

PC-based System **ELOP II**

First Steps



HIMA Paul Hildebrandt GmbH
Industrial Automation

HI 800 001 DEA

All HIMA products mentioned in this manual are protected by the HIMA trademark. This also applies to other manufacturers and their products, unless otherwise stated.

It is forbidden to pass on or copy this document or to exploit and reveal its contents without our expressed permission. Contraventions shall be liable to damage claims.

All technical information and instructions in this manual were created with great care and with the incorporation of effective control measures. However, the possibility of errors can never be completely excluded. HIMA must therefore point out that neither a guarantee nor legal responsibility nor any liability can be accepted for consequences arising from incorrect details. HIMA will always be grateful to readers who point out any errors.

We reserve the right to make technical modifications.

Further information can be found in the documentation on the CD-ROM and on our web site under www.hima.de.

Information enquiries must be directed to:

HIMA Paul Hildebrandt GmbH
Post-box 12 61
68777 Brühl

Tel: +49 (6202) 709 0
Fax: +49 (6202) 709 107

Email: info@hima.com

Contents

- 1 BRIEF OVERVIEW 1**
 - 1.1 SCOPE OF DELIVERY 1
 - 1.2 INFORMATION ABOUT THIS MANUAL 1
 - 1.3 SUPPORT 3
- 2 INSTALLATION 5**
 - 2.1 HARD- AND SOFTWARE REQUIREMENTS 6
 - 2.2 WHERE TO INSTALL..... 6
 - 2.3 HOW TO START THE INSTALLATION 7
 - 2.4 NETWORK INSTALLATION..... 9
 - 2.5 DEINSTALLATION 9
- 3 INTRODUCTION TO ELOP II 11**
 - 3.1 STARTING ELOP II 11
 - 3.2 STANDARD SCREEN ELEMENTS..... 12
 - 3.3 TITLE BAR..... 13
 - 3.4 MENU BAR 13
 - 3.5 TOOLBAR 14
 - 3.5.1 *ELOP II Project Management Toolbar*..... 14
 - 3.5.2 *Function Block Diagram Editor Toolbar* 15
 - 3.6 STATUS BAR 16
 - 3.7 STRUCTURE WINDOW 17
 - 3.7.1 *Context Menu for Objects* 18
 - 3.8 WORKING AREA 19
 - 3.8.1 *Function Block Diagram Editor (FBD Editor)* 19
 - 3.9 ERROR STATE VIEWER 20
 - 3.10 ONLINE HELP 20
- 4 OBJECTS IN THE STRUCTURE WINDOW 21**
 - 4.1 PROJECT 21
 - 4.2 LIBRARY..... 22
 - 4.2.1 *Program Type* 22
 - 4.2.2 *Function Block Type* 22
 - 4.2.3 *Function* 22
 - 4.3 CONFIGURATION..... 23
 - 4.3.1 *Resource*..... 23
 - 4.4 DOCUMENTATION 24
 - 4.5 STRUCTURE FOLDER 24

5	WINDOW ADJUSTMENT AND NAVIGATION.....	25
5.1	WINDOW ADJUSTMENT.....	25
5.1.1	<i>Maximizing the Working Area</i>	<i>25</i>
5.1.2	<i>Maximizing and Restoring Panes.....</i>	<i>27</i>
5.1.3	<i>Zoom</i>	<i>27</i>
5.2	NAVIGATION	28
5.2.1	<i>Function Diagrams with centered Starting Point.....</i>	<i>28</i>
5.2.2	<i>Moving in Logic using the Overview Window.....</i>	<i>29</i>
5.2.3	<i>Moving in Logic using the Page List</i>	<i>30</i>
6	EXERCISES	31
6.1	CREATING A PROJECT.....	31
6.1.1	<i>Creating a Library (optional).....</i>	<i>33</i>
6.1.2	<i>Creating Configuration and Resource.....</i>	<i>35</i>
6.1.3	<i>Creating a Type Instance (Program for the Resource).....</i>	<i>37</i>
6.1.4	<i>Creating a Function Block (option).....</i>	<i>38</i>
6.2	CREATING LOGIC	40
6.2.1	<i>Defining Variables</i>	<i>40</i>
6.2.2	<i>Variable Declaration Editor</i>	<i>42</i>
6.2.3	<i>Creating Logic</i>	<i>45</i>
6.3	OFFLINE SIMULATION OF LOGIC (OPTION).....	51
6.4	PLC CABINET LAYOUT AND VARIABLE ASSIGNMENT	56
6.5	RUNNING A PROGRAM ON A PES.....	62
6.5.1	<i>Compile Program</i>	<i>62</i>
6.5.2	<i>Setup Communication between PC and PES.....</i>	<i>64</i>
6.5.3	<i>Download and Start.....</i>	<i>74</i>
6.6	ONLINE TEST.....	80
6.7	FORCING INPUTS AND OUTPUTS	85
6.8	DOCUMENTATION.....	91
6.9	BACKUP OF A PROJECT	98
6.9.1	<i>Archiving.....</i>	<i>98</i>
6.9.2	<i>Restore a Project</i>	<i>100</i>
7	APPENDIX	103
7.1	INDEX.....	103
7.2	LIST OF ABBREVIATIONS	108

1 Brief overview

1.1 Scope of Delivery

The ELOP II scope of delivery includes:

- This manual
The manual “First Steps” enables you to obtain a speedy and simple introduction to the operation of ELOP II. For this purpose, in addition to an overview of the functions, it provides step-by-step instructions to create a project.
- A CD-ROM
Beside the ELOP II software, the CD-ROM also contains a few help programs and the documentation for the current HIMA automation systems.
- A hardlock (dongle)
The hardlock is for license administration (protection against unauthorized use) of the protected ELOP II software.

1.2 Information about this Manual

Screenshots and procedures correspond to ELOP II version 4.1. In principle the manual is also suitable for previous versions (3.0 or 3.5).

In this manual, the user finds details worth knowing about the most important functions of ELOP II to familiarize himself in the context of a training course or auto-didactically.

Basically this manual is an introduction to programming and operating (online functions) for the safety-related programmable electronic systems H41q/H51q.

For the hardware in use also regard the data sheets, the operating system manual and the safety manual.

The installation of ELOP II is explained in Chapter 2.

The chapters 3 to 5 describe in general terms how to use and operate ELOP II. Users without knowledge in ELOP II should read these sections carefully.

Chapter 6 contains exercise lessons, on the basis of which users, who already have some basic knowledge of ELOP II, can familiarize themselves on how to create projects and increase their skills.

In the appendix in chapter 7 you find explanations about the technical terms used, the index tab and the list of abbreviations.

1.3 Support

You have the choice of various possibilities if you have any questions on operation or reporting program errors, or if you wish to make suggestions about improvements.

Frequent questions	Chapter in this manual	Questions and answers on basic subjects
News, FAQ's, Download	Our Web page www.hima.com	News, frequently asked questions, functional objects
Questions and ideas	Via email: Support@hima.com Phone: +49-(0)6202-709 259 +49-(0)6202-709 261 Fax: +49-(0)6202-709 199	Between 9:00 and 17:00 Central European Time (on working days)

Note: *This manual is part of the documentation for the ELOP II seminars at HIMA. Because of the powerful nature of ELOP II, we can only cover the most important functions of the program.*

It is recommended that you participate in a seminar in order to deepen your knowledge.

2 Installation

This chapter covers:

- What is required?
- Where is ELOP II installed?
- How is installation started?
- Installation within a network
- Deinstallation

ELOP II is a hardware-protected program. The hardlock module (dongle) must be connected to the parallel or USB port.



Fig. 1: Hardlock for parallel and USB port

To address the hardlock module, a driver must be installed on the computer. Driver installation requires administrator rights! If in doubt, consult the system administrator.

2.1 Hard- and Software Requirements

For the installation of ELOP II on a personal computer, you need the hardlock and the CD-ROM.

The computer hardware requirements are as follows:

	Minimum	Recommended
Processor Intel Pentium II®	500 MHz	1,2 GHz
RAM	256 MB	512 MB
Graphic card	2 MB XGA (1024x768)	8 MB True Color (1280x1024)
Operating system	Windows 2000 or Windows XP (latest service pack recommended)	

Table 1: PC hardware requirements

If a printer is connected to the hardlock, the printer should be switched on. Some printers exhibit insufficient load resistance when switched off.

2.2 Where to install

You can perform the installation on a local hard disk or on a network drive (see chapter 2.4).

If ELOP II should be accessible by different users (with different user names) on one computer, ELOP II must be installed for every user individually.

The users must have at least main user rights. The procedure for multiple installation can be found in the ELOP II Online help, or in the installation program.

2.3 How to Start the Installation

The installation of ELOP II and all additional components can be started from the installation menu of the CD-ROM.

The CD-ROM also contains the complete documentation in PDF format for the ELOP II software and for the H41q/H51q system family. The Adobe Acrobat Reader®, required for displaying these files, is included.

How to start the installation:

1. Insert the CD-ROM in the CD drive. The installation menu is started automatically.
If the installation menu does not start automatically, open the root directory of the CD-ROM in the Windows Explorer and double click the file **START.EXE**.
2. From the Table of Contents open the folder “Software Installation\ELOP II”.
3. Select **Hardlock Installation** and run the installation of the hardlock driver.

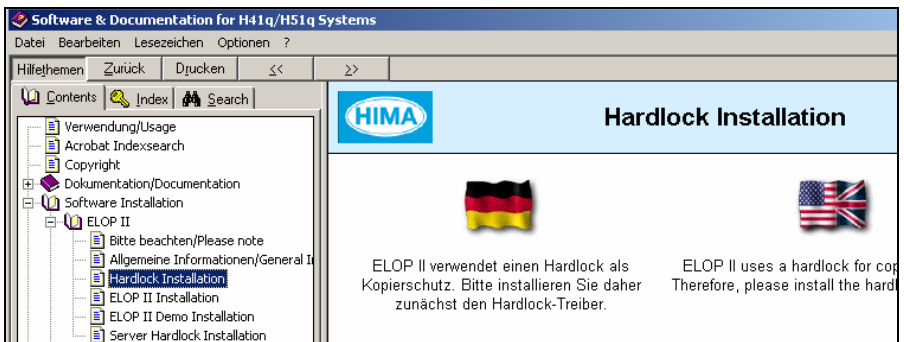


Fig. 2: Hardlock installation

4. After installing the hardlock driver, click on the button **ELOP II** for installing ELOP II. Please read the notes for installation.
Beginning with ELOP II version 4.1, both the English and German languages are installed simultaneously. Select **ELOP II Control Center, Administration** to choose your language.

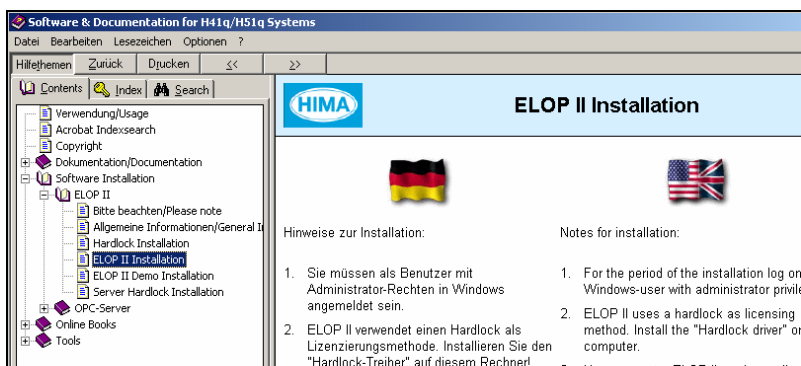


Fig. 3: ELOP II installation

5. The Setup automatically creates desktop icons on request.

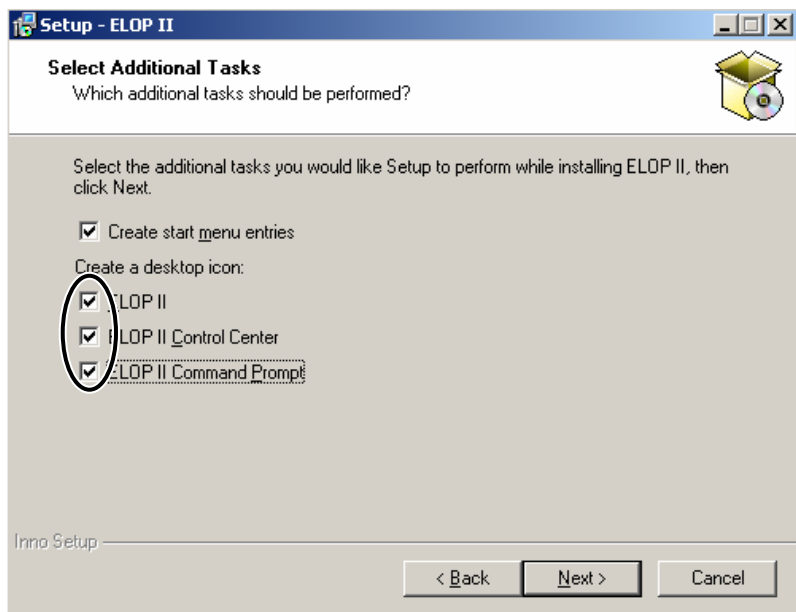


Fig. 4: Creating desktop icons

2.4 Network Installation

1. Installation is performed as described in Chapter 2.3, whereby a directory on a network server is selected for the destination.
2. Grant access to the network directory for the other users.
3. Perform the workstation installation for each workstation and user by running the program WS_SETUP from the installation directory of the server.

2.5 Deinstallation



Fig. 5: Uninstall

To uninstall ELOP II, start the **Control Center**, **Administration** and select **Uninstall**.

3 Introduction to ELOP II

This chapter covers:

- Starting ELOP II
- Elements of the ELOP II user shell
 - Menu and title bar
 - Symbol and status bar
 - Screen panes, Structure window and Working area
 - Error state viewer

ELOP II is a program with numerous functions whose access is simplified by the intuitive user guidance.

3.1 Starting ELOP II

In the Windows **Start** menu select **Programs** and then **ELOP II**.

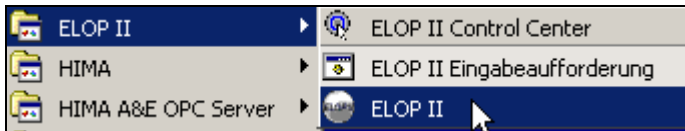


Fig. 6: Start menu ELOP II

Alternatively, you can start the program in the “ELOP II Control Center” or via the desktop icon.

3.2 Standard Screen Elements

After ELOP II has started the standard screen appears, as shown in Fig. 7. The standard screen basically comprises the following elements:

- 1 Title bar
- 2 Structure window
- 3 Menu bar
- 4 Toolbar for the project management
- 5 Working area
- 6 Toolbar for the Function Block Diagram editor (FBD editor)
- 7 Error state viewer
- 8 Status bar with coordinate information of the Function Block Diagram editor

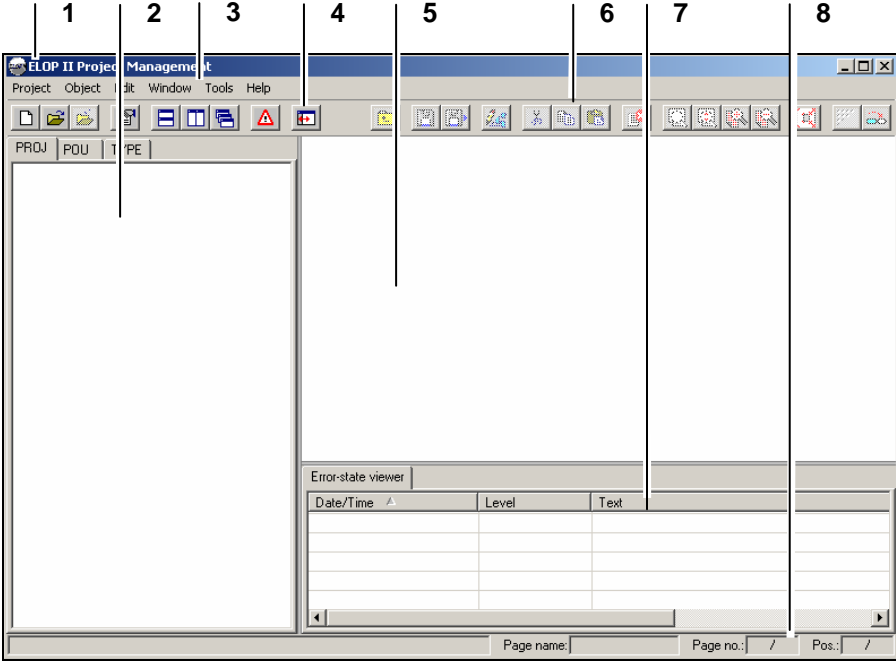


Fig. 7: ELOP II standard screen

3.3 Title Bar

In addition to the standard functions for maximizing, minimizing and closing the window, the title bar contains information about the project and about the object currently selected.

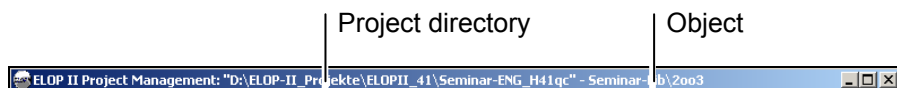


Fig. 8: Title bar

3.4 Menu Bar

Most functions of ELOP II are available in the menu bar.

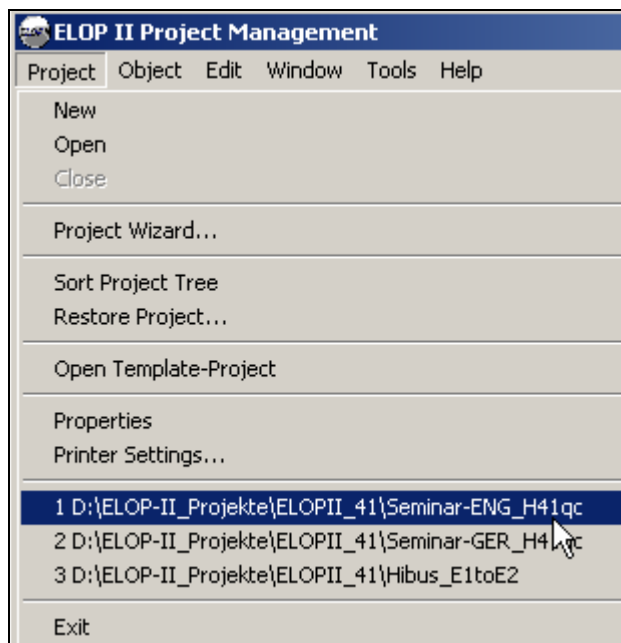


Fig. 9: The “Project” menu

3.5 Toolbar

The toolbar is located beneath the menu bar and is split into the parts “Project management” and “Function Block Diagram editor”.

3.5.1 ELOP II Project Management Toolbar

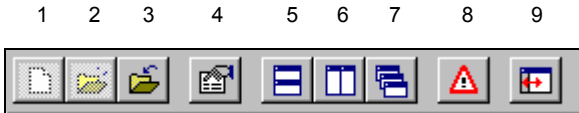


Fig. 10: Project management toolbar

The buttons (from left to right) represent the following functions:

1. *New*: Creates an empty project
2. *Open*: Opens a project already created
3. *Close*: Closes the current project
4. *Properties*: Defines global settings
5. *Tile horizontally*: Horizontally tiles the windows in the Working area
6. *Tile vertically*: Vertically tiles the windows in the Working area
7. *Cascade*: Cascades the windows in the Working area
8. *Error state viewer*: Minimizes the Error state viewer.
If the Error state viewer is already minimized you can restore it by clicking the button again.
9. *Minimize/Restore Structure Window*: Maximizes the Working area (= area in which the data object can be edited, e.g. by using the FBD editor).
If the Working area is already maximized you can restore the original setting by clicking the button again.
Note: This function is only available in the toolbar.

3.5.2 Function Block Diagram Editor Toolbar

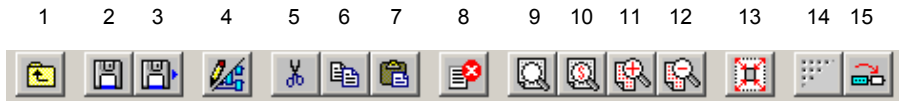


Fig. 11: Function Block Diagram editor toolbar

The buttons (from left to right) represent the following functions:

1. *Open*: Opens the calling POU
2. *Save*: Save the current object (e.g. POU)
3. *Save As...*: Save the current object (e.g. POU) under a different name
4. *Toggle OLT-/Edit-Mode*: Toggles the active editor between Online-Test mode and Edit mode (only for Offline simulation)
5. *Cut*: Moves the selected object from the POU editor into the clipboard
6. *Copy*: Copies the selection object from the POU editor into the clipboard
7. *Paste*: Inserts the object from the clipboard in the logic
8. *Start Early Error Detection*: Starts the early error detection for the POU editor
9. *Zoom to Page*: Displays the full page of the function block diagram
10. *Zoom to Selection*: Magnifies the current selection as the active area
11. *Zoom In*: Enlarges the current display by 30 percent
12. *Zoom Out*: Reduces the current display by 30 percent
13. *Maximize/Restore Current Focus*: Maximizes the current component of the FBD editor (= area)
14. *Toggle Grid*: Shows or hides the grid in the FBD editor drawing field
15. *Display/Hide Values*: Show current values in the Offline simulation or Online test.

Note: If the mouse pointer is positioned over a button for a short time, a "Quick Info " (short help text) appears.

3.6 Status Bar

The status bar at the lower edge of the window contains information messages and help texts from the project management and from the Function Block Diagram editor, as well as the current cursor position.

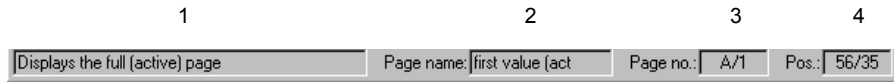


Fig. 12: Status bar

The fields in the status bar have the following meaning (from left to the right):

1. Info line
2. Page name
3. Page number
4. Cursor position

3.7 Structure Window

The Structure window displays the project in its hierarchical structure. You can choose between three views with different degrees of detail.

Entire project

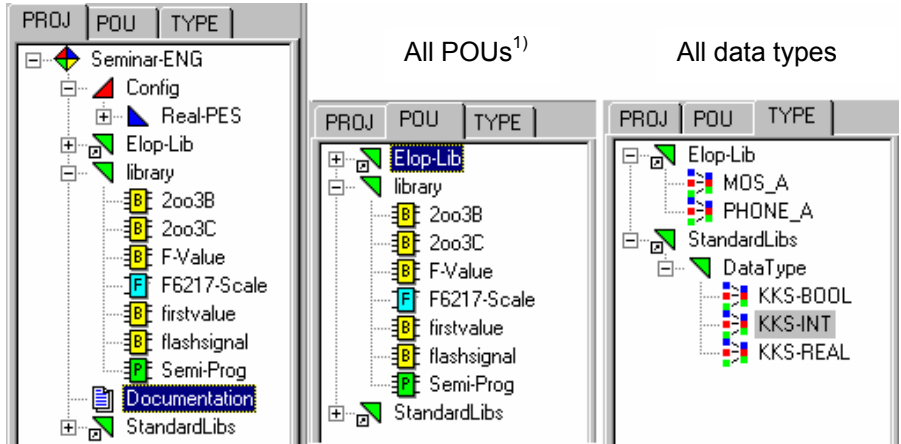


Fig. 13: Structure window

¹⁾ Program Organization Unit (object)

3.7.1 Context Menu for Objects

Open the context menu of an object by right-clicking on the object in the Structure window. The individual menu functions are selected as usual by clicking on the function with the left mouse button.

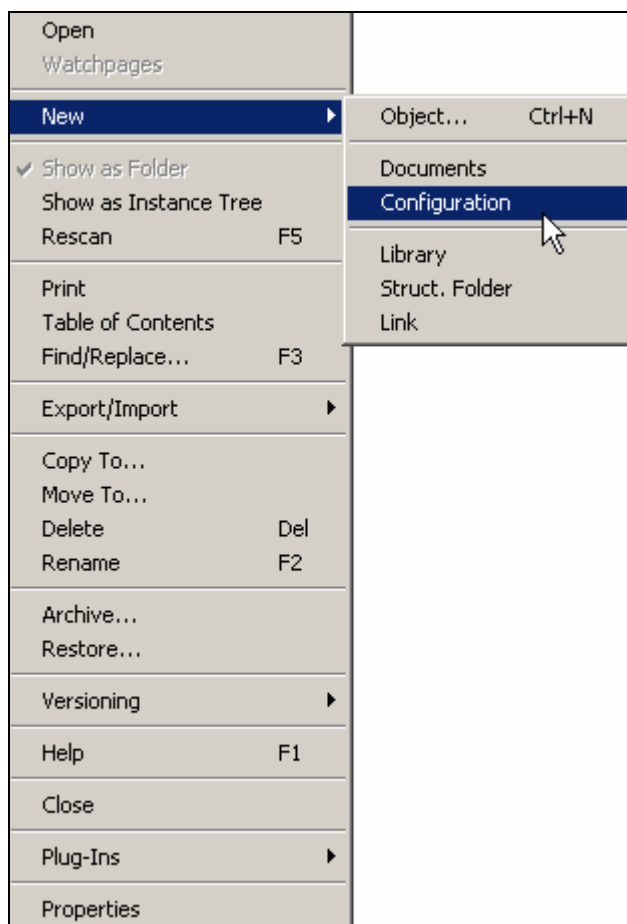


Fig. 14: Context menu of the project

3.8 Working Area

In the Working area (see Fig. 7), you process the data objects with the

- Function Block Diagram editor and the
- Document editor.

3.8.1 Function Block Diagram Editor (FBD Editor)

With the Function Block Diagram editor you create the function diagrams in the Function Block Diagram language (FBD) or in the Sequential Function Chart language (SFC, sequences).

The Function Block Diagram editor (Fig. 15) comprises the panes

- 1 Drawing area
- 2 Variable declaration editor
- 3 Overview window
- 4 Interface declaration editor

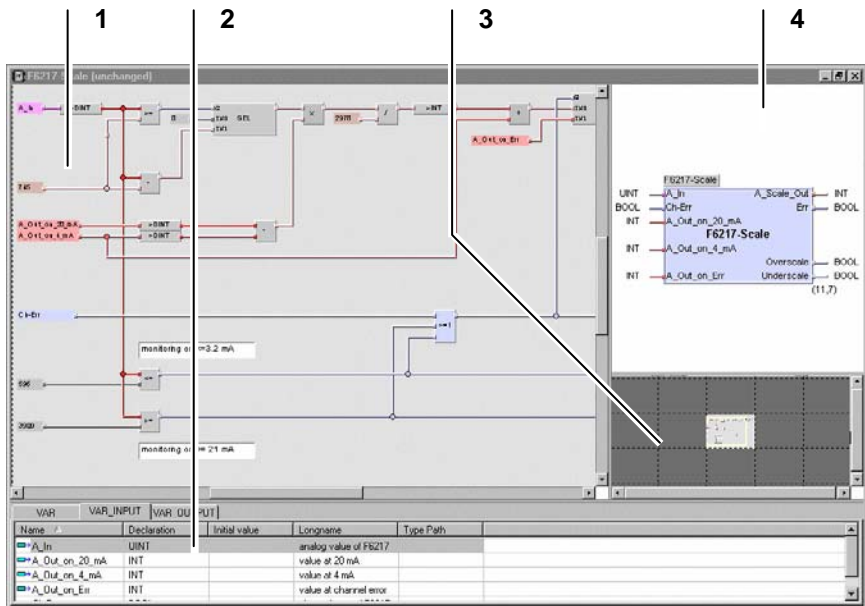

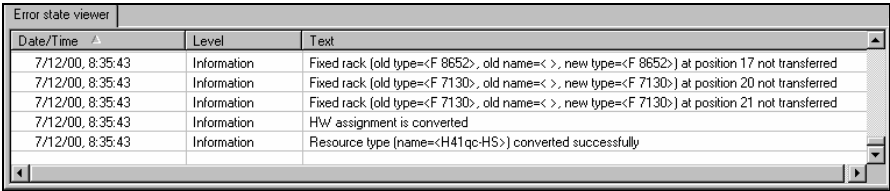


Fig. 15: Function Block Diagram editor

3.9 Error State Viewer

Error messages are displayed in the Error state viewer. The occurrence of a new error message is indicated by a flashing symbol  in the Windows task bar.



Date/Time	Level	Text
7/12/00, 8:35:43	Information	Fixed rack (old type=<F 8652>, old name=< >, new type=<F 8652>) at position 17 not transferred
7/12/00, 8:35:43	Information	Fixed rack (old type=<F 7130>, old name=< >, new type=<F 7130>) at position 20 not transferred
7/12/00, 8:35:43	Information	Fixed rack (old type=<F 7130>, old name=< >, new type=<F 7130>) at position 21 not transferred
7/12/00, 8:35:43	Information	HW assignment is converted
7/12/00, 8:35:43	Information	Resource type (name=<H41qc-HS>) converted successfully

Fig. 16: Error state viewer

3.10 Online Help

In the Online help you find detailed descriptions of all ELOP II functions. With the aid of the index you can quickly obtain help on key words.

Likewise you can get online help for all objects from the context menu.

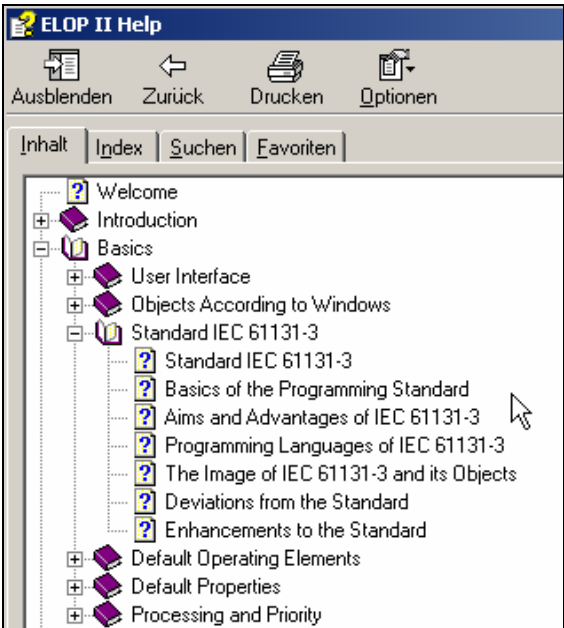


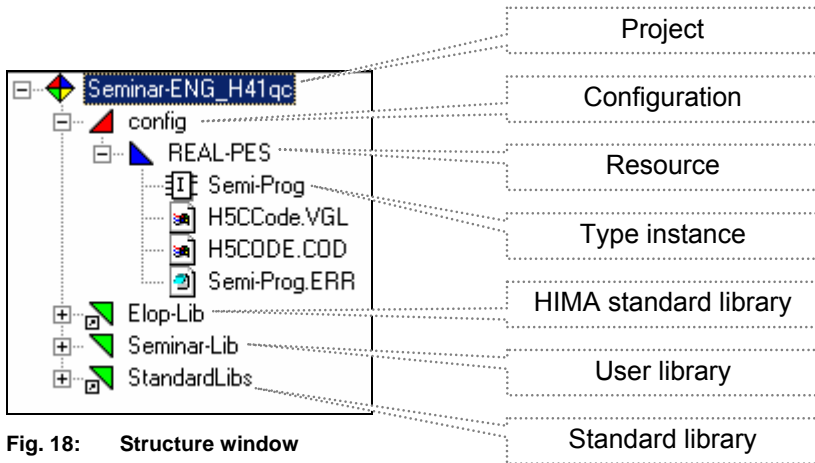
Fig. 17: ELOP II Online help

4 Objects in the Structure Window


This chapter covers:

- The hierarchical structure of the objects in the Structure window
- The meaning of the objects


The Structure window displays all objects of a project in hierarchical order (see Fig. 13). Numerous functions of the project management are carried out from the Structure window.



4.1 Project

The project  is the superordinated object. All other objects are created under a project. Only one project may be opened in ELOP II at a time.

4.2 Library


A Library  can contain functions, function blocks and programs. In the terminology of IEC 61131-3 functions, function blocks and programs are also referred to as Program Organization Units (POU).

The **StandardLib** contains all logic elements described in the IEC 61131-3. Use these elements to create your own more complex Program Organization Units.


The **ELOP Lib** contains additional function blocks provided by HIMA for special functions like communication control, redundancy evaluation for certain I/O modules etc.

The user can provide as many as desired additional own libraries.

4.2.1 Program Type


A Program type  contains all functions of an application. A Program type can be assigned to several controllers for execution. A controller executes a *Program instance* of the *Program type* created in the library.

4.2.2 Function Block Type


The Function block type  contains partial functions of an application, comparable with a sub-routine. The Function block type can also be used for structuring the program according to the system structure. The Function block type can temporarily store values in local variables. The output value depends on the input values and the temporarily stored values (typical example: Flip-Flop, Timer).

In addition, the Function block type can be used to access external variables (global variables of a program).

4.2.3 Function

A Function  contains basic functions of an application. Contrary to the Function block type a function cannot store any statuses. The output value depends only on the input values (typical example: AND, OR).




4.3 Configuration

The Configuration  groups controllers into logic units, between which a communication connection can exist.


4.3.1 Resource

A Resource  is the term defined in IEC 61131-3 for a target system that executes a program, i.e. a HIMA controller.

4.3.1.1 Program Instance

A Program instance  is a reference to a Program type that already exists in a library (see 4.2.1). The program is run on this resource (PES, Programmable Electronic System). Controller-specific settings such as creating communication variables and system variables must be defined in the Program instance  and not in the Program type  !


4.3.1.2 Type Instance

A Type instance  is a Program instance including a Program type in a hidden subfolder. One Type instance cannot be used for more instances, but is a 1-to-1 relation between type and instance.

Most users prefer a Type Instance because there is no need to think about the differences between Program instance and program.

4.3.1.3 Task


Creating a task for the offline simulation is required only in ELOP II version 3.0.

A task  defines the execution of a program in a resource. ELOP II supports multitasking, but HIMA controllers only allow the execution of a single program (= a single Program instance).


Therefore the task can be used on a PC for Offline simulation to define the cycle time for the execution of the assigned Program instance.

4.3.1.4 Project Root

From ELOP II version 3.5 on a Project root is no longer required and not created any more.

The Project root  is required for the internal search algorithm in ELOP II version 3.0. This object is used to influence the search order and may only be changed by experienced users.

4.4 Documentation

The documentation  allows the arrangement of the objects being documented by way of Drag & Drop. Users can perform revision management for all included documents.

4.5 Structure Folder

A Structure folder can be used for creating a better structure in the project tree (like creating subfolders in explorer).

A Structure folder can contain every object like “configuration”, “library”...

5 Window Adjustment and Navigation

5.1 Window Adjustment

5.1.1 Maximizing the Working Area

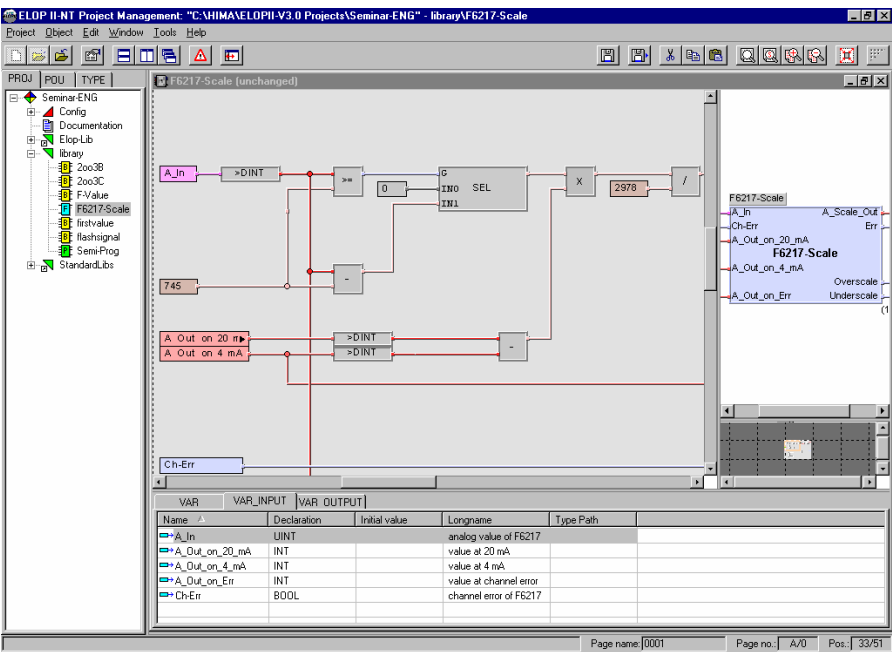




Fig. 19: Function Block Diagram editor in the Working area

First make sure that the Structure window is active.

You can hide and show the Structure window in Fig. 19 with the button  in the left group of buttons in the symbols bar.

To hide and show the Error state viewer, use the button .

This enables the Working area for the Function Block Diagram editor to be enlarged or reduced.

After clicking the button  the Working area is maximized.

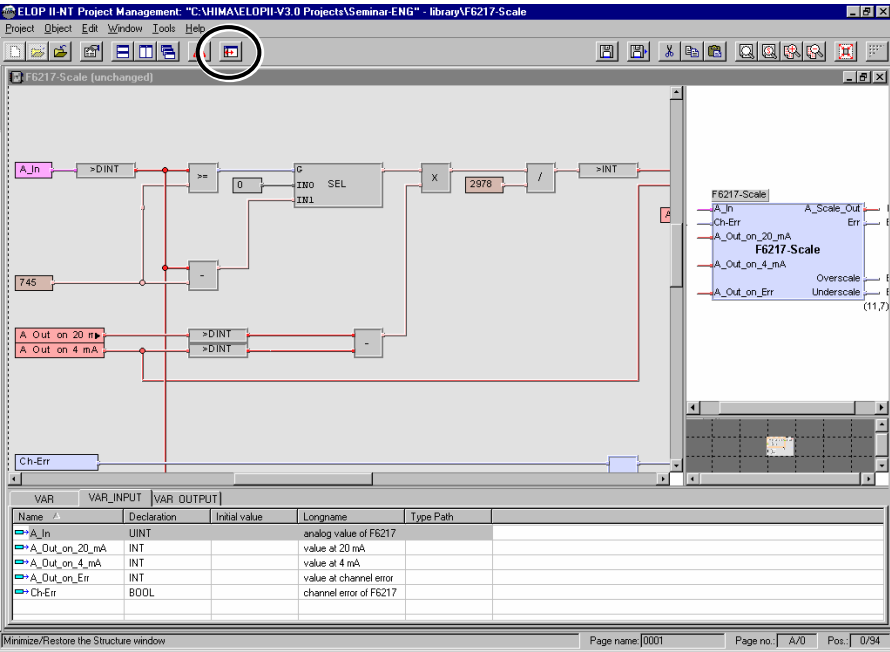





Fig. 20: FBD editor in the maximized Working area

Note: The size of the Working area can also be enlarged by dragging the boundaries towards the variable declaration editor and towards the interface declaration editor.

Note: There is a toggle function with the button . Click again on  and Structure window opens again.

5.1.2 Maximizing and Restoring Panes

To activate a window in the Working area click inside the window (Fig. 20).

The active window can be maximized using the button  in the right group of buttons of the symbols bar.

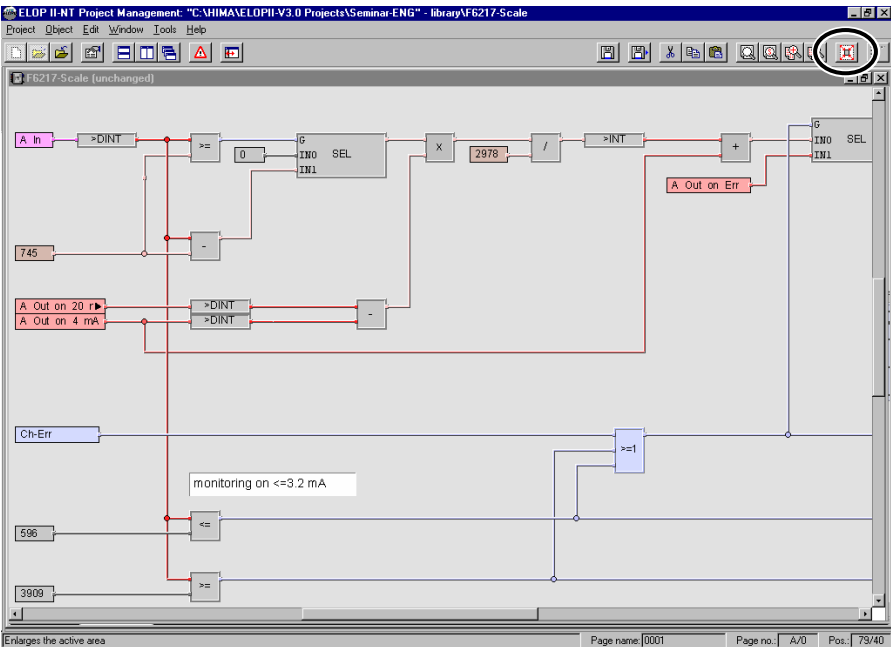



Fig. 21: Maximized drawing area

Pressing the button  again causes the panes of the Function Block Diagram editor to be shown in its original size again.

5.1.3 Zoom

By using the buttons  in the symbols bar you can enlarge and reduce the view in the drawing area and in the overview window.

5.2 Navigation

5.2.1 Function Diagrams with centered Starting Point

The concept of ELOP II eliminates the need to subsequently insert individual pages. The centered starting page can be expanded in any direction without page limitations. The position of a page is marked with coordinates. Columns are named with capital letters and lines with numbers.

The first page is the page with the coordinate A/0. As soon as an element is placed on this page, the page is activated.

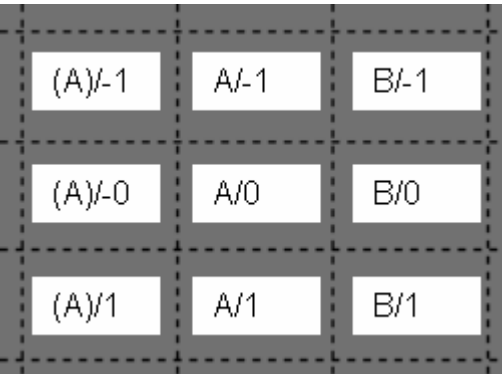


Fig. 22: Page numbering in the function diagram

Active pages are highlighted. As soon as you place an element on an adjacent page, this page is also activated and thereby highlighted. In this way, the function diagram can be extended in any desired direction.

Note: If you plan to insert a page between existing pages, you can move a page. To do this, select "Move page" from the menu "Plug-Ins" in the context menu of the page.

5.2.2 Moving in Logic using the Overview Window

The overview window shows a top view of the function diagram. You can navigate between pages by clicking on one of the pages in the overview window.

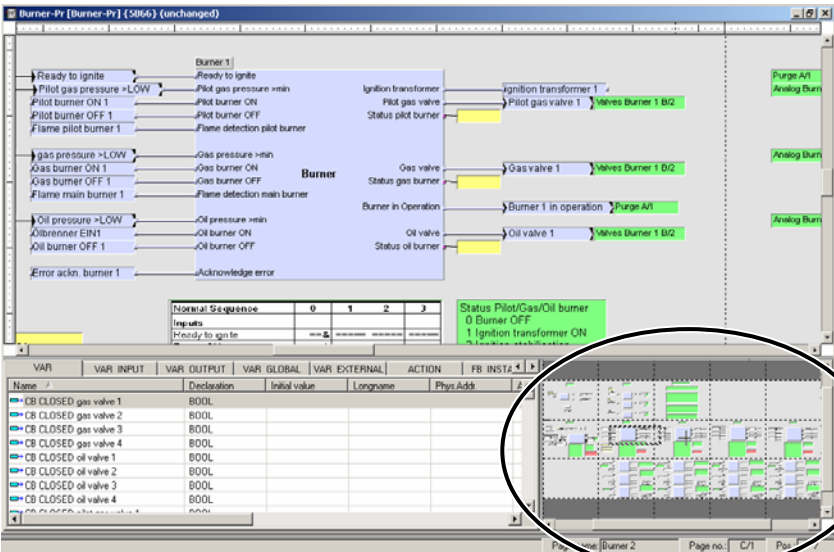


Fig. 23: Overview window

Moving the mouse pointer in the overview window shows the name of the pages in the status bar:

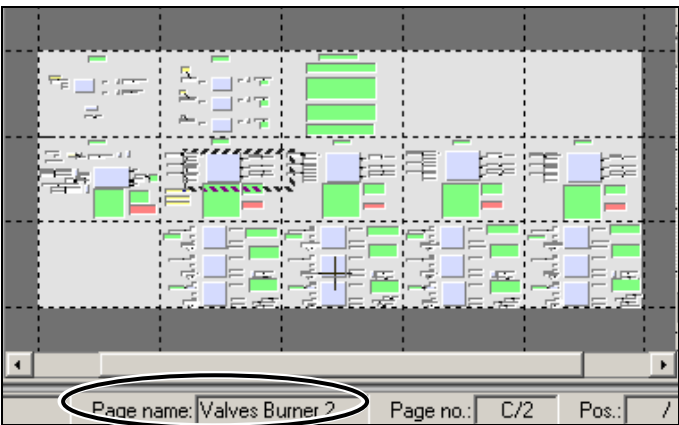


Fig. 24: Page name in overview window

5.2.3 Moving in Logic using the Page List

Give a right mouse click to the drawing area.

Go to “Plug-Ins”

Go to “Page list”

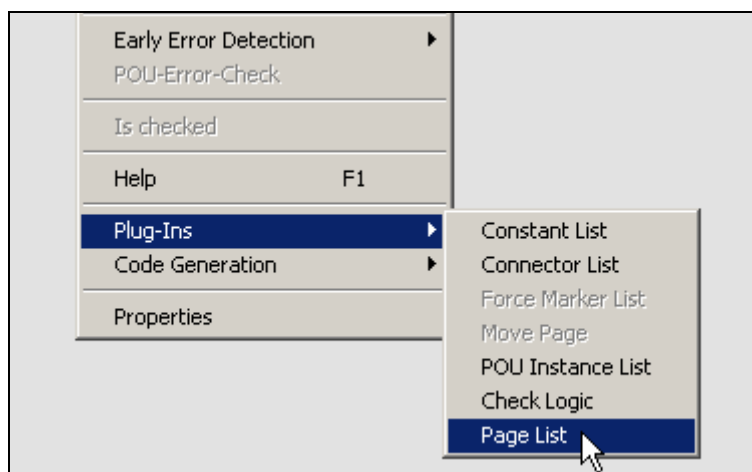


Fig. 25: Go to page list

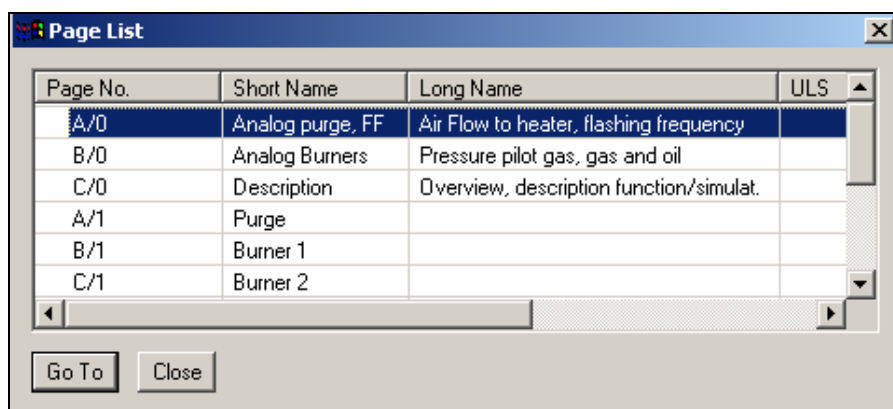


Fig. 26: Page list

6 Exercises

6.1 Creating a Project

Step 1: Start ELOP II:

- ☐ Click on **ELOP II** in the Windows start menu and select **ELOP II**.

Since ELOP II Version 4.1 now the Project Wizard appears.

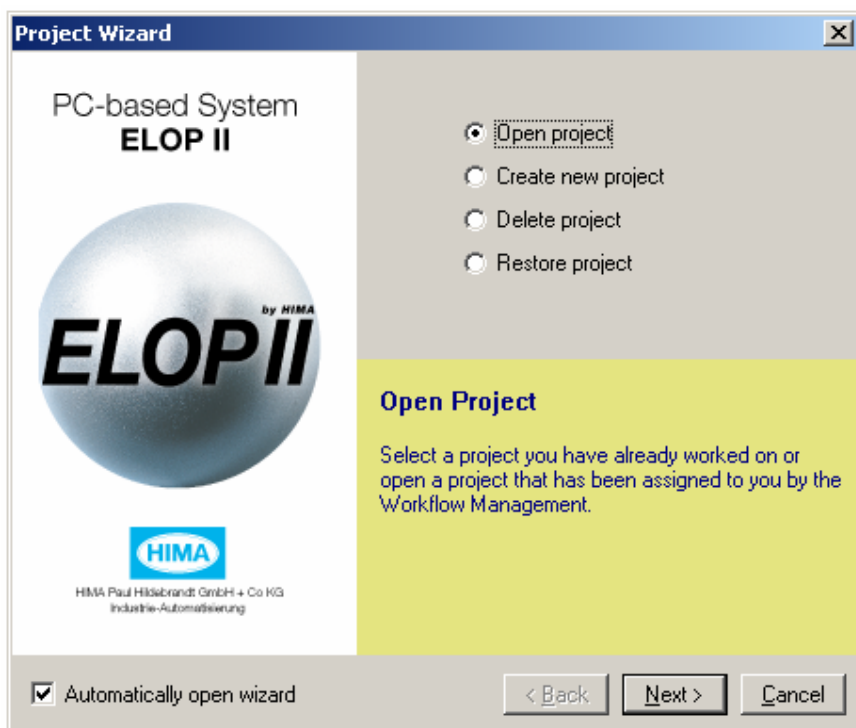



Fig. 27: Project Wizard

The Project Wizard can assist you in standard procedures like mentioned in the window above. Just follow the instructions.
In this manual we show the standard procedures without the wizard.

Click **Cancel** now to close the wizard.

Step 2: Create a new project:

- ☐ From the Project menu, select the option New or click on the symbol  in the symbols bar.

Step 3: Define the project path and name:

- ☐ In the figure below, left pane, select the directory where the new project should be created, enter the name for the new project in the Object name field and click OK.

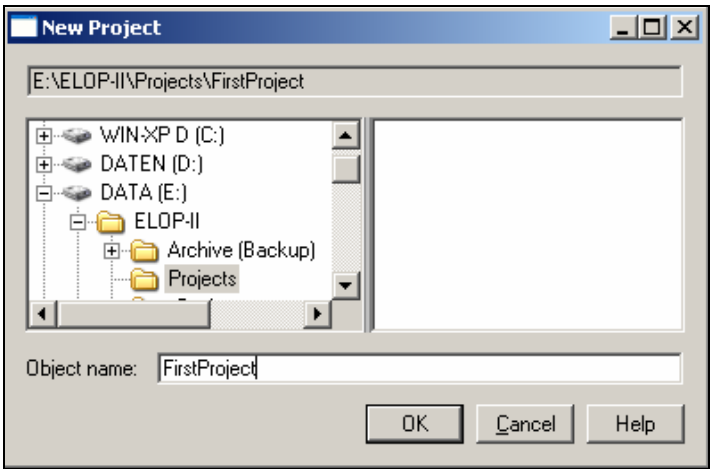


Fig. 28: Create project

The new project is displayed in the Structure window. All standard libraries are automatically included in the new project.

- ☐ Click on the “+” in front of the project name and open the folder.

6.1.1 Creating a Library (optional)

Step 1: Create a new library as the first element in the project. The library will contain the function blocks created by the user:

- ☐ In the Structure window, click on the project.
- ☐ Open the context menu with a right mouse click.
- ☐ Select **New, Library**.

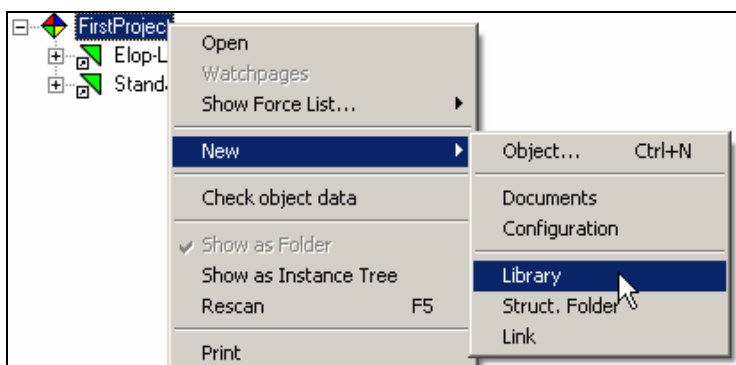


Fig. 29: Create library

A new library with the name "NewLib" is added to the project.

Step 2:

Change the name of the library “NewLib” to “Prog_Library”:

If the library has been created new, the cursor is already in the field to overwrite the name.



Fig. 30: Rename library in the field

or:

- ☐ In the Structure window, click on “NewLib”.
- ☐ Open the context menu of the library “NewLib”.
- ☐ Select **Rename** and change the name to e.g. “X-Function blocks”.

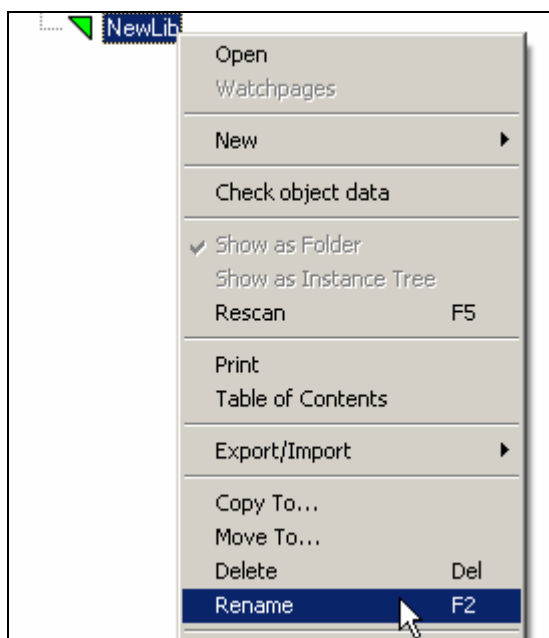


Fig. 31: Rename library with context menu

6.1.2 Creating Configuration and Resource

Step 1: Create a configuration. The configuration will incorporate the resource(s):

- ☐ Click on the project in the Structure window.
- ☐ Open the context.
- ☐ Select **New, Configuration**.

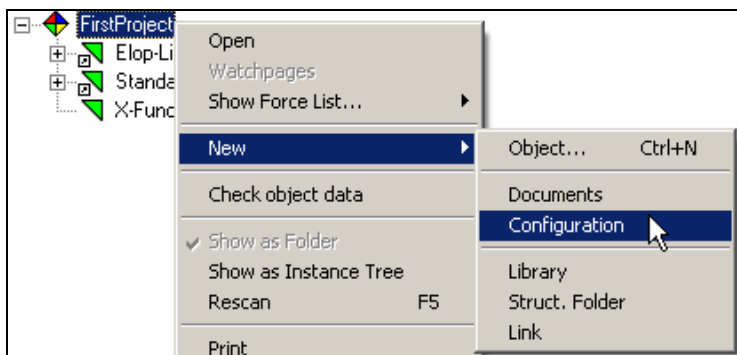


Fig. 32: Create a new configuration

Step 2: Rename the new configuration:

- ☐ In the Structure window, click twice slowly on the configuration name.

An input field opens and the name can be changed, e.g. "Configuration".

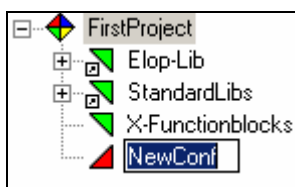


Fig. 33: Rename configuration

Step 3:

Create a new resource within the configuration:

- ☐ In the Structure window, click on the configuration.
- ☐ Open the context menu.
- ☐ Select **New, Resource**.

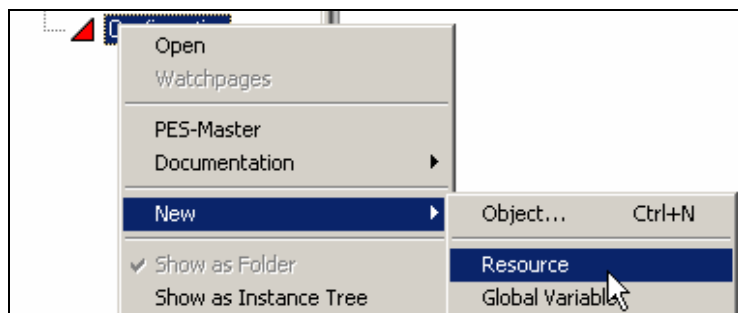


Fig. 34: Create a resource

Step 4:

Rename the new resource:

- ☐ Type in the new name of the resource.
The resource name must consist of eight characters with the last two characters being figures (resource ID).
See also chapter 6.5.2, Step 7.

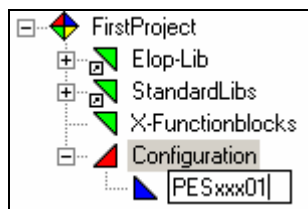


Fig. 35: Rename resource

6.1.3 Creating a Type Instance (Program for the Resource)

Step 1: Create a Type instance:

- ☐ In the Structure window, click on the resource.
- ☐ Open the context menu.
- ☐ Select **New, Type Instance**.

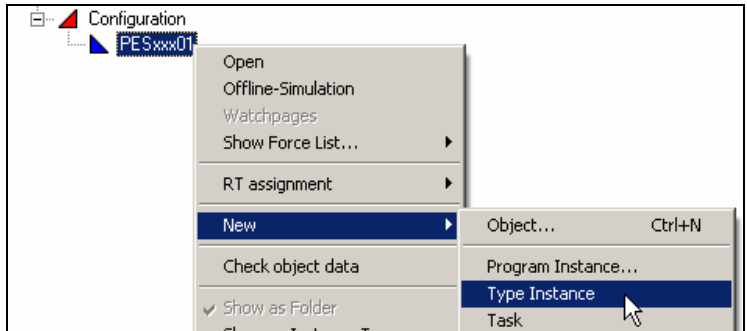


Fig. 36: Create a new Type instance

Step 2: Rename the Type instance:

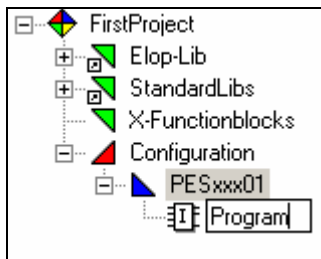


Fig. 37: Rename Type instance

6.1.4 Creating a Function Block (option)

Step 1:

Create a function block:

- ☐ In the Structure window, click on the library.
- ☐ Open the context menu.
- ☐ Select **New, Function Block Type (FBD)**.

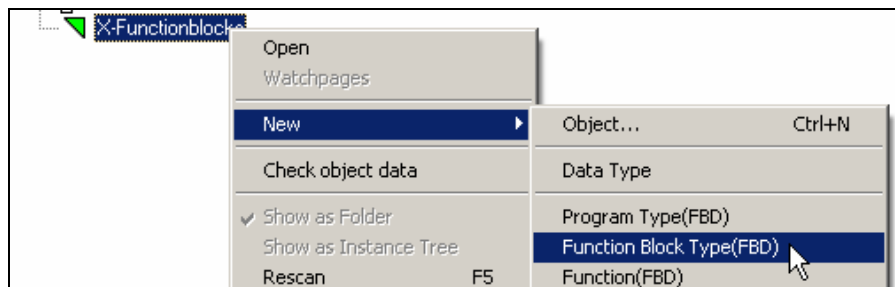


Fig. 38: Create a new function block

Step 2:

Rename the function block:

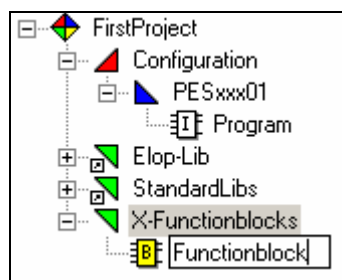


Fig. 39: Rename function block

Step 3:

Sort the Project Tree:

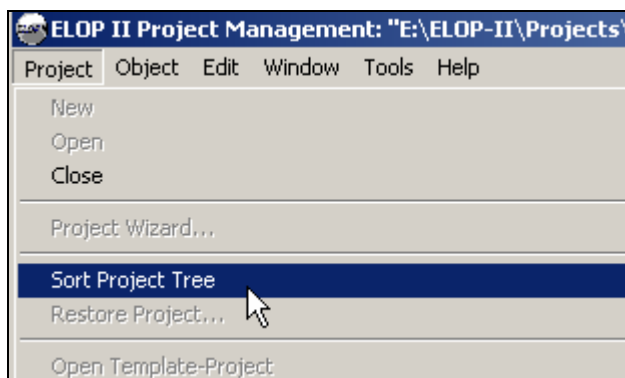


Fig. 40: Sort Project Tree

☐ Check the project structure. It should look like below:

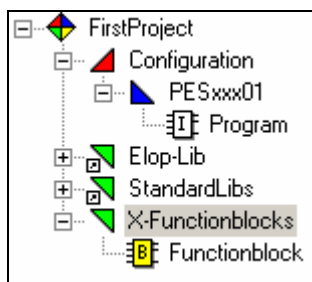



Fig. 41: Project structure

6.2 Creating Logic

Open the Program by double clicking on the symbol: . The FBD editor opens.

6.2.1 Defining Variables

Variables are created in the variable declaration editor. The following types of variables do exist:

Local Variables (VAR) are defined in a program, in a function or in a function block and are available only in this POU. Local variables in a program can be connected to hardware (e.g. DI or AI, see VARxx).

Interface Variables (VAR_INPUT, VAR_OUTPUT) can only be defined in a function or in a function block. Interface variables are the direct connection between the surrounding logic and the internal logic of the function or function block. Each time a function or function block is invoked, external values are transferred to the internal logic using VAR_INPUTs. The results of the internal logic are written back to the calling POU using VAR_OUTPUTs (Fig. 42, FBX).

Global Variables (VAR_GLOBAL) can only be defined in a program and can be used in every function block (FB), where they have to be defined as **VAR_EXTERNAL** with exactly the same variable name.

Example:

In Fig. 42 variable GE2 is created in function block FB Y and is also used in function block FB X. Therefore, this variable has to be defined in both function blocks as VAR_EXTERNAL.

Additionally variable GE2 must be defined as VAR_GLOBAL in the program and can also be used there.

*Note: Global Variables may be connected to the hardware (e.g. DI, AI, DO, AO). Input of initial-values for VAR_GLOBAL is possible only in the Program.
A printout of the cross-reference list, showing all uses of the VAR_GLOBAL / VAR_EXTERNAL, can be executed in the context menu of the resource.*

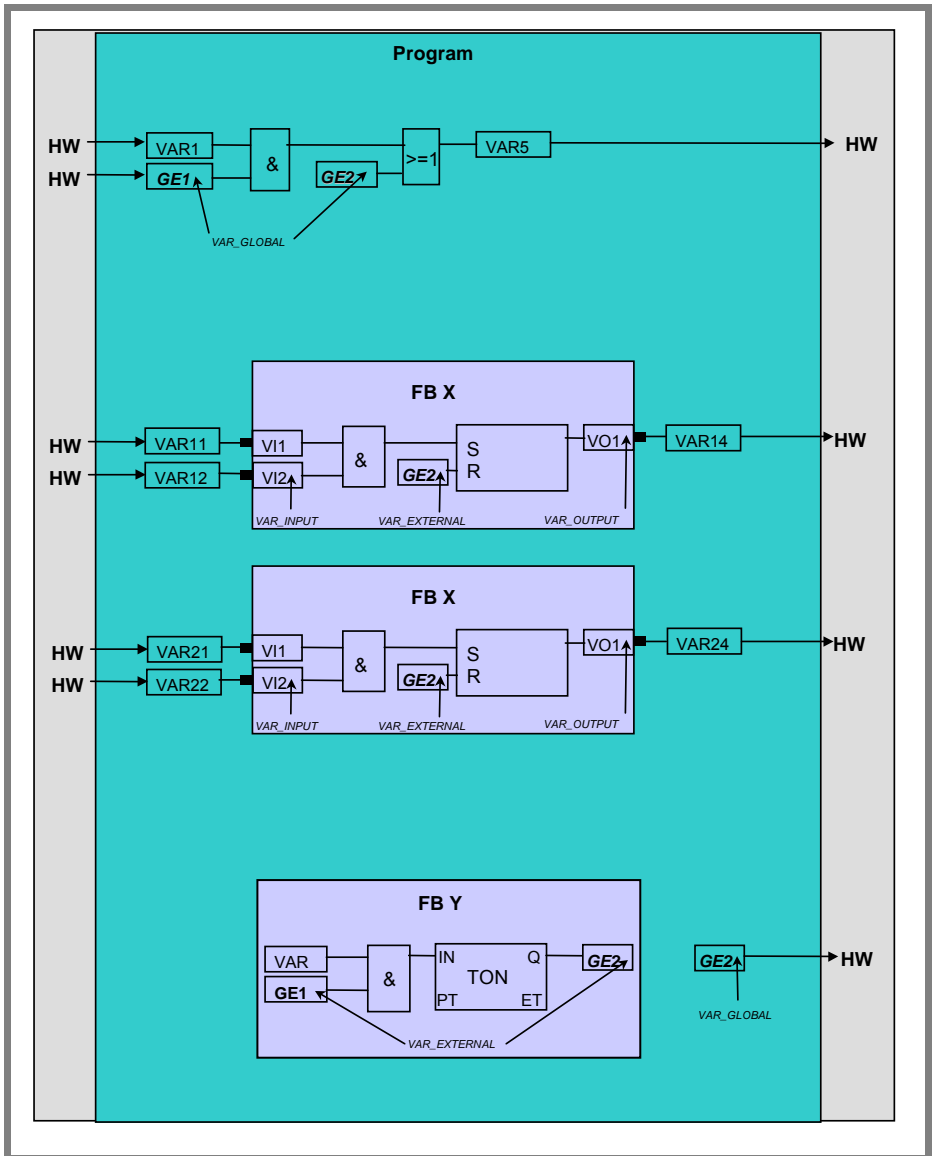


Fig. 42: Organizing variables

6.2.2 Variable Declaration Editor

Step 1: In the variable declaration editor select the variable type:

- Click on the tab with the required variable type.
Select VAR (see Fig. 43).

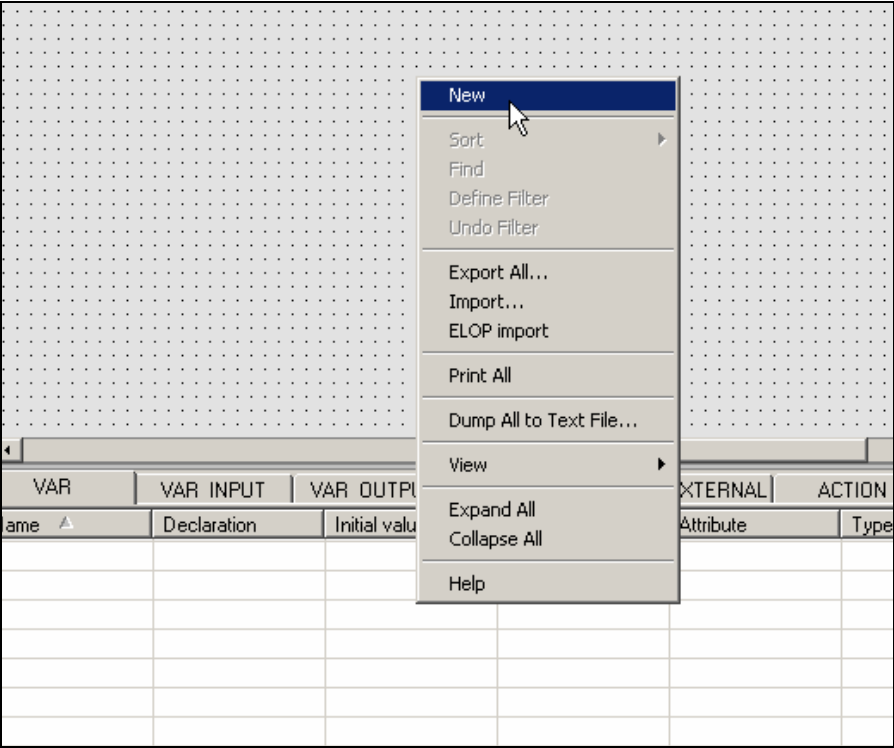


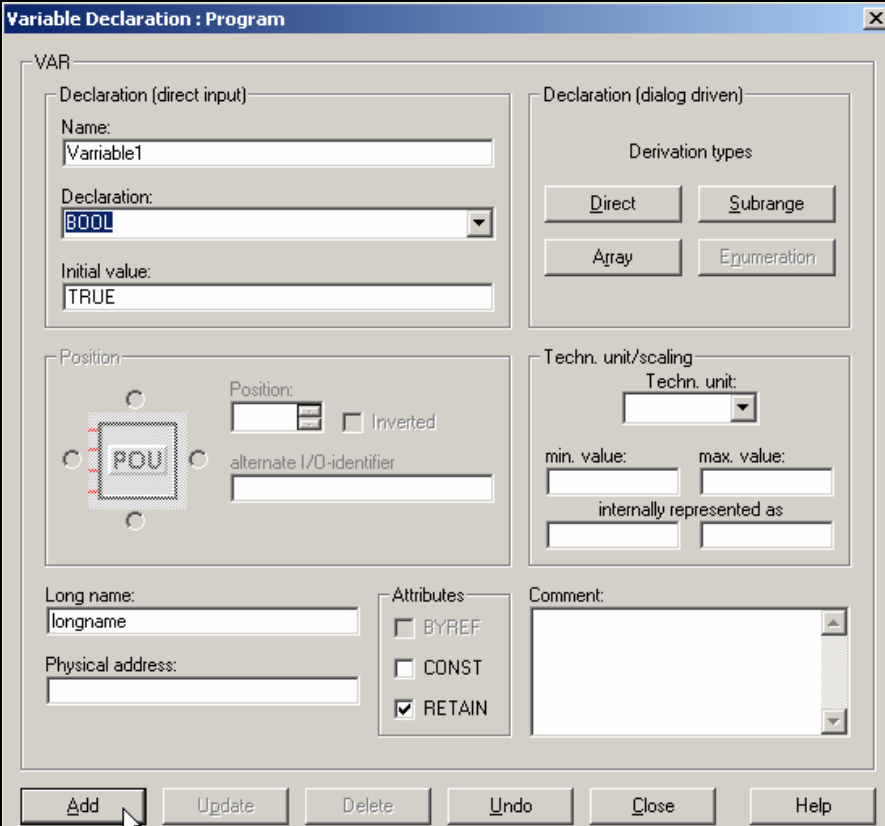
Fig. 43: Variable declaration editor

Note: Double clicking into an empty line in the table also opens the variable declaration editor.

Step 2:

Enter a variable:

- ☐ In the variable declaration overwrite the default “Var1” in the **Name** field with “Variable1”.
- ☐ In the **Declaration** field, select the data type “BOOL”.
- ☐ In the field **Long name** enter a good description for “Variable1”. The long name helps to explain the use of the variable.
- ☐ Give attributes and initial value if required.



The image shows a software window titled "Variable Declaration : Program". It contains several sections for configuring a variable:

- VAR:**
 - Declaration (direct input):** Includes fields for Name (containing "Variable1"), Declaration (a dropdown menu with "BOOL" selected), and Initial value (containing "TRUE").
 - Declaration (dialog driven):** Includes buttons for "Direct", "Subrange", "Array", and "Enumeration".
- Position:** Includes a diagram of a POU (Program Organization Unit) with input and output ports, a "Position:" field, an "Inverted" checkbox, and an "alternate I/O-identifier" field.
- Techn. unit/scaling:** Includes a "Techn. unit:" dropdown, "min. value:" and "max. value:" fields, and an "internally represented as" section with two empty fields.
- Long name:** A text field containing "longname".
- Physical address:** An empty text field.
- Attributes:** A group box containing three checkboxes: "BYREF" (unchecked), "CONST" (unchecked), and "RETAIN" (checked).
- Comment:** A large text area for additional notes.

At the bottom of the window are buttons for "Add", "Update", "Delete", "Undo", "Close", and "Help". A mouse cursor is pointing at the "Add" button.

Fig. 44: Declaration of a variable

- ☐ Add your input to the list of variables by pressing the **“Add”** button.

The dialog window remains open and the end number of the variable is counted up.

- ☐ Create more variables just by overwriting the new names and pressing the Add button.
- ☐ Close the window with **“Close”**

Result:






VAR	VAR INPUT	VAR OUTPUT	VAR GLOBAL	VAR EXTERNAL
Name ▲	Declaration	Initial value	Longname	Attribute
 Variable1	BOOL		longname	
 Variable2	BOOL		longname	
 Variable3	BOOL		longname	
 Variable4	BOOL		longname	
 Variable5	BOOL		longname	

Fig. 45: Variable table

6.2.3 Creating Logic

Step 1:

Use Drag & Drop to place logic elements from the libraries in the drawing area:

- ☐ In the Structure window, click on the “+” symbol to open the library **StandardLibs**.
- ☐ In the **StandardLibs**, open the library **IEC61131-3** and there **Bitstr**.
- ☐ Click on the **AND** function and drag the object out of the Structure window into the drawing area. A preview of the object is displayed while dragging.
- ☐ After you release the mouse button, the object is placed at the cursor position (Drag & Drop).

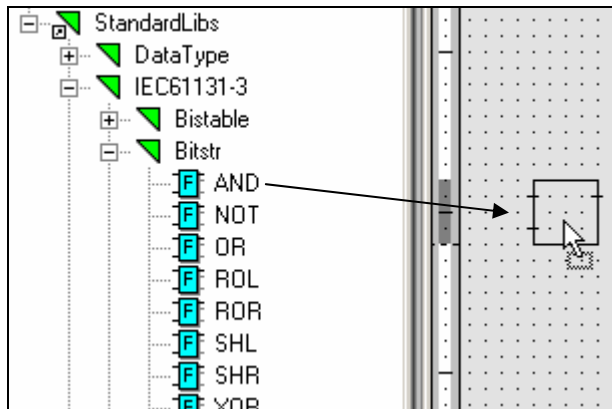


Fig. 46: POU drag & drop

Note: The procedure for adding standard objects from libraries to a logic also applies to user-defined function blocks.

Step 2:

Complete the page data:

Since placing the **AND** object is the first change to the contents of this page, the dialog window “Edit Page Data” opens automatically.

In the fields **Short name** and **Long name** enter a good description for the page.

Edit Page Data

Page data

Page no. **A/0** Short name: ESD logic

Long name:

created by (logon-ID): **e.kull** Date: **22/09/05**

tested by: Date:

Labeling systems

ULS:

PPLS:

☒ Inherit labeling systems from page options

Page-ID: ☒ Inherit

OK Cancel Help

Fig. 47: Editing page data

Step 3:

Complete the logic with additional function blocks:

- ☐ Insert additional objects from the libraries into the drawing area as described in **Step 1**.
- ☐ Duplicate identical objects by keeping the **Ctrl** key pressed while dragging an already existing object to another position in the drawing area.

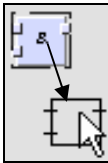


Fig. 48: Copy object

Note:

Objects cannot be positioned on top of each other. In this case, ELOP II aborts copying followed by a signal.

Step 4:

Switch-on the grid and zoom in:

- ☐ Switch-on the grid with the **Toggle grid** button in the symbols bar.
- ☐ Zoom into the screen section you want to enlarge.

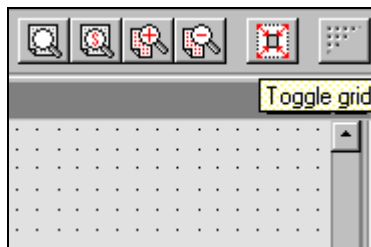


Fig. 49: Buttons for zoom and grid

Note:

The buttons on the right side refer to the drawing area of the open object.

Step 5:

Add variables to your logic:

Note: You use Drag & Drop to copy variables from the variables list into the value field area (light gray shaded margins on the left and right side of a page) or to a position in the drawing area. A preview of the value field is shown while dragging.

- In the variables list, click on a variable name and drag the variable into the value field area (light gray shaded margins on the left and right side of a page) or to a position in the drawing area. A preview of the value field is shown while dragging.

After the variable has been dropped, the variable name appears in the value field.

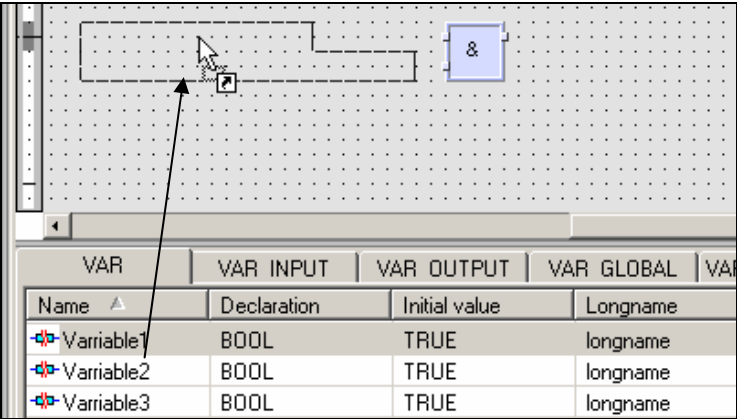


Fig. 50: Drag & drop variables

Note: You cannot drop variables on top of other objects in the drawing area.

Step 6:

Draw segments between the variables (value fields) and function blocks:

- ☐ Position the cursor above the node where you want to start the segment (variable output).
- ☐ Press the left mouse button and draw a segment to the node where you want the segment to end. Release the mouse button.

The result is an automatically routed connection between the two nodes.

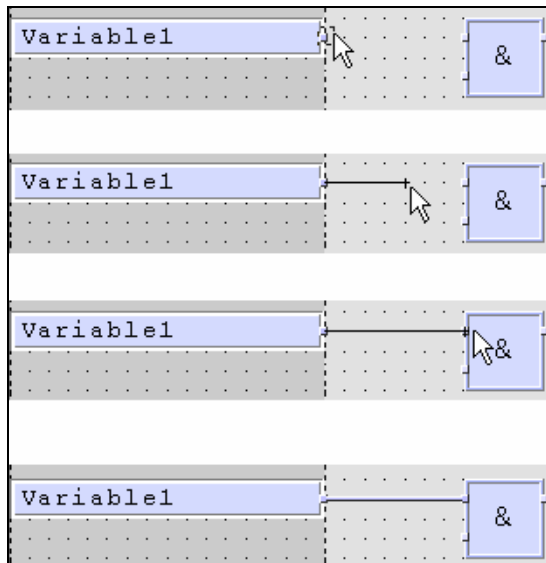


Fig. 51: Draw segments

*Note: The color of a segment depends on the data type (BOOL, Integer, Real etc., see Help).
You cannot connect nodes of different data types!*

Example of a simple exercise:

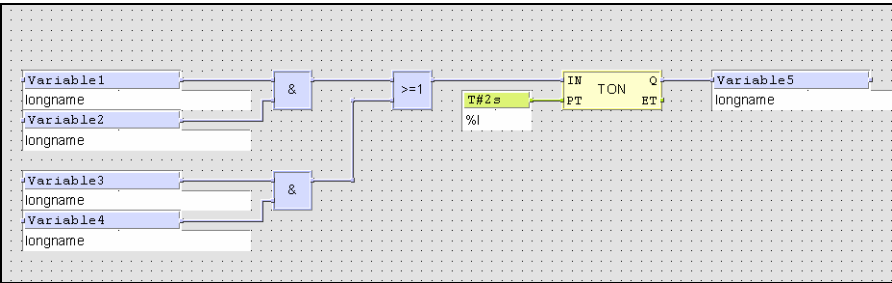


Fig. 52: Simple exercise

Step 7: Save the logic and close the program:

- Save the logic by clicking the **Save** button in the toolbar.

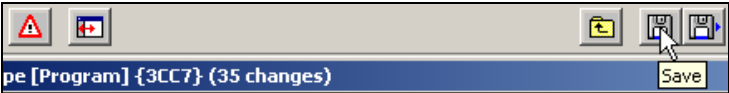


Fig. 53: Save program

- Close the program

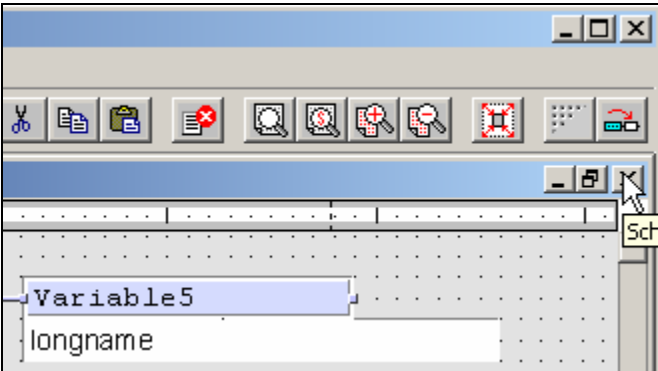


Fig. 54: Close program

6.3 Offline Simulation of Logic (Option)

The new logic can be tested without a real PES on the PC. This first test is always recommended (see Safety Manual)!

Step 1: Enter the Offline simulation:

- ☐ Click on the resource.
- ☐ Open the context menu of the resource.
- ☐ Select **Offline-Simulation**.

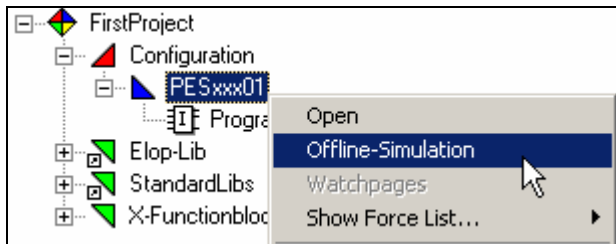


Fig. 55: Open the Offline simulation

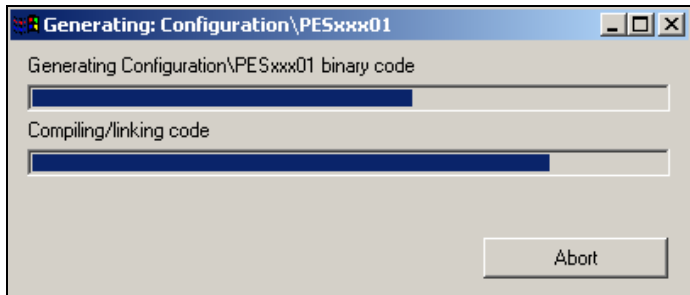


Fig. 56: Code generation for Offline simulation

Note: A new tab OLS (Offline Simulation) opens in the Structure window.

Step 2: Start the Offline simulation:

Note: Beginning with ELOP II version 4.1 the Offline simulation starts automatically.
For earlier versions a manual cold start is required:
Click on the **Cold boot** button (blue triangle).
The status changes from “Stopped” to “Running” after the start.

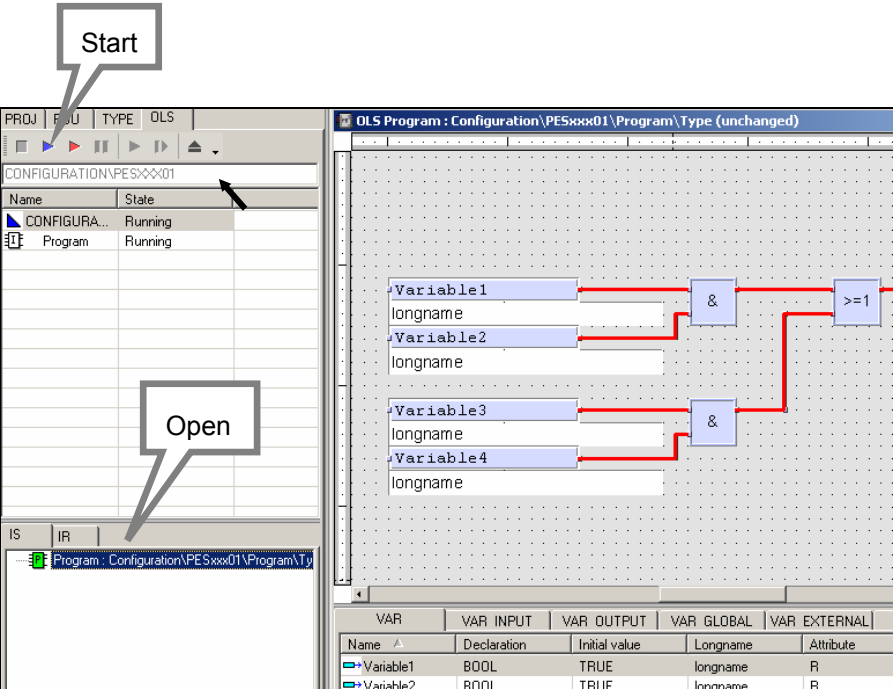


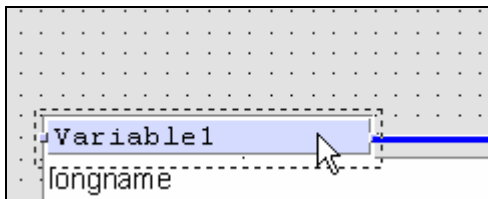
Fig. 57: Start the Offline simulation

Step 3:

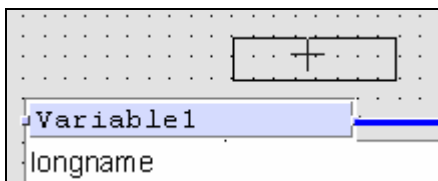
Change the state of the signals and test the logic:

Change signal state with an Online test field (OLT field):

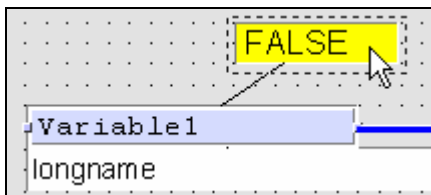
- ☐ Click inside a value field and keep the mouse button pressed.



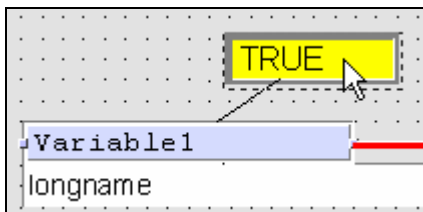
- ☐ Drag the cursor out of the value field and drop the OLT field that follows the cursor at a free location.



- ☐ Relocate the OLT field if necessary.



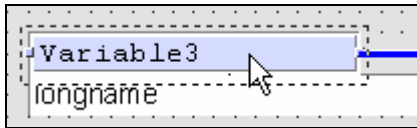
- ☐ Change the signal state by double clicking in the OLT field.



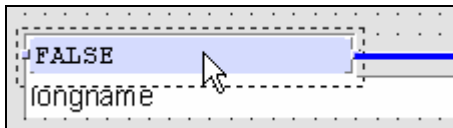
*Note: After having inserted OLT fields, the question “Save change?” appears when closing the function block in the drawing area. If you click **Yes**, the OLT fields will be saved. In case of **No**, the OLT fields you have created are rejected. New OLT fields will not affect Code version or RUN version.*

Change signal state directly in the value field:

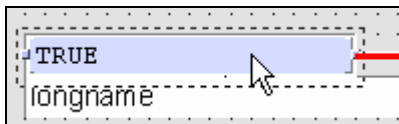
- ☐ Position the cursor over the value field whose value you want to change.



- ☐ Press and hold the ALT key. The signal state is displayed in the value field instead of the signal name.



- ☐ Change the signal state by clicking in the value field.



- ☐ Release the ALT key. The signal name is displayed again.

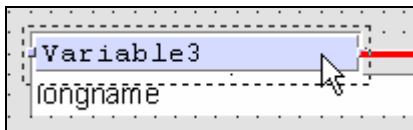


Fig. 58: Change the value field with the ALT key

Note: You can only change values which are not created by the logic.

Step 4:

Close the Offline simulation:

- Click on the **Close OLS** button in the symbols bar.

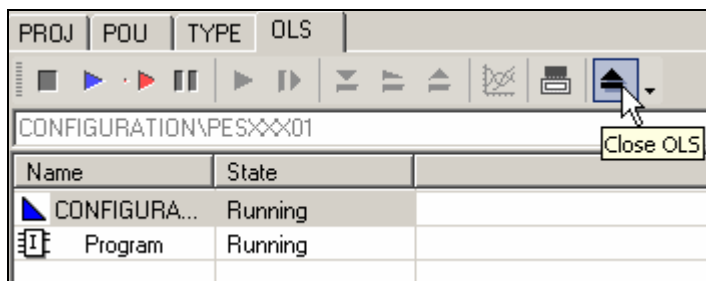


Fig. 59: Close the Offline simulation

6.4 PLC Cabinet Layout and Variable Assignment

Step 1: Assign the resource type (RT):

- ☐ Click on the resource in the Structure window.
- ☐ Open the context menu of the resource.
- ☐ Click on **RT assignment**, and select the type of **HIMA PLC** you have (refer to main catalogue H41q/H51q).
In example **H41q** with **F8652E, HS** configuration.

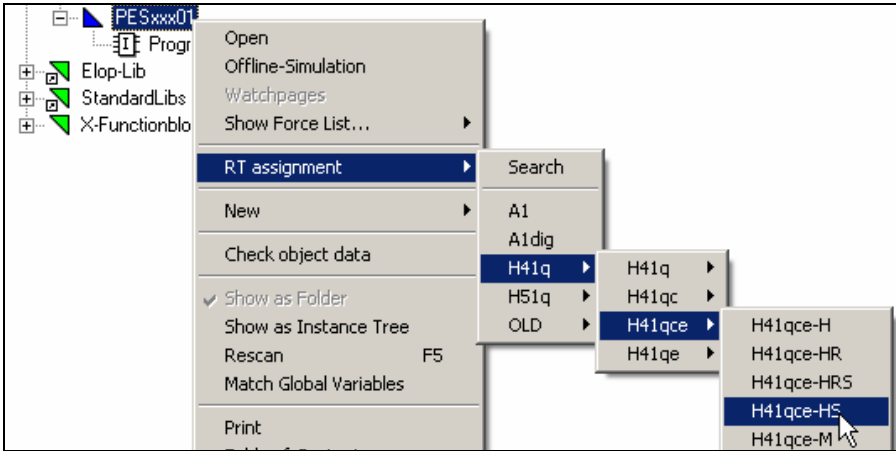


Fig. 60: Assigning a resource type

Step 2:

Open the cabinet layout:

- ☐ Open the context menu of the resource.
- ☐ Select **Edit cabinet layout**.

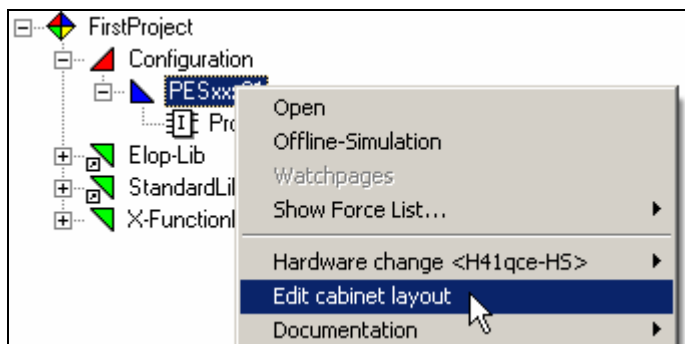


Fig. 61: Edit cabinet layout

Step 3:

Insert I/O Modules into the Rack (Example H41q):

Note: If resource H51q is selected, insert at first a rack with the right address (coding switch on F 7553)

- “Insert” modules into the rack by dragging and dropping components from the **Modules** tab on the bottom of the page to the required positions in the rack.

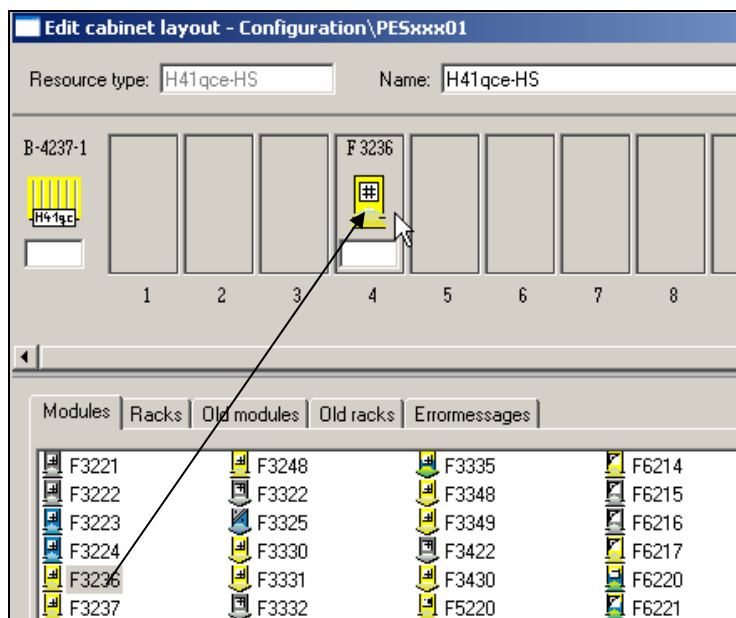


Fig. 62: Insert modules

The module icons have the following meanings:

Yellow I/O icon ⇒ safety-related
 Grey I/O icon ⇒ not safety-related
 Blue I/O icon ⇒ for explosion hazardous areas

Arrow to the icon ⇒ Input module
 Arrow from the icon ⇒ Output module

Examples:



⇒ digital I/O module

⇒ analog I/O module

Step 4:

Open the I/O module assignment:

- Double-click on a module icon (F 3236 in our example) to open the “Edit Tag name” window.

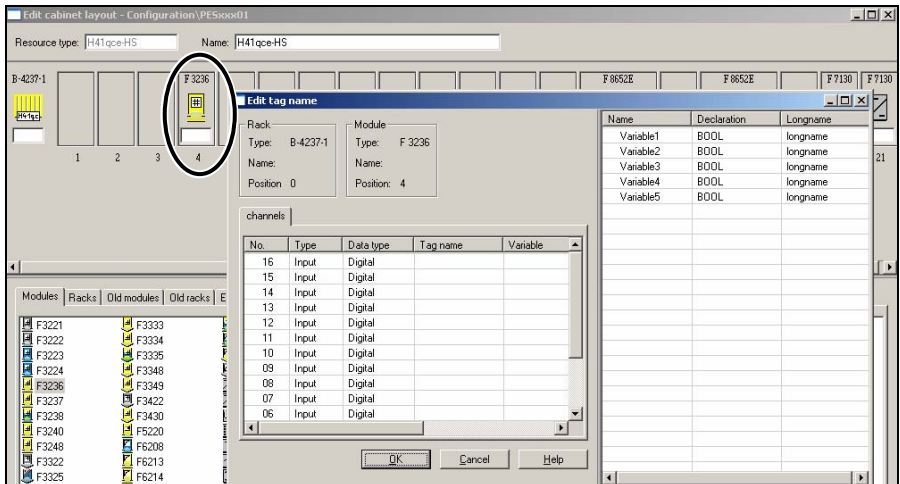


Fig. 63: Module assignment

Step 5: Assign variables to the inputs and outputs of an I/O module:

Note: Fig. 64, left, shows the 16 digital input channels of the F 3236 module. Fig. 64, right, lists all variables whose data type (BOOL) matches the channels of the F 3236 module and which have not been assigned to an I/O channel yet.

- Click on a name in the variable list, drag the variable into the I/O channel list and drop it on an I/O channel.

Assigned variables are displayed in the I/O channel list at their channel and are removed from the variable list.

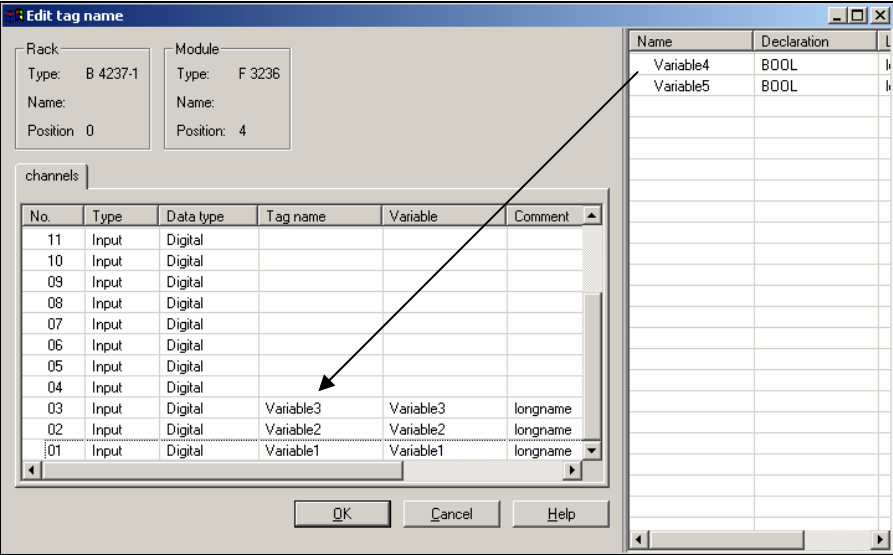


Fig. 64: Assign variables to I/O channels

Note: A variable can only be assigned to one I/O channel.

Step 6: Remove variable assignment (optional):

Sometimes it may become necessary to remove a variable assignment.

- ☐ Double click to the variable
- ☐ Mark and delete the Tag name
- ☐ Remove the check mark at “Assign variable”.
- ☐ Press **OK**.

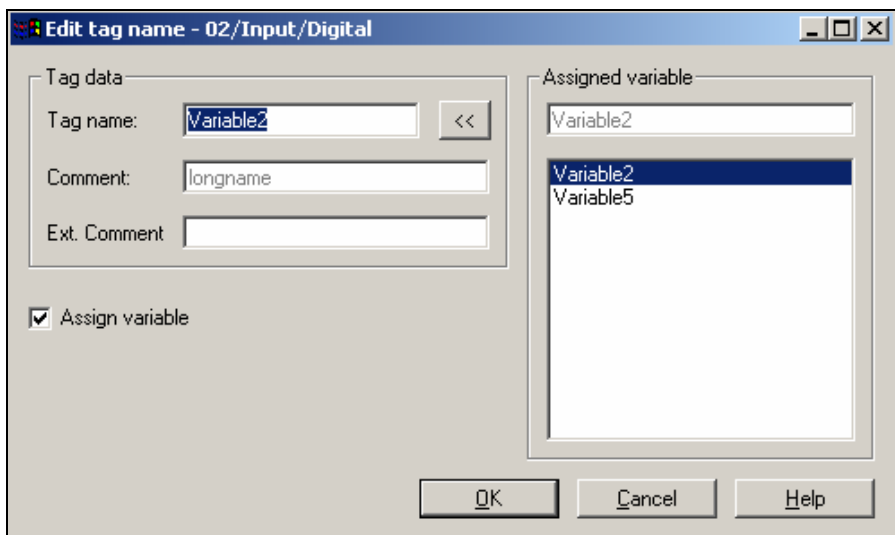


Fig. 65: Remove variable assignment

- ☐ Close the window “Edit tag name” with OK.
- ☐ Close the window “Edit cabinet layout” with OK.

6.5 Running a Program on a PES

6.5.1 Compile Program

Before the logic, created by the user, can run on a controller, the logic must be converted into code executable by the CPU:

Step 1: Start the Code generator:

- ☐ Click on the resource.
- ☐ Open the context menu of the resource.
- ☐ Select **Code Generator**.

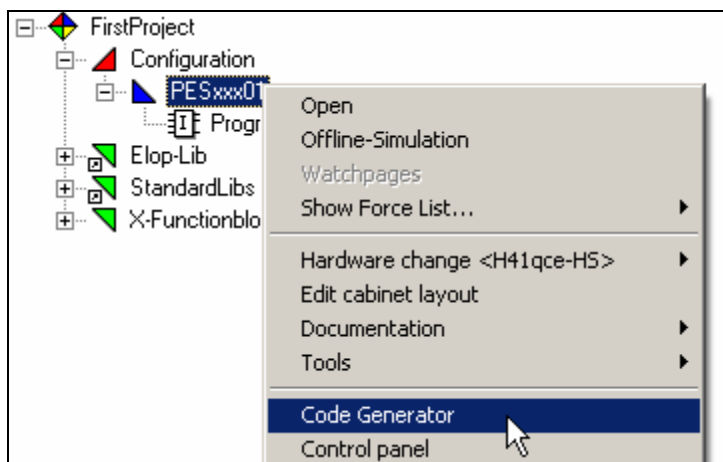


Fig. 66: Start the Code generator

Step 2: Confirm the start of the code generator:

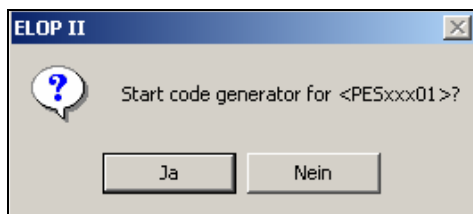


Fig. 67: Confirm start

The code generator reports the progress of the compilation in its own window.

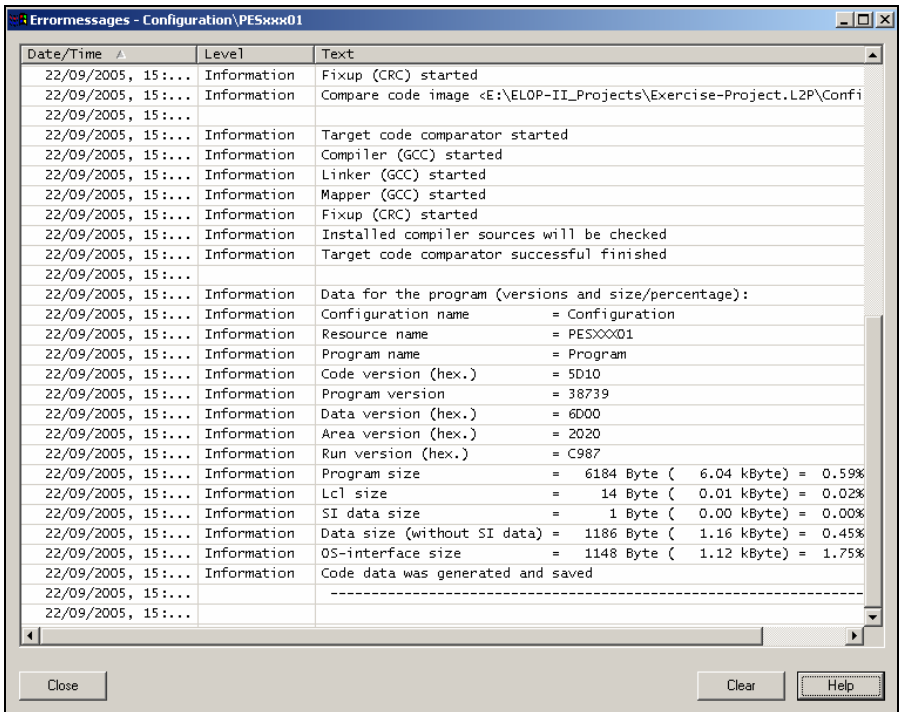



Fig. 68: Messages of the code generator

After the code generation has been successfully accomplished, additional elements appear in the Structure window. These are:

- H5CCode.VGL File for the comparison
- H5CODE.COD Code file with compiled logic
- Program.ERR File with error messages

Note: If errors occurred during code generation, more detailed information can be called up via the Error state viewer .

6.5.2 Setup Communication between PC and PES

Note: Before communication can be established, a configuration of the bus connection and the bus settings between the PC and the controller are necessary.

Step 1: Open the properties of the configuration:

- ☐ Click on the Configuration in the Structure window.
- ☐ Open the context menu of the configuration.
- ☐ Select **Properties**.

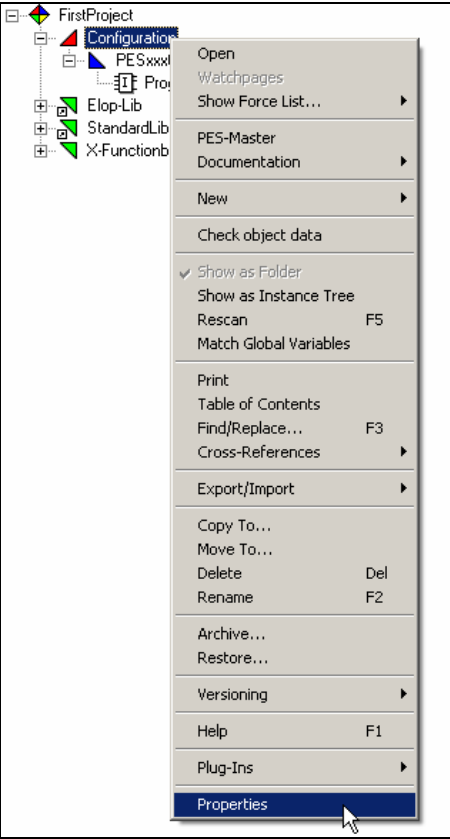


Fig. 69: Open the configuration properties

Step 2:

Add a new bus:

- ☐ In the window “Properties: Configuration“, select the tab **Busses**.
- ☐ Click on the **Add** button to define a new bus.
- ☐ In the dialog window, “Add communication system“ enter a name for the new bus in the **Name** field.
- ☐ The bus **Type** is always **HIBUS**.
- ☐ Press **OK** to add your entry.

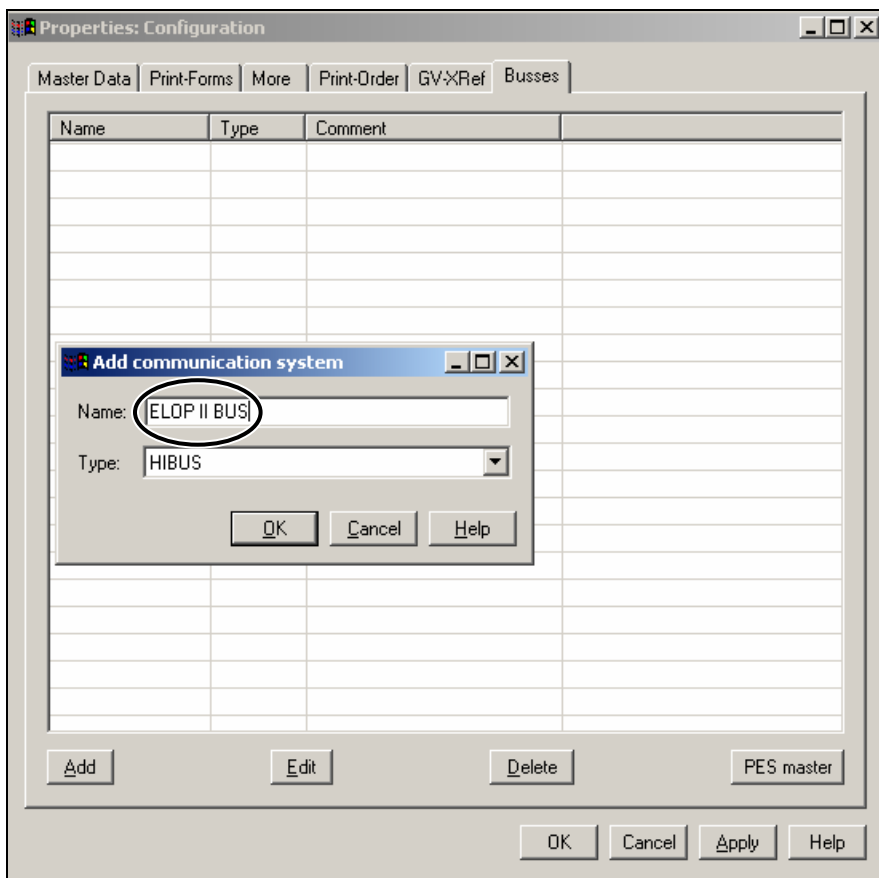


Fig. 70: Adding a bus

The window “Edit HIBUS – *Bus name*” opens. Select the tab **Station**.

Comment:

Station Parameter

Name	Type	BSN	CU	CM	

Add Edit Delete

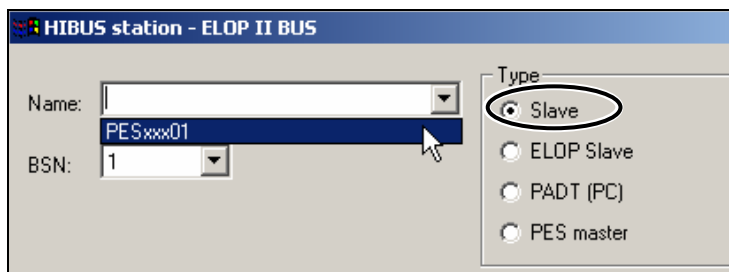
OK Cancel Help

Fig. 71: Edit HIBUS

Step 3:

Add a resource to the list of bus stations:

- ☐ Click on **Add** to define a new bus station.
- ☐ In the window “HIBUS station – *Bus name*”(see Fig. 72), select the resource (controller) with which you want to communicate from the drop-down list “**Name**”.



HIBUS station - ELOP II BUS

Name:

BSN:

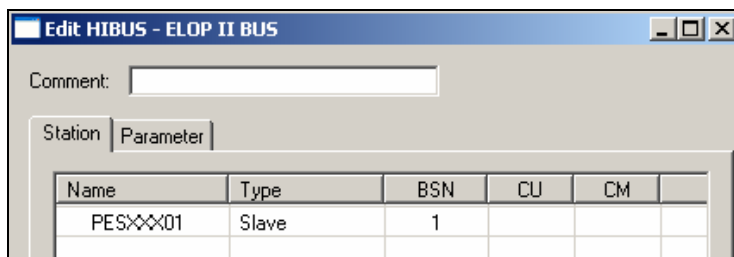
Type

- ☒ Slave
- ☐ ELOP Slave
- ☐ PADT (PC)
- ☐ PES master

Fig. 72: Add a resource to the list of bus stations

- ☐ Define the **Bus Subscriber Number (BSN)** in accordance with the setting on the central module of the controller.
- ☐ Select **Type** "Slave" for a resource.
- ☐ Confirm with **OK**.

Result:



Edit HIBUS - ELOP II BUS

Comment:

Station | Parameter

Name	Type	BSN	CU	CM
PESxxx01	Slave	1		

Fig. 73: Result after step 3

Step 4:

Add a PC to the list of bus stations:

- ☐ In the window “Edit HIBUS – *Bus name*” click on **Add** to define a new bus station.
- ☐ In the window “HIBUS station – *Bus name*”, enter the name of the PC that is connected to the resource in the **Name** field. The name can be chosen freely.
- ☐ Define the **Bus Subscriber Number (BSN)** for the PC. “31” is normally used.
Important: The BSN of the PC must not collide with a BSN of a controller.
- ☐ Select **Type** “PADT (PC)” for the PC.

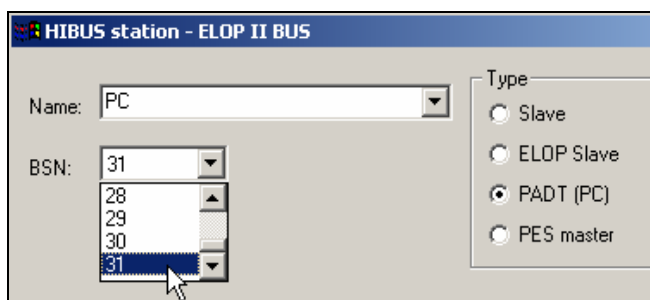
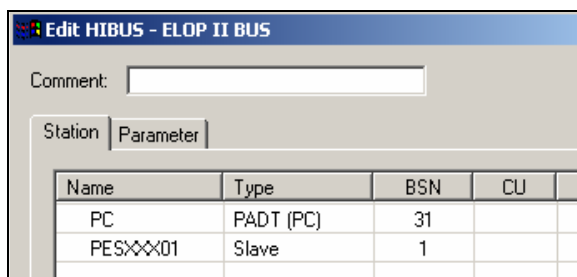


Fig. 74: Add a PC to the list of bus stations

Result:



Name	Type	BSN	CU
PC	PADT (PC)	31	
PESXXX01	Slave	1	

Fig. 75: Result communication system

Step 5:

Define the bus parameters:

- ☐ In the “Edit HIBUS – *Bus name*” window, select the tab **Parameter** to define the **Baud rate**, the number of **Stop bits** and the **Parity** of the serial bus.

Note:

The settings must correspond with the settings of the controller (standard baud rate is 57600 bps, adjusted by dip switches on the central module).

- ☐ Save all your bus settings with **OK**.

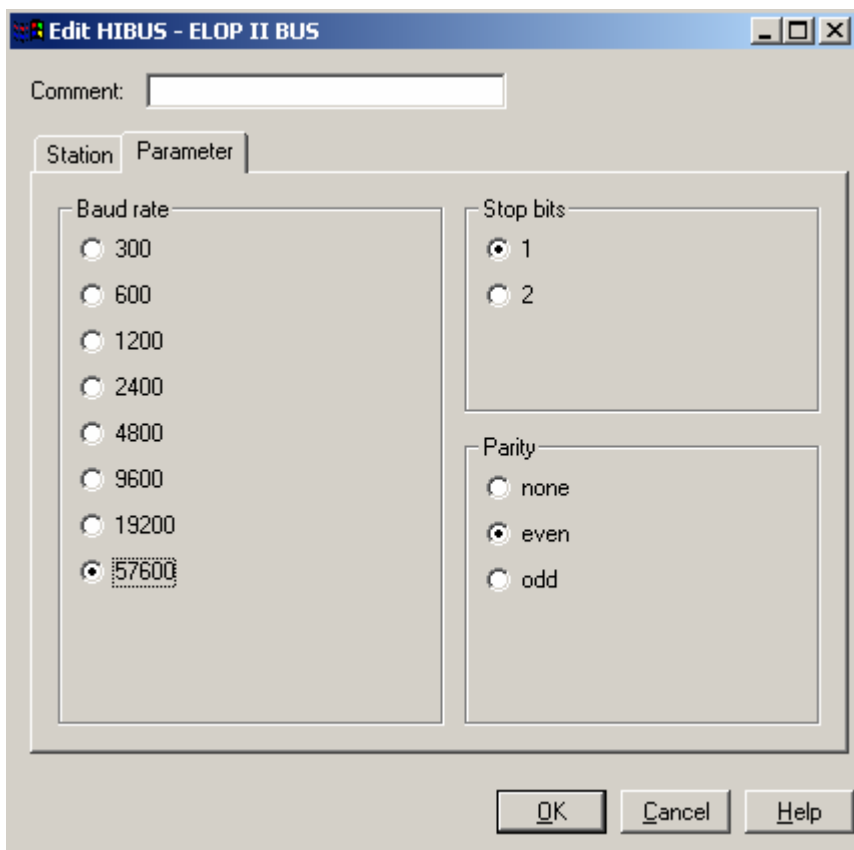


Fig. 76: Define the bus parameter

Step 6:

Open the properties of the resource:

- ☐ Click on the resource in the Structure window.
- ☐ Open the context of the resource.
- ☐ Select **Properties**.

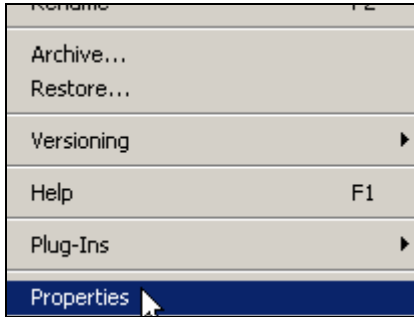


Fig. 77: Open the resource properties

Step 7:

Select type of communication:

The communication between programming device (PADT) and controller (PES) can be established either over serial link (RS485) or over Ethernet, depending on available hardware:

Hardware for serial link (see data sheets):

- Standard cable BV 7043 from PC (RS232) to PES (RS485) for temporal use, *or*
- RS485 Bus with H 7505 and H 7506, *or*
- Interface converter USB/RS485 with BV 7055, or special cable, *or*
- RS485 PCI module and B V7049.

Hardware for Ethernet communication (see data sheets):

- Communication module F 8627X or F 8628X, and
- Central module F 8652X (H41q) or F 8650X, and
- Network interface card in the PC.

- ☐ Open properties of resource, tab **PADT (PC)**.
- ☐ Select Communication Type and click on **Ethernet** or **Serial**.

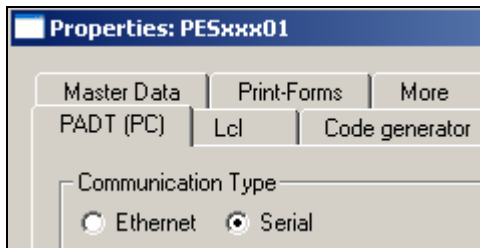


Fig. 78: Select communication type

- ☐ If you have selected **Serial** continue with **Step 8a**.
- ☐ If you have selected **Ethernet** continue with **Step 8b**.

Step 8a: Setup of serial communication in tab PADT (PC):

- ☐ In the **Bus** field, select your bus from the list of the configured busses.
- ☐ In the **PADT (PC)** field, select your PC from the list of configured PADTs on the bus.
- ☐ Under **PC interface** select the serial port which connects to the controller.
- ☐ Normally there is no need to adjust the connection settings (see “Help”).

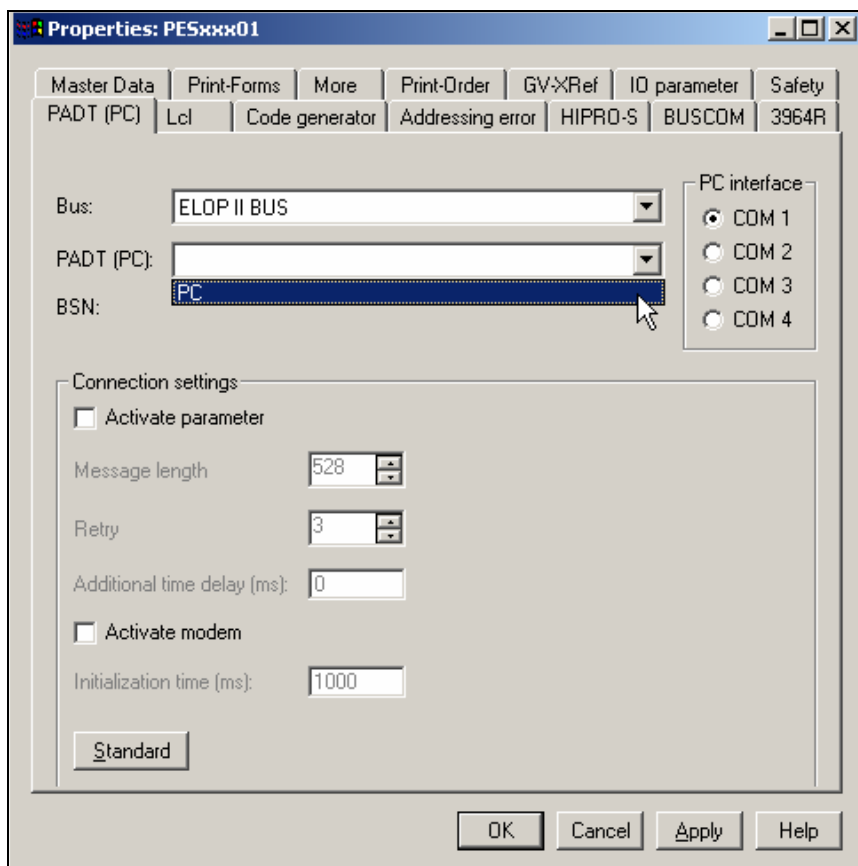


Fig. 79: Tab PADT (PC) of the resource properties

Step 8b:

Setup of Ethernet communication:

- ☐ Check the IP addresses of communication modules.
Therefore see data sheet of F 8627X or F 8628X.
- ☐ Check IP address of PC (network interface card).

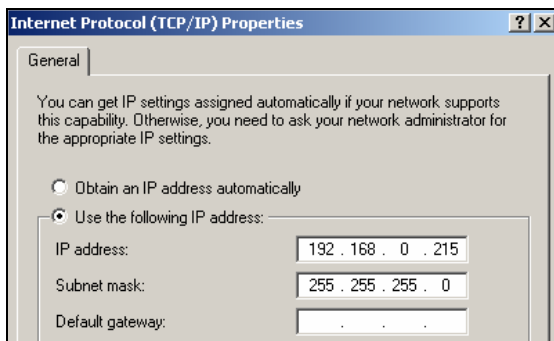


Fig. 80: IP address of PC

- ☐ Open the resource properties.
- ☐ Select **Channel 1** or **Channel 2**.

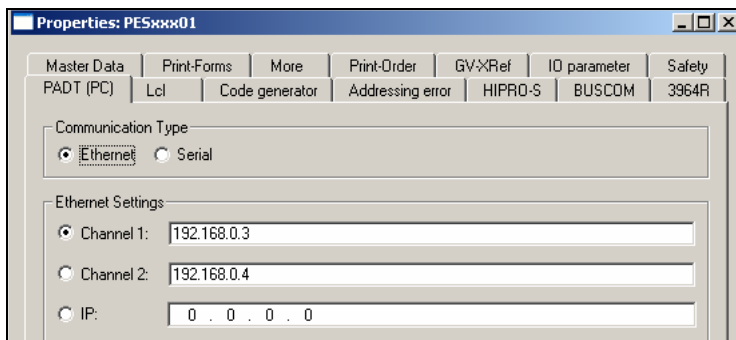


Fig. 81: Ethernet settings

Note:

The IP addresses of the controllers are automatically determined by ELOP II from the resource names. A manual input of an IP address in the field "IP" is only required (and possible) for special operating systems.

6.5.3 Download and Start

Step 1:

Open the control panel:

- ☐ Click on the resource.
- ☐ Open the context menu of the resource.
- ☐ Select **Control panel**.

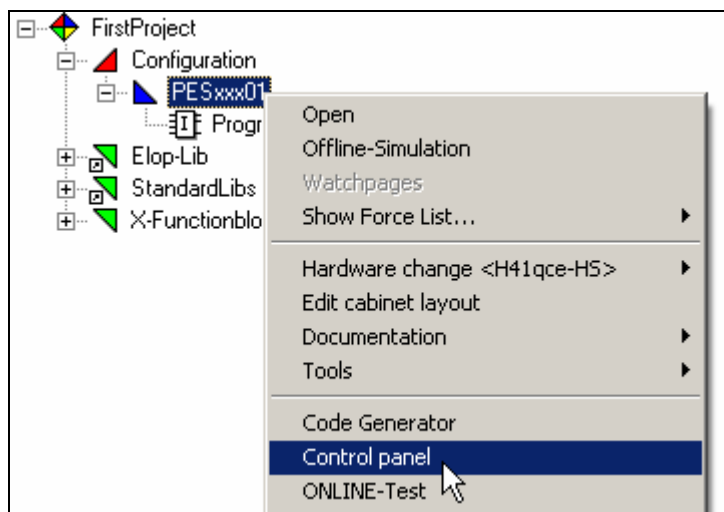


Fig. 82: Start the control panel

Note: You can leave the control panel open for further actions. Simply select the tab **CP** in the Structure window.

Step 2:

Start the download procedure:

- In the control panel, click on the button **Download/Reload**

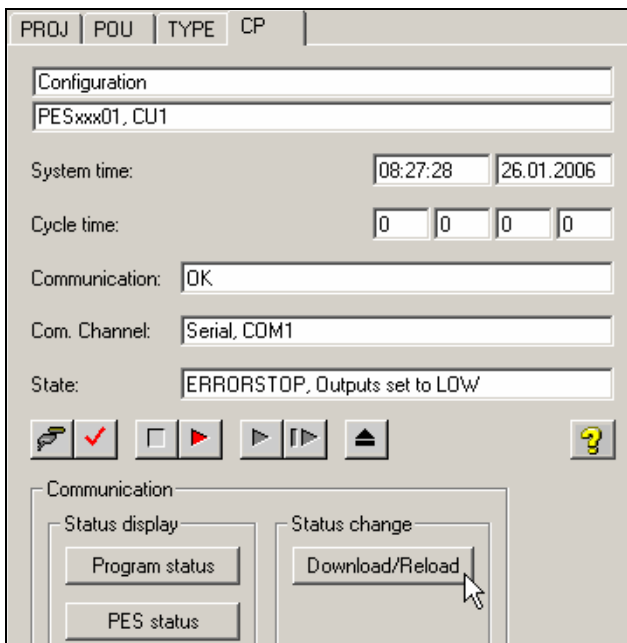


Fig. 83: Download/Reload

Note: In a running system download or reload must only be carried out by trained personnel and with the permission of the person responsible for the facility, because especially a download causes a shutdown.

Make sure that the names of function block instances and variables do **not begin with blanks or “.”** and do **not end with blanks!**



Otherwise errors in the program execution may occur after reload.

If names of function block instances or variables are changed, the corresponding function block instances or variables are initialised during reload the same way as during download!

- In the window “Download/Reload”, click on **OK** to start the download.

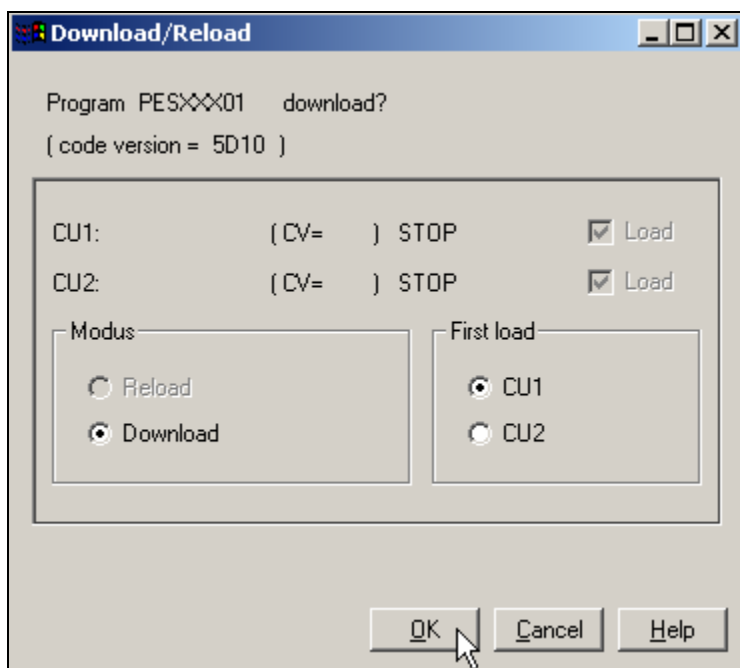


Fig. 84: Start download

- Wait for the download to be terminated.

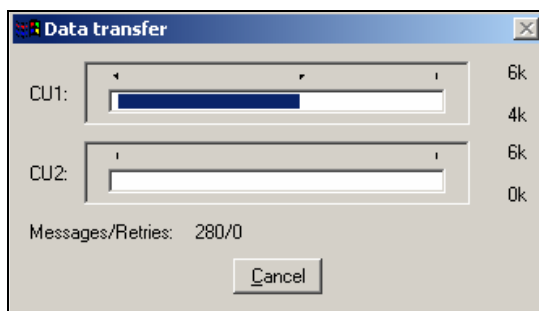


Fig. 85: Data transfer

Messages, indicating status change (must be confirmed with “OK”):

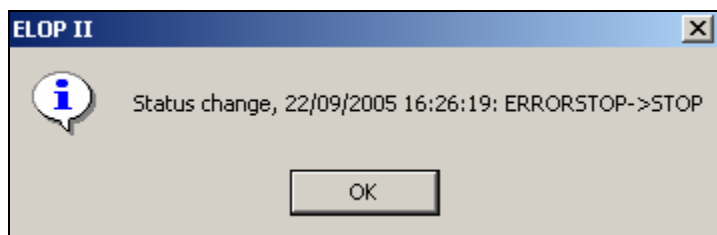


Fig. 86: Status change

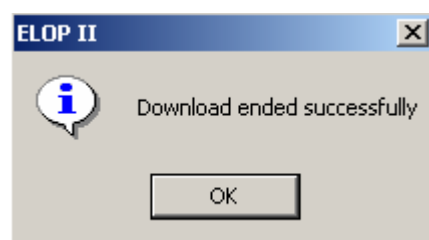


Fig. 87: Download successful

Step 3:

Start the controller:

- ☐ In the control panel, click on the **Start** button.

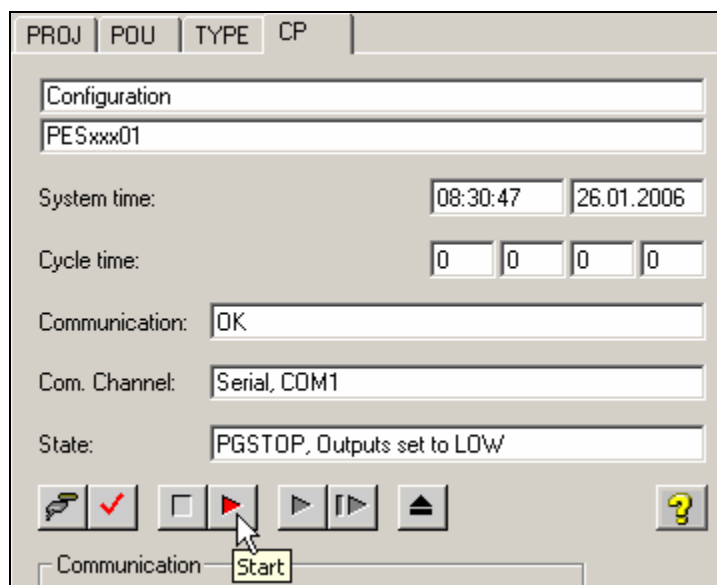


Fig. 88: Start

- ☐ For a new project select **Cold start** as start mode:

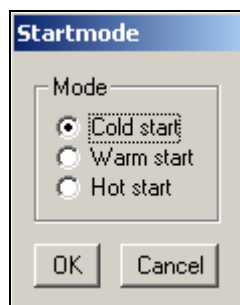


Fig. 89: Cold start of the controller

- After you pressed **OK**, a safety query makes sure that the controller is not started inadvertently.

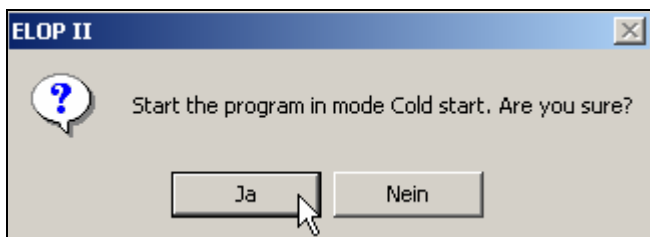


Fig. 90: Safety query

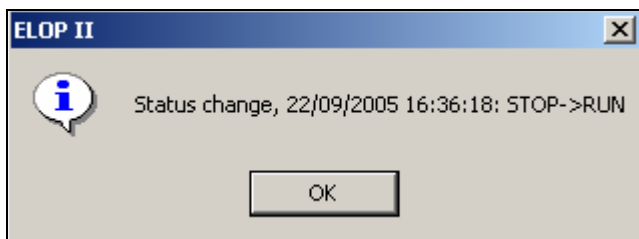


Fig. 91: status change

Result:

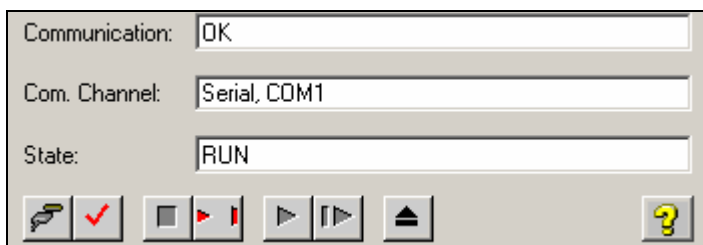


Fig. 92: PLC is in RUN mode

6.6 Online Test

Note: The Online Test can only be started if the code version of the user program in the controller is identical to the code version of the project on the PC.

Step 1:

Open the Online test:

- ☐ Click on the resource.
- ☐ Open the context menu of the resource.
- ☐ Select **ONLINE-Test**.

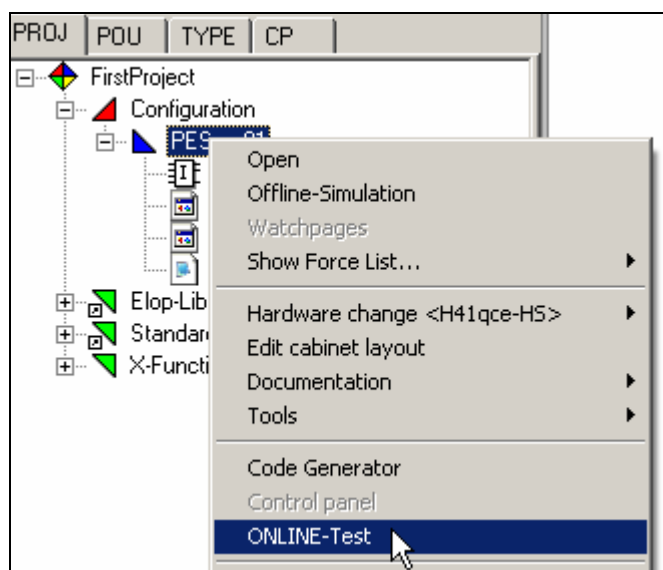


Fig. 93: Invoke the Online Test

Step 2:

Three ways to create Online test fields (OLT):

1. Direct mouse operation:

- ☐ Click to the value field or interface point and keep mouse key pressed

Drag the mouse from the object and release mouse key.
Screen shows now a preview of the OLT field.

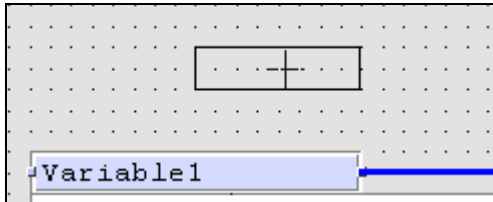


Fig. 95: Drag OLT field

- ☐ Place the OLT field with another mouse click to the drawing area

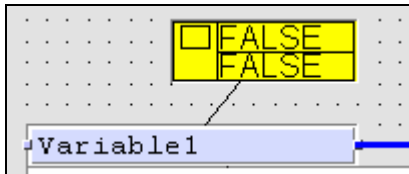


Fig. 96: OLT field

2. Context menu:

- ☐ Right mouse click to the value field or interface point.

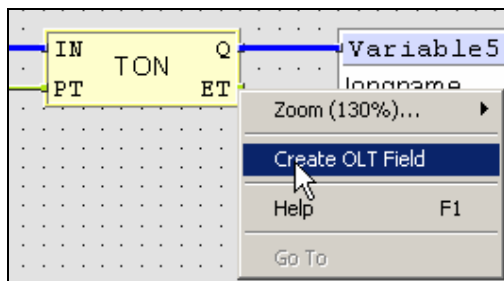


Fig. 97: Creating OLT field via context sensitive menu

3. Free OLT:

- ☐ Open context menu of drawing area (right mouse click to drawing area) and click **Create OLT Field**.

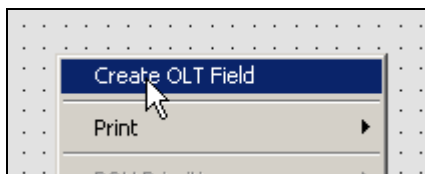


Fig. 98: Creating free OLT field

- ☐ Right mouse click to OLT field "Define Element"

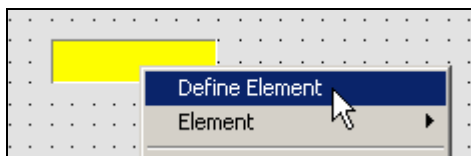


Fig. 99: Define element

- ☐ Insert the name of variable (here Variable2) you want to see in the OLT field

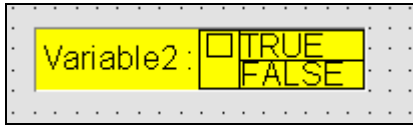


Fig. 100: Assign OLT field

Step 3:

Close the Online test:

- ☐ Click on the **Close** button.



Fig. 101: Close the Online test

Note: The Online test must be closed before modifications can be made to the logic!

6.7 Forcing Inputs and Outputs

Step 1:

Prepare for Online test:

- ☐ Start the Online test as described in Chapter 6.6, Step1.
- ☐ Create Online test field as described in Chapter 6.6 Step2.

Step 2:

Change and activate the force value:

The OLT field is divided into three sections:

1. Display of the single switch and the state of the main switch.
2. Display of the current signal state of the input or output.
3. Force value display.



Fig. 102: OLT field of a Boolean variable

- ☐ Change the force value and the force switch by double clicking in the respective section of the OLT field.

The force switch and force value **changes directly in the controller**. If the master force switch has not yet been set, ELOP II asks whether the master force switch should be set (see figure below).

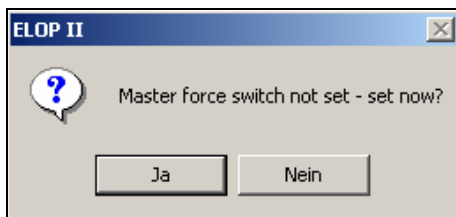


Fig. 103: Check back to set master force

Note: It is recommended not to set the master force switch via an OLT field, otherwise an already set single switch could become active inadvertently.

Possible states of the OLT field:

	Single force switch	Master force switch	Force value	Cur. value	Output
<input type="checkbox"/> TRUE <input type="checkbox"/> FALSE	Not active	Not active	FALSE	TRUE	Current value
<input type="checkbox"/> TRUE <input type="checkbox"/> FALSE	Active	Not active	FALSE	TRUE	Current value
<input checked="" type="checkbox"/> TRUE <input type="checkbox"/> FALSE	Not active	Active	FALSE	TRUE	Current value
<input type="checkbox"/> TRUE <input checked="" type="checkbox"/> FALSE	Active	Active	FALSE	TRUE	Force value
<input type="checkbox"/> TRUE <input checked="" type="checkbox"/> TRUE	Active	Active	TRUE	TRUE	Force value

Step 3:

Open the control panel:

- ☐ Click on the resource.
- ☐ Open the context menu of the resource.
- ☐ Select **Control panel**.

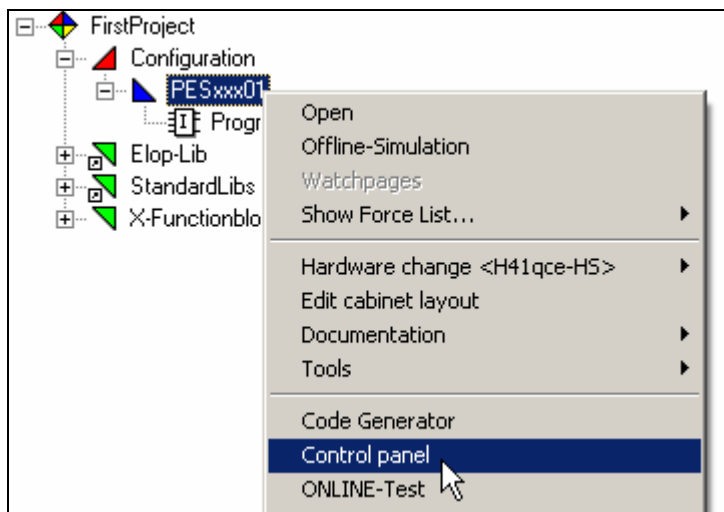


Fig. 104: Opening the control panel

Step 4:

Check Force switch settings:

- In the control panel, click on the **Force switch** button.

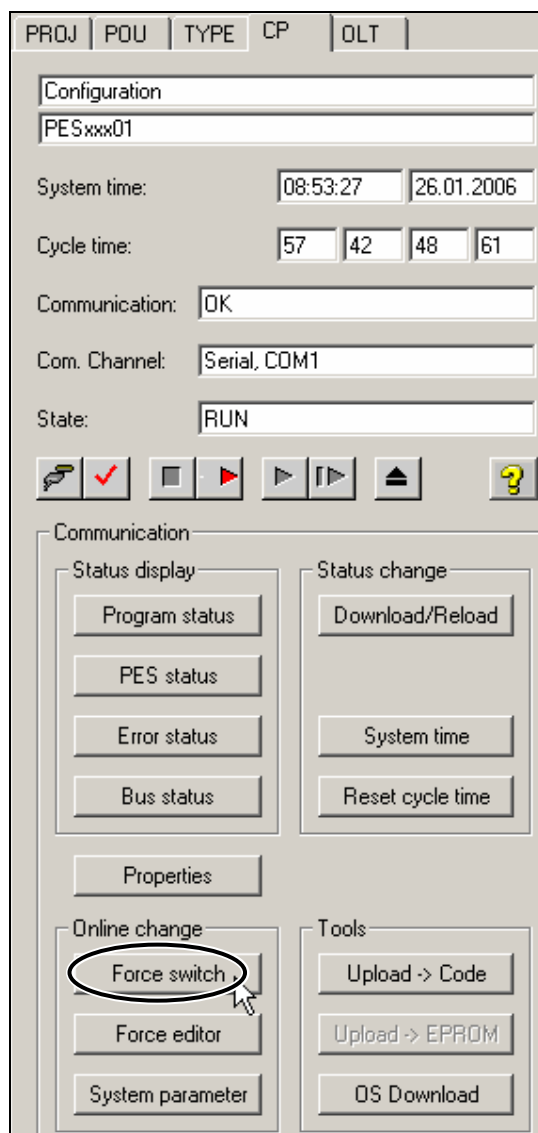


Fig. 105: Check Force switch settings

The “Force switch” window shows the status of the **Master force** switch and the number of **Activated force switches**, separately for inputs and outputs.

- ☐ Activate the **Set/Reset** checkboxes for the inputs or outputs (see figure below).

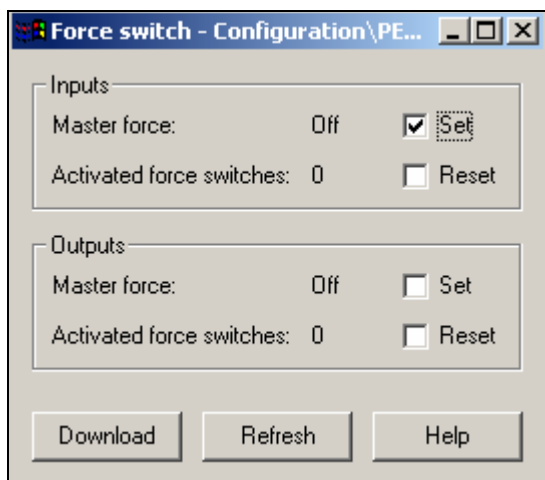


Fig. 106: Activate Force switch

Example:

- The actual value of the hardware input is FALSE.
- The Force Value is TRUE, set by a mouse double-click.
- Master force is on.
- Result: The logic is working with the forced signal TRUE.

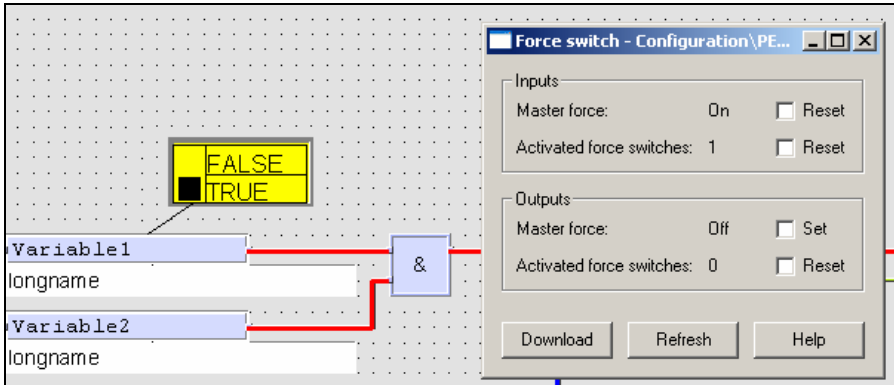


Fig. 107: Forcing of variable

Note: If OLT fields have been added to the logic and you close the function block, you are asked whether you want to save the changes for future use. Saving OLT fields does not affect the code version of the project.

*Note: If an input variable is forced, the forced value is used everywhere in the logic.
If an output variable is forced, only the physical output is forced. If the output variable is used in the logic, the value of the variable is not changed by the forcing command, but still contains the process value.*

6.8 Documentation

A printout of the logic can be structured and organized in a document object. All POUs are printed out with the documentation object. The hardware is documented separately, but can be inserted into the document object as a file.

Step 1: Insert project master data:

- ☐ Click to the project name in Structure window
- ☐ Open context menu (right mouse click)
- ☐ Click to properties, tab master data

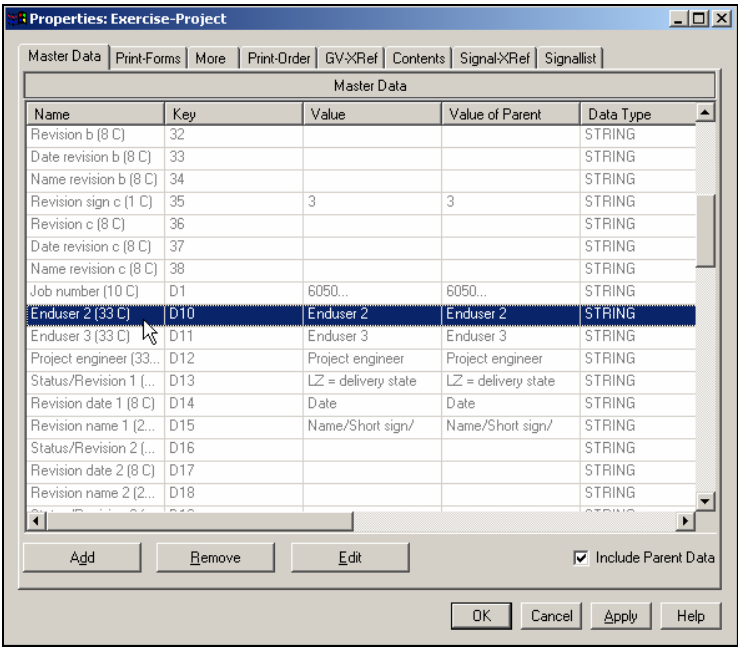


Fig. 108: Master data

Insert important master data like “end user, job number, project engineer.” These information appear on the cover sheet and in the bottom area of each page. (see “Help” for details)

Step 2:

Create a new document object:

- ☐ Click on the project.
- ☐ Open the context menu of the project.
- ☐ Select **New, Documents**.

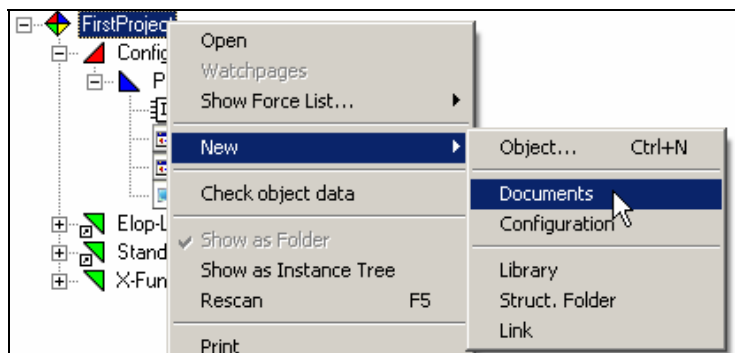


Fig. 109: Create a document object

Step 3:

Rename the new document object:

- ☐ Insert the new name immediately or slowly click twice on the name you want to change. An input field opens and you can enter a new name.

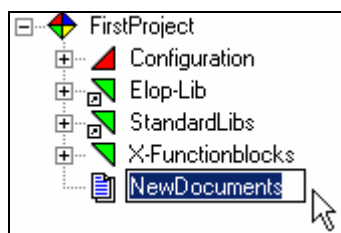


Fig. 110: Rename document object

Step4:

Put all data of your project into the document object:

- ☐ Open the document object by double clicking on the object.
- ☐ Click on your project name and hold down the mouse button.
- ☐ Drag the project onto the document object.

The document object now shows all elements of your project.

Note: You can also add single objects from your project to the documentation by dragging and dropping only one object.

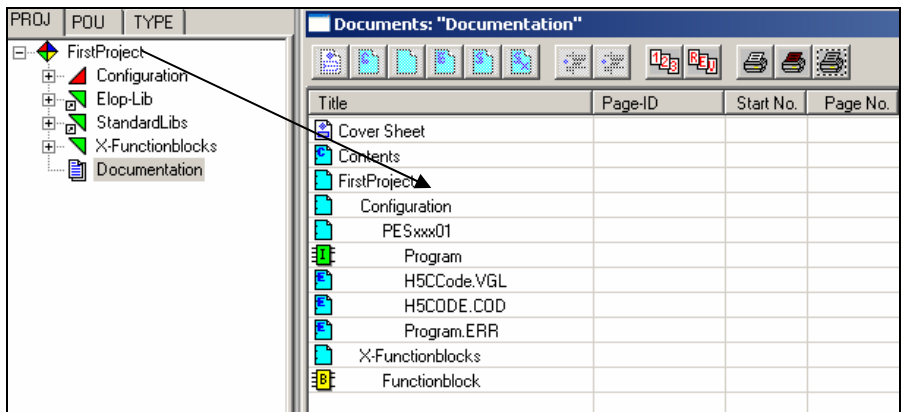


Fig. 111: Add a project to the documentation

Step 5: Update the table of contents:

- Click to the icon „Update Contents“

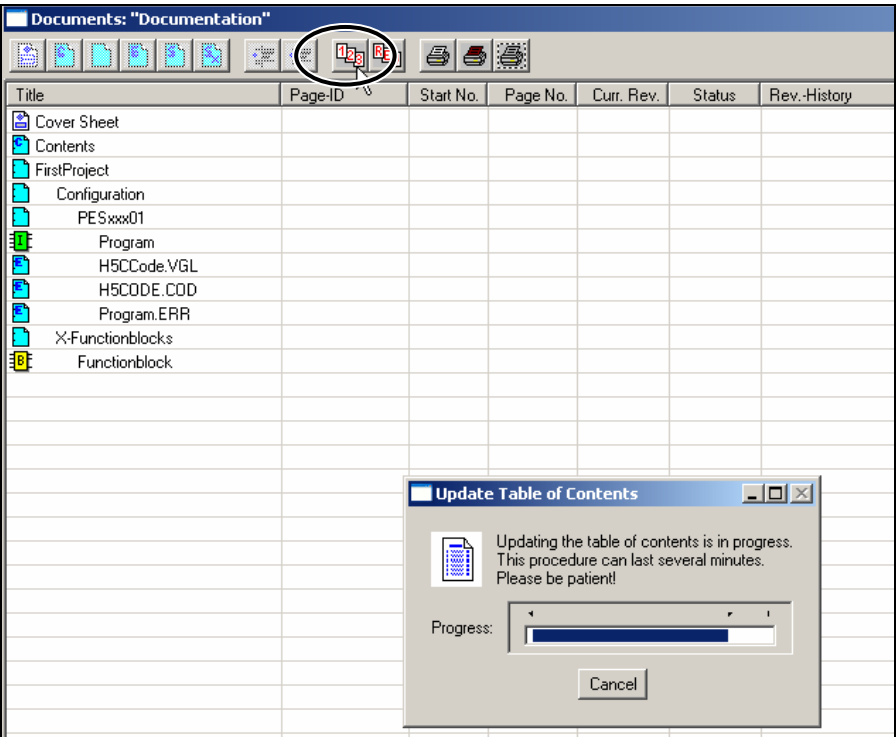


Fig. 112: Update the table of contents

After updating the table of contents, the pages are listed with their pagination.

The sections in the document object correspond to the project folder and the library folders with the indents reflecting the hierarchy in the Structure window.

Note: You can change the order of the elements or delete individual elements. Don't forget to update the table of contents!

Step 6:

Print out the documentation:

- ☐ Start the printout for all, modified or selected pages via the context menu or with the buttons in the symbol bar.

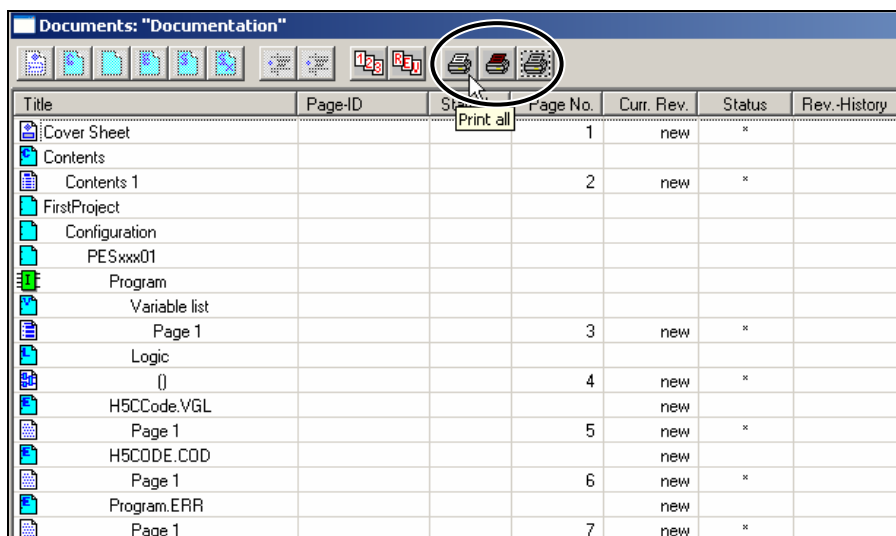


Fig. 113: Start printout

Step 7: Open the resource documentation:

Note: The resource documentation contains all hardware-related data.

- ☐ Click on the resource.
- ☐ Open the context menu of the resource.
- ☐ Select **Documentation, RES-DOCU (generated)**.

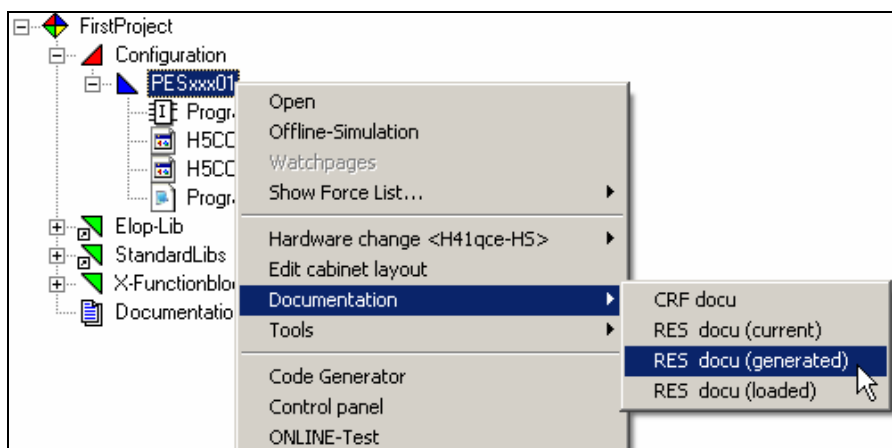


Fig. 114: Open the resource documentation

Step 8:

Start the printout of the resource documentation:

- ☐ Check on screen the information given in the RES docu generated.
- ☐ Print all tabs with the **Print** button.

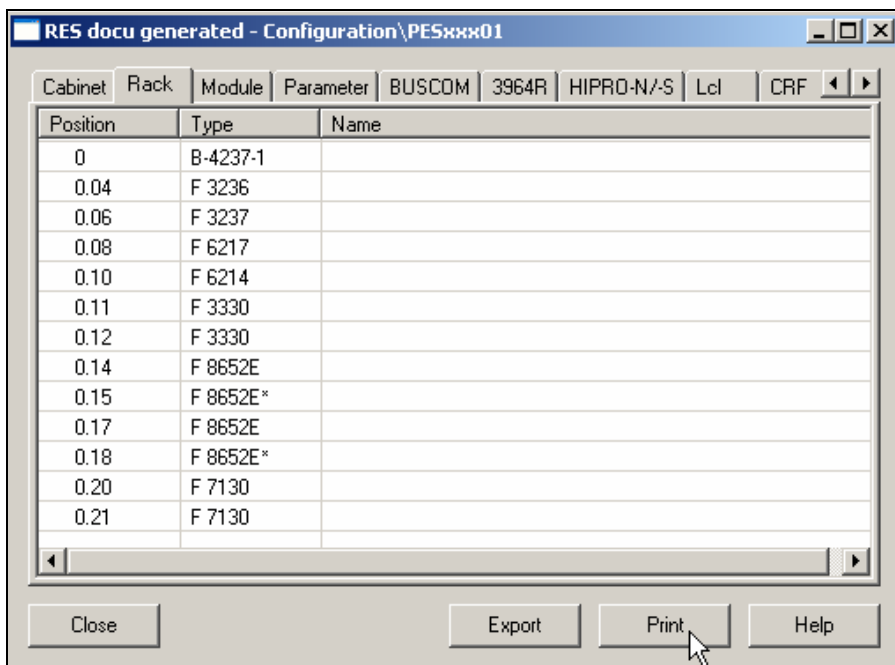


Fig. 115: Resource documentation

*Note: You can also export the resource documentation to a file (in the project path) and include it in the document object.
Attention: All folders will be exported or printed!*

6.9 Backup of a Project

*Note: Please take care you have always a backup of your actual ELOP II project available, in order to protect yourself by lost of data.
In addition the backup procedure is reducing data size (packing).
It is not possible to upload the user program from the controller.
The procedure archive/restore can be used for every object available in the Structure window.*

6.9.1 Archiving

Step 1: Archive the project:

- ☐ Click on the project.
- ☐ Open the context menu of the project.
- ☐ Select **Archive...**

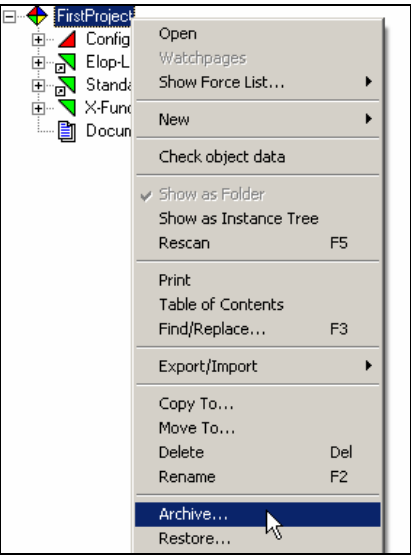


Fig. 116: Archiving a project

Step 2:

Enter the path of the archive:

- ☐ In the “Archive” window (Fig. 117) specify the path where the project should be archived. In the field **Target-file** enter the directory or press the **Browse...** button to select a directory (Fig. 118).
- ☐ In the field **Description** enter a description for the project (optional).
- ☐ Click on the **Backup** button.

The project is archived in the specified directory and now carries the extension ".L3P". The backup consists of three files (Fig. 119).

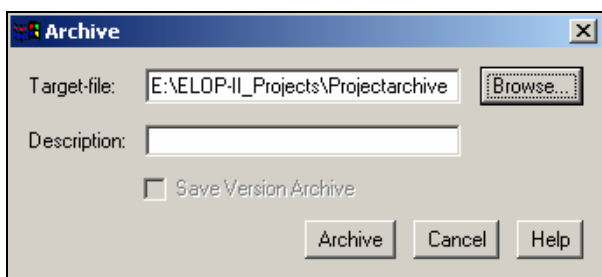


Fig. 117: Define the archive name

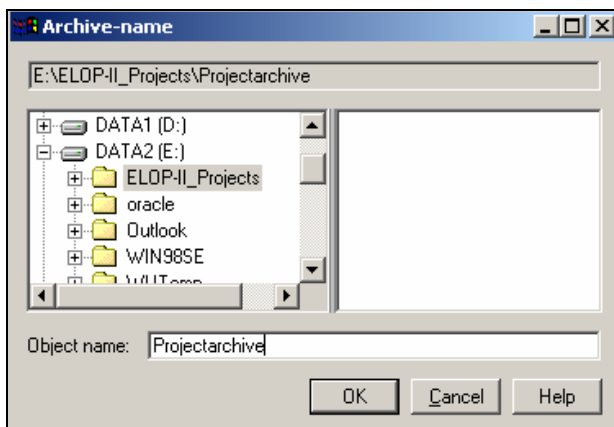


Fig. 118: Browse for the target path

s\Projectarchive.L3P				
	Dateiname ▲	Größe	Typ	Geändert
archive.L3P	archive.aci	1 KB	ACI-Datei	26/09/2005 17:08
	archive.pld	4 KB	PLD-Datei	26/09/2005 17:08
	archive.plp	107 KB	PLP-Datei	26/09/2005 17:08

Fig. 119: Archived project

6.9.2 Restore a Project

Note: To be able to restore a project from an archive, no other project may be opened in ELOP II.

Step 1:

Preceding activities:

- ☐ Close open projects.

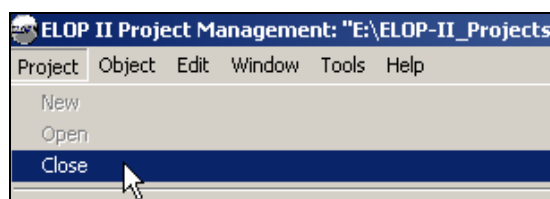


Fig. 120: Close project

- ☐ Start restoration of a project:

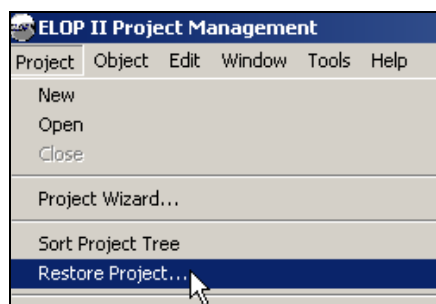


Fig. 121: Restore Project

Step 2:

Select the project to be restored:

- In the “Restore” window (Fig. 122) specify the **Target-directory** where you want to restore the project. Specify also the Archive-name, means your existing backup file. Use the **Browse...** button to select already existing directories.

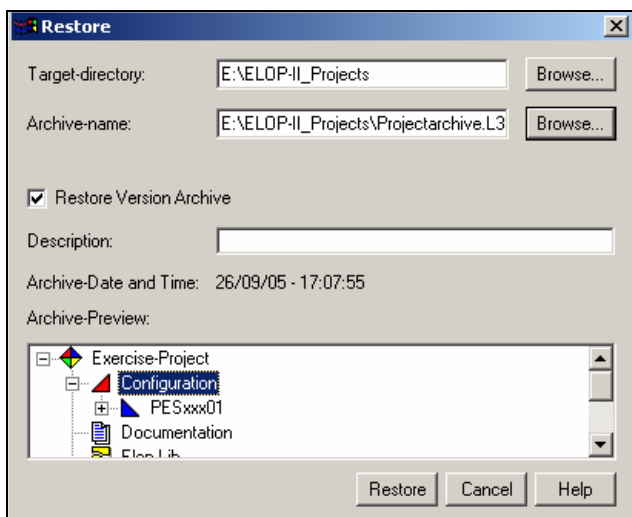


Fig. 122: Restore archive

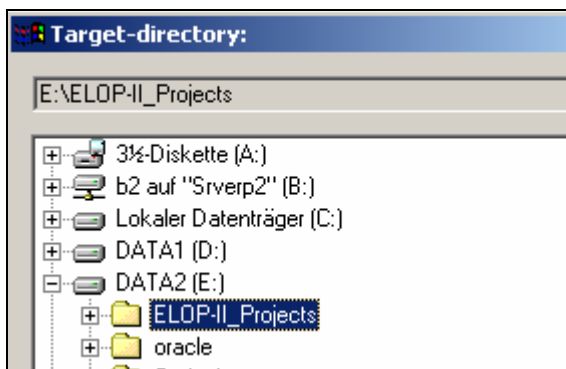


Fig. 123: Select Target-directory

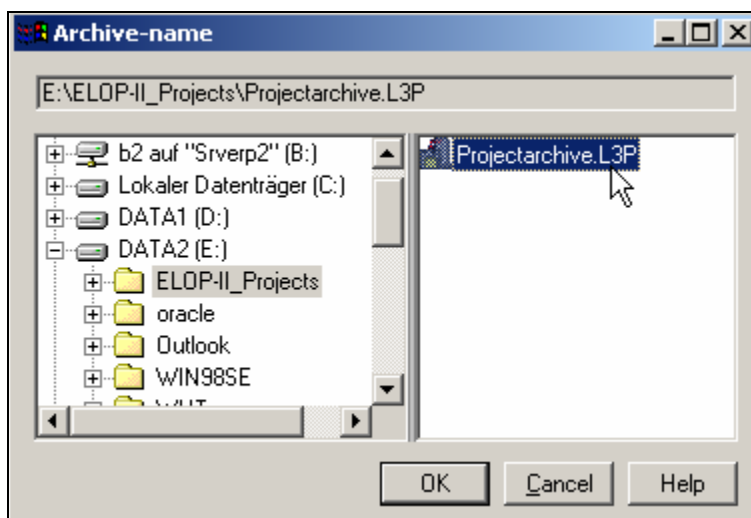


Fig. 124: Select archive

After the project has been successfully restored, it is automatically opened and shown in the Structure window.

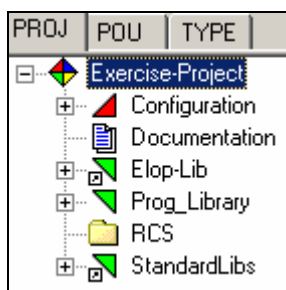


Fig. 125: Restored project

Note: You may also archive individual objects of a project and restore them in other projects.

7 Appendix

7.1 Index

.COD	63
.ERR	63
.L3P.....	99
.VGL.....	63

A

Administrator	
~ rights	5
Adobe Acrobat Reader©	7
Archiving	98

B

Baud rate	69
Bitstr	45
Bus	
~ parameters	69
~ station number	67
~ type	65
add ~	65
add ~ user	66

C

Cabinet layout	57
Code generator	62
~ messages	63
Code version	80, 90
Cold start	78
Communication	64
Add ~ system	65
Computer hardware	6
Configuration	23
new ~	35
Context menu	18
Control Center	11

Control panel.....	73, 74
Coordinates.....	28
columns.....	28
lines.....	28
Copy object.....	47

D

<i>Data type</i>	60
Document object	
Rename ~.....	92
Documentation.....	24, 91
Add project.....	93
new ~ object.....	92
Resource ~.....	96
Dongle.....	1
Download.....	74, 75
Drag & Drop	
~ variables.....	48
Draw segments.....	49, 50
Duplicate.....	47

E

Edit page data.....	46
Error message.....	20
Error state viewer.....	12, 20, 25
Ethernet.....	71
Exercises.....	31

F

Forcing.....	85
Force switch settings.....	88
Force value.....	85
Input variable.....	90
Master force.....	89
Output variable.....	90
Single switch.....	89
Function.....	22
Function block.....	22
create a ~.....	40
Function Block Diagram editor.....	12, 15, 19, 26

G

Grid	47
------------	----

H

Hardlock	1, 5, 6
HIBUS	65

I

Installation	5, 7
De~	9
multiple ~	6
Network ~	9
Interface declaration	
~ editor	19

L

Library	22
new ~	33
rename ~	34
Long name	43

M

Menu bar	12, 13
Modules	
~ assignment	59
analog IO ~	59
digital IO ~	59
insert ~	58
Move page	28

N

Network drive	6
---------------------	---

O

Object	13
Offline simulation	24, 51
start ~	52
Online Help	20
<u>Online test</u>	
~ <u>field</u>	53

Online test (OLT)	80, 81
Overview window	19

P

PADT (PC)	68, 72
Page numbering	28
Parity	69
Plug-Ins	28
Printer	6
Program	
~ instance	22, 23
new ~	37, 38
~ type	22
compile a ~	62
Program Organization Unit (POU)	22
Project	13, 17, 21
~ management	11, 12
~ root	24
create a ~	31

Q

Quick-Info	15
------------------	----

R

Reload	75
Resource	23
~ type	
assigning a ~	56
new ~	36
Restore	100
Revision management	24

S

Sequence Function Chart language	19
Serial	71
Shutdown	75
Slave	67
Standard screen	12
StandardLibs	45
Start menu	11
Starting ELOP II	11

Status bar	12, 16
Stop bits	69
Structure window	12, 17
Objects in the ~	21

T

Target-file	99
Task	24
Technical Support	3
Title bar	12, 13
Toolbar	12, 14

V

Value field	54
~ area	48
Variable	
~ declaration	42
~ editor	19, 42
~ list	48
add ~	48
external ~	22
Local	40
system ~	23
VAR_External	40
VAR_Global	40
VAR_Input	40
VAR_Output	40

W

Working area	12, 19, 25
--------------------	------------

Z

Zoom	27
------------	----

7.2 List of abbreviations

BSN	Bus Subscriber Number
CB	Coprocessor module
CFG	Configuration (bus documentation)
CG	Code Generator
CONST	Constant
CRC	Cyclic redundancy check
CRF	Cross reference (Info on inputs and outputs)
CSV	Data format for import / export function, ASCII format with comma separated value
CU	Central Unit
DXF	Standard-AutoCAD graphic format for printing copy
I/O	Input / output
FB	Function block
FBD	Function Block Diagram
FUN	Function type object
GV	Global variable
HIPRO-N / S	Hibus communication not safety-related / safety-related
HW	Hardware
LgP	Logic plan - controlled protocolling
OLS	Offline simulation
OLT	Online Test
PADT	Programming and debugging Tool (PC)

PES	Programmable Electronic System
PFR	Product Feedback Report
POU	Program Organization Unit (object)
RES	Resource
RETAIN	Retention performance
RT	Resource Type
RWP	Read / write parameter

HIMA
...the safe decision



HIMA Paul Hildebrandt GmbH

Industrial Automation

PO Box 1261 D-68777 Bruehl

Phone: +49 6202 709-0 Fax: +49 6202 709-107

(0638)