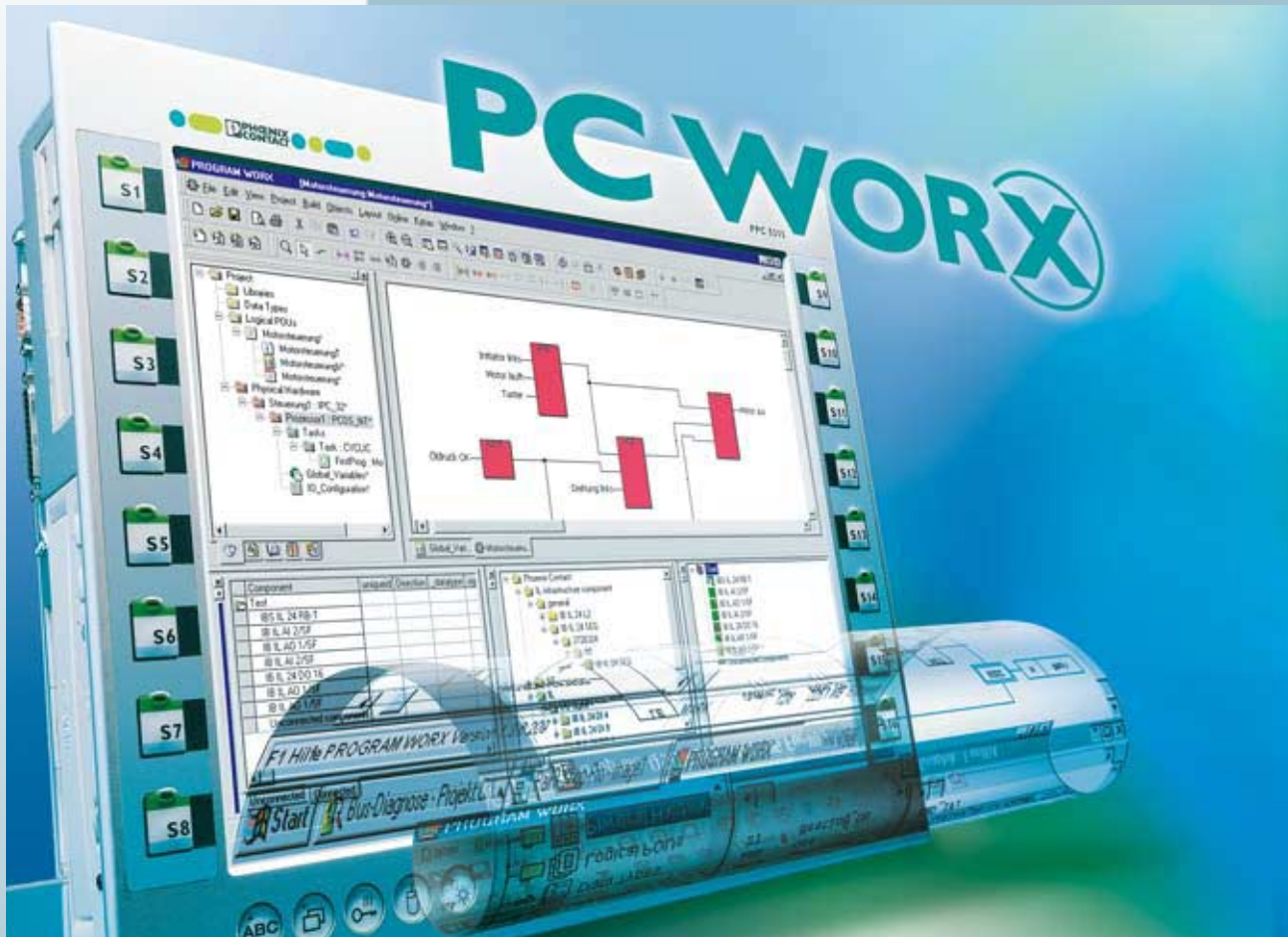


**AUTOMATION**



Quick Start

**UM QS EN PC WORX**

Order No.: —

PC WorX



# AUTOMATION

## Quick Start PC WorX

2011-07-27

---

Designation: UM QS EN PC WORX

Revision: 03

Order No.: —

This user manual is valid for:

Designation

PC WorX

Part of the AUTOMATIONWORX Software Suite 2010

Revision

ab 6.10

ab 1.60

## Please observe the following notes

In order to ensure the safe use of the product described, you have to read and understand this manual. The following notes provide information on how to use this manual.

### User group of this manual

The use of products described in this manual is oriented exclusively to qualified application programmers and software engineers, who are familiar with the safety concepts of automation technology and applicable standards.

Phoenix Contact accepts no liability for erroneous handling or damage to products from Phoenix Contact or third-party products resulting from disregard of information contained in this manual.

### Explanation of symbols used and signal words



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.



#### **DANGER**

This indicates a hazardous situation which, if not avoided, will result in death or serious injury.



#### **WARNING**

This indicates a hazardous situation which, if not avoided, could result in death or serious injury.



#### **CAUTION**

This indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

The following types of messages provide information about possible property damage and general information concerning proper operation and ease-of-use.



#### **NOTE**

This symbol and the accompanying text alerts the reader to a situation which may cause damage or malfunction to the device, either hardware or software, or surrounding property.



This symbol and the accompanying text provides additional information to the reader. It is also used as a reference to other sources of information (manuals, data sheets, literature) on the subject matter, product, etc.

---

### **General terms and conditions of use for technical documentation**

Phoenix Contact reserves the right to alter, correct, and/or improve the technical documentation and the products described in the technical documentation at its own discretion and without giving prior notice, insofar as this is reasonable for the user. The same applies to any technical changes that serve the purpose of technical progress.

The receipt of technical documentation (in particular data sheets, installation instructions, manuals, etc.) does not constitute any further duty on the part of Phoenix Contact to furnish information on alterations to products and/or technical documentation. Any other agreement shall only apply if expressly confirmed in writing by Phoenix Contact. Please note that the supplied documentation is product-specific documentation only and that you are responsible for checking the suitability and intended use of the products in your specific application, in particular with regard to observing the applicable standards and regulations. Although Phoenix Contact makes every effort to ensure that the information content is accurate, up-to-date, and state-of-the-art, technical inaccuracies and/or printing errors in the information cannot be ruled out. Phoenix Contact does not offer any guarantees as to the reliability, accuracy or completeness of the information. All information made available in the technical data is supplied without any accompanying guarantee, whether expressly mentioned, implied or tacitly assumed. This information does not include any guarantees regarding quality, does not describe any fair marketable quality, and does not make any claims as to quality guarantees or guarantees regarding the suitability for a special purpose.

Phoenix Contact accepts no liability or responsibility for errors or omissions in the content of the technical documentation (in particular data sheets, installation instructions, manuals, etc.).

The aforementioned limitations of liability and exemptions from liability do not apply, in so far as liability must be assumed, e.g., according to product liability law, in cases of premeditation, gross negligence, on account of loss of life, physical injury or damage to health or on account of the violation of important contractual obligations. Claims for damages for the violation of important contractual obligations are, however, limited to contract-typical, predictable damages, provided there is no premeditation or gross negligence, or that liability is assumed on account of loss of life, physical injury or damage to health. This ruling does not imply a change in the burden of proof to the detriment of the user.

### Statement of legal authority

This manual, including all illustrations contained herein, is copyright protected. Use of this manual by any third party is forbidden. Reproduction, translation, and public disclosure, as well as electronic and photographic archiving or alteration requires the express written consent of Phoenix Contact. Violators are liable for damages.

Phoenix Contact reserves all rights in the case of patent award or listing of a registered design, in as far as this concerns software of Phoenix Contact that meets the criteria of technicality or has technical relevance. Third-party products are always named without reference to patent rights. The existence of such rights shall not be excluded.

Windows 3.x, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, and Windows Vista are trademarks of the Microsoft Corporation.

All other product names used are trademarks of the respective organizations.

### How to contact us

#### Internet

Up-to-date information on Phoenix Contact products and our Terms and Conditions can be found on the Internet at:

[www.phoenixcontact.com](http://www.phoenixcontact.com).

Make sure you always use the latest documentation.

It can be downloaded at:

[www.phoenixcontact.net/catalog](http://www.phoenixcontact.net/catalog).

#### Subsidiaries

If there are any problems that cannot be solved using the documentation, please contact your Phoenix Contact subsidiary.

Subsidiary contact information is available at [www.phoenixcontact.com](http://www.phoenixcontact.com).

#### Published by

PHOENIX CONTACT GmbH & Co. KG  
Flachmarktstraße 8  
32825 Blomberg  
Germany  
Phone +49 - (0) 52 35 - 3-00  
Fax +49 - (0) 52 35 - 3-4 12 00

PHOENIX CONTACT  
P.O. Box 4100  
Harrisburg, PA 17111-0100  
USA  
Phone +1-717-944-1300

Should you have any suggestions or recommendations for improvement of the contents and layout of our manuals, please send your comments to

[tecdoc@phoenixcontact.com](mailto:tecdoc@phoenixcontact.com).

# Table of contents

1	General.....	1-1
1.1	Introduction.....	1-1
1.2	Information about this manual.....	1-1
1.3	System requirements.....	1-2
1.4	Ordering data .....	1-2
2	Installing and enabling the software.....	2-1
2.1	Prior to installation .....	2-1
2.2	AUTOMATIONWORX Software Suite .....	2-1
2.3	Starting the installation program .....	2-2
2.4	Starting PC WorX .....	2-2
2.5	Enabling the PC WorX license.....	2-3
2.6	License Manager.....	2-4
3	Helpful information about PC WorX .....	3-1
3.1	Online help .....	3-1
3.2	Selecting the language .....	3-1
3.3	The PC WorX user interface.....	3-3
3.4	Toolbars .....	3-4
3.5	Workspaces .....	3-5
3.5.1	Windows in the workspaces .....	3-8
3.5.2	Toggling windows on/off and docking/undocking windows .....	3-9
3.5.3	Auto-hide function .....	3-9
3.6	"Bus Structure" window .....	3-10
3.6.1	Icons in the "Bus Structure" window .....	3-10
3.6.2	Display in the "Bus Structure" window .....	3-12
3.7	"EXCEL Link" window.....	3-14
3.8	"Diag+" window .....	3-15
3.9	Visualization .....	3-19
3.10	FDT (Field Device Tool) workspace .....	3-20
3.10.1	Introduction .....	3-20
3.10.2	General .....	3-20
3.10.3	Installing DTM libraries .....	3-21
3.10.4	Creating a project and integrating DTMs in PC WorX .....	3-24
3.10.5	Reading in the bus configuration and/or manually inserting devices .....	3-25
3.11	Calling DTM functions .....	3-27
3.11.1	General DTM functions .....	3-27
3.11.2	DTM functions of the controller .....	3-28
3.11.3	DTM functions of devices .....	3-30

4	Sequence for creating a project .....	4-1
4.1	Sequence for creating a project.....	4-2
4.2	Creating a new project.....	4-3
4.3	Specifying project information .....	4-4
4.4	Preparing the PC for communication .....	4-6
4.5	Checking/modifying IP settings for the controller.....	4-7
4.6	Decision: Working online or offline .....	4-8
4.7	Inserting PROFINET IO devices .....	4-8
4.8	Checking/modifying the PROFINET settings for PROFINET IO devices .....	4-10
4.9	Manually inserting INTERBUS devices.....	4-12
4.9.1	Manually inserting INTERBUS devices - General .....	4-12
4.9.2	Inserting INTERBUS devices below an INTERBUS proxy .....	4-15
4.9.3	Inserting Inline terminals below a PROFINET IO bus coupler .....	4-15
4.9.4	INTERBUS devices below the PROFINET IO controller .....	4-15
4.10	Compiling after completing the bus topology .....	4-16
4.11	Creating the program.....	4-17
4.12	Compiling after creating the program.....	4-17
4.13	Generating variables and assigning process data .....	4-18
4.13.1	Generating variables .....	4-18
4.13.2	Assigning process data .....	4-19
4.14	Setting the communication path: EasySim simulation .....	4-22
4.15	Switching to working with the system (online).....	4-23
4.16	Assigning the IP address for the controller .....	4-23
4.16.1	Address assignment via the PC WorX BootP server .....	4-23
4.16.2	Address assignment with PC WorX via the serial interface .....	4-25
4.17	Setting the communication path .....	4-26
4.17.1	Communication via the serial interface .....	4-27
4.17.2	Communication via Ethernet .....	4-28
4.18	Reading in and importing PROFINET IO devices .....	4-29
4.18.1	Reading in PROFINET IO devices .....	4-29
4.18.2	Importing PROFINET IO devices into the project .....	4-31
4.18.3	Naming PROFINET IO devices without a PROFINET device name ..	4-31
4.18.4	Bus configuration with read in PROFINET IO devices .....	4-32
4.18.5	Subsequent naming of a PROFINET IO device .....	4-33
4.19	Checking/modifying the PROFINET settings for PROFINET IO devices .....	4-34
4.20	Transferring PROFINET device names and IP settings to PROFINET IO devices.....	4-35
4.21	Reading in INTERBUS .....	4-36
4.22	Compiling after reading in the bus topology.....	4-39
4.23	Creating the program.....	4-39



---

	4.24 Compiling after creating the program.....	4-39
	4.25 Assigning process data .....	4-39
	4.26 Compiling and sending a project, and performing a cold restart .....	4-40
	4.26.1 Compiling a project .....	4-40
	4.26.2 Sending a project .....	4-40
	4.26.3 Performing a cold restart .....	4-43
	4.27 Operation.....	4-45
	4.27.1 Setting the task properties .....	4-45
	4.27.2 Debug mode .....	4-47
	4.27.3 PLC stop/run .....	4-48
	4.28 Switching from simulation to real hardware .....	4-49
<b>5</b>	<b>Example project for an INTERBUS system.....</b>	<b>5-1</b>
	5.1 Project description.....	5-1
	5.2 Sequence for creating the INTERBUS project.....	5-2
	5.3 Creating a new project.....	5-4
	5.4 Specifying project information .....	5-4
	5.5 Checking/modifying IP settings for the controller.....	5-4
	5.6 Assigning the IP address for the controller .....	5-5
	5.7 Setting the communication path .....	5-5
	5.8 Reading in INTERBUS .....	5-6
	5.9 Compiling after completing the bus topology.....	5-6
	5.10 Creating the program.....	5-6
	5.11 Compiling after completing the program.....	5-7
	5.12 Assigning process data .....	5-7
	5.13 Compiling and sending a project, and performing a cold restart.....	5-8
	5.14 Operation.....	5-8
<b>6</b>	<b>Example project for a system consisting of PROFINET IO and INTERBUS.....</b>	<b>6-1</b>
	6.1 Project description.....	6-1
	6.2 Sequence for creating the PROFINET project.....	6-3
	6.3 Creating a new project.....	6-5
	6.4 Specifying project information .....	6-5
	6.5 Preparing the PC for communication .....	6-5
	6.6 Checking/modifying IP settings for the PROFINET IO controller.....	6-6
	6.7 Assigning the IP address for the PROFINET IOcontroller.....	6-7
	6.8 Setting the communication path .....	6-7
	6.9 Reading in and importing PROFINET IO devices .....	6-8
	6.10 Checking/modifying the PROFINET settings for PROFINET IO devices .....	6-9

6.11	Transferring PROFINET device names and IP settings to PROFINET IO devices .....	6-10
6.12	Reading in INTERBUS .....	6-11
6.13	Compiling after completing the bus topology .....	6-12
6.14	Creating the program.....	6-12
6.15	Compiling after creating the program.....	6-12
6.16	Assigning process data .....	6-13
6.17	Compiling and sending a project, and performing a cold restart .....	6-14
6.18	Operation.....	6-15
6.19	Additional information.....	6-15
6.19.1	Changing the PROFINET device name .....	6-15
6.19.2	Renumbering devices .....	6-18
6.19.3	Assigning IP parameters for the PROFINET IO device .....	6-19
<b>7</b>	<b>Example project for a simulation with processor type "IPC" controllers.....</b>	<b>7-1</b>
7.1	Project description.....	7-1
7.2	Sequence for creating the simulation project.....	7-2
7.3	Creating a new project.....	7-4
7.4	Specifying project information .....	7-4
7.5	Checking/modifying IP settings for the controller .....	7-4
7.6	Manually inserting INTERBUS devices.....	7-5
7.7	Compiling after completing the bus topology .....	7-5
7.8	Creating the program.....	7-6
7.9	Compiling after creating the program.....	7-6
7.10	Assigning process data .....	7-6
7.11	Setting the communication path .....	7-8
7.12	Compiling and sending a project, and performing a cold restart .....	7-8
7.13	Operation.....	7-10
7.13.1	Setting the task properties .....	7-10
7.13.2	Simulation and debug mode .....	7-12
7.14	Switching from simulation to real hardware .....	7-16
<b>8</b>	<b>Example program .....</b>	<b>8-1</b>
8.1	Program description .....	8-1
8.2	Function blocks used.....	8-2
8.3	Programming.....	8-4
8.4	Setting the initial value.....	8-9
8.5	Additional options for PROFINET .....	8-9

<b>A</b>	<b>Additional software functions .....</b>	<b>A-1</b>
A 1	Setting the realtime clock.....	A-1
A 2	Options for modifying a project.....	A-2
A 2.1	Download Changes .....	A-2
A 2.2	Online modifications .....	A-5
A 3	Extended retain handling.....	A-8
A 4	Simulation.....	A-9
A 5	Comparing projects .....	A-9
A 6	Visualization .....	A-9
A 7	Replacing a controller (hardware replacement).....	A-9
A 8	Device description files.....	A-16
A 8.1	Phoenix Contact device description files .....	A-16
A 8.2	Device description files for Inline Controllers .....	A-17
A 8.3	Device description files from other manufacturers (GSD files) .....	A-18
A 8.4	Creating device description files .....	A-18
A 9	Ethernet topology .....	A-19
A 10	INTERBUS topology.....	A-20
<b>B</b>	<b>Status information for a PROFINET IO system .....</b>	<b>B-1</b>
B 1	Status of the PROFINET IO controller .....	B-1
B 2	Status of a PROFINET IO device.....	B-2
B 3	INTERBUS registers.....	B-3
B 4	Setting all PROFINET IO devices to a defined state.....	B-4
<b>C</b>	<b>AX OPC Server and WebVisit.....</b>	<b>C-1</b>
C 1	AX OPC Server.....	C-1
C 1.1	Preparatory tasks in PC WorX .....	C-1
C 1.2	OPC Configurator .....	C-2
C 1.3	OPC Test Client .....	C-4
C 2	WebVisit .....	C-6
<b>D</b>	<b>Phoenix Contact controllers.....</b>	<b>D-1</b>



# 1 General

## 1.1 Introduction

PC WorX is the automation software, which combines programming according to IEC 61131, fieldbus configuration, and diagnostics.

The programming system is based on modern 32-bit Windows technology and enables easy handling for the user by means of zooming, drag & drop, and dockable windows. IEC configuration elements can be processed and libraries can be integrated. In addition, the programming system has a powerful debugging system. In PC WorX, all functions can be easily accessed via the menu and you can create a project using only a few dialog boxes. You can then immediately start developing your program.

## 1.2 Information about this manual

This document helps you to parameterize a bus configuration and to program the application program (according to IEC 61131-3) using example projects.

It is assumed the user has knowledge of and experience in the operation of PCs and Windows operating systems, and knowledge of IEC 61131 and Ethernet basics.



More detailed information about the individual functions of PC WorX can be found in the online help for the program. The entire help function can be called via "Help" in the menu bar. Help for specific functions can be called via F1.



Functions and commands requiring communication with the controller cannot be executed without a physical bus configuration. However, complete parameterization is possible in the "offline" state. The application program can also be created and compiled.

## 1.3 System requirements

PC system, requirements	
Supported operating systems	MS Windows® XP Professional SP3 MS Windows® Vista Business MS Windows® 7 Professional (32-bit; 64-bit as of AUTOMATIONWORX Software Suite 1.60 SP1)
Hardware requirements	
CPU	Pentium 4/Celeron 1.6 GHz (minimum), 2 GHz (recommended)
Main memory	1 GB (minimum), 2 GB (recommended)
Hard disk space	2 GB free memory space
DVD-ROM drive	Yes
Interfaces	1 x Serial (V.24 (RS-232)), 1 x Ethernet (TCP/IP)
Monitor	XGA, resolution of 1024 x 768 pixels (minimum); SXGA, resolution of 1280 x 1024 pixels (recommended)
Operating devices	Keyboard, mouse
Web browser	Internet Explorer Version 7 or later

## 1.4 Ordering data

### Products

Description	Type	Order No.	Pcs. / Pkt.
PC WORX DEMO license (software DVD), contains all 5 IEC languages, with MSFC compiler, maximum of 16 bytes of I/O data (mix)	PC WORX DEMO	2985725	1
PC WORX BASIC license, contains all 5 IEC languages, without MSFC compiler, maximum of 256 bytes of I/O data (mix), version-specific license key	PC WORX BASIC LIC	2985275	1
PC WORX PRO license, contains all 5 IEC languages, with MSFC compiler, maximum of 128 kbytes of I/O data (mix), version-specific license key	PC WORX PRO LIC	2985385	1
PC WORX PRO license, contains all 5 IEC languages, with MSFC compiler, maximum of 128 kbytes of I/O data (mix), version-specific license key, upgrade of an existing basic license to a full license with MSFC compiler	PC WORX BASIC-PRO LIC	2985259	1
PC WORX BASIC update, version update for PC WORX BASIC LIC (e.g., from 5.xx to 6.xx), maximum of 256 bytes of I/O data (mix), version-specific license key	PC WORX BASIC UPD	2985262	1
PC WORX PRO update, version update for PC WORX PRO LIC (e.g., from 5.xx to 6.xx), maximum of 128 kbytes of I/O data (mix), version-specific license key	PC WORX PRO UPD	2985372	1



All PC WorX versions support all five IEC programming languages according to IEC 61131-3.

## 2 Installing and enabling the software

### 2.1 Prior to installation



Prior to installation, close all open Windows applications.  
This PC WorX version can be installed parallel to an existing earlier version.

### 2.2 AUTOMATIONWORX Software Suite

PC WorX is part of the AUTOMATIONWORX Software Suite.

The AUTOMATIONWORX Software Suite includes the following programs:

- **Config+**  
Easy configuration and startup of INTERBUS networks
- **Diag+**  
User-friendly network diagnostics during startup and operation
- **Diag+ NetScan**  
User-friendly monitoring of multiple INTERBUS networks
- **PC WorX**  
Uniform IEC 61131 programming environment for all Phoenix Contact controllers
- **PC WorX Express**  
Easy-to-use version of the PC WorX software tool designed to provide a more friendly introduction to the world of IEC 61131 programming using controller class 100 (ILC 1xx) from Phoenix Contact
- **AX OPC Server**  
Software used for data exchange between distributed INTERBUS networks and visualization systems
- **WebVisit**  
Tool for creating web pages for web-based operator panels from Phoenix Contact

The desired programs can be selected individually or simultaneously for installation.

Diag+ should only be selected if you wish to use it independently of PC WorX. When installing PC WorX, Diag+ is installed as part of PC WorX.

When one of the software suite programs is started for the first time, it runs in demo mode with limited resources. A registration code is required to enable the full version. You will receive the registration code when a full version of the relevant program is purchased.

### 2.3 Starting the installation program

- Insert the "AUTOMATIONWORX Software Suite" DVD in your DVD-ROM drive. The installation program usually starts automatically after a few seconds.
- If it does not, start the "SETUP.EXE" file from the "[Drive]:\SETUP\" directory on the DVD-ROM. This file calls the installation wizard, which guides you through the installation process.
- Follow the instructions in the installation program.

The installation program generates all the directories required for operation and copies the files for the selected programs.



#### Restart the PC

Following successful installation, you must restart your computer for the changes to the configuration files to take effect. To do this, click "Finish" at the end of the installation process.

- To restart the PC, click "Finish" at the end of the installation process.

### 2.4 Starting PC WorX

- For installation using the default settings, start PC WorX via "Start, All Programs, PHOENIX CONTACT, AUTOMATIONWORX Software Suite 201x 1.6x, PC WORX 6.1x".

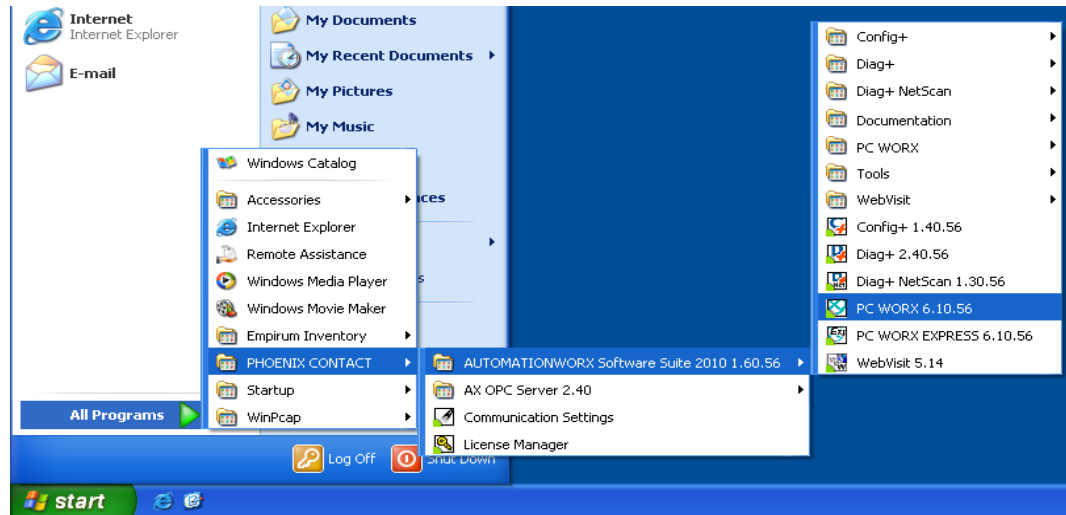


Figure 2-1 Starting PC WorX





When PC WorX is started for the first time, it runs in demo mode with limited resources. A maximum of 16 bytes is available for I/O data.

If the upper limits for demo mode are exceeded, the following error messages are generated **when compiling**:

- Too many inputs/outputs in I/O configuration (16 maximum, XXXXX specified)! PC WORX running with limited resources.
- Resource file cannot be compiled!

Regardless of the bus configuration, these error messages always appear when compiling for various controllers (e.g., RFC 470S PN 3TX) in demo mode as all system variables are entered in the I/O configuration.

If you wish to switch from demo mode to the full version, enable your PC WorX license.

## 2.5 Enabling the PC WorX license

After starting PC WorX, proceed as follows to enable your license:

- Select the "Register..." command in the "?" menu.

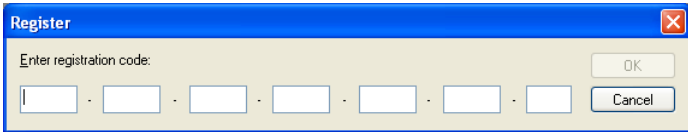


Figure 2-2 Registration dialog box for entering the license code

- Enter your registration code.
- Confirm your entry with "OK".



Figure 2-3 Registration dialog box for entering the license code



Alternatively, you can also license PC WorX via the License Manager in the AUTOMATIONWORX Software Suite.

Registration comes into effect the next time PC WorX is started.

## 2.6 License Manager

The License Manager can be used to enable all programs in the AUTOMATIONWORX Software Suite. Select the relevant program to be licensed and then enter the corresponding license key in the field provided.

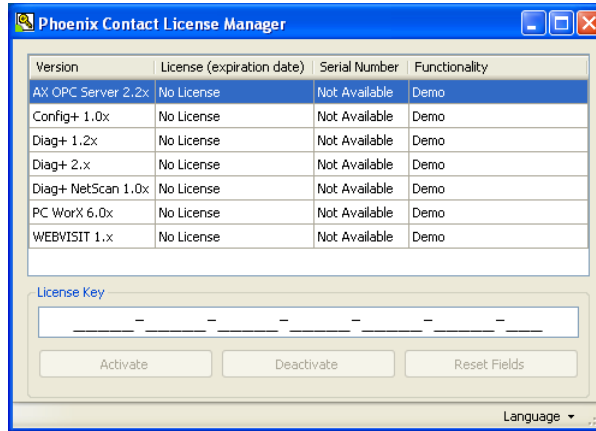


Figure 2-4 License Manager

## 3 Helpful information about PC WorX

### 3.1 Online help

More detailed information about the individual functions of PC WorX can be found in the online help for the program. The entire help function can be called via "Help" in the menu bar. Help for specific functions can be called via F1.

### 3.2 Selecting the language

When installing the software, the language in which PC WorX should be started can be selected. The program language can be changed at any time.

- Select the "Extras, Options..." menu.
- Select the "General" tab.
- Select the language.
- Confirm your selection with "Apply" and "OK".

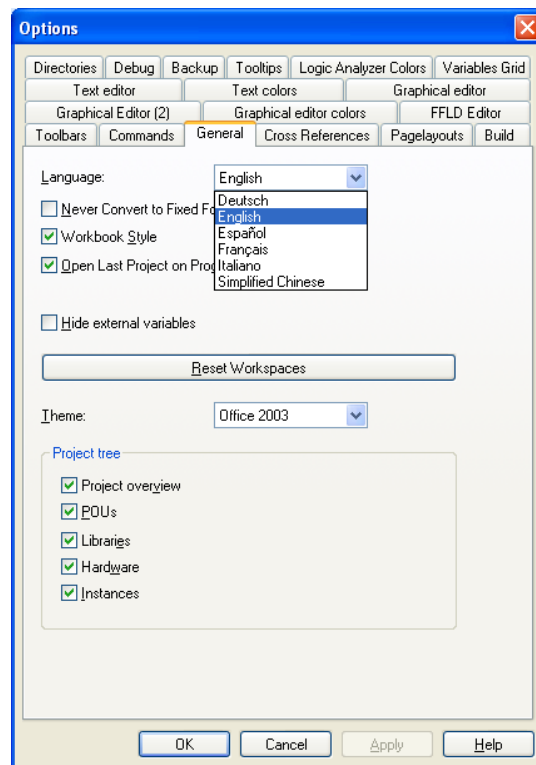


Figure 3-1 Changing the language setting



The selected language is activated the next time PC WorX is started.



The "Theme" element can be used to change the design of the PC WorX user interface. Default setting: "Office 2003".

- Close and restart the program.

**Exporting texts/importing translations**

Texts previously created in one language can be exported and translated. The translated texts can then be imported back into PC WorX.

Please ensure that the project language has been set:

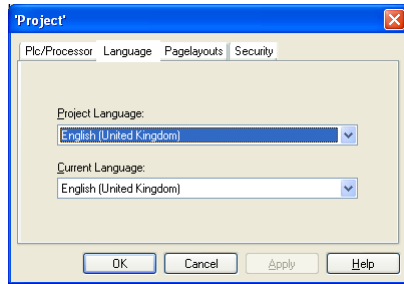


Figure 3-2 Project language

- Open the "File, Export..." menu.
- Select the data you wish to export.
- Confirm your selection with "OK".

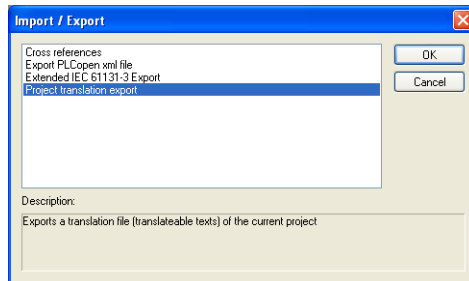


Figure 3-3 Exporting files from the project

Once the files have been translated:

- Open the "File, Import..." menu.
- Select the data you wish to import.
- Confirm your selection with "OK".
- Specify the path for the data to be imported.
- Confirm your selection with "Import".

### 3.3 The PC WorX user interface

The user interface consists of the following main components: menu bar, toolbars, main window, and status bar. The contents of the main window depend on the workspace.

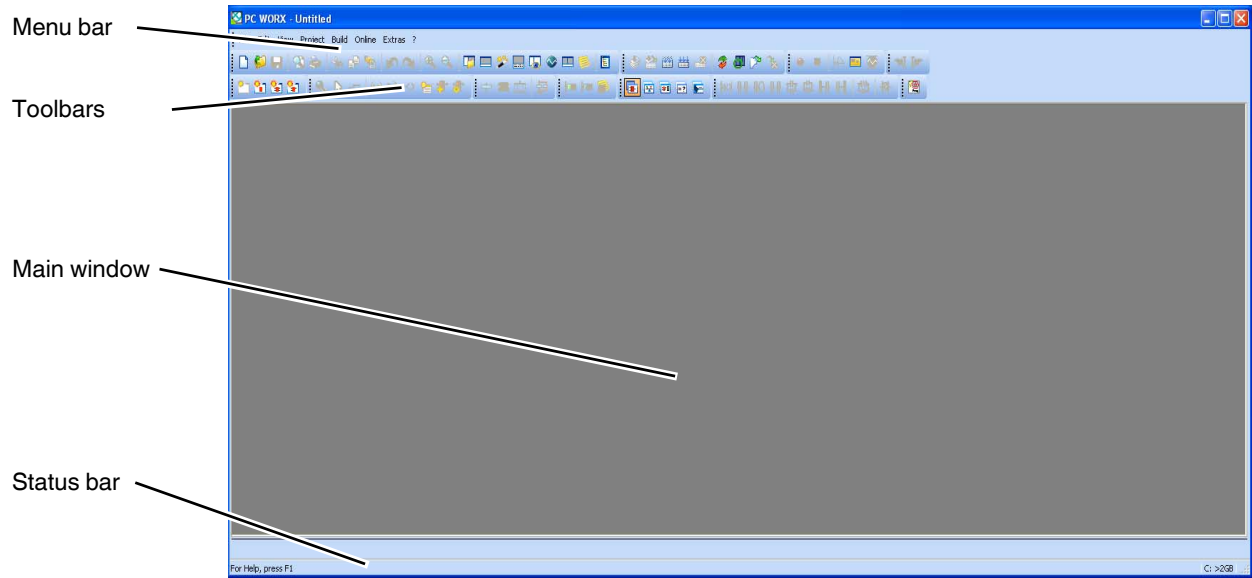


Figure 3-4 User interface

## 3.4 Toolbars

The program contains several toolbars with different icons, which enable frequently used operations to be executed quickly. Alternatively, these operating steps can be called via menu items or predefined shortcuts.

By default, all the toolbars are shown. To display or hide a specific toolbar, use the "Extras, Options" dialog box.

When the mouse pointer is placed over an icon (without clicking on it), a tool tip appears. The tool tip displays the name of the current icon. In addition, a short function description appears in the status bar. If tool tips are not displayed, this feature can be activated in the "Extras, Options, Toolbars" dialog box.

### Icons for selecting the workspace

The workspace can be changed via the icons in the toolbar:



Activate IEC programming workspace.



Activate bus configuration workspace.



Activate process data assignment workspace.



Activate project comparison workspace.



Activate FDT (Field Device Tool) workspace.



Which windows will actually be displayed depends on which windows have been toggled on (see also "Windows in the workspaces" on page 3-8). The last setting for each workspace is saved when the program is closed and restored when it is started again.

### Frequently used icons for compiling and debugging



Online modifications.



Make (compile project; corresponds to "Build, Make" in the menu bar).



Rebuild project (corresponds to "Build, Rebuild Project" in the menu bar).



Switch debug mode on/off.



Display project control dialog box.

## 3.5 Workspaces

PC WorX is divided into five workspaces:

- IEC programming
- Bus configuration
- Process data assignment
- Project comparison
- FDT (Field Device Tool)

The "View" menu or the corresponding icon in the toolbar can be used to switch between the workspaces. Following initial installation, the IEC programming workspace is the default setting.

Figure 3-5 to Figure 3-9 below show the default workspaces. Table 3-1 on page 3-8 provides an overview of the windows that can be usefully added to the default setting.

### IEC programming workspace

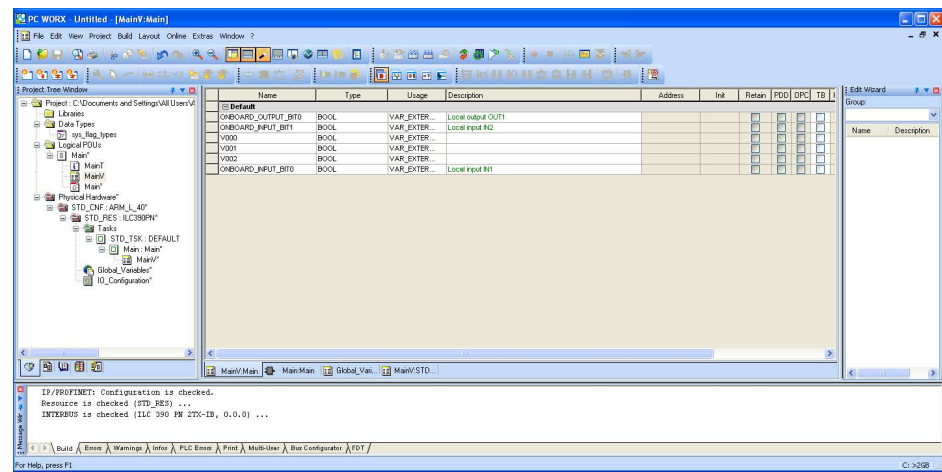


Figure 3-5 IEC programming workspace

### Bus configuration workspace

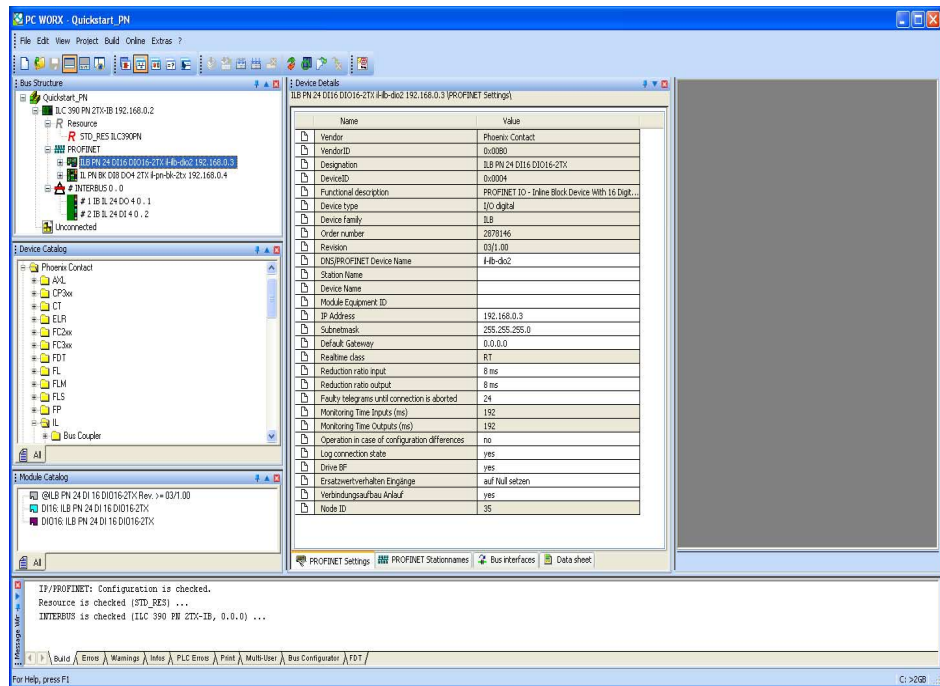


Figure 3-6 Bus configuration workspace

### Process data assignment workspace

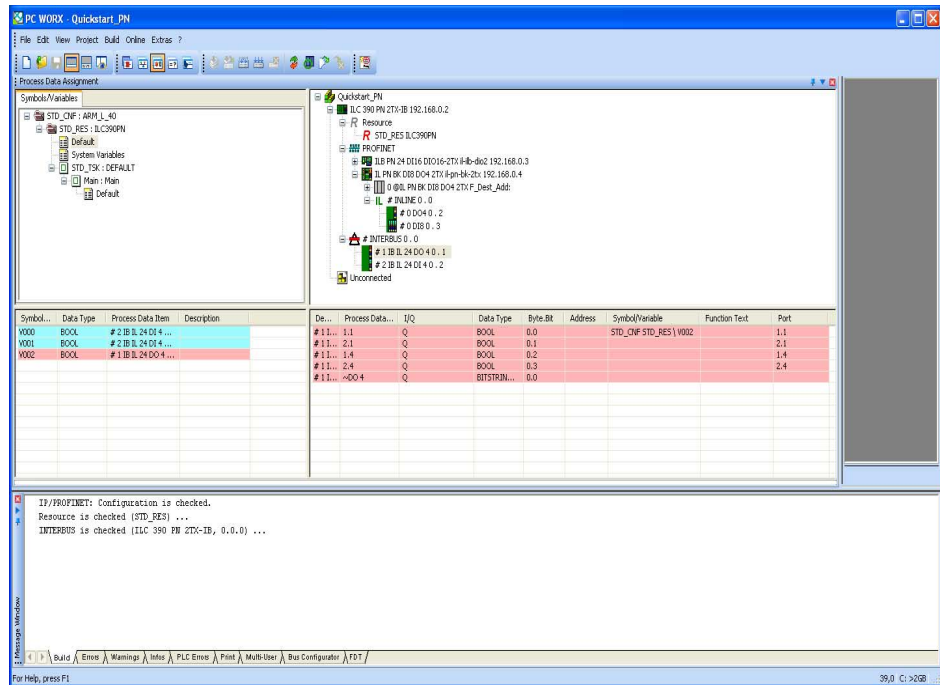


Figure 3-7 Process data assignment workspace



### Project comparison workspace

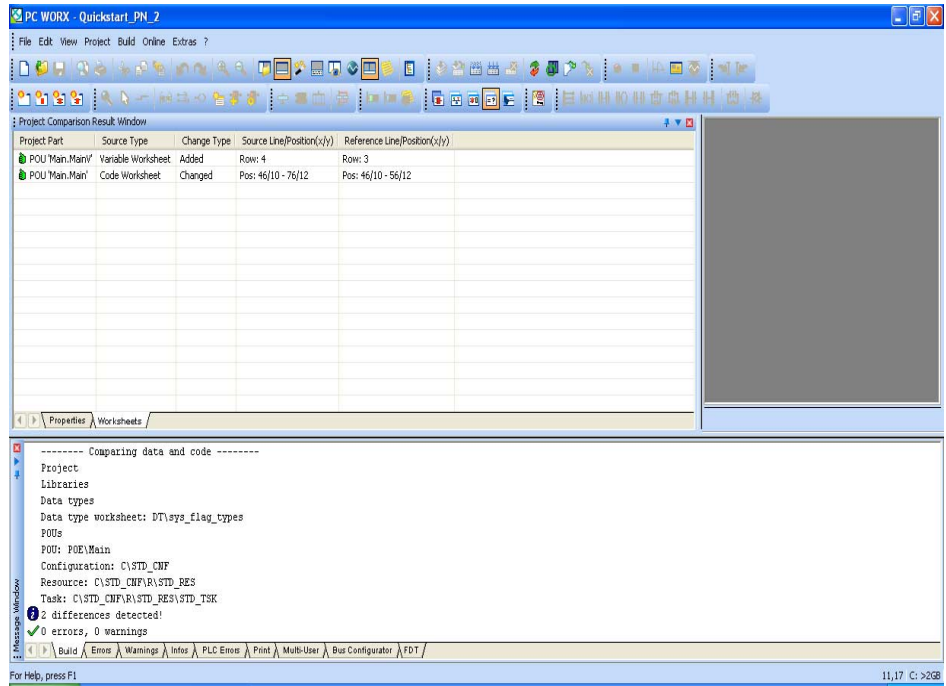


Figure 3-8 Project comparison workspace

### FDT workspace

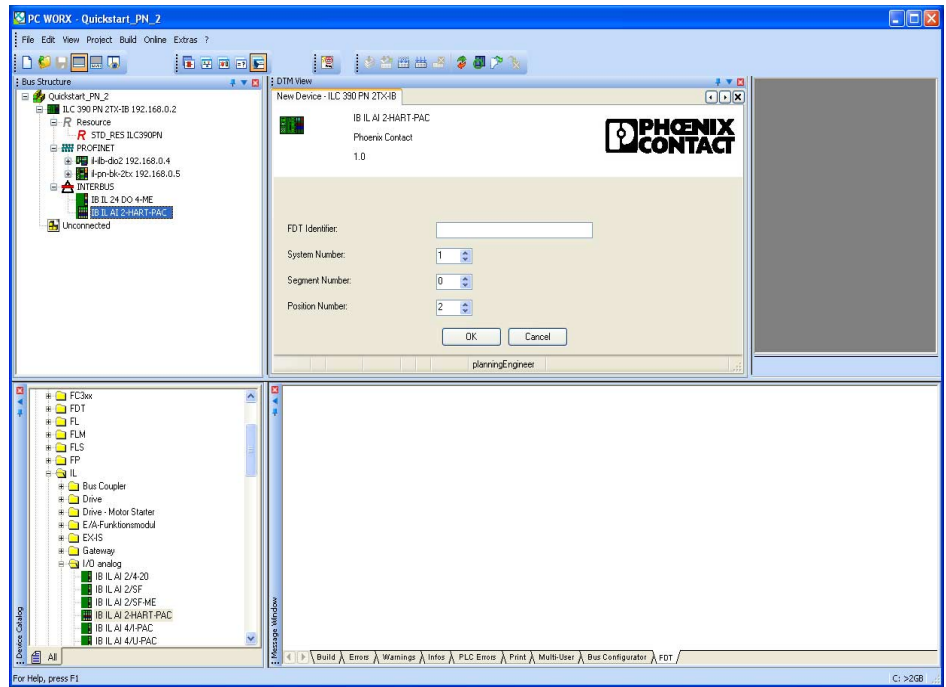


Figure 3-9 FDT workspace

### 3.5.1 Windows in the workspaces

The windows that you wish to display can be defined at any time for each workspace. Table 3-1 lists which windows are set by default when PC WorX is started for the first time and which windows can be usefully added.

Table 3-1 Windows in the workspaces

Window	IEC programming		Bus configuration		Process data assignment		Project comparison		FDT	
	D	A	D	A	D	A	D	A	D	A
Project Tree Window	Yes	Yes	-	-	-	-	-	-	-	-
Message Window	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Edit Wizard	Yes	Yes	-	-	-	-	-	-	-	-
Cross References Window	-	Yes	-	-	-	-	-	-	-	-
Watch Window	-	Yes	-	-	-	-	-	-	-	-
Logic Analyzer	-	Yes	-	-	-	-	-	-	-	-
Project Comparison Result Window	-	-	-	-	-	-	Yes	Yes	-	-
Status bar	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Bus Structure	-	-	Yes	Yes	-	-	-	-	Yes	Yes
Device Details	-	-	Yes	Yes	-	-	-	-	-	Yes
Device Catalog	-	-	Yes	Yes <sup>1</sup>	-	-	-	-	-	Yes
Module Catalog	-	-	-	Yes <sup>2</sup>	-	-	-	-	-	-
Process Data Assignment	-	-	-	-	Yes	Yes	-	-	-	-
Diag+	-	-	-	Yes <sup>3</sup>	-	-	-	-	-	-
Connected Bus	-	-	-	Yes	-	-	-	-	-	-
EXCEL Link	-	-	-	Yes	-	Yes <sup>4</sup>	-	-	-	-
LCD	-	Yes	-	Yes	-	Yes	-	-	-	-
Ethernet Topology	-	-	-	Yes	-	-	-	-	-	-
INTERBUS Topology	-	-	-	Yes	-	-	-	-	-	-
DTM View	-	-	-	-	-	-	-	-	Yes	Yes

Key:

- 1 When inserted manually
- 2 For PROFINET
- 3 For troubleshooting
- 4 When using Excel
- D Default
- A Useful addition

### 3.5.2 Toggling windows on/off and docking/undocking windows

#### Toggling on/off

Each window can be toggled on/off via the "View" menu by selecting the corresponding menu item.

For the windows recommended for the relevant workspace, please refer to "Workspaces" on page 3-5.

#### Docking/undocking

For each window, you can specify whether or not it is to be docked in the other windows. There are various options for docking/undocking windows:

1. Permanent docking/undocking:  
In the title bar or the blue/gray frame of the relevant window, right-click with the mouse and activate/deactivate the "Allow Docking" menu item.  
Move an undocked window to the position where it is to be inserted in the desktop.
2. Temporary undocking:  
Double-click on the gray window frame or the title bar of the window. The window is then displayed as a "normal" window. Its size can be modified and it can be moved to any position on the screen. In order to re-dock the window, i.e., to reinsert it in the desktop, double-click on the title bar.

### 3.5.3 Auto-hide function

The auto-hide function can be toggled on/off for each window. This function ensures that the window is automatically displayed or hidden according to the cursor position. This function is particularly useful, for example, when working with a small monitor (e.g., a notebook).

The auto-hide function is enabled/disabled via the pin icon in the title bar.



Auto-hide is disabled. When you move the cursor out of the window, the window stays open (see A in Figure 3-10).



Auto-hide is enabled. When you move the cursor out of the window, the window is minimized. To show the window again, place the cursor on the minimized window (see B in Figure 3-10).

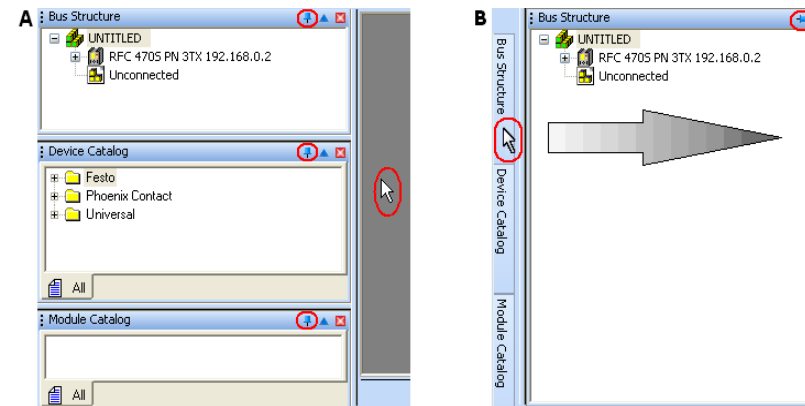


Figure 3-10 Auto-hide function

### 3.6 "Bus Structure" window






The "Bus Structure" window is used to display and edit the bus topology of the project.

#### 3.6.1 Icons in the "Bus Structure" window

##### 3.6.1.1 Logical device functions

The individual device functions are identified in the "Bus Structure" window with logical icons. The logical icons in the example bus configuration have the following meaning:

Table 3-2 Icons for logical device functions (examples)

Icon	Meaning
 Resource:	<b>Controller resource</b> When creating the project using a template, the controller resource is inserted below this icon. When creating the project using the wizard, the controller resource is inserted below this icon.
 PROFINET:	<b>PROFINET IO controller</b> Insert all PROFINET IO devices below this icon in the lower level.
 INTERBUS:	<b>INTERBUS master</b> Below this icon, insert the INTERBUS devices that are directly connected to the device that this icon refers to (local bus and remote bus).
 Inline:	<b>Inline</b> Below this icon, insert the Inline terminals that are connected to the Inline bus coupler (local bus).
	<b>PROFINET IO devices: Device proxy and modules</b> For PROFINET IO devices, the display is divided into the device proxy (bus interface) and its modules. The device proxy is indicated with "@". The slot number is specified after the graphical representation of the device proxy or modules (setting in context menu under "Edit Device Representation").  The icons for the device proxy and modules are stored in the device description file (e.g., FDCML, GSD). They may vary from manufacturer to manufacturer.  PROFINET IO devices from Phoenix Contact (Example: ILB PN 24 DI16 DIO16-2TX)  Default icons, if none are stored in the device description file.

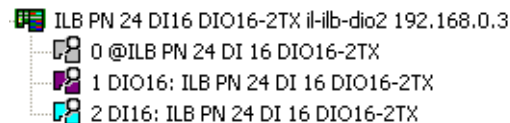
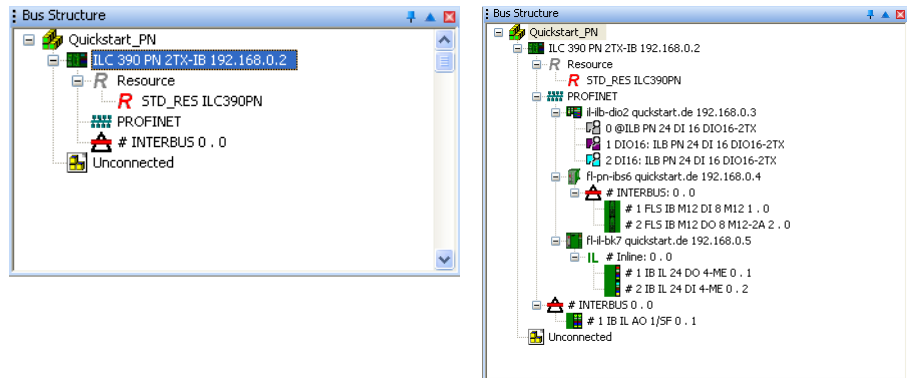


Figure 3-11 Device proxy and modules of a PROFINET IO device

For the example in Figure 3-11:

- Slot 0 is the device proxy for the PROFINET IO device
- Slot 1 is the DIO module for the PROFINET IO device
- Slot 2 is the DI module for the PROFINET IO device

Since the modules are fixed components of the PROFINET IO device, each one is marked with a pin. These elements cannot be deleted or replaced.



PROFINET IO controller

Overall structure with PROFINET IO devices and INTERBUS devices

Figure 3-12 Example for the representation of devices in the bus configuration

### 3.6.1.2 Validity of actions

When editing the bus configuration with the mouse, the mouse pointer indicates the validity of your actions.





Table 3-3 Icons when editing the bus configuration

Icon	Meaning
	Insert at the same level The device can only be inserted in/moved to the same level as existing devices.
	Insert in the lower level The device can only be inserted in/moved to a lower level than existing devices.
	Replace The device under the mouse pointer can be replaced by holding down the <Ctrl> key and placing the mouse pointer on the existing device.
	Not permitted This icon indicates a mouse pointer position for which actions are not permitted.

### 3.6.1.3 Display of status information

In the "Bus Structure" window some icons, which superimpose other graphics, are used to display important status information.

Table 3-4 Icons for displaying status information

Icon	Meaning
	The device is hidden or the bus is deactivated.
	Errors have occurred for the device.
	Warnings have occurred for the device.
	The pin indicates a module that is a fixed component of a device. This element cannot be deleted or replaced.

### 3.6.2 Display in the "Bus Structure" window

The display in the "Bus Structure" window can be adapted to your requirements.



- Switch to the bus configuration workspace.
- In the context menu, open a device via the "Edit Device Representation..." menu item.

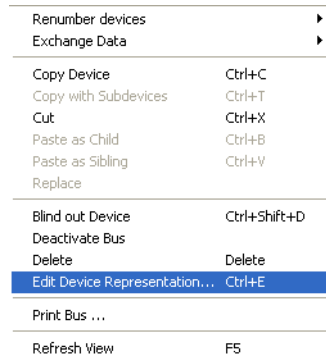


Figure 3-13 "Edit Device Representation..." menu item

- Select the criteria that you would like to see in the display.

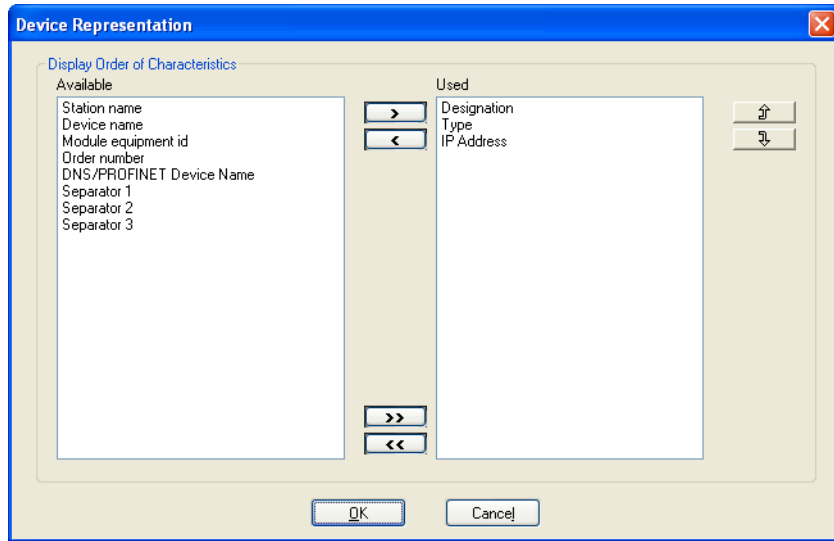


Figure 3-14 Device representations

Examples for different device representations:



Product designation

DNS/PROFINET device name, separator, and IP address

Figure 3-15 Examples for different device representations

The selected representation only affects the representation of a specific device group.

Groups include, for example:

- Controllers
- PROFINET IO devices
- INTERBUS devices

For example, the setting for PROFINET IO devices does not have any influence on the representation of the ILC 390 PN 2TX-IB (see Figure 3-15).

### 3.7 "EXCEL Link" window

The Excel link enables:

- Data to be exported to an Excel file
- Data to be imported from an Excel file

Data that can be exported/imported includes:

- IEC variables
- IEC variable connections to signals
- Associated data and text

The data for all IEC variables can therefore be edited in an Excel worksheet rather than in various dialog boxes in the PC WorX user interface.

Since IEC variables are edited in this window, it is useful to toggle the window on in the process data assignment workspace.

The following functions can be executed with the Excel link:

- Export existing IEC variables.
- Adjust and import the IEC variables available in an Excel file with a specified format with the IEC variables available in a PC WorX project.
- Settings that specify how you wish to identify devices.



For more detailed information about the Excel link, please refer to the online help for PC WorX.

To toggle the "EXCEL Link" window on, select the "View, EXCEL Link" menu item.



### 3.8 "Diag+" window

Diag+ is a diagnostics tool for the seamless diagnostics of INTERBUS and PROFINET. When installing PC WorX, Diag+ is installed automatically and is integrated in PC WorX. It can be called from PC WorX via the "Diag+" window.

However, Diag+ can also be started separately via "Start, All Programs, PHOENIX CONTACT, AUTOMATIONWORX Software Suite 201x 1.6x, Diag+ 2.xx".



An online connection to the controller is required in order to use Diag+.

For the following description, the example bus configuration for the PROFINET and INTERBUS system has been selected (see Figure 3-15 on page 3-13). The IB IL 24 DI 4-ME Inline terminal has been removed from the bus configuration in order to prompt an error.

- Start Diag+ via "View, Diag+".

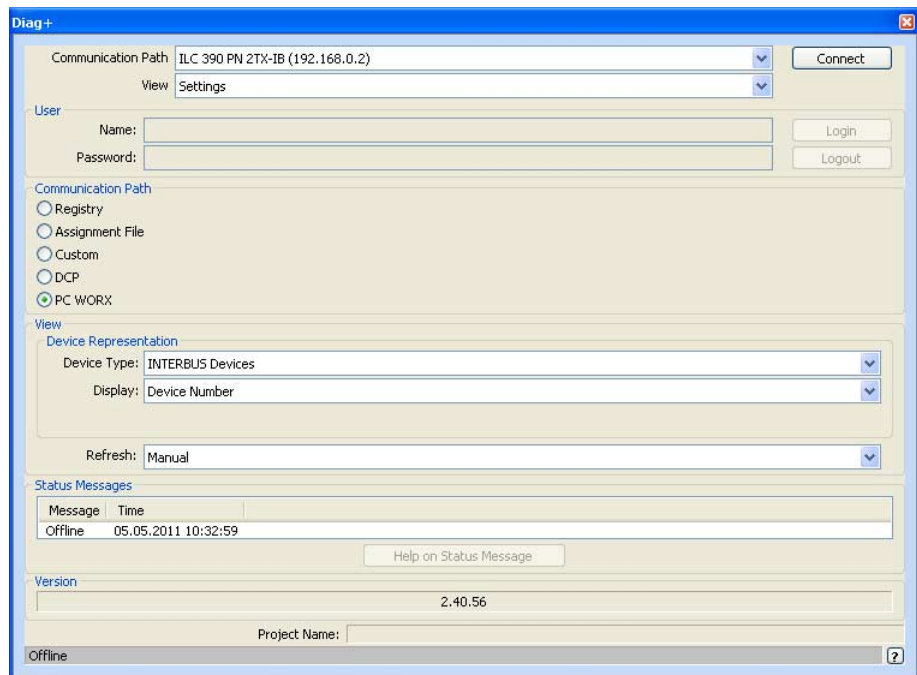


Figure 3-16 Diag+ start screen

- Under "Communication Path", select the device with which you would like to establish a connection.

The diagnostics represented depend on the communication path:

- PROFINET IO controller: INTERBUS diagnostics and diagnostics archive
- PROFINET IO devices: INTERBUS diagnostics

- Select, e.g., the ILC 390 PN 2TX-IB PROFINET IO controller as the communication path.

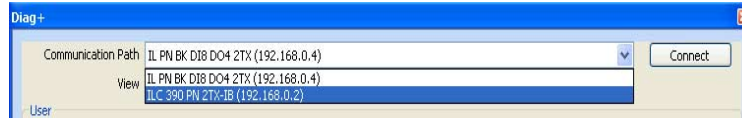


Figure 3-17 Selecting the communication path

- Click on "Connect" to activate the communication path.
- At the top under "View", select which general information is to be represented.

**View: Settings**

- At the bottom under "View", select which information is to be represented.

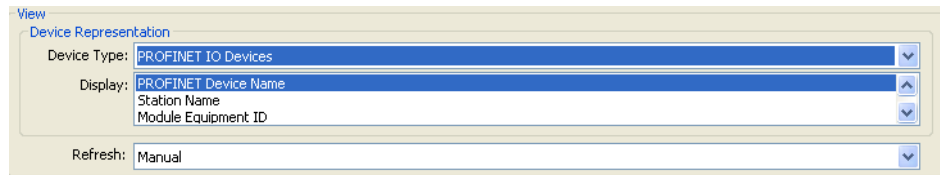


Figure 3-18 Example for device representation view

**View: Diagnostic archive**

All events that have occurred are displayed in list format. The display can be adapted to your individual requirements via the filter options at the top.

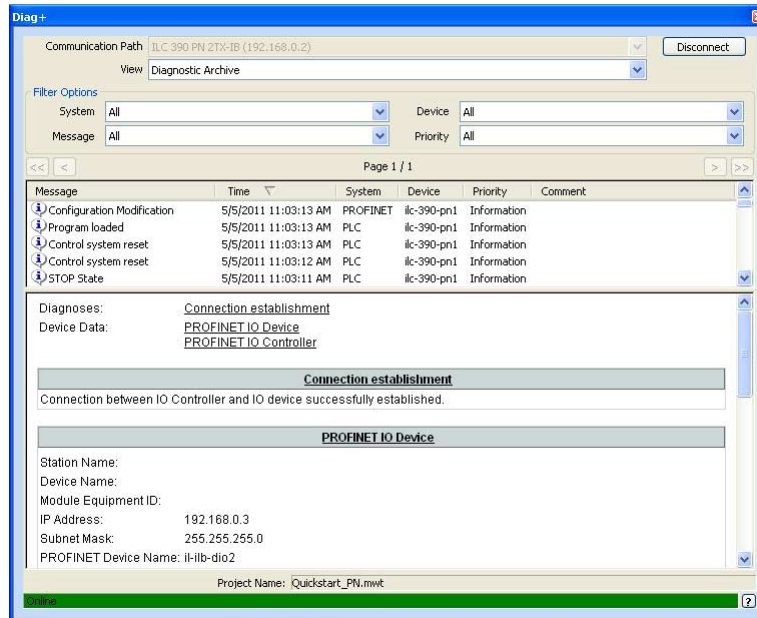


Figure 3-19 Diagnostic Archive

**View: INTERBUS Diagnostic**

INTERBUS modules in an INTERBUS system can be diagnosed with Diag+ as usual. Various information can be called via the different tabs.



**Diagnostics in lower-level INTERBUS systems**

Please note that for direct diagnostics of the Inline devices that are connected to the PROFINET bus terminal in the bus configuration in Figure 3-12 on page 3-11, the communication path can now be changed from the ILC 390 PN 2TX-IB to the FL IL 24 BK-PN-PAC. To this end, you must disconnect the communication path to the ILC 390 PN 2TX-IB, change it to the FL IL 24 BK-PN-PAC, and then activate the communication path ("Connect" button).

Switch to "View: INTERBUS Diagnostic".

- Under "View, Device Representation", select "INTERBUS Devices" as the device type.
- At the top, select "INTERBUS Diagnostic" as the view.

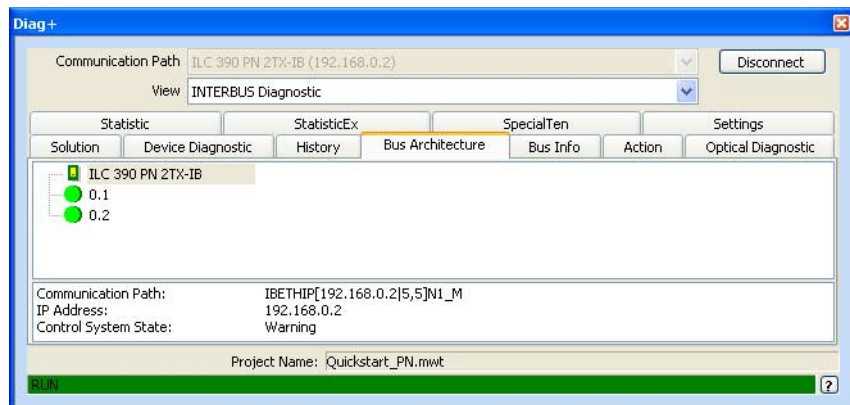


Figure 3-20 INTERBUS Diagnostic: Bus Architecture (no error)

An error is simulated, the IB IL 24 DI 4-ME Inline terminal is disconnected.

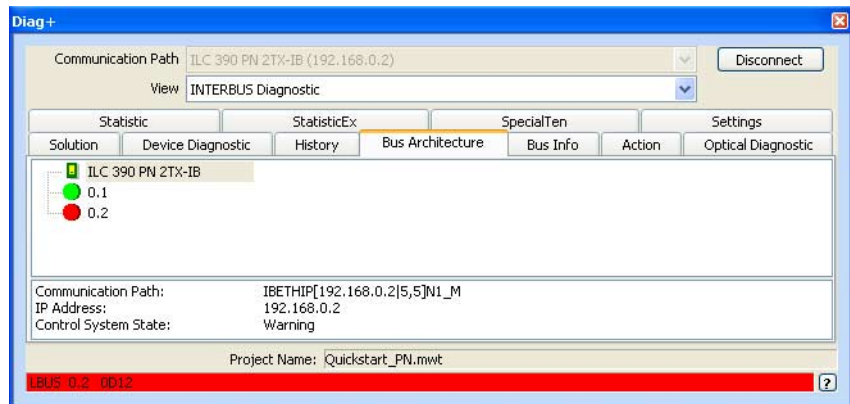


Figure 3-21 INTERBUS Diagnostic: Bus Architecture (error)

- Switch to the "Solution" tab.

Here, information is provided about how to remove the error.

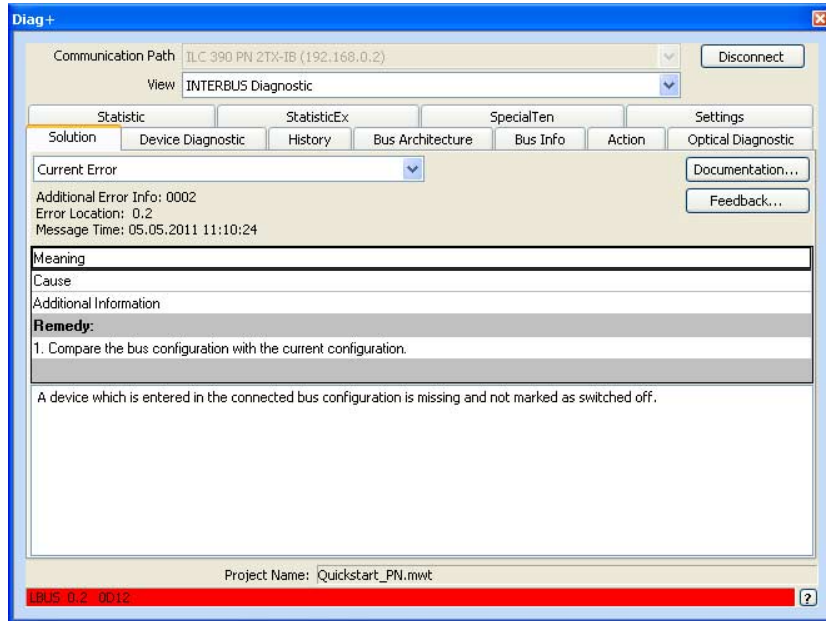


Figure 3-22 INTERBUS Diagnostic: Solution

- Remove the error (here: insert Inline terminal again).

For PROFINET, the entire system is automatically restarted by the PROFINET IO controller.

If the bus is not restarted automatically, the "Action" tab can be used, for example, to acknowledge errors, reset the controller board, and start the bus.

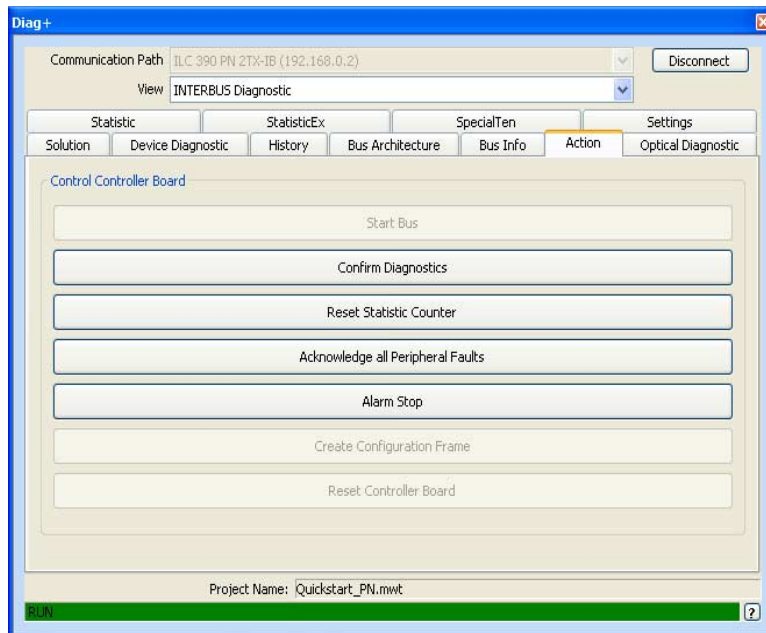


Figure 3-23 Diag+: Action

**View:  
Ethernet Topology**



The Ethernet topology can also be displayed in Diag+.

Please note that the Ethernet topology can only be displayed if it has been previously generated in PC WorX under "View, Ethernet Topology".

### **3.9 Visualization**

The data generated in PC WorX can be used in other programs, e.g., to visualize processes (e.g., Visu+, WebVisit).

The AX OPC Server is provided in order to enable the use of data in OPC clients.

See also "AX OPC Server and WebVisit" on page C-1.

## 3.10 FDT (Field Device Tool) workspace

### 3.10.1 Introduction

FDT (Field Device Tool) is the name of the software specification that standardizes the fieldbus-independent configuration, parameterization, and diagnostics of devices in an automation network.

To this end, the device manufacturer provides device-specific drivers and user interfaces known as DTMs (Device Type Managers). The DTM of a device or a device group contains, for example, a view of selected data from the device description and a device-specific user interface. Using the DTM, a device is integrated into the automation network via an FDT framework application (also known as FDT container software, e.g., PC WorX).

#### FDT Group

Phoenix Contact is a member of the FDT Group ([www.fdtgroup.org](http://www.fdtgroup.org)). Various DTMs are available from Phoenix Contact, some of which are certified by the FDT Group.

#### Internet

Device-specific DTMs, which are not included in the DTM library installed as part of the AUTOMATIONWORX Software Suite, can be downloaded at [www.phoenixcontact.net/catalog](http://www.phoenixcontact.net/catalog). The latest DTM libraries are available at [www.automationexplorer.phoenixcontact.com](http://www.automationexplorer.phoenixcontact.com).

For DTMs for devices from other manufacturers, please contact the relevant manufacturer.

### 3.10.2 General

PC WorX software is an FDT framework application for integrating DTMs. Once DTM support has been activated for PC WorX, the DTM library will be available the next time PC WorX is started. Any other DTMs from different manufacturers can be integrated.

Using DTMs, point-to-point communication, even beyond network boundaries, enables user-friendly parameterization and diagnostics of devices and sensors/actuators via, e.g., Ethernet, INTERBUS, PROFIBUS, HART, IO-Link protocol, and also PROFINET IO in the future.

### 3.10.3 Installing DTM libraries

#### Activating DTM support



DTM support must be activated in PC WorX. Please note that this functionality is deactivated by default.

Close PC WorX before activating/deactivating DTM support.

- To activate the DTM functionality in PC WorX, select "Start, All Programs, PHOENIX CONTACT, AUTOMATIONWORX Software Suite 201..., Tools, Activate DTM support".

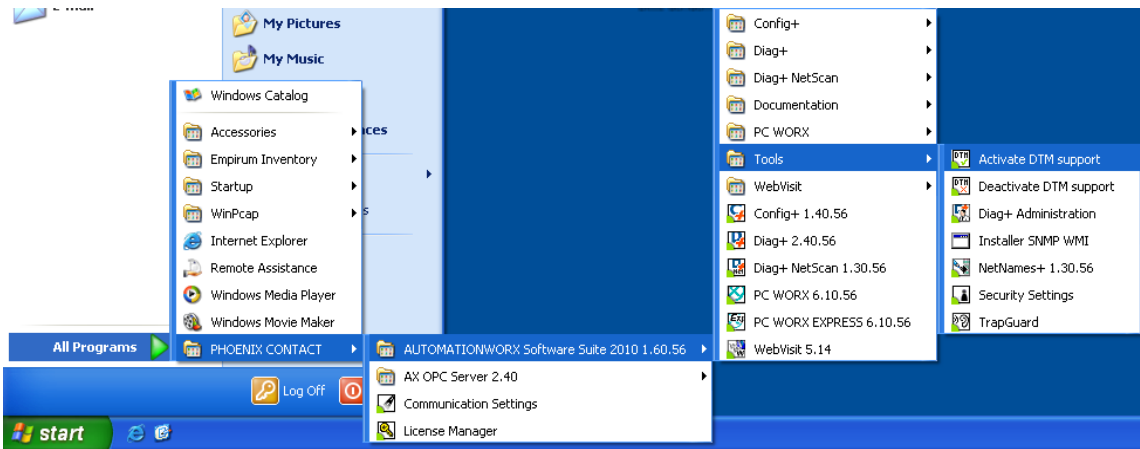


Figure 3-24 Activating DTP support

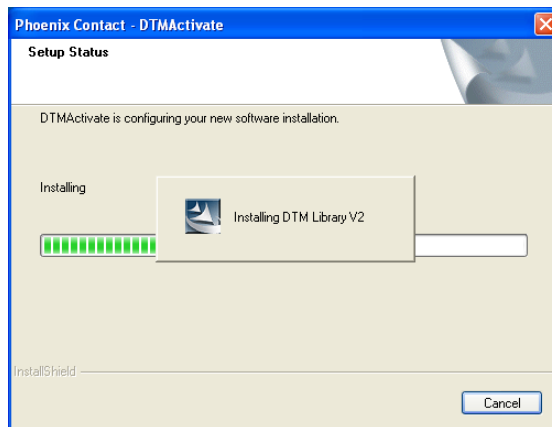


Figure 3-25 DTM support: Installing the DTM library

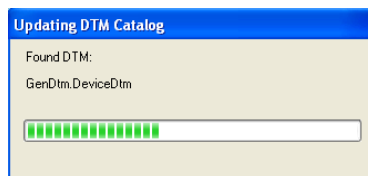


Figure 3-26 DTM support: Updating the device catalog

**Installing the AX DTM Library**

The AX DTM Library is also installed with AUTOMATIONWORX Software Suite 201x 1.6x. To update the AX DTM Library, e.g., in order to integrate new devices in the software, a setup option is available which can be used to install a new version of the AX DTM Library.



**Prior to commencing installation of DTM libraries, close PC WorX.**

- If the AX DTM Library is already installed, an update installation will be performed whenever the setup program is started. Only the features that were previously installed will be updated.
- If the AX DTM Library has not yet been installed, a brand new installation will be performed when the setup program is started. The features that have been selected automatically or manually will be installed.

**Installing the DTM**

If a library has already been installed, there are two options for installing the DTM for the IB IL IFS-MA-PAC terminal:

1. Uninstall the old library:  
Next time setup is executed, all automatically or manually selected features will be installed.
2. Install the new setup package without uninstalling the old library:  
Following installation, the newly added features (e.g., the IB IL IFS-MA-PAC terminal) are not yet available.  
To activate the new features, proceed as follows:
  - a) Select "Start, Control Panel, Add or Remove Programs".
  - b) Select "AX DTM Library".
  - c) Click on "Change".
  - d) In the welcome screen of the installation wizard, click "Next".
  - e) In the "Program Maintenance" window that appears, select the "Modify" option and confirm your selection with "Next".
  - f) The installation wizard displays all the features available in the setup.  
The symbol before an entry indicates the status of the feature:  
Hard disk = Installed/install; Red cross = Not installed/remove

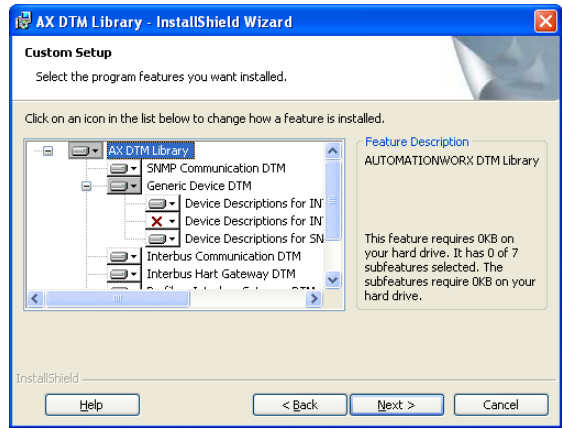


Figure 3-27 Features in the setup



- g) Click on the symbol before the entry to change the state of the feature (e.g., IB IL IFS-MA DTM).

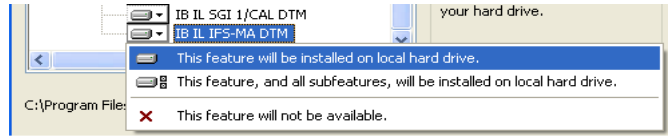


Figure 3-28 Activating a feature

- h) Confirm your selection with "Next".
- i) Close the installation wizard by clicking on "Install".

**Installing additional device DTMs individually (example: CONTACTRON DTM)**



Please note that individual DTMs for devices from Phoenix Contact only have to be installed if these devices are not included in the installed DTM library. For devices from other manufacturers, installation is always required.

- Save the setup file for the device DTM to be installed onto your PC.
- Start setup for the CONTACTRON DTM by executing the setup file.
- Follow the instructions of the installation wizard.
- In the "Required components..." window, select the entry highlighted in Figure 3-29.

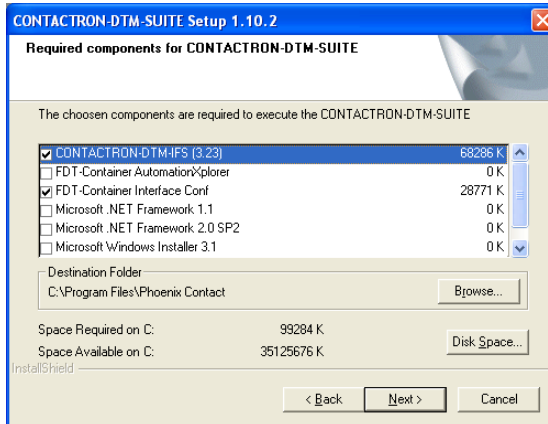


Figure 3-29 CONTACTRON DTM - Selection for PC WorX

- Follow the instructions of the installation wizard.

### 3.10.4 Creating a project and integrating DTMs in PC WorX

The next time PC WorX is started, a window appears indicating the progress of the search for new installed DTMs in the registry.

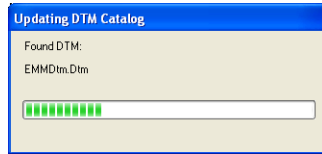


Figure 3-30 Search progress

- Create a new project with the controller used (here: ILC 150 ETH) via the "File, New Project" menu.
- Save the project via "File, Project, Save Project As / Zip Project As" (here: Quickstart\_PCW\_IFS\_MA, for example).
- To integrate the detected DTMs into the device catalog, select the "Import from DTM Catalog..." menu item.

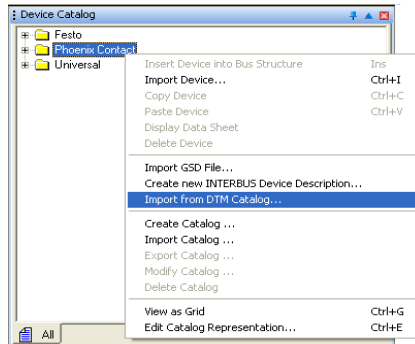


Figure 3-31 Integrating DTMs into the device catalog

The message "Do you want to scan the complete registry again to find new installed DTMs?" appears. As the registry was scanned when PC WorX was restarted, you can answer this prompt with "No".

A prompt may then appear (possibly multiple times) for various device descriptions asking whether the existing devices in the device catalog should be replaced. Answer "Yes" to these prompts..



The process described applies to DTMs from Phoenix Contact as well as DTMs from other manufacturers.

The PC WorX message window indicates which devices have been imported.

In the device catalog, all the devices are now displayed sorted by company.

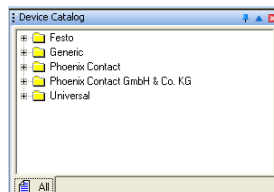


Figure 3-32 Device catalog following import

### 3.10.5 Reading in the bus configuration and/or manually inserting devices

At this point you must now read in the current bus configuration or manually configure the desired bus configuration. For detailed instructions on how to proceed, please refer to Section 4, "Sequence for creating a project".

When you manually insert a DTM-compatible device into the bus configuration, depending on the device and the position of the device in the bus configuration in "DTM View" (menu: "View, DTM View"), more detailed information about the device is displayed. The following is displayed in the example in Figure 3-33:

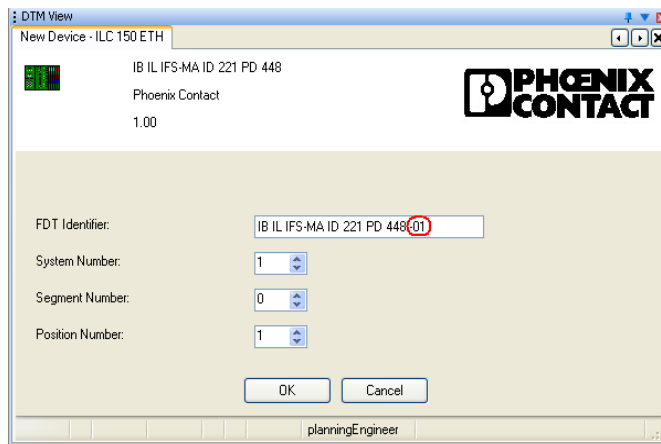


Figure 3-33 DTM View: New Device – ...

The following settings are shown in Figure 3-33:

- FDT Identifier: Specify a unique ID for the device here. This is particularly important if several devices of the same type are used in the bus configuration.
  - System Number: Specifies the number of the controller to which the devices used in your application are connected. In an application with just one controller, the controller number is "1".
  - Segment Number\*: Number of the level in which the device is used in the application.  
Segment number = 1: Corresponds to the control level  
Segment number = x: Lower level than the controller
  - Position Number\*: Position of the device in the segment/bus configuration.
- \* = For the segment and position number, you can refer to the "Bus Structure" window in PC WorX (bus segment number/position number)

**Settings for DTM-compatible devices**

Depending on the DTM-compatible devices to be inserted in the bus configuration, settings are required that may differ according to the device.

The following example shows the input window for a required device address and a device name. If the "DTM View" window is hidden, it can be displayed via the "View, DTM View" menu. The name ("Tag") and the device address of the DTM-compatible device can be edited.

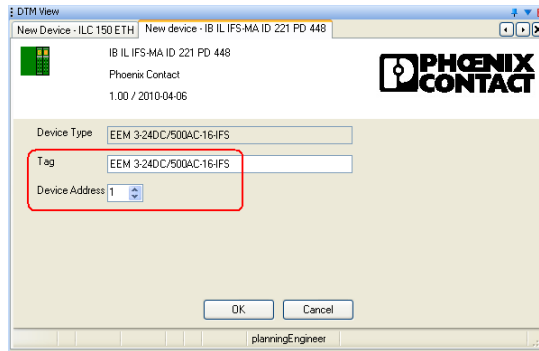


Figure 3-34 "DTM View" window: Device address and name for EMM...

Changes to the default settings for the name and device address must be confirmed with "OK".

### 3.11 Calling DTM functions

- To call the DTM functions of a device, select the "DTM Functions" item in the context menu (right mouse button) followed by the desired function.

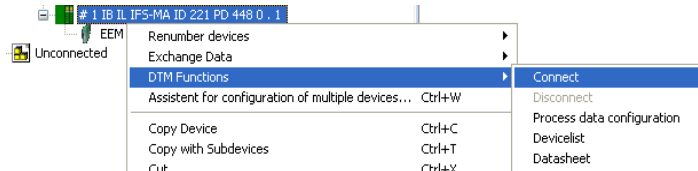


Figure 3-35 Calling DTM functions

#### 3.11.1 General DTM functions

##### Connect

Establishes the connection between the DTM and the device.

##### Disconnect

Disconnects the connection between the DTM and the device.

Connected devices are represented with a connection symbol.

#### Display of connected devices

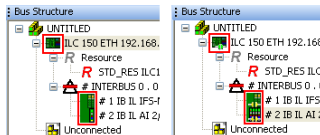





Figure 3-36 Connection representation

Key:

-  Device not connected
-  Device connected
-  Device actively connected



The connection representation in PC WorX only changes to green (active connection) when another DTM function is selected on this device.

### 3.11.2 DTM functions of the controller

- To use the DTM functions of the controller, connect the controller.

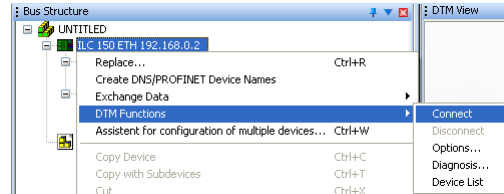


Figure 3-37 DTM functions of the controller

#### Settings

The settings for the communication path are displayed and can be modified, if required.



Make sure that the settings in this window match the settings in the "Communication" tab in the "Device Details" window of the controller.

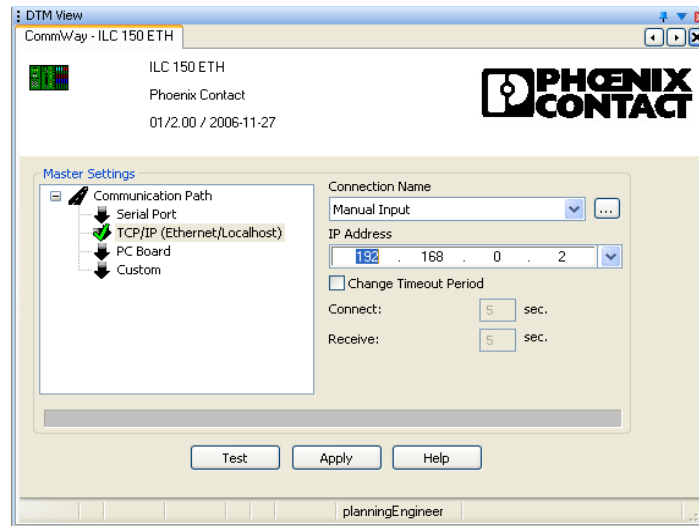


Figure 3-38 DTM functions: Settings – Communication path of the controller

#### Diagnostics

Diagnostic information is made available for the communication DTM. With regard to DTM functions, the same functions as those in Diag+ are available here.



This function has been implemented within the DTM and is therefore also available in PC WorX. It corresponds to the displays in Diag+. This diagnostic display can be used in other FDT framework applications, such as AutomationXplorer+, which do not offer diagnostics.

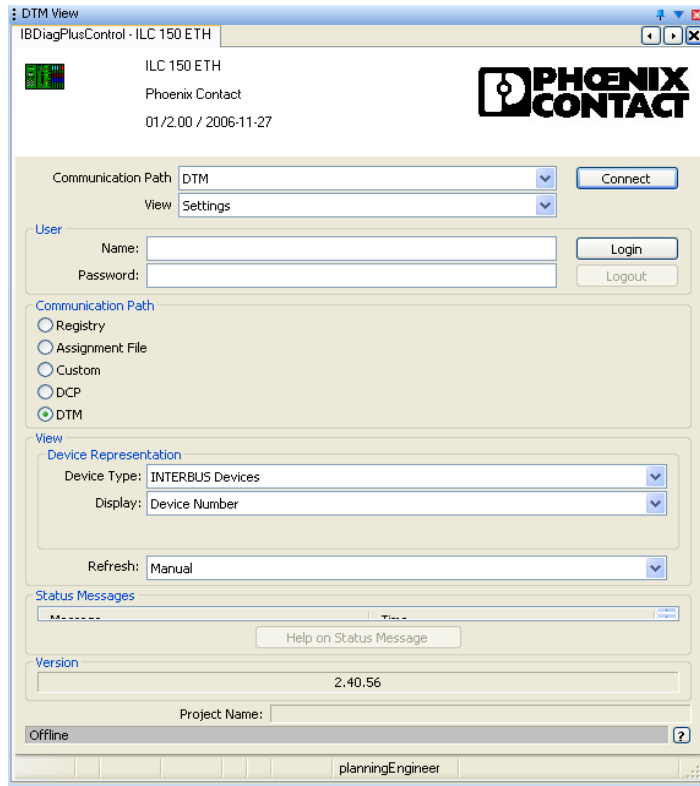


Figure 3-39 DTM functions: Diagnostics

**Device list**

The list of DTM-comptabile devices connected to a communication DTM or gateway DTM is displayed and can be modified, if required (e.g., entry of FDT identifier).

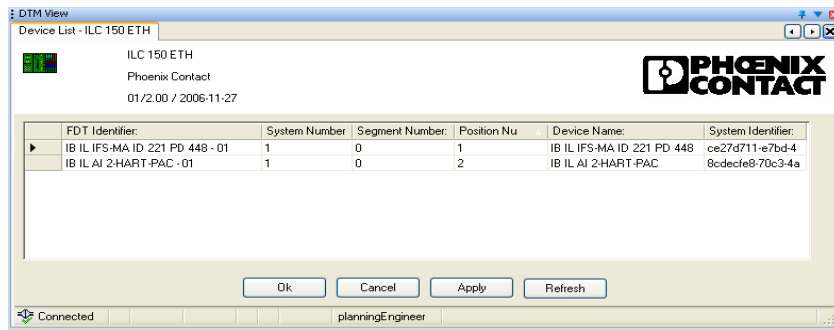


Figure 3-40 DTM functions: Controller device list

### 3.11.3 DTM functions of devices

- To use the DTM functions of the devices used, connect the relevant device.

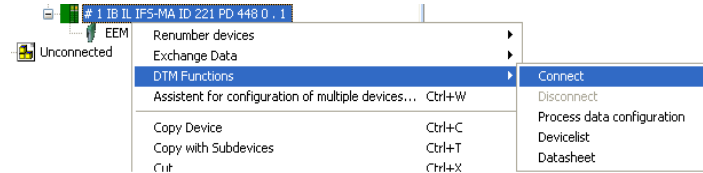


Figure 3-41 DTM functions of devices



**DTM functions of devices – Refer to user documentation**

The order of the functions displayed in the context menu is not the same as the order in which they are processed.

For the execution of the relevant function, please refer to the device-specific user documentation.



## 4 Sequence for creating a project

This section describes the general procedure for creating a project.




**The aim of this section is** to provide an overview of the various options available for achieving a particular objective and to explain some points in detail that are only referred to briefly in later sections.

**The aim is not** to create a comprehensive project or to have an operational project at the end of the section. If you create a project using the description in this section, please note that you must adapt the explanations for your project.

If you have little or no experience in creating projects, please proceed as described in Sections 5, 6 or 7. Each of these sections provides a brief description of a specific project creation. The various sections focus on the following topics:

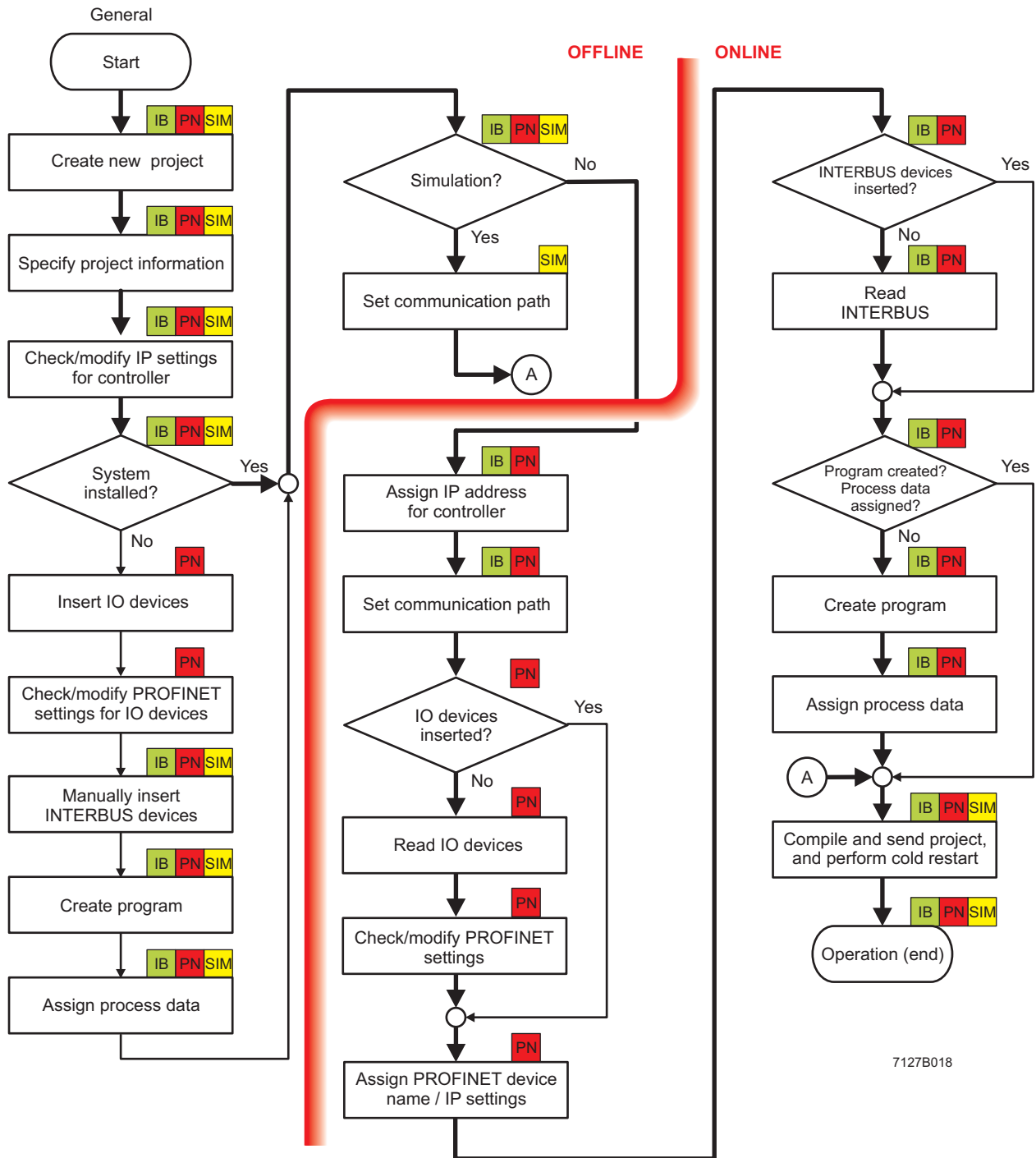
- Section 5, "Example project for an INTERBUS system"
- Section 6, "Example project for a system consisting of PROFINET IO and INTERBUS"
- Section 7, "Example project for a simulation with processor type "IPC" controllers"

The sections are labeled. The symbol next to the heading indicates the applicable versions:

 IB	INTERBUS project
 PN	PROFINET project
 SIM	Simulation project



### 4.1 Sequence for creating a project



7127B018

Figure 4-1 Sequence for creating a project



## 4.2 Creating a new project

- Select the "New Project..." command from the "File" menu to create a new project using a template.

The tree structure and the selection of the controller are now prepared.

- Select the controller.  
For some controllers there are several templates depending on the hardware and firmware version of the controller used.
- Confirm your selection with "OK".

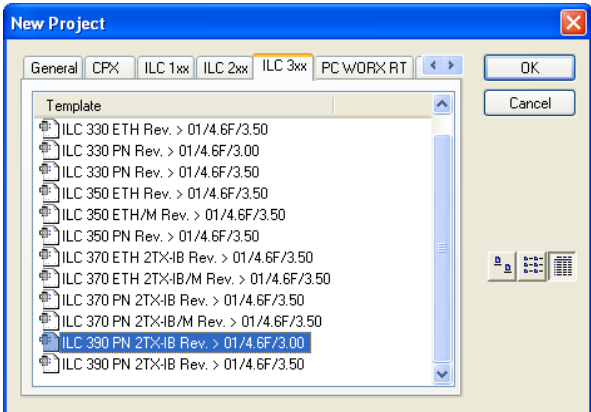


Figure 4-2 Project templates (here: "ILC 390 PN 2TX-IB Rev. > 01/4.6F/3.00")



"List" view has been selected in Figure 4-2. The view can be switched via the icons.



The ">" character in the selection stands for "later than" and not "greater than" in a mathematical sense. The specified version is the first supported version.

- Select the "File, Save Project As / Zip Project As..." command.

- Enter a project name and save the project.

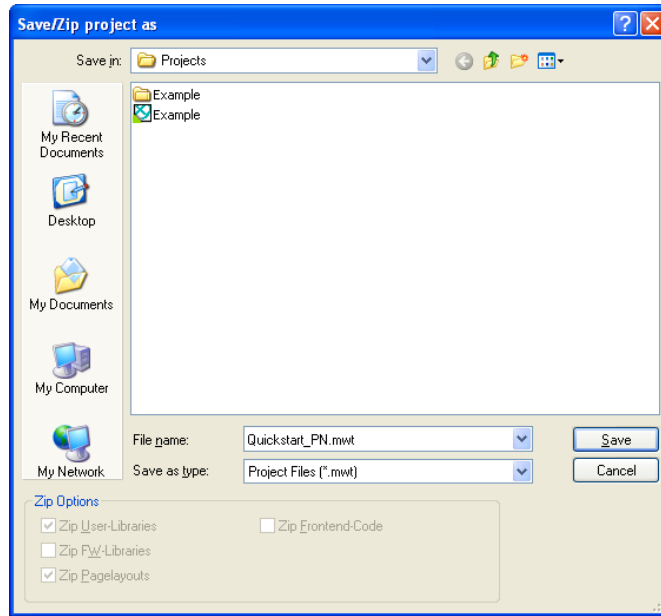


Figure 4-3 "Save/Zip project as" window



### 4.3 Specifying project information



- Switch to the bus configuration workspace.

After creating a new project, the project information is displayed in the bus configuration workspace.

- Adapt the project information to your project.

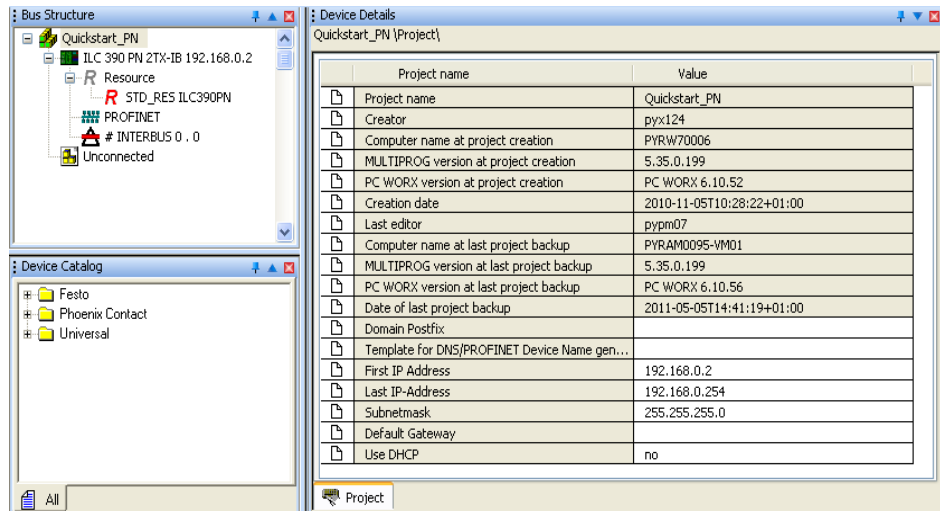


Figure 4-4 Project information after creating a new project

During project creation, PC WorX automatically assigns an IP address area for a **local network** (area from 192.168.0.2 to 192.168.0.254). If you would like to use another address area (e.g., a **global network**), adapt the start and end address on the project node (here: Quickstart\_PN) to your application.



Make sure that all the components are in the same subnetwork.  
 For information about the IP parameters to be used within your system, please contact your system administrator.  
 If the start address of the new address area to be entered is higher than the previous end address, please modify the end address first.

When inserting PROFINET IO controllers and PROFINET IO devices, PC WorX automatically assigns IP parameters that are within the specified area. If later on you manually assign IP parameters that are outside the defined area, they will not be accepted.

If you are using a default gateway:

- Assign the address for the default gateway under "Default Gateway" in the "Device Details" window.

This address must be within the specified IP address area. If the default gateway is entered at the start of the project, it will be used automatically for all subsequent PROFINET IO devices that are inserted (see also "Default Gateway" on page 4-11).



- Assign the name suffix for the PROFINET device name (here: quickstart.de) in the "Device Details" window under "Domain Postfix".

The following characters are permitted for the "Domain Postfix":

Lower case letters without umlauts     a to z

As separator: Dot     .



Upper case letters without umlauts are permitted, however, they will be converted to lower case letters internally. The entry is therefore not case-sensitive.

If the "Domain Postfix" is entered at the start of the project, it will be used automatically for all subsequent PROFINET IO devices that are **inserted** in the PROFINET device name (see also "DNS/PROFINET Device Name" on page 4-10).



If the PROFINET IO devices are **read in**, the PROFINET device name for the devices will also be read in. In this case, the "Domain Postfix" may differ. Adapt it accordingly.

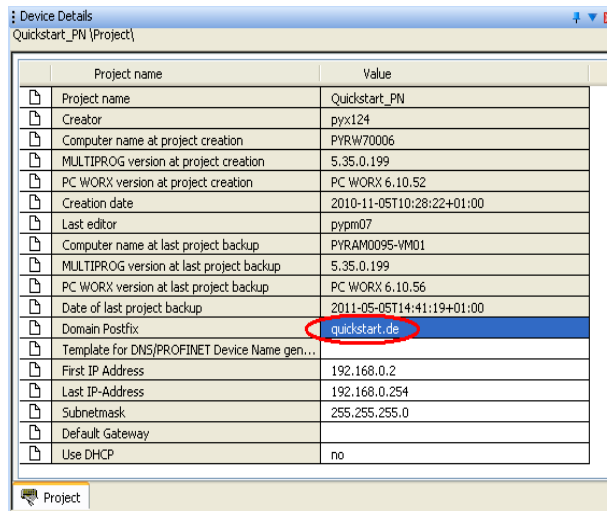


Figure 4-5 Assigning the "Domain Postfix"



#### 4.4 Preparing the PC for communication

- In the "Extras, PROFINET Configuration..." menu in PC WorX, select the network card for your computer that is to be used for communication.

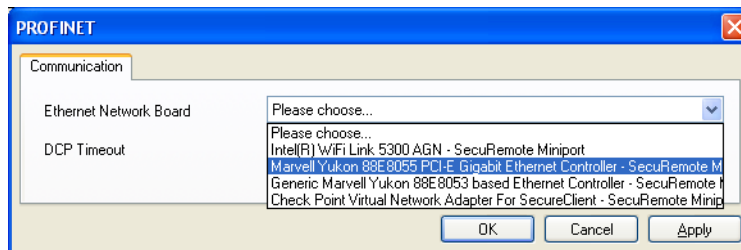


Figure 4-6 Selecting the network card

- Make sure that your PC has been assigned IP parameters that can be used to communicate with the connected network.  
If, e.g., when using PROFINET, you are working with a local network that is set by default (192.168.0.2 to 192.168.0.254), the IP parameters for the PC must be 192.168.0.1, with subnet mask 255.255.255.0.



## 4.5 Checking/modifying IP settings for the controller

The IP settings for the controller are made when the project is created.



### **NOTE: Changes to project information are not applied automatically**

If any modifications are made to the project information that affect the IP settings for the controller, a warning is displayed. However, the modification is not implemented automatically. When a new project is created, the default settings are specified under the IP settings.

Adapt these settings, if necessary.



- Switch to the bus configuration workspace.
- Select the controller node.
- In the "Device Details" window, switch to the "IP Settings" tab.
- Check the IP settings and modify them, if necessary.

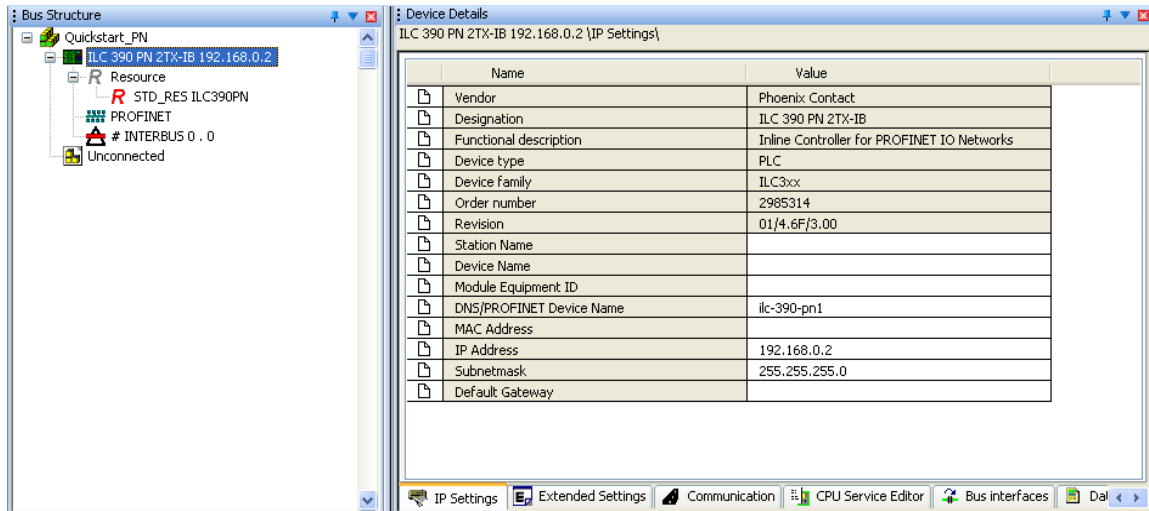


Figure 4-7 IP settings

- If an IP address has still not been assigned, assign one according to "Assigning the IP address for the controller" on page 4-23.



The IP parameters that are assigned here for the controller are also implemented as the IP parameters for the communication path via TCP/IP.



## 4.6 Decision: Working online or offline

If your system is installed, you can work online. In this case, skip this section and proceed to Section 4.17, "Setting the communication path".

If your system is not installed or you wish to work offline, proceed as follows:

- For PROFINET, see Section 4.7, "Inserting PROFINET IO devices".
- For INTERBUS or Simulation, see Section 4.9, "Manually inserting INTERBUS devices".



## 4.7 Inserting PROFINET IO devices



- Make sure you are in the bus configuration workspace.

Insert the PROFINET IO devices below the PROFINET IO controller node.

- If the device catalog is hidden, show it by selecting the "View, Device Catalog" menu.
- Open the "Phoenix Contact" device catalog.

For example, the specified folders contain the following PROFINET IO devices:

PROFINET IO device	Folder
FL IL 24 BK-PN-PAC	FL - Gateway
ILB PN 24 DI16 DIO16-2TX	ILB - I/O digital
FL PN/IBS	New: FL - Gateway

- Select the first PROFINET IO device (here: FL IL 24 BK-PN-PAC).

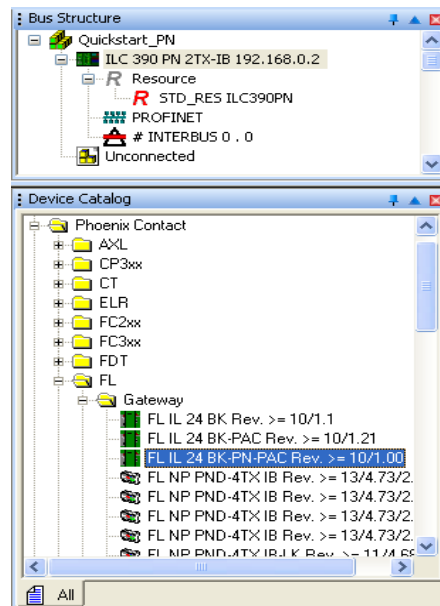


Figure 4-8 Selecting the PROFINET IO device



- Hold down the left mouse button and move the first PROFINET IO device in the "Bus Structure" window to the right of the PROFINET IO controller icon until the "Insert in the lower level" icon appears (see page 3-11).
- Move all other PROFINET IO devices to below the preceding PROFINET IO device until the "Insert at the same level" icon appears (see page 3-11).

Figure 4-9 shows the bus configuration with inserted PROFINET IO devices.

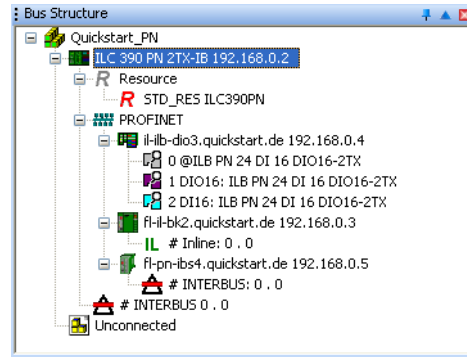


Figure 4-9 PROFINET IO devices inserted



## 4.8 Checking/modifying the PROFINET settings for PROFINET IO devices

When **inserting** each PROFINET IO device, the PROFINET settings are assigned automatically according to the entries defined in the project information.

The PROFINET settings can be checked and modified via the "PROFINET Settings" tab.



- Make sure you are in the bus configuration workspace.
- In the "Bus Structure" window, select the PROFINET IO device.
- Under "Device Details", select the "PROFINET Settings" tab.
- Check the PROFINET settings and modify them, if necessary.

Name	Value
Vendor	Phoenix Contact
VendorID	0x00B0
Designation	ILB PN 24 DI16 DIO16-2TX
DeviceID	0x0004
Functional description	PROFINET IO - Inline Block Device With 16 Digit...
Device type	I/O digital
Device Family	ILB
Order number	2878146
Revision	03/1.00
DNS/PROFINET Device Name	il-lb-dio3.quickstart.de
Station Name	
Device Name	
Module Equipment ID	
IP Address	192.168.0.4
Subnetmask	255.255.255.0
Default Gateway	
Realtime class	RT
Reduction ratio input	8 ms
Reduction ratio output	8 ms
Faulty telegrams until connection is aborted	24
Monitoring Time Inputs (ms)	192
Monitoring Time Outputs (ms)	192
Operation in case of configuration differences	no
Log connection state	yes
Drive BF	yes
Ersatzwertverhalten Eingänge	auf Null setzen
Verbindungsaufbau Anlauf	yes
Node ID	41

Figure 4-10 PROFINET settings for the ILB PN 24 DI16 DIO16-2TX

The PROFINET settings comprise:

### DNS/PROFINET Device Name

This name is the unique identification for the PROFINET IO device in the network. It must be known to the PROFINET IO device before it can be used in the network.

PC WorX automatically ensures that each device is assigned a PROFINET device name that is unique throughout the entire project. The "Domain Postfix" that was set on the project node is attached to this name (here: quickstart.de).

The PROFINET device name can be freely modified.

The following characters are permitted for the PROFINET device name:

Lower case letters without umlauts	a to z
Numbers	0 to 9
Hyphen	-

If the naming conventions have not been observed, an error message is output in the "Message Window" and the name is rejected.



Upper case letters without umlauts are permitted, however, they will be converted to lower case letters internally. The entry is therefore not case-sensitive.

The procedure that makes the device name known to the PROFINET IO device is either called "assigning device names" or "device naming". This procedure is described under "Transferring PROFINET device names and IP settings to PROFINET IO devices" on page 4-35.

**IP Address**

During operation, the device can be accessed via the IP address. PC WorX selects the address from the area that is set on the project node.



If the area for the IP addresses is later modified in the project node, you will also have to adapt the addresses of the PROFINET IO devices accordingly.

Specify the IP address according to "Transferring PROFINET device names and IP settings to PROFINET IO devices" on page 4-35.

**Subnetmask**

The subnet mask that was specified on the project node is assigned to each PROFINET IO device as the default subnet mask. It can be modified specifically for each individual device.

**Default Gateway**

If a default gateway is used, enter its IP address here. The default gateway specified on the project node is automatically assigned to each PROFINET IO device.



"IP Address", "Subnetmask", and "Default Gateway" represent the **IP parameters** for a device.

**Reduction ratio input/  
Reduction ratio output**

The time in which the inputs/outputs of the PROFINET IO device are updated. It can be set individually for each PROFINET IO device for both data directions.

The reduction ratio (update time) directly affects the number of PROFINET IO devices that can be operated via a PROFINET IO controller. The default setting displayed in PC WorX is the value that was stored in the FDCML file for the corresponding PROFINET IO device. If this setting exceeds the performance level of the PROFINET IO controller, a corresponding error message appears when translating the project. In this case, select a longer update time for certain devices.

**Monitoring Time**

The monitoring time can be set automatically by PC WorX according to the update times. The monitoring time specifies how long PROFINET IO devices may receive no valid PROFINET telegrams before substitute values are enabled at the outputs. This setting can affect the "noise immunity" of the network.



**NOTE:** Please note that excessively high settings can result in errors in the process as the PROFINET IO device "freezes" the last output value until the error is detected.



## 4.9 Manually inserting INTERBUS devices



### 4.9.1 Manually inserting INTERBUS devices - General

If the bus configuration is not actually available yet or you wish to create the configuration offline, the bus configuration can be created manually.

- Select a point in the "Bus Structure" window where a device is to be inserted (e.g., INTERBUS node for the controller).
- In the device catalog, open the product range for the inserted device (e.g., IL for Inline) under "Phoenix Contact".
- Under the product range, open the product group (e.g., I/O analog).
- Select the device to be inserted (e.g., IB IL AO 1/SF).

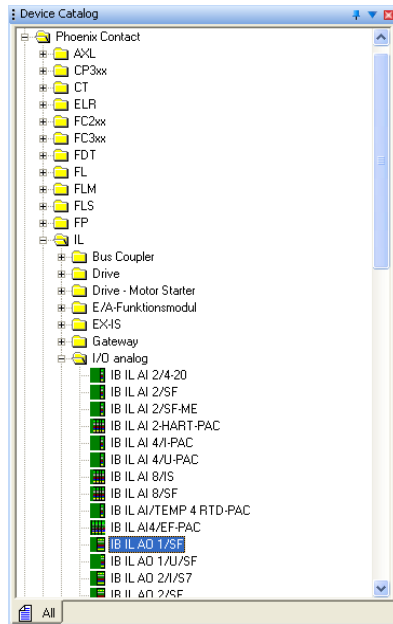


Figure 4-11 Selecting the device

- Hold down the mouse button and drag the selected device to the insertion point. Please refer to the information about the mouse pointer in "Icons in the "Bus Structure" window" on page 3-10.

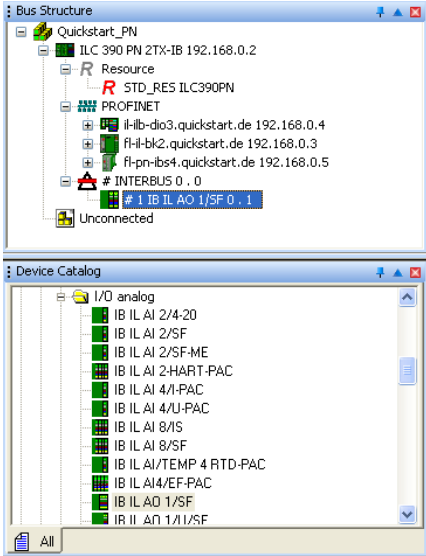


Figure 4-12 Device inserted (here: IB IL AO 1/SF)

- Insert all the other devices.

**Example bus configurations**

Example 1: Simple bus configuration

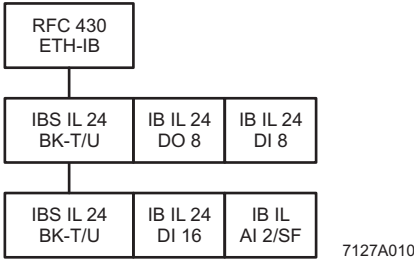
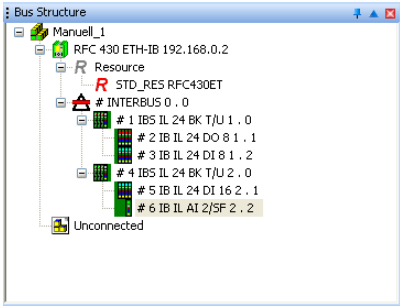


Figure 4-13 Simple bus configuration

Folders in the device catalog where the listed devices can be found:

- |                  |                  |
|------------------|------------------|
| RFC 430 ETH-IB   | RFC4xx - PLC     |
| IBS IL 24 BK-T/U | IL - Bus coupler |
| IB IL 24 D....   | IL - I/O digital |
| IB IL A....      | IL - I/O analog  |

Example 2: Bus configuration with branch terminals

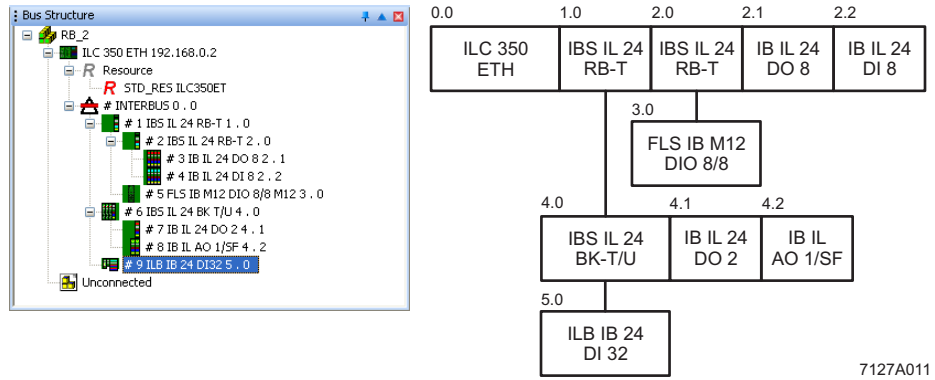


Figure 4-14 Bus configuration with branch terminals

Example 3: Bus configuration with bus coupler for connecting a remote bus branch

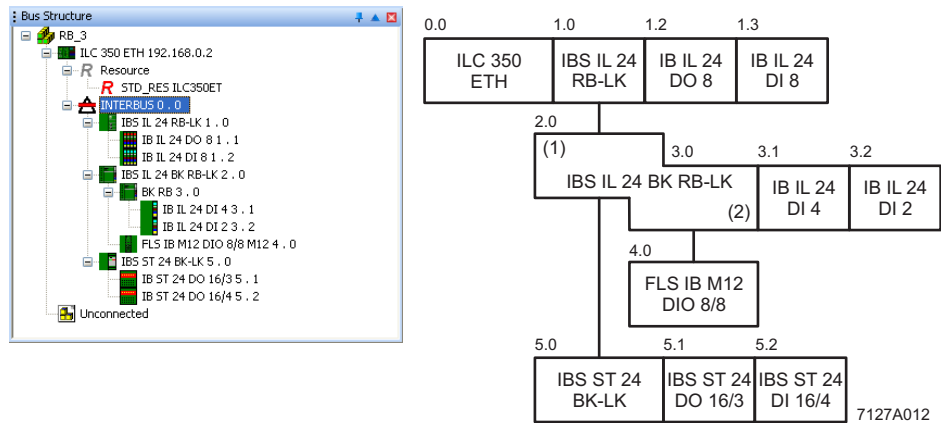


Figure 4-15 Bus configuration with branch terminals

Folders in the device catalog where the devices listed in Figure 4-14 and Figure 4-15 can be found:

- |                                 |                   |
|---------------------------------|-------------------|
| ILC 350 ETH                     | ILC3xx - PLC      |
| IBS IL 24 RB-T; IBS IL 24 RB-LK | IL - Bus coupler  |
| IBS IL 24 BK RB-LK              | IL - Bus coupler  |
| IB IL 24 D....                  | IL - I/O digital  |
| IB IL A....                     | IL - I/O analog   |
| FLS IB M12 DIO 8/8              | FLS - I/O digital |
| ILB IB 24 DI 32                 | ILB - I/O digital |
| IBS ST 24 BK-LK                 | ST - Bus coupler  |
| IBS ST 24 D....                 | ST - I/O digital  |

PN

#### 4.9.2 Inserting INTERBUS devices below an INTERBUS proxy

- Insert the required INTERBUS devices from the device catalog at the "INTERBUS" node of the INTERBUS proxy (e.g., FL PN/IBS).

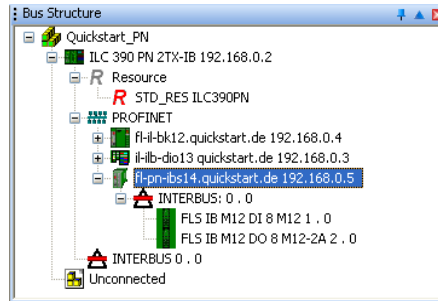


Figure 4-16 FLM modules inserted below an INTERBUS proxy

PN

#### 4.9.3 Inserting Inline terminals below a PROFINET IO bus coupler

- Insert the required Inline terminals from the device catalog at the "Inline" node of the PROFINET IO bus coupler (e.g., FL IL 24 BK-PN-PAC).

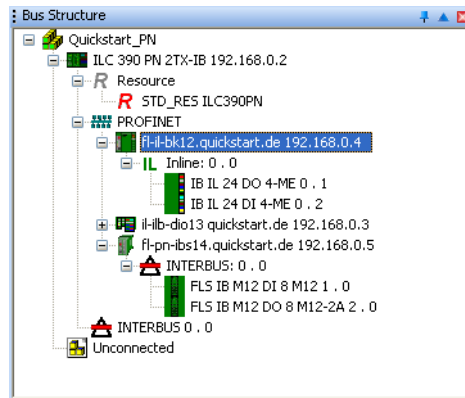


Figure 4-17 Inline terminals inserted below a PROFINET IO bus coupler

PN

#### 4.9.4 INTERBUS devices below the PROFINET IO controller

Manually insert the PROFINET IO modules connected to the INTERBUS controller in the same way as described above.

- Insert the required INTERBUS modules from the device catalog at the "INTERBUS" node of the PROFINET IO controller (e.g., ILC 390 PN 2TX-IB).



## 4.10 Compiling after completing the bus topology

At this point you can compile your project in order to detect any errors that may have occurred.

### Compiling a project

- When compiling a project for the first time, select the "Rebuild Project" command from the "Build" menu. For subsequent compiling processes, the "Make" command in the "Build" menu can also be used (see also "Compiling (additional information)" on page 4-16).

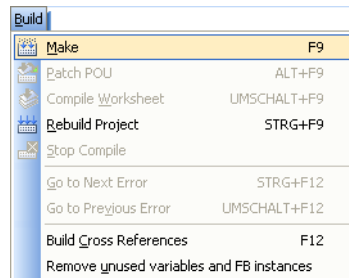


Figure 4-18 Compiling a project

- If errors occur when compiling, remove the errors and repeat the compiling process until it is completed successfully. Error messages must be removed. Warning messages do not have to be removed.



If you are compiling the project but have not yet programmed anything, you will receive the "Empty worksheet" warning message. This warning message does not affect the next step and can be ignored.

The results of the compiling process are displayed in the "Message Window" together with details of the number of errors and warnings.

### Sending a project: Special notes for PROFINET

When translating the controller project, the PROFINET IO configuration is generated automatically. This is also displayed in the "Message Window".

### Compiling (additional information)

There are two options for compiling:

- "Build, Make"
- "Build, Rebuild Project"

### Rebuild Project

Use this command to compile an **entire** project for the first time or after modifying a user library.

"Rebuild Project" compiles and links all worksheets. Errors and warnings that are discovered by the compiler are logged in the "Message Window". After the syntax has been checked successfully, the system automatically generates the IEC code and the special PLC code. Finally, the project can be sent to the PLC.

The "Rebuild Project" command should only be used if errors occurred when compiling with "Make" or your project was unpacked without frontend.



**Make**

The "Make" command is the standard mode for compiling. Use this command after editing and completing your project.

When the "Make" menu item is executed, all **modified** worksheets are compiled/linked and the modified PLC code is generated.

This command can be executed from the menu bar via "Build, Make", with the "Make" icon in the toolbar or using the shortcut <F9>.

Modified worksheets in the project tree are identified by an asterisk that appears after the worksheet name.

**Both**

After the compiler has been started, the "Message Window" appears automatically if it was closed before. This window shows the steps the compiler is currently performing. In addition, errors, warnings, and additional information about the process are logged here.

Once the compiling process has been completed successfully, i.e., no errors have been reported, the modified project can be sent to the PLC.



## 4.11 Creating the program

- Create the program.

To program the example program, proceed as described in "Example program" on page 8-1.



It is now assumed that you have created the program. If you skip this point, this may result in deviations.



## 4.12 Compiling after creating the program

At this point you can compile your project in order to detect any errors that may have occurred.



- Select the "Build, Make" command.



## 4.13 Generating variables and assigning process data

This section provides a general description for generating variables and assigning process data. Process data assignment for the example bus configuration is explained in later sections.



### 4.13.1 Generating variables

Usually, variables are either generated during **program creation** or they are created individually. If all the previous steps for creating a project including programming have been performed, variables will have been created.

Variables can also be **generated automatically** for the process data of all devices (PROFINET IO and INTERBUS). The variable names are assigned automatically according to the following pattern:

<I or Q>\_<PROFINET device name>\_<IBS segment>\_<IBS position>\_<PD name>

Key:

I or Q	I = Input; Q = Output
IBS	INTERBUS
PD name	Name of the process data item

For direct inputs/outputs or PROFINET IO status data, no INTERBUS data is required.

To generate variables, proceed as follows:



- Switch to the process data assignment workspace.
- Select the controller in the top right window. The standard configuration is displayed in the top left window, "Symbols/Variables".
- In the top left window, select the resource or program (in Figure 4-19: resource "STD\_RES: ILC390PN").



Please note the following:

- If the resource has been selected, global variables are generated/displayed that can be used in all POU's of the project (VAR\_GLOBAL).
- If the program (POU) has been selected, global variables are generated/displayed that can be used in all POU's of the project (VAR\_GLOBAL\_PG).

(POU = Program Organization Unit; see online help for PC WorX or IEC 61131-3)

- In the top right window, select the device for which you would like to link the process data to variables (e.g., IB IL AO 1/SF in Figure 4-21).
- Select the process data item for which you would like to generate a variable (~AO 16 in Figure 4-19).
- In the context menu for the process data item, select the "Create Variable" menu item.

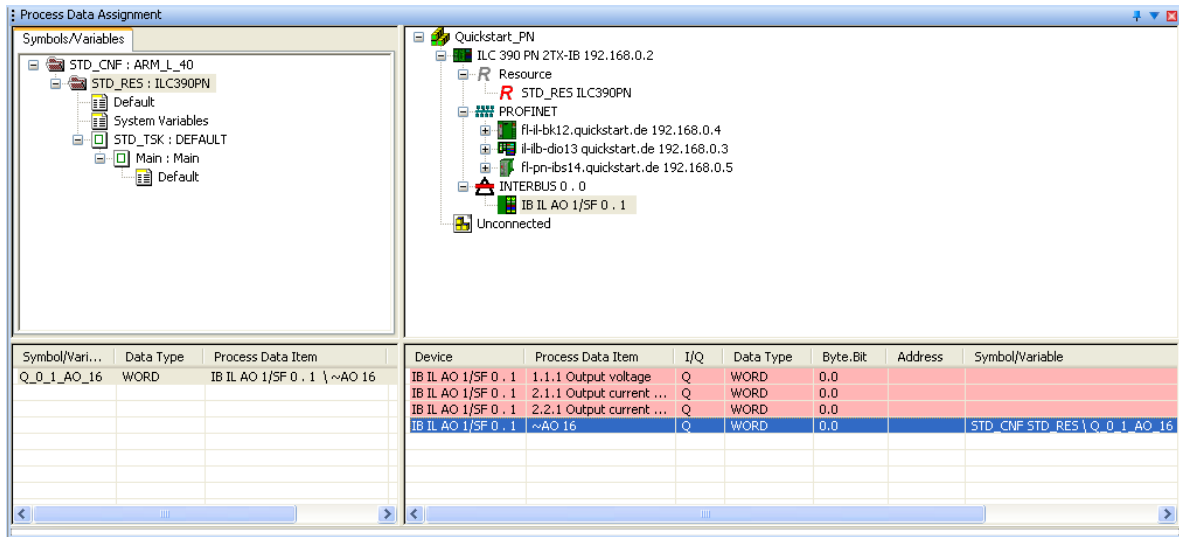


Figure 4-19 Variable generated for process data item "~AO 16"

Figure 4-20 shows examples of automatically generated variables.

Symbol/Variable	Data ...	Process Data Item
I_FL_PV_IB54_1_2_I3	BOOL	3 FLM DI 8 M12 \ I3
Q_FL_PN_IB54_1_3_O1	BOOL	4 FLM DIO 4/4 M12-2A \ O1
I_IL_ILB_DIO3_DI32	DWORD	ILB PN 24 DI16 DIC16-2TX \ ~DI32
I_FL_PV_IB54_PNIO_DATA_VALID	BOOL	FL PN/IBS \ PNIO_DATA_VALID

Figure 4-20 Examples of generated variables



### 4.13.2 Assigning process data

Process data and control variables are assigned in the process data assignment workspace.

The assignment is seamless for INTERBUS and PROFINET IO.



Please note the following:

- If the resource has been selected, global variables are generated/displayed that can be used in all POU's of the project (VAR\_GLOBAL).
- If the program (POU) has been selected, global variables are generated/displayed that can be used in all POU's of the project (VAR\_GLOBAL\_PG).

(POU = Program Organization Unit; see online help for PC WorX or IEC 61131-3)



#### EasySim simulation

Only the states of the global variables that have been linked to a physical address (inputs/outputs) as part of process data assignment are displayed in the simulation.



- Switch to the process data assignment workspace to assign the variables to the process data.
- Select the controller in the top right window. The standard configuration is then displayed in the top left window, "Symbols/Variables".
- In the top left window, "Symbols/Variables", select the standard resource (STD\_RES RFC450ET in Figure 4-21).
- In the top right window, select the device for which you would like to link the process data to variables (e.g., IB IL 24 DO 8 in Figure 4-21).
- Select the process data item to be linked (1.1.1 in Figure 4-21).



Please note that the variables shown in Figure 4-21 had to be created in the IEC programming workspace first before they could appear under "Symbols/Variables" in the process data assignment workspace.

- Using drag & drop, link the selected variable to one of the variables on the left-hand side (OUT1 in Figure 4-21).

In the bottom left window, the assignment between variables and process data is displayed.

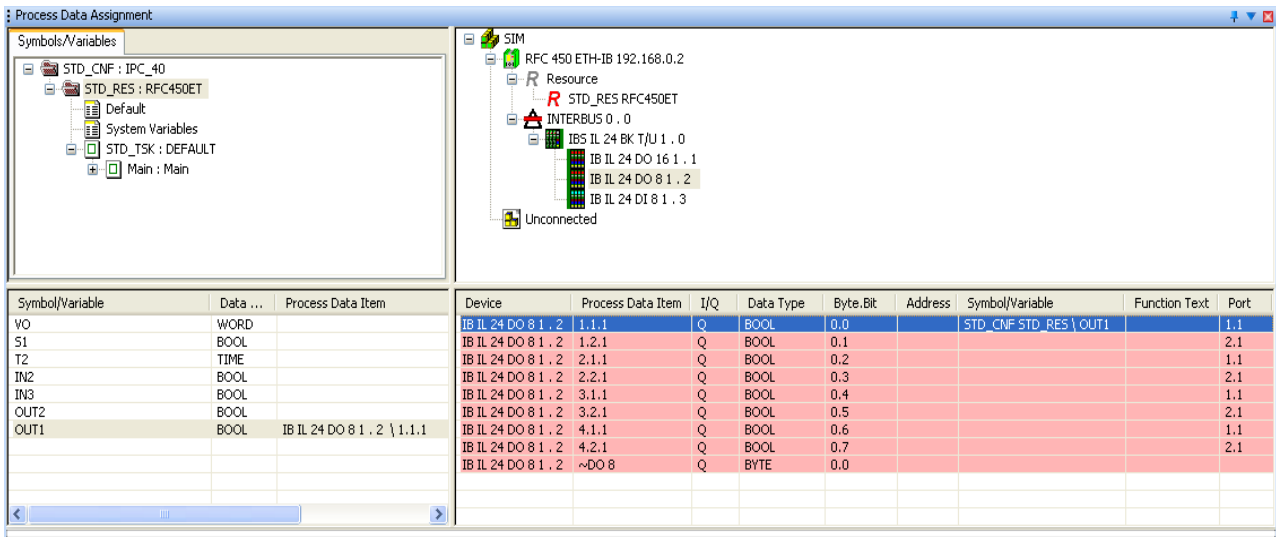


Figure 4-21 Process data item 1.1.1 assigned to variable OUT1

- Repeat this procedure for all inputs to be evaluated and for all outputs to be controlled.



Please note for a PROFINET IO device with inputs and outputs that this is represented with its modules.



Figure 4-22 Representation of a PROFINET IO device with its modules

- In this case, select the module where the required inputs or outputs are located.

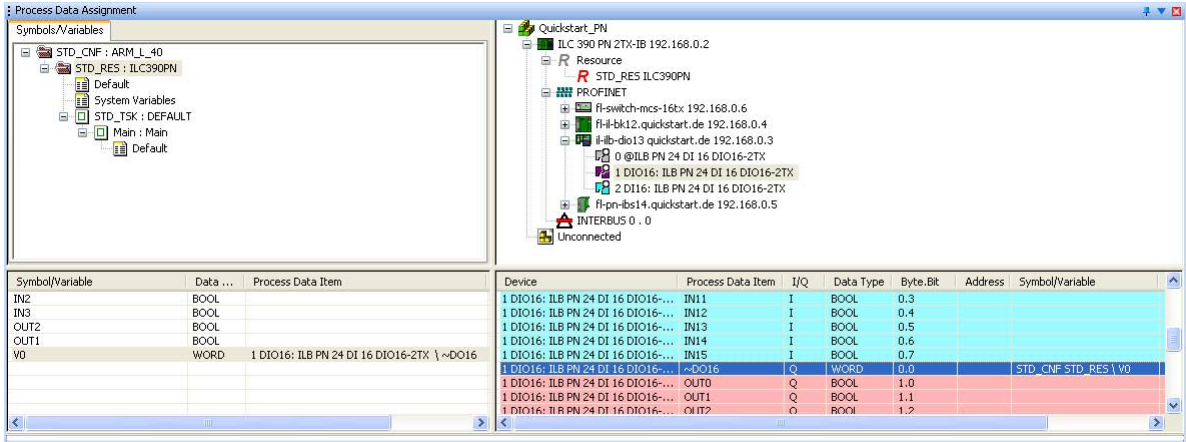


Figure 4-23 Assigning process data for a PROFINET IO device

The result of the process data assignment process is shown in the figure below.

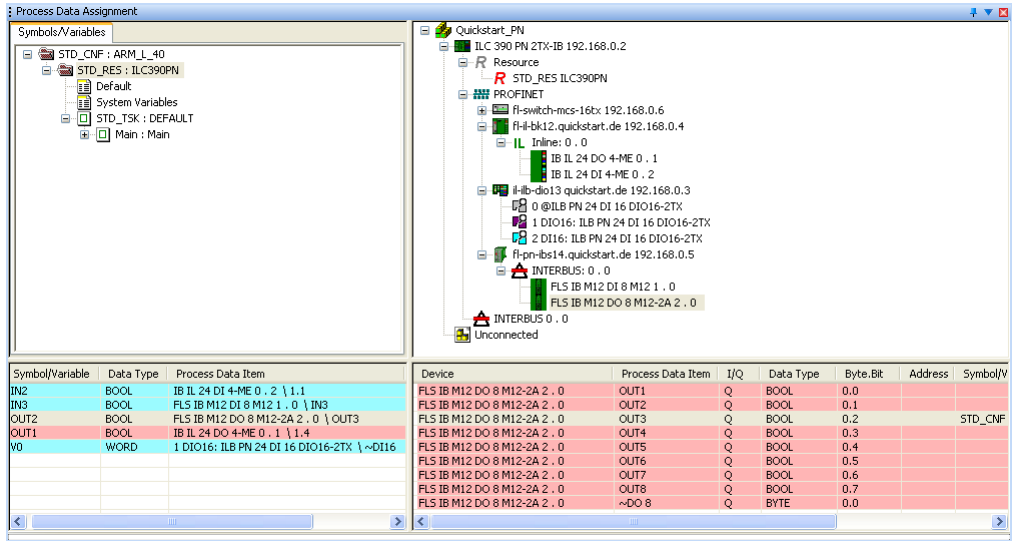


Figure 4-24 All used process data assigned to variables



System variables (e.g., ONBOARD\_INPUT\_BIT10) are not displayed in this process data view.



## 4.14 Setting the communication path: EasySim simulation



At present, this function is only available for processor type "IPC" controllers (RFC 4xx and S-MAX 4xx).

Select the EasySim simulation with this function. The simulation can be used to create a bus and develop and test a program without the hardware having to be available.



For more detailed information about the EasySim simulation, please refer to the online help for PC WorX.



- Switch to the bus configuration workspace to set the simulation (select the communication path).
- In the "Bus Structure" window, select the node for the controller.
- Select the "Communication" tab in the "Device Details" window.
- Select the "Simulation" interface type/communication path.
- Click on "Apply" to activate the simulation.

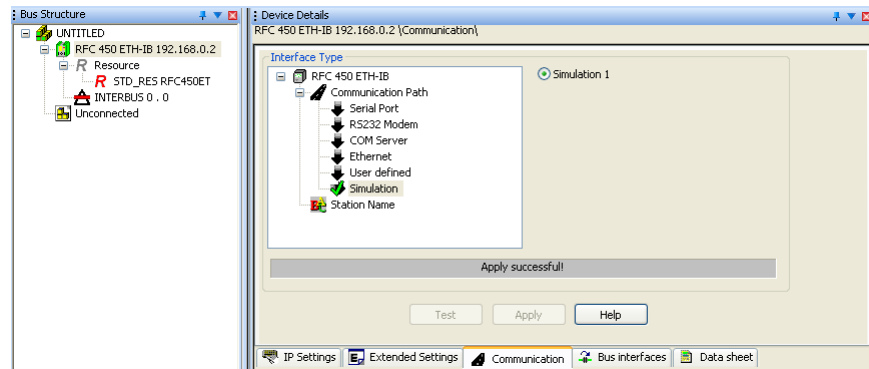


Figure 4-25 "Simulation" communication path

After setting the simulation, the project must be compiled and then written to the memory area of the simulated controller. Please refer to "Compiling and sending a project, and performing a cold restart" on page 4-40.



## 4.15 Switching to working with the system (online)

Apart from during simulation, the following work can only be carried out if a system has been installed and a connection has been established between your PC with PC WorX and the controller.



## 4.16 Assigning the IP address for the controller

An IP address must first be assigned to a controller that communicates via Ethernet in order to enable communication.



For some controllers (e.g., RFC 430 ETH-IB), the IP address can only be set via the diagnostic display or a special firmware service (see user manual for the controller).

For a number of controllers (e.g., ILC 350 ..., ILC 370 ...) BootP is activated by default upon delivery for assigning the IP address via Ethernet. In this case, when setting the IP address for the first time a BootP server can be used.

However, the first IP address can also be set manually using the PC WorX software via the serial interface. The IP address can be changed later via the serial connection or Ethernet using the PC WorX software. Address assignment using PC WorX is described below.



After assigning the IP parameters, PC WorX automatically sets the connection via TCP/IP as the communication path to the controller.

### 4.16.1 Address assignment via the PC WorX BootP server

- Establish an Ethernet connection between your PC and the controller.
- In the menu bar, select the "Extras, BootP/SNMP/TFTP-Configuration ..." menu.

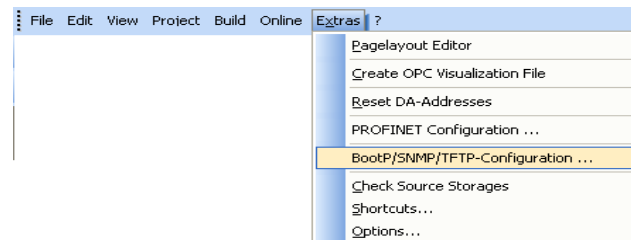


Figure 4-26 Extras, BootP/SNMP/TFTP-Configuration ...

- Activate the "BootP Server active" checkbox.



- Switch to the bus configuration workspace.
- Select the controller node (e.g., "ILC 390 PN 2TX-IB").
- Select the "IP Settings" tab in the "Device Details" window.
- Enter the MAC address of the controller. This can be found on a label on the device. It starts with "00.A0.45".

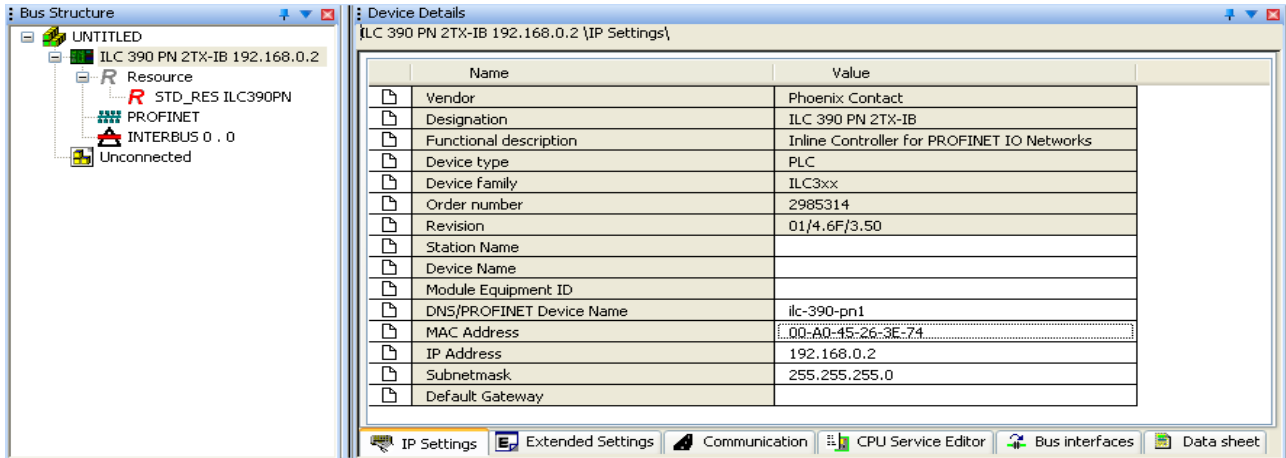


Figure 4-27 Entering the MAC address

- Reset the controller.

The controller is temporarily assigned the IP address which is specified in the project for the controller.

- Select the "Extended Settings" tab in the "Device Details" window.

The specified IP parameters are displayed under "Manual definition of the TCP/IP settings".

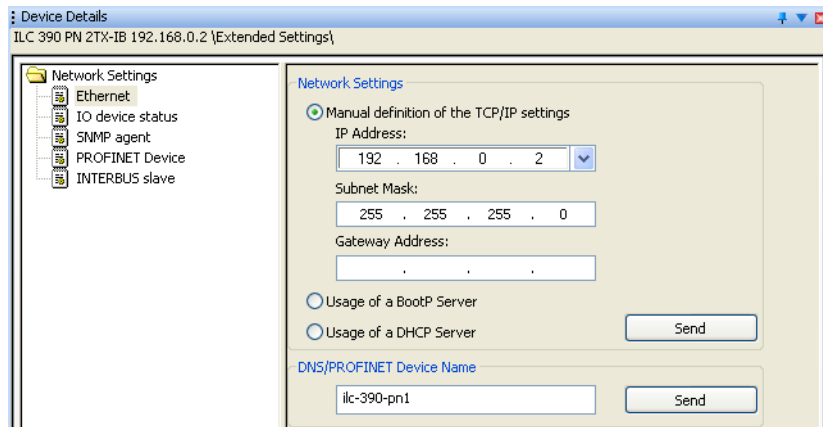


Figure 4-28 IP address of the PROFINET IO controller

- Confirm the displayed IP parameters or your modifications via "Send".

The IP address is now permanently stored on the controller memory card.



## 4.16.2 Address assignment with PC WorX via the serial interface



If the IP parameters were assigned using the BootP server, this section can be skipped.



- Establish a serial connection between your PC and the controller.
- Switch to the bus configuration workspace.
- Select the controller node (e.g., "ILC 390 PN 2TX-IB").
- Select the "Communication" tab in the "Device Details" window.
- Set the communication path (here: serial interface COM1).
- Click on "Test" to check the specified communication path.  
The communication path has been tested successfully if a green status indicator appears in the window. If a red status indicator appears, check the communication path and change it, if necessary.
- If the PROFINET IO controller had an IP address, the IP address of the PROFINET IO controller read via the serial connection is displayed under "Extended Settings".

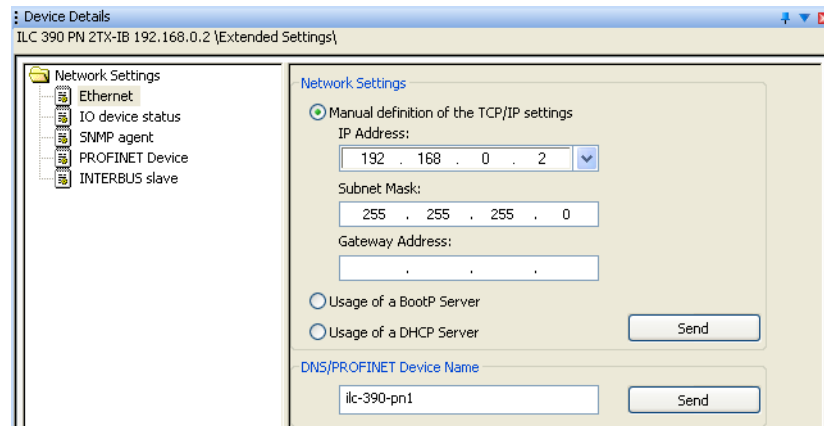


Figure 4-29 Read IP address of the PROFINET IO controller

- If you would like to change the network settings, specify the new settings.
- Confirm the display or your entry with "Send".
- In the "Settings Communication Path" dialog box that opens, set serial interface COM1 and confirm your setting with "OK".
- Reset the controller in the "Activate Network Settings" area via "Reset Control System".



The controller needs some time before it is ready to operate again. It indicates that it is ready via the RUN LED that is either permanently on or flashing cyclically.



## 4.17 Setting the communication path



For a controller that supports Ethernet (e.g., ILC 3xx ETH, ILC 350 PN, ILC 390 PN 2TX-IB, RFC 4xx), the communication path is automatically set to "Ethernet" by default with the IP parameters of the controller specified under "IP Settings". If you would like to send your project to the controller via this Ethernet connection, the communication path does not have to be set.

Since both the INTERBUS and PROFINET project have been created using the example of an ILC 390 PN 2TX-IB, the communication path setting displayed applies to this controller.



- Switch to the bus configuration workspace to set the communication path between the programming PC with PC WorX and the controller.
- In the "Bus Structure" window, select the controller node.
- Select the "Communication" tab in the "Device Details" window.
- Select the communication path to the controller.

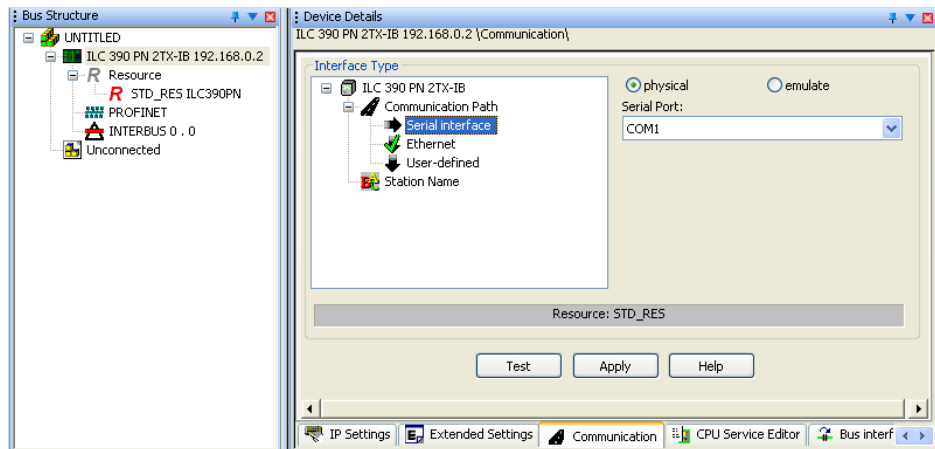


Figure 4-30 Setting the communication path

- Depending on the communication path, proceed as described in Section 4.17.1, "Communication via the serial interface" or Section 4.17.2, "Communication via Ethernet".



BootP is activated by default upon delivery for assigning IP parameters with an Inline Controller via Ethernet.

If you would like to change this setting, select the serial interface as the communication path and specify the IP parameters in the "Extended Settings" dialog box. This setting is activated following a controller reset.



#### 4.17.1 Communication via the serial interface

The controller must be connected to the programming PC via the connecting cable.



Ordering data:

Connecting cable for connecting the controller to a PC (V.24 (RS-232)) for PC WorX, 3 m in length (order designation: PRG CAB MINI DIN, Order No. 2730611).

- In the "Device Details" window, select the "Serial Port" interface type in the "Communication" tab.
- Select the interface (e.g., COM1).
- Click on "Test" to check the specified communication path.  
The communication path has been tested successfully if a green status indicator appears in the window. If a red status indicator appears, check the communication path and change it, if necessary.
- Save your settings in the project via "Apply".

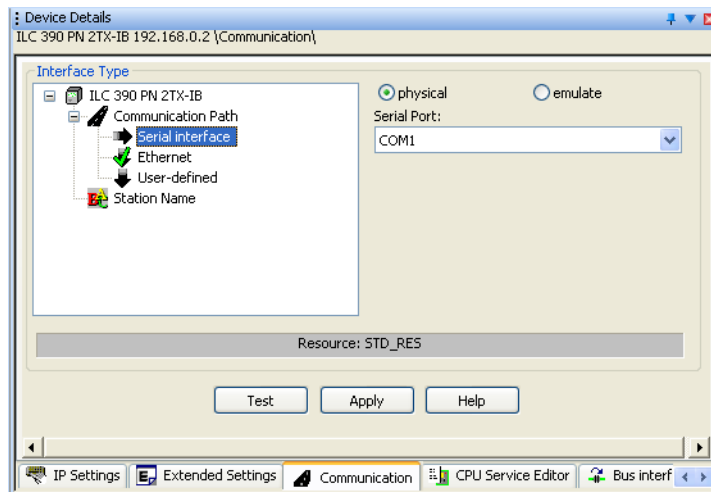


Figure 4-31 "Serial interface" communication path



## 4.17.2 Communication via Ethernet

A physical Ethernet connection to the controller is required for communication via Ethernet.

Furthermore, the IP address must also be set in the controller.

- In the "Device Details" window, select the "Ethernet" interface type in the "Communication" tab.
- Enter the IP address set in the Inline Controller in the "IP Address" field or select it from the menu.

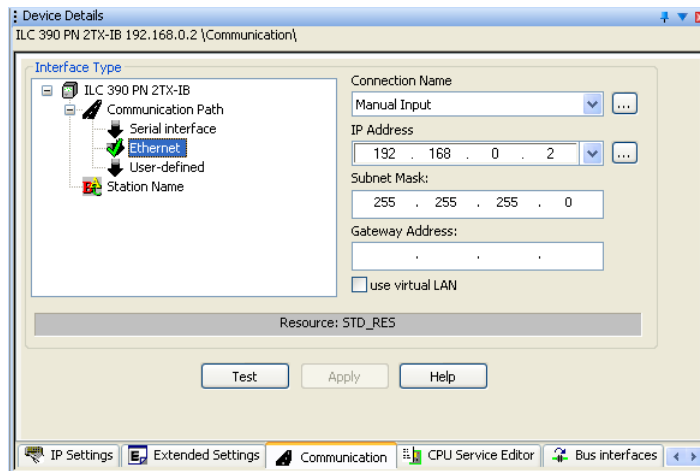


Figure 4-32 "Ethernet" communication path

- Click on "Test" to check the specified communication path and therefore the IP address entered.  
The communication path has been tested successfully if a green status indicator appears in the window. If a red status indicator appears, check the communication path and settings and change them, if necessary.  
If a red status indicator appears, an IP address may not have been assigned or the IP address needs to be changed. To assign and change the TCP/IP settings, please proceed as described in "Assigning the IP address for the controller" on page 4-23 (general), "Assigning the IP address for the controller" on page 5-5 (for INTERBUS) or "Assigning the IP address for the PROFINET IOcontroller" on page 6-7 (for PROFINET).
- Click on "Apply" to save your settings in the project.



**NOTE:** The Inline Controller can be accessed via a network using the Ethernet interface and the TCP/IP protocol. Please note that with all networked devices/computers, there is a risk that third parties may access the Inline Controller or the PC WorX computer and make changes, either intentionally or unintentionally. Prevent unauthorized access to the Inline Controller and your PC.



## 4.18 Reading in and importing PROFINET IO devices

If you have inserted the PROFINET IO devices manually, skip this section.

### 4.18.1 Reading in PROFINET IO devices



- Make sure you are in the bus configuration workspace.
- In the "Bus Structure" window, select the PROFINET node.
- Open the context menu and select the "Read PROFINET..." menu item.

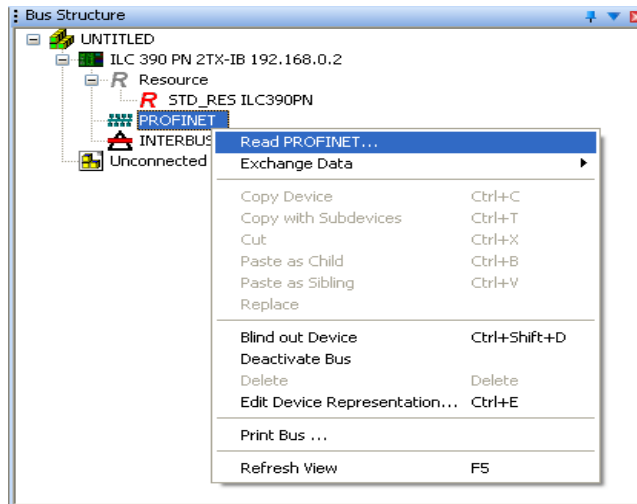


Figure 4-33 Reading in PROFINET

All the connected PROFINET IO devices are displayed. When reading in, the device name and the IP parameters are imported for each device - if present.

- Select the devices that are to be included in your project.



If your PROFINET IO controller already contains a project, a different name can be specified for the controller in the "Read PROFINET" window from the one assigned in the current project. If you are sure you are communicating with the right controller this will not have any adverse effects. The PROFINET IO devices displayed are connected to the controller - regardless of its name. The current name is also transmitted when the current project is sent to the controller.

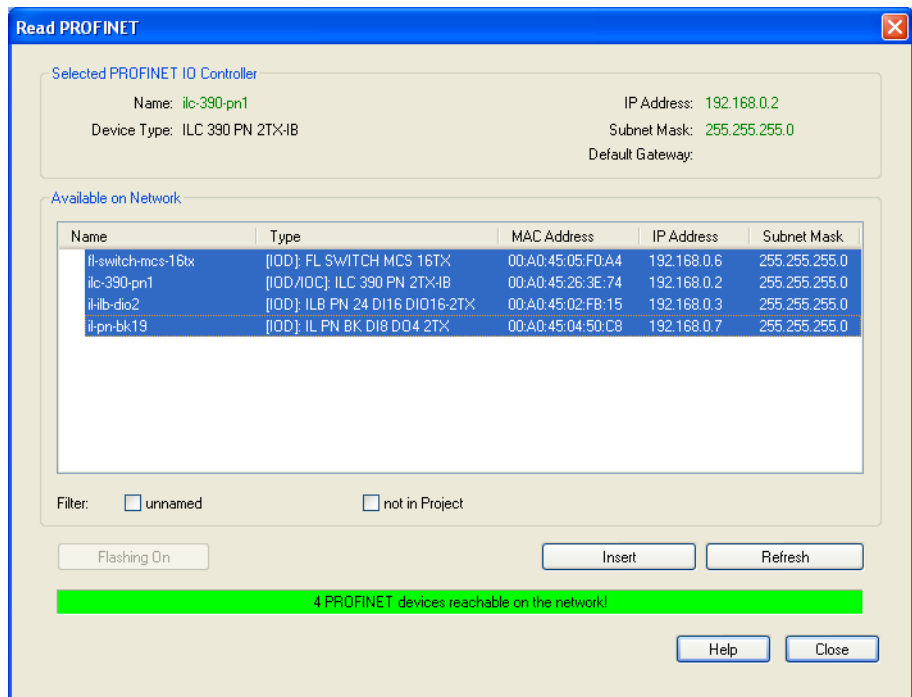


Figure 4-34 Connected and selected PROFINET IO devices for the project; PROFINET device names and IP addresses are already present

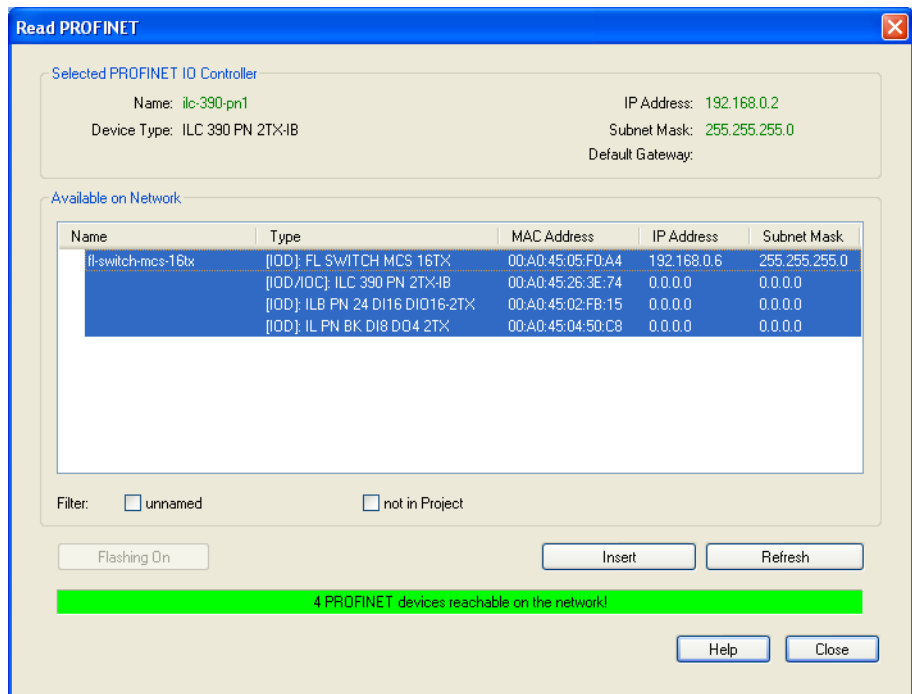


Figure 4-35 Connected and selected PROFINET IO devices for the project; without PROFINET device names and IP addresses

## 4.18.2 Importing PROFINET IO devices into the project

- Click on "Insert" to include the selected PROFINET IO devices in your project.

### 4.18.3 Naming PROFINET IO devices without a PROFINET device name

If a PROFINET device name does not yet exist for a device, a name should be assigned at this point.

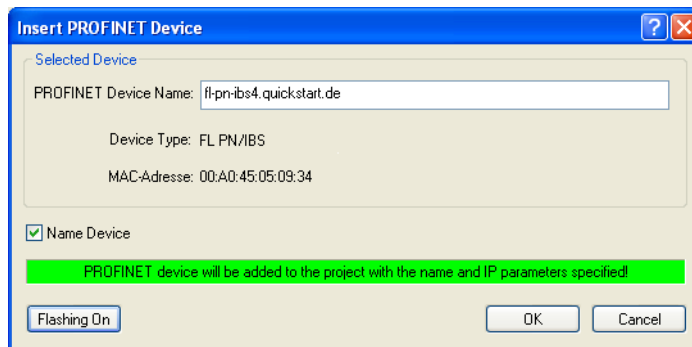


Figure 4-36 Naming a device

- Activate the "Name Device" checkbox.
- Check the displayed PROFINET device name (see also "DNS/PROFINET Device Name" on page 4-10).
- Assign the name to the relevant device.
- Confirm the display or your entry with "OK".

If you do not wish to assign a name at this point, deactivate the "Name Device" checkbox. A name will have to be assigned later to the relevant device (here: FL PN/IBS).

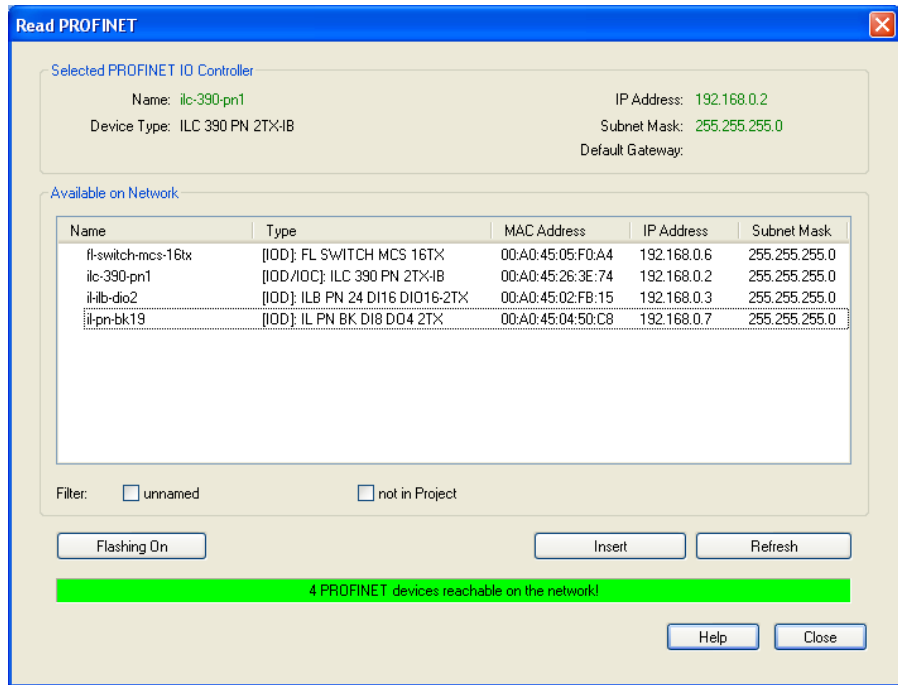


Figure 4-37 Devices named

- Click "Close" to close the "Read PROFINET" dialog box.

#### 4.18.4 Bus configuration with read in PROFINET IO devices

The PROFINET structure is displayed in the "Bus Structure" window.

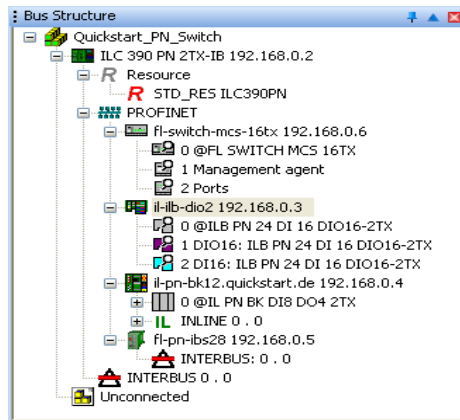


Figure 4-38 Bus configuration with read in PROFINET IO devices

The settings can also be checked in the "Read PROFINET" window. All PROFINET IO devices should now have a PROFINET device name. The IP parameters are assigned later.



### 4.18.5 Subsequent naming of a PROFINET IO device

If a device is displayed in the list without a device name, assign the PROFINET device name to this device.

- Select the device in the bus configuration.
- In the "Device Details" window, switch to the "PROFINET Stationnames" tab.
- Select the device with no name.

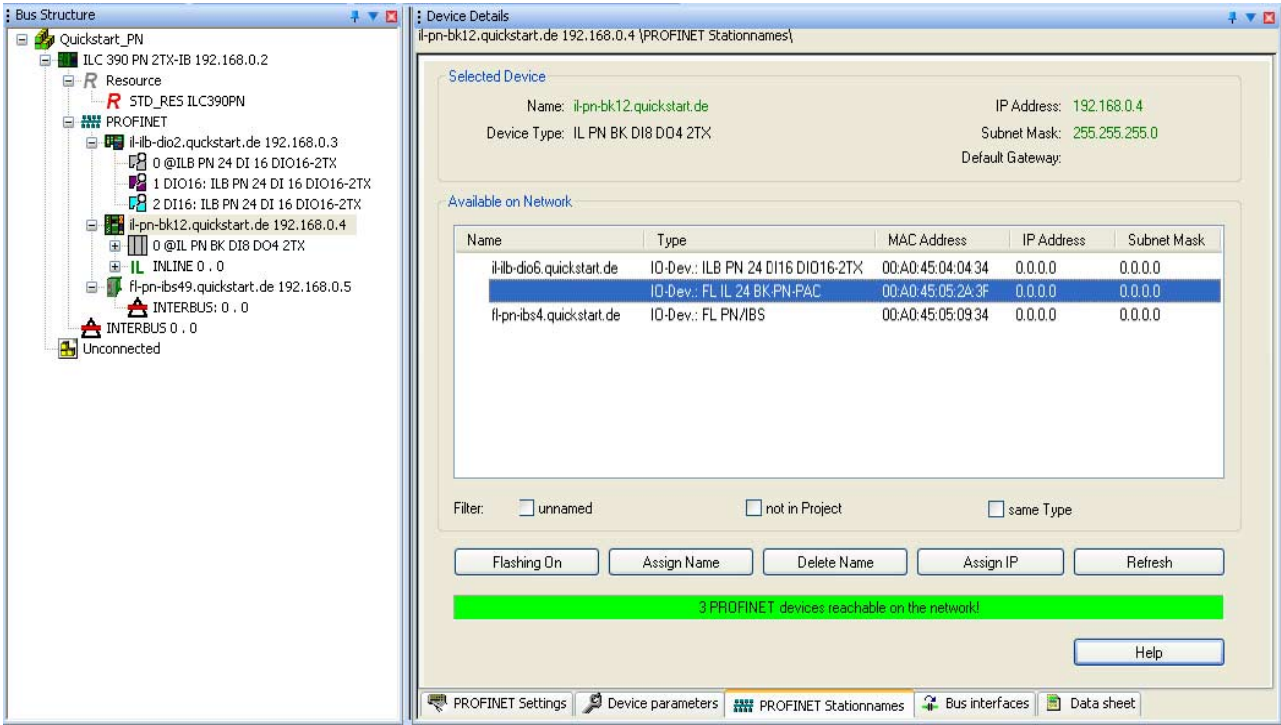


Figure 4-39 Assigning the PROFINET device name

- Click on "Assign Name".

The updated list contains the PROFINET IO devices together with their PROFINET device names.

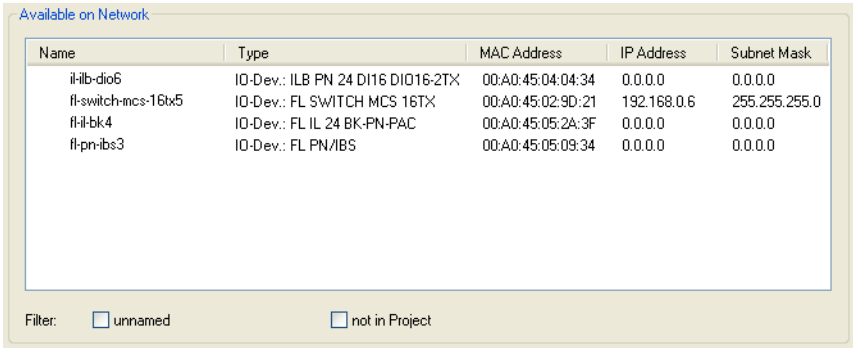


Figure 4-40 Example: All devices with PROFINET device names



## 4.19 Checking/modifying the PROFINET settings for PROFINET IO devices

If you have made the PROFINET settings for the PROFINET IO devices offline after inserting the PROFINET IO devices, skip this section.

When **reading in**, the PROFINET settings of the relevant PROFINET IO devices are imported with default values. Check these settings and modify them, if necessary.



- Make sure you are in the bus configuration workspace.
- In the "Bus Structure" window, select the PROFINET IO device.
- Under "Device Details", select the "PROFINET Settings" tab.
- Check the PROFINET settings and modify them, if necessary.

Name	Value
Vendor	Phoenix Contact
VendorID	0x00B0
Designation	ILB PN 24 DI16 DIO16-2TX
DeviceID	0x0004
Functional description	PROFINET IO - Inline Block Device With 16 Digit...
Device type	I/O digital
Device family	ILB
Order number	2878146
Revision	03/1.00
DNS/PROFINET Device Name	il-hb-dio3.quickstart.de
Station Name	
Device Name	
Module Equipment ID	
IP Address	192.168.0.4
Subnetmask	255.255.255.0
Default Gateway	
Realtime class	RT
Reduction ratio input	8 ms
Reduction ratio output	8 ms
Faulty telegrams until connection is aborted	24
Monitoring Time Inputs (ms)	192
Monitoring Time Outputs (ms)	192
Operation in case of configuration differences	no
Log connection state	yes
Drive BF	yes
Ersatzwertverhalten Eingänge	auf Null setzen
Verbindungsaufbau Anlauf	yes
Node ID	41

Figure 4-41 PROFINET settings for the ILB PN 24 DI16 DIO16-2TX

The individual setting options are described in "Checking/modifying the PROFINET settings for PROFINET IO devices" on page 4-10.



## 4.20 Transferring PROFINET device names and IP settings to PROFINET IO devices

Before a PROFINET IO device can be operated in a PROFINET IO network, the PROFINET device name and IP address configured in PC WorX must also be made known to the device itself.

- Make sure that the PROFINET IO controller can establish communication with the PROFINET IO devices.
- Compile the project, send it to the controller, and perform a cold restart. Proceed as described in "Compiling and sending a project, and performing a cold restart" on page 4-40.

During startup, the PROFINET controller automatically assigns the IP settings and device parameterizations specified in the project to the PROFINET IO devices.

- To check the assignment of device names and IP settings, select the "PROFINET Stationnames" tab in the "Device Details" window.

The updated list contains the PROFINET IO devices together with their PROFINET device names and the IP parameters.

Name	Type	MAC Address	IP Address	Subnet Mask
fl-switch-mcs-16tx	[I0D]: FL SWITCH MCS 16TX	00:A0:45:05:F0:A4	192.168.0.6	255.255.255.0
ilc-390-pn1	[I0D/I0C]: ILC 390 PN 2TX/IB	00:A0:45:26:3E:74	192.168.0.2	255.255.255.0
ilb-dio2	[I0D]: ILB PN 24 DI16 DIO16-2TX	00:A0:45:02:FB:15	192.168.0.3	255.255.255.0
il-pn-bk12.quickstart.de	[I0D]: IL PN BK DI8 DO4 2TX	00:A0:45:04:50:C8	192.168.0.4	255.255.255.0

Filter:  unnamed  not in Project  same Type

Figure 4-42 All devices with PROFINET device names and IP addresses

When the tab is selected, the list is updated. If you would like to update the list again later, click on "Refresh".

In order to limit the search results, various options are available:

No constraints	All devices that are available in the network are listed.
Unnamed	All those devices that do not yet have a PROFINET device name are listed.
Not in project	All those devices that are not included in the project are listed.
Same type	Only those devices that are the same type as the device selected in the bus configuration are listed.

The BF LED is now off on all PROFINET IO devices.  
The PROFINET network is running.



For additional information about assigning PROFINET device names and IP addresses, please also refer to "Additional information" on page 6-15.



## 4.21 Reading in INTERBUS

If you have inserted the INTERBUS devices manually, skip this section.

The bus configuration must actually be available and power must be supplied to the modules.

- Select the "Connected Bus" command from the "View" menu to read in the connected INTERBUS system.

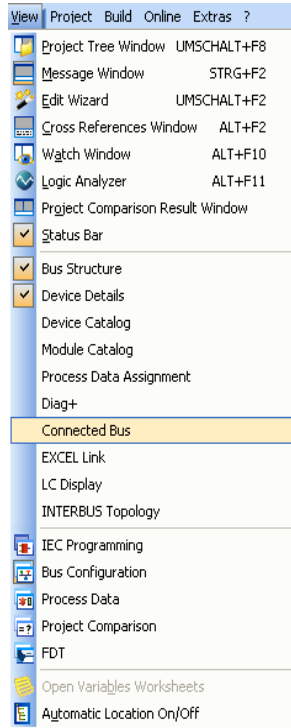


Figure 4-43 "View, Connected Bus" command

- Select the controller in the "Connected Bus" window.



For PROFINET, the PROFINET IO devices can also be selected as the controller (FL IL 24 BK-PN-PAC in Figure 4-44).

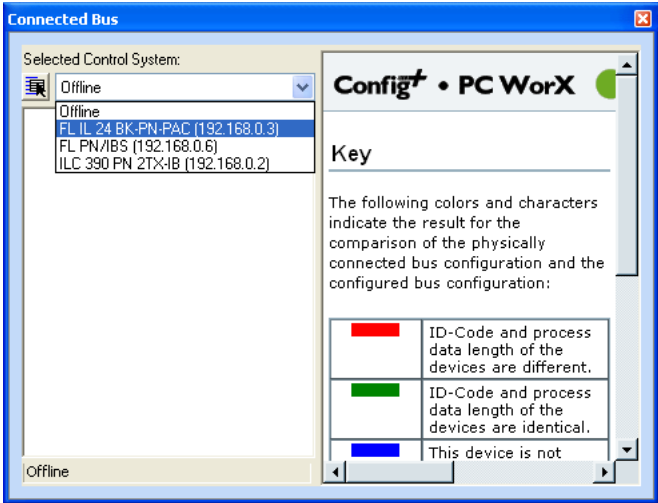


Figure 4-44 Selecting the controller

This activates the "online" operating state and the controller reads in the connected INTERBUS configuration.

Once the controller has read in the connected INTERBUS system, the INTERBUS devices must be imported into the project.

- Select the controller in the "Connected Bus" window.
- Open the context menu and select the "Import to Project, With Device Description" command.

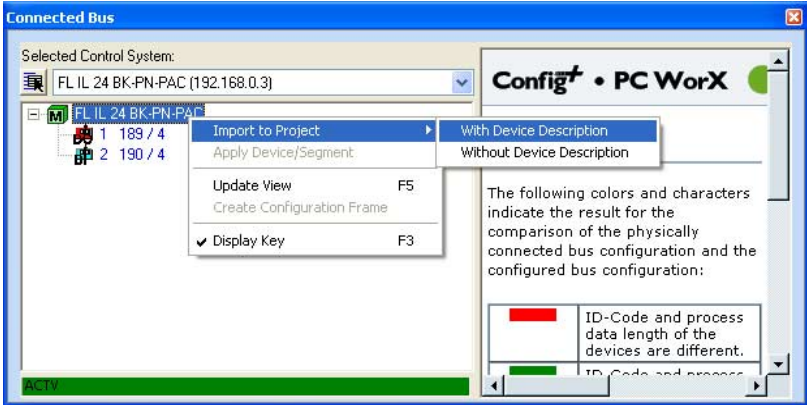


Figure 4-45 "Connected Bus" window

The "Select Device" window that opens lists the modules whose device description corresponds to that of the connected devices.

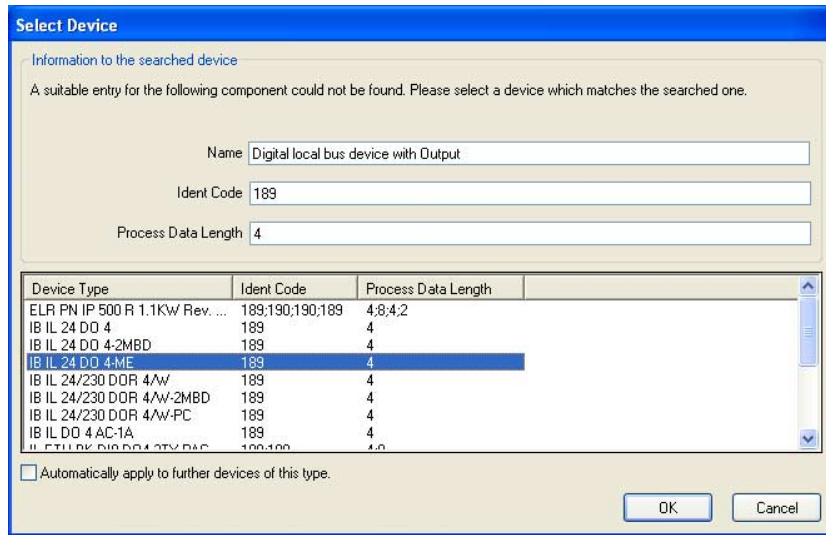


Figure 4-46 "Select Device" window

- Select the device that is actually connected in the INTERBUS system, and repeat this step until all the devices are linked to their device description.
- Disconnect the link to the controller by selecting "Offline" under "Selected Control System" in the "Connected Bus" window.

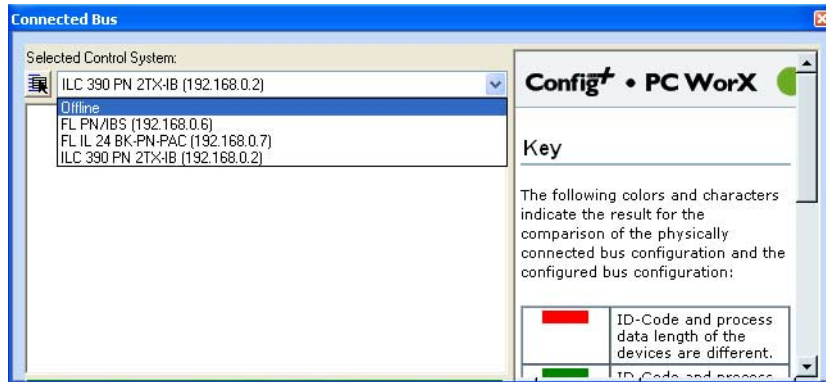


Figure 4-47 "Connected Bus" window



**Please note for PROFINET:**

The process for reading in INTERBUS modules that are connected to a PROFINET IO controller or a PROFINET IO device is essentially the same.

**PROFINET IO controller**

After the communication path to the PROFINET IO controller has been set, the INTERBUS devices connected to the controller can be read in.

**PROFINET IO device**

After device names and IP parameters for the PROFINET IO devices have been assigned, the PROFINET IO devices connected to the INTERBUS devices can be read in.

- Select all the "controllers" that are connected to the INTERBUS devices. These include the Inline Controller as well as all the PROFINET IO devices displayed in the "Connected Bus" window.
- Read in the INTERBUS devices as described above.



The product designation or the station name is displayed under "Selected Control System". If several devices of the same type are used in a project, first assign the devices a station name. This enables the unique identification of the devices.



## 4.22 Compiling after reading in the bus topology

At this point you can compile your project in order to detect any errors that may have occurred.



- Select the "Build, Make" command.



## 4.23 Creating the program

If the program has been created offline, skip this section.

To program the example program, proceed as described in "Example program" on page 8-1.



## 4.24 Compiling after creating the program

At this point you can compile your project in order to detect any errors that may have occurred.



- Select the "Build, Make" command.



## 4.25 Assigning process data

If the process data has been assigned offline, skip this section.

To assign the process data, proceed as described in "Generating variables and assigning process data" on page 4-18.



## 4.26 Compiling and sending a project, and performing a cold restart



### 4.26.1 Compiling a project

- To compile a project, proceed as described in "Compiling after completing the bus topology" on page 4-16.



- Select the "Build, Make" command.



### 4.26.2 Sending a project

When working online, the project is written to the main memory of the controller. When working in simulation mode, the project is written to the simulated main memory of the controller.



- Open the project control dialog box.



If only the "Close" button is enabled in the project control dialog box that opens, there is no valid connection to the controller. In this case, check the communication path.

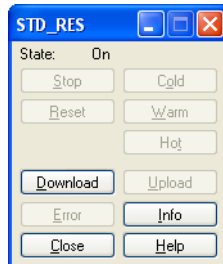


Figure 4-48 Project control dialog box

- Click on "Download".

Another dialog box opens. There are various options for sending the project to the controller or the simulated controller.

- Click on "Download" in the "Project" area.
- Click on "Download" in the "Bootproject" area.
- Activate the "Include Bootproject" checkbox in the "Project" area and click on "Download" in this area.

In the first option, the project is sent. However, it is not loaded automatically when booting the controller.

For option 2 and 3, the project is written as a boot project to the parameterization memory of the controller and is loaded automatically after the controller has been booted.

Option 3 is shown in Figure 4-49 on page 4-41.



- Activate the "Include Bootproject" checkbox in the "Project" area and click on "Download" in this area.

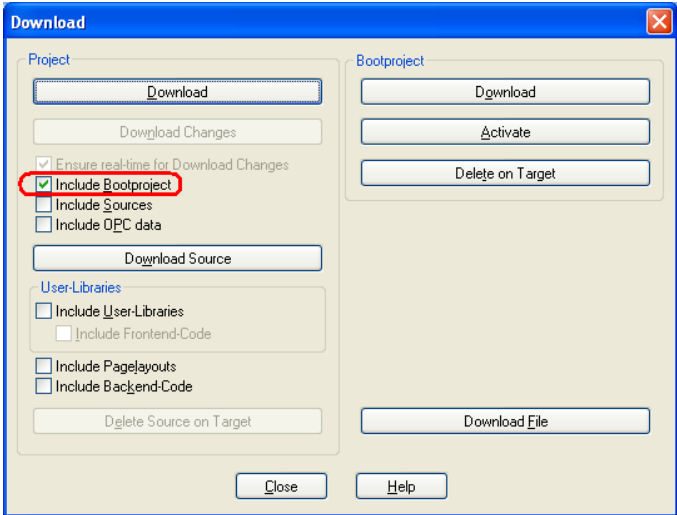


Figure 4-49 "Download" dialog box

For controllers that support the "Download Changes" function, the "Download" dialog box appears as shown in Figure 4-50. If the "Download Changes" function is used, activate the "Ensure real-time for Download Changes" and "Include Bootproject" checkboxes.

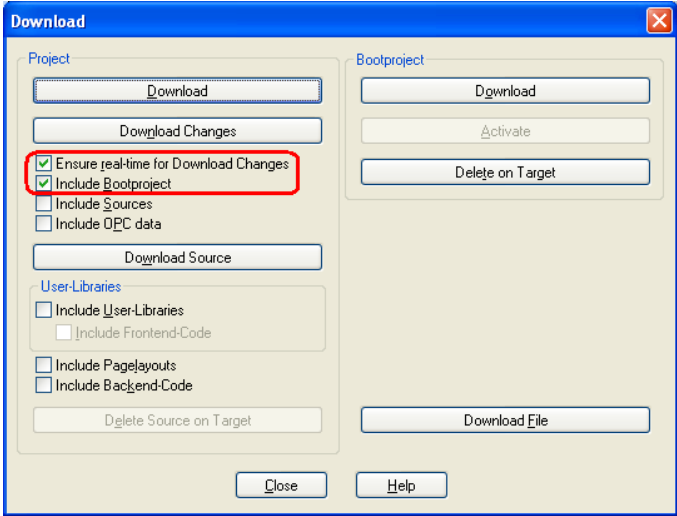


Figure 4-50 "Download" dialog box

The project is now in the real or simulated main memory of the controller.



**4.26.2.1 Sending a project: Special notes for PROFINET**

When sending the project to the controller, the desired PROFINET IO configuration is transmitted and activated. If activation was successful, the PNIO\_CONFIG\_STATUS\_ACTIVE system variable is set.

The PROFINET IO controller attempts to start up all the devices in the desired configuration.



**4.26.2.2 Sending a project: Special notes for the EasySim simulation**

The EasySim simulation is started when the project control dialog box is opened. This is shown in the taskbar with the following icon:

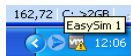


Figure 4-51 EasySim simulation: Icon in the taskbar

The following options are available for the simulation in the context menu:

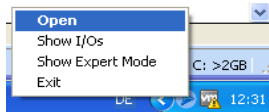


Figure 4-52 EasySim simulation: Context menu

Selecting the "Open" menu item opens the simulation window.



Figure 4-53 EasySim simulation opened

The "Show I/Os" menu item or "I/Os" button opens EasySim input and output view.

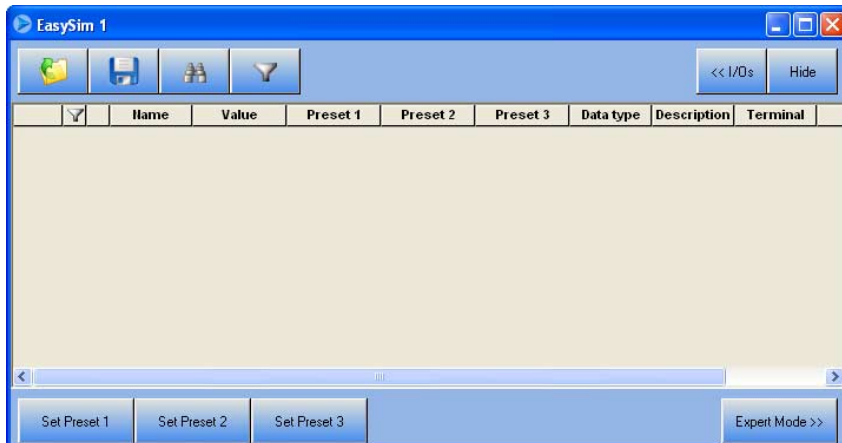


Figure 4-54 EasySim simulation: I/O view

The "Show Expert Mode" menu item or "Expert Mode" button opens expert mode for the simulation. This enables periods to be shown according to the simulated input and output states.



### 4.26.3 Performing a cold restart

In order to activate the project, it is necessary to perform a cold restart.

- To do so, click on "Cold" in the project control dialog box.

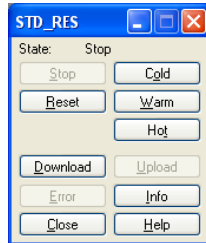


Figure 4-55 Project control dialog box

If the system cannot be started up, a corresponding error message appears on the diagnostic display (for controllers with diagnostic display) and in the "Message Window" in PC WorX and Diag+.



For an explanation of the error message, please refer to the Diagnostics Guide (IBS SYS DIAG DSC UM E, Order No. 2747280).

For example, the ILC 350 ETH controller has been started up successfully if the green PLC RUN LED is on.

Communication to the PROFINET IO devices has been established successfully if the BF LED does not flash on any of the PROFINET IO devices.

**Simulation following cold restart**

When sending, the project is written to the main memory of the simulated controller. Following a successful cold restart, you can simulate the behavior of your application.



Only the states of the global variables that have been linked to a physical address (inputs/outputs) as part of process data assignment are displayed in the simulation.

Figure 4-56 below shows expert mode for the simulation of an example project, in which a run sequence is illustrated for various I/O states in the area at the bottom.

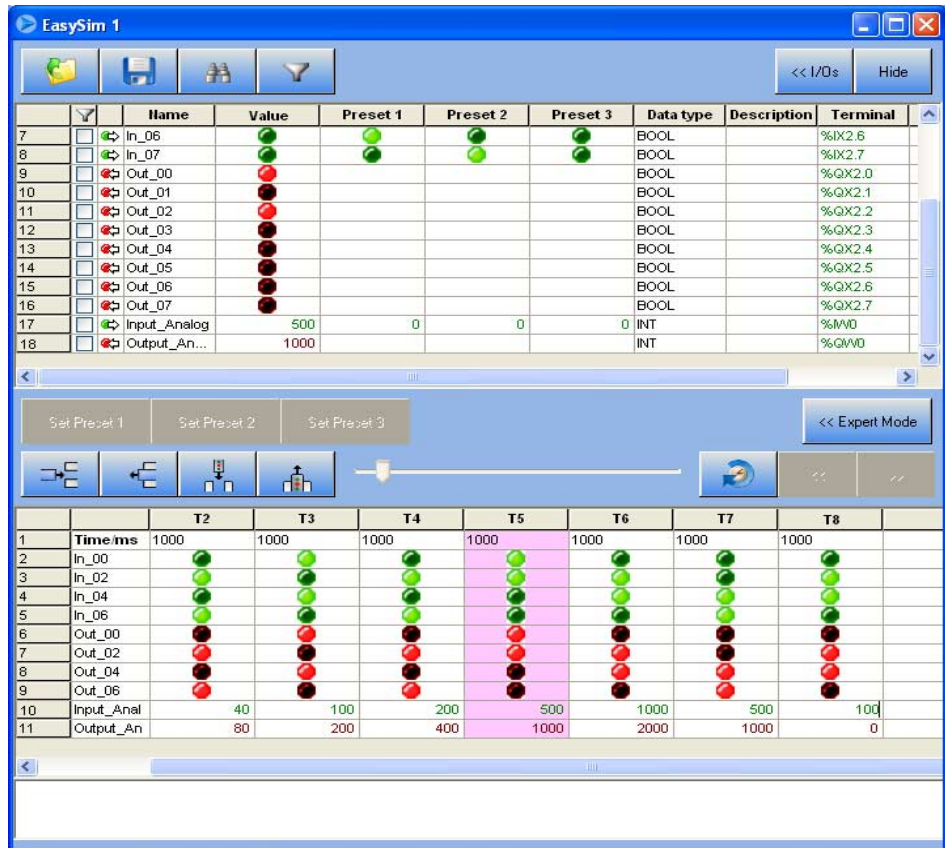


Figure 4-56 EasySim simulation: I/O view - Expert mode



## 4.27 Operation



### 4.27.1 Setting the task properties

If no settings have been made for the task, a default task is used. If necessary, change the properties of the task (in the example below a default task is changed to a cyclic task).



- Switch to the IEC programming workspace.
- Select "STD\_TSK : DEFAULT".
- In the context menu, select the "Properties..." menu item.

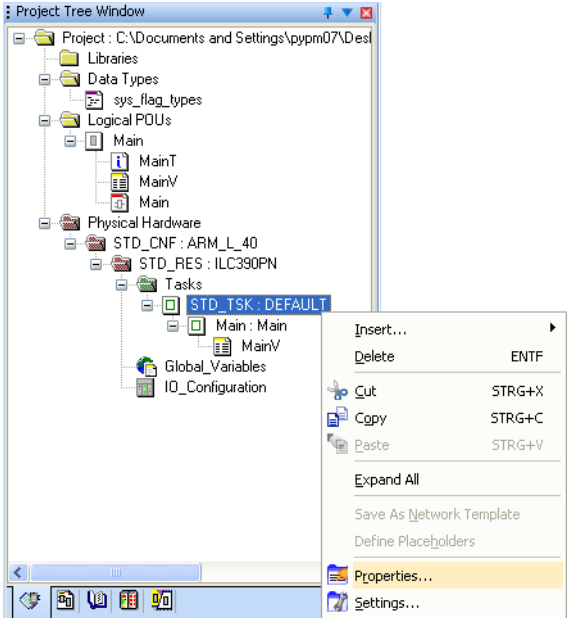


Figure 4-57 Standard task: Properties

- Under type, select "Cyclic". This selects a cyclic task.

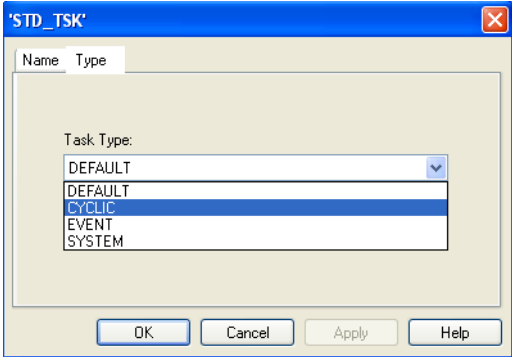


Figure 4-58 Standard task: "Cyclic" type

- Apply your setting with "Apply".
- In the standard task context menu, select the "Settings..." menu item (see Figure 4-57 on page 4-45).
- Change the watchdog time setting, e.g., to 1000 ms.

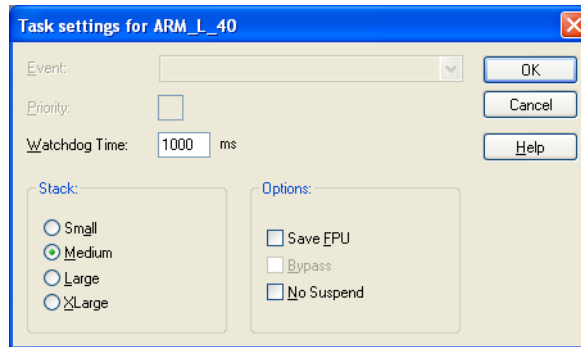


Figure 4-59 Changing the setting

- Confirm your settings with "OK".
- Compile the project, send it to the controller, and perform a cold restart.

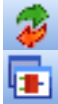
If the process is now running too slowly, change the settings.

- In the standard task context menu, select the "Settings..." menu item.
- Change the watchdog time setting, e.g., to 250 ms.
- Compile the project, send it to the controller, and perform a cold restart.



### 4.27.2 Debug mode

In order to use debug mode, a program must have been created in your project. The method of operation of the program can be monitored in debug mode.



- Activate debug mode.
- Switch to the IEC programming workspace.

The status of all global variables used in the program is displayed in the "MainV" variables worksheet in the "Project Tree Window".

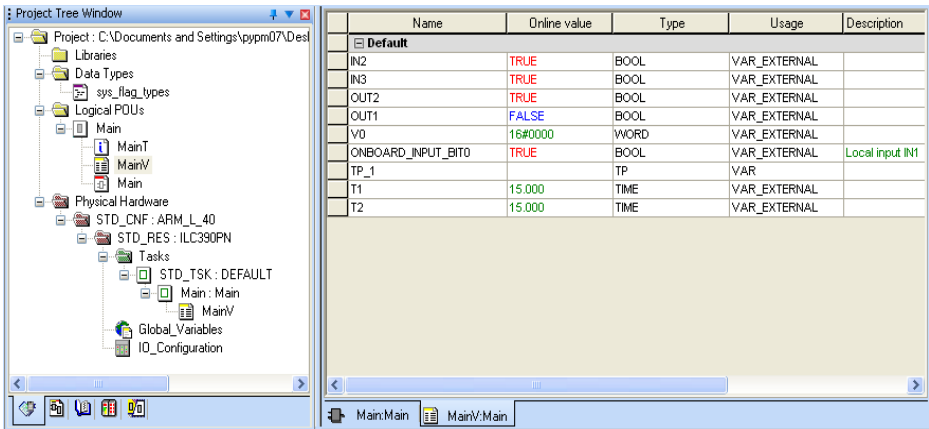


Figure 4-60 Variables in debug mode: Global variables (VAR\_EXTERNAL)



For PROFINET IO devices, the PNIO\_DATA\_VALID system variable can be assigned, for example. You can read the status here. This bit must be set for the PROFINET IO device to supply valid data and all other process values to be valid. In this way, you can check in this window whether all PROFINET IO devices supply valid data. For information about system variables, please refer to Appendix B "Status information for a PROFINET IO system".

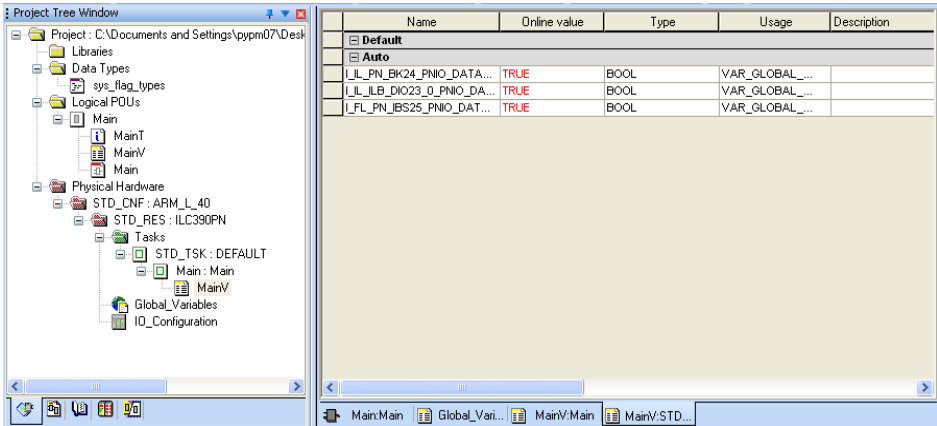


Figure 4-61 Variables in debug mode: System variables of PROFINET IO devices

The program status of a POU is displayed by opening the program worksheet (in Figure 4-62 under "Main" in the "Project Tree Window" or under "Main:Main" on the tab in the workspace).

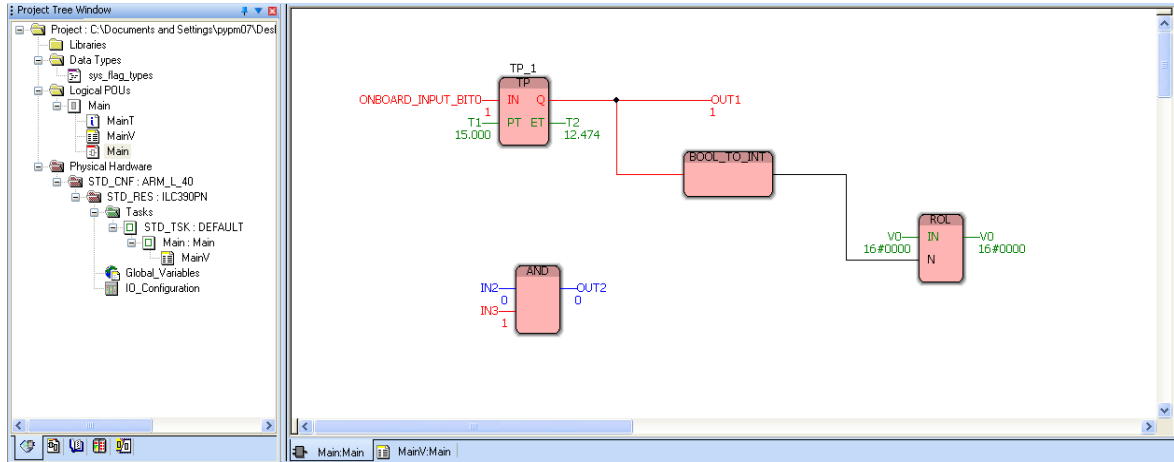


Figure 4-62 Variables in debug mode: Program variables

The status of all global variables is displayed under "Global Variables" in the "Project Tree Window".

Name	Online value	Type	Usage	Description
<b>Default</b>				
IN2	TRUE	BOOL	VAR_GLOBAL	
IN3	FALSE	BOOL	VAR_GLOBAL	
OUT2	FALSE	BOOL	VAR_GLOBAL	
OUT1	TRUE	BOOL	VAR_GLOBAL	
OUT11	TRUE	BOOL	VAR_GLOBAL	
<b>System Variables</b>				
PLCMODE_ON	FALSE	BOOL	VAR_GLOBAL	PLC status ON
PLCMODE_RUN	TRUE	BOOL	VAR_GLOBAL	PLC status R...
PLCMODE_STOP	FALSE	BOOL	VAR_GLOBAL	PLC status S...
PLCMODE_HALT	FALSE	BOOL	VAR_GLOBAL	PLC status H...
PLCDEBUG_BPSET	FALSE	BOOL	VAR_GLOBAL	Breakpoint set
PLCDEBUG_FORCE	FALSE	BOOL	VAR_GLOBAL	Variable(s) f...
PLCDEBUG_POWERFLOW	FALSE	BOOL	VAR_GLOBAL	Powerflow ON
PLC_TICKS_PER_SEC	1000	INT	VAR_GLOBAL	System ticks
PLC_SYS_TICK_CNT	160193910	DINT	VAR_GLOBAL	Number of s...
PLC_TASK_AVAILABLE	16	INT	VAR_GLOBAL	Number of a...
PLC_TASK_DEFINED	1	INT	VAR_GLOBAL	Number of ta...
PLC_TASK_1		Ext...	VAR_GLOBAL	Task 1 infor...
PLC_TASK_2		Ext...	VAR_GLOBAL	Task 2 infor...
PLC_TASK_3		Ext...	VAR_GLOBAL	Task 3 infor...
PLC_TASK_4		Ext...	VAR_GLOBAL	Task 4 infor...
PLC_TASK_5		Ext...	VAR_GLOBAL	Task 5 infor...
PLC_TASK_6		Ext...	VAR_GLOBAL	Task 6 infor...

Figure 4-63 Variables in debug mode: Global variables

- Switch to the window that provides the best display of the required information.



### 4.27.3 PLC stop/run

If the PLC is set to STOP, all outputs are set to their safe state. When the controller is started, process values are output again.





## 4.28 Switching from simulation to real hardware

Once you have successfully completed the simulation for your program, you can switch to your hardware. To do this, the actual bus configuration must correspond to the simulated bus configuration.

- Stop debug mode. To do this, click on "Debug on/off".
- Stop the simulation. To do so, click on "Stop" in the project control dialog box.
- Close the project control dialog box.
- Exit the simulation, e.g., via the "Exit" menu item in the context menu for the simulation icon in the taskbar.



- Switch to the bus configuration workspace.
- In the "Device Details" window, switch to the "Communication" tab.
- Select the required communication path to enable you to communicate between PC WorX and your hardware (see "Setting the communication path" on page 4-26).
- Send the project to the controller and perform a cold restart (see "Compiling and sending a project, and performing a cold restart" on page 4-40).



## 5 Example project for an INTERBUS system

The creation of a project is described in detail in Section 4, "Sequence for creating a project".

This section covers:

- All steps to be taken in reference to the corresponding section in Section 4, "Sequence for creating a project"
- Project-specific settings
- Information and special notes for an INTERBUS project

### 5.1 Project description

In the following, an example project is developed using function block diagram (FBD).



In order to obtain the best possible results, please use the same identifiers and names as used in this manual.

**Project name**

Quickstart

**Project hardware**

- Controller: ILC 390 PN 2TX-IB Inline Controller
- I/O modules: IB IL 24 DO 16, IB IL 24 DO 8, and IB IL 24 DI 8

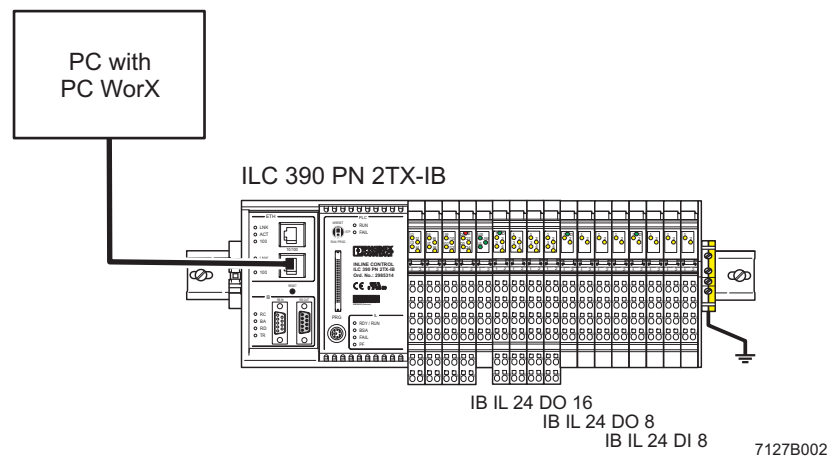


Figure 5-1 Example bus configuration

**Inputs and outputs/process data**

Inputs and outputs used in the example including process data and assigned variables.

Device	Inputs/outputs according to the device data sheet	Signal at	Variable	Process data item
ILC 390 PN 2TX-IB	Input IN11	Co 3 TP 1.4	ONBOARD_INPUT_BIT10	3.1.4
IB IL 24 DI 8	Input IN2	Co 1 TP 2.1	IN2	1.2.1
	Input IN5	Co 3 TP 1.1	IN3	3.1.1
IB IL 24 DO 8	Output OUT1	Co 1 TP 1.1	OUT1	1.1.1
	Output OUT6	Co 3 TP 2.1	OUT2	3.2.1
IB IL 24 DO 16	Outputs OUT1 to OUT16	Co 1 to 4	V0	~DO16
		TP x.1 and x.4		
No assignment required			T1	
			T2	

- Co Connector
- TP Terminal point
- x Row on the Inline connector

## 5.2 Sequence for creating the INTERBUS project

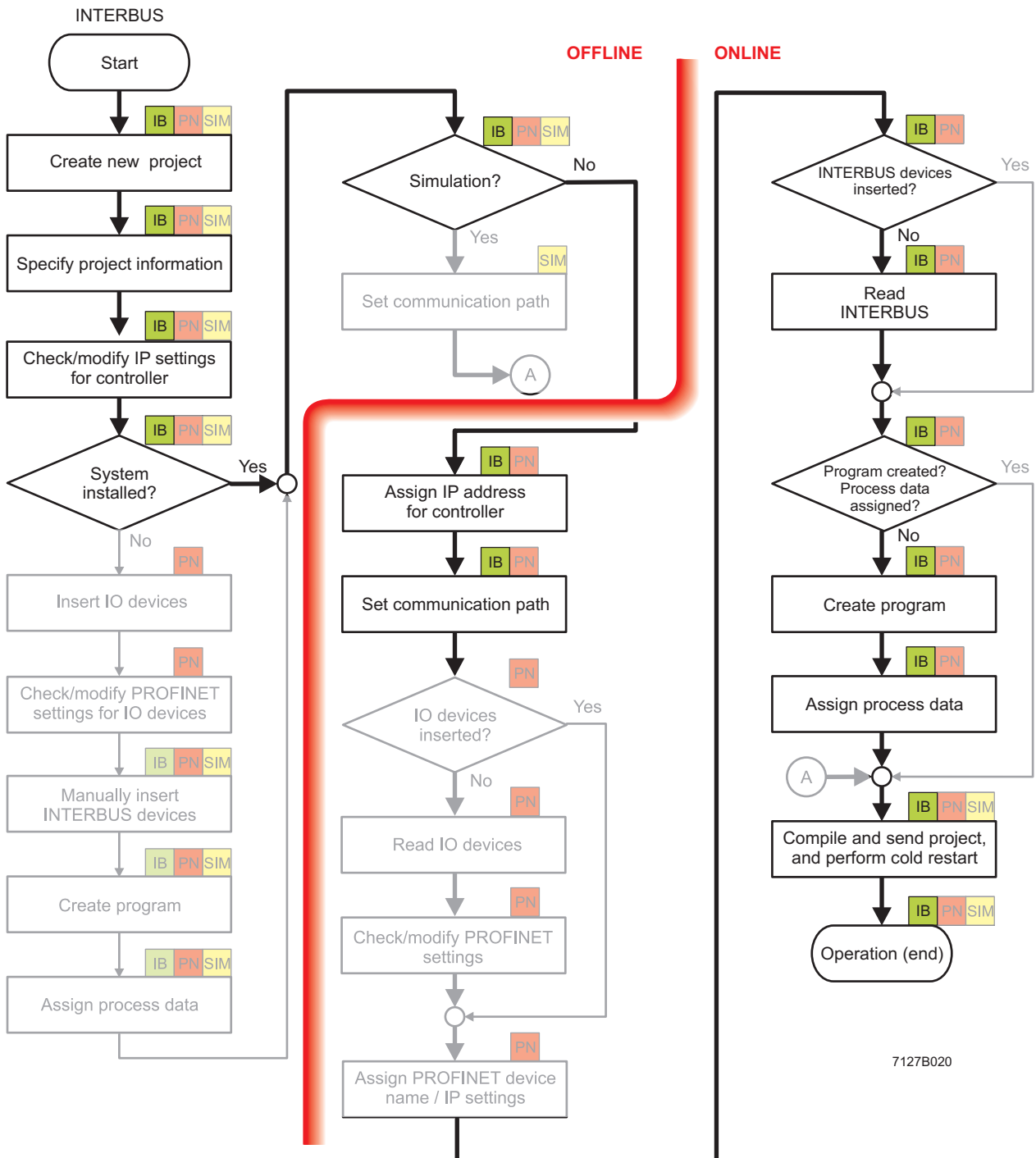
The sequence for creating the Figure 5-2 project is shown in INTERBUS.

When implementing the project, some of the tasks can be performed offline (without a connection to the INTERBUS system).

All tasks relating to communication must be performed online (with connection to the INTERBUS system).

In Figure 5-2 and in the description in this section, it is assumed that the system has been fully installed and all tasks are performed online. This is the quickest way to start up an INTERBUS system.

If the INTERBUS system has not yet been fully installed or if you would like to perform as many preparatory tasks as possible offline, follow the sequence described in Section 4, "Sequence for creating a project".



7127B020

Figure 5-2 Sequence for creating the INTERBUS project

### 5.3 Creating a new project



See also "Creating a new project" on page 4-3.

- Select the "New Project..." command from the "File" menu.
- Select the controller (here: ILC 390 PN 2TX-IB Rev. > 01/4.6F/3.00) and confirm your selection with "OK".
- Select the "File, Save Project As / Zip Project As..." command.
- Enter the project name "Quickstart" and save the project.

### 5.4 Specifying project information



See also "Specifying project information" on page 4-4.



- Switch to the bus configuration workspace.
- Adapt the project information to your project.

### 5.5 Checking/modifying IP settings for the controller



See also "Checking/modifying IP settings for the controller" on page 4-7.

The IP settings for the controller are made when the project is created.



**NOTE: Changes to project information are not applied automatically**

If any modifications are made to the project information that affect the IP settings for the controller, a warning is displayed. However, the modification is not implemented automatically. When a new project is created, the default settings are specified under the IP settings.

Adapt these settings, if necessary.



- Switch to the bus configuration workspace.
- Select the controller node.
- In the "Device Details" window, switch to the "IP Settings" tab.
- Check the IP settings and modify them, if necessary.
- If an IP address has still not been assigned, assign one according to "Assigning the IP address for the controller" on page 5-5.



The IP address that is assigned here for the controller is also implemented as the IP address for the communication path via TCP/IP.

## 5.6 Assigning the IP address for the controller



See also "Assigning the IP address for the controller" on page 4-23.

An IP address must first be assigned to the controller in order to enable communication. For the ILC 390 PN 2TX-IB controller, BootP is activated by default upon delivery for assigning the IP address via Ethernet. In this case, when setting the IP address for the first time a BootP server can be used. The corresponding procedure is described below. For all other options, please refer to "Assigning the IP address for the controller" on page 4-23.



- Establish an Ethernet connection between your PC and the controller.
- In the menu bar, select the "Extras, BootP/SNMP/TFTP-Configuration ..." menu.
- Activate the "BootP Server active" checkbox.
- Switch to the bus configuration workspace.
- Select the controller node (e.g., "ILC 390 PN 2TX-IB").
- Select the "IP Settings" tab in the "Device Details" window.
- Enter the MAC address of the controller. This can be found on a label on the device. It starts with "00.A0.45".
- Reset the controller.

The controller is temporarily assigned the IP address which is specified in the project for the controller.

- Select the "Extended Settings" tab in the "Device Details" window.

The specified IP parameters are displayed under "Manual definition of the TCP/IP settings".

- Confirm the displayed IP parameters or your modifications via "Send".

The IP address is now permanently stored on the controller parameterization memory.

## 5.7 Setting the communication path



See also "Setting the communication path" on page 4-26.



For a controller that supports Ethernet (e.g., ILC xxx ETH, ILC 390 PN 2TX-IB, RFC 4xx), the communication path is automatically set to "Ethernet" by default with the IP address of the controller specified under "IP Settings". If you would like to send your project to the controller via this Ethernet connection, the communication path does **not** have to be set. If not using this path, please proceed as described in "Setting the communication path" on page 4-26.

## 5.8 Reading in INTERBUS



See also "Reading in INTERBUS" on page 4-36.

- Select the "Connected Bus" command from the "View" menu to read in the connected INTERBUS system.
- Select the controller in the "Connected Bus" window (here: "ILC 390 PN").
- Open the context menu and select the "Import to Project, With Device Description" command.

The "Select Device" window that opens lists the modules whose device description corresponds to that of the connected devices.

- Select the device that is actually connected to the INTERBUS system, and repeat this step until all the devices are linked to their device description. According to the example, the following I/O modules are connected: IB IL 24 DO 16, IB IL 24 DO 8, and IB IL 24 DI 8.
- Disconnect the link to the controller by selecting "Offline" under "Selected Control System" in the "Connected Bus" window.

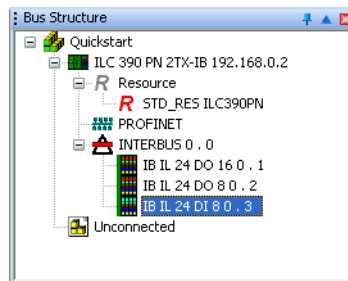


Figure 5-3 Complete bus configuration for the example project

## 5.9 Compiling after completing the bus topology



See also "Compiling after completing the bus topology" on page 4-16.



- Select the "Build, Make" command.

## 5.10 Creating the program

To program the example program, proceed as described in "Example program" on page 8-1.

- Set variable V0 to initial value 1. This set bit is then rotated to the left during program processing.



It is now assumed that you have created the program. If you skip this point, this may result in deviations.



## 5.11 Compiling after completing the program



See also "Compiling after creating the program" on page 4-17.



- Select the "Build, Make" command.

## 5.12 Assigning process data



See also "Generating variables and assigning process data" on page 4-18.



- Switch to the process data assignment workspace to assign the variables to the process data.
- Select the controller in the top right window. The standard configuration is then displayed in the top left window, "Symbols/Variables".
- In the top left window, "Symbols/Variables", select the standard resource (STD\_RES ILC390PN in the example).
- In the top right window, select the device for which you would like to link the process data to variables (IB IL 24 DI 8 in Figure 5-4).
- Select the process data item to be linked (3.1.1 in Figure 5-4).
- Variables are created when the program is created. Using drag & drop, link the selected variable to one of the variables on the left-hand side (IN3 in Figure 5-4).  
If you would like to link further process data but no corresponding variables have been created yet, select "Create Variable" in the context menu.

The created variable is displayed in the bottom left window.

- Repeat this procedure for all inputs to be evaluated and for all outputs to be controlled.

**Process data assigned**

The result of the process data assignment process is shown in the figure below.

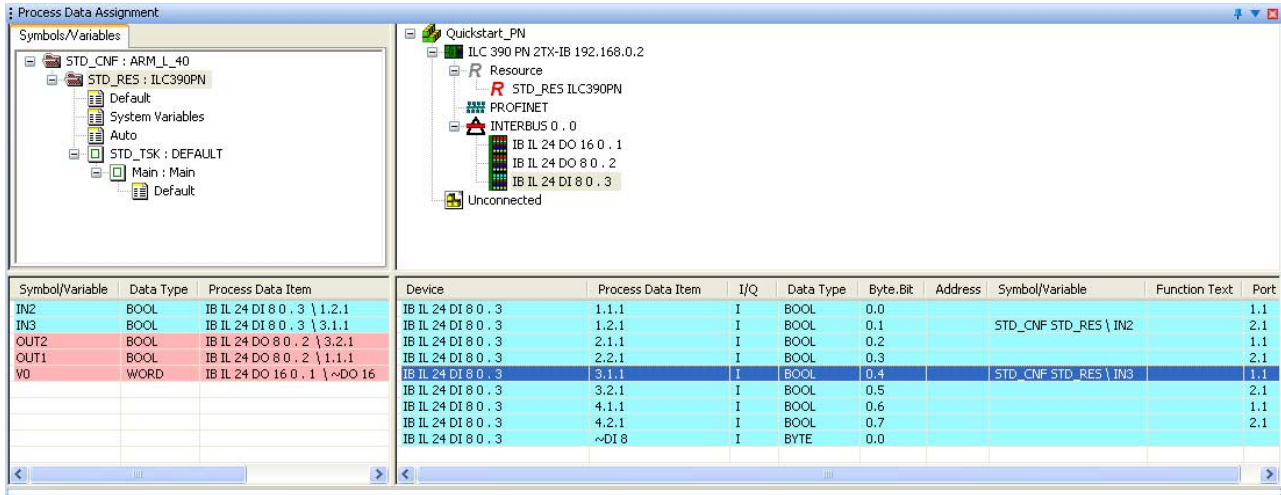


Figure 5-4 All used process data assigned to variables



Since the ONBOARD\_INPUT\_BIT10 variable is a system variable, it is not shown in Figure 5-4.

### 5.13 Compiling and sending a project, and performing a cold restart



See also "Compiling and sending a project, and performing a cold restart" on page 4-40.

#### Compile project



- Select the "Build, Make" command.

#### Send project



- Open the project control dialog box.
- Activate the "Include Bootproject" checkbox in the "Project" area.
- Click on "Download" in the area on the left.

### 5.14 Operation



See "Operation" on page 4-45.

## 6 Example project for a system consisting of PROFINET IO and INTERBUS

The creation of a project is described in detail in Section 4, "Sequence for creating a project".

This section covers:

- All steps to be taken in reference to the corresponding section in Section 4, "Sequence for creating a project"
- Project-specific settings
- Information and special notes for a project with PROFINET IO and INTERBUS

### 6.1 Project description

In the following, an example project is developed using function block diagram (FBD).



In order to obtain the best possible results, please use the same identifiers and names as used in this manual.

**Project name**

Quickstart\_PN

**Project hardware**

Specify the system structure during configuration. A system consisting of PROFINET IO and INTERBUS comprises the PROFINET IO controller, PROFINET IO devices, and INTERBUS devices. The example system is shown in Figure 6-1.

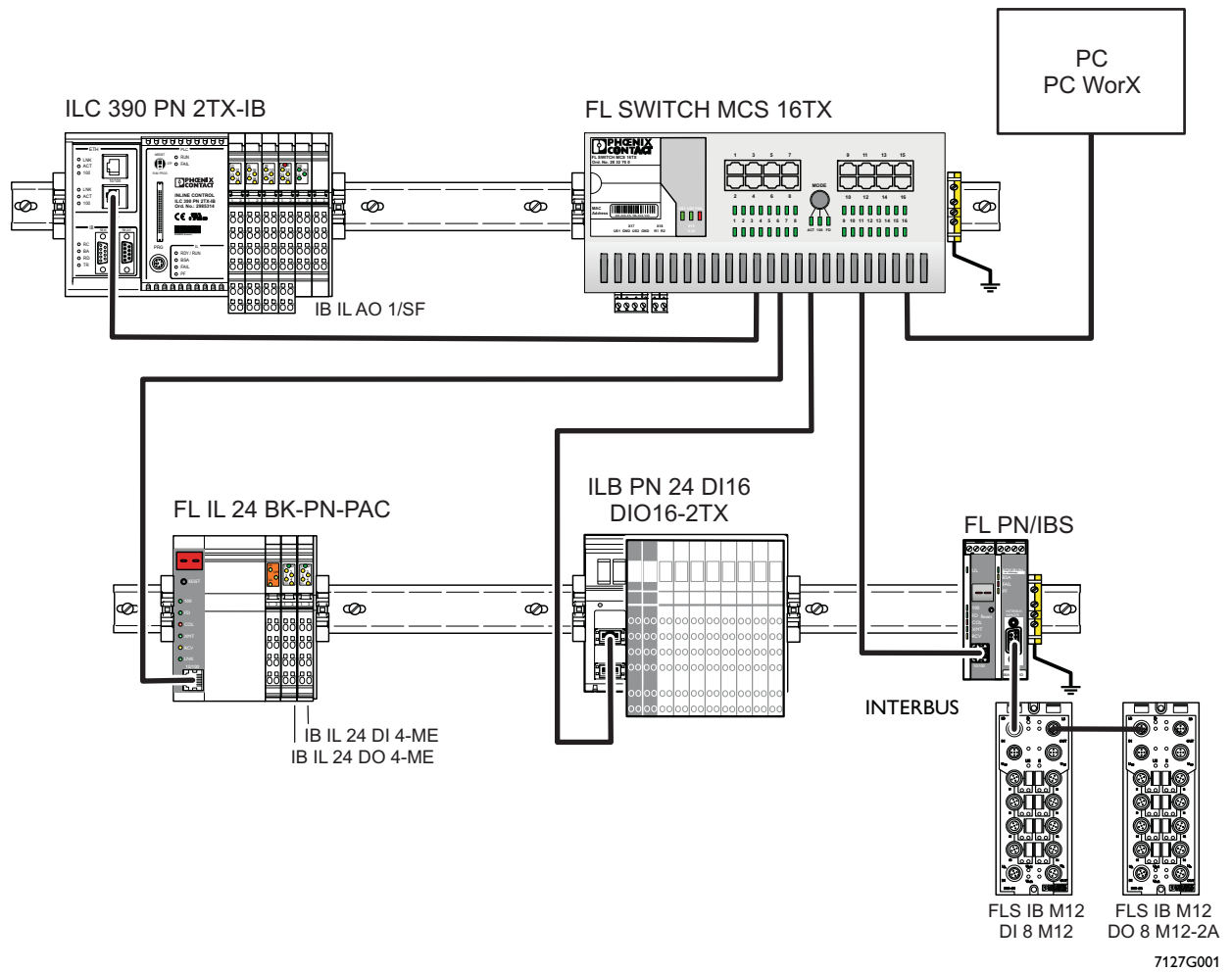


Figure 6-1 Example system

Devices in the example system:

Switch:	FL SWITCH MCS 16TX
PROFINET IO controller:	ILC 390 PN 2TX-IB
PROFINET IO device:	FL IL 24 BK-PN-PAC, ILB PN 24 DI16 DIO16-2TX
PROFINET IO device:	FL PN/IBS
INTERBUS proxy	
INTERBUS devices	At ILC 390 PN 2TX-IB: IB IL AO 1/SF
	At FL IL 24 BK-PN-PAC: IB IL 24 DO 4-ME, IB IL 24 DI 4-ME
	At FL PN/IBS: FLS IB M12 DI 8 M12, FLS IB M12 DO 8 M12-2A

Several PROFINET IO controllers can be used within a project. In this example, only one PROFINET IO controller is used.

## Example project for a system consisting of PROFINET IO and INTERBUS

### Inputs and outputs/process data

Inputs and outputs used in the example including process data and assigned variables.

Device	Connected to	Input/output according to the device data sheet	Signal at	Variable	Process data item
FL SWITCH MCS 16TX					
ILC 390 PN 2TX-IB		Input IN11	Co 3 TP 1.4	ONBOARD_INPUT_BIT10	3.1.4
ILB PN 24 DI16 DIO16-2TX		Output OUT1 to OUT16	Co 1 to 4 TP x.1 and x.4	V0	~DO16
IB IL AO 1/SF	ILC 390 PN 2TX-IB				
IB IL 24 DO 4-ME	FL IL 24 BK-PN-PAC	Output OUT3	TP 1.4	OUT1	1.4
IB IL 24 DI 4-ME	FL IL 24 BK-PN-PAC	Input IN1	TP 1.1	IN2	1.1
FLS IB M12 DI 8 M12	FL PN/IBS	Input IN3	FC 3 pin 4	IN3	IN3
FLS IB M12 DO 8 M12-2A	FL PN/IBS	Output OUT3	FC 3 pin 4	OUT2	OUT3
No assignment required				T1 T2	

Co Connector  
 TP Terminal point  
 FC Female connector

## 6.2 Sequence for creating the PROFINET project

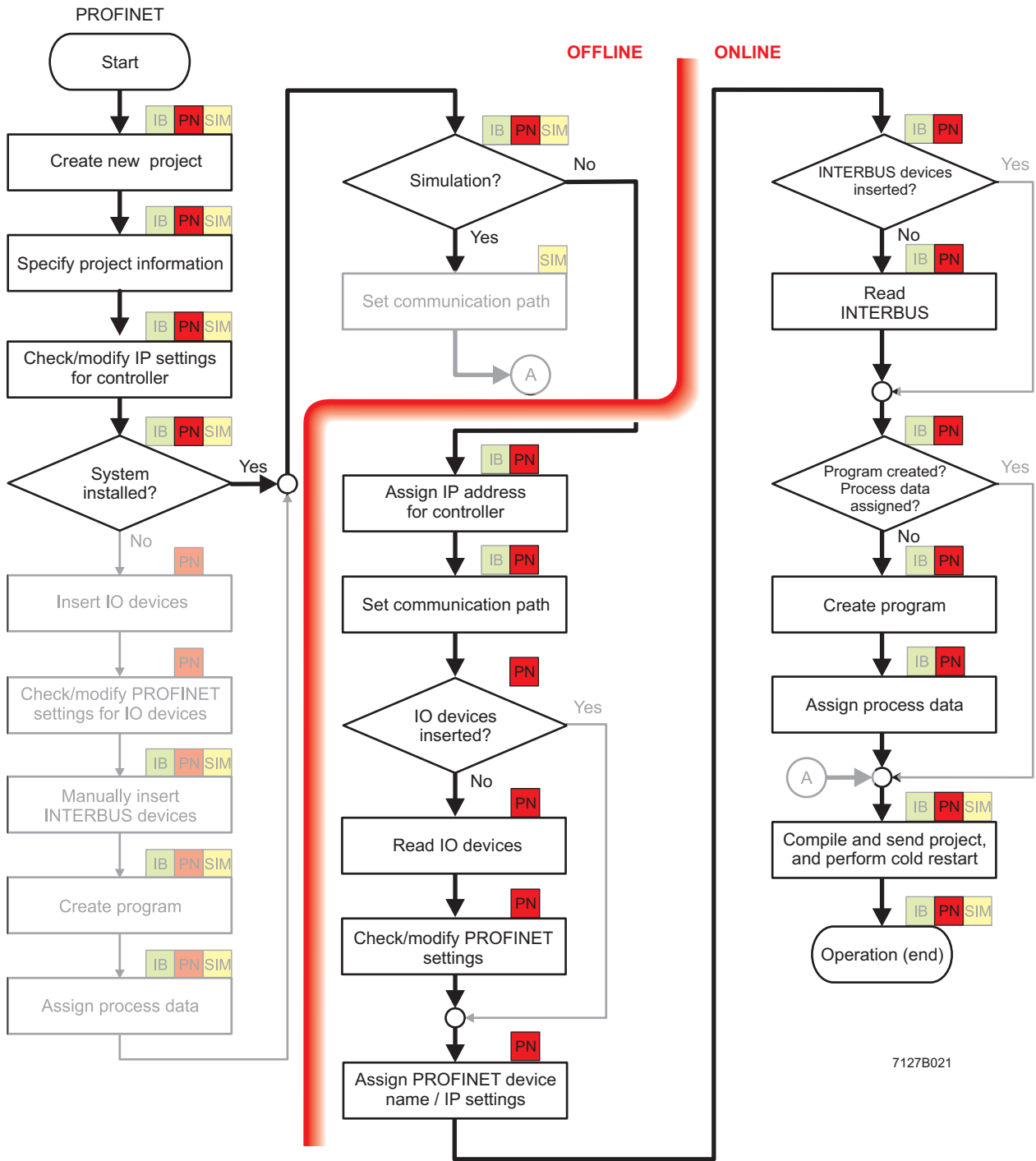
The sequence for creating the Figure 6-2 project is shown in PROFINET.

When implementing the project, some of the tasks can be performed offline (without a connection to the PROFINET IO system).

All tasks that require communication must be performed online (with connection to the PROFINET IO system).

In Figure 6-2 and in the description in this section, it is assumed that the system has been fully installed and all tasks are performed online. This is the quickest way to start up a PROFINET system. This path is shown in a dark color in Figure 6-2. Other possible paths are grayed out.

If the PROFINET system has not yet been fully installed or if you would like to perform as many preparatory tasks as possible offline, follow the grayed out sequence. This is described in detail in Section 4, "Sequence for creating a project".



7127B021

Figure 6-2 Sequence for creating the PROFINET project

## 6.3 Creating a new project



See also "Creating a new project" on page 4-3.

- Select the "New Project..." command from the "File" menu.
- Select the controller (here: ILC 390 PN 2TX-IB Rev. > 01/4.6F/3.00) and confirm your selection with "OK".
- Select the "File, Save Project As / Zip Project As..." command.
- Enter the project name "Quickstart\_PN" and save the project.

## 6.4 Specifying project information



See also "Specifying project information" on page 4-4.



- Switch to the bus configuration workspace.
- Adapt the project information to your project.
- Assign the name suffix for the PROFINET device name (here: quickstart.de) in the "Device Details" window under "Domain Postfix".

## 6.5 Preparing the PC for communication



See also "Preparing the PC for communication" on page 4-6.

- In the "Extras, PROFINET Configuration..." menu in PC WorX, select the network card for your computer that is to be used for communication.

## 6.6 Checking/modifying IP settings for the PROFINET IO controller



See also "Checking/modifying IP settings for the controller" on page 4-7.

The IP settings for the controller are made when the project is created.



**NOTE: Changes to project information are not applied automatically**

If any modifications are made to the project information that affect the IP settings for the controller, a warning is displayed. However, the modification is not implemented automatically. When a new project is created, the default settings are specified under the IP settings.

Adapt these settings, if necessary.



- Switch to the bus configuration workspace.
- Select the controller node.
- In the "Device Details" window, switch to the "IP Settings" tab.
- Check the IP settings and modify them, if necessary.
- If an IP address has still not been assigned, assign one according to "Assigning the IP address for the PROFINET IOcontroller" on page 6-7.



The IP address that is assigned here for the controller is also implemented as the IP address for the communication path via TCP/IP.



## 6.7 Assigning the IP address for the PROFINET IO controller



See also "Assigning the IP address for the controller" on page 4-23.

An IP address must first be assigned to the PROFINET IO controller in order to enable communication.

For the ILC 390 PN 2TX-IB PROFINET IO controller, BootP is activated by default upon delivery for assigning the IP address via Ethernet. In this case, when setting the IP address for the first time a BootP server can be used. The corresponding procedure is described below. For all other options, please refer to "Assigning the IP address for the controller" on page 4-23.



- Establish an Ethernet connection between your PC and the PROFINET IO controller.
- In the menu bar, select the "Extras, BootP/SNMP/TFTP-Configuration ..." menu.
- Activate the "BootP Server active" checkbox.
- Switch to the bus configuration workspace.
- Select the controller node (e.g., "ILC 390 PN 2TX-IB").
- Select the "IP Settings" tab in the "Device Details" window.
- Enter the MAC address of the controller. This can be found on a label on the device. It starts with "00.A0.45".
- Reset the controller.

The controller is temporarily assigned the IP address which is specified in the project for the controller.

- Select the "Extended Settings" tab in the "Device Details" window.

The specified IP parameters are displayed under "Manual definition of the TCP/IP settings".

- Confirm the displayed IP parameters or your modifications via "Send".

The IP address is now permanently stored on the controller parameterization memory.

## 6.8 Setting the communication path



See also "Setting the communication path" on page 4-26.



For a controller that supports Ethernet (e.g., ILC 350 PN, ILC 390 PN 2TX-IB, RFC 4xx), the communication path is automatically set to "Ethernet" by default with the IP address of the PROFINET IO controller specified under "IP Settings". If you would like to send your project to the controller via this Ethernet connection, the communication path does **not** have to be set.

If not using this path, please proceed as described in "Setting the communication path" on page 4-26.

## 6.9 Reading in and importing PROFINET IO devices



See also "Reading in and importing PROFINET IO devices" on page 4-29.



- Make sure you are in the bus configuration workspace.
- In the "Bus Structure" window, select the PROFINET node for the controller.
- Open the context menu and select the "Read PROFINET..." menu item.

All the connected PROFINET IO devices are displayed. When reading in, the device name and the IP parameters are imported for each device - if present.

- Select the devices that are to be included in your project.
- Click on "Insert" to include the selected PROFINET IO devices in your project.

If a PROFINET device name does not yet exist for a device, a name should be assigned at this point.

- Activate the "Name Device" checkbox.
- Check the displayed PROFINET device name (see also "DNS/PROFINET Device Name" on page 4-10).
- Assign the name to the relevant device.
- Confirm the display or your entry with "OK".
- Click "Close" to close the "Read PROFINET" dialog box.

The PROFINET structure is displayed in the "Bus Structure" window.

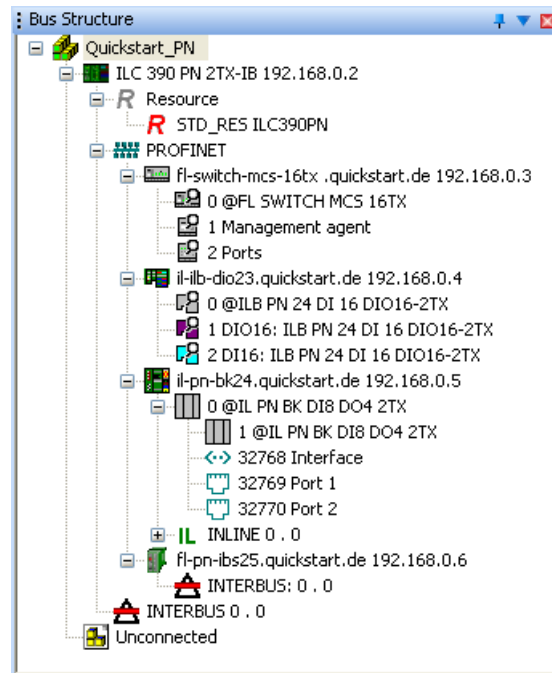


Figure 6-3 Bus configuration with read in PROFINET IO devices

## 6.10 Checking/modifying the PROFINET settings for PROFINET IO devices



See also "Checking/modifying the PROFINET settings for PROFINET IO devices" on page 4-34.

When **reading in**, the PROFINET settings of the relevant PROFINET IO devices are imported with default values. Check these settings and modify them, if necessary.



- Make sure you are in the bus configuration workspace.
- In the "Bus Structure" window, select the PROFINET IO device.
- Under "Device Details", select the "PROFINET Settings" tab.
- Check the PROFINET settings and modify them, if necessary.

## 6.11 Transferring PROFINET device names and IP settings to PROFINET IO devices



See also "Transferring PROFINET device names and IP settings to PROFINET IO devices" on page 4-35.

Before a PROFINET IO device can be operated in a PROFINET IO network, the PROFINET device name and IP address configured in PC WorX must also be made known to the device itself.

- Make sure that the PROFINET IO controller can establish communication with the PROFINET IO devices.
- Compile the project, send it to the controller, and perform a cold restart. Proceed as described in "Compiling and sending a project, and performing a cold restart" on page 4-40.

During startup, the PROFINET controller automatically assigns the IP settings and device parameterizations specified in the project to the PROFINET IO devices.

- To check the assignment of device names and IP settings, select the "PROFINET Stationnames" tab in the "Device Details" window.

The updated list contains the PROFINET IO devices together with their PROFINET device names and the IP parameters.

When the tab is selected, the list is updated. If you would like to update the list again later, click on "Refresh".

In order to limit the search results, various options are available:

No constraints	All devices that are available in the network are listed.
Unnamed	All those devices that do not yet have a PROFINET device name are listed.
Not in project	All those devices that are not included in the project are listed.
Same type	Only those devices that are the same type as the device selected in the bus configuration are listed.

The BF LED is now off on all PROFINET IO devices.  
The PROFINET network is running.



For additional information about assigning PROFINET device names and IP addresses, please also refer to "Additional information" on page 6-15.

## 6.12 Reading in INTERBUS



See also "Reading in INTERBUS" on page 4-36.

The process for reading in INTERBUS modules that are connected to a PROFINET IO controller or a PROFINET IO device is the same.

### PROFINET IO controller

After the communication path to the PROFINET IO controller has been set, the INTERBUS devices connected to the controller can be read in.

### PROFINET IO device

After device names and IP parameters for the PROFINET IO devices have been assigned, the PROFINET IO devices connected to the INTERBUS devices can be read in.

- Select the "Connected Bus" command from the "View" menu to read in the connected INTERBUS system.
- Select one of the controllers displayed in the "Connected Bus" window.
- Open the context menu and select the "Import to Project, With Device Description" command.

The "Select Device" window that opens lists the modules whose device description corresponds to that of the connected devices.

- Select the device that is actually connected to the INTERBUS system, and repeat this step until all the devices are linked to their device description.
- Select all the "controllers" that are connected to the INTERBUS devices. These include the Inline Controller as well as all the PROFINET IO devices displayed in the "Connected Bus" window.
- Disconnect the link to the controller by selecting "Offline" under "Selected Control System" in the "Connected Bus" window.

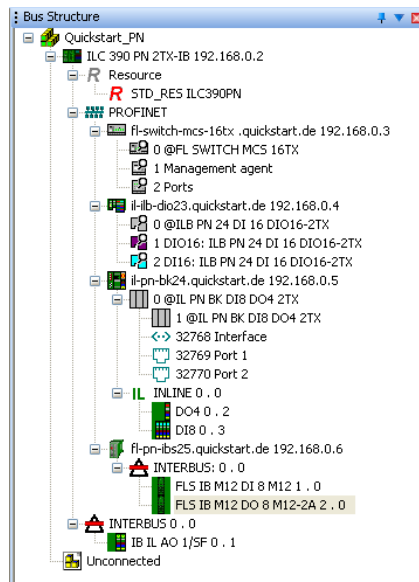


Figure 6-4 Complete bus configuration for the example project

### 6.13 Compiling after completing the bus topology



See also "Compiling after completing the bus topology" on page 4-16.



- Select the "Build, Make" command.

### 6.14 Creating the program

To program the example program, proceed as described in "Example program" on page 8-1.

- Set variable V0 to initial value 1. This set bit is then rotated to the left during program processing.



It is now assumed that you have created the program. If you skip this point, this may result in deviations.

### 6.15 Compiling after creating the program



See also "Compiling after creating the program" on page 4-17.



- Select the "Build, Make" command.

## 6.16 Assigning process data



See also "Generating variables and assigning process data" on page 4-18.



- Switch to the process data assignment workspace to assign the variables to the process data.
- Select the controller in the top right window. The standard configuration is then displayed in the top left window, "Symbols/Variables".
- In the top left window, "Symbols/Variables", select the standard resource (STD\_RES ILC390PN in the example).
- In the top right window, select the device for which you would like to link the process data to variables.

Please note for a PROFINET IO device with inputs and outputs that this is represented with its modules.

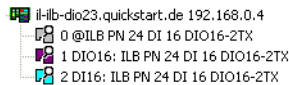


Figure 6-5 Representation of a PROFINET IO device with its modules

- In this case, select the module where the required inputs or outputs are located (DIO16: ILB PN 24 DI16 DIO16-2TX in Figure 6-6).
- Select the process data item to be linked (~DO16 in Figure 6-6).
- Variables are created when the program is created. Using drag & drop, link the selected variable to one of the variables on the left-hand side (V0 in Figure 6-6).

The created variable is displayed in the bottom left window.

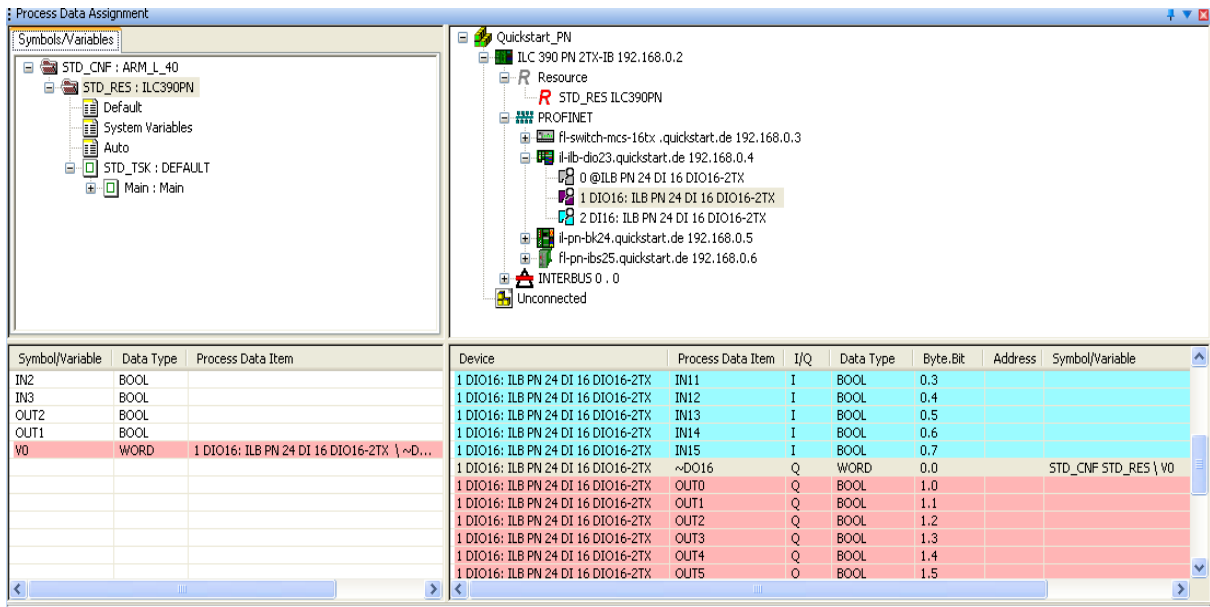


Figure 6-6 Assigning process data for a PROFINET IO device

- Repeat this procedure for all inputs to be evaluated and for all outputs to be controlled.
- For the specified example, assign the inputs and outputs to the created variables according to "Inputs and outputs/process data" on page 6-3.

The result of the process data assignment process is shown in the figure below.

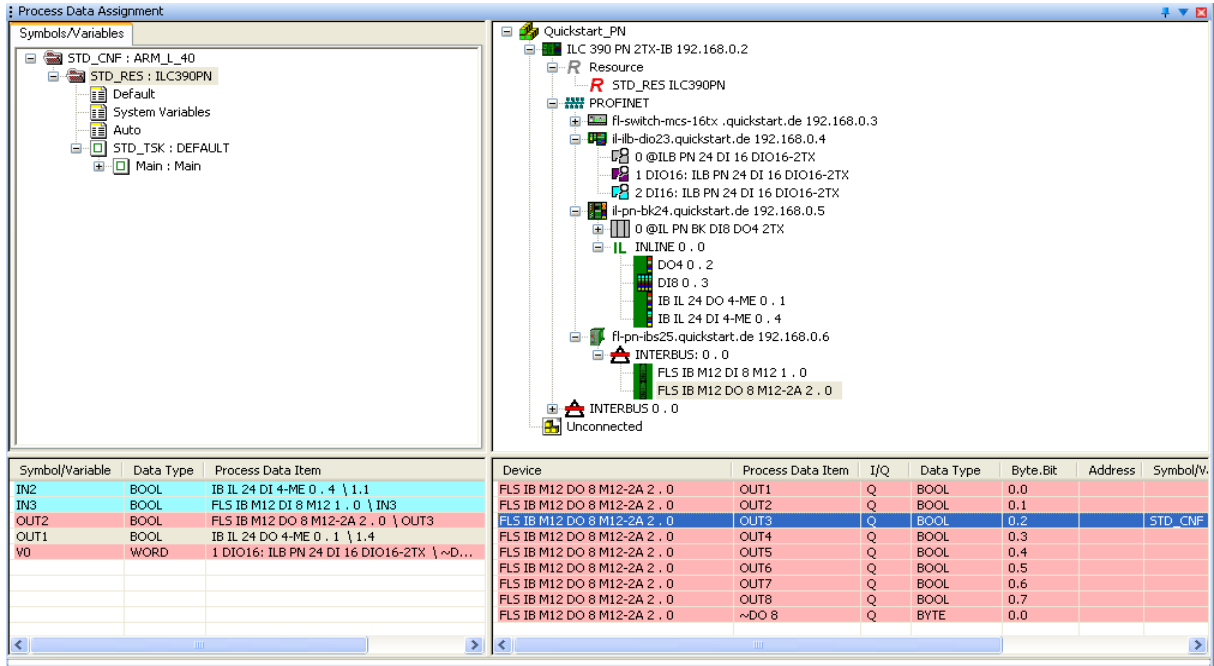


Figure 6-7 All used process data assigned to variables



Since the ONBOARD\_INPUT\_BIT10 variable is a system variable, it is not shown in Figure 6-7.

## 6.17 Compiling and sending a project, and performing a cold restart



See also "Compiling and sending a project, and performing a cold restart" on page 4-40.

### Compile project



- Select the "Build, Make" command.

### Send project



- Open the project control dialog box.
- Activate the "Include Bootproject" checkbox in the "Project" area.
- Click on "Download" in the area on the left.



## 6.18 Operation



See "Operation" on page 4-45.

## 6.19 Additional information

### 6.19.1 Changing the PROFINET device name

If, following device naming, you wish to change device names, proceed as follows:

If a project is located on the PROFINET IO controller, the controller attempts to start up the PROFINET IO devices with the information stored in the project. In this case, a PROFINET device name cannot be assigned. Therefore, if there are Ethernet connections between the PROFINET IO devices and the PROFINET IO controller with the project, these connections must be aborted first. To do this, proceed as follows:



- Open the project control dialog box.
- Click on "Stop".
- Click on "Reset".

All Ethernet connections are aborted and the PROFINET device names can now be assigned.



- Make sure you are in the bus configuration workspace.
- In the bus configuration, select the PROFINET IO device which is to be assigned the name.
- Select the "PROFINET Settings" tab in the "Device Details" window.
- Change the device name under "DNS/PROFINET Device Name".
- In the "Device Details" window, select the "PROFINET Stationnames" tab.

When the tab is selected, the list is updated. If you would like to update the list again later, click on "Refresh".

In order to limit the search results, various options are available:

No constraints	All devices that are available in the network are listed.
Unnamed	All those devices that do not yet have a PROFINET device name are listed.
Not in project	All those devices that are not included in the project are listed.
Same type	Only those devices that are the same type as the device selected in the bus configuration are listed.

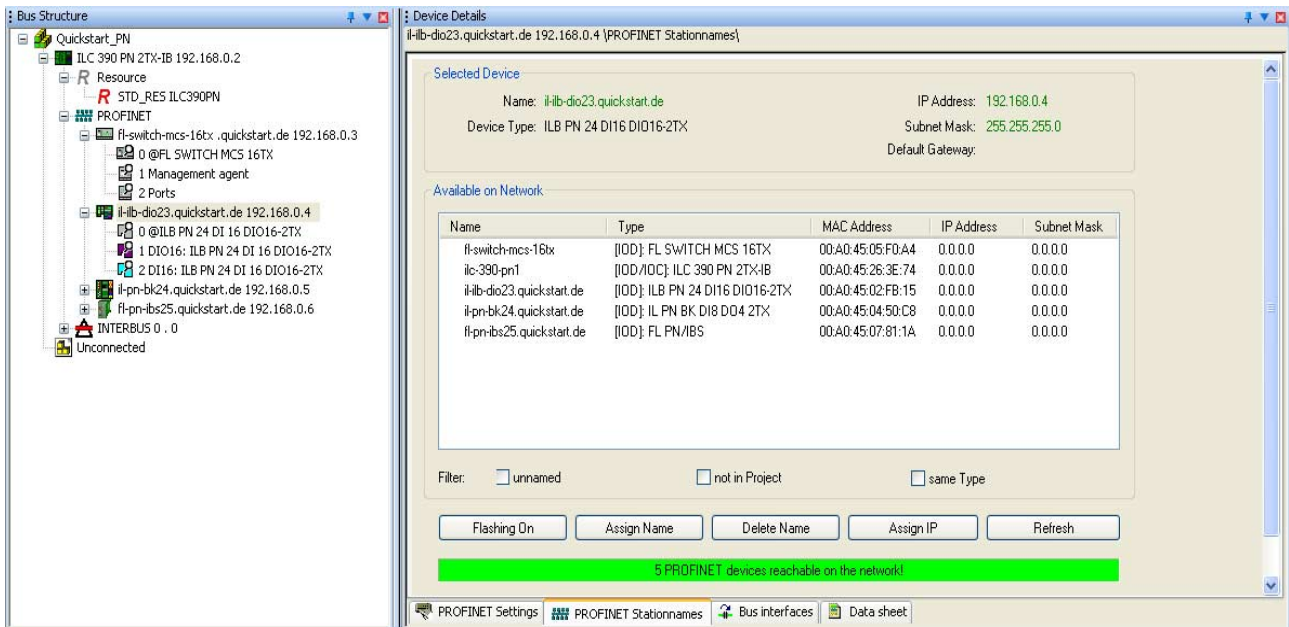


Figure 6-8 List of all PROFINET IO devices that can be accessed in the network

In the "Selected Device" area, the PROFINET IO device name specified under "IP Settings" is displayed.

- From the list, select the corresponding PROFINET IO device (e.g., using the indicated MAC address).

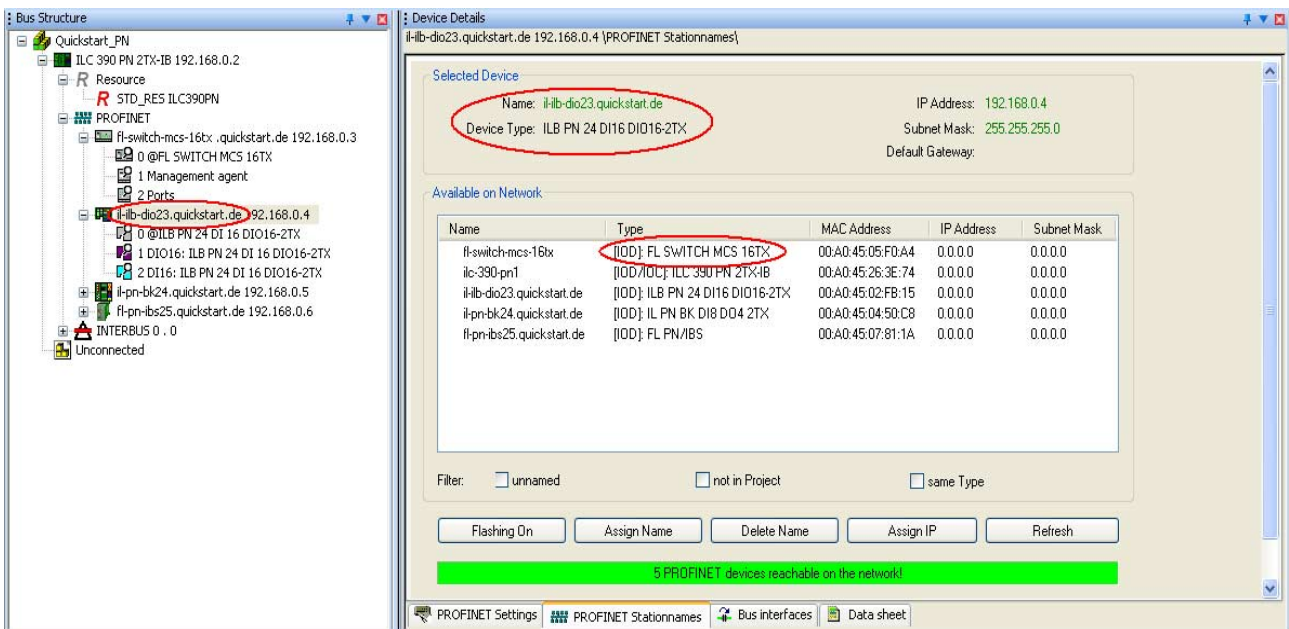


Figure 6-9 Selecting the device

A flashing display can be triggered on the device via "Flashing On". This display can be used to find the selected device in the field.

When the button is activated, the devices in the example system flash as follows:

ILB PN 24 DI16 DIO16-2TX	RDY LED flashing
FL IL 24 BK-PN-PAC	Double-zero flashing on the 7-segment display
FL PN/IBS	Double-zero flashing on the 7-segment display

The flashing process is also indicated in the list.

- Send the name to the device via "Assign Name".



If you receive an error message, first delete the name via "Delete Name" and then assign the name again.

The PROFINET device name is transferred from the PC to the PROFINET IO device. The previous parameters are still displayed as the IP parameters. The IP parameters are not assigned until the next step.

- Assign the appropriate PROFINET device name to each PROFINET IO device.
- Compile the project, send it to the controller, and perform a cold restart (see "Compiling and sending a project, and performing a cold restart" on page 4-40).

### 6.19.2 Renumbering devices

There is also the option of automatically regenerating the IP parameters (e.g., IP addresses, Domain Postfix) for all devices. The settings from the project information are used for this.

- Select the project node.
- Select "Renumber Devices/Whole project" and then the menu item for renumbering the devices (e.g., "All DNS/PROFINET Device Names").

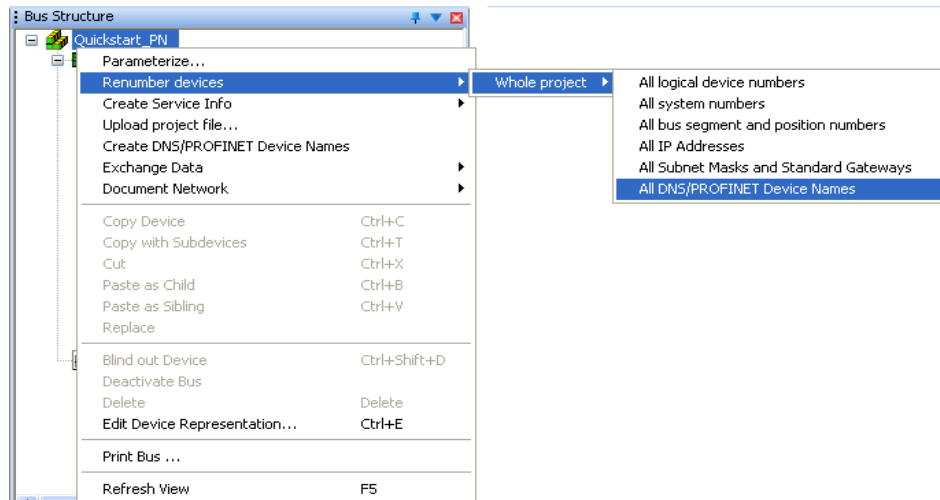


Figure 6-10 Renumbering devices

In this example, the newly specified names (e.g., il-ib-dio27) are extended to include the Domain Postfix "quickstart.de" specified in the project node. This is the complete PROFINET device name.

- If necessary, the IP addresses can also be renumbered, e.g., to view the IP addresses in the bus configuration in ascending order.

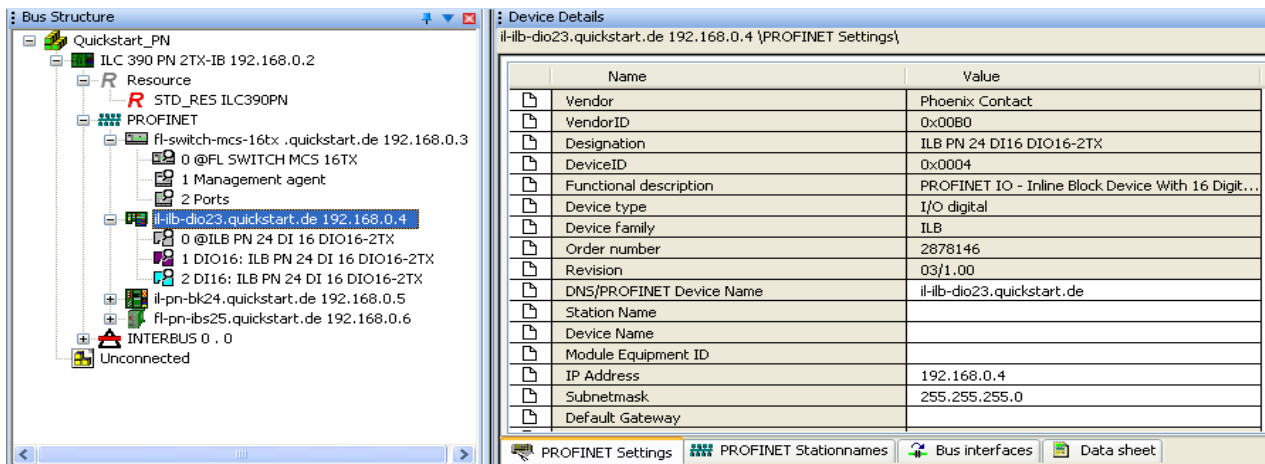


Figure 6-11 Renumbering devices (PROFINET device names and IP addresses)

All the devices in the project now have the correct IP parameters. In the section below, these parameters must **actually** be assigned to the devices.

- Compile the project, send it to the controller, and perform a cold restart (see "Compiling and sending a project, and performing a cold restart" on page 4-40).

### 6.19.3 Assigning IP parameters for the PROFINET IO device

In order to enable the connected INTERBUS modules to be read into the bus configuration, the IP parameters must be assigned to the PROFINET IO devices.

There are two options for assigning IP parameters. Select an option based on the specified requirements.

#### 6.19.3.1 Option 1: IP parameter assignment during PROFINET IO controller startup

This option always works **provided the PROFINET IO controller is able to establish communication with the PROFINET IO devices.**



- Make sure you are in the bus configuration workspace.
- In the "Device Details" window, select the "PROFINET Stationnames" tab. You can later check whether the IP parameters have been assigned.
- Compile the project, send it to the controller, and perform a cold restart. Proceed as described in "Compiling and sending a project, and performing a cold restart" on page 4-40.

During startup, the PROFINET IO controller assigns the IP parameters specified in the project to the PROFINET IO devices.

- Click on "Refresh".

The updated list contains the PROFINET IO devices together with their PROFINET device names and the IP parameters.

Name	Type	MAC Address	IP Address	Subnet Mask
fl-switch-mcs-16tx	[I0D]: FL SWITCH MCS 16TX	00:A0:45:05:F0:A4	192.168.0.3	255.255.255.0
ilc-390-pn1	[I0D/I0C]: ILC 390 PN 2TX-IB	00:A0:45:26:3E:74	192.168.0.2	255.255.255.0
il-ib-dio23.quickstart.de	[I0D]: ILB PN 24 DI16 DIO16-2TX	00:A0:45:02:FB:15	192.168.0.4	255.255.255.0
il-pn-bk24.quickstart.de	[I0D]: IL PN BK DI8 DO4 2TX	00:A0:45:04:50:C8	192.168.0.5	255.255.255.0
fl-pn-ibs25.quickstart.de	[I0D]: FL PN/IBS	00:A0:45:07:81:1A	192.168.0.6	255.255.255.0

Filter:  unnamed  not in Project  same Type

Figure 6-12 All devices with PROFINET device names and IP parameters

### 6.19.3.2 Option 2: IP parameter assignment without project on the PROFINET IO controller

This option is specifically designed to read in INTERBUS modules that are connected to a PROFINET IO device **without** first sending a project to the PROFINET IO controller.

This option **only works under the following conditions:**

1. There is no project on the PROFINET IO controller, i.e., the PROFINET IO controller did not yet establish a connection to the PROFINET IO devices.  
**Or**
2. There is no connection between the PROFINET IO controller and the PROFINET IO devices (Ethernet cable not plugged in or no supply voltage at the PROFINET IO controller).

In each case, the **BF LED must flash** on each of the PROFINET IO devices. If the BF LED of a PROFINET IO device does not flash, no IP parameters can be assigned to this device as it is currently communicating.

- Select the device to which you would like to assign the IP parameters. Proceed as described in "Transferring PROFINET device names and IP settings to PROFINET IO devices" on page 6-10.
- Send the IP parameters to the device via "Assign IP".
- Assign the IP parameters to each PROFINET IO device.

Name	Type	MAC Address	IP Address	Subnet Mask
fl-switch-mcs-16tx	[I0D]: FL SWITCH MCS 16TX	00:A0:45:05:F0:A4	192.168.0.3	255.255.255.0
ilc-390-pn1	[I0D/I0C]: ILC 390 PN 2TX/IB	00:A0:45:26:3E:74	192.168.0.2	255.255.255.0
il-ib-dio23.quickstart.de	[I0D]: ILB PN 24 DI16 DIO16-2TX	00:A0:45:02:FB:15	192.168.0.4	255.255.255.0
il-pn-bk24.quickstart.de	[I0D]: IL PN BK DI8 DO4 2TX	00:A0:45:04:50:C8	192.168.0.5	255.255.255.0
fl-pn-ibs25.quickstart.de	[I0D]: FL PN/IBS	00:A0:45:07:81:1A	192.168.0.6	255.255.255.0

Filter:  unnamed  not in Project  same Type

Figure 6-13 All devices with PROFINET device names and IP parameters

Now the INTERBUS modules connected to the devices can be read in.

If there was already a project on the PROFINET IO controller and the connection was aborted in order to send the IP parameters, please note the following:

On startup, the PROFINET IO controller starts up the PROFINET IO devices with the IP parameters stored in this project (which may be an old one). Only after compiling, sending, and performing a cold restart for the new project can the IP parameters be transferred from this project to the PROFINET IO devices.



If error messages are displayed and the IP parameters are not applied:

- Check whether the **BF LED is flashing**.  
**If so:** Check all Ethernet connections as well as your previous actions.  
**If not:** You cannot assign any IP parameters. Either interrupt communication or proceed according to "Option 1: IP parameter assignment during PROFINET IO controller startup" on page 6-19.

## 7 Example project for a simulation with processor type "IPC" controllers

For processor type "IPC" controllers (RFC 4xx and S-MAX 4xx), PC WorX can be used to simulate a project completely without hardware. A program can be created and tested without having to actually connect hardware. The EasySim simulation in PC WorX is available for this.



For more detailed information about the EasySim simulation, please refer to the online help for PC WorX.

### 7.1 Project description

In the following, an example project is developed for the simulation using function block diagram (FBD).



In order to obtain the best possible results, please use the same identifiers and names as used in this manual.

Project name: SIM

Project hardware:

- Controller: RFC 430 ETH-IB
- Bus coupler: IBS IL 24 BK T/U
- I/O modules: IB IL 24 DO 16, IB IL 24 DO 8, and IB IL 24 DI 8

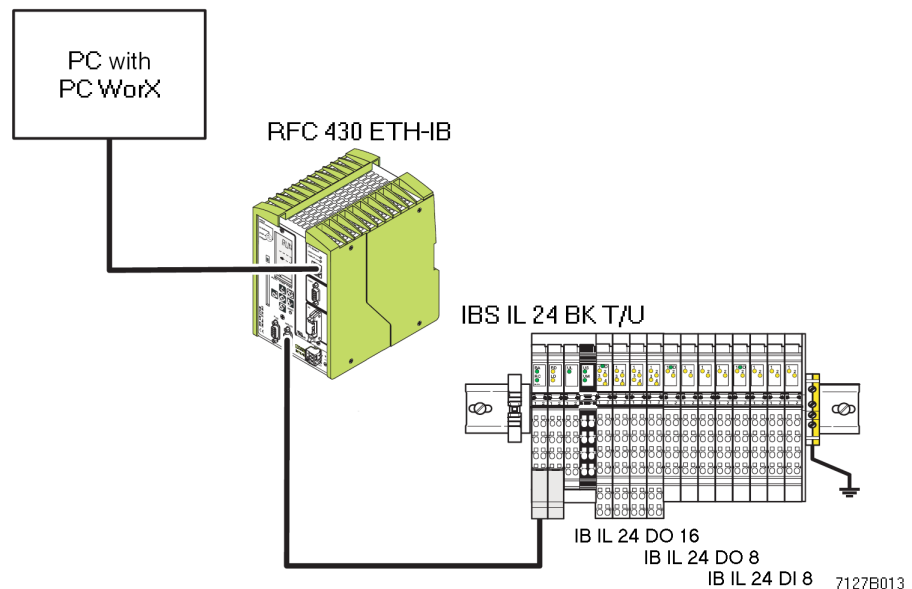


Figure 7-1 Example bus configuration

**Inputs and outputs/process data**

Inputs and outputs used in the example including process data and assigned variables.

Device	Inputs/outputs according to the device data sheet	Signal at	Variable	Process data item
IB IL 24 DI 8	Input IN1	Co 1 TP 1.1	S1	1.1.1
	Input IN2	Co 1 TP 2.1	IN2	1.2.1
	Input IN5	Co 3 TP 1.1	IN3	3.1.1
IB IL 24 DO 8	Output OUT1	Co 1 TP 1.1	OUT1	1.1.1
	Output OUT6	Co 3 TP 2.1	OUT2	3.2.1
IB IL 24 DO 16	Outputs OUT1 to OUT16	Co 1 to 4 TP x.1 and x.4	V0	~DO16
No assignment required			T1 T2	

- Co Connector
- TP Terminal point
- x Row on the Inline connector

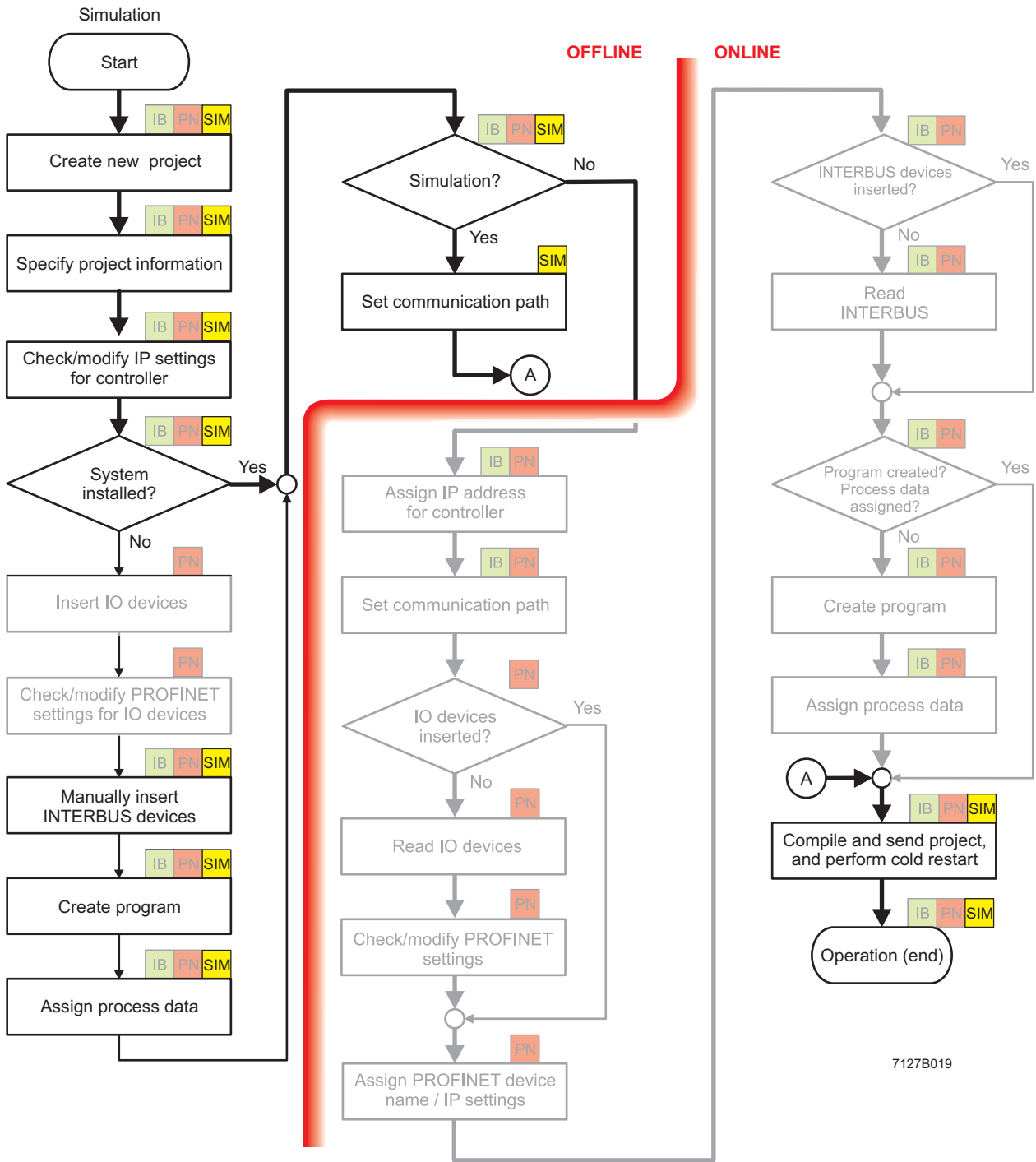
## 7.2 Sequence for creating the simulation project

The sequence for creating the simulation project is shown in Figure 7-2.

When implementing the project, all of the tasks are performed offline (without a connection to the system).



Example project for a simulation with processor type "IPC" controllers



7127B019

Figure 7-2 Sequence for creating the simulation project

## 7.3 Creating a new project



See also "Creating a new project" on page 4-3.

- Select the "New Project..." command from the "File" menu.
- Select the RFC 430 ETH-IB controller and confirm your selection with "OK".
- Select the "File, Save Project As / Zip Project As..." command.
- Enter the project name "SIM" and save the project.



If you want to use a different controller for the example project then select a different project template. Please note that the following descriptions may differ. Always select only one processor type "IPC" controller.

## 7.4 Specifying project information



See also "Specifying project information" on page 4-4.



- Switch to the bus configuration workspace.
- Adapt the project information to your project.

## 7.5 Checking/modifying IP settings for the controller



See also "Checking/modifying IP settings for the controller" on page 4-7.

The IP settings for the controller are made when the project is created.



**NOTE: Changes to project information are not applied automatically**

If any modifications are made to the project information that affect the IP settings for the controller, a warning is displayed. However, the modification is not implemented automatically. When a new project is created, the default settings are specified under the IP settings.

Adapt these settings, if necessary.



- Switch to the bus configuration workspace.
- Select the controller node.
- In the "Device Details" window, switch to the "IP Settings" tab.
- Check the IP settings and modify them, if necessary.



The IP address that is assigned here for the controller is also implemented as the IP address for the communication path via TCP/IP.

## 7.6 Manually inserting INTERBUS devices



See also "Manually inserting INTERBUS devices" on page 4-12.

The following devices are used for the example project:

- Controller: RFC 430 ETH-IB
- Bus coupler: IBS IL 24 BK-T/U
- I/O modules: IB IL 24 DO 16, IB IL 24 DO 8, and IB IL 24 DI 8
- Select a point in the "Bus Structure" window where a device is to be inserted. To insert the bus coupler, select the INTERBUS node for the controller, as the first INTERBUS device can only be inserted here.
- In the device catalog, open the product range for the inserted device (e.g., IL for Inline) under "Phoenix Contact".
- Under the product range, open the product group (e.g., bus coupler, I/O digital).
- Select the device to be inserted (e.g., IBS IL 24 BK-T/U).
- Hold down the mouse button and drag the selected device to the insertion point.
- Insert all the other devices.  
In the example: Insert the Inline modules (I/O digital) below the bus coupler.

The bus configuration with the INTERBUS modules used in the example is shown in Figure 7-3.

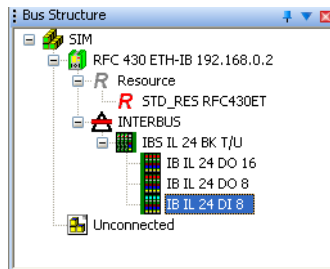


Figure 7-3 Complete bus configuration for the example project

## 7.7 Compiling after completing the bus topology



See also "Compiling after completing the bus topology" on page 4-16.

- Select the "Rebuild Project" command from the "Build" menu.

## 7.8 Creating the program

To program the example program, proceed as described in "Example program" on page 8-1.

- Set variable V0 to initial value 1. This set bit is then rotated to the left during program processing.



It is now assumed that you have created the program. If you skip this point, this may result in deviations.

## 7.9 Compiling after creating the program



See also "Compiling after creating the program" on page 4-17.



- Select the "Build, Make" command.

## 7.10 Assigning process data



See also "Generating variables and assigning process data" on page 4-18.



- Switch to the process data assignment workspace to assign the variables to the process data.
- Select the controller in the top right window. The standard configuration is then displayed in the top left window, "Symbols/Variables".
- In the top left window, "Symbols/Variables", select the standard resource (STD\_RES: RFC430ET in the example).
- In the top right window, select the device for which you would like to link the process data to variables (IB IL 24 DO 16 in Figure 7-4).
- Select the process data item to be linked (~DO16 in Figure 7-4).
- Variables are created when the program is created. Using drag & drop, link the selected variable to one of the variables on the left-hand side (V0 in Figure 7-4).

## Example project for a simulation with processor type "IPC" controllers

The screenshot shows the 'Process Data Assignment' window. The left pane displays a tree view of symbols/variables under 'STD\_CNCF : IPC\_40', with 'STD\_RES : RFC430ET' expanded to show 'Main : Main'. The right pane shows a SIM tree view for 'RFC 430 ETH-IB 192.168.0.2', with 'IB IL 24 DO 16' selected. The bottom table shows the assignment of 'V0' to 'IB IL 24 DO 16 \~DO 16'.

Symbol/Variable	Data Type	Process Data Item	Description
S1	BOOL		
OUT1	BOOL		
V0	WORD	IB IL 24 DO 16 \~DO 16	
IN2	BOOL		
IN3	BOOL		
OUT2	BOOL		

Device	Process Data Item	I/Q	Data Type	Byte.Bit	Address	Symbol/Vari
IB IL 24 DO 16	1.1.1	Q	BOOL	1.0		
IB IL 24 DO 16	1.2.1	Q	BOOL	1.1		
IB IL 24 DO 16	1.1.4	Q	BOOL	1.2		
IB IL 24 DO 16	1.2.4	Q	BOOL	1.3		
IB IL 24 DO 16	2.1.1	Q	BOOL	1.4		
IB IL 24 DO 16	2.2.1	Q	BOOL	1.5		
IB IL 24 DO 16	2.1.4	Q	BOOL	1.6		
IB IL 24 DO 16	2.2.4	Q	BOOL	1.7		
IB IL 24 DO 16	3.1.1	Q	BOOL	0.0		
IB IL 24 DO 16	3.2.1	Q	BOOL	0.1		
IB IL 24 DO 16	3.1.4	Q	BOOL	0.2		
IB IL 24 DO 16	3.2.4	Q	BOOL	0.3		
IB IL 24 DO 16	4.1.1	Q	BOOL	0.4		
IB IL 24 DO 16	4.2.1	Q	BOOL	0.5		
IB IL 24 DO 16	4.1.4	Q	BOOL	0.6		
IB IL 24 DO 16	4.2.4	Q	BOOL	0.7		
IB IL 24 DO 16	\~DO 16	Q	WORD	0.0		STD_CNCF ST

Figure 7-4 Variable V0

The created variable is displayed in the bottom left window.

- Repeat this procedure for all inputs to be evaluated and for all outputs to be controlled.

The result of the process data assignment process is shown in the figure below.

The screenshot shows the 'Process Data Assignment' window after multiple assignments. The left pane is the same as in Figure 7-4. The right pane shows the SIM tree view with 'IB IL 24 DI 8' selected. The bottom table shows the final assignment of 'V0' to 'IB IL 24 DI 8 \3.1.1'.

Symbol/Variable	Data Type	Process Data Item	Description
S1	BOOL	IB IL 24 DI 8 \1.1.1	
OUT1	BOOL	IB IL 24 DO 8 \1.1.1	
V0	WORD	IB IL 24 DO 16 \~DO 16	
IN2	BOOL	IB IL 24 DI 8 \1.2.1	
IN3	BOOL	IB IL 24 DI 8 \3.1.1	
OUT2	BOOL	IB IL 24 DO 8 \3.2.1	

Device	Process Data Item	I/Q	Data Type	Byte.Bit	Address	Symbol/Vari
IB IL 24 DI 8	1.1.1	I	BOOL	0.0		STD_CNCF ST
IB IL 24 DI 8	1.2.1	I	BOOL	0.1		STD_CNCF ST
IB IL 24 DI 8	2.1.1	I	BOOL	0.2		
IB IL 24 DI 8	2.2.1	I	BOOL	0.3		
IB IL 24 DI 8	3.1.1	I	BOOL	0.4		STD_CNCF ST
IB IL 24 DI 8	3.2.1	I	BOOL	0.5		
IB IL 24 DI 8	4.1.1	I	BOOL	0.6		
IB IL 24 DI 8	4.2.1	I	BOOL	0.7		
IB IL 24 DI 8	\~DI 8	I	BYTE	0.0		

Figure 7-5 All used variables and assigned process data

## 7.11 Setting the communication path



See also "Setting the communication path" on page 4-26.



- Switch to the bus configuration workspace to set the communication path.
- In the "Bus Structure" window, select the node for the controller.
- Select the "Communication" tab in the "Device Details" window.
- Select the "Simulation" interface type.
- Activate the EasySim simulation by clicking on "Apply".

## 7.12 Compiling and sending a project, and performing a cold restart



See also "Compiling and sending a project, and performing a cold restart" on page 4-40.

### Compile project



- Select the "Build, Make" command.

### Send project



- Open the project control dialog box.

The EasySim simulation is started when the project control dialog box is opened. This is shown in the taskbar with the following icon:

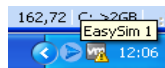


Figure 7-6 EasySim simulation: Icon in the taskbar

- Click on "Download" in the project control dialog box.
- In the "Download" dialog box that opens, activate the "Ensure real-time for Download Changes" and "Include Bootproject" checkboxes in the "Project" area.
- Click on "Download" in the area on the left.

When sending, the project is written to the main memory of the simulated controller.

- Open I/O view of the simulation by selecting the corresponding menu item in the context menu for the EasySim icon in the taskbar.

## Example project for a simulation with processor type "IPC" controllers

Only the states of the global variables that have been linked to a physical address (inputs/outputs) as part of process data assignment are displayed in the simulation.

	Name	Value	Preset 1	Preset 2	Preset 3	Data type	Description	Terminal
1	S1					BOOL	%IX0.0	
2	OUT1					BOOL	%QX2.0	
3	V0	16#0000				WORD	%QW0	
4	IN2					BOOL	%IX0.1	
5	IN3					BOOL	%IX0.4	
6	OUT2					BOOL	%QX2.5	

Figure 7-7 Status of the inputs and outputs



Using the "Preset..." column, the states of the input values can be preset (click on green LEDs). These can be activated using the corresponding buttons.

### Cold restart

Perform a cold restart as described in "Compiling and sending a project, and performing a cold restart" on page 4-40.

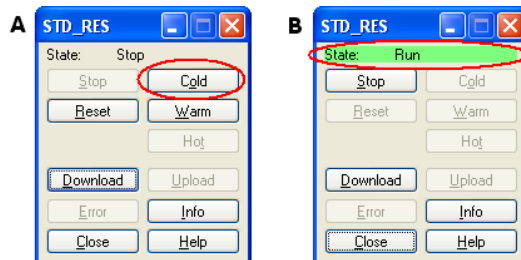


Figure 7-8 Performing a cold restart

The cold restart activates the project. The simulation can now be started.

## 7.13 Operation

### 7.13.1 Setting the task properties

If no settings have been made for the task, a default task is used. If necessary, change the properties of the task (in the example below a default task is changed to a cyclic task).



- Switch to the IEC programming workspace.
- Select "STD\_TSK : DEFAULT".
- In the context menu, select the "Properties..." menu item.

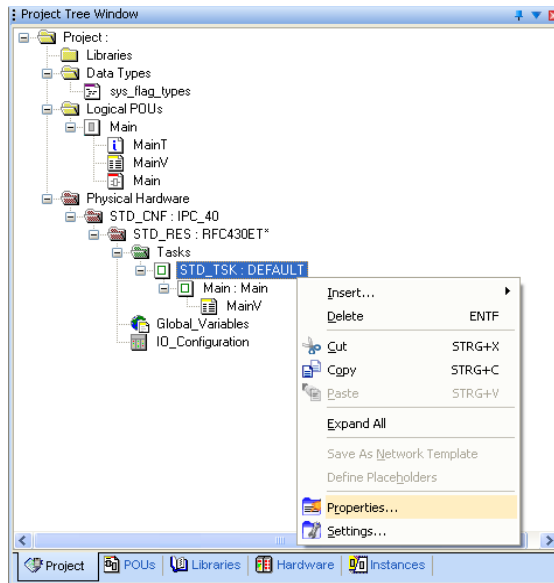


Figure 7-9 Standard task: Properties

- Under type, select "Cyclic". This selects a cyclic task.

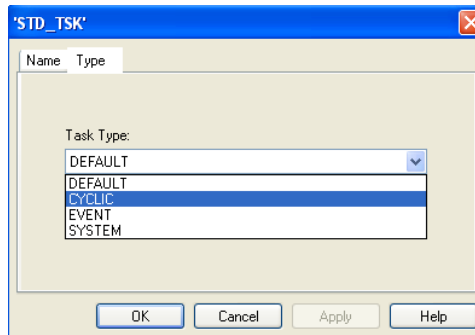


Figure 7-10 Standard task: Type

- Apply your setting with "Apply".



- Change the setting, e.g., to 1000 ms (see Figure 7-11 on page 7-11).

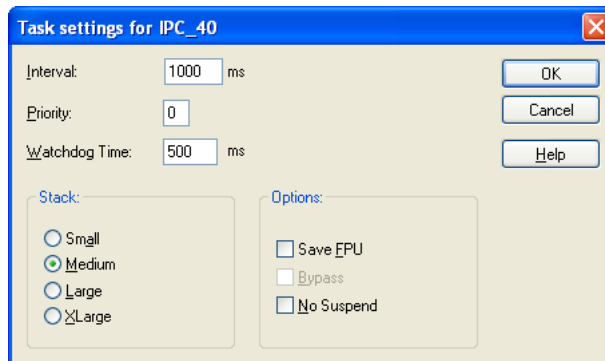


Figure 7-11 Changing the setting

- Confirm your settings with "OK".
- Compile the project, send it to the controller, and perform a cold restart.

If the process is now running too slowly, change the settings.

- In the standard task context menu, select the "Settings..." menu item.
- Change the setting, e.g., to 250 ms.
- Compile the project, send it to the controller, and perform a cold restart.

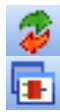
### 7.13.2 Simulation and debug mode

Following the cold restart, the animated arrows (see figure below) indicate that the simulation is running. The arrows "move" from left to right.



Figure 7-12 Simulation running

The program simulation can now be started. To monitor all variables, switch to debug mode. The status of the variables can be monitored online in debug mode.



- Activate debug mode.
- Switch to the IEC programming workspace.
- Switch to the window that provides the best display of the required information. The "MainV" window has been selected for the description below.
- Arrange the screen display so that the "MainV" and "EasySim – I/O View" windows can be seen.

The screenshot shows the software interface with the following components:

- Project Tree Window:** Shows a tree structure with folders like Libraries, Data Types, Logical POU's, Main, MainT, MainV, and Physical Hardware.
- Variable Table:** A table with columns: Name, Online value, Type, Usage, Description, Address, Init, Retain, PDD, OPC, TB, H. It lists variables like S1, T1, OUT1, T2, V0, IN2, IN3, OUT2, and TP\_1.
- EasySim 1 I/O View:** A window showing a table of I/O variables with columns: Name, Value, Preset 1, Preset 2, Preset 3, Data type, Description, Terminal. It includes buttons for 'Set Preset 1', 'Set Preset 2', 'Set Preset 3', and 'Expert Mode >>'.

Figure 7-13 Program output state

In the program output state, S1 = FALSE, all LEDs are off, and variable V0 is set to initial value "1" (16#0001).

## Example project for a simulation with processor type "IPC" controllers

In I/O view, click on the green LED for input S1. The following actions are performed (see Figure 7-14 on page 7-13):

- Variable S1 = TRUE (see S1 in the "MainV" window)
- The counter for the elapsed time, T2, is started (see T2 in the "MainV" window).
- Output 0 of IB IL 24 DO 8 (variable OUT1) is set.
- The chasing light is started (see change to V0 in the "MainV" window or in I/O view). Since only the physical outputs that have been linked to a variable as part of process data assignment are displayed in the simulation, the individual outputs of the IB IL 24 DO 16 cannot be shown in this example project. Display is by means of WORD variable V0.

The screenshot shows the EasySim 1 software interface. The top window displays a project tree on the left and a table of variables on the right. The bottom window shows the I/O view with a table of variables and their current states.

Name	Online value	Type	Usage	Description	Address	Init	Retain	PDD	DPC	TB	H
Default											
S1	TRUE	BOOL	VAR_EXTERNAL								
T1	15.000	TIME	VAR			T#15000ms					
OUT1	TRUE	BOOL	VAR_EXTERNAL								
T2	3.240	TIME	VAR								
V0	16#0008	WORD	VAR_EXTERNAL								
IN2	FALSE	BOOL	VAR_EXTERNAL								
IN3	FALSE	BOOL	VAR_EXTERNAL								
OUT2	FALSE	BOOL	VAR_EXTERNAL								
TP_1		TP	VAR								

	Name	Value	Preset 1	Preset 2	Preset 3	Data type	Description	Terminal
1	S1					BOOL		%IX0.0
2	OUT1					BOOL		%QX2.0
3	V0	16#4000				WORD		%QW0
4	IN2					BOOL		%IX0.1
5	IN3					BOOL		%IX0.4
6	OUT2					BOOL		%QX2.5

Figure 7-14 S1 has been activated

After 15 seconds the pulse output (variable OUT1) is set to FALSE, output OUT0 of the IB IL 24 DO 8 is switched off, and the chasing light is switched off (variable V0 stops with a random final value). Timer T2 has reached its final value of 15 seconds.

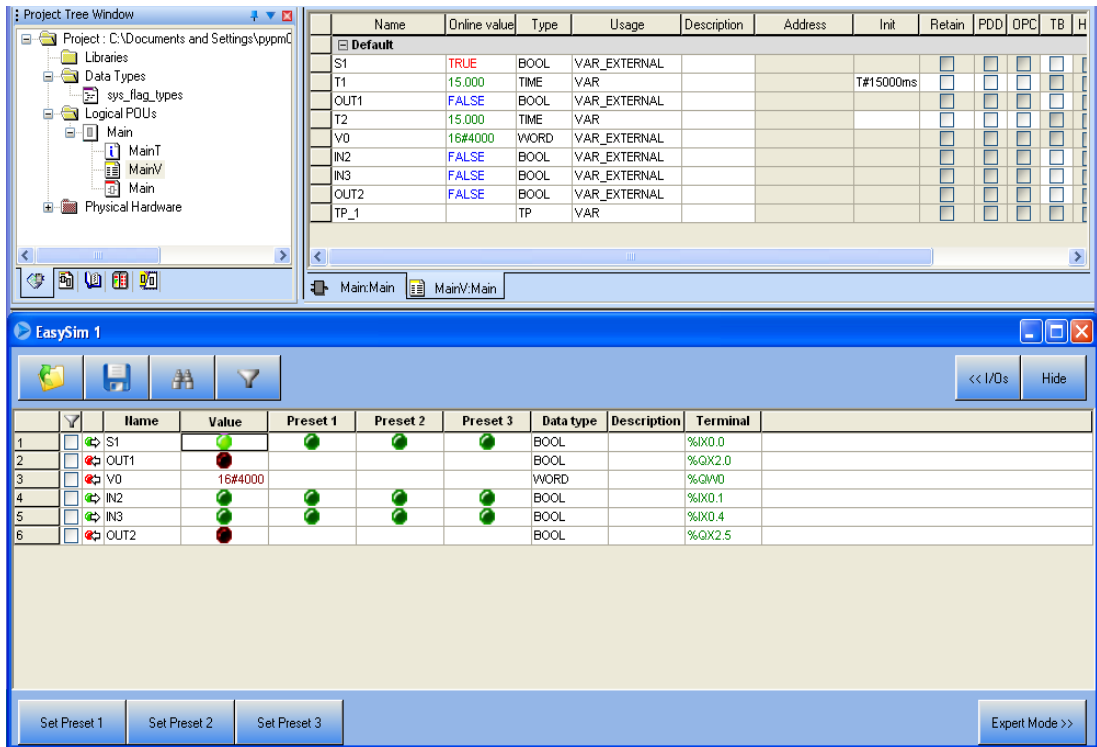


Figure 7-15 Program run complete

If input S1 is set to TRUE when the maximum time is reached, T2 remains set to 15000. As soon as S1 switches to the FALSE state, T2 is set to 0.

- Activate and deactivate S1 to check the validity of this program part.

The program part with the AND block operates independently of the program part with the pulse timer (see Figure 7-16 on page 7-15).

- Activate/deactivate the simulated LEDs (by clicking) that are assigned to inputs IN2 and IN3.

Output OUT5 of the IB IL 24 DO 8 module responds according to the result of ANDing.

Example project for a simulation with processor type "IPC" controllers

The screenshot displays the EasySim 1 interface. On the left is the Project Tree Window showing a project structure with folders for Libraries, Data Types, Logical POU's, and Physical Hardware. The main workspace shows a logic diagram with the following components and connections:

- TP\_1** (Timer): Inputs S1 (1) and T1 (15.000); outputs Q and ET (T2, 3.000).
- AND** gate: Inputs IN2 (1) and IN3 (1); output OUT2 (1).
- BOOL\_TO\_DINT** converter: Input OUT1 (1) from the Q output of TP\_1.
- ROL** (Rotate Left) register: Input IN (16#0020) from the output of the BOOL\_TO\_DINT converter; output N (16#0020).

Below the diagram is a data table with the following columns: Name, Value, Preset 1, Preset 2, Preset 3, Data type, Description, and Terminal. The table contains 6 rows of data:

	Name	Value	Preset 1	Preset 2	Preset 3	Data type	Description	Terminal
1	S1	1	●	●	●	BOOL	%IX0.0	
2	OUT1	1	●	●	●	BOOL	%QX2.0	
3	V0	16#0400	●	●	●	WORD	%QW0	
4	IN2	1	●	●	●	BOOL	%IX0.1	
5	IN3	1	●	●	●	BOOL	%IX0.4	
6	OUT2	1	●	●	●	BOOL	%QX2.5	

At the bottom of the interface, there are buttons for 'Set Preset 1', 'Set Preset 2', 'Set Preset 3', and 'Expert Mode >>'. The 'I/Os' button is also visible in the top right of the data table area.

Figure 7-16 ANDing in "Main"

## 7.14 Switching from simulation to real hardware

Once you have successfully completed the simulation for your program, you can switch to your hardware. To do this, the actual bus configuration must correspond to the simulated bus configuration.

- Stop debug mode. To do this, click on "Debug on/off".
- Stop the simulation. To do so, click on "Stop" in the project control dialog box.
- Close the project control dialog box.
- Exit the simulation, e.g., via the "Exit" menu item in the context menu for the simulation icon in the taskbar.



- Switch to the bus configuration workspace.
- In the "Device Details" window, switch to the "Communication" tab.
- Select the required communication path to enable you to communicate between PC WorX and your hardware (see "Setting the communication path" on page 4-26).
- Send the project to the controller and perform a cold restart (see "Compiling and sending a project, and performing a cold restart" on page 4-40).

## 8 Example program

A different example project is described in each of the previous sections. The various sections focus on the following topics:

- Section 5, "Example project for an INTERBUS system"
- Section 6, "Example project for a system consisting of PROFINET IO and INTERBUS"
- Section 7, "Example project for a simulation with processor type "IPC" controllers"

The procedure for creating the program in the software is the same. This is described in this section.

### 8.1 Program description

The program is created in function block diagram (FBD).

The program controls the following sequence:

On an edge change from FALSE to TRUE at input **IN**, a timer (**TP function block**) generates a pulse, which is output at output **Q** for the period of time **PT**. The associated output is set for this time. The state of the output is indicated at the corresponding status LED.

The elapsed time is output at output **ET**.

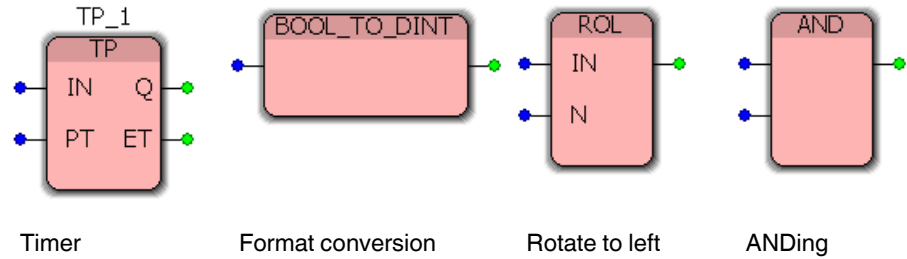
If IN switches from FALSE to TRUE for a second time while the pulse is still active (within PT), this does not affect the duration of the pulse generated at output Q. This signal is ignored.

If a TRUE signal is present at output **Q**, the **ROL function block** is activated, which rotates the operands associated with the **IN** input parameter bit-by-bit to the left. This means that for an initial value equal to 1, the associated outputs (e.g., OUT1 to OUT16 of a 16-bit module) switch in succession from FALSE to TRUE and back again. This state change is indicated by a "chasing light" at the corresponding status LEDs.

Once the pulse time has elapsed, the output that indicates the presence of the pulse and the chasing light are switched off.

Independent of this, inputs **IN1** and **IN2** are logically ANDed. IN1 and IN2 map the status of the inputs to which the toggle switches are connected. The result of ANDing is output at output **OUT** (OUT2).

## 8.2 Function blocks used



**TP timer function block**

Table 8-1 Timer TP\_1

Parameter	Variable name	Data type	Use	Initial value	Description
IN	S1*	BOOL	VAR_GLOBAL		Start: If a rising edge is detected, a pulse is generated.
PT	T1	TIME	VAR	T#15000ms	Pulse time Setting: 15 s = 15000 ms
Q	OUT1	BOOL	VAR_GLOBAL		Pulse output TRUE if IN = TRUE and ET < PT; FALSE if IN = FALSE or ET >= PT
ET	T2	TIME	VAR		Elapsed time

\* In this example, variable S1 is defined as a global variable to match the simulation example with the RFC 430 ETH-IB controller in Section 7. This means that the variable status can be displayed in the EasySim simulation.

In the examples in Section 5 and Section 6, a system variable is used for each of the onboard inputs of the ILC 390 PN 2TX-IB Inline Controller (ONBOARD\_INPUT\_BIT10). If you would like to use the example in this section for the projects in Section 5 and Section 6, do not connect input IN of the TP function block to variable S1, but to system variable ONBOARD\_INPUT\_BIT10 instead.

### VAR/VAR\_GLOBAL (VAR\_EXTERNAL)

If a variable will only be used internally and will not be assigned to a process data item, it can be declared as VAR. However, if you would like to monitor this variable in the simulation, declare it as VAR\_GLOBAL. This is why in the example program the variables that are linked to a process data item (input/output) are declared as VAR\_GLOBAL.

In the "Global\_Variables" worksheet, these variables are indicated as VAR\_GLOBAL in the "Usage" column (see Figure 4-63 on page 4-48) and as VAR\_EXTERNAL in the POU variables worksheet (see Figure 4-60 on page 4-47).

### BOOL\_TO\_DINT format conversion function block

To use the output signal of the TP\_1 block as the input signal for the ROL block, the format must be converted from BOOL to DINT.

Variables are not declared for this block.



**ROL rotation function block**

Table 8-2 Rotate to left ROL

Parameter	Variable name	Data type	Use	Initial value	Description
IN	V0	WORD	VAR_GLOBAL		Input value
N		DINT			Number of characters to be rotated (in the example 0 or 1, defined by OUT1)
(Output)	V0	WORD	VAR_GLOBAL		Output value ROL

**AND ANDing function block**

Table 8-3 AND ANDing function block

Parameter	Variable name	Data type	Use	Initial value	Description
IN1	IN2	BOOL	VAR_GLOBAL		Input value 1
IN2	IN3	BOOL	VAR_GLOBAL		Input value 2
OUT	OUT2	BOOL	VAR_GLOBAL		Output value AND

### 8.3 Programming

For programming, proceed as follows:



- Switch to the IEC programming workspace.
- Double-click on "Main" in the "Project Tree Window" to activate the IEC programming interface.
- In the "Main" worksheet, click on the empty position where you would like to insert the function block.

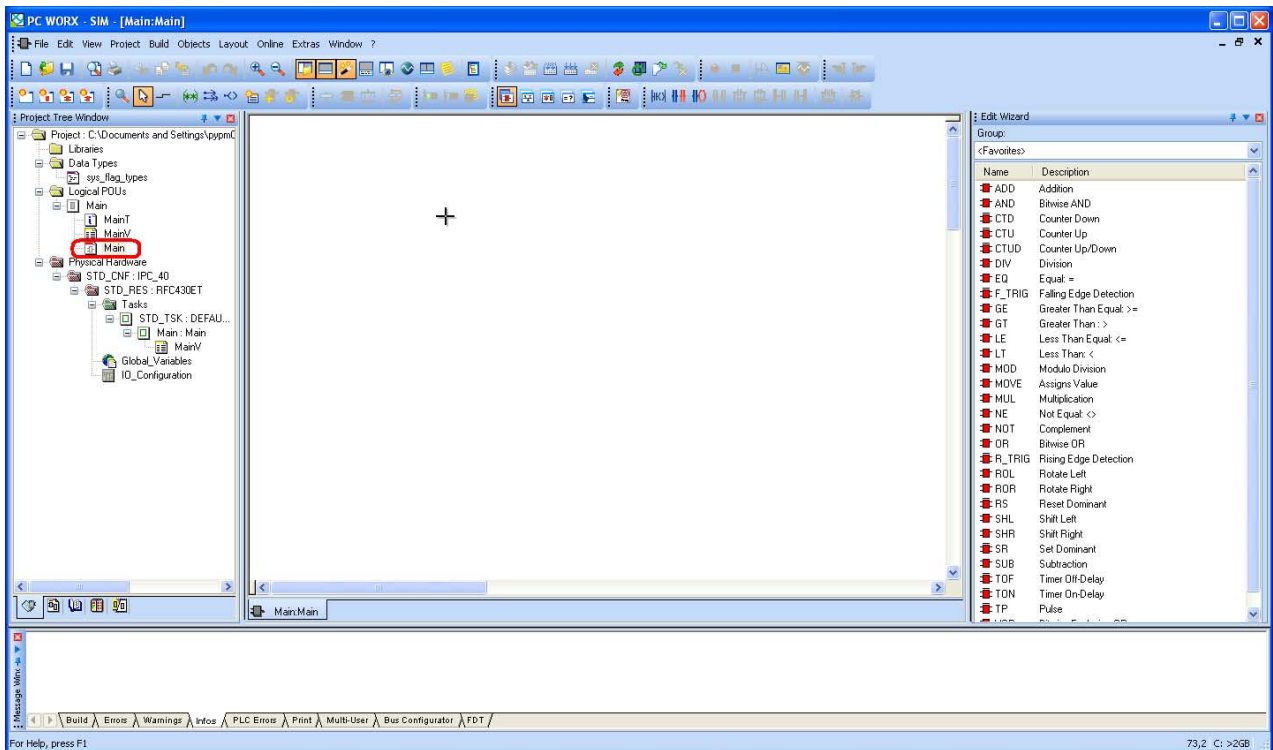


Figure 8-1 IEC programming workspace

- In the "Edit Wizard" window, select the "TP" function block by double-clicking on the corresponding function block.

- Enter the name "TP\_1" for this function block in the "Variable Properties" window and confirm your entry with "OK".

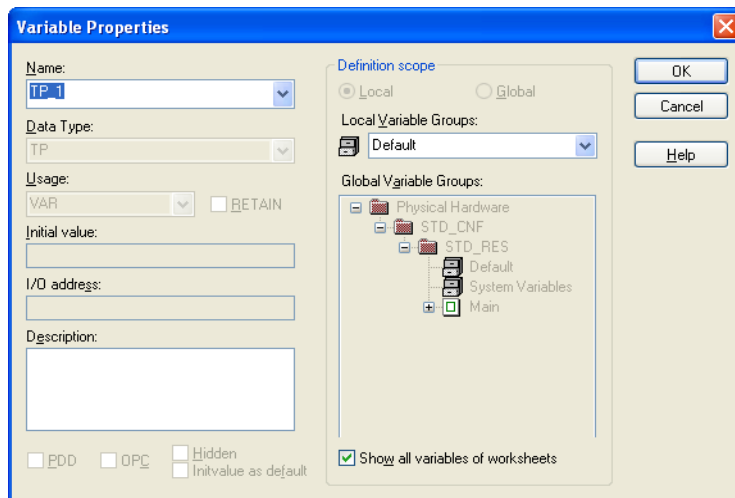


Figure 8-2 "Variable Properties" window

The function block is inserted in the worksheet.

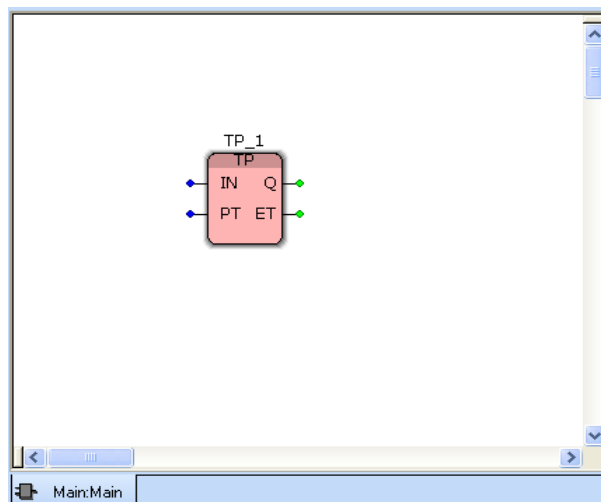


Figure 8-3 Function block in the IEC programming workspace

- Double-click on the "IN" input parameter of the function block to define the variable properties.
- In the "Variable Properties" window, enter or select the name for the "IN" input parameter under "Name".
- Enter the name S1.

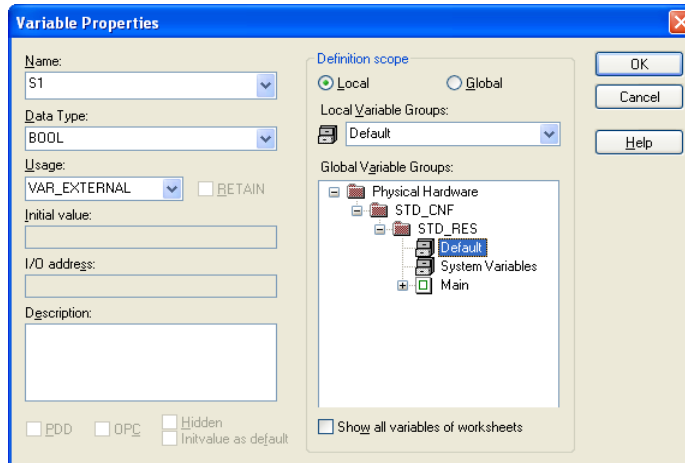


Figure 8-4 "Variable Properties" window: S1 (VAR\_GLOBAL)

- Specify the data type. The variable is a bit variable and therefore "BOOL" data type.
- Select "VAR\_GLOBAL" in the "Usage" selection box because the variable will later be assigned to a process data item.



If you have created a variable as VAR, and you would like to use this later as VAR\_GLOBAL, to change it select "VAR\_GLOBAL" in the "Usage" selection box and confirm your selection with "OK".

- Confirm your entries with "OK".
- Double-click on the "PT" input parameter of the function block to specify the variable properties.
- In the "Variable Properties" window, enter the name "T1".
- "TIME" is specified in the "Data Type" selection box.
- Select "VAR" in the "Usage" selection box because the variable will only be used internally and will not be assigned to a process data item.
- Enter a pulse width of 15 seconds (T#15000 ms) as the "Initial value".

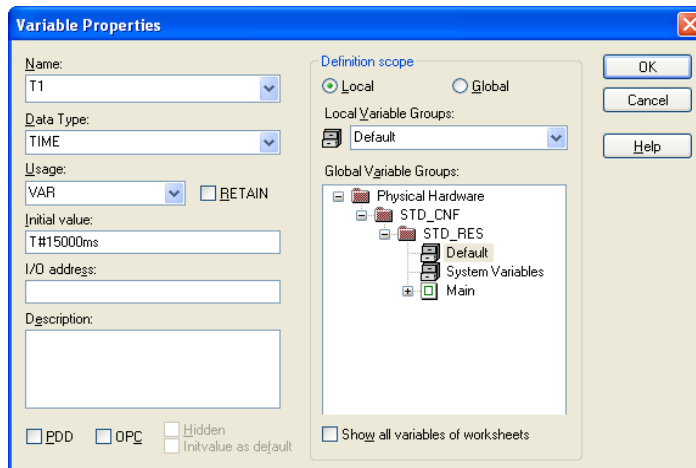


Figure 8-5 "Variable Properties" window: T1 (VAR)

- Confirm your entries with "OK".
- Proceed in the same way for the output parameters of this block. Use the values in Table 8-1 "Timer TP\_1" on page 8-2.

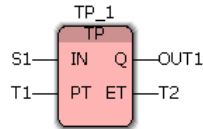


Figure 8-6 TP\_1 function block with all parameters

- Insert the BOOL\_TO\_DINT function block. A parameter declaration is not required for this block.
- Insert the ROL block.
- Declare the required parameters according to Table 8-2 "Rotate to left ROL" on page 8-3.
- Insert the AND block.
- Declare the required parameters according to Table 8-3 "AND ANDING function block" on page 8-3.

All function blocks are shown in Figure 8-7 with the required parameters.

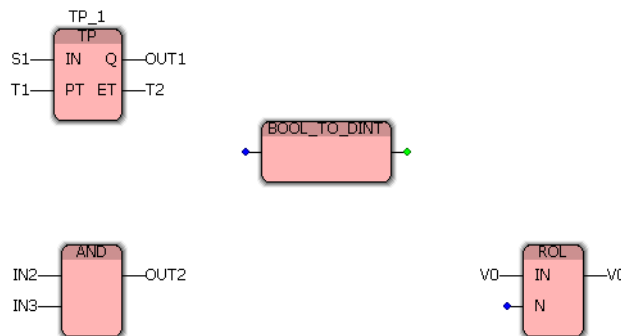


Figure 8-7 All function blocks inserted with the required parameters

Link the required inputs and outputs according to Figure 8-8 on page 8-8. To do this, proceed as follows:

- Select variable OUT1 and drag it to the same level as the BOOL\_TO\_DINT block.



- Activate the "Link" icon to link the objects.
- Click in the part of the line for output OUT1.
- Link this item to the input of BOOL\_TO\_DINT by dragging the mouse pointer onto the input and clicking on the input.



Alternatively, without having to activate the "Link" icon, you can click in the part of the line for output OUT1 and, holding down the left mouse button, drag the mouse pointer onto the input of BOOL\_TO\_DINT. Release the left mouse button as soon as the input of BOOL\_TO\_DINT is highlighted.

- Link the output of BOOL\_TO\_DINT to input N of the ROL block.

The finished program is shown in Figure 8-8 below.

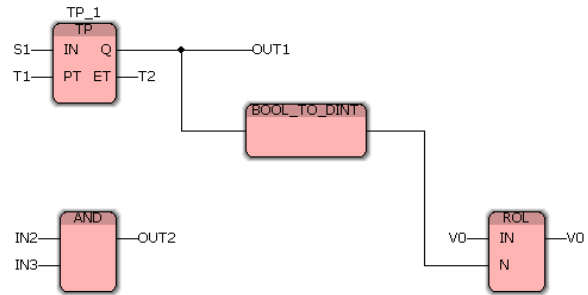


Figure 8-8 Finished program

**Deleted and renamed variables**

Variables created during program creation are created in different worksheets. If you do not rename or delete a variable in all the workspaces it appears in, this will generate error messages when compiling. Please follow the corresponding instructions.

## 8.4 Setting the initial value

If you would like to set an initial value, proceed as follows:



- Switch to the IEC programming workspace.
- Double-click on the "Global\_Variables" item.
- Set the initial value for the variables (here: V0 to initial value 1).

The screenshot shows the 'Global Variables' table in the software. The table has columns for Name, Type, Usage, Description, Address, Init, Retain, PDD, OPC, TB, and Hid. The variable V0 is highlighted, and its 'Init' value is set to 1, which is circled in red. The table also lists various system variables like PLCMODE\_ON, PLCMODE\_RUN, etc.

Name	Type	Usage	Description	Address	Init	Retain	PDD	OPC	TB	Hid...
Default										
S1	BOOL	VAR_GLOBAL				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
OUT1	BOOL	VAR_GLOBAL				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
V0	WORD	VAR_GLOBAL			1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IN2	BOOL	VAR_GLOBAL				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IN3	BOOL	VAR_GLOBAL				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
OUT2	BOOL	VAR_GLOBAL				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
System Variables										
PLCMODE_ON	BOOL	VAR_GLOBAL	PLC mode ON	%MX1....		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLCMODE_RUN	BOOL	VAR_GLOBAL	PLC mode RUN	%MX1....		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLCMODE_STOP	BOOL	VAR_GLOBAL	PLC mode STOP	%MX1....		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLCMODE_HALT	BOOL	VAR_GLOBAL	PLC mode HALT	%MX1....		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLCDEBUG_BPSET	BOOL	VAR_GLOBAL	Breakpoint set	%MX1....		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLCDEBUG_FORCE	BOOL	VAR_GLOBAL	Variable(s) forced	%MX1....		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLCDEBUG_POWERFLOW	BOOL	VAR_GLOBAL	Powerflow ON	%MX1....		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLC_TICKS_PER_SEC	INT	VAR_GLOBAL	System ticks per second	%MW1....		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLC_SYS_TICK_CNT	DINT	VAR_GLOBAL	System tick count	%MD1....		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLC_TASK_AVAILABLE	INT	VAR_GLOBAL	Number of available PLC tasks	%MW1....		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLC_TASK_DEFINED	INT	VAR_GLOBAL	Number of used tasks	%MW1....		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PLC_TASK_1	Extended_Task_I...	VAR_GLOBAL	Task 1 information	%MB1....		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 8-9 Setting the initial value (here: V0 = 1)

## 8.5 Additional options for PROFINET

The following additional data can be evaluated in a PROFINET IO system:

- For each PROFINET IO controller: System variables that indicate the configuration status
- For each PROFINET IO device: Pre-defined process data that indicates the status



See also "Status information for a PROFINET IO system" on page B-1.





# A Additional software functions

## A 1 Setting the realtime clock

In the "Extended Settings" tab, the time and date can be set for the internal system clock of the controller.

The current controller time is read and displayed every time the "Extended Settings" tab is opened. The display then indicates this value until the "Extended Settings" tab is opened again.

If you want to apply the system time and date from your PC:

Requirement: Connection between the PC and the controller.



- Switch to the bus configuration workspace.
- Select the controller node (e.g., "ILC 390 PN").
- Select the "Extended Settings" tab in the "Device Details" window.
- Click on the "System Time" button.
- Then immediately click on "Send" to transmit the data in the "Real-time Clock Settings" area to the controller.

If you would like to set different values:

- Enter the values for the time and date.
- Then click on "Send" to transmit the data in the "Real-time Clock Settings" area to the ILC 390 PN.

Real-time Clock Settings

Time: 14:43:38 System Time

Date: Mittwoch , 25. Mai 2011 Send

Figure A-1 Realtime clock settings

## A 2 Options for modifying a project

PC WorX provides two functions that can be used to send project modifications to the controller(s) and activate them there without having to stop the active PLC.

Table A-1 Options for modifying a project

Options:	Download changes	Online modifications
<b>Special features:</b>	Send the entire project (also known as "Download Changes")	Only send the modifications that have been made, e.g., individual function blocks
<b>Sequence:</b>	Debug mode ON: An error is detected Debug mode OFF Modify Compile Send via project control dialog box	
<b>Application:</b>	Extensive modifications	Minor modifications
<b>Advantage:</b>	Fewer restrictions	Fast

### A 2.1 Download Changes



For details of the controllers for which the "Download Changes" function is available, please refer to the "AH EN BASIC DATA ILC... / RFC... / FC..." application note.

#### Download

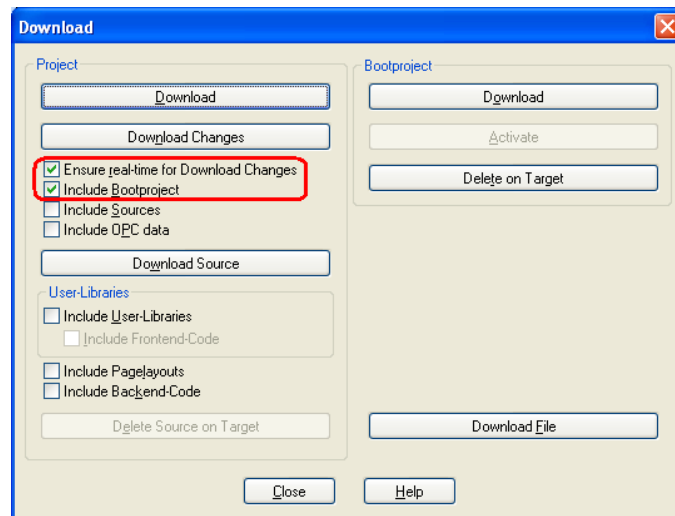


Figure A-2 Download changes



With the "Download Changes" function, the retain data is only retained if the current project has been saved to the parameterization memory. Therefore, when sending activate "Include Bootproject" (see Figure A-2 on page A-2) to avoid inconsistencies. For additional information about retain handling, please refer to "Extended retain handling" on page A-8.

When downloading changes to one or more controllers, it is possible to specify whether the system should ensure the realtime behavior of the PLC before starting the download process.

If the "Ensure real-time for Download Changes" option is activated, the system attempts to execute the "Download Changes" function without violating the realtime conditions of the tasks executed on the controller.

If the modifications cannot be sent, for example, because the amount of data to be copied is too large, an error message is output. In this case, disable the "Ensure real-time..." option (i.e., permit realtime violations) so that all modifications can be executed.

**NOTE: Realtime violations – Consider the effects**

If realtime violations are permitted, first ensure that your PLC application is running within safe limits. Please note that each realtime violation may result in unexpected consequences in the automation system, as under certain circumstances user tasks are not processed for a short time. Therefore, check the level of risk before disabling this option.



For more detailed information about the "Download Changes" function and ensuring realtime with detailed background information, please refer to the online help for PC WorX.

To determine when the "Download Changes" function can or cannot be used, please refer to the table below.

Table A-2 Application options and restrictions for "Download Changes"

Area	Can be used after	Do not use after
POU	<ul style="list-style-type: none"> <li>– Modifying the code in existing POUs</li> <li>– Using new POU types in a resource</li> </ul>	<ul style="list-style-type: none"> <li>– Using an MSFC POU in the project</li> </ul>
	<div style="border: 1px solid black; padding: 5px;"> <p>In principle, modifications in SFC POUs are permitted. However, they can result in critical PLC states if the "Download Changes" function is executed when the PLC is running. For example, the execution of an SFC step chain could be blocked if an active step is deleted. These SFC modifications include:</p> <ul style="list-style-type: none"> <li>– Modifying a previous/subsequent relationship.</li> <li>– Modifying step properties (type).</li> <li>– Modifying step/action relationships.</li> <li>– Modifying qualifiers (action qualifiers).</li> </ul> <p>In these cases, a warning is generated. Check the existing level of risk before continuing the process by confirming the warning.</p> </div>	
Variables	<ul style="list-style-type: none"> <li>– Adding/deleting local and global variables</li> <li>– Modifying local and global variables (including initial value)</li> </ul>	

Table A-2 Application options and restrictions for "Download Changes" (Continued)

Area	Can be used after	Do not use after
Data types	<ul style="list-style-type: none"> <li>- Modifying the data type of variables, as a result of which the variables are reinitialized</li> <li>- Adding/deleting data types</li> <li>- Modifying data type declarations. Modified data variables are initialized.</li> <li>- Modifying the data area</li> </ul>	
Function blocks	<ul style="list-style-type: none"> <li>- Declaring new function block instances (user/firmware)</li> </ul>	
Program	<ul style="list-style-type: none"> <li>- Modifying comments in the program code</li> <li>- Adding/deleting program instances</li> </ul>	
Tasks	<ul style="list-style-type: none"> <li>- Modifying task cycle times</li> <li>- Modifying watchdog times for tasks</li> </ul>	<ul style="list-style-type: none"> <li>- Adding or deleting tasks</li> <li>- Modifying task properties other than the cycle time and the watchdog time (i.e., task type, priority, event number, SPG assignment, stack size)</li> </ul>
I/O configuration	<ul style="list-style-type: none"> <li>- Modifying driver parameters in the I/O configuration</li> <li>- Adding/deleting an I/O group in the I/O configuration</li> </ul>	<ul style="list-style-type: none"> <li>- Modifying VAR_CONF groups in the I/O configuration</li> <li>- Modifying I/O group properties other than the driver parameters (e.g., addresses, the driver itself, task assignment, etc.)</li> </ul>
Other	<ul style="list-style-type: none"> <li>- Adding new configurations and resources</li> </ul>	

If a restriction is not observed, the system displays an error message in the "Message Window". Call the help information for the error message by right-clicking on the entry in the "Message Window" and selecting the "Message Help" context menu item. Undo the relevant modification, rebuild the project, and execute "Download Changes" again.



For more detailed information, please refer to the online help for PC WorX.

## A 2.2 Online modifications

"Online modification" of a POU means that in a single step the modifications in the project are compiled and the code is generated and automatically sent to the active PLC. During the entire process the PLC continues running, i.e., the program executed on the PLC is not interrupted during compilation and transmission.



The "Online Modifications" menu item is only available in offline mode. The affected worksheet can be set to online and offline mode via the "Debug on/off" icon in the toolbar.

"Online Modifications" can be called via the "Build, Online Modifications" menu item or via the corresponding icon (see "Frequently used icons for compiling and debugging" on page 3-4).

The "Online Modifications" command can be used in two ways:

1. Correcting errors in the code.  
If a programming error is discovered in online mode and you have switched to offline mode to correct the error, the modifications made can be compiled with "Online Modifications".
2. Further development of a basic project.  
In some cases it is easier and more convenient to develop a project using online modifications rather than via the "Make" command. Once a basic project has been compiled for the first time with "Make", this "skeleton project" can be completed by editing the individual POU code worksheets and using "Online Modifications" to add these extensions to the PLC program, which is already running on the target system.

In both cases, the modifications are sent automatically to the PLC without interrupting PLC operation, i.e., the modifications can then be seen in online mode.

To determine when the "Online Modifications" function can be used, please refer to the table below.

Table A-3 Application options and restrictions for "Online Modifications"

Area	Can be used after	Do not use after
Variables	<ul style="list-style-type: none"> <li>- Adding new local and global variables</li> <li>- Deleting variables which are not used in the CSV file (*.csv)</li> <li>- Adding global variables which are not on direct inputs and outputs</li> <li>- Removing variables from the program worksheet</li> <li>- Adding a user-defined function block*</li> </ul> <p>* Requirement: An instance of this function block has already been declared in a variable worksheet of any POU of the same resource</p>	<ul style="list-style-type: none"> <li>- Modifying variable properties</li> <li>- Modifying variable initial values (initialized variables cannot be modified online)</li> </ul>
Functions	<ul style="list-style-type: none"> <li>- Adding new IEC-defined functions</li> <li>- Adding a user-defined function*</li> <li>- Adding a function from a firmware or user library*</li> </ul> <p>* Requirement: This function has already been called in any POU of the same resource.</p>	<ul style="list-style-type: none"> <li>- Modifying the formal parameters of functions</li> <li>- Adding new formal parameters</li> </ul>
Function blocks	<ul style="list-style-type: none"> <li>- Adding new IEC-defined function blocks</li> <li>- Adding a user-defined function block*</li> <li>- Adding a function block from a firmware library or a user library*</li> </ul> <p>* Requirement: An instance of this function block has already been declared in a variable worksheet of any POU of the same resource</p>	<ul style="list-style-type: none"> <li>- Modifying the formal parameters of function blocks (VAR_INPUT, VAR_OUTPUT, and VAR_IN_OUT)</li> <li>- Adding new formal parameters</li> </ul>
Other	See Table A-4 "Rules for the specific programming languages" on page A-7	<ul style="list-style-type: none"> <li>- Modifying character strings/constants or user-defined character strings</li> <li>- Deleting POUs or libraries</li> <li>- Modifying the hardware structure</li> </ul>

Please note the following when modifying functions and function blocks:



For online modifications to be possible in the above cases, the project must have been compiled with "Make" and sent to the target system. In addition, online mode must already be active.

Table A-4 Rules for the specific programming languages

Language	Online modification possible after
All	<ul style="list-style-type: none"> <li>- Adding line breaks and blank lines</li> <li>- Adding or modifying comments</li> </ul>
IL	<ul style="list-style-type: none"> <li>- Adding or modifying IL operators</li> <li>- Modifying the nesting depth</li> </ul>
ST	<ul style="list-style-type: none"> <li>- Modifying instructions or expressions</li> </ul>
FBD	<ul style="list-style-type: none"> <li>- Modifying existing networks</li> <li>- Modifying graphics, e.g., moving lines or function blocks</li> <li>- Networks have been modified or added</li> </ul>
LD	<ul style="list-style-type: none"> <li>- Modifying existing networks</li> <li>- Modifying graphics, e.g., moving lines or function blocks</li> <li>- Networks have been modified or added</li> </ul>
SFC	<ul style="list-style-type: none"> <li>- Modifying variables, FBD or LD networks which are directly connected</li> <li>- Modifying time intervals for time qualifiers</li> <li>- Modifying variable names in action blocks</li> </ul>



Use "Make" if online modifications cannot be implemented.  
 If some of the restrictions are not observed, the online modification process is aborted and a corresponding error message appears in the "Message Window".  
 Once the "Online Modifications" command has been aborted, compile the project with "Make", and send it manually to the controller. Modifications can now be implemented online again.



For more detailed information, please refer to the online help for PC WorX.

## A 3 Extended retain handling



Retentive variables are declared with the keyword "RETAIN". The value of a retentive variable is retained even if the controller is switched off. The last value of the variable is used for a warm start.

**Application:** Your application program is already running on the controller. Modifications to the retentive variables are now required or new RETAIN variables must be added.

Without the retain handling function, a **cold restart** would have to be carried out after the modifications. In this case, all the variables are reinitialized.

With the retain handling function of the programming system, a controller cold restart is no longer required after the retentive variables have been modified. Instead, a **warm start** is possible.

The retain handling function receives the value of all retentive **symbolic** variables, by uploading them before the modified and newly compiled project is sent to the controller. Once the modified program has been sent to the controller, the system sends the previously saved variable values back to the relevant memory addresses.

Please note:

- When adding new retentive variables:  
Each newly added variable is initialized with the assigned initial value on a warm start. If an initial value was not assigned when the new variable was declared, it is initialized with the default value for the relevant data type.
- When modifying existing retentive variables:  
The retain handling function identifies each retentive variable using its instance path, variable name (not case-sensitive), and data type. If any one of these three items has been modified, the retentive variable is considered a newly declared variable and initialized (either with the declared initial value or with the default value, as described above).  
Please note that modifications within a user-defined data type result in the initialization of all variables of this data type. Example: Following the modification of the data type of a structure, all elements in this structure are initialized (see above).
- When deleting retentive variables:  
Deleted retentive variables are not relevant for the retain handling function. However, it may be that the remaining retentive variables are moved to the memory image of the controller.
- When moving retentive variables:  
Please note that variables can be moved to the memory image of the controller if variables are added, deleted, and moved to the variables table.



With the "Download Changes" function, the retain data is only retained if the current project has been saved to the parameterization memory. Therefore, when sending activate "Include Bootproject" (see Figure A-2 on page A-2) to avoid inconsistencies.



For more detailed information, please refer to the online help for PC WorX.



## A 4 Simulation

For processor type "IPC" controllers (RFC 4xx and S-MAX 4xx), PC WorX can be used to develop a project completely without hardware and to simulate operation. If you then install the hardware according to the simulated configuration, you can start up your project on the hardware by changing the communication path.

For an example of the simulation, please refer to Section 7, "Example project for a simulation with processor type "IPC" controllers".

## A 5 Comparing projects

The "Project, Compare Projects" menu can be used to compare two **compiled** projects.

- Open the first of the projects to be compared.
- In the "Project" menu, select the "Compare Projects" menu item.
- In the dialog box that opens, open the second of the projects to be compared.
- Open the project comparison workspace.

All the differences are displayed in table format in the "Project Comparison Result Window".

## A 6 Visualization

The data generated in PC WorX can be used in other programs, e.g., to visualize processes.

In order to use the data, AX OPC Server and/or a visualization software program are required.



For more detailed information about visualization and AX OPC Server, please refer to "AX OPC Server and WebVisit" on page C-1.

## A 7 Replacing a controller (hardware replacement)

To simply replace one controller with another, a wizard is provided for replacing complex devices. It may be necessary to replace the controller, e.g., if you have carried out the simulation of a project with an RFC 450 ETH-IB, but would like to use an ILC 370 PN 2TX-IB for the actual project.



- Switch to the bus configuration workspace.
- In the "Bus Structure" window, select the controller to be replaced (RFC 450 ETH-IB in Figure A-3 on page A-10).
- Open the context menu for the controller.
- Select the "Replace..." menu item.

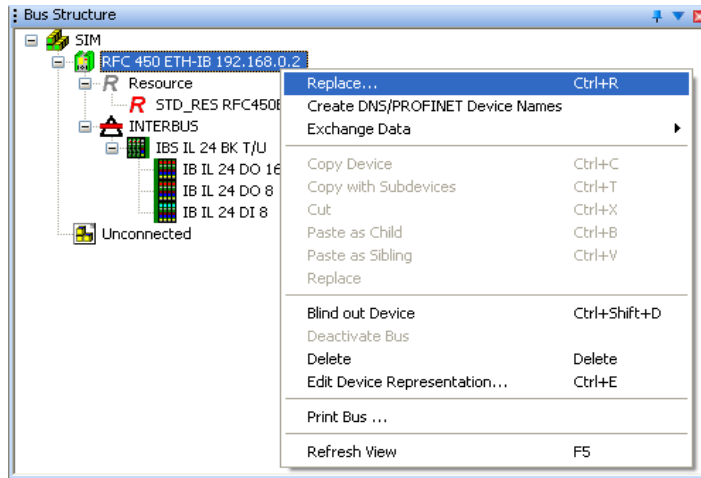


Figure A-3 Replacing a device

The wizard opens.

- Confirm the window that opens with "Next".

The window displays a selection of all the controllers that can replace the controller currently used in the project.



If the window is empty, the controller that is currently being used cannot be replaced with another one.

- Select the new controller to be used (e.g., ILC 370 PN).

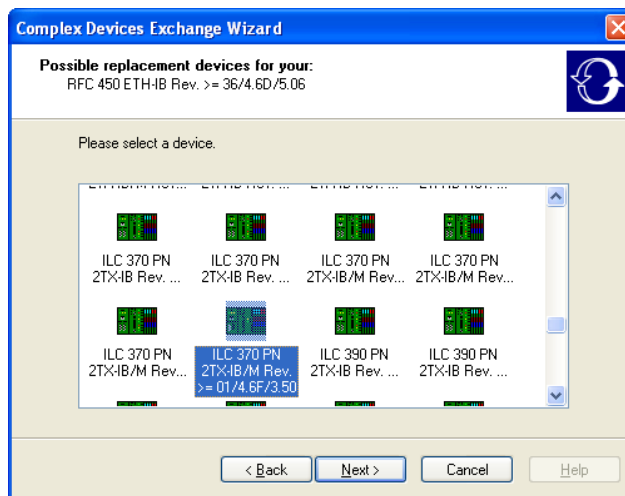


Figure A-4 Selecting the new controller

- Confirm your selection with "Next".

The wizard replaces the device.

Depending on which controllers are replaced with one another, the bus topology may have to be adapted. In addition, it may be necessary to insert additional devices.

In this example, a bus coupler (IBS IL 24 BK-T/U) is connected to the RFC 450 ETH-IB. This cannot be simply connected to the ILC 370 PN 2TX-IB. In order to continue using the bus coupler, a terminal with remote bus branch (e.g., IBS IL 24 RB-T) must be connected to the ILC 370 PN 2TX-IB.

If this type of adaptation is required, a window opens with a list of devices recommended for the adaptation.

- Select the device that you would like to use for the adaptation.

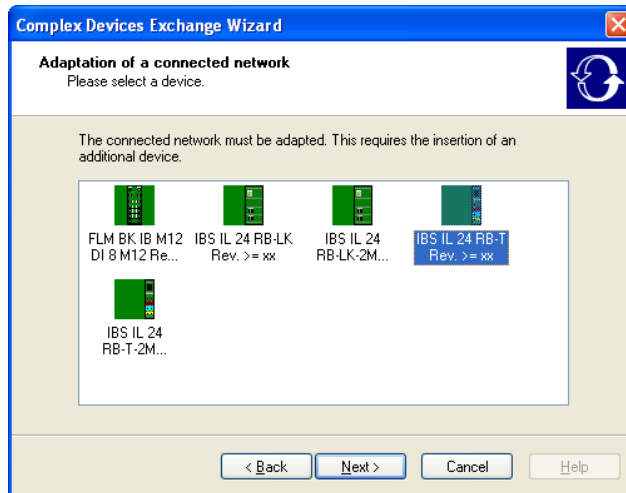


Figure A-5 Selecting the device to adapt the structure

The wizard indicates whether the controller was replaced successfully and also displays special information.



Figure A-6 Completing replacement

- Confirm replacement with "Finish".

The bus configuration with the new controller is displayed.

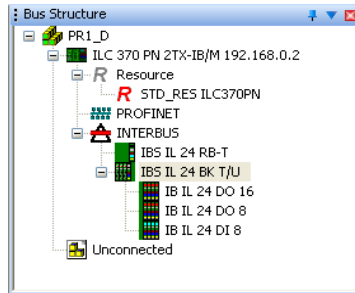


Figure A-7 Bus configuration with new controller

In the example, an INTERBUS controller was replaced with a PROFINET IO controller. When comparing the bus topology in Figure A-3 and Figure A-7, you can see that the structure has been adapted accordingly as a result of replacing the controller. The dedicated INTERBUS structure has been replaced by a PROFINET structure.

In this example, it would be better to connect the Inline terminals directly to the Inline Controller instead of using a branch terminal and a bus coupler.

- In this case, adapt the bus configuration.
- Move the Inline I/O terminals directly to the INTERBUS node of the controller.
- First delete the bus coupler (IBS IL 24 BK-T/U) and then the branch terminal (IBS IL 24 RB-T).

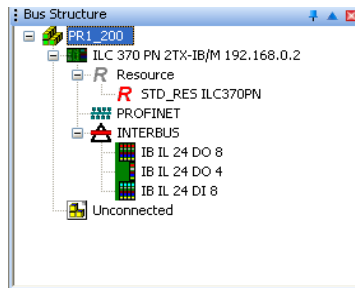


Figure A-8 Manually adapted bus configuration with new controller

**Adapting the standard configuration resource**

In rare cases, the following error message may appear in the "Message Window" following controller replacement: "Resource: Invalid configuration type for resource "STD\_CNF.STD\_RES". The current type is "IPC\_40". However, type "ARM\_L\_40" is expected.", if so proceed as follows:



- In this case, first adapt the configuration then the resource.
- Switch to the IEC programming workspace.
- Open the context menu for the configuration and select the "Properties..." menu item.

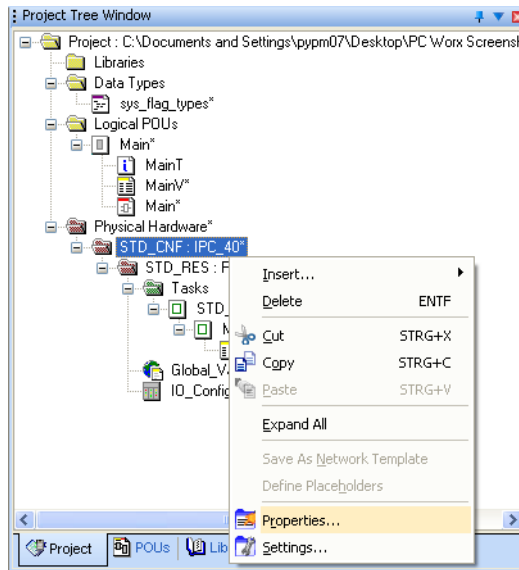


Figure A-9 Configuration context menu: Properties

- Switch to the "Plc/Processor" tab.
- Select the corresponding processor type for the controller. In the example this is ARM\_L\_40.

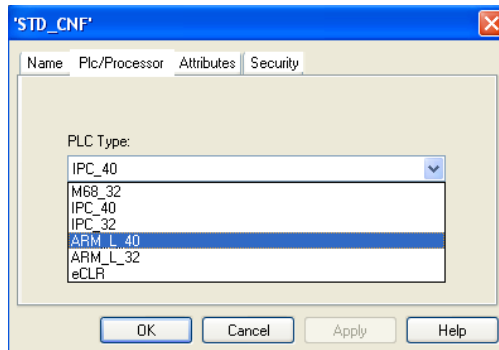


Figure A-10 Selecting the processor type

- Confirm your selection with "Apply".

A message then informs you that under certain circumstances other parts of your project may have to be adapted.

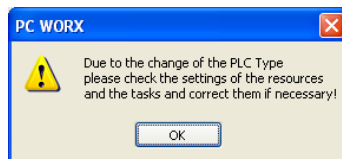


Figure A-11 Warning

- Confirm the warning with "OK".
- Confirm the processor type selection with "OK".
- Open the context menu for the resource and select the "Properties..." menu item.

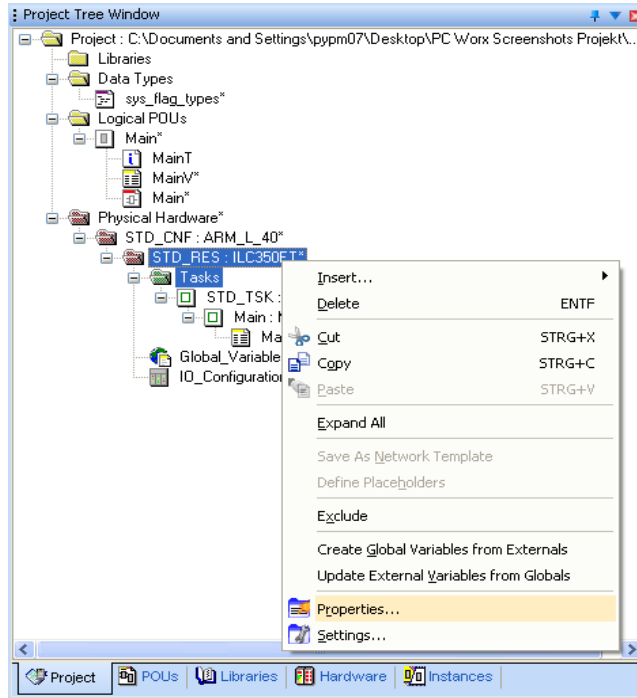


Figure A-12 Resource context menu: Properties

- Switch to the "Plc/Processor" tab.
- Select the corresponding processor type for the controller. In the example this is ILC370PN.

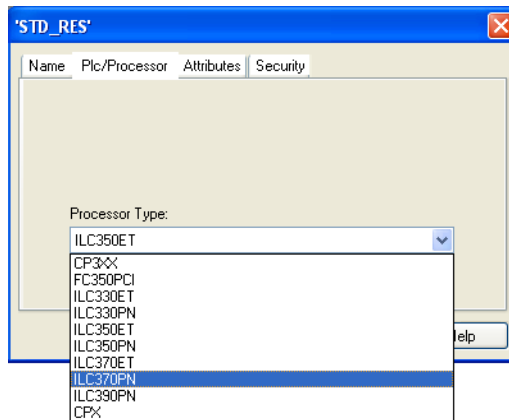


Figure A-13 Selecting the processor type

- Confirm your selection with "Apply".

A message then informs you that under certain circumstances other parts of your project may have to be adapted.

- Confirm the warning with "OK".
- Confirm the processor type selection with "OK".
- Compile the project to detect any errors.
- If special POU's have been used, adapt them accordingly.
- Compile the project.

## A 8 Device description files

Device description files are FDCML or GSD files, which provide a complete description of a device. If the device catalog does not contain a corresponding device description file, it must be imported.

### A 8.1 Phoenix Contact device description files

When PC WorX is reinstalled, the device description files for the devices available at the time are also installed. These descriptions are also found in the device catalog. When installing PC WorX service packs, new device description files are installed in folder "...FDCML10\...\Phoenix Contact". These new descriptions are not included in the device catalog yet. They must be imported as necessary.

- Select "Phoenix Contact" in the "Device Catalog" window in PC WorX.
- Open the context menu and select "Import Device...".

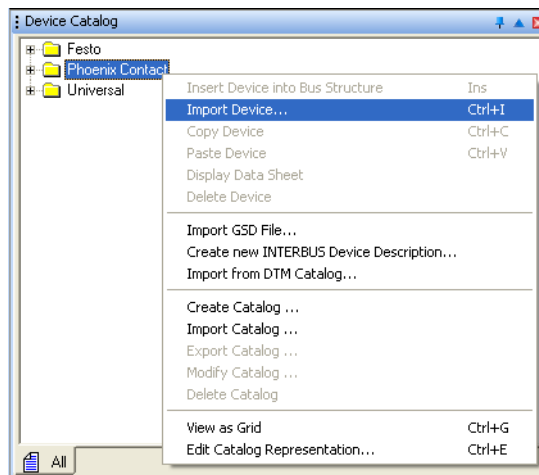


Figure A-14 Device Catalog: Import Device

- Select the device description file. If you have used the suggested standard installation, the files are located in directory:  
**C:\Documents and Settings\All Users\Shared Documents\FDCML10\xxx\Phoenix Contact**  
xxx = System (e.g., ETHERNET, INTERBUS, PROFINET)



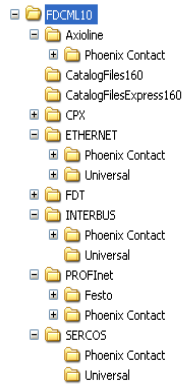


Figure A-15 Directory for the device description files

- Confirm your selection.

The "Message Window" indicates whether the device has been imported successfully.

## A 8.2 Device description files for Inline Controllers

If you have used the suggested standard installation, the device description files together with all configuration options for Inline Controllers are located in directory:

**C:\Documents and Settings\All Users\Shared Documents\FDCML10\INTERBUS\Phoenix Contact\modification**

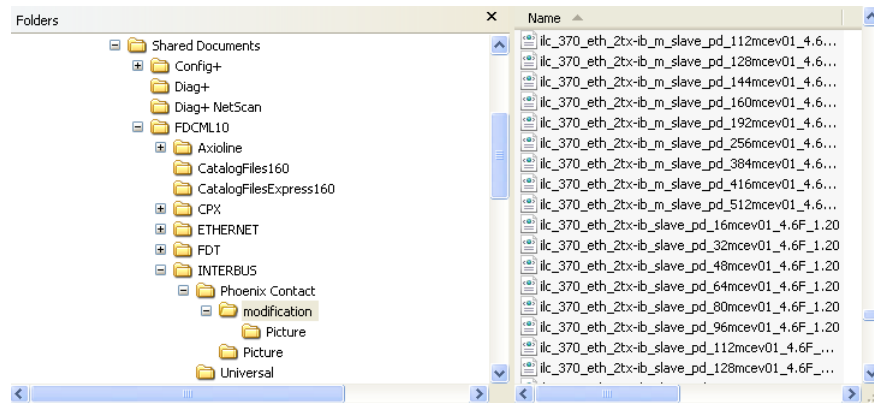


Figure A-16 Directory for the device description files

- Copy the required device description file to directory:  
**...FDCML10\INTERBUS\Phoenix Contact**
- Proceed as described above.

### A 8.3 Device description files from other manufacturers (GSD files)

GSD files can be used to integrate devices from manufacturers other than Phoenix Contact in PC WorX.

- Copy the GSD file for your device, e.g., from the Internet to your PC hard disk.
- Select "Phoenix Contact" in the "Device Catalog" window in PC WorX.
- Open the context menu and select "Import GSD File..."

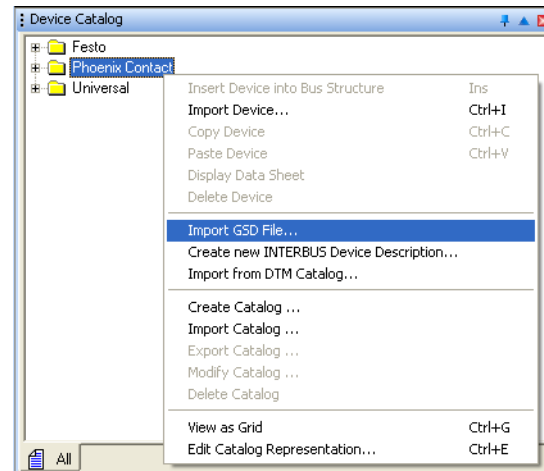


Figure A-17 Device Catalog: Import GSD File

- Select the device description file and confirm with "Open".

PC WorX creates an entry for the device in the device catalog. This entry is located in directory: C:\Documents and Settings\All Users\Shared Documents\FDCML10\... (for standard installation).

The "Message Window" indicates whether the device has been imported successfully.

When reading a modular device, only the device, and not its modules, is displayed in the bus configuration.

Manually insert the modules for these devices. To do this, proceed as follows:

- Open the module catalog via "View, Module Catalog".
- Select the device modules one at a time from the module catalog and drag and drop them in the lower level below the device entry.

### A 8.4 Creating device description files

If a device description file is not available, one can be created. For this, a device description editor is included on the AUTOMATIONWORX Software Suite DVD.

## A 9 Ethernet topology

PC WorX can be used to automatically detect Ethernet topologies. All the Ethernet devices in the project can be displayed with the connections in the "Ethernet Topology" window. For all devices with LLDP functions, the connections are displayed automatically with the port number.

For this, there must be an online connection between PC WorX and a complete bus configuration.



For more detailed information about Ethernet topology view, please refer to the online help.

Ethernet topology view offers the following display options, which can be selected in the context menu:

### Device context menu

Device Properties	Call the "Device Properties" window via the context menu of a device. Properties include, e.g., IP Address, DNS/PROFINET Device Name, Device Name, Firmware Version, etc. . If required by your application, you can adapt the device properties using this dialog box.
Web Interface	Call the web interface of a device in the browser, if available (e.g., for a switch), via the context menu of the device in question.

### Topology context menu

Topology	You can update, delete, and read the Ethernet topology in again manually or in adjustable cycles. In addition, you can save a desired topology status or reopen a saved one.
Online View	The availability of devices can be shown online. Devices in the network that cannot be accessed are indicated.  Devices that can be accessed are indicated with a green dot, devices that cannot be accessed are indicated with a red dot with a white cross. Any changes with regard to availability are displayed when the view is refreshed.
Link Mode	In link mode, the connections between connected devices can be marked if they are not displayed.
View	The display in the "Ethernet Topology" window can be enlarged/reduced, printed, and product information can be displayed.
Magnifier	The magnifier function enables you to enlarge/reduce the size of sections of the network in order to provide a detailed view of individual sections or a better overview of extensive network sections.

The connection information is displayed by clicking on a connection or port number in the topology.

For a clearer view of extensive networks, switches that have a lower-level network branch are displayed in a tab. To display a different network branch, click on the tab for the corresponding switch.

## A 10 INTERBUS topology

PC WorX can be used to automatically detect INTERBUS topologies. The INTERBUS structure can be viewed in the "INTERBUS Topology" window.

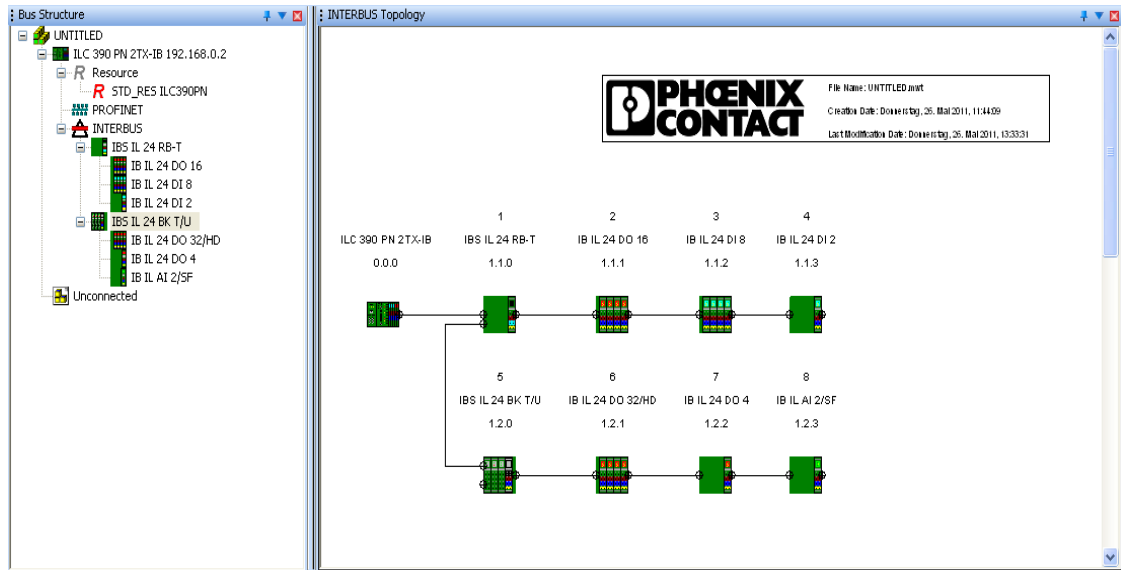


Figure A-18 INTERBUS topology for an INTERBUS system



If this function is used with PROFINET, only the INTERBUS topology is displayed that is connected to the INTERBUS node for the controller.

To print the entire bus topology, another function can be used in PC WorX:

- In the bus configuration, activate the context menu via any element.
- Select the "Print Bus Configuration" menu item.

## B Status information for a PROFINET IO system

### B 1 Status of the PROFINET IO controller

The PROFINET IO controller has status information that indicates its configuration status. This can be requested via the PNIO\_CONFIG\_STATUS system variable. The system variable is available as a word and as individual bits under "Global\_Variables".

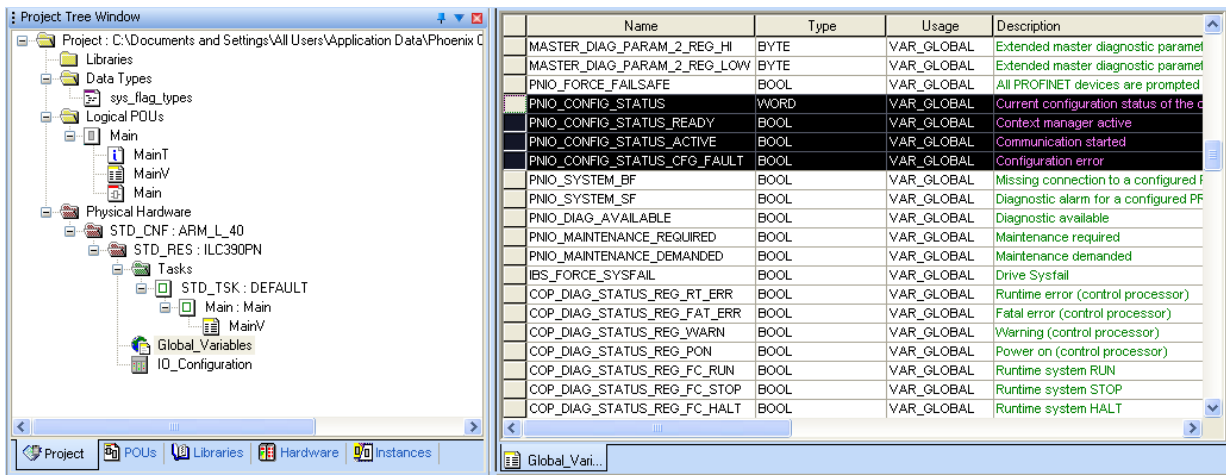


Figure B-1 PNIO\_CONFIG\_STATUS system variable

#### PNIO\_CONFIG\_STATUS\_READY

The variable is set if the PROFINET IO controller has been initialized correctly. No desired configuration has been loaded by PC WorX yet. Connection establishment to the devices has been completed.

#### PNIO\_CONFIG\_STATUS\_ACTIVE

The variable is set if the desired configuration for the PROFINET IO controller has been loaded. In this state the PROFINET IO controller attempts to establish a connection cyclically to all devices in the desired configuration (under the PROFINET icon). In this way, it attempts to find each device with the corresponding "PROFINET device name" in the current subnetwork.

If the connection to a PROFINET IO device has been established successfully, the corresponding "PNIO\_DATA\_VALID" process data item is set.

For inaccessible devices, the PROFINET IO controller cyclically attempts to establish a connection approximately every 5 seconds.

A connection cannot be established, e.g., if the corresponding device is ready, but a correct PROFINET device name has not yet been assigned to it.

#### PNIO\_CONFIG\_STATUS\_CFG\_FAULT

This system variable is set if an error occurred when configuring the PROFINET IO controller.

## B 2 Status of a PROFINET IO device

The status of a PROFINET IO device can be queried using predefined process data.

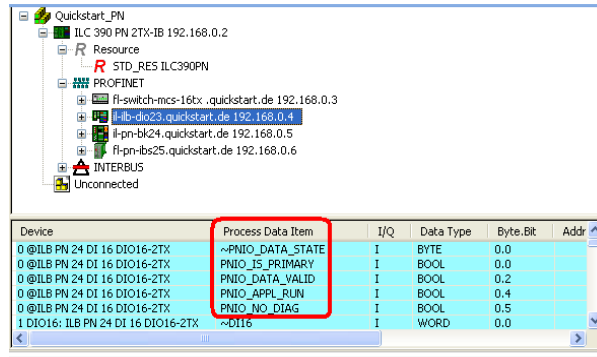


Figure B-2 PROFINET IO device: Predefined process data

The "~PNIO\_DATA\_STATE" byte that contains corresponding status bits is available. A PROFINET IO device only provides valid data if this bit is set.

### PNIO\_DATA\_VALID

For PROFINET IO, it may be normal during operation that a PROFINET IO device within the network cannot be accessed (e.g., undocking devices).

Use this variable to specify for your application whether or not the operation of all other PROFINET IO devices is to be continued despite a device failure.

In either case, the application program must receive information as to whether or not a PROFINET IO device is supplying valid data. This is why each PROFINET IO device has a PNIO\_DATA\_VALID process data item.

This bit must be set for the PROFINET IO device to supply valid data and all other process values to be valid.

### PNIO\_IS\_PRIMARY

This process data item is reserved for future redundancy functions.

### PNIO\_APPL\_RUN

This bit indicates that the application is running on the PROFINET IO device.

### PNIO\_NO\_DIAG

If this bit is set, there are no device diagnostics present.



Not all bits of the "~PNIO\_DATA\_STATE" byte are used. To prevent unused bits being used by mistake, hide them in the control program by linking to a corresponding bit mask.



To query the status, assign the process data to variables.

- Switch to the process data assignment workspace.
- In the top left window, "Symbols/Variables", select the resource ("STD\_RES : ILC390PN" in the example).
- In the top right window, select the device for which you would like to link the process data to variables (e.g., ILB PN 24 DI16 DIO16-2TX).
- Select the variable in order to link it (e.g., PNIO\_NO\_DIAG).

- If variables have already been created, use drag & drop to link the selected variable to one of the variables on the left-hand side.  
If a corresponding variable has not yet been created, select "Create Variable" in the context menu.

The created variable is displayed in the bottom left window.

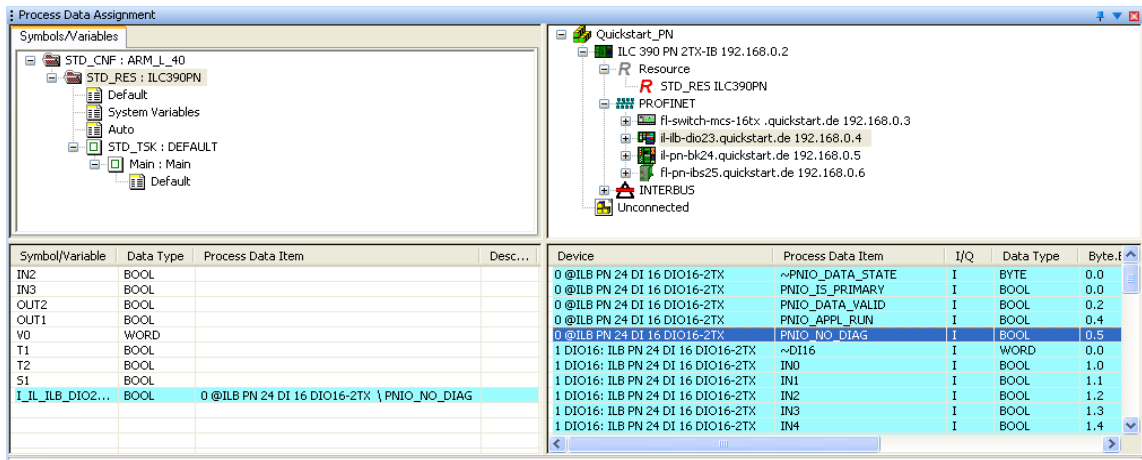


Figure B-3 PROFINET IO device: Variable created for querying the status

### B 3 INTERBUS registers

The registers of an INTERBUS master are available directly as process data. If necessary, assign the corresponding process data to the variables of the controller.

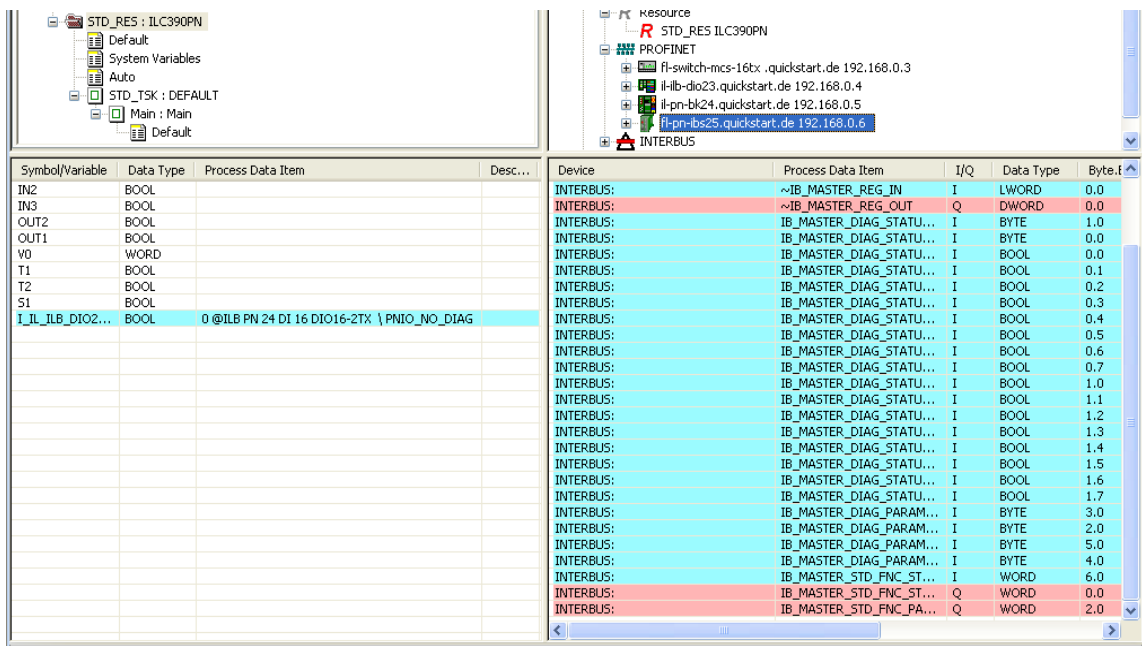


Figure B-4 INTERBUS registers

## B 4 Setting all PROFINET IO devices to a defined state

In many cases, a machine/system can no longer be operated in the event of a PROFINET IO device failure. This is normal in modern INTERBUS systems where neither bus couplers nor isolated disconnection are used.

In the example system, a PROFINET IO device failure does not affect the operation of all the other PROFINET IO devices. If you would like to set all the outputs to a defined state in the event of an error, the PNIO\_FORCE\_FAILSAFE system variable can be used for this. It can be found under "Global\_Variables" in the IEC programming workspace.

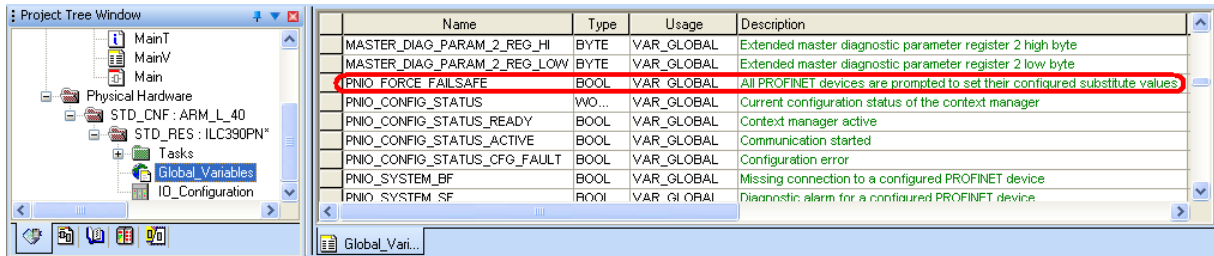


Figure B-5 PNIO\_FORCE\_FAILSAFE system variable

If the variable is set, all the PROFINET IO devices output zero values or defined substitute values (the substitute values are currently set to "0" by default). The corresponding inputs are set to zero.



## C AX OPC Server and WebVisit

### C 1 AX OPC Server

For information about using AX OPC Server, please refer to the corresponding documentation.

AX OPC Server includes:

- The "OPC Configurator" for establishing a connection between PC WorX and an OPC client (e.g., Visu+).
- An "OPC Test Client" for testing the connection.

AX OPC Server is also installed as a demo version. To register your license, proceed as follows:

- Start the "OPC Test Client".
- In the taskbar, activate the context menu for the "OPC Test Client" icon.

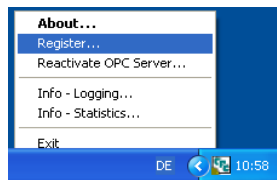


Figure C-1 Registering the "OPC Test Client"

- Select the "Register" menu item and enter the registration code.

#### C 1.1 Preparatory tasks in PC WorX

- In order to use a variable in AX OPC Server, activate the "OPC" checkbox:
  - When creating variables in the "Variable Properties" window (Figure C-2)
  - Or in the variables worksheet (Figure C-3 on page C-2)

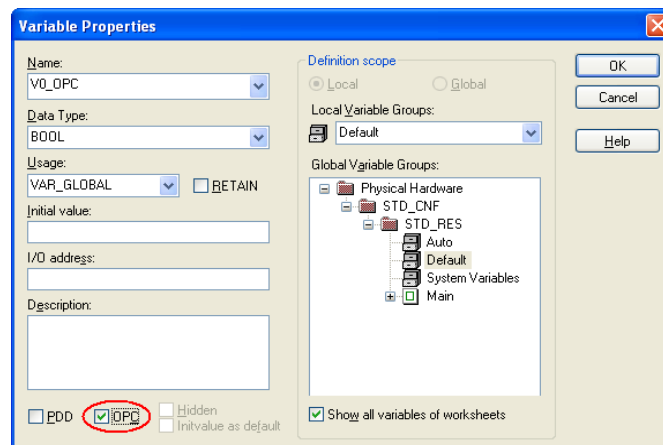


Figure C-2 Creating variables for AX OPC Server

Name	Retain	PDD	OPC
[-] Default			
IN2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
IN3	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
OUT2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
OUT1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
V0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
T1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
T2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure C-3 Activating "OPC" in the variables worksheet

- When sending the project, activate the "Include OPC data" checkbox.

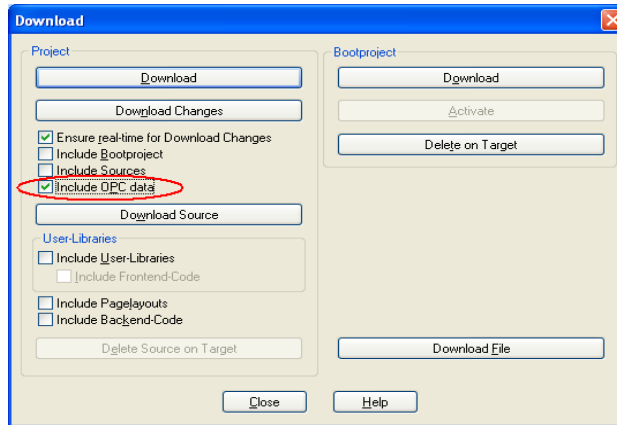


Figure C-4 Download: Include OPC data

## C 1.2 OPC Configurator

- Start the "OPC Configurator" program.

The configurator commands can be accessed via context menus.

- Select the resource type.

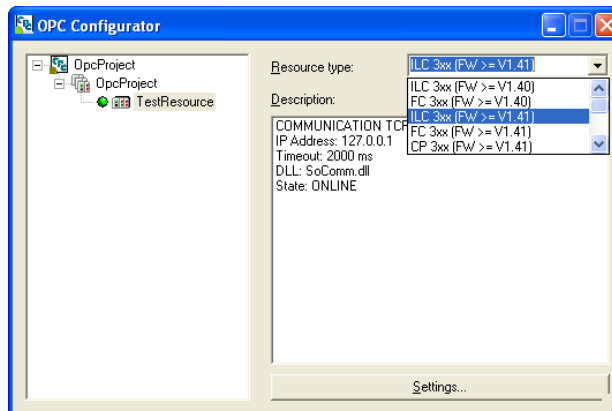


Figure C-5 Selecting the resource type

- Open the context menu for the "TestResource" entry.
- Select the "Settings" menu item.

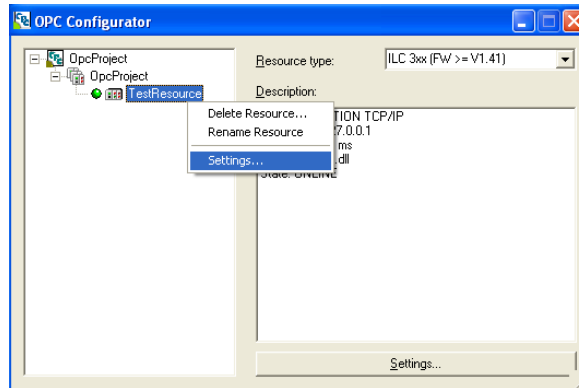


Figure C-6 Settings

- Under "IP Address", enter the IP address of the controller for which you would like to use the OPC data.

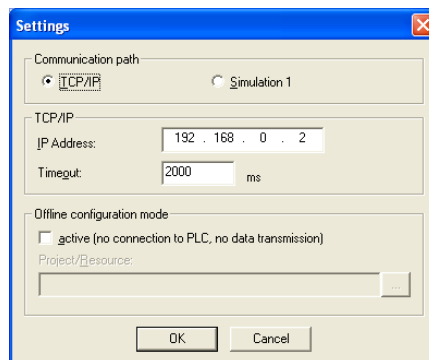


Figure C-7 Specifying the IP address of the controller

This completes the configuration for using OPC data through an OPC client.

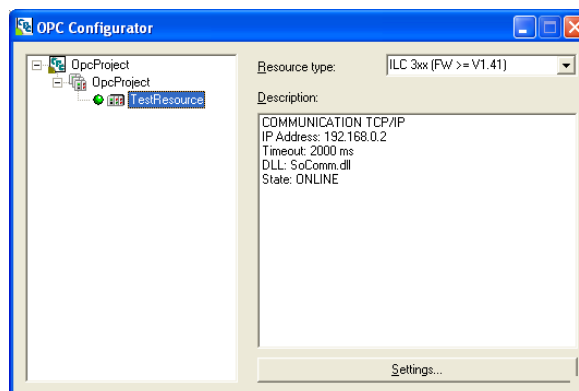


Figure C-8 OPC configuration completed

### C 1.3 OPC Test Client

The test client can be used to test the OPC configuration.

- Open the "OPC Test Client" program.
- Connect the test client to the OPC server by selecting "Connect..." in the "Server" menu.

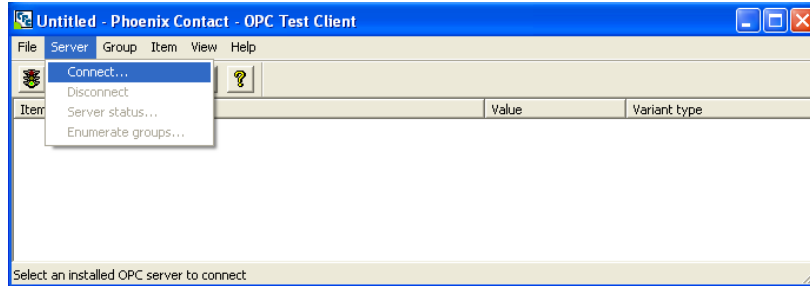


Figure C-9 Connecting the test client

- In the "Group" menu, select "Add all items" to display all variables defined as OPC variables.

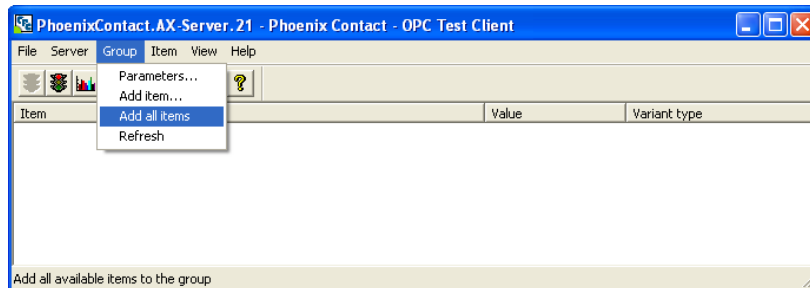


Figure C-10 Add all Items – Displaying OPC variables

The defined variables are displayed.

To set variables in the test client, e.g., to test the connection, proceed as follows:

- Select the variable to be set and then click on the "Write item" button.

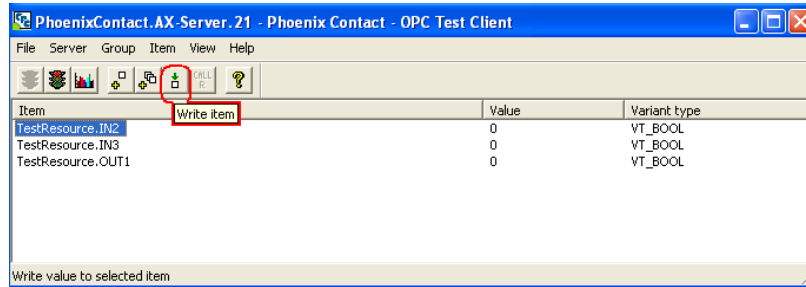


Figure C-11 OPC variables and "Write item" button

- In the "Write Item Value" window, enter the required value (in the example: IN2 = 1 (BOOL)).

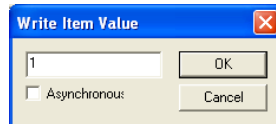


Figure C-12 OPC variable: Specifying the value

- Confirm your entry with "OK".

The result is displayed in the "Value" field.

## C 2 WebVisit

The WebVisit software is used to visualize the global variables of the Inline Controller used under PC WorX. WebVisit is a software tool used to generate web pages. The software runtime component is a web server, which is stored in PC WorX. The variable values are actually visualized via a Java-compatible standard browser.

To visualize variables from your PC WorX project in WebVisit, activate the "PDD" checkbox:

- In the "Variable Properties" window (Figure C-13)
- Or in the variables worksheet (Figure C-14)

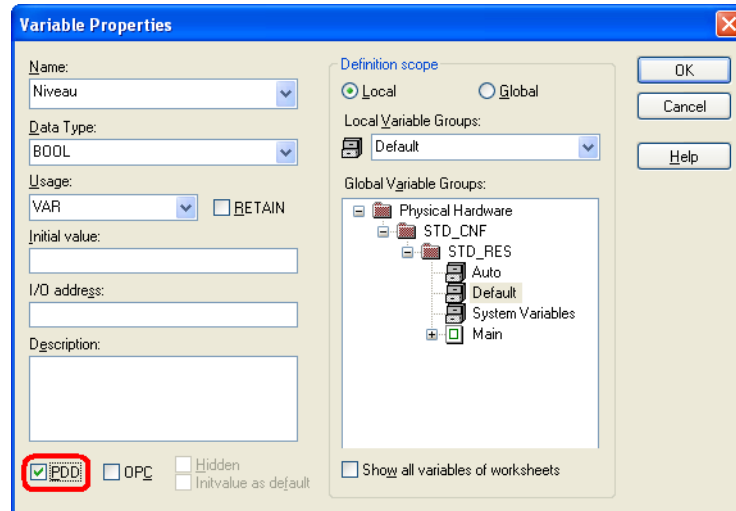


Figure C-13 Creating variables for WebVisit

Name	Type	Usage	Description	Address	Init	Retain	PDD	OPC
Default								
Zeitdauer	TIME	VAR			T#5000ms	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Runtime	TIME	VAR				<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Pulseingang	BOOL	VAR_EXTERNAL				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
START	BOOL	VAR_EXTERNAL				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure C-14 Activating "PDD" in the variables worksheet

When compiling the project, a file called "pdd.csv" is generated, which is used by WebVisit for the visualization.

In WebVisit, enter the PC WorX project used (recommended) or the corresponding pdd.csv file.



For information about using WebVisit, please refer to the corresponding documentation.

## D Phoenix Contact controllers

When creating a new project, several versions are provided for different controllers. The versions differ with regard to the hardware and firmware version used. The scope of functions in PC WorX may therefore differ for the various controller versions.

For additional information about the various controllers, please refer to the "AH EN BASIC DATA ILC... /RFC... /FC..." application note.

The aim of this document is to provide an overview of Phoenix Contact controllers for control technology. It contains the basic specifications for controllers from the Phoenix Contact controller range broken down into controller classes, as well as the ordering data for the individual controllers.





Компания «Океан Электроники» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Поставка оригинальных импортных электронных компонентов напрямую с производств Америки, Европы и Азии, а так же с крупнейших складов мира;
- Широкая линейка поставок активных и пассивных импортных электронных компонентов (более 30 млн. наименований);
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Помощь Конструкторского Отдела и консультации квалифицированных инженеров;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Поставка электронных компонентов под контролем ВП;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- При необходимости вся продукция военного и аэрокосмического назначения проходит испытания и сертификацию в лаборатории (по согласованию с заказчиком);
- Поставка специализированных компонентов военного и аэрокосмического уровня качества (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Actel, Aeroflex, Peregrine, VPT, Syfer, Eurofarad, Texas Instruments, MS Kennedy, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics и др.);

Компания «Океан Электроники» является официальным дистрибьютором и эксклюзивным представителем в России одного из крупнейших производителей разъемов военного и аэрокосмического назначения «JONHON», а так же официальным дистрибьютором и эксклюзивным представителем в России производителя высокотехнологичных и надежных решений для передачи СВЧ сигналов «FORSTAR».



## JONHON

«JONHON» (основан в 1970 г.)

Разъемы специального, военного и аэрокосмического назначения:

(Применяются в военной, авиационной, аэрокосмической, морской, железнодорожной, горно- и нефтедобывающей отраслях промышленности)

«FORSTAR» (основан в 1998 г.)

ВЧ соединители, коаксиальные кабели, кабельные сборки и микроволновые компоненты:

(Применяются в телекоммуникациях гражданского и специального назначения, в средствах связи, РЛС, а так же военной, авиационной и аэрокосмической отраслях промышленности).



Телефон: 8 (812) 309-75-97 (многоканальный)

Факс: 8 (812) 320-03-32

Электронная почта: [ocean@oceanchips.ru](mailto:ocean@oceanchips.ru)

Web: <http://oceanchips.ru/>

Адрес: 198099, г. Санкт-Петербург, ул. Калинина, д. 2, корп. 4, лит. А