0...59

12.4.8 Enter time

Type: BOOL
This input must be assigned if a time input is assigned.
The values present at the **Hour, Minute and Second** inputs are read in when a positive-going edge change occurs. The time is not entered if invalid values are present.

12.4.9 Minute pulse

Type: BOOL

The minutes are synchronised when a rising trigger occurs. If the second value is less than 30 it is set to equal 0. If the second value is greater than or equal to 30 it is set to equal 0 and the minute value is incremented by 1.

12.4.10 Hour pulse

Type: BOOL

The hours are synchronised when a rising trigger occurs. If the minute value is less than 30, the minute and second values are set to equal 0. If the minute value is greater than or equal to 30, the minute and second values are set to equal 0 and the hour value is incremented by 1.

12.4.11 Day pulse

Type: BOOL

The days are synchronised when a rising trigger occurs. If the hour value is less than 12, the second, minute and hour values are set to equal 0. If the hour value is greater than or equal to 12, the second, minute and hour values are set to equal 0 and the day value is incremented by 1.

12.4.12 Year

Type: UINT

Current year of the controller as a 2-digit number (0...99).

12.4.13 Month

Type: UINT

Current month of the controller as a 2-digit number (1...12).

12.4.14 Day

Type: UINT

Current day of the controller as a 2-digit number (1...31).

12.4.15 Hour

Type: UINT

Current hour of the controller as a 2-digit number (0...23).

12.4.16 Minute

Type: UINT

Current minute of the controller as a 2-digit number (0...59).

12.4.17 Second

Type: UINT

Current second of the controller as a 2-digit number (0...59)

12.4.18 Millisecond

Type: UINT
Current millisecond of the controller as a 3-digit number (0...999).

12.4.19 Minute pulse

Type: BOOL
A TRUE signal with t = 1 cycle time is output for each full minute, when the second value = 0.

12.4.20 Hour pulse

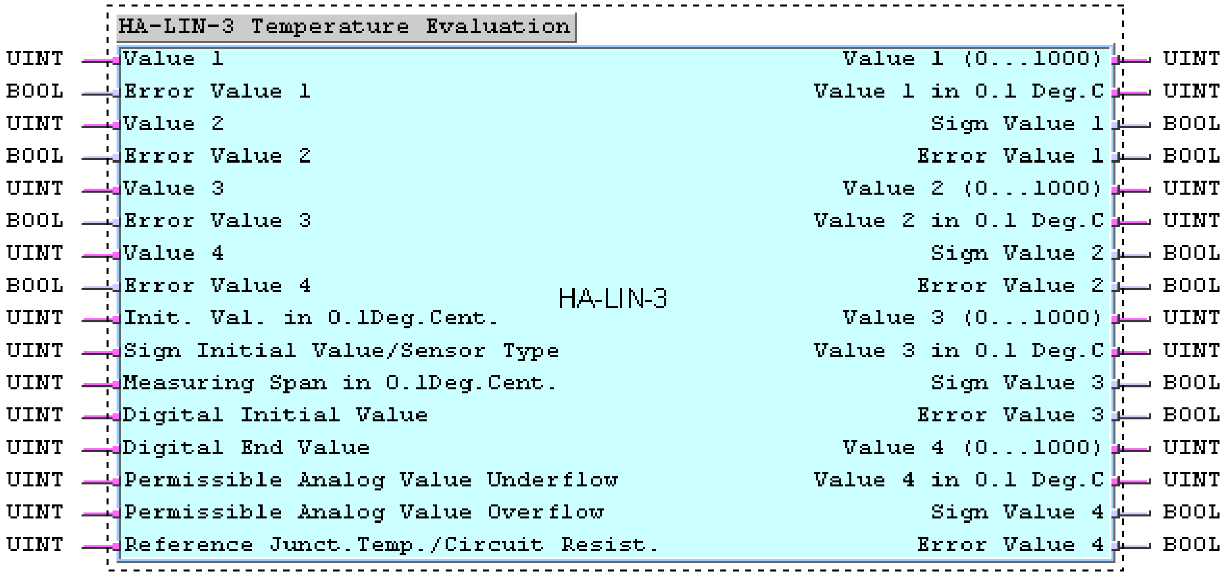
Type: BOOL
A TRUE signal with t = 1 cycle time is output for each full hour, when the second value = 0 and minute value = 0.

12.4.21 Day pulse

Type: BOOL
A TRUE signal with t = 1 cycle time is output for each full day, when the second value = 0, minute value = 0 and hour value = 0.

13HA-LIN-3

Evaluation of temperature measurement



Inputs	Type
Measurement 1...4	UINT
Faulty measurement 1...4	BOOL
Lower range limit in deci °C	UINT
Lower limit sign/type of sensor	UINT
Range in deci °C	UINT
Lower range limit, digital (bool)	UINT
Upper range limit, digital (bool)	UINT
Lowest permitted analog value	UINT
Highest permitted analog value (overflow)	UINT
Ref. temp./line resistor	UINT
Output	
Measurement 1...4 (0...1000)	UINT
Measurement 1...4 in decidegrees	UINT
Sign, measurement 1...4	BOOL
Faulty measurement 1...4	BOOL
Function information	Short information

table 15: Input and output specification

Block information	General block information
General	Description of block function
Programming note	

table 15: Input and output specification

13.1 Function information

Conversion of thermocouple voltages into temperatures in deci °C (tenths of a degree).

Conversion of resistance values of resistance thermometer (PT 100) into temperatures in deci °C.

Fault monitoring of the measurement.

13.2 General block information

Valid from operating system	BS41/51 V6.0-6 (9636) BS51-M, H, HR V6.0-6 (9636) BS51 A1 V6.0-6 (9636) BS41q/51q V7.0-7 (9737)
Reload (load program), change	Permitted
Use in safety controllers with TÜV test	Permitted
Assignment rule	The inputs for defining the measurement range must be assigned. Input ‘lower range limit’ in d °C to input ‘upper range limit’ digital (bool)

table 16: General block information

13.3 GENERAL

The HA-LIN-3 block is used to convert voltages of thermocouples or resistance values of resistance thermometers (PT 100) to the equivalent temperature in deci-degrees Celsius, within a user-defined range. Values are converted on the basis of the tables in the following standards:

- DIN IEC 584 Part 1 Thermocouples, January 1984
- DIN 43710 Electrical temperature sensors, December 1985
- DIN IEC 751 Industrial platinum resistance thermometers, December 1990

Universal applications

The block can be used for all A/D converters of the HIMA PES that are provided for Pt100 elements and thermocouples. The user can define the conversion range by means of two fixed points according to plant requirements. A series logic will be required for processing negative input values (see application example).

13.4 Notes on the assignment of inputs/outputs

13.4.1 Measurement 1...4

Type: UINT
Measurements 1...4 as analog values. Input values are processed as signless integers. If the value contains a sign, it must be processed by a series logic (see programming notes).

13.4.2 Faulty measurement 1...4

Type: BOOL
An appropriate signal at these inputs marks the measurement as valid or invalid.

TRUE	Measurement invalid (fault)
FALSE	Measurement valid (no fault)

13.4.3 Lower range limit in deci °C

Type: UINT; **This input must be assigned**
Initial value of temperature range in deci °C.

13.4.4 Sign start/type of sensor

Type: UINT; **This input must be assigned.**
Sign of the lower range limit. The 1st digit of the valued to be entered indicates the sign.

0	plus
1	minus

The 2nd to 5th digit of the value to be entered indicates the type of measurement sensor:

0001	Platinum resistance thermometer (Pt100) DIN IEC 751, 12.90 -200 °C ... +850 °C (+18.49 Ω... +390.26 Ω)
0002	Thermocouple: Platinum - 13% Rhodium/Platinum Code letter to DIN IEC 584, 1.84: R -50 °C ... +1769 °C (-226 µV ... +21121 µV)
0003	Thermocouple: Platinum - 10% Rhodium/Platinum Code letter to DIN IEC 584, 1.84: S -50 °C ... +1769 °C (-236 µV ... +18709 µV)
0004	Thermocouple: Platinum - 30% Rhodium/Platinum -

	6% Rhodium Code letter to DIN IEC 584, 1.84: B +40 °C ... +1820 °C (0 µV ... +13814 µV)
0005	Thermocouple: Iron/Copper - Nickel Code letter to DIN IEC 584, 1.84: J -210 °C ... +1200 °C (-8096 µV ... +69536 µV)
0006	Thermocouple: Copper/Copper - Nickel Code letter to DIN IEC 584, 1.84: T -270 °C ... +400 °C (-6258 µV ... +20896 µV)
0007	Thermocouple: Nickel - Chrome/Copper - Nickel Code letter to DIN IEC 584, 1.84: E -270 °C ... +1000 °C (-9835 µV ... +76358 µV)
0008	Thermocouple: Nickel - Chrome/Nickel Code letter to DIN IEC 584, 1.84: K -270 °C ... +1372 °C (-6485 µV ... +54875 µV)
0009	Thermocouple: Copper - Copper/Nickel Code letter to DIN IEC 584, 1.84: U -200 °C ... +600 °C (-5700 µV ... +34310 µV)
0010	Thermocouple: Iron - Copper/Nickel Code letter to DIN IEC 584, 1.84: L -200 °C ... +900 °C (-8150 µV ... +53140 µV)

The ranges (°C, µV) for types 2 to 10 relate to a reference temperature of 0 °C. This also applies to the values from the DIN tables.

Example:

10005	minus analog (word) lower range limit Iron/Copper - Nickel thermocouple

13.4.5 Range in deci °C

Type: UINT; **This input must be assigned**.
Range of the analog values in deci °C.
Lower range limit + range = upper range limit

13.4.6 Lower range limit digital (bool)

Type: UINT; **This input must be assigned**.
Digital value corresponding to the temperature at the ‘Lower range limit’ input in deci °C.

Lower range limit digital (bool) = DOFF + UDIN * DEW / UEW

UDIN	Voltage corresponding to the temperature at the ‘Lower range limit’ input in deci °C. The voltage must be taken from the appropriate DIN table.
UEW	Voltage upper range limit. This should be taken from the

	hardware description of the transducer used (Analog signal upper limit).
DEW	Digital upper range limit of the transducer used (dual upper limit), see hardware description.
DOFF	DOFF must be set equal to DEW if negative voltages are processed. If only positive voltages are processed, then DOFF = 0

13.4.7 Upper range limit digital (bool)

Type: UINT; **This input must be assigned.**
Digital upper limit corresponding to the upper temperature according to lower range limit + upper range limit. Calculated according to 'Lower range limit' digital (bool) formula.

13.4.8 Lowest permitted analog value (underflow)

Type: UINT
Limit for the underflow of a measurement. The value entered must be less than or equal to the value at the **Lower range limit digital (bool)** input. If the measurement is between the value defined here and the value at the **Lower range limit digital (bool)** input, the value configured at the **Lower range limit in deci** °C input is output at the corresponding measurement output. If the measurement is less than the value defined here, the corresponding output 'Fault measurement 1...4' is set, and 0 is output as the measurement. If the input is set to 0 or is not assigned, it is set equal to the **Lower range limit digital (bool)** input.

13.4.9 Highest permitted analog value (overflow)

Type: UINT
Limit for the overflow of a measurement. The value entered must be greater than or equal to the value at the **Upper range limit digital (bool)** input. If the measurement is between the value at the **Upper range limit digital (bool)** input and the value defined here, the upper range limit is output at the corresponding measurement output. If the measurement is greater than the value defined here, the corresponding output 'Fault measurement is set, and 0 is output as the measurement. If the input is set to 0 or is not assigned, it is set equal to the Upper range limit digital (bool) input.

13.4.10 Ref. temp./Line resistance

Type: UINT
Reference temperature / line resistance: Correction value for the measurement. The correction value is added or subtracted depending on the configured type (input type sensor). The unit of the value also depends on the sensor type:

Type 1	Line resistance in dΩ. For 2-wire circuits allowing for the trimming resistor. For 3-wire circuits twice the line resistance must be given unless the transducer compensates the line resistance itself. 4-wire resistor: 0

	Compensates the line resistance by a subtraction from the input value.
Types 2 - 10	Reference temperature in d°C. Compensates the voltage value of the reference temperature by an addition to the input value. The reference temperature is the temperature of the compensation point (isothermal block).

If the reference temperature is greater than 0, there is a shift in the lower temperature limit of the DIN tables.

Example:
Thermocouple Type J

Reference temperature 0°C :	-210°C ... 1200°C
Reference temperature 25°C :	-160°C ... 1200°C
	210°C = -8096 µV
	25°C = 1277 µV
Lower limit: -8096 µV + 1277 µV =	-6819 µV = -160°C

13.4.11 Measurement 1...4 (0...1000)

Type: UINT<R>
Linearized values in ‘per mil’ depending on the data at the inputs for the measurement range. 0 per mil is the lower range limit and 1000 per mil the upper range limit. Because a defined reference temperature >0 can be corrected by addition, the conversion result can be greater than 1000 per mil. In the event of a fault, 0 is output for the measurement concerned.

13.4.12 Measurement 1...4 in deci °C

Type: UINT
Linearized values in deci °C depending on the data at the inputs for the measurement range. In the event of a fault, 0 is output for the measurement affected.

13.4.13 Sign measurement 1...4

Type: BOOL
Sign of the measurements at ‘Measurement 1...4 in deci °C’ outputs.

TRUE	positive
FALSE	negative

In the event of a fault, the affected output for the sign is written to FALSE.

13.4.14 Faulty measurement 1...4

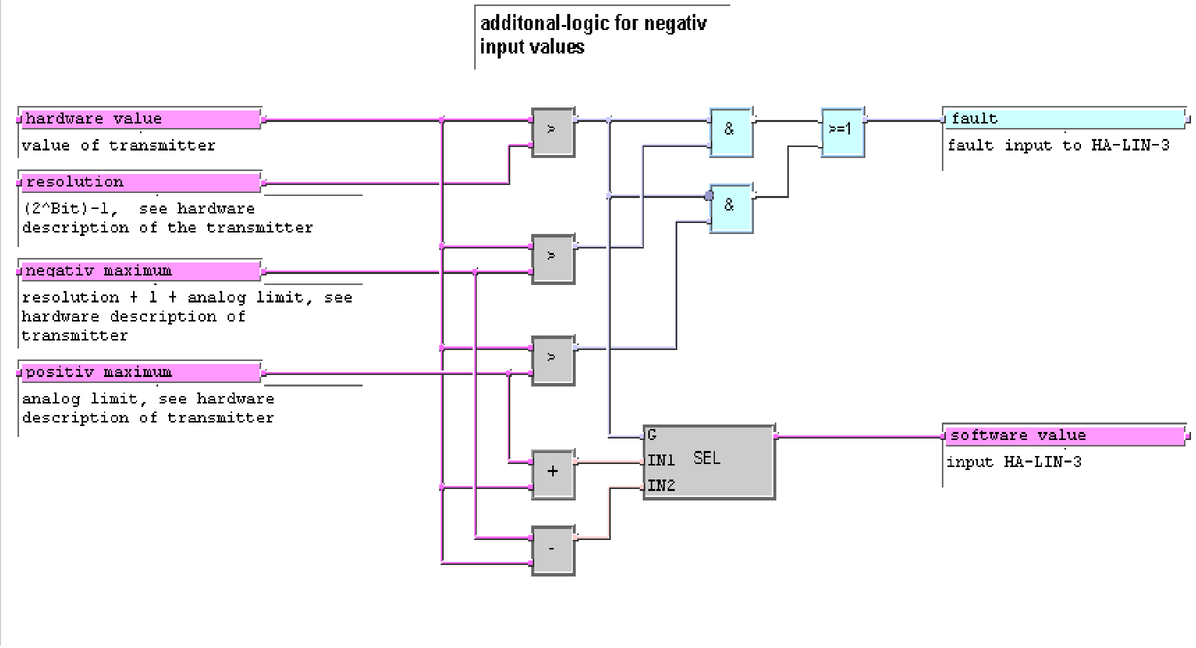
Type: BOOL
If the measurement is faulty, the corresponding output is written to TRUE.
A fault exists when:

- the corresponding input carries faulty measurement TRUE.
- there is a configuration error (meaningless entries at the inputs for the measurement range:´range in deci °C =0
  lower range limit + Range > DIN table range
  lower range limit digital (bool) >= upper range limit digital (bool)
  underflow > lower range limit digital (bool)
  oerflow << upper range limit digital (bool)
- Input value < underflow
- Input value > overflow
- Conversion result > 65535
- Input value < lower range limit digital (bool) and underflow not assigned or equal to 0

13.5 Programming notes and example

13.5.1 Programming with negative input values

An additional ELOP logic is required for negative input values.


additonal-logic for negativ input values
hardware value
value of transmitter
resolution
(2^Bit)-1, see hardware description of the transmitter
negativ maximum
resolution + 1 + analog limit, see hardware description of transmitter
positiv maximum
analog limit, see hardware description of transmitter
>
>
>
+
-
&
&
>=1
G
IN1 SEL
IN2
fault
fault input to HA-LIN-3
software value
input HA-LIN-3

Necessary ELOP logic for negative input values

Measurement_Hardware = Hardware measurement (value of the transducer)
Measurement_SoftwareSoftware measurement (value at HA-LIN-3)
Constant_4095Resolution (2^{Bit}-1)
Negative_Maximumnegative maximum (resolution + 1+ digital upper limit)
Constant_3840positive maximum (digital upper limit)
(See the hardware description of the transducer for resolution, negative maximum and positive maximum)
Fault Fault (fault input at HA-LIN-3)

13.5.2 Assignment of the block when using PT100 sensors with F 6215

Information from hardware description and DIN table:

- Temperature range -200 °C ... +850 °C
- Upper resistance range limits 18.49 Ω to 390.26 Ω
- Digital upper range limit 3840 corresponds to 1 V
- Constant current PT100 IK= 2.5 mA (1V: 2.5 mA = 400 Ω)

Calculating lower and upper range limit digital (bool)

For PT100 sensors, the resistance values must be used for calculation and not the voltages required in the formula.

LOWER DIGITAL $=D_{OFF} + U_{DIN} * D_{EW}/ U_{EW}$

$= 0 + 18.49 * 3840/400$

$= 177.5$

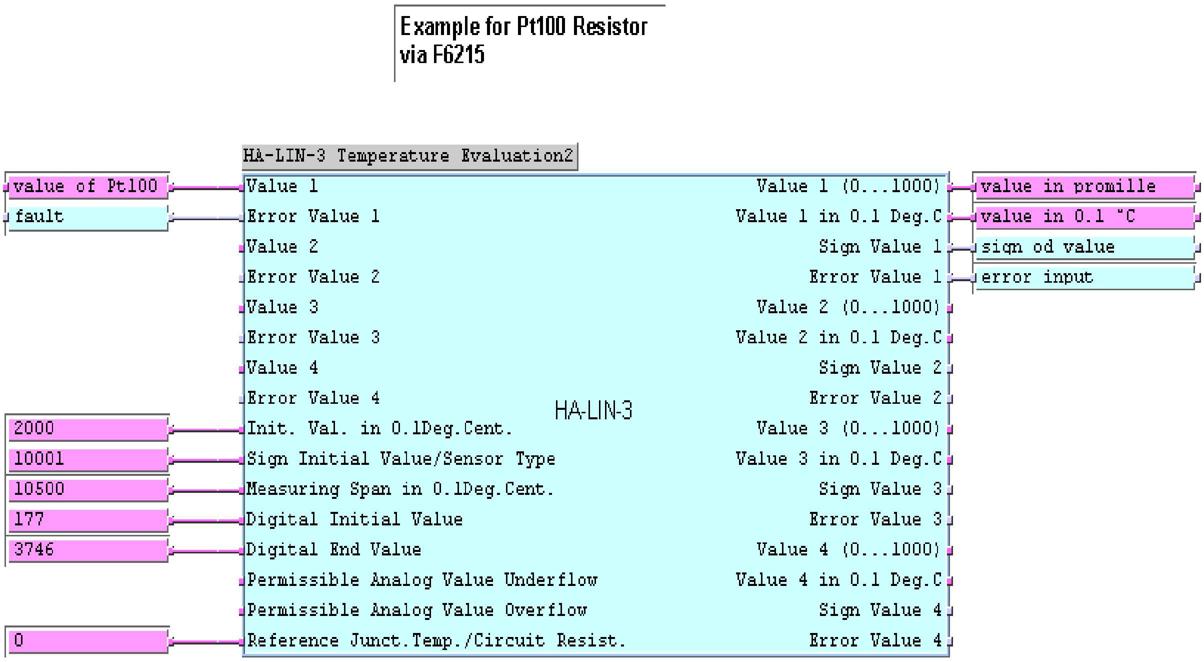
N.B.:D_{OFF}= 0 as negative resistances are not processed.

UPPER DIGITAL$=D_{OFF} + U_{DIN} * D_{EW}/ U_{EW}$

$=0 + 390.26 * 3840/400$

$= 3746.5$

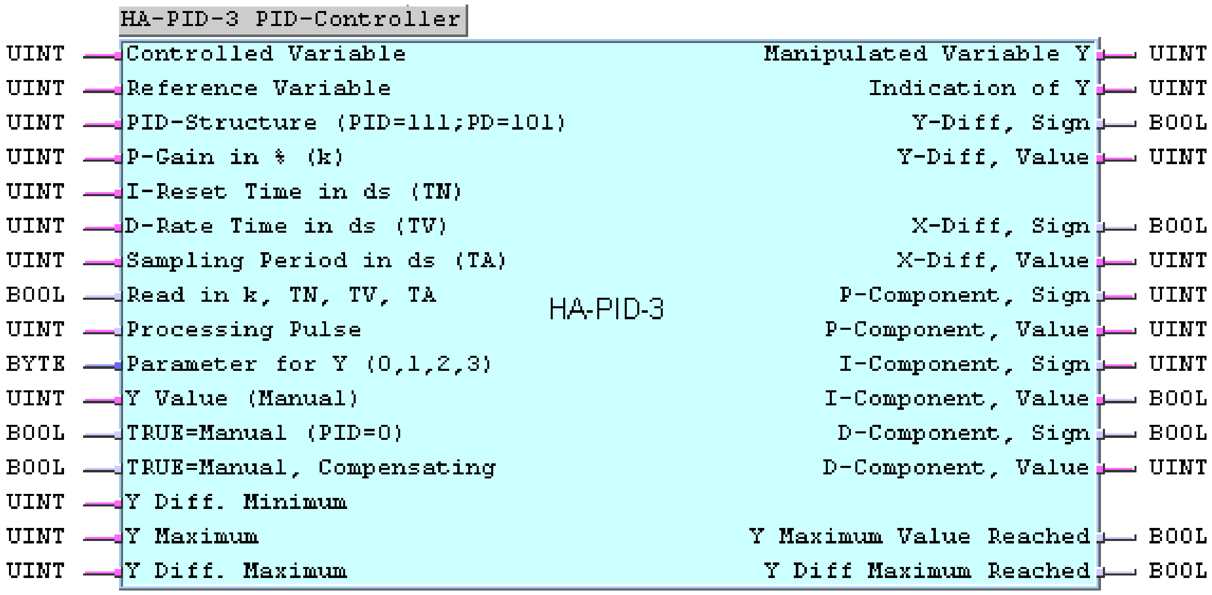
The example below assumes that the line resistance is compensated by other measures (3-wire circuit and compensation by transducer (HIMA standard), 4-wire circuit).



Assignment of the block when using PT100 sensors

14HA-PID-3

PID Controllers



HA-PID-3 PID-Controller

HA-PID-3

UINT	Controlled Variable	Manipulated Variable Y	UINT
UINT	Reference Variable	Indication of Y	UINT
UINT	PID-Structure (PID=111;PD=101)	Y-Diff, Sign	BOOL
UINT	P-Gain in % (k)	Y-Diff, Value	UINT
UINT	I-Reset Time in ds (TN)		
UINT	D-Rate Time in ds (TV)	X-Diff, Sign	BOOL
UINT	Sampling Period in ds (TA)	X-Diff, Value	UINT
BOOL	Read in k, TN, TV, TA	P-Component, Sign	UINT
UINT	Processing Pulse	P-Component, Value	UINT
BYTE	Parameter for Y (0,1,2,3)	I-Component, Sign	UINT
UINT	Y Value (Manual)	I-Component, Value	BOOL
BOOL	TRUE=Manual (PID=0)	D-Component, Sign	BOOL
BOOL	TRUE=Manual, Compensating	D-Component, Value	UINT
UINT	Y Diff. Minimum		
UINT	Y Maximum	Y Maximum Value Reached	BOOL
UINT	Y Diff. Maximum	Y Diff Maximum Reached	BOOL

Inputs	Type
Controlled variable	UINT
Reference variable	UINT
PID structure (PID=111;PD=101)	UINT
P gain in % (k)	UINT
I reset time in ds (TN)	UINT
D rate time in ds (TV)	UINT
sampling period in ds (TA)	UINT
Enter k, TN, TV, TA	BOOL
Processing pulse	BOOL
Parameter (0, 1, 2, 3)	BYTE
Manual setpoint correcting variable	UINT
TRUE=Manual (PID=0)	BOOL
TRUE=Manual compensating	BOOL
Min. correcting deviation	UINT
Max. correcting variable	UINT
Max. correcting deviation	UINT

table 17: Input and output specification

Outputs	
Correcting variable	UINT
Position display	UINT
Correcting deviation sign	BOOL
Correcting deviation value	UINT
System deviation sign	BOOL
System deviation value	UINT
P component sign	BOOL
P component value	UINT
I component sign	BOOL
I component value	UINT
D component sign	BOOL
D component value	UINT
Max. correcting variable achieved	BOOL
Max. correcting deviation achieved	BOOL
Function information	Short information
Block information	General block information
General	Description of block function

table 17: Input and output specification

14.1 Function information

This block contains a digital (boolean) controller that can be operated in the P, I, D, PI, PD and PID modes by parameterising.

14.2 General block information

Valid from operating system	BS41/51 V6.0-6 (9636) BS51-M, H, HR V6.0-6 (9636) BS51 A1 V6.0-6 (9636) BS41q/51q V7.0-7 (9737)
Reload (load program), change the input and output assignment	Not permitted
Use for safety controllers with TÜV test	Permitted

Assignment rule	The following inputs must be assigned: Actual-value controlled variable Setpoint controlled variable PID structure (PID=111;PD=101) Scan time in ds (TA) Read in k, TN, TV, TA Processing pulse Correcting parameter (0, 1, 2, 3) Max. correcting variable Plus one of these inputs: P gain in % (k) I reset time in ds (TN) D rate time in ds (TV)

table 18: General block information

14.3 GENERAL

The block contains a digital (boolean) controller that can be operated in the P, I, D, PI, PD and PID modes by setting the appropriate parameters. Options are provided for parameter changes for the P, I and D components and for switching from Auto to Manual mode.

Controller algorithm

Y = Yo + (k * en) + k * TA / TN * (e1 + ... + en) + k * TV / TN * (en - en-1)
Y = Yo + P component + I component + D component
The calculation error is less than 0.4 %

Y	Correcting variable at the block output
Yo	Initial correcting variable when switching from Manual to Auto: last manual value
k	Gain
en	System deviation, most recent value
en-1	System deviation, one scan time before en
e1	System deviation, first (oldest) value
TN	Reset time
TV	Rate time
TA	Scan time

14.4 Notes on the assignment of inputs/outputs

14.4.1 Actual-value controlled variable

Type: UINT; **This input must be assigned**.
The actual value is entered at this input.

14.4.2 Setpoint controlled variable

Type: UINT; **This input must be assigned**.
The setpoint is entered at this input.

14.4.3 PID structure (PID=111;PD=101)

Type: UINT; **This input must be assigned**.
The numerical value at the input defines the controller structure according to the following table:

Value	Controller function
001	D controller
010	I controller
011	ID controller
100	P controller
101	PD controller
110	PI controller
111	PID controller

The input therefore makes it possible to change the structure of the controller. Entering the value ‘zero’ at a position causes the corresponding structure component to be turned off.
Only the above values may be created.

14.4.4 P gain in %

Type: UINT
Gain P corresponds with k in the control algorithm. The value is given as a percentage of k = 1.
If this input is not assigned, k = 1 is set in the algorithm for the I and D components.

14.4.5 I reset time in ds (TN)

Type: UINT
Corresponds with TN in the control algorithm. The value is input in increments of 0.1 second. Integral saturation: A limiter for the integral component is provided in the block.

The total of the system deviations in the I component is limited
upwards by: e1 + ... + en = (Ym-Yo) / k * TN / TA
downwards by: e1 + ... en = -Yo / k * TN / TA

Ym = maximum correcting variable defined by the **Max. correcting variable** input.

The calculation of the I component limit is initiated by entering the control parameters or by switching to Manual.

The upper saturation limit may not be greater than 65535, and the lower not less than 65535.

14.4.6 D rate time in ds (TV)

Type: UINT
Corresponds to TV in the control algorithm. The value is entered in increments of 0.1 second.

14.4.7 Scan time in ds (TA)

Type: UINT; **This input must be assigned**.
Corresponds to TA in the control algorithm. The value is entered in increments of 0.1 second.
The scan time must agree with the time intervals of processing pulses at the **Processing pulse** input.

14.4.8 Enter k, TN, TV, TA

Type: BOOL; This input must be assigned.
The control parameters are entered with TRUE at this input. They should be entered by a pulse to optimise the block time.

14.4.9 Processing pulse

Type: BOOL; **This input must be assigned**.
With TRUE at this input the controlled variable is entered and the value of the correcting variable is computed with the control algorithm. External processing pulses should be generated and their time interval must agree with the scan time at the Scan time in ds (TA) input.
When several controllers are used, a cyclical staggering of the processing pulses is recommended to achieve shorter cycle times.

14.4.10 Correcting parameter (0,1,2,3)

Type: BYTE; **Values 0,1,2 and 3 only as permitted; this input must be assigned**.
The correcting parameter controls inverting in relation to the maximum value of the correcting variable. The table below gives the assignments depending on the input values:

Vue	Yo	Y	Ya
0	Yw	Yn	Yn
1	Yw	Ym - Yn	Yn
2	Ym - Yw	Ym - Yn	Ym - Yn
3	Ym - Yw	Yn	Ym - Yn

table 19: Calculating the controlling variable

Ya	Position display
Yw	Correcting variable: Value
Yn	Correcting variable computed internally according to the algorithm

This input may only be assigned values 0, 1, 2, 3.

The following consideration applies provided that the reference variable is the controlled variable:

Parameter	Application example
0	Valve must open, closes by spring force
1	Valve must open, opens by spring force
2	Valve must close, closes by spring force
3	Valve must close, opens by spring force

14.4.11 Manual setpoint correcting variable

Type: UINT

Input for manual setpoint.

The correcting variable is influenced independently of the control algorithm in conjunction with the inputs for manual mode (the next 2 inputs). If one of the next two inputs carries a TRUE, the system switches over to the manual setpoint.

If this input is not assigned, the value 0 is used.

14.4.12 TRUE=Manual (PID=0)

Type: BOOL

The controller is changed over to manual with TRUE. Independently of the processing pulse, the manual setpoint is written to the Correcting Variable output in each program cycle. The P, I and D components in the control algorithm are set to 0. The value of the correcting variable present at the time of the changeover is saved as the new initial correcting variable Yo. The controller is switched back to Auto with FALSE, and the correcting variable is calculated by the control algorithm on the basis of the new Yo value.

14.4.13 TRUE Manual compensating

Type: BOOL

The controller is changed over to manual with TRUE. Independently of the processing pulse, the manual setpoint is written to the Correcting Variable output in each program cycle. If the controller has an I component, the P and D components are calculated again. The I component is calculated by correcting variable - (P component) - (D component). For a bumpless changeover the Position Display output must be brought back externally to the Manual Setpoint Correcting Variable input.

The controller is switched back to Auto with FALSE. A bumpless changeover is achieved by calculating the I component. Because of the high computing load, it is advisable to trigger the input with pulses to optimise the cycle time.

14.4.14 Min. correcting deviation

Type: UINT

This value defines a neutral zone for the correcting variable. The new correcting variable is not written to the Correcting Variable output until the computed value becomes greater or less than the last output correcting variable ± neutral zone. This reduces the amount of correction.

14.4.15 Max. correcting variable

Type: UINT

This value determines the upper limit of the correcting variable. If the internal correcting variable exceeds the limit, then only the limit is written to the Correcting Variable output.

14.4.16 Max. correcting deviation

Type: UINT; **This input must be assigned**
This value determines the upper limit between the last output correcting variable and the new correcting variable. This can affect the correcting rate of the controller. If the internal correcting variable exceeds this limit, then only the limit is output as the correcting deviation.

14.4.17 Correcting variable

Type: UINT
The correcting variable can assume values between 0 and 65535 depending on the inputs Min., Max. Correcting Deviation and Max. Correcting Variable.

Auto mode:	Value depends on control algorithm
Manual mode:	Value depends on the inputs for Manual.

14.4.18 Position display

Type: UINT
This output always shows the opening of the actuator. The output value is influenced by the correcting variable and the input correcting parameter. See the table under Correcting Parameter

14.4.19 Correcting deviation sign

Type: BOOL
This output indicates the sign of the correcting deviation of the next output. TRUE is +, FALSE is -.

14.4.20 Correcting deviation value

Type: UINT
The correcting deviation is the difference in the correcting variable between the last two processing pulses. It indicates the change in correcting variable between the current and the previous value.

14.4.21 System deviation sign

Type: BOOL
This output indicates the sign of the system deviation of the next output. TRUE is + or 0, FALSE is -.

14.4.22 System deviation value

Type: UINT
The system deviation is the difference between the reference variable (setpoint) and the controlled variable (actual value).

14.4.23 P component sign

Typ: BOOL
This output indicates the sign of the P component of the next output. TRUE is + or 0, FALSE is -.

14.4.24 P component value

Type: UINT

This output indicates the value of the P component according to the control algorithm.

14.4.25 I component sign

Type: BOOL

This output indicates the sign of the I component of the next output. TRUE is + or 0, FALSE is -.

14.4.26 I component value

Type: UINT

This output indicates the value of the I component according to the control algorithm.

14.4.27 D component sign

Type: BOOL

This output indicates the sign of the D component of the next output. TRUE is + or 0, FALSE is -.

14.4.28 D component value

Type: UINT

This output indicates the value of the D component according to the control algorithm.

14.4.29 Max. correcting variable achieved

Type: BOOL

TRUE at this output indicates that the maximum value of the correcting variable has been reached. The maximum value is entered at the Max. Correcting Variable input.

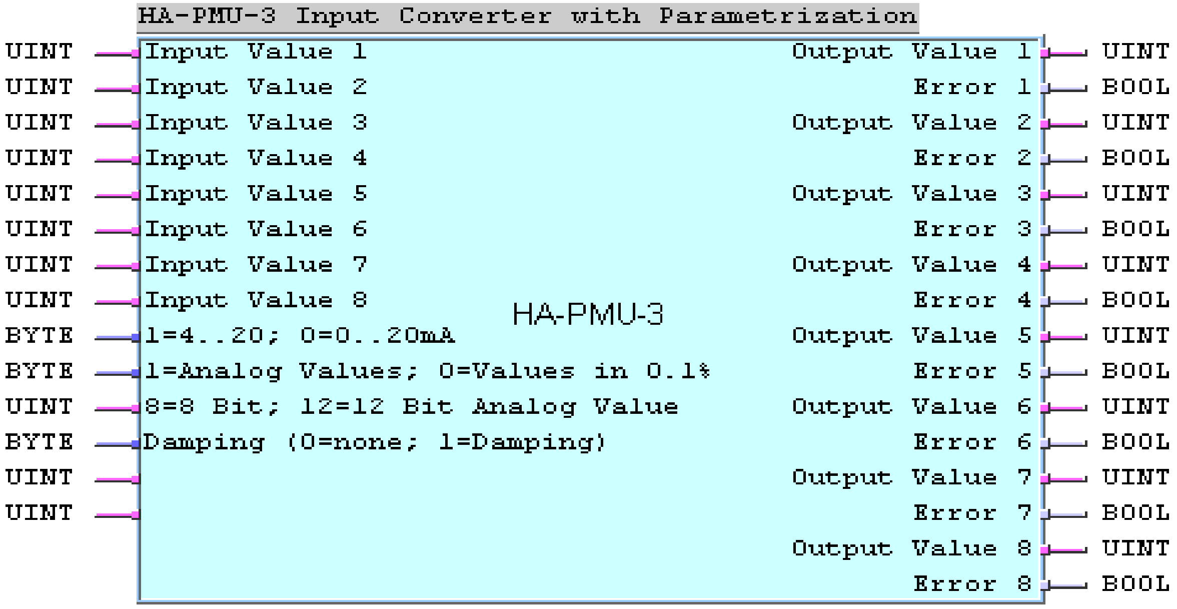
14.4.30 Max. correcting deviation achieved

Type: BOOL

TRUE at this output indicates that the maximum value of the correcting deviation has been reached. The maximum value is entered at the Max. Correcting Deviation input.

15HA-PMU-3

Conversion of values of analog (word) I/O modules



Inputs	Type
Input value 1...8	UINT
1=4...20; 0=0...20 mA	BYTE
1=analog value; 0=per mil value	BYTE
8=8 bit; 12=12 bit analog value	UINT
Damping (0=none; 1=damping)	BYTE
Outputs	
Output value 1...8	UINT
Error 1...8	BOOL
Function information	Short information
Block information	General block information
General	Description of block function

table 20: Input and output specification

15.1 Function information

Measuring transducer for digitised analog values and per mil values
Digitised 12 bit analog value to per mil value (0...20 mA and 4...20 mA)
Per mil value to digitised 12 bit analog value (0...20 mA and 4...20 mA)
Per mil value to digitised 8 bit analog value (0...20 mA and 4...20 mA)

15.2 General block information

Valid from operating system	BS41/51 V6.0-6 (9636) BS51-M, H, HR V6.0-6 (9636) BS51 A1 V6.0-6 (9636) BS41q/51q V7.0-7 (9737)
Reload (load program), change of the input and output assignment	Permitted
Use for safety controllers with TÜV test	Permitted
Assignment rule	If one input is assigned, the following inputs must be assigned: 1=4...20mA; 0=0...20 mA 1=analog value; 0=per mil value 8=8 bit; 12=12 bit analog value

table 21: General block information

15.3 GENERAL

This block converts digitised analog values (0, 768...4095) of analog (word) input modules to per mil values (0..1066, 1083). The conversion is from 0...4095 (see Value table 1) or 768...4095 (see Value table 2) depending on the defined input range.

The block also converts per mil values to digitised analog values for analog output modules. Depending on the defined output range the conversion is to 0...4095 with 12 bit resolution, 0...255 with 8 bit resolution or 768...4095 with 12 bit resolution, 51...255 with 8 bit resolution..

Analogvalue (mA)	Digitised value (decimal)		Per mil value
	12 bit	8 bit	
0	0	0	0
1	192	13	50
2	384	26	100
3	576	38	150
4	768	51	200
5	960	64	250
6	1152	77	300
7	1344	89	350
8	1536	102	400
9	1728	115	450

table 22: 0...20 mA, shown in 1 mA increments

10	1920	128	500
11	2112	140	550
12	2304	153	600
13	2496	166	650
14	2688	178	700
15	2880	191	750
16	3072	204	800
17	3264	217	850
18	3456	230	900
19	3648	242	950
20	3840	255	1000

table 22: 0...20 mA, shown in 1 mA increments

Analog value (mA)	Digitised value (decimal)		Per mil value
	12 bit	8 bit	
4	768	51	0
5	960	64	63
6	1152	77	125
7	1344	89	188
8	1536	102	250
9	1728	115	313
10	1920	128	375
11	2112	140	437
12	2304	153	500
13	2496	166	563
14	2688	178	625
15	2880	191	688
16	3072	204	750
17	3264	217	813
18	3456	230	875
19	3648	242	938
20	3840	255	1000
21.2	4095	-	1083

table 23: 4...20 mA, shown in 1 mA increments

The function of the block is determined by the block’s last 4 inputs and is valid for all 8 input values.

15.4 Notes on the assignment of inputs/outputs

15.4.1 Input value 1...8

Type: UINT
Depending on whether an analog value or a per mil value is being defined, the inputs must be assigned a per mil value to define a per mil value or a digitised analog value to define an analog value.

15.4.2 1=4...20mA; 0=0...20mA

Type: BYTE; **Values 1 and 0 only are permitted, this input must be assigned**.

1.: Conversion of digitised analog values (768...4095) to per mil values or per mil values to digitised analog values (51...255 with 8 bit, 768...4095 with 12 bit according to Table 4 depending on the assignment of the next two inputs.

0: Conversion of digitised analog values (0...4095) to per mil values or per mil values to digitised analog values (0...255 with 8 bit, 0...4095 with 12 bit according to Table 3 depending on the assignment of the next two inputs.

15.4.3 1=analog values; 0=per mil value

Type: BYTE; Values 1 and 0 only are permitted, this input must be assigned.

1: Input values 1...8 must be assigned a digitised analog value (0, 768...4095)

0: Input values 1...8 must be assigned a per mil value (0...1000 with 8 bit resolution; 0...1066 or 1083 with 12 bit resolution).

15.4.4 8=8 bit; 12=12 bit analog value

Type: UINT; Values 8 and 12 only are permitted, this input must be assigned.

8: Conversion of the per mil values (0...1000) of input values 1...8 to a digitised analog value (0, 51...255). If the value 8 is selected, the preceding input must be assigned 0. Otherwise all 8 output values carry the value 0 and the corresponding error outputs carry TRUE.

12: Conversion of the per mil values (0...1066, 1083) of input values 1...8 to a digitised analog value (0, 768...4095) or vice versa depending on the definition of analog values and per mil values.

If this input carries a value that is not 8 or 12, all output values are assigned 0 and the error outputs are set (TRUE).

15.4.5 Damping (0=none; 1=damping)

Type: BYTE; **Values 1 and 0 only are permitted.**

1:	The value is damped on the input side, i.e. transient minor input fluctuations within 3 per mil are not passed to the block output. The block output is compensated if the input is changed permanently. The function is only performed when converting from digitised analog value to per mil value.
0:	There is no damping.

The value “0” is valid if the input is not assigned.

15.4.6 Output values 1...8

Type: UINT

Carries per mil values 0...1066, 1083 (see Tables 1 and 2) that correspond to the input value if the input values are configured as analog values.

Carries digitised analog values that correspond to the input value if the input values are configured as per mil values. 0...255 with 8 bit resolution or 0...4095 with 12 bit resolution (see Tables 3 and 4).

15.4.7 Error 1...8

Type: BOOL

Error output of the relevant input value

Carries FALSE when the input value is inside the input value range.

Carries TRUE when the input value is outside the input value range, i.e. the input value is outside the valid range of values or the block is incorrectly configured at the **8=8 bit; 12=12 bit Analog Value** input. See table below for valid ranges of values:

Input assignment			Range	Kind of conversion
4...20 mA	Value	Resolution		
no	yes	12	0...4095	digitised analog value to per mil value, 12 bit
yes	yes	12	768..4095	digitised analog value to per mil value, 12 bit
no	no	12	0...1066	per mil to digitised analog value, 12 bit
yes	no	12	0...1083	per mil to digitised analog value, 12 bit
no	no	8	0...1000	per mil to digitised analog value, 8 bit
yes	no	8	0...1000	per mil to digitised analog value, 8 bit

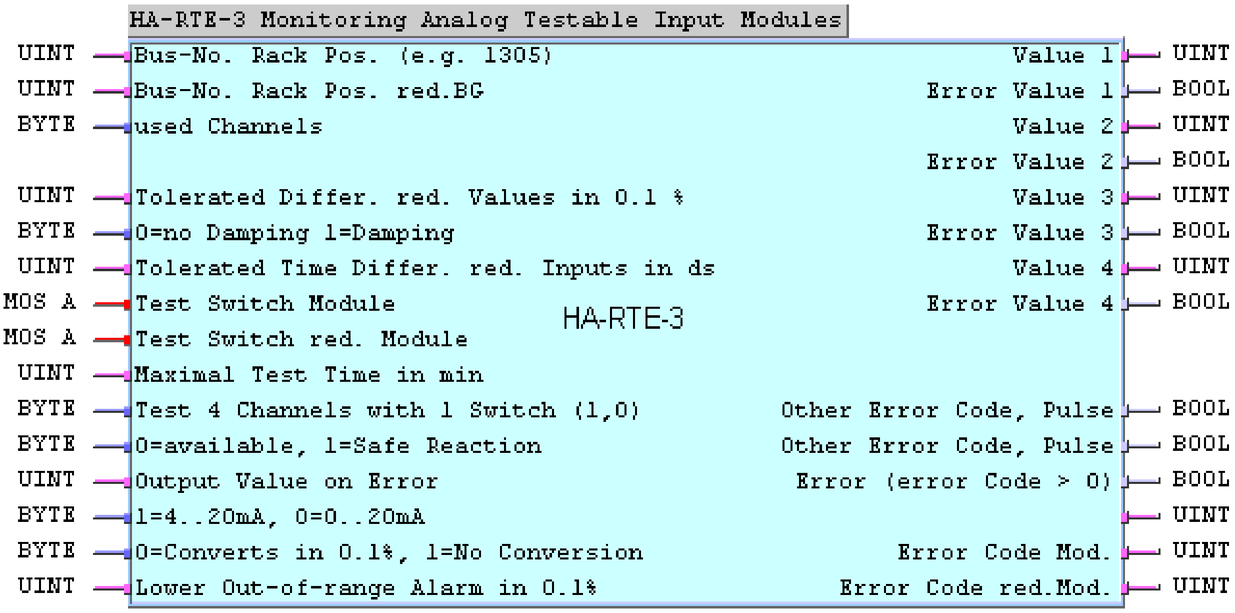
table 24: Conversions

If the block is configured differently from the options listed above, then all error outputs carry TRUE.

If an error output carries TRUE, then the corresponding output carries the value 0.

16HA-RTE-3

Watchdog for analog (word) testable input modules


HA-RTE-3 Monitoring Analog Testable Input Modules
UINT Bus-No. Rack Pos. (e.g. 1305)
Value 1 UINT
UINT Bus-No. Rack Pos. red.BG
Error Value 1 BOOL
BYTE used Channels
Value 2 UINT
Error Value 2 BOOL
UINT Tolerated Differ. red. Values in 0.1 %
Value 3 UINT
BYTE 0=no Damping 1=Damping
Error Value 3 BOOL
UINT Tolerated Time Differ. red. Inputs in ds
Value 4 UINT
MOS A Test Switch Module
Error Value 4 BOOL
HA-RTE-3
MOS A Test Switch red. Module
UINT Maximal Test Time in min
BYTE Test 4 Channels with 1 Switch (1,0)
Other Error Code, Pulse BOOL
BYTE 0=available, 1=Safe Reaction
Other Error Code, Pulse BOOL
UINT Output Value on Error
Error (error Code > 0) BOOL
BYTE 1=4..20mA, 0=0..20mA
UINT
BYTE 0=Converts in 0.1%, 1=No Conversion
Error Code Mod. UINT
UINT Lower Out-of-range Alarm in 0.1%
Error Code red.Mod. UINT

Inputs	Type
Bus No. Rack Pos. (e.g. 1305)	UINT
Bus No. Rack Pos. red. module	UINT
Assigned channels	BYTE
Permitted difference red. values in per mil	UINT
0=no Damping; 1=Damping	BYTE
Permitted time difference red. inputs in ds	UINT
Module test switch	MOS_A
Red. module test switch	MOS_A
Max. test time min	UINT
Test 4 channels with 1 switch (1,0)	BYTE
0=available, 1=safe reaction	BYTE
Output in event of error	UINT
1=4...20 mA, 0=0...20 mA	BYTE
0=conversion to per mil, 1=no conversion	BYTE

table 25: Input and output specification

Max. underflow in per mil (1000=20 mA)	UINT
Outputs	
Value 1...4	UINT
Error value 1...4	BOOL
Change error code, pulse	BOOL
Error	BOOL
Error code, module	UINT
Error code, red. module	UINT
Function information	Short information
Block information	General block information
General	Description of block function

table 25: Input and output specification

16.1 Function information

Conversion of input values of F6213 and F6214 to per mil values

Configuration of error reaction and error display

Value calculation with redundant input modules

16.2 General block information

Valid from operating system	BS41/51 V6.0-6 (9636) BS51-M, H, HR V6.0-6 (9636) BS41q/51q V7.0-7 (9737)
Reload (load program), change the input and output assignment	The two first inputs cannot be changed with reload.
Use for safety controllers with TÜV test	Must be used precisely once in the project for each F 6213 or F 6214 module
Assignment rule	The following inputs must be assigned: Bus No. BT Pos. (e.g. 1305) Assigned channels 1=4...20 mA; 0==...20 mA 0=conversion to per mil, 1 no conversion

table 26: General block information

16.3 GENERAL

Monitoring of analog (word) testable input modules

This block is used to process values and to evaluate and display errors in the analog (word) testable input modules F6213 and F6214 during single-channel or redundant operation of these modules. Value processing includes converting the input values of the module into per mil values. The block can be used for one single or two redundant F6213 or F6214 modules. Integrating input value smoothing is carried out within the accuracy range of the modules.

16.4 Notes on the assignment of inputs/outputs

16.4.1 Bus No. Rack Pos. (e.g. 1305)

Type: UINT; **This input must be assigned**
The position of module F 6213 or F 6214 is shown here as 4-digit decimal numbers according to the position of the board in the CABINET program section.

Example:	Cabinet (1- 2):	1
	Subrack (1- 8):	3
	Board position (1-16):	5
	Position:	1305

If the position of another module is given here, then the output values are set to 0 and code 2 is output as a error code at the module error code BG output.

The assignment must be checked for correctness on commissioning. If the next input (Bus No. Rack Pos. red. module) is not assigned and no errors are present within the module and input circuits, then the block is only used to process values and display module errors. The values are passed to the corresponding outputs.

16.4.2 Bus No. BT Pos. red. module

Type: UINT
The position of the redundant module is displayed here as a 4-digit decimal number - see above.

If the position of another module is given here, then the output value is set to 0 and code 2 is output as the error code at output **Error code red. module**. Here again, the assignment must be checked for correctness on commissioning.

If the input is not assigned, the value 0 is output as the error code of module 2.

16.4.3 Assigned channels

Type: BYTE; **This input must be assigned**.
Assigned channels are represented by the 4 lowest bits. This means bit 0 = channel 1, bit 1 = channel 2, bit 2 = channel 3 and bit 3 = channel 4. When the complete board is assigned the value is 15 or 0F H.

If the input is not assigned, a value of 15 (0F H) is the default. (Input according to IEC 1131-3: 16#0F or 2#00001111)

16.4.4 Permitted difference red. values in per mil

Type: UINT
When operating with two redundant modules, this indicates how great the difference between the two redundant channels may be before an error is displayed. The input in per mil refers to the measurement range 0...20 mA.

16.4.5 0=none; 1=attenuation

Type: BYTE; values 0 and 1 only are permitted.
With attenuation (1, with redundant operation only), a difference is computed from the current measurement and the value of the previous cycle for each of the four channels of both modules in each program cycle; this is added to the permitted difference (input according to IEC 1131-3: 16#01 or 2#00000001).

16.4.6 Permitted time difference red. input in ds

Type: UINT
When operating with redundant modules, the maximum time difference in deciseconds (0.1 s) which can be caused by different measurements at two transmitters, for example, is entered here. An error message is suppressed during this time. This time has no effect on the output of the value at the outputs.

16.4.7 Module test switch

Type: MOS_A, see ELOP_LIB
The field of the service bypass switch that is used to test the transmitters of the input in module 1 is entered here. Depending on the configuration of input Test 4 channels with 1 switch (1,0), all 4 channels are tested together when one of the 4 service switch is operated, or each channel is tested separately. The values of the associated block outputs are held during the defined test time. The inputs of the test switches must be assigned to the individual field elements. If the service bypass switch signal is still present after the test time has timed out, then the current value is output again, and the **Error code, module** output displays a corresponding error code.

16.4.8 Test switch red. module

Type: MOS_A, see ELOP-LIB
Function is as for the previous input for the redundant module; output of the error code at output **Error code, red. module**.

16.4.9 Maximum test time in min

Type: UINT
The test time is given in minutes for the four channels together. Each channel has a separate timer that is activated by the assigned service bypass switch. If TRUE is still present at the test switch outputs when the test time times out, then the current value is converted and output again, and the outputs for the error code display a corresponding error code.

16.4.10 Test 4 channels with 1 switch (1,0)

Type: BYTE; **Values 0 and 1 only are permitted**.

Value input 1: When a service bypass switch is operated all four channels are tested together.
Value input 0: If one of the four service bypass switches is operated, only the corresponding channel is tested.

If the input is not assigned, a value of 0 is the default. (Input according to IEC 1131-3: 16#01 or 2#00000001)

16.4.11 0=available, 1=safe reaction

Type: BYTE; **Values 0 and 1 only are permitted**.
The response of the four **Value n** outputs is defined according to the following table by entering 0 (highly available) or 1 (safety oriented):

Operating status		0	1
1-channel	Normal	AW	AW
	Test	LG	LG
	Error	WF	WF

table 27: Error response

2-channel	Normal	AW	AW
	Channel 1 under test	RW	RW
	Channels 1 and 2 under test	LG	WB
	Channel 1 under test and error in 1	RW	RW
	Channel 1 under test and error in 2	LG	WF
	Error in channel 1	RW	WF/ RW*
	Error in channels 1 and 2	WF	WF

table 27: Error response

*in AK6 RW if single-channel operation is permitted with BS V6.0-6

RW with BS V7.0-7

Key to abbreviations:

AW	Current value
RW	Current value of redundant channel
LG	Last value (held), only valid for the permitted test time
WF	Value in event of error, according to configuration of output in event of error
WB	Value as for module error, according to configuration of output in event of error

If the input is not assigned the default value is 0.

16.4.12 Output in event of error

Type: UINT

The response of the block in the event of an error is configured for each channel by entering a 4-digit number:

	>21,3 mA Priority 1		< min. error or channel error Priority 2		Module error Priority 3	
	Bit	Value	Bit	Value	Bit	Value
0	TRUE	0	TRUE	0	TRUE	0
1	TRUE	max.	TRUE	0	TRUE	0
2	TRUE	max.	TRUE	max.	TRUE	0
3	FALSE	max.	TRUE	0	TRUE	0
5	TRUE	0	TRUE	0	TRUE	max.
6	TRUE	max.	TRUE	0	TRUE	max.
7	TRUE	max.	TRUE	max.	TRUE	max.
8	FALSE	max.	TRUE	0	TRUE	max.

table 28: Response in event of error

Error priority: If different errors occur in two redundant inputs simultaneously, the block responds according to the configuration with the highest error priority (1 = highest priority).

The simultaneous entry of four digits defines the response for each channel separately.

Example: 5 0 1 8
Configuration 8 for channel 4
Configuration 1 for channel 3
Configuration 0 for channel 2
Configuration 5 for channel 1

If the input is not assigned the default value is 0000.

If the number entry is not valid (more than four digits or prohibited values) the value outputs are set to 0 and the error outputs are set to TRUE; the error code outputs display error code 99.

16.4.13 1=4...20 mA, 0=0...20 mA

Type: BYTE; **Values 0 and 1 only are permitted, this input must be assigned**.

1:	Measurement with suppressed zero (4...20 mA)
0:	Measurement without suppressed zero (0...20 mA)

(Input according to IEC 1131-3: 16#01 or 2#00000001)

16.4.14 0=conversion to per mil, 1=no conversion

Type: BYTE; **Values 0 and 1 only are permitted, this input must be assigned**.

0:	The measurements are output in per mil at the value n outputs.
1:	The numerical values of the A/D converters are output as measurements at the Value n outputs (12 bit resolution).

(Input according to IEC 1131-3: 16#01 or 2#00000001) The input values are converted linearly.
0...20 mA = 0...3840 = 0...1000, 21.3 mA = 4095 = 1066
4...20 mA = 768...3840 = 0...1000, 21.3 mA = 4095 = 1083

16.4.15 Max. underflow in per mil (1000 = 20 mA)

Type: UINT
The underflow can be freely selected by entering a per mil value. If the input is not assigned, the default values are 0 (without suppressed zero) and 150 (with suppressed zero, equivalent to 3 mA).

16.4.16 Value 1...4

Type: UINT
The measurements are output as shown in the following table:

Operating status		Output
1-channel	Normal	Value of the channel
	Fault	Parameterised value
2-channel	< Difference	Mean value of both channels
	> Difference < time difference	Mean value of both channels
	> Difference > time difference	Mean value / parameterised value (safe)
	One channel faulty	Value of the red. channel / parameterised value (safe)
	Both channels faulty	Parameterised value

table 29: Value output

16.4.17 Error value 1...4

Type: BOOL
Errors in the event of a fault are displayed as shown in the following table:

Operating status		Output
1-channel	Normal	FALSE
	Fault	Parameterised signal
2-channel	< Difference	FALSE
	> Difference < time difference	FALSE
	> Difference > time difference	TRUE
	One channel faulty	FALSE / Paramet. signal (safe)
	Both channels faulty	Parameterised signal

table 30: Error bit output

16.4.18 Change error code pulse

Type: BOOL
With every value change of the error codes this output carries TRUE for the duration of one program cycle.

16.4.19 Error

Type: BOOL
The output carries TRUE, when an error code >0 is present at one of the

Module or Red. Module error code outputs.

16.4.20 Module error code

Type: UINT
Output of the error code for the first defined module:

Code	Description
0	No error
1	Component error
2	Invalid module position
4	Test time of the module is exceeded
n1	Difference between the redundant channels of two modules is more than permitted
n2	Input value at channel n is 4095, no output when ëOutput in event of errorí input is assigned 3 or 8
n3	Value at channel n less than permitted
n4	Test time of channel n exceeded
n5	Channel error
99	Invalid number entry at Output in event of error input

table 31: Error codes

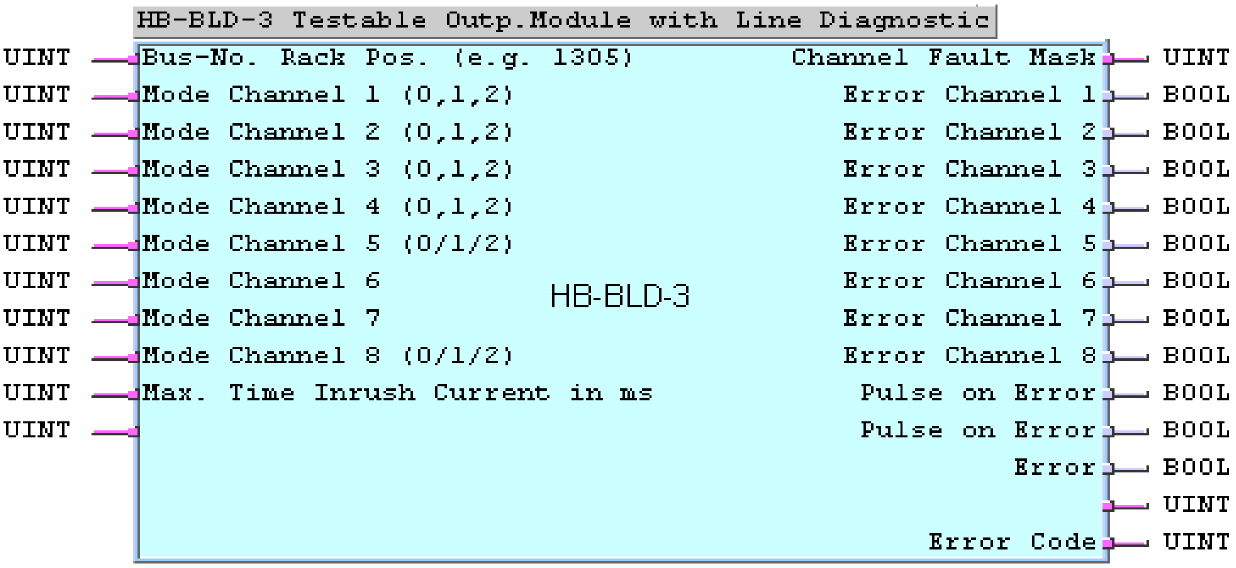
Error codes are reset immediately without an acknowledge as soon as the fault is cleared, except error codes 1 (component error) and n5 (channel error).
If several error codes are present, they are displayed according to the following priorities:
2 > 1 > 99 > 4 > error channel 4 > error channel 3 > ... > error channel 1
An error in channel n is only detected when the associated channel is assigned according to the ëAssigned channelsí input.

16.4.21 Error code red. module

Type: UINT
Output of the error code for the redundant module:
Function as above.

17HB-BLD-3

Testable module, line diagnostic, mono operation



Inputs	Type
Bus No. Rack Pos. (e.g. 1305)	UINT
Mode channel 1...8	UINT
Max. time inrush current in ms	UINT
Outputs	
Error channel 1...8	BOOL
Pulse on error	BOOL
Error	BOOL
Error code	UINT
Function information	Short information
Block information	General block information
General	Description of block function

table 32: Input and output specification

17.1 Function information

Error evaluation and display for the digital (boolean) testable output modules F3323, F3331 and F3334.

Error detection for the function of the module, open circuit and short circuit.

3 modes: no error evaluation, normal mode and inverse mode.

Time setting for open circuit detection, tolerancing the current limiter

17.2 General block information

Valid from operating system	BS 41/51 V6.0-6 (9636) BS51-M, H, HR V6.0-6 (9636) BS41q/51q V7.0-7 (9737)
Reload (load program), change of the input and output assignment	All inputs/outputs can be changed by Reload except input **Bus No. Rack Pos.**
Use for safety controllers with TÜV test	Permitted
Assignment rule	Input **Bus No. Rack Pos.** must be assigned

table 33: General block information

17.3 GENERAL

This block is used to evaluate and display errors for the digital (boolean) testable output modules F3323, F3331 and F3334. It may only be used once per module. Error detection is provided for the module function and for open circuit and short circuit in the output circuits.
The block must be programmed precisely once for each F3323, F3331 and F3334 module when channel errors are processed depending on the MODE. The module must not be used on several blocks (HB-BLD-3, HB-BLD-4). This block is not required for pure error detection because faults are also displayed at the central module and by the system variables.

The MODE of a channel can be switched between the 3 permitted values at any time. Transitions 2-0-1 and 1-0-2 are recommended. The intermediate step via MODE 0 suppresses all error displays during changeover. MODE 1 or MODE 2 must remain valid as values for at least 2 PES cycles after changeover.

The **Max. time inrush current in ms** input can be used to configure the time for tolerancing the current limiter for the module. This is essential when high lamp loads are switched with module F3334.

The normal group shutdown block is used to configure group shutdown with module F3323, F3331 or F3334.

17.4 Notes on the assignment of inputs/outputs

17.4.1 Bus No. Rack Pos. (e.g.1305)

Type: UINT; **This input must be assigned**.
The position of module F3323, F3331 or F3334 is entered as a 4-digit dec-

imal number here, depending on the position of the board in the program section CABINET:

Example:

Cabinet (1-2):	1
Subrack (1-8):	3
Board position(1-16):	5
Position	1305

If the position of another module is entered here, then all 8 error outputs are set to TRUE and the error code output displays error code 2.

The user must verify the validity of the position himself when commissioning; if error code 2 appears the position is not correct.

17.4.2 Mode channel 1...8 (0/1/2)

Type: UINT; **Values 0, 1 and 2 only are permitted**.
Setting the error display and evaluation in the event of an open circuit.

Value 0: No error evaluation for this channel, error messages are suppressed.

Value 1: Normal mode, the output circuit of the output module closed, a detected error is reported by TRUE at the associated **Error channel n** output.

Value 2: "Inverse" mode, i.e. the output circuit should be open. Example: an open contact in the output circuit of the module is required to prevent through-switching of the output signal at certain times. The open contact is detected as an open circuit, but this is OK and is not displayed. In this case a closed contact is reported as an error by **Error channel n**. If the output signal is switched through, i.e. the contact in the output circuit is closed, then the mode must be changed to 1.

Unassigned inputs are treated as mode 0.

Value > 2: The channel is interpreted as faulty (TRUE at the output)
and a channel-related error message is output.

With the 4-channel output module F3334 the inputs for channels 5...8 must be kept free or be assigned 0.

17.4.3 Max. time inrush current in ms

Type: UINT
Defines the delay for detecting open circuit, or the time for tolerancing the current limiter.

not assigned:	0 µs (default)
1...50	valid value in ms
Value > 50	The maximum value of 50 ms is used

A time setting is essential when high lamp loads are switched, and is used to suppress the error display during switch-on.

Extending the delay also increases the cycle time, especially during mode changeover. To keep the cycle time as short as possible, measures should be implemented to ensure that a mode changeover only takes place on one block in a cycle.

17.4.4 Error channel 1...8

Type: BOOL
FALSE indicates correct function, TRUE means an error in the associated output circuit depending on the mode set at the inputs for channel 1...8.

17.4.5 Pulse on error

Type: BOOL
Positive pulse of one cycle timeís duration if the error code changes at the Error Code output.

17.4.6 Error

Type: BOOL
The output carries TRUE when the error code at the Error Code output is not equal to 0.

17.4.7 Error code

Type: UINT

Code	Meaning
0	No error
1	Component error (faulty module) TRUE is output at all error channels. The component fault can only be cleared by ACK or restart. If the board is pulled during operation then an error (TRUE) is reported for all error channels assigned mode 1 until the component fault is detected. The component fault is detected immediately when the outputs of the module are triggered, otherwise detection is delayed slightly by test routines.
2	Invalid position at **Bus No. Rack Pos.** input. All error outputs are set to TRUE. The error display of the actual controller is unaffected, i.e. the diagnostic display does not show any error for the module belonging to this block. Because the error is a programming error, it can only be cleared by modifying the user program and recompiling (code generator) and loading.
3	Module dead through group shutdown. The module is de-energised following a group shutdown. It is not necessarily faulty however. Output is FALSE at all error channels.
n1	Input value of the mode input of **Channel n** is > 2. Example: Error code 51 shows that an invalid mode has been entered for channel 5 at the block input.

table 34: Error codes

n2	Channel error (open circuit) in **Channel n.** Example: Error code 62 means open circuit in the output circuit of channel 6.
n3	Channel error (short circuit) in **Channel n.** Example: Error code 13 means a short in the output circuit of channel 1.

table 34: Error codes

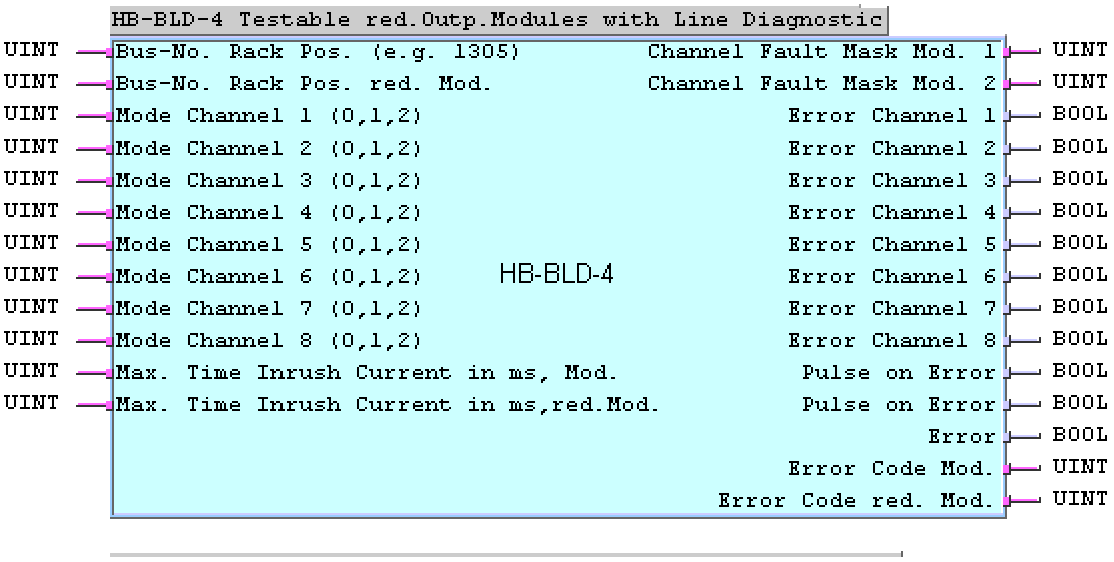
If there is more than one error, the error code of the error which occurred last is always displayed.

If several errors occur simultaneously for error codes n1, n2 and n3, the channel with the highest number is displayed. When this error is cleared, the next highest faulty channel appears. Outgoing channel errors (open circuit only) are detected and cleared automatically. Shorts and component faults must be acknowledged.

A short circuit is safely detected for short circuit currents greater than 1.5 A. In rare cases it is possible for an open circuit to be reported first. The line error is correctly reported.
The response range for an open circuit is 0.5...9.5 mA.

18HB-BLD-4

Testable module, line diagnosis, redundant operation


HB-BLD-4 Testable red.Outp.Modules with Line Diagnostic
UINT Bus-No. Rack Pos. (e.g. 1305)
UINT Bus-No. Rack Pos. red. Mod.
UINT Mode Channel 1 (0,1,2)
UINT Mode Channel 2 (0,1,2)
UINT Mode Channel 3 (0,1,2)
UINT Mode Channel 4 (0,1,2)
UINT Mode Channel 5 (0,1,2)
UINT Mode Channel 6 (0,1,2)
UINT Mode Channel 7 (0,1,2)
UINT Mode Channel 8 (0,1,2)
UINT Max. Time Inrush Current in ms, Mod.
UINT Max. Time Inrush Current in ms,red.Mod.
HB-BLD-4
Channel Fault Mask Mod. 1 UINT
Channel Fault Mask Mod. 2 UINT
Error Channel 1 BOOL
Error Channel 2 BOOL
Error Channel 3 BOOL
Error Channel 4 BOOL
Error Channel 5 BOOL
Error Channel 6 BOOL
Error Channel 7 BOOL
Error Channel 8 BOOL
Pulse on Error BOOL
Pulse on Error BOOL
Error BOOL
Error Code Mod. UINT
Error Code red. Mod. UINT

Inputs	Type
Bus No. Rack Pos. (e.g. 1305)	UINT
Mode channel 1...8	UINT
Max. time inrush current in ms module	UINT
Max. time inrush current in ms red. module	UINT
Outputs	
Error channel 1...8	BOOL
Pulse on error	BOOL
Error	BOOL
Error code module	UINT
Error code red. module	UINT
Function information	Short information
Block information	General block information
General	Description of block function

table 35: Input and output specification

18.1 Function information

Error evaluation and display for the digital (boolean) testable output modules F 3323, F 3331 and F 3334 with redundant use.

Error detection for the function of the module, open circuit and short circuit.

3 modes: no error evaluation, normal mode and inverse mode.

Time setting for open circuit detection, tolerancing the current limiter

18.2 General block information

Valid from operating system	BS 41/51 V6.0-6 (9636) BS51-M, H, HR V6.0-6 (9636) BS41q/51q V7.0-7 (9737)
Reload (load program), change of input and output assignment	All inputs/outputs can be changed by reload except the inputs **for the positions**
Use for safety controllers with TÜV test	permitted
Assignment rule	Inputs **Bus No. Rack Pos.** and **Bus No. Rack Pos. red. module** must be assigned

table 36: General block information

18.3 GENERAL

This block is used to evaluate and display errors for the digital (boolean) testable output modules F3323, F3331 and F3334 with redundant use. It may only be used once per module. Error detection is provided for the module function and for open circuit and short circuit in the output circuits, with the short being displayed only.
The block must be programmed precisely once for each F3323, F3331 and F3334 module when channel errors are processed depending on the MODE. The module must not be used on several blocks (HB-BLD-3, HB-BLD-4). This block is not required for pure error detection because faults are also displayed at the central module and by the system variables.

The MODE of a channel can be switched between the 3 permitted values at any time. Transitions 2-0-1 and 1-0-2 are recommended. The intermediate step via MODE 0 suppresses all error displays during changeover. MODE 1 or MODE 2 must remain valid as values for at least 2 PES cycles after changeover.

Input **Max. time inrush current in ms module** and **Max. time inrush current in ms red. module** can be used to configure the time for tolerancing the current limit for the modules. This is essential when high lamp loads are switched with module F 3334.

The normal group shutdown block is used to configure group shutdown

Note:
A short circuit is safely detected for short circuit currents greater than 1.5 A. In rare cases it is possible for an open circuit to be reported first. The line error is correctly reported.
The response range for an open circuit is 0.5...9.5 mA.

18.4 Notes on the assignment of inputs/outputs

18.4.1 Bus No. Rack Pos. (e.g. 1305)

Type: UINT; **This input must be assigned**
The position of module F 3323, F3331 or F 3334 is entered as a 4-digit decimal number here, depending on the position of the board in the program section CABINET:

Example:	Cabinet (1-2):	1
	Subrack (1-8):	3
	Board position(1-16):	5
	Position	1305

If the position of another module is entered here, then all 8 error outputs are set to TRUE and the error code output displays error code 2. The user must verify the validity of the position himself when commissioning; if error code 2 appears the position is not correct.

18.4.2 Bus No. Rack Pos. red. module

Typ: UINT; **This input must be assigned**.
The position of redundant module F 3323, F3331 or F 3334 is entered as a 4-digit decimal number here, depending on the position of the board in CABINET, see above:

18.4.3 Mode channel 1...8 (0/1/2)

Type UINT, Values 0, 1 and 2 are permitted.
Setting the error display and evaluation in the event of an open circuit.

Value 0: No error evaluation for this channel, error messages are suppressed.
Value 1: Normal mode, the output circuits of the output modules are closed, a detected error (in both output circuits) is reported by TRUE at the associated **Error channel n** output.
Value 2: "Inverse" mode, i.e. the output circuits should be open.
Vaue >2 The channel is interpreted as faulty and a channel related error message is output.

Example: an open contact in the output circuits of the module is required to prevent through-switching of the output signal at certain times. The open contact is detected as an open circuit, but this is OK and is not displayed. In this case a closed contact (in both output circuits) is reported as an error by **Error channel n**. If the output signal is switched through, i.e. the output circuit is closed, then the mode must be changed to 1.

Unassigned inputs are treated as mode 0.

With the 4-channel output module F 3334 the inputs for channels 5...8 must be kept free or be assigned 0.

18.4.4 Max. time inrush current in ms, module ; red. module

Type: UINT
Defines the delay for detecting open circuit, or the time for tolerating the current limit of modules 1 and 2.

not assigned:	0 ms (default)
1...50	valid value in ms
Value > 50	The maximum value of 50 ms is used

Extending the delay also increases the cycle time, especially during mode changeover. To keep the cycle time as short as possible, measures should be implemented to ensure that only one mode changeover takes place in a cycle.

18.4.5 Error channel 1...8

Type: BOOL
Error output of the particular channel as defined in the table below.

Channel 1st module	Channel 2nd module	Module	Error Mode 0	rror Mode 1	Error Mode 2
okay	okay	okay	FALSE	FALSE	TRUE
okay	open circuit	okay	FALSE	FALSE	TRUE
okay	-	faulty	FALSE	FALSE	TRUE
open circuit	okay	okay	FALSE	FALSE	TRUE
open circuit	open circuit	okay	FALSE	TRUE	FALSE
open circuit	-	faulty	FALSE	TRUE	FALSE
-	okay	faulty	FALSE	FALSE	TRUE
-	open circuit	faulty	FALSE	TRUE	FALSE
-	-	faulty with group shutdown	TRUE FALSE	TRUE FALSE	TRUE FALSE

table 37: Output in the event of an error

-: see Module column

Module faulty means that the module is dead following a group shutdown, or that there is a component fault or a short circuit.

18.4.6 Pulse on error

Type: BOOL
Positive pulse of one cycle timeís duration if the error code changes at one of the two Error Code outputs.

18.4.7 Error

Type: BOOL
The output carries TRUE when the error code at one of the Error Code outputs is not equal to 0.

18.4.8 Error code module or red. module

Type: UINT
Error code of the 1st or 2nd module

Code	Meaning
0	No error
1	Component error (faulty module) TRUE is output at all error channels (see error output table) when both modules are faulty. The component fault can only be cleared by ACK or restart. A component fault is detected immediately when the outputs of the module are triggered, otherwise detection is delayed slightly by test routines.
2	Invalid position for one or both modules. All error outputs are set to TRUE. The error display of the actual controller is unaffected, i.e. the diagnostic display does not show any error for the module belonging to this block. Because the error is a programming error, it can only be cleared by modifying the user program and recompiling and loading.
3	Module dead through group shutdown. The module is de-energised following a group shutdown. It is not necessarily faulty however. If both modules are dead following a group shutdown, then FALSE is output at all error channels (see error output table).
n1	Input value of the mode input of **Channel n** is > 2. Example: Error code 51 shows that an invalid mode has been entered for channel 5 at the block input.
n2	Channel error (open circuit) in **Channel n.** Example: Error code 62 means open circuit in the output circuit of channel 6.
n3	Channel error (short circuit) in **Channel n.** Example: Error code 13 means a short in the output circuit of channel 1.

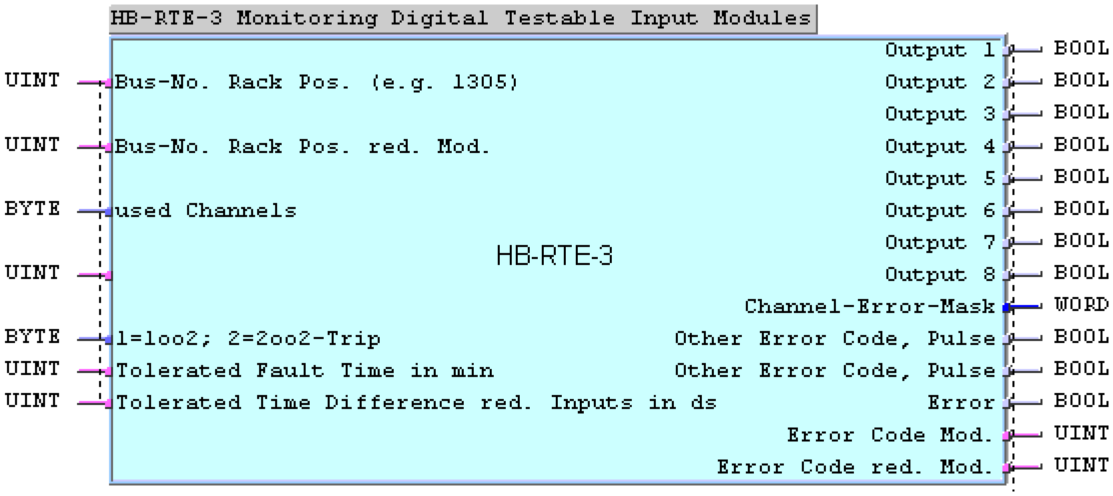
table 38: Error Codes

If there is more than one error, the error code of the error which occurred last is always displayed.

If several errors occur simultaneously for error codes n1, n2 and n3, the channel with the highest number is displayed. When this error is cleared, the next error with the highest channel appears. Outgoing channel errors (open circuit only) are detected and cleared automatically. Shorts and components faults must be acknowledged.

19HB-RTE-3

Watchdog for digital testable input modules



Inputs	Type
Bus No. Rack Pos. (e.g. 1305)	UINT
Bus No. Rack Pos. red module	UINT
Assigned channels	BYTE
1=1of2; 2=2of2 shutdown	BYTE
Permitted error time in min	UINT
Permitted time difference red. inputs in ds	UINT
Outputs	
Output 1...8	BOOL
Channel error mask	WORD
Error code change, pulse	BOOL
Error	BOOL
Error code module	UINT
Error code red. module	UINT
Function information	Short information
Block information	General block information
General	Description of block function

table 39: Input and Ouput specification

19.1 Function information

Evaluation and display of errors in the digital (boolean) testable input modules F 3235, F 3237 and F 3238.
Distinguishes between component faults, short circuit and open circuit; selection of error response with redundant operation (1of2, 2of2 shutdown)

19.2 General block information

Valid from operating system	BS 41/51 V6.0-6 (9636) BS51-M, H, HR V6.0-6 (9636) BS41q/51q V7.0-7 (9737)
Reload (load program), change the input and output assignment	The positions of the modules, the number of assigned channels, the shutdown and the outputs cannot be changed by Reload. All other inputs/outputs can be changed by Reload.
Use for safety controller with TÜV test	Must be used precisely once for each module F 3235, F 3237 and F 3238.
Assignment rule	Inputs **Bus No. Rack Pos. (e.g. 1305)**, **assigned channels** and **1=1of2; 2=2of2 shutdown** must be assigned. In single-channel mode the input must be assigned 0. If **Bus No. Rack Pos. red. module** is assigned, then input **Permitted time difference red. inputs in ds** must also be assigned.

table 40: General block information

19.3 GENERAL

Monitoring of digital (boolean) testable input modules

The block is used to evaluate and display errors in the digital (boolean) testable input modules F3235, F3237 and F3238 during single-channel or redundant operation.

The error detection feature distinguishes between components faults on the module, short circuits and open circuits in the input circuit.

Two modes can be selected for redundant operation: the ‘1 of 2’ shutdown or the ‘2 of 2’ shutdown.

The block can be used for one single or two redundant modules F3235, F3237 or F3238. It may only be used once per module. Combining modules is not permitted.

19.4 Notes on the assignment of inputs/outputs

19.4.1 Bus No. Rack Pos. (e.g. 1305)

Type: UINT; **This input must be assigned**.
The position of module F 3235, F 3237 or F 3238 is shown here as 4-digit decimal numbers according to the position of the board in the CABINET program section.

Example:	Cubicle (1- 2):	1
	Subrack (1- 8):	3
	Board position (1-16):	5
	Position:	1305

If the position of another module is given here, then the output values are set to 0 and code 2 is output as a error code at the module error code output.

The assignment must be checked for correctness on commissioning; if error code 2 appears the position is not correct.

If the **Bus No. Rack Pos. red. module** input is not assigned and no errors are present within the module and input circuits, then the input signals of channels 1...8 of the module are passed to the corresponding block outputs.

When an error occurs in a channel the last status at the corresponding block output is held and reset to FALSE when the defined error time times out, if the error is still present.

19.4.2 Bus No. BT Pos. red. module

Type: UINT
The position of the redundant module F 3235, F 3237 or F 3238 is given here, See above.

The type of shutdown (1of2 or 2of2) must also be configured and the permitted time difference must be defined for redundant operation.

If the input is not assigned, the value 0 is output at the **Error code red. module** output.

19.4.3 Assigned channels

Type: BYTE; **This input must be assigned**.
Assigned channels are represented by the lowest 8 bits. This means bit 0 = channel 1, bit 1 = channel 2, bit 2 = channel 3, bit 3 = channel 4, bit 4 = channel 5, bit 5 = channel 6, bit 6 = channel 7 and bit 7 = channel 8. The value is therefore 255 (0FF H) when the complete board is assigned (input according to IEC 1131: 16#FF or 2#11111111).

If it is not assigned the value 0 is used, i.e. the block always sends FALSE to the outputs whatever the input signals.

Unused channels should be terminated with a resistor.

19.4.4 1=1of2; 2=2of2 shutdown

Type: BYTE; **Values 1 and 2 only are permitted, this input must be assigned if redundancy is used**.
The input is only read in once during a cold restart of the system. Changes cannot be made during ongoing operation.

Value input 0 (assignment for single-channel operation)
In single-channel operation the input must be assigned 0.

Value input 1 (1 of 2 shutdown, equivalent to ANDing):
With the 1 of 2 shutdown the redundancy of the modules is used to enhance safety.

If no errors of the input modules and input circuits are present, the input signals of channels 1...8 of the modules are sent ANDed to the corresponding outputs of the block.

When an error occurs in a channel the last status at the corresponding block output is held and reset to FALSE when the defined error time times out, if the error is still present. If FALSE is at the other error-free input or if errors occur simultaneously in both channels (double error) the block is set to FALSE without a delay.

Value input 2 (2 of 2 shutdown, equivalent to ORing):
With the 2 of 2 shutdown the redundancy of the modules is used to enhance availability.

If no errors of the input modules and input circuits are present, the input signals of channels 1...8 of the modules are sent ORed to the corresponding outputs of the block.

If an error occurs in one channel the input signal of the other channel is passed to the block output.

Only when errors occur simultaneously in both channels (double error) is the last status at the corresponding block output held and reset to FALSE when the defined error time has timed out, if the double error is still present.

19.4.5 Permitted error time in min

Type: UINT
Do not assign. Shutdown is performed directly.

19.4.6 Permitted time difference red. inputs in ds

Type: UINT, **This input must be assigned if redundancy is used**.

When operating with redundant modules, the maximum time difference that can be caused by different switching instants in two sensors is entered here in deciseconds (0.1 s). An error message is suppressed during this time.

19.4.7 Output 1...8

Type: BOOL
Usable outputs are defined by the **Assigned channels** input:

Table 3: Output circuits

Operating status		Status at output n	
1-channel:	Normal	Current signal	
	Fault	Last status, after error time: FALSE	
2-channel:		Parameter 1 v 2	Parameter 2 v 2
	Normal	Current signal	Current signal
	One channel disturbed	Last status, after error time FALSE*	Signal of the undisturbed channel
	Two channels disturbed	FALSE	Last status, after error time: FALSE

table 41: Output circuits

*)BS V6.0-6: If single-channel operation in AK6 is permitted, the signal of the undisturbed channel is processed
BS V7.0-7: The signal of the undisturbed channel is processed

19.4.8 Channel error mask

Type: WORD

Bit 0 ... Bit 7: Channel 1 ... Channel 8 1st module
Bit 8 ... Bit 15: Channel 1 ... Channel 8 2nd (redundant) module

‘Bit set’ means channel error. If there is no redundant module, bits 8...15 = 0. This means that only values between 0 and 255 are possible with single-channel operation.

A channel error is only displayed for the input channels that are also processed by the block. Errors are defined with the **Assigned channels** input. The 8x channel error and component error are exceptions; bit 0...7 is set for the 1st module (value = 16#00 FF = 255) and bit 8...15 is set for the 2nd (redundant) module (value = 16#FF 00 = 65280) irrespective of the **Assigned channels** input. Remember that the value at the block output is always a combination of all errors.

19.4.9 Error code change, pulse

Type: BOOL
With each change of the error code this output carries TRUE for the duration of a program cycle.

19.4.10 Error

Type: BOOL
The output carries TRUE when an error code > 0 is present at one of the two error code outputs.

19.4.11 Error code module

Type: UINT
Output of the error code for the first defined module:

Code	Meaning
0	No error
1	Component error
2	Invalid input name at **Bus No. BT Pos.** input
n1	Difference between the two channels n (1...8) of two redundant modules; is only output for both channels when no other error is present
n2	Line short in input circuit channel n (1...8)
n3	Open circuit in input circuit channel n (1...8)

table 42: Error codes

If some channels are not used and line faults occur in all other channels simultaneously, then error code 1 (component fault) is output. This also happens when line faults occur simultaneously in each eight used channels.

Error codes are reset after the fault is cleared and acknowledged. Error code n1 is an exception to this rule - it is only output when no other error is present and is reset when cleared but without acknowledgment.

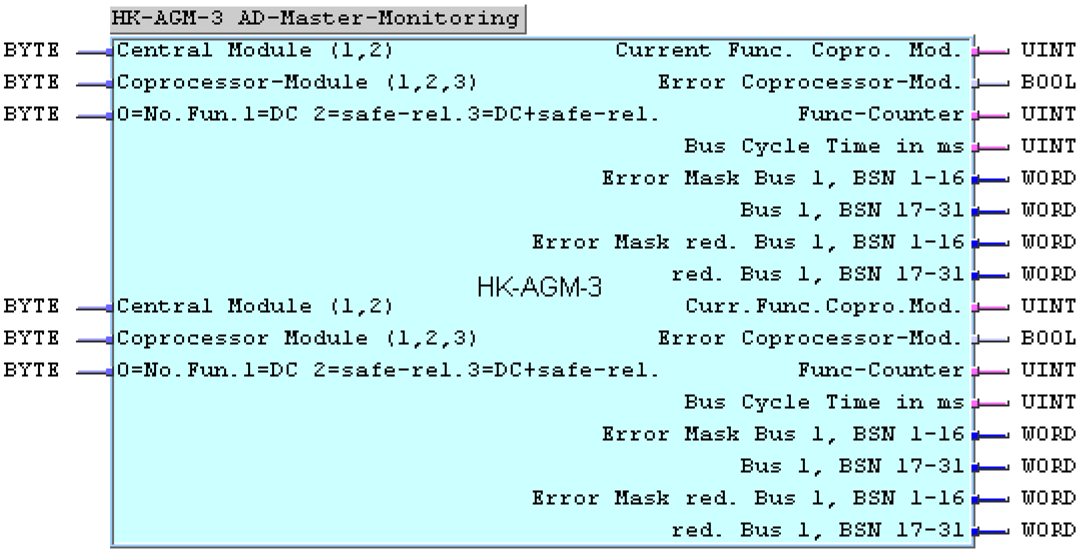
The error code of the error that occurred last is output. If this has been cleared and acknowledged and other faults are still present, then the system displays the error with the highest channel number.

19.4.12 Error code red. module

Type: UINT<R>
Output of the error code for the redundant module:
Function as described above.

20HK-AGM-3

PES Master Watchdog


HK-AGM-3 AD-Master-Monitoring
HK-AGM-3
BYTE Central Module (1,2)
BYTE Coprocessor-Module (1,2,3)
BYTE 0=No.Fun.1=DC 2=safe-rel.3=DC+safe-rel.
Current Func. Copro. Mod. UINT
Error Coprocessor-Mod. BOOL
Func-Counter UINT
Bus Cycle Time in ms UINT
Error Mask Bus 1, BSN 1-16 WORD
Bus 1, BSN 17-31 WORD
Error Mask red. Bus 1, BSN 1-16 WORD
red. Bus 1, BSN 17-31 WORD
BYTE Central Module (1,2)
BYTE Coprocessor Module (1,2,3)
BYTE 0=No.Fun.1=DC 2=safe-rel.3=DC+safe-rel.
Curr.Func.Copro.Mod. UINT
Error Coprocessor-Mod. BOOL
Func-Counter UINT
Bus Cycle Time in ms UINT
Error Mask Bus 1, BSN 1-16 WORD
Bus 1, BSN 17-31 WORD
Error Mask red. Bus 1, BSN 1-16 WORD
red. Bus 1, BSN 17-31 WORD

Inputs	Type
Central module (1,2)	BYTE
Coprocessor module (1, 2, 3)	BYTE
0=no func. 1=DC 2=safety-ori. 3=DC+safety-ori.	BYTE
Outputs	
Current function coprocessor module	UINT
Error in coprocessor module	BOOL
Function counter	UINT
Bus cycle time in ms	UINT
Error mask bus 1, BTN 1-16	WORD
Bus 1, BTN 17-31	WORD
Error mask red. bus, BTN 1-16	WORD
Red. bus, BTN 17-31	WORD
Function information	Short information
Block information	General block information
General	Description of block function

table 43: Input and output specification

20.1 Function information

Monitoring the PES masters for operation reliability

Monitoring connections between the PES master and the PESs on the HIBUS (normal bus) and reserve bus

Function display of the PES master

Displays of current bus cycle time

20.2 General block information

Valid from operating system	BS41/51 V6.0-6 (9636) BS51-M, H, HR V6.0-6 (9636) BS41q/51q V7.0-7 (9737)
Reload (load program), change the input and output assignment	Permitted
Use for safety controllers with TÜV test	Permitted
Assignment rule	If the **Central module (1,2)** input is assigned then the associated inputs **Coprocessor module (1,2,3) 0=no func. 1=DC 2=safety-ori., 3=DC+safety-ori.** must also be assigned.

table 44: General block information

20.3 GENERAL

Block HK-AGM-3 is used to monitor the operation and configuration of a coprocessor module (F 8621) in PESs of the type H51, H51q. The block can monitor 2 coprocessor modules and can be called several times in the user program.

The block outputs the current function of the coprocessor module. If this deviates from the block’s configured function, an error output is set.

It also displays the current connections of the PES master to the slaves for the HIBUS (normal bus) and the reserve bus.

The current bus cycle time is also displayed.

20.4 Notes on assigning the inputs/outputs

20.4.1 Central module (1,2)

Type: BYTE; **Values 1 and 2 only are permitted.**
Value 1 or 2 defines the central module to which the tested coprocessor module belongs.

1:	left central module
2:	right central module

20.4.2 Coprocessor module (1, 2, 3)

Type: BYTE; **Values 1, 2 and 3 only are permitted. This input must be assigned if the Central module (1,2) input is assigned.**
The position of the coprocessor module is defined at this input; the coprocessor modules are numbered in sequence from left to right (1, 2, 3).

20.4.3 0=no func. 1=DC 2=safety-rel. 3=DC+safety-rel.

Type: BYTE; **Values 1, 2 and 3 only are permitted. This input must be assigned if input Central module (1,2) is assigned.**
The function of the PES master must be entered here.

0	The selected coprocessor module contains no PES master project
1	PES master as data centre
2	PES master for starting safety-related transmissions
3	PES master both as a data centre and for starting safety-related transmissions.

20.4.4 Current. funct. copro. mod.

Type: UINT
The output carries the value of the current function of the coprocessor module (see above). If the coprocessor module fails the value is 0.

20.4.5 Error in coprocessor module

Type: BOOL
The output carries FALSE if the current function is the same as the function present at inputs **0=no.func. 1=DC 2=safety-rel. 3=DC+safety-rel.** (including connection failures). If the current function is not the same or if the coprocessor module has failed (e.g. faulty or not present), the output carries TRUE.

20.4.6 Function counter

Type: UINT
The function counter is incremented by 1 each time the PES master performs a read or write transmission or starts a safety-oriented data transfer.
Value range: 0...65535, after 65535 the counter starts again from 0.

If a connection between the coprocessor module and the HIBUS fails, the output holds its last value. If the coprocessor module fails, the output is set to 0.

20.4.7 Bus cycle time in ms

Type: UINT
The current bus cycle time is output here. If the coprocessor module has no master project or if it has failed, the value 0 is output.

If a connection between the coprocessor module and the HIBUS fails, the output holds its last value.

20.4.8 Error mask bus 1, BTN 1 - 16

Type: WORD
Output of the status of the connection from the PES master to the PESs with bus station numbers 1...16 via the HIBUS, in which

1st bit (lowest significance)	= PES with bus station number 1
...	
16th bit (highest significance)	= PES with bus station number 16

The corresponding bit of the status mask carries FALSE if the PES master has a connection to the PES with the corresponding bus station number via the HIBUS, or no slave with this bus station number has been configured in the bus configuration. If there is no connection via the HIBUS, then the corresponding bit of the status mask carries TRUE. If the coprocessor module has no master project or has failed, the value set to 0.

20.4.9 Bus 1, BTN 17 - 31

Type: WORD
Output of the status of the connection from the PES master to the PESs with bus station numbers 17...31 via the HIBUS, in which:

1st bit (lowest significance)	=PES with bus station number 17
...	
15th bit	=PES with bus station number 31
16th bit (highest significance)	=always 0 (no meaning)

The output is as described above.

20.4.10 Error mask red. bus, BTN 1 - 16

Type: WORD
Output of the status of the connection from the PES master to the PESs with bus station numbers 1...16 via the reserve bus.

Function as for Bus 1.

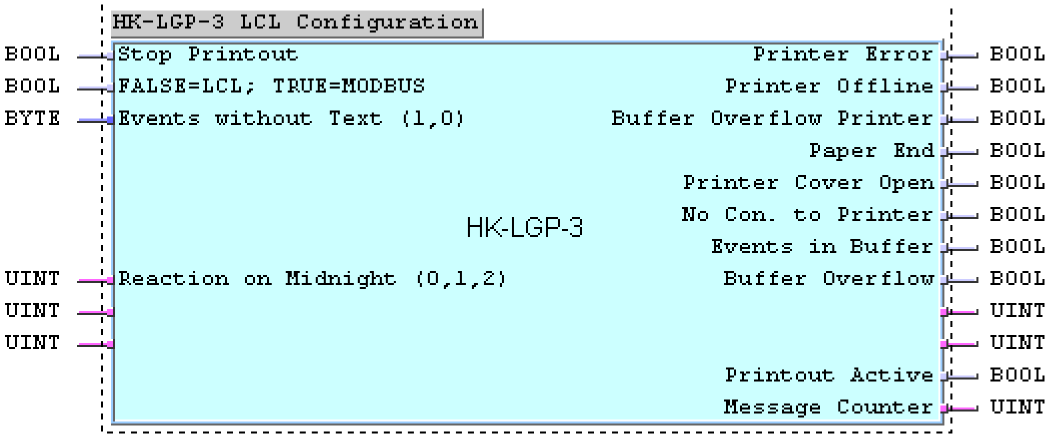
20.4.11 red. Bus, BTN 17 - 31

Type: WORD
Output of the status of the connection from the PES master to the PESs with bus station numbers 17...31 via the reserve bus.

Function as for Bus 1.

21HK-LGP-3

Configuring LgP



Inputs	Type
Stop event printout	BOOL
False=LGP; TRUE=MODBUS	BOOL
Output emergency message texts (1,0)	BYTE
Response to day change (0, 1, 2)	UINT
Outputs	
Printer error	BOOL
Printer offline	BOOL
Printer buffer overflow	BOOL
No paper	BOOL
Printer cover open	BOOL
Printer conn. lost	BOOL
Event in buffer	BOOL
Buffer overflow	BOOL
Printout active	BOOL
Message counter	UINT
Function information	Short information

table 45: Input and output specification

Block information	General block information
General	Description of block function
Programming example	Several controllers, one printer

table 45: Input and output specification

21.1 Function information

Evaluation and configuration of logic plan controlled reporting in systems H11, A1, H41, H51, H41q, H51q.

Issue of print enable when several controllers use a common printer

Output of emergency message texts

Setup of page numbering on day change

21.2 General block information

Valid from operating system	BS41/51 V6.0-6 (9636 BS51-M, H, HR V6.0-6 (9636) BS11 V6.0-6 (9636) BS51 A1 V6.0-6 (9636) BS41q/51q V7.0-7 (9737)
Special features	Can only be used once, block may only be used in the program type
Reload (load program), change of the input and output assignment	Permitted
Use for safety controllers with TÜV test	Permitted

table 46: General block information

21.3 GENERAL

Block HK-LGP-3 is used to evaluate and configure logic plan controlled reporting in the H11, A1, H41, H51, H41q and H51q systems. Printer feedback is evaluated by means of this block.
It also enables the printer where several PESs are accessing a common printer. PESs are coupled to the printer by the interface converter H 7505.

Planning guidelines
Logic plan controlled reporting is carried out on SIO channel 2 of the central module (A1,H41, H51, H41q, H51q) or interface 2 (plug connector 1 pins) with the H11. The interface is always occupied by the LgP.

In the resource type the variables must be assigned the attribute LgP (HW assignment) and the appropriate printer text.

Reporting only takes place in the RUN and MONO modes.

Suitable printers and printer settings are given in the printer data sheets.

The printer is connected with interface converter H 7505. The settings of application 10 must be made on the H 7505.

This block is needed when a number of PESs are accessing a common printer or when feedback from the printer is evaluated, or for changeovers between Modbus and LgP.

A redundant connection of the printer with HIMA PES H41, H51, H41q and H51q with 2 central modules is possible.

21.4 Notes on the assignment of inputs/outputs

21.4.1 Stop event printout

Type: BOOL
TRUE stops the printout at the printer. Stopping is necessary in order to enable the print authorization to be passed to other controllers.
Non-assignment is identical to FALSE and means ‘printer is enabled’.

21.4.2 FALSE=LGP, TRUE=MODBUS

Type: BOOL
TRUE activates event polling via Modbus (an additional interface is required).

LgP is active with FALSE or non-assignment.

21.4.3 Output emergency message texts (1,0)

Type: BYTE; **Values 0 and 1 only are permitted**.
1: Output of "emergency message texts" for events with no defined output text on the printer
0: No output of "emergency message texts” (or no assignment).

Format of emergency message text: n: E.xxxx <<time>> <<value>>

n:	Bus station number
E.xxxx	Event number
<time>	Time of event
<value>	TRUE or FALSE

The event number is the sequence number of the event. See the event list of ELOP II.

21.4.4 Response to day change (0,1,2)

Type BYTE; **Values 0, 1 and 2 only are permitted.**
This input determines the response of the printout to day changes. The following values are possible at input 4:
0 No new page at day change, page numbering continu-

	es. Range 1...65535, then start at 1.
1	New page at day change, page numbering continues. Range 1...65535, then start at 1.
2	New page at day change and page number 1.

If the input is not assigned or is assigned a value > 2, then the statements re value 0 apply.

21.4.5 Printer error

Type: BOOL
The output carries TRUE if the printer is in a fault condition. FALSE means ‘printer OK’. Details of the fault are given in the following outputs provided the printer supplies detailed information about the cause of its faults.

21.4.6 Printer offline

Typ: BOOL
This output carries TRUE if the printer is not ready. FALSE means ‘printer ready’.

21.4.7 Printer buffer overflow

Type: BOOL
This output caries TRUE when the printer buffer has overflowed. FALSE means no overflow is present.

21.4.8 No paper

Type: BOOL
This output carries TRUE when no paper is present. FALSE means paper is present.

21.4.9 Printer cover open

Type: BOOL
This output carries TRUE when the printer cover is open and printing is therefore interrupted. FALSE means the printer cover is closed.

21.4.10 Printer conn. lost

Type: BOOL
This output carries TRUE when the connection between PES and printer is faulty or interrupted. FALSE means that the connection between PES and printer is present.

21.4.11 Event in buffer

Type: BOOL
FALSE means that there is no event in the buffer. TRUE means that the buffer contains events.

This signal can be used to enable the printer when a number of controllers use a common printer.

The signal must not be defined as an event.

21.4.12 Buffer overflow

Type: BOOL
This output is TRUE when the event buffer is full.

21.4.13 Printout active

Type: BOOL
This output carries TRUE while data are being sent to the printer. If the "Stop event printout" input is triggered, the TRUE remains present until the current message has finished. FALSE means that the printer is not triggered.

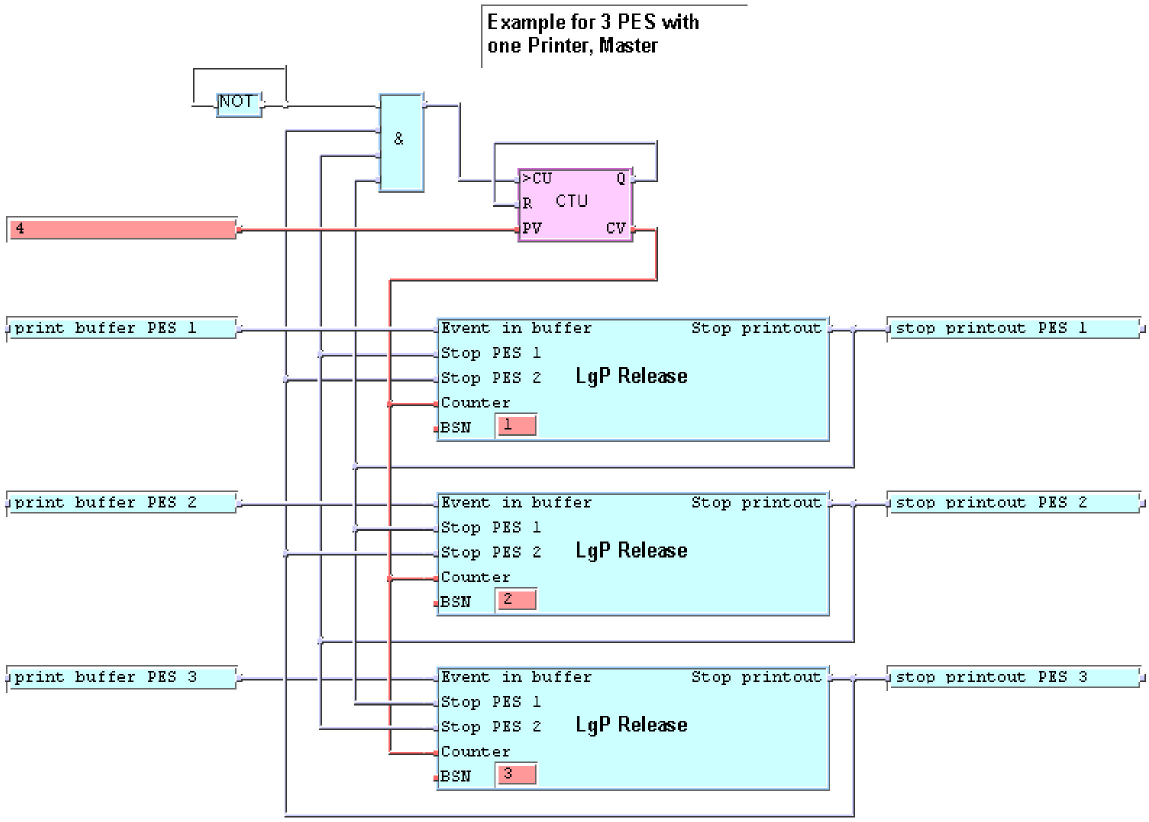
21.4.14 Message counter

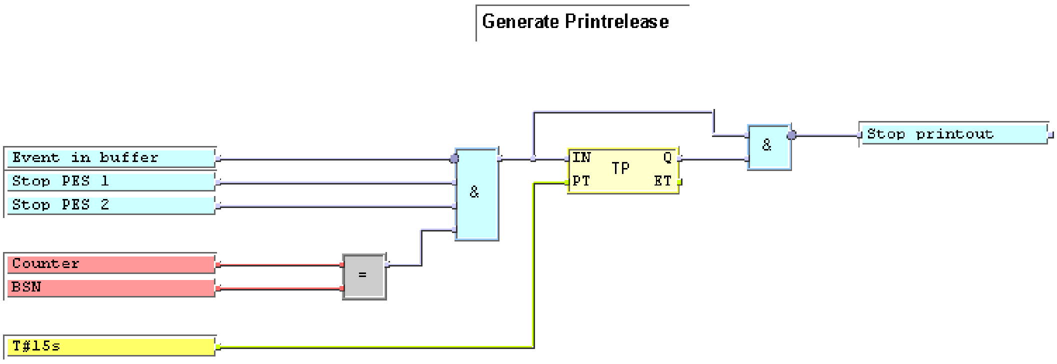
Type: UINT
The counter increments by 1 with each successful printer output. It goes back to 1 when it has reached 65535.

21.5 Program example

Printer for 3 PESs with a maximum print time of 15s per PES. Printer enable is generated in a central PES.

1st PES (Master)



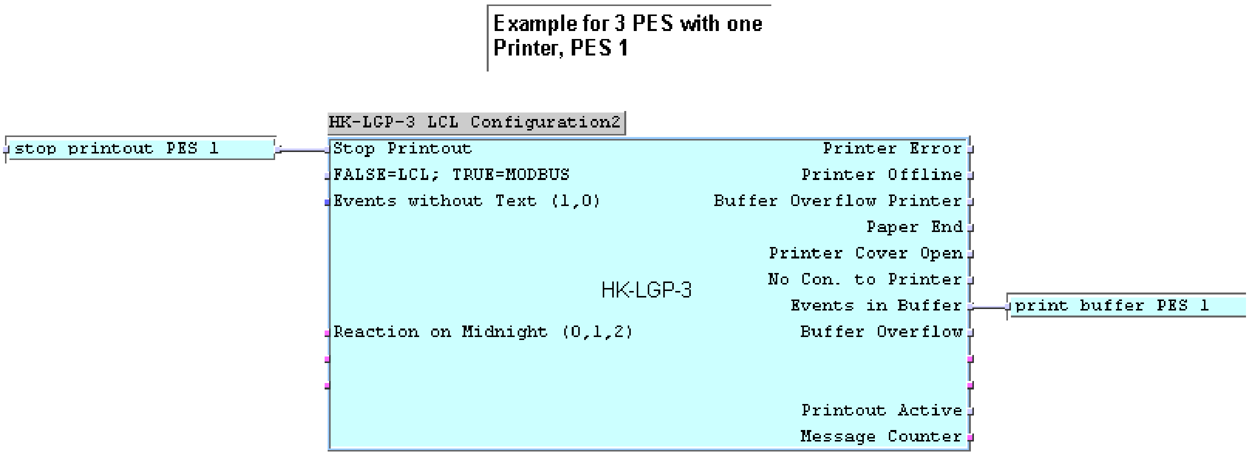


Block LGP_W

The EVENT signals must be transmitted from the slave systems to the master system. The STOPP1 and 2 signals must be transmitted from the master system to the individual slave systems. Transmission can be via

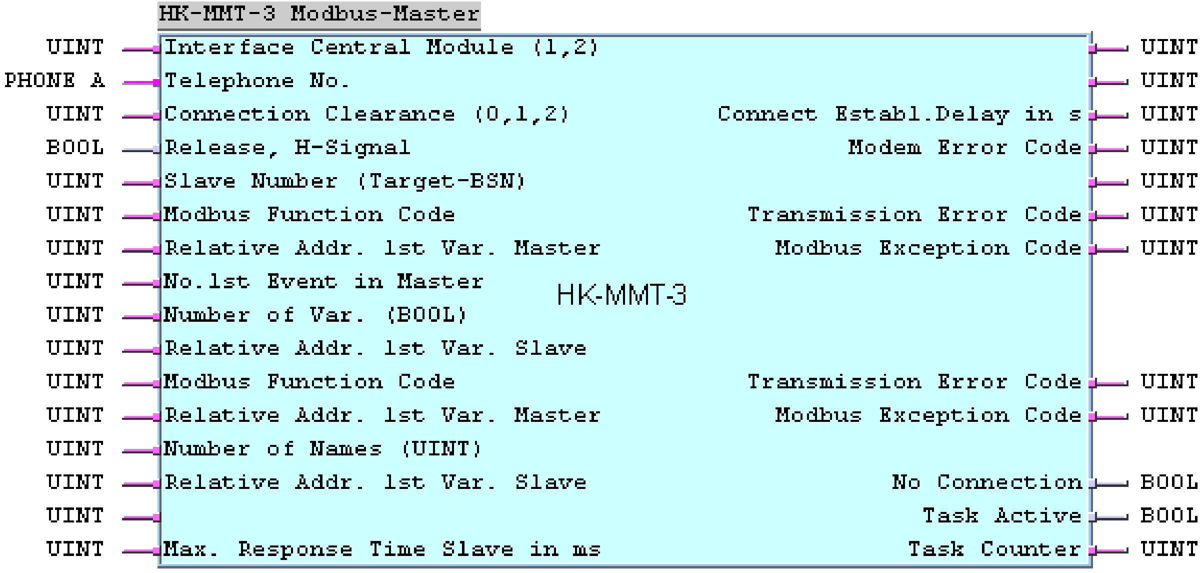
HIPRO or via input and output modules. **Block HK-LGP-3 must be used in each PES and assigned the appropriate signals**.

e.g. PES 1:



22HK-MMT-3

MODBUS Master



Inputs	Type
Central module interface	UINT
Telephone number	PHONE_A
Enable, TRUE	BOOL
Slave number (target bus stn.)	UINT
Modbus function code	UINT
Relative address 1st.var. master	UINT
No. 1st event master	UINT
Number var. (BOOL)	UINT
Relative address 1st var. slave	UINT
Modbus function code	UINT
Relative address 1st var. master	UINT
Number var. (UINT)	UINT
Relative address 1st var. slave	UINT
Max. slave answer time in ms	UINT
Outputs	
Establishment delay in s	UINT

table 47: Input and output specification

Modem error code	UINT
Error code transmission	UINT
Modbus error code transmission	UINT
Error code transmission (UINT)	UINT
Modbus error code transmission	UINT
No connection	BOOL
Task active	BOOL
Task counter	UINT
Function information	Short information
Block information	General block information
General	Description of block function
Applications	
Time consideration	

table 47: Input and output specification

22.1 Function information

MODBUS Master Block

Call establishment by telephone

Error monitoring of the MODBUS

22.2 General block information

Valid from operating system	BS41/51 V6.0-6 (9636) BS51-M, H, HR V6.0-6 (9636) BS11 V6.0-6 (9636) BS51 A1 V6.0-6 (9636) BS41q/51q V7.0-7 (9737)
Reload (load program), change of the input and output assignment	Permitted
Use in safety controllers with TÜV test	Yes (but not suitable for safety-oriented data transmission!)

Assignment rule	The following inputs must be assigned: Central module interface (1,2) Telephone number Release, TRUE Slave number (target bus station) Max. slave response time in ms, plus one of the ‘Modbus function code’ inputs. If the ‘Modbus function code’ input is assigned, the next three inputs must also be assigned.

table 48: General block information

22.3 GENERAL

This block allows an H11, H41 or H51 PES to be used as a MODBUS master. MODBUS masters can send data to MODBUS slaves or request data from MODBUS slaves independently.
The MODBUS master function is implemented for both interfaces of the central module. If the master function is not active, the corresponding interface is available as a MODBUS slave, i.e. master and slave functions are possible on the same interface.
Only one MODBUS master can be active on any one bus. However with the use of suitable logic it is possible to switch between several MODBUS masters on the bus.
With appropriate configuration the block can also be used for automatic connection (call) establishment with a suitable telephone modem. The block is not suitable for safety-oriented data transfers; safety-oriented HIPRO communication is provided for this purpose.

Block functions
The block is designed for the serial interfaces of the central module. Unless you intend to use standard interface parameters (9600 bps or 57600 bps depending on the setting of DIP switches S1-8 on the central module), you can configure the interfaces with the SIO parameter setup window, and the settings will then be valid for all MODBUS master blocks that use that interface. One boolean and one unsigned integer MODBUS master transmission can be configured and initiated per block. It is executed when the block is enabled (see the **Enable, TRUE** input).
If both the boolean and the unsigned integer task are configured on a block, the boolean task is executed first, followed by the unsigned integer task. The block can be called more than once in the program for other send/request tasks. If all the used blocks are enabled, the tasks will be executed in the order of the block calls.

22.3.1 Potential applications

22.3.1.1 Direct connection:

Data interchange between HIMA PESs if the HIMA coprocessor module F8621 cannot be used as a PES master (e.g. H41 as master) or HIMA PES and external systems (external system as the slave).

A modem link is also possible:
A dedicated line can be used as a transmission link, but a telephone modem can also be used. The MODBUS master can itself dial and establish the connection.

22.3.1.2 Bus connection:

A HIMA PES as a master reads data and writes data in slaves on the bus. This configuration is advisable when coprocessor module F 8621 cannot be used as the PES master (e.g. H41 as master or external system as slave).

A modem interface is also possible (dedicated line or telephone modem).

NOTE
The system only supports modems with AT syntax. The block uses a subset of the AT command set:

ATDx..xDial a telephone number, x..x = telephone number
ATE0Echo off: Commands are not sent back
ATX0Messages 0 - 4, where 0 = OK
ATQ0Return of result messages
ATQ1No return of result messages
ATV0Output modem answerbacks in short form
ATHHang up: Clear a connection
ATS2=43Sign for mode change

22.3.1.3 Data transmission

Data are transmitted using the Gould MODBUS protocol. The following MODBUS function codes are implemented in the HK-MMT-3 block:

1	READ COIL STATUS
2	READ INPUT STATUS
3	READ HOLDING REGISTERS
4	READ INPUT REGISTERS
5	FORCE SINGLE COIL
6	PRESET SINGLE REGISTER
8	LOOPBACK DIAGNOSTIC TEST
15	FORCE MULTIPLE COILS
16	PRESET MULTIPLE REGISTERS

For a detailed description of the function codes and the protocol please refer to the Gould MODBUS Reference Manual.
The MODBUS master can import variable types BOOL and UINT from a slave controler; in the PES which holds the master the attribute ’BUSCOM Import’ must be assigned to the variables. It can also export these variable types to a slave controler; in the PES which holds the master the attribute ’BUSCOM Export’ must be assigned to the variables.
With HIMA slave systems the variables exported to the master must have the attribute ’BUSCOM Export’ and variables imported from the master must have the attribute ’BUSCOM Import’.

You will find the corresponding addresses for individual variables with HIMA in the MODBUS lists in the ELOP II-NT software package. They can be printed out (generated) with the Res-Docu resource documentation.

Please refer to the ‘Functions of the Operating System’ manual for the addresses with connections to earlier H41, H51 systems (pre-version 6.0-6).

22.3.1.4 Transmitting events

Events are boolean signal changes with a time stamp. They are stored in the PES in a special event buffer. These events can be transmitted by MODBUS Code 1 (READ COIL STATUS) and MODBUS Code 3 (READ HOLDING REGISTERS) (HIMA specific).
The MODBUS master can therefore read events from MODBUS slave systems. In conjunction with event evaluation by LcL (Logic controlled Logging) and PLESY II, this means that events from different MODBUS slaves can be collected by a MODBUS master and printed out centrally (see Application).

22.4 Notes on the assignment of inputs/outputs

22.4.1 Central module interface (1,2)

Type: UINT, **Values 1 and 2 only are permitted, this input must be assigned**.
Number of the serial interface of the central module across which the communication takes place. Only the central module interfaces are supported.
Caution: If you wish to use the MODBUS master block together with LGP, only interface 1 on the central module can be used because interface 2 of the central module is occupied by the report printer terminal.

22.4.2 Telephone number

Type: PHONE_A, **This input must be assigned.**
In the case of an existing connection (established phone link, dedicated (leased) line, bus system) the input with a PHONE_A type field is seized and an Underscore (value 95) or Blank (value 32) assigned to the first element of the field. If you wish to establish an automatic modem link before the MODBUS communication however, the PHONE_A type field must contain the phone number to be dialed. The end of the phone number must be marked by the Underscore or Blank.
See Application for entering a telephone number.

22.4.3 Type of call establishment (0,1,2)

Type: UINT; **Values 0, 1, 2 only are permitted.**
This input can be used to determine the response of the block to a call clearance:

0	an existing call is cleared immediately if no task is present
1	an existing call is cleared after one minute if no task is present
2	an existing call is not cleared.

If the input is not assigned the default response is as 0.

22.4.4 Enable, TRUE

Type: BOOL, **This input must be assigned**.

TRUE at this input initiates a task provided the block is ready to receive. The block is ready to receive when a connection is established and any previous tasks have been executed (‘Task active‘ [Auftrag aktiv] output = FALSE).

If there is no connection, the block tries to establish one, and the task is executed when the call is successfully established. If the block is still busy with a task, the TRUE of the enable is ignored for this period.

22.4.5 Slave number (target bus stn.)

Type: UINT, **This input must be assigned**.

This input contains the address of the addressed MODBUS slave. Values from 0 to 255 are possible.

Address 0 is used for broadcast messages. Broadcast messages are transmissions to all slaves simultaneously. The slaves do not send an answer to the master in this case. Broadcast transmissions are possible with MODBUS function codes 5, 6, 15 and 16.

22.4.6 Modbus function code

Type: UINT, **Values 1, 2, 3, 5 and 15 only are permitted. This input must be assigned for boolean tasks**.

Function code for write or read transmissions of boolean variables. The following codes can be used:

1	READ COIL STATUS with HIMA, read from: boolean variables with attribute BUSCOM Export status of events (without time stamp)
2	READ INPUT STATUS with HIMA, read from: boolean variables from earlier H41, H51 systems (pre V6.0-6)
3	READ HOLDING REGISTERS with HIMA, read from: Events with a time stamp
5	FORCE SINGLE COIL with HIMA, write from: boolean variables with attribute BUSCOM Export
15	FORCE MULTIPLE COILS with HIMA, write from: several boolean variables with attribute BUSCOM Export

22.4.7 Relative address 1st var. master

Type: UINT

Address of the variable in the master system. See BUSCOM list of ELOP II-NT. This list is printed out (generated) with the Res-Docu resource documentation.

When events are transmitted, event number 0000 corresponds to relative address 2048 (master).

22.4.8 No. 1st event in master

Type: UINT
The number in the master for the 1st event exported from the slave. The event number can be taken from the BUSCOM list of ELOP II-NT. The list is printed out (generated) with the Res-Docu resource documentation.
Please refer to the ‘Functions of the Operating System’ manual for the addresses with connections to earlier H41, H51 systems (pre-version 6.0-6).

22.4.9 Number var. (BOOL)

Type: UINT
The number of boolean variables that are to be imported or exported.

Special cases:
Function code 5: Precisely one value is processed whatever the number entered here.
Caution: Number = 0 causes an error code.
Function code 3: because an event with a time stamp consists of several bytes, the number must be calculated (see Application).

22.4.10 Relative address 1st var. slave

Type: UINT
The target address or source address in the slave starting from which data are to be imported or exported. If the slave is a HIMA system, the assignments of the addresses to the individual names will be found in the BUSCOM list of ELOP II-NT. The list is printed out (generated) with the Res-Docu resource documentation.
Please refer to the ‘Functions of the Operating System’ manual for the addresses with connections to earlier H41, H51 systems (pre-version 6.0-6).
If the system is an external system, you should consult its documentation.

22.4.11 Modbus function code

Type: UINT, **Values 3, 4, 6 and 16 only are permitted, this input must be assigned for unsigned integer tasks.**
Function code for write or read transmissions for integer data.
The following codes can be used:

3	READ HOLDING REGISTERS with HIMA, read from: Variables (UINT) with attribute BUSCOM Import
4	READ INPUT REGISTERS with HIMA, read from: Variables (UINT) of earlier H41, H51 systems
6	PRESET SINGLE REGISTER with HIMA, write from: individual variables (UINT) with attribute BUSCOM Import
16	PRESET MULTIPLE REGISTERS with HIMA, write from: several variables (UINT) with attribute BUSCOM Import

22.4.12 Relative address 1st var. master

Type: UINT

The address of the variable in the master system. See MODBUS list of ELOP II. The list is printed out (generated) with the Res-Docu resource documentation.

22.4.13 Number var. (UINT)

Type: UINT
Number of unsigned integer variables to be imported or exported.
Special case:
Function code 6: precisely one value is processed whatever the number entered here.
Caution: Number = 0 causes an error code.

22.4.14 Relative address 1st var. slave

Type: UINT
The target address or source address in the slave starting from which data are to be imported or exported. If the slave is a HIMA system, the assignments of the addresses to the individual names will be found in the MODBUS list of ELOP II. The list is printed out (generated) with the Res-Docu resource documentation.
Please refer to the ‘Functions of the Operating System’ manual for the addresses with connections to earlier H41, H51 systems (pre-version 6.0-6).
If the system is an external system, you should consult its documentation.

22.4.15 Max. slave answer time in ms

Type: UINT, **This input must be assigned**.
The maximum time within which the master must have received the answer from the slave must be entered here. If no answer is received inside the time, the transmission is repeated twice, i.e. an error is output at the **No connection** output after a total of three time-outs. Time monitoring starts when the first character in the master transmission is sent, i.e. in the case of modem communication, after the call is established. The answer time (TA) consists of the actual transfer time (TÜ) and the processing times in the master and slave. The transfer time can be seen from the table below - it is calculated on the basis of 11 bits per character transmitted (1 start bit, 8 data bits, 1 parity and 1 stop bit) and a maximum message length of 256 characters.

Calculation of answer time:

HIMA PES as slave:

Read task: $T_A = T_Ü + ZZ_M$
Write task: $T_A = T_Ü + ZZ_M + ZZ_S$

External system as slave:
$T_A = T_Ü + ZZ_M + TX$

where:

T_A = answer time in ms
$T_Ü$ = transfer time according to table
ZZ_M = cycle time of master in ms

ZZ_S = cycle time of HIMA slave in ms
T_X = processing time of external system in ms (see documentation of external system)

b	300	600	1200	2400	4800	9600	19200	57600
ms	9600	4800	2400	1200	600	300	150	50

table 49: Transfer time TÜ in ms for baud rates 300 to 57600

Entered values over 60 000 are limited to 60 000. If both tasks (BOOL and UINT) of the block are used, the longer answer time must be entered if applicable.

22.4.16 Establishment delay in s

Type: UINT
This output is only active when a telephone number is specified (‘Telephone number’ input). It outputs the time that is needed to establish the connection. If no telephone number is configured (Blank or Underscore as the 1st character), the output carries 0.

22.4.17 Modem error code

Type: UINT
This output carries the error code that is output by the modem in the case of errors. The error codes are explained in the Modem Manual.

22.4.18 Error code transmission

Type: UINT
Error code for block errors (1-digit) and errors in boolean task processing.
0 = no error
1 = incorrect interface (only 1 or 2 possible)
2 = incorrect slave number (number >255)
3 = incorrect dialing response (value >2)
4= address of an exported event exceeds the event range of the master (not enough event names defined in the master)
11= incorrect function code with BOOL task
12= incorrect number or incorrect relative address in the master with BOOL task
13= incorrect relative address, address in safety-oriented range with BOOL task
15= unexpected answer from slave with BOOL task

22.4.19 Modbus error code transmission

Type: UINT
Output of the error code as supplied in the answer message from the slave. Please refer to the MODBUS Manual and the description of the external system for an explanation and interpretation of the error code (for HIMA see ‘Functions of the Operating Systems’ manual).

22.4.20 Error code transmission

Type: UINT
Error code for block errors (1-digit) and errors in UINT task processing.

0 =	no error
1 =	incorrect interface (only 1 or 2 possible)
2 =	incorrect slave number (number >255)
3 =	incorrect dialing response (value >2)
21=	incorrect function code with UINT task
22=	incorrect number or incorrect relative address in the master with UINT task
23=	incorrect relative address, address in safety-oriented range with UINT task
24=	incorrect diagnostic code with MODBUS Code 8
25=	unexpected answer from slave with UINT task.

22.4.21 Modbus error code transmission

Type: UINT
Output of the error code as supplied in the answer message from the slave. Please refer to the MODBUS Manual and the description of the external system for an explanation and interpretation of the error code (for HIMA see ‘Functions of the Operating Systems’ manual).

22.4.22 No connection

Type: BOOL
The output carries TRUE when a master transmission has not been answered or has been incorrectly answered after the third request. If there is no connection between master and slave, the output carries TRUE after three times the time specified at the **Max. slave answer time in ms** input. The output also carries TRUE after an existing call is cleared according to the dialing response configured at the **Type of call establishment (0, 1, 2)** input.
Irrespective of the dialing response (see **Type of call establishment (0, 1, 2)** input) the call is cleared after the third incorrect answer in succession. The output is reset after the next correct task execution. The cause of the error may be a faulty or cleared link between the two modems or a fault in the connection between modem and slave.

22.4.23 Task active

Type: BOOL
This output carries TRUE when the task is being processed (with repeat attempts if required) or when several blocks are waiting for the enable and a task has been accepted. While the output is carrying TRUE no input is read in, i.e. not even the enable signal.

22.4.24 Task counter

Type: UINT
The counter increments by 1 for each executed task. It is set to 0 on cold start with initialising and without initialising.

22.5 Applications

22.5.1 Input of the telephone number

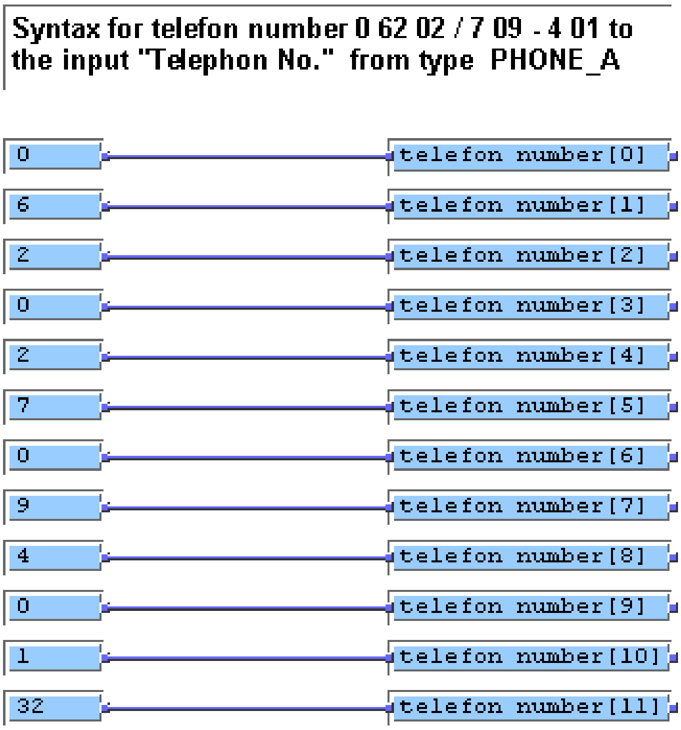
The telephone number can be up to 32 characters long. Shorter numbers must end with a terminating character. The terminating character is either a Blank (ASCII code 32) or an Underscore (ASCII code 95).
As well as the digits, the telephone number may contain other special characters which have the following meanings:

P	= pulse dialing from here
T	= frequency dialing from here
W	= waiting for dialing tone
>	= press earth button (the b-contact of the line is connected to earth for 300 ms to get a line from outside into the PABX. Must be given as the first character if necessary)
!	= Flash (the line is interrupted for 80 ms to get a line from outside into the PABX. Must be given as the first character if necessary)
_	(Underscore) or Blank: = terminating character to end the telephone number

See also ‘ATD Command’ in the Modem Manual for the use of special characters.

The telephone number can be entered in the PES as follows:

Configuration in ELOP II-NT



Syntax for telefon number 0 62 02 / 7 09 - 4 01 to
the input "Telephon No." from type PHONE_A

```
0     telefon number[0]
6     telefon number[1]
2     telefon number[2]
0     telefon number[3]
2     telefon number[4]
7     telefon number[5]
0     telefon number[6]
9     telefon number[7]
4     telefon number[8]
0     telefon number[9]
1     telefon number[10]
32    telefon number[11]
```

In ELOP II-NT the telephone number is entered by assigning it to the field variables. The digits are entered as digits, characters and terminating characters are entered as ASCII code.

Character	P	T	W	>	!	_	" "
Entry	80	84	87	62	33	95	32

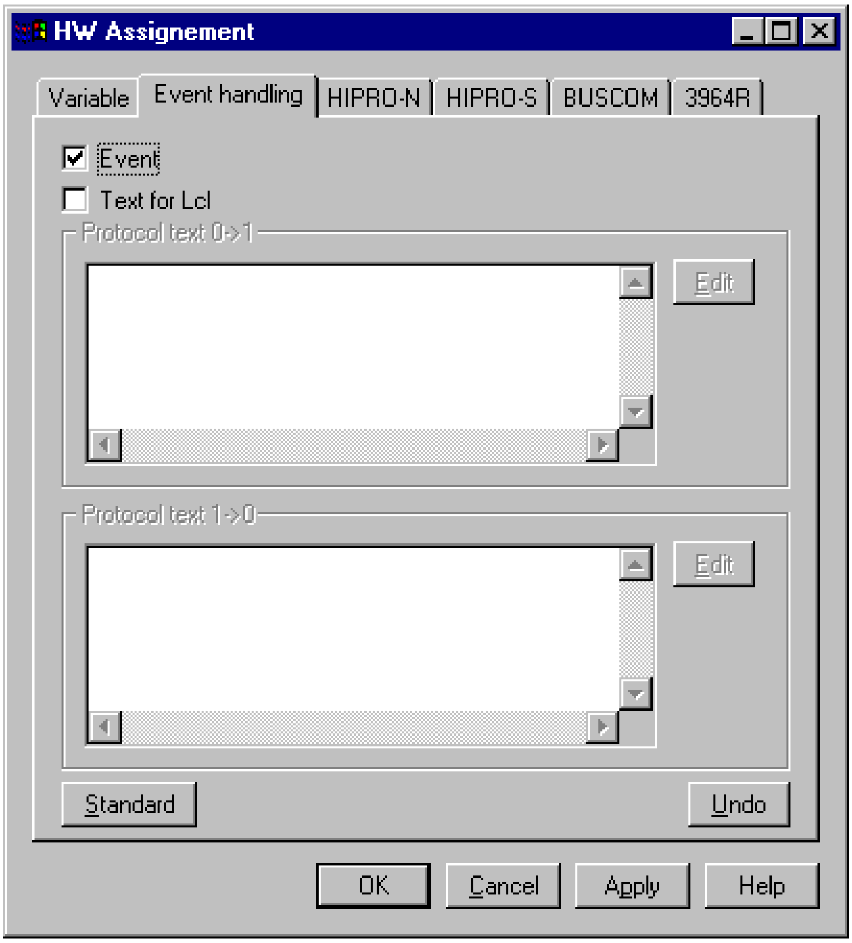
table 50: Extract from ASCII Table (Special characters)

22.5.2 Entering events

Events are digital (boolean) signal changes with a time stamp. They can be exported by the MODBUS master from slave PESs (HIMA).
In the PES events are stored in an event buffer. By reading with function code 1 or 3 starting from certain <->relative addresses, the status of the events or events with time stamp can be exported from the slave.
The relative addresses to be used can be taken from the 'Functions of the Operating Systems' manual or from MODBUS list ELOP II. The list is printed out (generated) with the Res-Docu resource documentation.
The following application assumes a slave of a PES H41 or H51 with H41/H51 V6.0-6 as the operating system. The task is to export events with a time stamp from this PES.

Configuring in the slave

In the slave the individual variables are assigned the attribute "event-driven" out of the properties during hardware assignment.



☑ Event

□ Text for Lcl

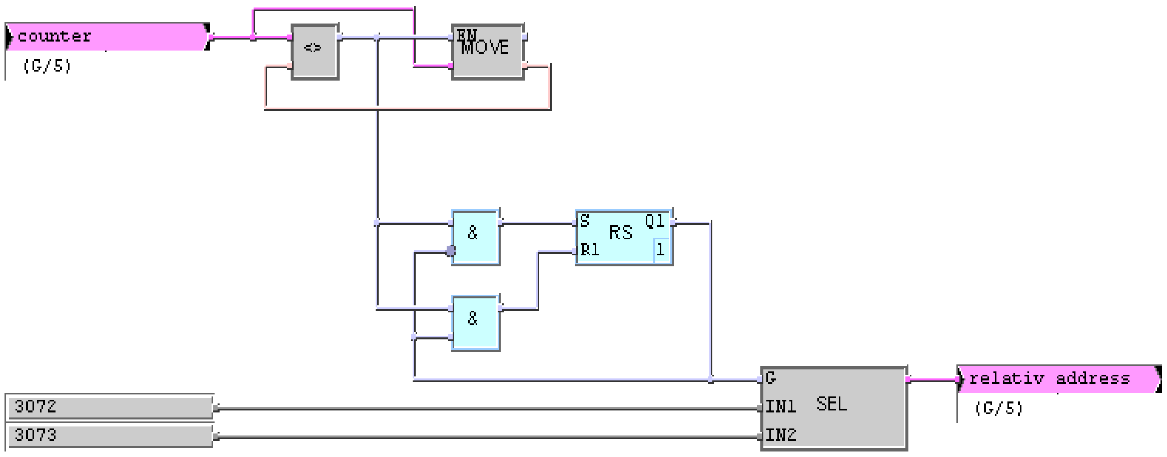
In case of an overflow in the slave PES, a certain event is generated that can be evaluated in the master:
The event number is 0 (very first event), the event value is TRUE (coming event) and the time is 99.99.99.xxx, with xxx representing the bus station number of the slave from which the overflow label came. If this overflow event also defined in the master, then defined text can be output about it with LGP. PLESY II masks out the overflow event from the slave automatically and outputs a system message. That is why these overflow events do not need to be explicitly agreed for PLESY II.

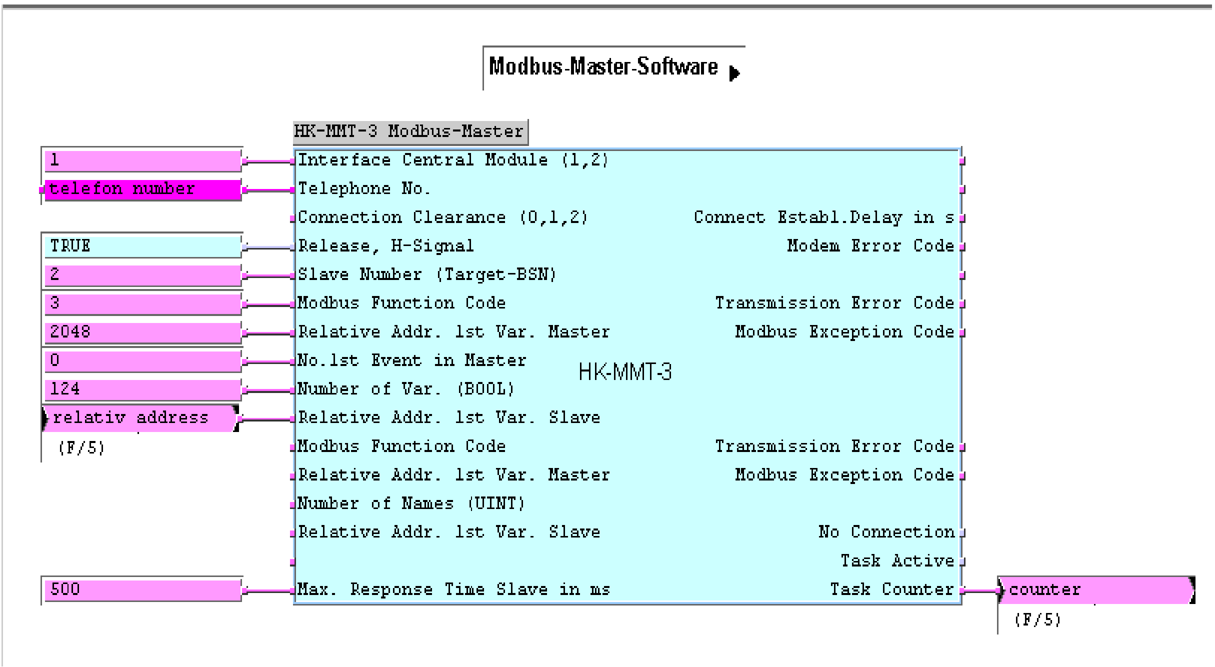
Configuring in the master

In the master the individual variables are assigned the attribute “event-driven” out of the properties during hardware assignment.

Logic:
To allow new events to be exported from the slave, the master must interrogate with varying addresses. If the master always interrogated the same address, the slave would interpret this as if the previous events had not been received and the old events would be repeated.
Interrogation with varying addresses must be programmed with a small pre-logic. The following logic makes use of the fact that the task counter on the block increments following the correct execution of the task. This ensures that the address is only changed when the previous events have been received, thus allowing new events to be fetched.

Address-Switching to Read Events





Notes:

- 1. Exporting alternately from addresses 3072 and 3073 is sufficient for new events to be read, as implemented in the above logic.
- 2. An event consists of a total of 8 bytes (address + signal change + time information). Because a UINT variable has two bytes, four variables are required per event.
- In the example above, the number given is the maximum number for events (124 = 31 events) per transmission; this is a function of the maximum message length of 256 bytes in the MODBUS report. A smaller number can be exported if there are fewer events; this makes the messages shorter and saves on transfer time. This should be tried out by tests in individual cases.
- 4. Variable values in the report text are added with their current value at the time when the event is entered in the master PES. Variable values from the slaves are not transferred automatically, if necessary they should be transferred in advance by the MODBUS to the master.
- Transferring additional gates is not possible.½hlung2

Reading events from several slaves

The appropriate number of event-driven boolean variables (hardware assignment) must be provided sequentially numbered and for each slave in the master. The event number can be taken from the MODBUS list of ELOP II. The list is printed out (generated) with the Res-Docu resource documentation.

A Modbus master block must be provided for each slave, also the logic for the address changeover.

The event number of the boolean variables that corresponds to the 1st event in the slave must be entered in the master at the **Event No. 1 event in the master** input.

22.5.3 Time consideration

The bus cycle time and response time can be calculated approximately using the following information (assuming a trouble-free transfer!).
If the master is continuously processing all tasks (the ENABLE [FREIGABE] input is always TRUE), then the average cycle time is the time needed to process all tasks once. The response time indicates how much time is needed until inputs of one system (master or slave) become effective as outputs in the other system.

The new transfer time should be calculated first with Table 3. The transfer time is the net transmission time needed for the master’s transmission and the slave’s answer. The two most common baud rates, 9600 bps and 57 600bps, are used, the factors given (1.15 and 0.19) relate to 11 bits per transmitted character (1 start bit, 8 data bits, 1 parity bit and 1 stop bit). Corresponding factors for other baud rates are given in the table below.

The following abbreviations are used:

$T_Ü$	Transfer time, includes time for master transmission and answerback from slave
T_X	Time until slave answers
ZZ_M	Cycle time of master
ZZ_S	Cycle time of slave
T_R	Response time between export inputs and import outputs on the other controller
BZZ_L	Bus cycle time with a read transmission of the master
BZZ_S	Bus cycle time with a write transmission of the master
BZZ_G	Total bus cycle time with several tasks

Typ	Code	Value	No. of characters	$T_Ü$ at 9600	TÜ at 57600
BOOL	1	n	x = n : 8 + 13	x . 1,15	x . 0.19
BOOL	2	n	x = n : 8 + 13	x . 1,15	x . 0.19
Event	3	n	x = n . 2 +13	x . 1,15	x . 0.19
UINT	3	m	x = m . 2 + 13	x . 1,15	x . 0.19
UINT	4	m	x = m . 2 +13	x . 1,15	x . 0.19
BOOL	5	1	x = 16	x . 1,15	x . 0.19
UINT	6	1	x = 16	x . 1,15	x . 0.19
BOOL	15	n	x = n : 8 + 17	x . 1,15	x . 0.19
UINT	16	m	x = m . 2 + 17	x . 1,15	x . 0.19

table 51: Calculation of transfer time

bps	300	600	1200	2400	4800	9600	19200	57600
ms	36.7	18.3	9.2	4.6	2.3	1.15	0.57	0.19

table 52: Transfer time per character

Bus cycle time (one task)

General:
BZZ =TÜ + TX +2 . ZZM

The processing time TX in the slave is not known and must be taken from the relevant documentation.
Where a HIMA PES (H41 or H51) is used as the slave, TX is known, so the bus cycle time can be calculated as follows:

Bus cycle time for a read transmission from the master:
BZZL = TÜ + 5 ms + 2 . ZZM

Bus cycle time for a write transmission from the master:
BZZS = TÜ + 0.5 . ZZS + 2 . ZZM

Total bus cycle time

The total bus cycle time for several tasks is the sum of the bus cycle times for each task. With several tasks however the next task is started immediately, whereas with a single task the system must wait for the beginning of the next cycle. This shortens the total time by a certain amount:

Number of tasks: n
Reduction in total time: (n- 1) . ZZM
Example:
2 read tasks and 2 write tasks are processed:

BZZG = BZZL1 + BZZL2 + BZZS1 + BZZS2 - 3 . ZZM

Response time

The processing times in external systems used as slaves are not known, so the following statements refer to HIMA PESs (H41, H51) only. Different times of external systems must be adjusted accordingly.
Specifically, these are:

- Export inputs of the slave
- Receive by slave and response
- Receive by slave and acknowledge
- Write the outputs in the slave

Export slave inputs, import master outputs

The task is to export the inputs of the slave controller and set the outputs of the master controller.

Complete time sequence:
Export inputs of slave: 0.5 ZZS
Master waits for transfer: 0.5 BZZL
Message transfer time: TÜ
Slave receives and responds: 5 ms
Master receives answer:0.5 ZZM
Write the outputs: ZZM

Formula:

TR = 0.5 ZZS + 0.5 BZZL + TÜ + 5ms + 1.5 ZZM

Export master inputs, import slave outputs

The task is to export the inputs of the master controller and set the outputs of the slave controller.

Complete time sequence:

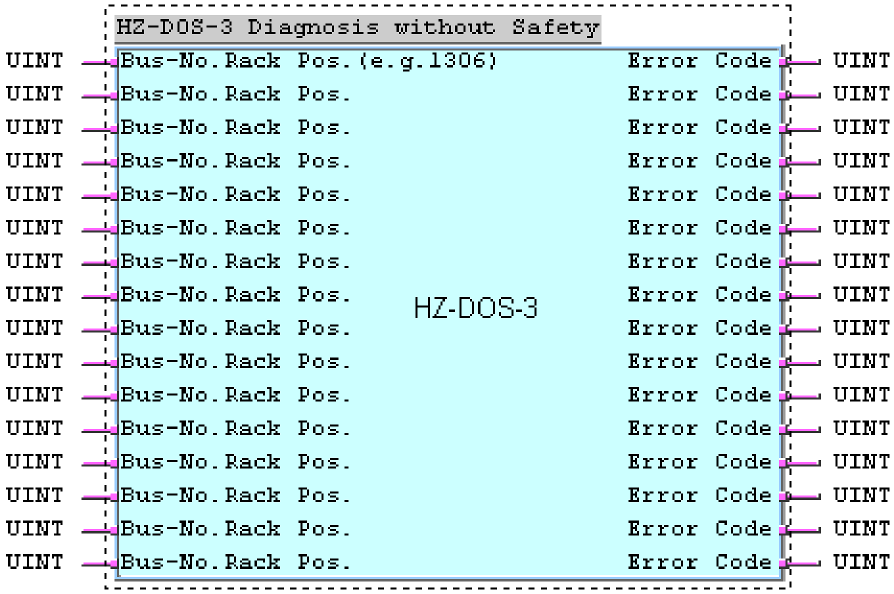
Export inputs of master: 0.5 ZZM
Master waits for transfer: 0.5 BZZS
Message transfer time: TÜ
Slave receives and responds: 0.5 ZZS
Write the outputs in the slave: ZZS

Formula:

TR = 0.5 ZZM + 0.5 BZZS +TÜ + 1.5 ZZS

23HZ-DOS-3

Diagnosis without safety


HZ-DOS-3 Diagnosis without Safety
UINT Bus-No.Rack Pos.(e.g.1306) Error Code UINT
UINT Bus-No.Rack Pos. Error Code UINT
UINT Bus-No.Rack Pos. Error Code UINT
UINT Bus-No.Rack Pos. Error Code UINT
UINT Bus-No.Rack Pos. Error Code UINT
UINT Bus-No.Rack Pos. Error Code UINT
UINT Bus-No.Rack Pos. Error Code UINT
UINT Bus-No.Rack Pos. Error Code UINT
UINT Bus-No.Rack Pos. Error Code UINT
UINT Bus-No.Rack Pos. Error Code UINT
UINT Bus-No.Rack Pos. Error Code UINT
UINT Bus-No.Rack Pos. Error Code UINT
UINT Bus-No.Rack Pos. Error Code UINT
UINT Bus-No.Rack Pos. Error Code UINT
UINT Bus-No.Rack Pos. Error Code UINT
UINT Bus-No.Rack Pos. Error Code UINT
HZ-DOS-3

Inputs	Type
Bus No. Rack Pos.	UINT
Outputs	
Error code	UINT
Function information	Short information
Block information	General block information
General	Description of block function

table 53: Input and output specification

23.1 Function information

Diagnostic mode call for modules F3235, F3236, F3237, F3238, F3330, F3331, F3333, F3334, F6213, F6214, F6217 and 6705.

Displays all faulty modules without shutdown

23.2 General block information

Valid from operating system	BS 41/51 V6.0-6 (9636) BS51-M, H, HR V6.0-6 (9636) BS51 A1 V6.0-6 (9636) BS41q/51q V7.0-7 (9737)
Special features	Block may only be used in the program type
Reload (load program), change the input and output assignment	Permitted, acknowledge required
Use for safety controllers with TÜV test	Permitted, **but not for modules in safety circuits, and block outputs may not be used for safety-relevant logic**

table 54: General block information

23.3 GENERAL

This block defines the input and output modules that are operated in diagnostic mode. The modules F3235, F3236, F3237, F3238, F3330, F3331, F3333, F3334, F6213, F6214, F6217 and 6705 can be set in diagnostic mode.

Diagnostic mode means that faulty modules are reported, even if the fault is caused by a short circuit, but are not turned off. Input values are not set to 0 therefore. The input values and the channel error information that is actually read is processed. Output modules are not turned off.

The block is used to optimise the availability of non-redundant modules in non-safety relevant circuits.

The following points should be remembered when using the diagnostic mode:

- Modules in the diagnostic mode must not be used in safety-relevant circuits.
- The block outputs must not be used for safety-relevant logic.
- Use is only possible for modules in I/O subrack B 9302.
- Faulty modules are only reported, they are not shut down.
- Input values are not set to 0.
- All the read values of faulty modules can be corrupted.
- Faulty output modules are not turned off by the integrated safety shutdown in the diagnostic mode; there is no shutdown by the connecting module in the event of an internal double error.
- The diagnostic mode precludes the redundant processing of input values with blocks HA-RTE-3 and HB-RTE-3.
- A group shutdown using block H8-STA-3 is not possible for modules in the diagnostic mode.

23.4 Notes on the assignment of inputs/outputs

23.4.1 Bus No. Rack Pos. (e.g. 1306)

Type: UINT

Defines the testable modules that will be operated in the diagnostic mode. The position of the module is a 4-digit decimal number corresponding to the position of the board in the CABINET program section.

Example:	Cabinet (1-2):	1
	Subrack (1-8):	3
	Board position (1-16):	6
	Module position:	1306

The user must check whether a module is defined and testable at the position indicated.

23.4.2 Error code

Type: UINT

Displays the error status of the channels of the testable input and output module as an 8 bit value:

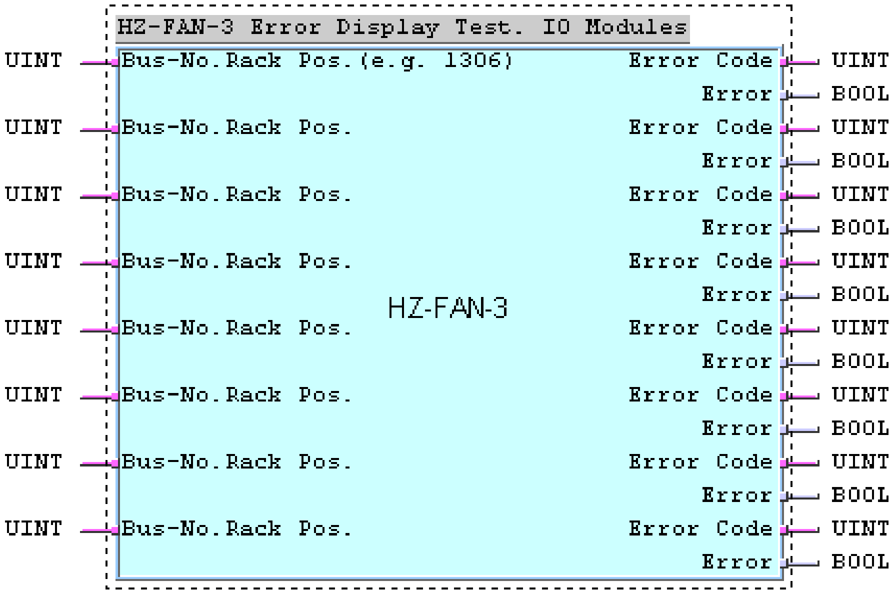
Bit 1...8	Value	Meaning
00 000 000	0	No error
00 000 001	1	Channel 1 of testable I/O module faulty
00 000 010	2	Channel 2 of testable I/O module faulty
00 000100	4	Channel 3 of testable I/O module faulty
00 001 000	8	Channel 4 of testable I/O module faulty
00 010 000	16	Channel 5 of testable I/O module faulty
00 100 000	32	Channel 6 of testable I/O module faulty
01 000 000	64	Channel 7 of testable I/O module faulty
10 000 000	128	Channel 8 of testable I/O module faulty
11 111 111	255	Testable I/O module faulty
11 111 111 11 111 111	65535	Invalid position at **Bus No. RACK Pos.** input

table 55: Error code

Where several channel errors exist, the decimal value which corresponds to the dual code is output. With modules which have no channel error detection, only the decimal values 0, 255 and 65535 are output if the assignment is incorrect.

24HZ-FAN-3

FAULT OF TESTABLE I/O MODULES



Inputs	Type
Bus No. Rack Pos.	UINT
Outputs	
Error code	UINT
Error	BOOL
Function information	Short information
Block information	General block information
General	Description of block function
Programming example	

table 56: Input and output specification

24.1 Function information

Error display for testable I/O modules

24.2 General block information

Valid from operating system	BS41/51 V6.0-6 (9636) BS51-M, H, HR V6.0-6 (9636) BS51 A1 V6.0-6 (9636) BS41q/51q V7.0-7 (9737
Reload (load program), change the input and output assignment	Permitted
Use for safety controllers with TÜV test	Permitted
Assignment rule	One of the input must be assigned

table 57: General block information

24.3 GENERAL

This block is used to detect faulty testable I/O modules. Specifically, it is needed for the redundant use of testable analog module F 6705 to detect the faulty module and to be able to drive the serviceable channel with 100% of the analog value (otherwise 50%, see also the data sheet for the F 6705). The changeover to 2 channels must be delayed by one cycle. The error output can also be used to isolate the supply voltage in the current sink mode, an option that is absolutely essential in safety-oriented (fail-safe) systems. See the Safety Manual.

Up to 8 I/O modules can be monitored with 1 block. The block can be used more than once in the ELOP program.

24.4 Notes on the assignment of inputs/outputs

24.4.1 Bus No. Rack Pos. (e.g. 1306)

Type: UINT
The positions of the testable I/O modules are entered as 4-digit decimal numbers according to the position of the board in the CABINET program section.

Example:	Cabinet (1-2):	1
	Subrack (1-8):	3
	Board position (1-16):	6
	Position of the unit:	1306

24.4.2 Error code

Type: UINT
Displays the channels of the corresponding testable I/O module on the input side as an 8 bit value (1...255) if faults are detected in the external current circuits of the channels (e.g. open circuit, line short). The set error bits are displayed as integers. Each bit corresponds to one channel, as the following table shows:

Bit 1...8	Value	Error location
00 000 000	0	No error
00 000 001	1	Fault in external circuit in channel 1
00 000 010	2	Fault in external circuit in channel 2
00 000 100	4	Fault in external circuit in channel 3
00 001 000	8	Fault in external circuit in channel 4
00 010 000	16	Fault in external circuit in channel 5
00 100 000	32	Fault in external circuit in channel 6
01 000 000	64	Fault in external circuit in channel 7
10 000 000	128	Fault in external circuit in channel 8
11 111 111	255	Module faulty

table 58: Error code

If more than one external circuit is faulty, a decimal value is output that corresponds to the dual code.

Example: Decimal value 18 corresponds to bit pattern 00 010 010; this means that the external circuits in channels 2 and 5 are faulty.

If 255 is displayed, then the actual module is faulty (modules without line diagnosis), not plugged in, not supplied with voltage or all the external circuits are faulty.

Faults in external current circuits can only be displayed when you use modules that have suitable integrated measuring stages for open circuit and short circuit, and when their software blocks for the I/O modules are used in the program.

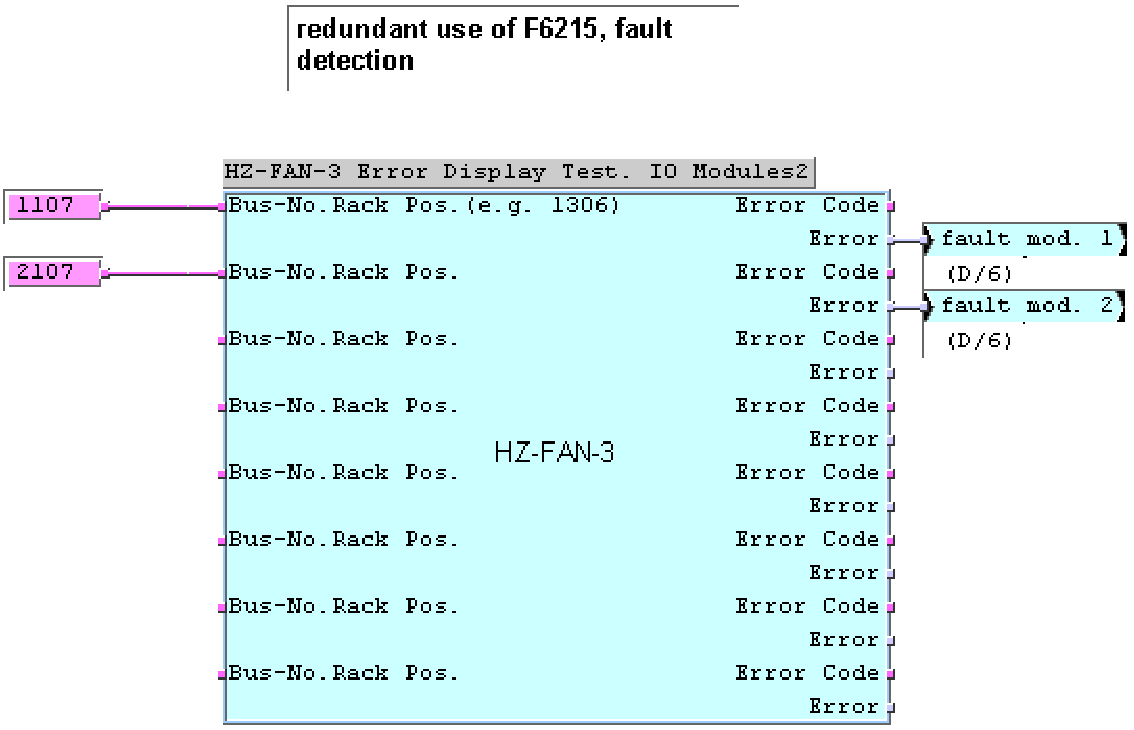
24.4.3 Error

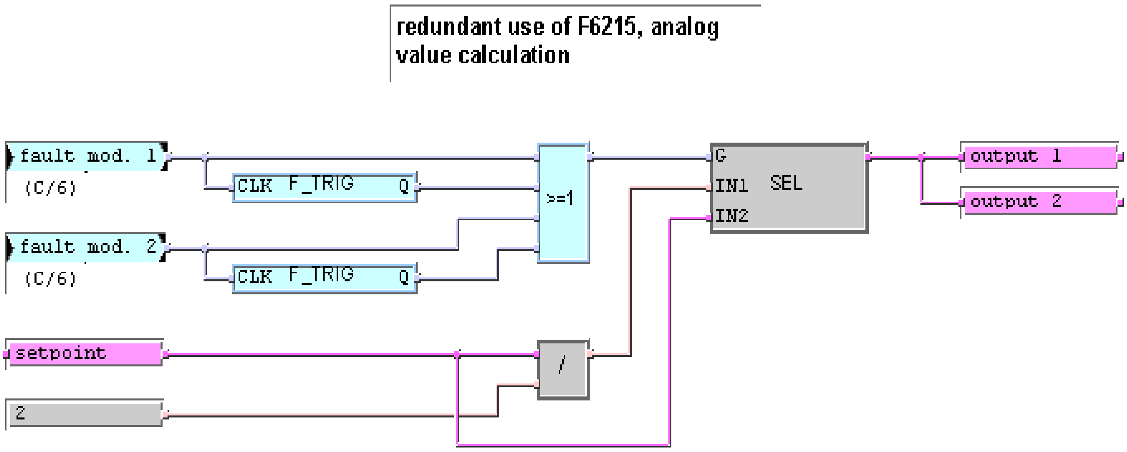
Type: BOOL
This output carries TRUE when there is an error in the corresponding module.
When negated, this block can be used to supply voltage for current loop F 6705 in the current-sink mode, in which case it is output by a testable output module. Alternatively it can be used to drive a fail-safe relay with which the voltage supply is switched.

24.5 Applications

Redundant used of the F 6705

With the redundant use of the F 6705, each of the two output channels sees 50% of the analog value (see the F 6705 data sheet). In the event of an error therefore it is necessary to switch the serviceable channel over to 100%. This can be done with the following logic.





Output1
Analog output 1 of the first analog output module

Output2
Analog output 1 of the redundant analog output module (redundant channel for output1)

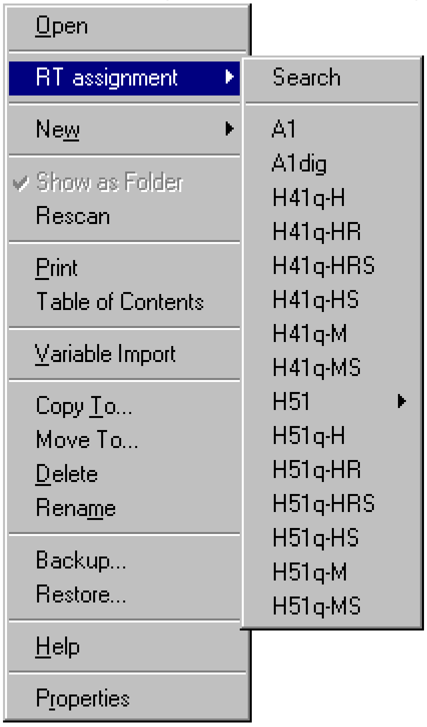
Lists and References

25 Resource Types

The selection of resource types is called from the resource context menu. Find the *RT Assignment* option if no resource is assigned yet. If a resource is already assigned, this option is replaced by the *Hardware Change* option.

25.1 Current Resource Types

The following current resource types are available:

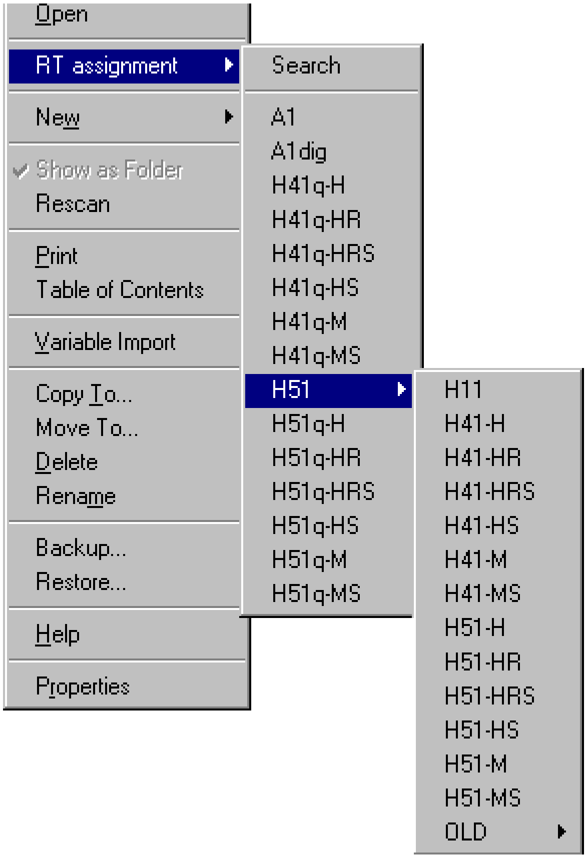


- A1
- A1dig
- H41q-H
- H41q-HR
- H41q-HRS
- H41q-HS
- H41q-M
- H41q-MS
- H51q-H
- H51q-HR
- H51q-HRS
- H51q-HS
- H51q-M
- H51q-MS

First selection in the RT assignment or with Hardware Change

25.2 Predecessor Resource Types

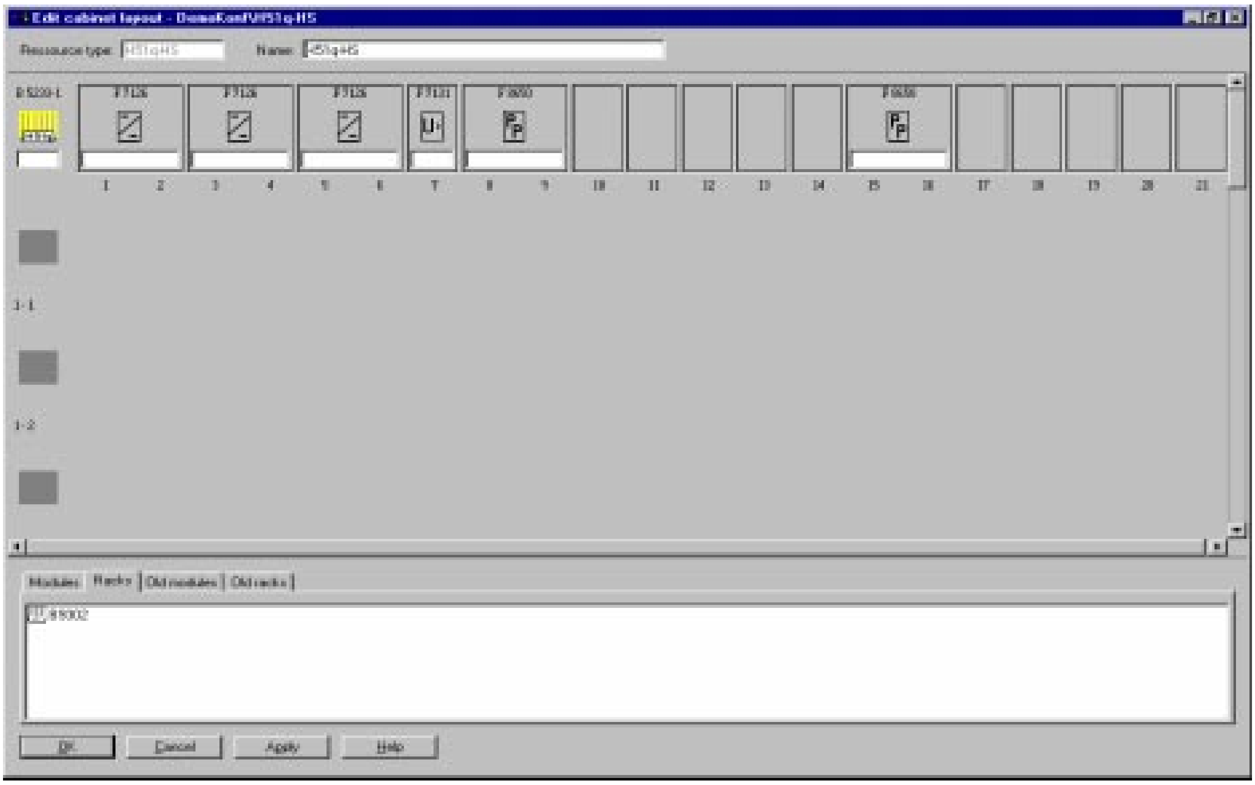
After the entry H51 you will find all the H41 and H51 systems that are operated with the V 6.0-6 operating system. They are:



- H11
- H41-H
- H41-HR
- H41-HRS
- H41-HS
- H41-M
- H41-MS
- H51-H
- H51-HR
- H51-HRS
- H51-HS
- H51-M
- H51-MS

26 Subracks

When using H51 or H51q systems, you must also define the required I/O subracks. This is done in the *Edit Cabinet* option of the resource.



Edit cabinet layout - DemoKonf\H51q-HS

Resource type: H51q-HS

Name: H51q-HS

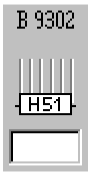
Edit cabinet

In the H11, H41, A1, A1dig and H41q systems there is no need to assign an I/O subrack because here the I/O modules belong to the central module or in the central subrack. No further extension is possible in these systems.

We distinguish between *Subracks* and *Old Subracks*.

26.1 Current subracks

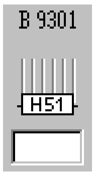
The I/O subrack B 9302 is available for the H51q and H51 systems.
To insert the subrack, Drag & Drop it to the left onto the grey box with the desired position.



You must use this subrack with new systems.

26.2 Old subracks

The B 9301 subrack is still available for H51 systems. This subrack cannot be used in the H51q system.



27 Modules

As with the subracks, we distinguish between *Modules* and *Old Modules*. Only the current modules in the *Modules* directory are used with new systems, of course.

27.1 Current modules

The current modules include:

- F 3221, 16x input module, digital (bool)
- F 3222, 8x input module, digital (bool) for proximity switches
- F 3223, 4x input module, digital (bool) for proximity switches in intrinsically safe circuits
- F 3224, 4x input module, digital (bool) for proximity switches in intrinsically safe circuits, with open-circuit monitoring
- F 3236, 16x input module, digital (bool), testable
- F 3237, 8x input module, digital (bool), testable, for proximity switches
- F 3238, 8x input module, digital (bool), testable, for proximity switches in intrinsically safe circuits, with line diagnosis
- F 3322, 16x output module, digital (bool), 500 mA (12 W)
- F 3330, 8x output module, digital (bool), testable, 500 mA (12 W)
- F 3331, 8x output module, digital (bool), testable, 500 mA (12 W), with line diagnosis
- F 3332, 4x output module, digital (bool), 2 A (48 W)
- F 3333, 4x output module, digital (bool), 2 A (48 W), testable
- F 3334, 4x output module, digital (bool), 2 A (48 W), testable, with line diagnosis
- F 3422, 8x relay module, switching voltage 60 V =/~
- F 6208, signal converter (Ex)i analog (word)
- F 6213, 4x input module, analog (word), testable
- F 6214, 4x input module, analog (word), testable, for transmitter
- F 6215, 8x input module, analog (word)
- F 6216, 8x input module, analog (word), for transmitter
- F 6217, 8x input module, analog (word), testable
- F 6705, 2x output module, analog (word), testable
- F 6706, 2x output module, analog (word)
- F 7126, power supply for H51, H51q, 24V/5V
- F 7130, power supply for H41, H41q, 24V/5V
- F 7131, power supply monitor for H51, H51q

- F 7132, 4x distribution board
- F 7133, 4x distribution board with fuse monitor
- F 7553, connecting module I/O bus for 9302
- F 8621, communication module, left CPU
- F 8621x, communication module, right CPU
- F 8625, ethernet module, left CPU
- F 8625x, ethernet module, right CPU
- F 8626, Profibus-DP module, left CPU
- F 8626x, Profibus-DP module, right CPU

Move the required module to the desired position in the desired subrack with Drag & Drop.

You do not need to define central modules, they are defined by their resource type assignment.

27.2 Old modules

These modules can only be used in H51 systems. They cannot be used in the H51q.
The following modules are available:

- F 3225, 16x input module, digital (bool), testable
- F 3227, 8x input module, digital (bool)
- F 3228, 16x input module, digital (bool)
- F 3235, 8x input module, digital (bool), testable, with line diagnosis
- F 3311, 16x output module, digital (bool), 200 mA
- F 3312, 4x output module, digital (bool), 1 A
- F 3313, 8x output module, digital (bool), testable, 400 mA
- F 3314, 4x output module, digital (bool), testable, 1 A
- F 3321, 16x output module, digital (bool), 500 mA
- F 3323, 8x output module, digital (bool), testable, 500 mA, with line diagnosis
- F 3412, 8x relay module
- F 3413, 8x relay module
- F 5202, 14 bit ring counter
- F 5203, 14 bit ring counter
- F 6103, input module, analog (word), for PT100 (Ex)i
- F 6204, input module, analog (word), (Ex)i
- F 6207, input module, analog (word), (Ex)i, for thermocouples
- F 6701, 2x output module, analog (word)
- F 7105, 6x distribution board with fuse monitor
- F 7129, 4x distribution board with fuse monitor
- F 7531, 6x distribution board
- F 7541, connecting module for B 9301

None of the old modules has integrated safety shutdown; they can only be used in subrack B 9301.

Use current modules only for new projects.

28 Supported IEC Functions and Data Types

IEC 61131-3 contains standard functions and standard blocks but they are not all supported. The level of support depends on the PES operating system that is used.

28.1 Supported data types

A distinction is made between operating systems with versions 6.0-6 and 7.0-7.
All HIQUAD systems, i.e. H41q and H51q, have OS version 7.0-7, all other systems have OS version 6.0-6.

Type	Version 6.0-6	Version 7.0-7
ANY		
ANY_NUM		
ANY_REAL		
REAL	available	available
LREAL		
ANY_INT		
SINT	available	available
INT	available	available
DINT		available
LINT		
USINT	available	available
UINT	available	available
UDINT		available
ULINT		
ANY_BIT		
BOOL	available	available
BYTE	available	available
WORD	available	available
DWORD		available
LWORD		
STRING		
ANY_DATE		
TIME_OF_DAY		available
DATE		available

Tabelle 59: Supported data types

DATE_AND_TIME		available
TIME	available	available

Tabelle 59: Supported data types

Empty boxes mean not available

28.2 Supported functions and function blocks

A distinction is made between operating systems with versions 6.0-6 and 7.0-7.
All HIQUAD systems, i.e. H41q and H51q, have OS version 7.0-7, all other systems have OS version 6.0-6.

Function	Version 6.0-6	Version 7.0-7
ANY_TO_BOOL		available
ANY_TO_BYTE		available
ANY_TO_WORD		available
ANY_TO_DWORD		except REAL
ANY_TO_LWORD		
ANY_TO_STRING		
ANY_TO_SINT		available
ANY_TO_INT	available	available
ANY_TO_DINT		except REAL
ANY_TO_LINT		
ANY_TO_USINT		available
ANY_TO_UINT	available	available
ANY_TO_DINT		except REAL
ANY_TO_ULINT		
ANY_TO_REAL	available	available
ANY_TO_LREAL		
ANY_TO_TIME_OF_DAY		except REAL
ANY_TO_DATE		except REAL
ANY_TO_DATE_AND_TIME		except REAL
ANY_TO_TIME		except REAL
TRUNC_SI		available

Tabelle 60: Supported functions and function modules

TRUNC_I	available	available
TRUNC_DI		
TRUNC_LI		
TRUNC_US		available
TRUNC_UI	available	available
TRUNC_UD		
TRUNC_UL		
BCD_TO_INT		available
BCD_TO_USINT		available
BCD_TO_UINT		available
BCD_TO_UDINT		available
INT_TO_BCD		available
USINT_TO_BCD		available
UINT_TO_BCD		available
UDINT_TO_BCD		available
ABS		available
MOVE	available	available
ADD	available	available
MUL	available	available
SUB	available	available
DIV	available	available
MOD		available
EXPT		
SQRT	available	available
LN	available	available
LOG		
EXP	available	available
SIN		
COS		
TAN		
ASIN		
ACOS		
ATAN		
SHL		available

Tabelle 60: Supported functions and function modules

SHR		available
ROL		available
ROR		available
AND	available	available
OR	available	available
XOR	available	available
NOT	available	available
SEL	available	available
MAX		
MIN		
LIMIT		available
MUX		available
GT	available	available
GE	available	available
EQ	available	available
LE	available	available
LT	available	available
LE	available	available
LEN		
LEFT		
RIGHT		
MID		
CONCAT		
INSERT		
DELETE		
REPLACE		
FIND		
ADD_TOD_T		available
ADD_DT_T		available
ADD_T_T		available
SUB_TOD_T		available
SUB_DT_T		available
SUB_T_T		available
SUB_TOD_TOD		available

Tabelle 60: Supported functions and function modules

SUB_DT_DT		available
SUB_T_T		available
MUL_T		except REAL
DIV_T		except REAL
CONCAT_D_TOD		available
SR	available	available
RS	available	available
SEMA	available	available
R_TRIG	available	available
F_TRIG	available	available
CTU		available
CTD		available
CTUD		available
TP	available	available
TON	available	available
TOF	available	available
RTC		available

Tabelle 60: Supported functions and function modules

29 System Variables

The system variables provide information from the system and are used to transfer information to the system. The following system variables are available:

- UINT SIO.ZB1: SIO1 receive counter
- UINT SIO.ZB1: SIO2 receive counter
- UINT SIO.ZG1/CB1: SIO1 receive counter
- UINT SIO.ZG1/CB1: SIO2 receive counter
- UINT SIO.ZG1/CB2: SIO1 receive counter
- UINT SIO.ZG1/CB2: SIO2 receive counter
- UINT SIO.ZG1/CB3: SIO1 receive counter
- UINT SIO.ZG1/CB3: SIO2 receive counter
- UINT SIO.ZB2: SIO1 receive counter
- UINT SIO.ZB2: SIO2 receive counter
- UINT SIO.ZG2/CB1: SIO1 receive counter
- UINT SIO.ZG2/CB1: SIO2 receive counter
- UINT SIO.ZG2/CB2: SIO1 receive counter
- UINT SIO.ZG2/CB2: SIO2 receive counter
- UINT SIO.ZG2/CB3: SIO1 receive counter
- UINT SIO.ZG2/CB3: SIO2 receive counter

- BOOL I/O.error acknowledgment
- UINT I/O.error code 2nd I/O bus
- UINT I/O.error position 2nd I/O bus
- UINT I/O.error code 1st I/O bus
- UINT I/O.error position 1st I/O bus
- BOOL I/O.error
- BOOL SYSTEM.Logic-Emergency stop
- UINT SYSTEM.RAM/EPROM
- UINT SYSTEM.Runversion
- UINT SYSTEM.Codeversion
- BOOL SYSTEM.Force individual switch outputs
- BOOL SYSTEM.Force individual switch inputs
- BOOL SYSTEM.Force main switch outputs
- BOOL SYSTEM.Force main switch inputs
- UINT SYSTEM.Number of prohibited accesses
- BOOL SYSTEM.Prohibited access
- BOOL SYSTEM.Single-channel
- UINT SYSTEM.Errormask2
- UINT SYSTEM.Errormask1
- UINT SYSTEM.Errorcode
- BOOL SYSTEM.normal

The system variables of the type UINT can also be assigned to a variable of the type WORD.

29.1 READ System variables of the BOOL type

READ system variables provide you with information from the operating system.

29.1.1 I/O Error

This system variable is TRUE when the operating system has detected one or more testable I/O modules as faulty. Use this system variable to display an I/O error.

29.1.2 HIBUS.Resource-Name.error

The system variable exists for each resource with safety-related communication via HIPRO. ´Resource-Name´ is replaced by the original resource name.
The variable is TRUE, if the resource does not receive any data inside the entered supervision time.

29.1.3 SYSTEM.Force individual switch outputs

This system variable is TRUE when at least one output variable is forced. An output variable is a variable to which an I&C name is assigned, i.e. it is assigned to an output module.

29.1.4 SYSTEM.Force individual switch inputs

This system variable is TRUE when at least one input variable is forced. An input variable is a variable to which an I&C name is assigned, i.e. it is assigned to an input module.

29.1.5 SYSTEM.Force main switch outputs

This system variable is TRUE when the FORCE main switch for outputs is on.

29.1.6 SYSTEM.Force main switch inputs

This system variable is TRUE when the FORCE main switch for inputs is on.

29.1.7 SYSTEM.Prohibited access

This system variable is TRUE for a cycle when an attempt has been made to run a prohibited function.
It is configured in the properties (safety) of the resource.

29.1.8 SYSTEM.Single-channel

This system variable is TRUE when one central module has failed in a system with two central modules.

29.1.9 SYSTEM.normal

This system variable is TRUE when no errors are present in the system. Use this system variable for a general status display of the system.

29.2 WRITE System variables of the BOOL type

You use WRITE system variables to transfer information to the operating system.

29.2.1 I/O Error acknowledgment

When this system variable is set to TRUE the I/O error is acknowledged. The following functions are performed:

- Acknowledge a displayed I/O error.
  Error display is reset and the system is checked again. If the system finds the error again, the position of the faulty module is displayed.
- Restart the test routines of the testable I/O modules that were shut down.

29.2.2 SYSTEM.Logic emergency stop

When this system variable is TRUE there is a general system shutdown. All outputs are de-energized and the system immediately enters the safe condition.

The system variable can be wired with an external signal or with a signal generated from the logic.
The system is set to RUN again by pressing the ACK button on the central modules.

29.3 READ System variables of the UINT/WORD type

READ system variables provide you with information from the operating system.
You can use either the UINT or WORD data type for the following system variables. Depending on the data type used, the value will subsequently be displayed in decimal or hexadecimal in the OLT field.

29.3.1 HIBUS.Resource-Name.Receive counter

The system variable exists for each resource with safety-related communication via HIPRO. ´Resource-Name´ is replaced by the original resource name.
The counter will be increased with each received transmission. The value has the range of 0...65535. After the value has reached the end value, the value starts again with 0.

29.3.2 SIO Receive Counter

All receive counters work as follows:
The receive counter is increment by 1 for every transmission received at this interface. The value range is from 1 to 65535, or 0001 to FFFF. The counter is reset to 1 when the maximum value is reached.

29.3.2.1 SIO.ZB1: SIO1-receive counter

Receive counter first interface of the left hand central module.

29.3.2.2 SIO.ZB1: SIO2-receive counter

Receive counter second interface of the left hand central module.

29.3.2.3 SIO.ZG1/CB1: SIO1 Receive counter

Receive counter first interface of the first coprocessor module that is assigned to the left hand central module.

29.3.2.4 SIO.ZG1/CB1: SIO2 Receive counter

Receive counter second interface of the first coprocessor module that is assigned to the left hand central module.

29.3.2.5 SIO.ZG1/CB2: SIO1 Receive counter

Receive counter first interface of the second coprocessor module that is assigned to the left hand central module.

29.3.2.6 SIO.ZG1/CB2: SIO2 Receive counter

Receive counter second interface of the second coprocessor module that is assigned to the left hand central module.

29.3.2.7 SIO.ZG1/CB3: SIO1 Receive counter

Receive counter first interface of the third coprocessor module that is assigned to the left hand central module.

29.3.2.8 SIO.ZG1/CB3: SIO2 Receive counter

Receive counter second interface of the third coprocessor module that is assigned to the left hand central module.

29.3.2.9 SIO.ZB2: SIO1-receive counter

Receive counter first interface of the right hand central module.

29.3.2.10SIO.ZB2: SIO2-receive counter

Receive counter second interface of the right hand central module.

29.3.2.11SIO.ZG2/CB1: SIO1 Receive counter

Receive counter first interface of the first coprocessor module that is assigned to the right hand central module.

29.3.2.12SIO.ZG2/CB1: SIO2 Receive counter

Receive counter second interface of the first coprocessor module that is assigned to the right hand central module.

29.3.2.13SIO.ZG2/CB2: SIO1 Receive counter

Receive counter first interface of the second coprocessor module that is assigned to the right hand central module.

29.3.2.14SIO.ZG2/CB2: SIO2 Receive counter

Receive counter second interface of the second coprocessor module that is assigned to the right hand central module.

29.3.2.15SIO.ZG2/CB3: SIO1 Receive counter

Receive counter first interface of the third coprocessor module that is assigned to the right hand central module.

29.3.2.16SIO.ZG2/CB3: SIO2 Receive counter

Receive counter second interface of the third coprocessor module that is assigned to the right hand central module.

29.3.3 I/O.Error code 2nd I/O bus

Displays the faulty channels of the module shown in system variable I/

O.Error position 2nd I/O bus. A display is only possible when the module has line diagnosis.
Value output:

Bit No. 1...8	Dec.	Hex	Error
00 000 000	0	0	No error
00 000 001	1	1	Error in circuit channel 1
00 000 010	2	2	Error in circuit channel 2
00 000 100	4	4	Error in circuit channel 3
00 001 000	8	8	Error in circuit channel 4
00 010 000	16	10	Error in circuit channel 5
00 100 000	32	20	Error in circuit channel 6
01 000 000	64	40	Error in circuit channel 7
10 000 000	128	80	Error in circuit channel 8
11 111 111	255	FF	Faulty module

Tabelle 61: Faulty channel display

If several external circuits are faulty then a corresponding value is displayed.

29.3.4 I/O.Error position 2nd I/O bus

This system variable contains the position of a faulty I/O module of the second I/O bus. The value corresponds to the bus number, the subrack and the position of the module. If there is more than one faulty module the module with the lowest position is always displayed.
1405 means: bus 1, subrack 4, position 05.

You should use the UINT data type so the value is displayed in this format.

29.3.5 I/O.Error code 1st I/O bus

Displays the faulty channels of the module shown in system variable I/O.Error position 1st I/O bus. A display is only possible when the module has line diagnosis.
Value output:

Bit No. 1...8	Dec.	Hex	Error
00 000 000	0	0	No error
00 000 001	1	1	Error in circuit channel 1
00 000 010	2	2	Error in circuit channel 2
00 000 100	4	4	Error in circuit channel 3
00 001 000	8	8	Error in circuit channel 4

Tabelle 62: Faulty channel display

00 010 000	16	10	Error in circuit channel 5
00 100 000	32	20	Error in circuit channel 6
01 000 000	64	40	Error in circuit channel 7
10 000 000	128	80	Error in circuit channel 8
11 111 111	255	FF	Faulty module

Tabelle 62: Faulty channel display

If several external circuits are faulty then a corresponding value is displayed.

29.3.6 I/O.Error position 1st I/O bus

This system variable contains the position of a faulty I/O module of the first I/O bus. The value corresponds to the bus number, the subrack and the position of the module. If there is more than one faulty module the module with the lowest position is always displayed.
1405 means: bus 1, subrack 4, position 05.

You should use the UINT data type so the value is displayed in this format.

29.3.7 SYSTEM.RAM/EPROM

This system variable displays the type, the memory used for the resource and the memory mapping.

Value		Redundant System			
Hex	Dec	left central module		right central module	
		NLS	SLP	NLS	SLP
0	0	RAM	RAM	RAM	RAM
1	1	EPROM	RAM	RAM	RAM
2	2	EPROM	EPROM	RAM	RAM
100	256	RAM	RAM	EPROM	RAM
101	257	EPROM	RAM	EPROM	RAM
102	258	EPROM	EPROM	EPROM	RAM
200	512	RAM	RAM	EPROM	EPROM
201	513	EPROM	RAM	EPROM	EPROM
202	514	EPROM	EPROM	EPROM	EPROM

Tabelle 63: RAM/EPROM Mapping

29.3.8 SYSTEM.Runversion

This system variable displays the current RUN version of the resource.
You must use the WORD data type for this variable so that the display is identical with the representation on the diagnostic display of the central

module.

29.3.9 SYSTEM.Codeversion

This system variable displays the current Code version of the resource. You must use the WORD data type for this variable so that the display is identical with the representation on the diagnostic display of the central module.

29.3.10 SYSTEM.Number of prohibited accesses

This system variable shows how often an attempt has been made to call a prohibited action or function.

29.3.11 SYSTEM.Errormask2

Error mask 2 displays general faults in the power supply, faults in coprocessor modules, active fault blanking, and the allocation of faults to the central modules. The set error bits have the following meaning:

Error bit 1...16	Hex	Dec	Fault type
0000 0000 0000 0000	0	0	No fault
0000 0000 0000 0001	1	1	Backup battery F 71xx ZB1
0000 0000 0000 0010	2	2	Backup battery F 71xx ZB2
0000 0000 0000 0100	4	4	Power supply 1
0000 0000 0000 1000	8	8	Power supply 2
0000 0000 0001 0000	10	16	Power supply 3
0000 0000 0010 0000	20	32	Fault blanking active
0000 0000 0100 0000	40	64	Fault ZB1
0000 0000 1000 0000	80	128	Fault ZB2
0000 0001 0000 0000	100	256	Copr. module 1 ZB 1
0000 0010 0000 0000	200	512	Copr. module 2 ZB 1
0000 0100 0000 0000	400	1024	Copr. module 3 ZB 1
0000 1000 0000 0000	800	2048	Copr. module 1 ZB 2
0001 0000 0000 0000	1000	4098	Copr. module 2 ZB 2
0010 0000 0000 0000	2000	8192	Copr. module 3 ZB 2
0100 0000 0000 0000	4000	16384	Backup battery on ZB 1
1000 0000 0000 0000	8000	32768	Backup battery on ZB 2

Tabelle 64: Error mask 2

If several faults occur simultaneously, a value is output that shows the fault bits at their corresponding positions. This means that several bits can be

set at the same time.
Faults for the coprocessor modules are only output when they are defined in the cubicle.

29.3.12 SYSTEM.Errormask1

Error mask 1 displays faults detected in the central modules and in the I/O bus. The set error bits have the following meaning:

Error bit 1...16	Hex	Dec	Fault type
0000 0000 0000 0000	0	0	No fault
0000 0000 0000 0001	1	1	CPU
0000 0000 0000 0010	2	2	CTC (time-IC)
0000 0000 0000 0100	4	4	Hardware watchdog
0000 0000 0000 1000	8	8	Memory fault
0000 0000 0001 0000	10	16	Program crash
0000 0000 0010 0000	20	32	Time-out
0000 0000 0100 0000	40	64	Dev. CTXC/hardware clock
0000 0000 1000 0000	80	128	Hardware clock
0000 0001 0000 0000	100	256	Connection to I/O level
0000 0010 0000 0000	200	512	Power supply monitor
0000 0100 0000 0000	400	1024	Address test I/O BT
0000 1000 0000 0000	800	2048	Time delay other ZB
0001 0000 0000 0000	1000	4098	Outputs not 0 on start-up
0010 0000 0000 0000	2000	8192	Dev. CTC/hardware clock can be tolerated
0100 0000 0000 0000	4000	16384	Not used
1000 0000 0000 0000	8000	32768	Memory unequal

Tabelle 65: Error mask 1

If several faults occur simultaneously, a value is output that shows the fault bits at their corresponding positions. This means that several bits can be set at the same time.
Faults for the coprocessor modules are only output when they are defined in the cubicle.

29.3.13 SYSTEM.Errorcode

The display of the error code is used for a detailed analysis of a fault that has occurred. The meaning of the error code can be taken from the description of the appropriate operating system.

Index

Numerics

3964R base addresses 29
3964R communication 45
3964R variables 35

A

Address conflicts 26

B

Building an application 72
BUS docu 63
Bus documentation 64
BUSCOM 43
BUSCOM variables 34
BUSCOM, MODBUS base addresses 28

C

C-code comparator 71
CFG docu 63
CFG documentation 66
Code generator 25, 49
Code generator settings 26
Compiling program code 49
Configuration 1
Control panel 51
Creating a configuration 1
CRF Docu 67
CRF docu 66
CRF documentation 67
CU, CM configure interfaces 14

D

Data types 180
Documentation configuration 63
Documentation resource 66

E

Edit cabinet 13
Editing modules 14
ELOP II bus 1
ELOP II bus assignment 2
ELOP n-file 42
Event creating 31

F
fieldbuses 43
Force image, graphic 60
Forcing I/Os 58
Forcing in force image 59
Forcing variables 58
Function check of the PES 72, 74

H
H8-STA-3 77
H8-UHR-3 81
HA-LIN-3 87
HA-PID-3 95
HA-PMU-3 103
HA-RTE-3 109
HB-BLD-3 119
HB-BLD-4 125
HB-RTE-3 131
HIBUS 2
HIBUS parameter 4
HIMA Standard blocks 75
HIPRO 37
HIPRO edit 3
HIPRO Hardware 37
HIPRO redundancy 38
HIPRO stations 4
HIPRO-N 37
HIPRO-N variables 32, 39
HIPRO-N/-S variables 31
HIPRO-S 27, 37
HIPRO-S add resource 39
HIPRO-S PES master 38
HIPRO-S variables 33
HK-AGM-3 137
HK-LGP-3 141
HK-MMT-3 149
HZ-DOS-3 167
HZ-FAN-3 171

I
I/O parameters 19
IEC functions 181
Instancing program type 11

L
Lcl page header 24
Lcl page header editing 24

M
MODBUS 43

Mode configuring serial interface 15
Modification of an application 73
Modules current 178
Modules old 179

O
OLT field 57
OLT panel 54
Online Test 53
Online test 57
OPC 43
OS download 53

P
PADT bus 23
PES master 7
PES master compiling 7, 49
PES master CRF 9
PES master docu 63
PES master documentation 65
PES master download 7, 50
PES master functions 32
PES master update 8
Program download/reload 52
proven GNU-C-compiler 71

R
Redundancy parameters 20
RES docu (current) 66
RES docu (generated) 66
RES docu (loaded) 66
Res documentation 68
Res documentation generated 69
Res documentation loaded 70
Resource 9
Resource hardware change 10
Resource properties 12, 19
Resource type assignment 9
Resource types current 175
Resource types predecessor 176

S
Safety parameters 21
Saftey related functions 71
Special interface settings 16
Subracks 176
Supported functions and function blocks 181
System variables 184
System variables assignment 30

T

Tag name editing 16, 17

target code comparator 71