AUTOMATION



User manual

UM EN PROFINET CTRL DEV

PROFINET IO controller/device functions



AUTOMATION

User manual PROFINET IO controller/device functions

2010-04-19

Designation: UM EN PROFINET CTRL DEV

Revision: 00

This user manual is valid for:

PROFINET IO devices from Phoenix Contact

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 electricians or persons instructed by them, who are familiar with applicable national standards and other regulations regarding electrical engineering and, in particular, the relevant safety concepts.

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 measures 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, will 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	autho	rity
-----------	----	-------	-------	------

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. Third-party products are always named without reference to patent rights. The existence of such rights shall not be excluded.

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

1

Make sure you always use the latest documentation. It can be downloaded at: 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 <u>www.phoenixcontact.com</u>.

Published by

 PHOENIX CONTACT GmbH & Co. KG

 Flachsmarktstraße 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

PHOENIX CONTACT

Table of contents

1	PROFINET IO controller	r/device fu	nctions	1-1
	1.1	1 User gr	oup of the manual	1-1
	1.2	2 Basic k	nowledge required	1-1
	1.3	3 Additio	nal PROFINET documentation	1-1
	1.4	4 Svstem	requirements	1-2
	1.5	•	NET IO controller/device functions	
2	Network topologies			2-1
	2.1	1 Topolo	gy 1: Mechatronic unit with lower-level compact controllers	2-1
	2.2		gy 2: Four identical machine controllers under a e park controller	2-2
	2.3		gy 3: System control with lower-level subsystems	
3	Description of a typical a (all devices in one netwo			3-1
	3.1		tion on how it was carried out	
	3.2	2 Typical	application	3-4
	3.3	3 Offline	configuration	
		3.3.1	Lower-level project	
		3.3.2	Higher-level project	3-12
	3.4	4 Online	configuration	3-19
		3.4.1	Preparing the PC for communication	
		3.4.2	Configuring the ILC 170 ETH 2TX	3-20
		3.4.3	Configuring the ILC 330 PN	3-25
		3.4.4	Observe startup behavior	3-26
		3.4.5	Checking the program start of the higher-level project	
		3.4.6	Checking the program start of the lower-level project	3-28
4	Description of a typical a			4 1
		-		
	4.1		configuration	
		4.1.1	Lower-level project	
		4.1.2	RFC 470 PN-3TX higher-level/lower-level project	
		4.1.3	Higher-level project	
	4.2		configuration	
		4.2.1	Preparing the PC for communication	
		4.2.2	Configuring the ILC 170 ETH 2TX	
		4.2.3 4.2.4	Configuring the RFC 470 PN-3TX Configuring the ILC 330 PN	
		4.2.4 4.2.5	Observe startup behavior	
		4.2.3	Observe startup benavior	

4.2.6	Checking the program start of the higher-level project
4.2.7	Checking the program start of the lower-level project4-33

1 PROFINET IO controller/device functions

The "PROFINET IO controller/device functions (UM EN PROFINET CTRL DEV) user manual provides an overview of the PROFINET communication system with device functions. This system description provides support when installing, starting up or operating a PROFINET device system. Examples show you how to program IO device diagnostics.

1.1 User group of the manual

Use this user manual if your are responsible for programming user programs or configuring, starting up and servicing automation systems.

1.2 Basic knowledge required

The following knowledge is required to understand the user manual:

- General knowledge with regard to automation technology
- Knowledge on how to use computers or equipment similar to a PC (e.g., programming devices) under the Windows operating system
- Knowledge of how to use PC WorX
- Good knowledge of the PROFINET IO communication method.

1.3 Additional PROFINET documentation

The PROFINET documentation is modular, providing you with optimum information.

Available PROFINET documents

"PROFINET basics" user manual UM EN PROFINET SYS

The manual describes PROFINET system basics. This includes:

- PROFINET development
- PROFINET versions
- PROFINET properties
- PROFINET installation and startup
- PROFINET and wireless

Quick start guides

- "Installing and starting up the starterkit 3.0" quick start guide UM QS EN PROFINET STARTERKIT 3.0.
- "Configuring INTERBUS devices in a PROFINET IO network using the example of STEP 7"
 UM QS EN PROFINET PROXY IB

Device-specific data sheets

The data sheets describe the specific properties of PROFINET IO devices. These include:

- Function description
- Ordering data and technical data
- Local diagnostic and status indicators
- Pin assignment and connection example
- Programming data/configuration data

PROFINET documents in preparation

- "Acyclic communication" application note AH EN PROFINET AZY KOM
- "PROFINET diagnostics" application note AH EN PROFINET DIAG



Make sure you always use the latest documentation. It can be downloaded at:

www.phoenixcontact.net/catalog

1.4 System requirements



Please note that the PROFINET IO device function of the ILC170 ETH 2TX is only available in the PC WorX software from version 6.00 Service Pack 2 or later (part of the AUTOMATIONWORX Software Suite 2009 1.50 Service Pack 2). The PC WorX Express software does not support these functions.

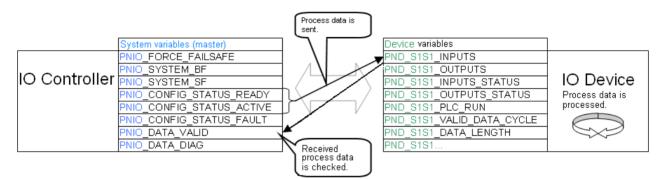
i

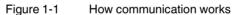
Firmware 3.5x for all controllers, including the PROFINET IO device function, is at least required to use the PROFINET IO device functions.

1.5 PROFINET IO controller/device functions

The master-slave procedure known from PROFIBUS was transferred to a providerconsumer model with PROFINET. A Provider generates and transmits data which the Consumer receives and processes. In terms of communication all devices in the PROFINET network have equal rights. The configuration specifies how the field devices are assigned to a central control system. PROFINET IO divides the control devices into IO controllers and IO devices. IO controllers are typically control systems (e.g., a central vehicle control).

The interface for IO devices was standardized by the PNO (Profibus User Organization) (PROFINET IO and GSD). This allows control systems from various manufacturers to communicate with IO devices. During configuration IO devices are assigned logically to an IO controller.





System variable	Туре	Meaning
PNIO_FORCE_FAILSAFE	BOOL	All outputs are set to the safe state "0".
PNIO_CONFIG_STATUS	BOOL	Status of the active configuration in the context manager
PNIO_CONFIG_STATUS_ACTIVE	BOOL	Communication is active.
PNIO_CONFIG_STATUS_READY	BOOL	The context manager is active.
PNIO_SYSTEM_BF	BOOL	An error occurred in the PROFINET network, that means, there is no connection to at least one configured device. This value is not set if the "Drive BF" parameter was set to FALSE for a device. This device is removed from connection monitoring.
PNIO_SYSTEM_SF	BOOL	At least one device reported a system error (diagnostic alarm or maintenance alarm).
PNIO_DIAG_AVAILABLE	BOOL	At least one device reported a diagnostic alarm with an active connection.
PNIO_MAINTENANCE_REQUIRED	BOOL	At least one device reported the "maintenance demand" alarm with an active connection.

UM EN PROFINET CTRL DEV

PNIO_MAINTENANCE_DEMANDED	BOOL	At least one device reported the "maintenance request" alarm with an active connection.
PNIO_DATA_DIAG		If this bit is set, no device diagnostics is present.
PNIO_DATA_VALID	BOOL	The application program must receive information on whether a PROFINET IO device is supplying valid data or not. For this reason, the "PNIO_DATA_VALID" process date exists on each PROFINET IO device. Only if this bit is set does the PROFINET device supply valid data and all other process values are valid.

PROFINET system variables (PROFINET IO device)

System variable	Туре	Meaning
PND_S1S1_PLC_RUN	BOOL	Status of the higher-level control system/IO controller
PND_S1S1_VALID_DATA_CYCLE	BOOL	The higher-level control system/IO controller has established the connection.
PND_S1S1_OUTPUT_STATUS_GOOD	BOOL	I/O provider status of the higher-level control system/ IO controller
PND_S1S1_INPUT_STATUS_GOOD	BOOL	I/O consumer status of the higher-level control system/ IO controller
PND_S1S1_DATA_LENGTH	WORD	Process data length that was configured for the IO device.
PND_S1S1_OUTPUTS	PND_IO_512 [256] [128] [64] [32]	OUT process data Memory area for OUT process data that the IO device receives from the higher-level control system/IO controller.
PND_S1S1_INPUTS	PND_IO_512 [256] [128] [64] [32]	IN process data Memory area for IN process data that the IO device receives from the higher-level control system/IO controller.

2 Network topologies

The following pages show three typical examples of network topologies. These topology examples are to explain the dependence and/or independence of the PROFINET IO controller/device functions.

The following hardware was used for the network structure:

ILC 330 PN	2988191-03
ILC 170 ETH 2TX	2916532-04
RFC 470 PN-3TX	2916600-07
FL SWITCH SMCS 4TX-PN	2989093-06

2.1 Topology 1: Mechatronic unit with lower-level compact controllers

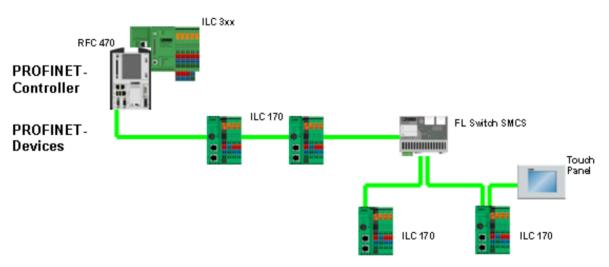
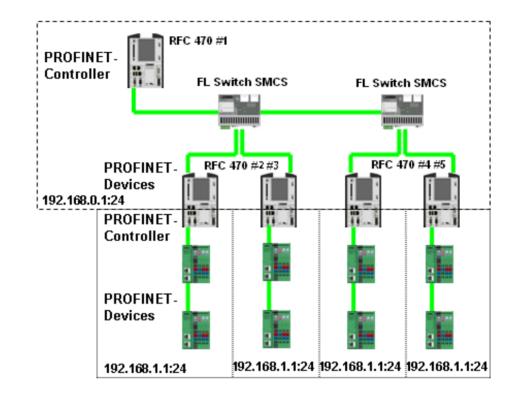


Figure 2-1 Topology 1: All devices in one network

Topology 1 describes a central concept with lower-level compact controllers. Every compact controller (ILC 1xx) is an independent PROFINET IO device and handles a local mechatronic unit with inputs and outputs. All controllers are available in a local network. Realtime communication over the central controller (RFC 470 PN-3TX, ILC 3xx) takes place over PROFINET. Controllers that are connected over a switch can be disconnected from the network at any time.



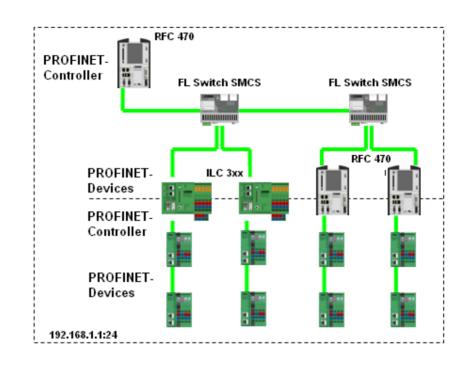
2.2 Topology 2: Four identical machine controllers under a machine park controller

Figure 2-2 Topology 2: Devices in several networks

Topology 2 describes a central concept with lower-level machine controllers.

Every machine controller (RFC 470 #2, #3, #4, #5) comprises a PROFINET IO device. This machine controller comprises in parallel IO controllers with their own IO devices. The lower-level networks can use the identical IP address range since they are separated by the controller.

The RFC #1 controller as well as the RFCs #2, #3 and RFCs #4, #5 on the device side are located in a higher-level network. Being PROFINET IO devices, the individual I/Os are located in a lower-level network.



2.3 Topology 3: System control with lower-level subsystems

Figure 2-3 Topology 3: Devices in several networks

Topology 3 describes a central concept with lower-level system controllers. Every controller comprises a PROFINET IO device. This controller is also an IO controller with its own IO devices. All controllers and IO devices are located in one network.

3 Description of a typical application (all devices in one network)

3.1 Information on how it was carried out

Alignment

The alignment of the data elements in the Inline controller memory can result in "data gaps" when storing data in the memory. The compiler automatically fills these gaps with padding bytes during the compiler process in order to prevent incorrect processing.

The disadvantage of the "automatic" filling of data gaps becomes apparent when data is transmitted from the Inline controller to another controller. If this controller does not know the memory algorithm of the Inline controller it will interpret the received data incorrectly.

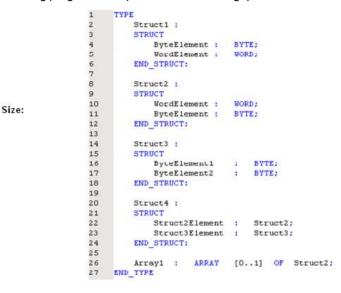
It is therefore useful to program the filling of data gaps in your application program. Data transmissions to other controllers can thus be taken into consideration. For example, use byte arrays with an even number of bytes and/or word arrays in order to avoid data gaps in your application program.

The following should be taken into consideration when creating the program:

- Create data types in flat structures, i.e., do not nest user-defined data types.
- Insert padding bytes manually in order to ensure the uniform size and layout of the data types.
- When inserting padding bytes, please observe the memory alignment method of the controllers used in the application (1-byte, 2-byte or 4-byte alignment).

Programming example with data gaps

The following program example shows how data gaps are filled.



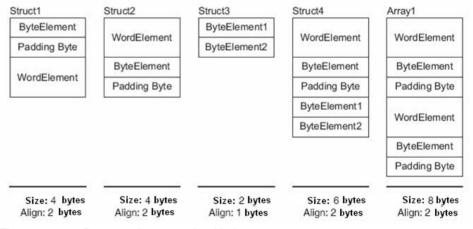


Figure 3-1 Programming example with data gaps

Struct1 receives a padding byte after the ByteElement so that the WordElement is at a WORD address (address that can be divided by 2 leaving no remainder). The alignment of the overall structure is based on the data type used with maximum alignment. In this case the WordElement specifies the alignment.

The size of Struct2 is calculated based on the elements used and the resulting alignment. The corresponding number of padding bytes is inserted so that the size of the data type with the value of the alignment can be divided by 2 leaving no remainder (data type size modulo alignment = 0).

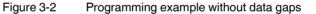
Struct3 does not receive any padding bytes as the maximum alignment corresponds to one byte.

Due to the padding bytes that belong to the Struct2 structure, the Struct3 structure starts at an even address. The number of padding bytes in array 1 corresponds to that of two consecutive Struct2 structures.

Programming example without data gaps

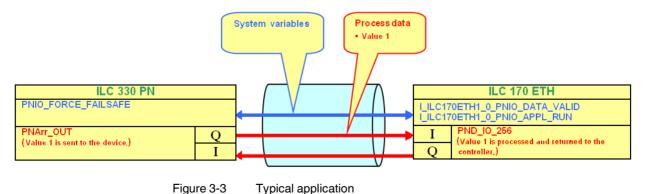
The following program shows an example of how data gaps may be filled in your application program. Fill data gaps, which are to be expected due to the memory alignment, with application data.

```
1
     TYPE
2
         Struct1 :
3
         STRUCT
4
             DyteElement : DYTE;
             ByteElement : BYTE; (*Padding-Byte*)
WordElement : WORD;
5
6
7
         END STRUCT:
8
9
         Struct2 :
10
         STRUCT
             WordElement : WORD;
11
12
             ByteElement : BYTE;
             ByteElement : BYTE; (*Padding-Byte*)
13
14
         END STRUCT:
15
16
         Struct3 :
17
         STRUCT
             ByteElement1 : BYTE;
ByteElement2 : BYTE;
18
                                BYTE:
19
20
         END STRUCT:
21
22
         STRUCT4 :
23
         STRUCT
24
            Struct2Element : Struct2;
25
            Struct3Element : Struct3;
26
         END_STRUCT:
27
         Array1 : ARRAY [0..1] OF Struct2;
28
29
     END TYPE
```



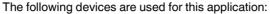
3.2 Typical application

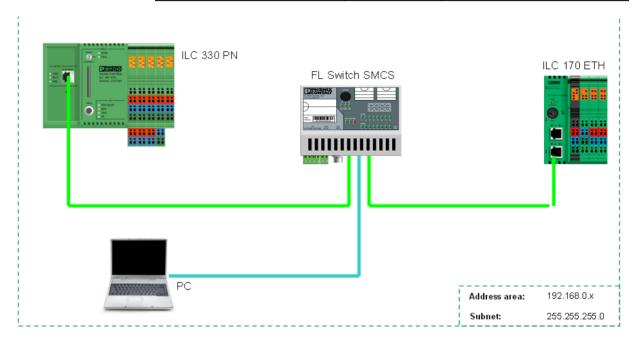
In the following application all devices are in one network, see also the topology example on page 2-1.



UM EN PROFINET CTRL DEV

Device	Order No.	IP address
ILC 330 PN as master	2988191	192.168.0.5
ILC 170 ETH 2TX as device	2916532	192.168.0.7
FL SWITCH SMCS 4TX (optional)	2989093	
Notebook as programming device		192.168.0.10







In this example, a project is created on the lower-level controller (ILC 170 PN) by requesting the status variables of PROFINET communication (PND_S1S1). For this purpose, a function block is created in structured text that sets the value "true" on the ONBOARD_OUTPUT_BIT0 system variable. The LED is ON when the ILC 330 PN sends the value "1".

In the example, a function block is used for logical ANDing. The I_ILC170ETH1_0_PNIO_DATA_VALID and I_ILC170ETH_0_PNIO_APPL_RUN variables (both system variables) map the status of the inputs to which the PNIO_FORCE_FAILSAFE system variable is connected.

The PNIO_DATA_VALID system variable indicates for each PROFINET IO device whether the connection to this PROFINET IO device was established successfully. Only if this bit is set does the PROFINET IO device supply valid data and all other process values are active.

A negated result is linked to the PNIO_FORCE_FAILSAFE variable. The PROFINET system is stable when the system variable PNIO_FORCE_FAILSAFE = 0. All outputs are set according to the process data If PNIO_FORCE_FAILSAFE = 1 (at least one PNIO_DATA_VALID variable set to 0), the safe state "0" is output for all PROFINET IO device outputs.

In addition, the value 1 is assigned to the PNArr_OUT[0] variable (user variable). This is done via the negated status of the PNIO_FORCE_FAILSAFE system variable. The value 1 is converted in the BYTE data type, since the PROFINET IO process data (PND_IO_256) are assigned as ARRAY OF BYTE data type for the variable.

3.3 Offline configuration

3.3.1 Lower-level 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 control system are now prepared.

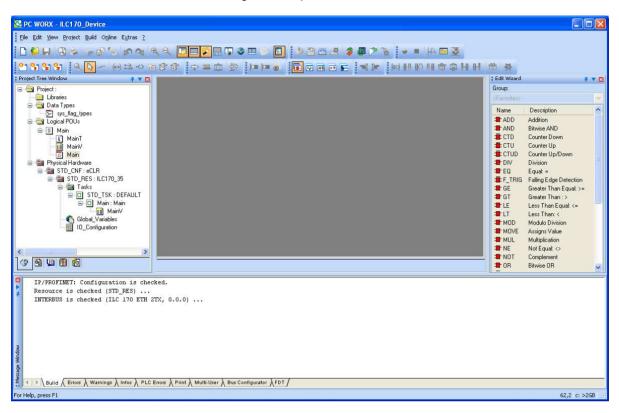
 Select the "ILC 170 ETH Rev. >01/3.50" control system and confirm your selection with "OK".

New Project	X
General CP 3xx CPX FC xxx ILC 1xx ILC 2xx ILC 3xx	OK Cancel
ILC 155 ETH Rev. > ILC 170 ETH Rev. > 01/3.50 01/3.00	<u>•</u>
ILC 170 ETH Rev. > 01/3.50	

Figure 3-5 Selecting the controller

- Select the "File, Save Project As/Zip Project As..." command.
- Enter a project name (here: ILC170_Device) and save the project.

Save/Zip pr	oject as
Save:	🗁 Projects 💽 🕑 🍺 📂 🖽 -
È Example ∑Example.r	nwt
File name:	ILC170_Device.mwt
File type:	Project Files (*.mwt)
- Zip Options ☑ Zip Us ☑ Zip P <u>w</u> ☑ Zip Pa	er-Libraries Zip Erontend-Code (-Libraries
Figure 3-6	Save project



The following window opens:

Figure 3-7 PC WorX start screen

- Right-click on Logical POUs.
- Insert the function block.

Project Tree Window			7 💌 🔛
Project : Libraries Data Types Sys_flag_types Cogical POUs	3		
🖻 🔲 Main	Insert	🎦 Program	
— 🚺 Main — 🔝 Main 🍲	Paste STRG+V	🚼 Eunction	
	Expand All	🚡 Function <u>B</u> lock	
🖮 🍘 Physical Hard		P <u>O</u> U group	
	Save As <u>N</u> etwork Template		
	Define Place <u>h</u> olders		
ė 🛛 🚞	P <u>r</u> operties		
	MainV		
	I_Variables onfiguration		
<u>10</u> 10	er migen soldt t		
<			>
Project 🖸 POUs 🖉	Libraries 🔠 Hardware 🛄 In	stances	

Figure 3-8 Inserting the function block

- Select the ST (Structured Text) language.
- Name the block "Data_Acknowledge".

Insert		
Name: Data_Acknowlegde	∼ Language	OK Cancel
 Program F<u>unction</u> Function <u>B</u>lock Action Transition Step Worksheet 	 □ IL ○ STI ○ SFC ○ EBD ○ LD ○ FFLD ○ MSFC ○ WAR ○ Data Types ○ Description 	Help
Datatype of return value:	~	
PLC type: <independent></independent>	Pr <u>o</u> cessor	

Figure 3-9 Selecting the programming language and naming the function block

• Open the worksheet by double-clicking on "Data_Acknowledge".

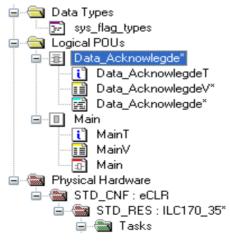


Figure 3-10 Opening the worksheet

Insert the following program to your worksheet.

•

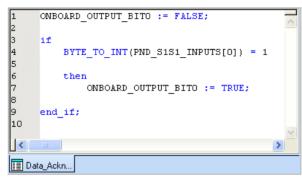


Figure 3-11 Inserting the program

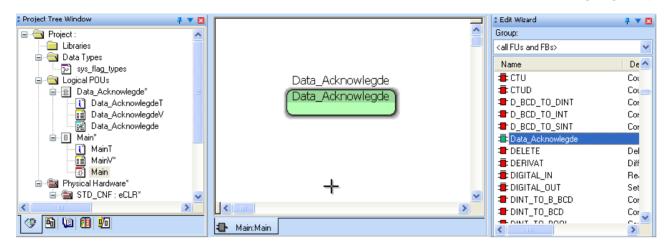
The ONBOARD-OUTPUT_BIT0 system variable and the PROFINET IO device status variable PND_S1S1_INPUTS for the process data can be found under the **Global_Variables**.

• Select the maximum process data length of 256 bytes (PND_IO_256) for the data exchange between master and device.

🚦 Project Tree Window 🛛 📮 🔻 🔀		Nama	Tun	Manualina	Developitume	^
🖃 🔄 Project :		Name	Тур	Verwendung	Beschreibung	
Libraries		ONBOARD_INPUT_BIT2	BOOL	VAR_GLOBAL	Local input IN3	%MC
Elblands		ONBOARD_INPUT_BIT3	BOOL	VAR_GLOBAL	Local input IN4	%MC
sys_flag_types		ONBOARD_INPUT_BIT4	BOOL	VAR_GLOBAL	Local input IN5	%M0
		ONBOARD_INPUT_BIT5	BOOL	VAR_GLOBAL	Local input IN6	%MC
🖨 🔄 Logical POUs		ONBOARD_INPUT_BIT6	BOOL	VAR_GLOBAL	Local input IN7	%M0
Data_Acknowlegde*		ONBOARD_INPUT_BIT7	BOOL	VAR_GLOBAL	Local input IN8	%M0
Data_AcknowlegdeT		ONBOARD_OUTPUT_BIT0	BOOL	VAR_GLOBAL	Local output OUT1) %M
Data_AcknowlegdeV*	11-	ONBOARD_OUTPUT_BIT1	BOOL	VAR_GLOBAL	Local output OUT2	%M
Data_Acknowlegde*		ONBOARD_OUTPUT_BIT2	BOOL	VAR_GLOBAL	Local output OUT3	%M
🗎 🔲 Main		RTC_DAY	INT	VAR_GLOBAL	Stud	%M
MainT			INT	VAD	Jorem time (month)	%M
[î] MainT ≣i MainV ⊡i Main		RTC_YEAK		GLOBAL	System time (year)	96MR
		PND_S1S1_PLC_RUN	BOOL	VAR_GLOBAL	Status of the higher-level control system	96DO
😑 📸 Physical Hardware		PND_S1S1_VALID_DATA	BOOL	VAR_GLOBAL	IO Controller has established the connection	%DXI
🖻 🖓 STD_CNF : eCLR		PND_S1S1_OUTPUT_STAT	BOOL	VAR_GLOBAL	IOP status of the higher-level control system	%(X)
😑 📸 STD_RES : ILC170_35*		PND_S1S1_INPUT_STATUS	BOOL	VAR GLOBAL	IOC status of the higher-level control system	%DXI
🖻 🖓 📷 Tasks		PND_S1S1_DATA_LENGTH	WORD	VAR GLOBAL	Process data length	%M
😑 🔲 STD_TSK : DEFAULT		PND S1S1 OUTPUTS	PND 10 256	VAR GLOBAL	Output process data	%QE
🖻 🔲 Main : Main	r	PND S1S1 INPUTS	PND 10 256	VAR GLOBAL	Input process data	%IBI =
🔜 🛄 MainV	n–	IB_DEVICE_PARAM_ACTIV	BOOL	VAR_GLOBAL	Interbus device configuration activated	%M0
Global_Variables		IB_DEVICE_PARAM_READY	BOOL	VAR GLOBAL	Interbus device configuration completed	%M0
IO_Configuration		IB_DEVICE_PARAM_ERROR	BOOL	VAR_GLOBAL	Interbus device configuration error	%MD 🗸
<	<]				>
· · · · · · · · · · · · · · · · · · ·		Data_Ackn 📑 Global_Vari.				

Figure 3-12 Selecting the process data

٠



Afterwards insert the created function block in the "Main" worksheet using drag & drop.

Figure 3-13 Inserting the function block into the worksheet

- Compile the project and save it.
- Close the project.

3.3.2 Higher-level 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 control system are now prepared.
- Select the "ILC 330 PN Rev. > 01/4.6F/3.50" control system and confirm your selection with "OK".

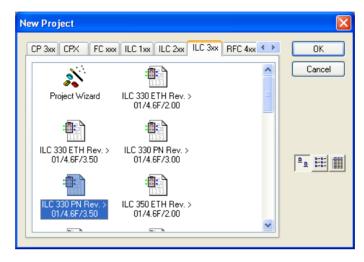


Figure 3-14 Selecting the controller

- Select the "File, Save Project As/Zip Project As..." command.
- Enter a project name (here: ILC330_Controller and save the project.

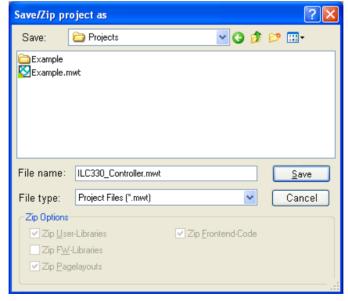


Figure 3-15 Save project

UM EN PROFINET CTRL DEV



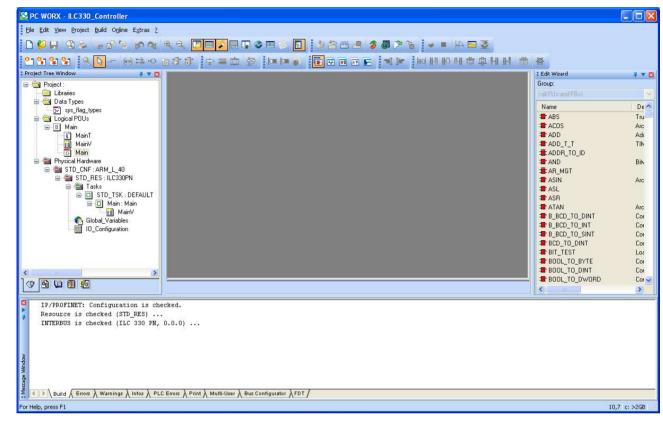


Figure 3-16 Start screen

- First integrate the ILC 170 ETH 2TX as a PROFINET IO device into the bus structure.
- Change to the bus structure. To do this, click on the "Bus Structure" icon in the toolbar.
- Insert the ILC 170 ETH 2TX as a device into the bus structure (right click).

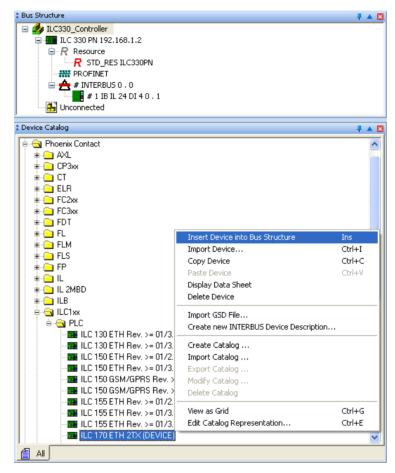


Figure 3-17 Inserting the ILC 170 ETH 2TX as a device in the bus structure

The PROFINET device inserted will be displayed in the Bus Structure workspace. The IP address is created depending on the IO controller address.

Bus Structure 📮 🔺 🔟	2 Device		÷ 🔻 🖞
🖃 🍰 ILC330_Controller	ILC 170) ETH 2TX (DEVICE) ilc-170-device32 192.168.0.3 \P	PROFINET Settings\
🗐 🚮 ILC 330 PN 192.168.0.2			
R Resource		Name	Value
R STD_RES ILC330PN		Vendor	Phoenix Contact
	B	VendorID	0x00B0
E - 📶 ILC 170 ETH 2TX (DEVICE) ilc-170-device32 192.168.0.3	B	Designation	ILC 170 ETH 2TX (DEVICE)
0 @ILC 170 ETH 2TX (DEVICE)	B	DeviceID	0x0035
1 EA256: ILC 170 ETH 2TX (DEVICE)	B	Functional description	
🖃 📥 # INTERBUS 0 . 0	B	Device type	PLC
# 1 IB IL 24 DI 4 0 . 1	D	Device family	ILC1xx
		Order number	2916532
	B	Revision	xx
		DNS/PROFINET Device Name	ilc-170-device32
	B	Station Name	
		Device Name	
	D	Module Equipment ID	
Device Catalog 🕴 🔺 🗖	B	IP Address	192.168.0.3
⊯- 🔁 Festo	B	Subnetmask	255.255.255.0
B Phoenix Contact	B	Default Gateway	
🖮 🧰 Universal	B	Realtime class	RT
	B	Reduction ratio input	16 ms
	B	Reduction ratio output	16 ms
	B	Faulty telegrams until connection is aborted	12
	B	Monitoring Time Inputs (ms)	192
	B	Monitoring Time Outputs (ms)	192
	B	Operation in case of configuration differences	no
	B	Log connection state	yes
	B	Drive BF	yes
		Node ID	215
			•
AII	P 🔍 P	ROFINET Settings I III PROFINET Stationnames	🖀 Bus interfaces 📄 Data sheet
	1 × ·		11 1

Figure 3-18 The ILC 170 ETH 2TX integrated as PROFINET device in the bus structure

The process data of the PROFINET device will be displayed in the Device Details workspace of the "Process Data" tab.

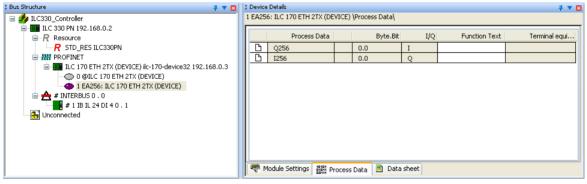


Figure 3-19 Process data of the PROFINET device

Description of a typical application (all devices in one network)

Switch to the IEC programming

and open the "Main" worksheet.

- Add the mapped function blocks.
- Negate the output at the AND block.

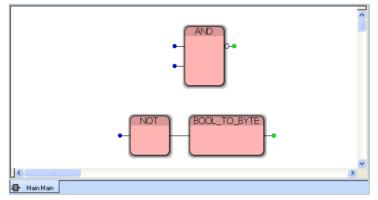


Figure 3-20 Adding function blocks

For the system variables for displaying the status of a PROFINET IO device, the process data is generated automatically.

- Switch to the process data assignment workspace.
- In the top left window, "Symbols/Variables", select the program (here: Main : Main).
- Highlight the PROFINET IO device in the top right window.
- Highlight the PNIO_APPL_RUN variable in the bottom right window.
- Enable the context menu on the variable and select the "Create Variable" command. In this case, a variable is generated automatically.
- Proceed in the same way for the PNIO_DATA_VALID variable.

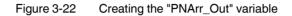
Process Data Assignment						₽ ▼
Symbols/Variables	PN les FAULT		■ 1.C330_controller ■ 1.C330_Controller ■ 1.C330_PN 192.168.0.2 ■ 1.C330_PN 192.168.0.2 ■ 2.C330_C0 192.170.0 ■ 2.C330_C0 192.0 ■	(DEVICE) ilc-170-device: TH 2TX (DEVICE) 170 ETH 2TX (DEVICE)	12 192.168.0.3	
Symbol/Variable	Data Type	Process Data Item	Device	Process Data Item	I/Q	Data Type
PNIO APPL RUN	BOOL		0 @ILC 170 ETH 2TX (DEVICE)	~PNIO DATA STATE	I	BYTE
PNIO_DATA_VALID	BOOL	0 @ILC 170 ETH 2TX (DEVICE) \ PNIO_DATA_VALID	0 @ILC 170 ETH 2TX (DEVICE)	PNIO_IS_PRIMARY	I	BOOL
			0 @ILC 170 ETH 2TX (DEVICE)	PNIO_DATA_VALID	I	BOOL
			0 @ILC 170 ETH 2TX (DEVICE)	PNIO_APPL_RUN	I	BOOL
			0 @ILC 170 ETH 2TX (DEVICE) 1 EA256: ILC 170 ETH 2TX (DEVICE)	PNIO_NO_DIAG Q256	Select all	Ctrl+A
			1 EA256: ILC 170 ETH 2TX (DEVICE) 1 EA256: ILC 170 ETH 2TX (DEVICE)		Search	Ctrl+F
					Filter	• • L
					 Color-coded view 	
					Connect	Ctrl+B
					Disconnect	Ctrl+Y
					Create Variable	Ctrl+E
					Search Variable	Ctrl+Q
					Search Cross Refere	nce Ctrl+R
<			<			

Figure 3-21 Creating variables

UM EN PROFINET CTRL DEV

• In addition, create the "PNArr_Out" variable with the "PND_IO_256" data type as "VAR-EXTERNAL".

	Name	Тур 🗸	Verwendung	Beschreibung
	🖃 Default			
	PNArr_Out	PND_IO_256	VAR_EXTERNAL	
	I_ILC170ETH1_0_PNIO_APPL_RUN	BOOL	VAR_EXTERNAL_PG	
	I_ILC170ETH1_0_PNIO_DATA_VALID	BOOL	VAR_EXTERNAL_PG	
	PNIO_FORCE_FAILSAFE	BOOL	VAR_EXTERNAL	All PROFINET devices are prompted to set th
۲				
₽	Main:Main 🗾 MainV:Main			



- To link the process data to the variables, proceed as follows:
- In the top left window (Symbols/Variables) select "Default".
- Highlight the PROFINET IO controller in the top right window.
- Highlight the "ILC 170 ETH ..." variable in the bottom right window.
- Connect the "PNArr_Out" variable to the "I256" process data item of the ILC 170 ETH 2TX device.

The total available data width of 256 bytes was selected in this example. You can change it later in the online configuration.

Process Data Assignment Symbols/Variables STD_CNF: ARM_L STD_RES: LLC StopPart STD_RES: LLC STD_TSK: Default STD_TSK: Default STD_TSK: Default Au	330PN riables DEFAULT Main			1 192.168.0.2 rce D_RES ILC330PN NET 1 770 ETH 2TX (DEVICE) IIc- 0 @0LC 170 ETH 2TX (DEV 1 EA256: ILC 170 ETH 2TX IRBUS 0 . 0 II BIL 24 DI 4 0 . 1	ICE)		<u>,</u> , , ,
Symbol/Variable PNArr_Out	Data Type PND_IO_256	Process Data Item	Device 0 @ILC 170 ETH 2TX (0 @ILC 170 ETH 2TX (1 EA256 ILC 170 ETH	DEVICE) PNIO_I5_F DEVICE) PNIO_DAT DEVICE) PNIO_DAT DEVICE) PNIO_APP DEVICE) PNIO_NO_	ATA_STATE I PRIMARY I A_VALID I L_RUN I		Data Type BYTE BOOL BOOL BOOL BOOL BYte_256
			i EA255: ILC 170 ETF # 1 IB IL 24 D1 40.1 # 1 IB IL 24 D1 40.1	2TX (DEVICE) I256	Select all Search Filter Color-coded view Connect Disconnect Create Variable Search Variable	Ctrl+A Ctrl+F Ctrl+B Ctrl+B Ctrl+Y Ctrl+P Ctrl+C Ctrl+Q	Byte_256 BOOL BOOL BOOL

Figure 3-23 Connecting the "PNArr_Out" variable to the process data

- Switch to IEC programming and link the variables as shown in the figure below.
- Add a negation to the output of the AND block.

The "PNIO_FORCE_FAILSAFE" system variable is used at the output of the AND block and the input of the "NOT" block.

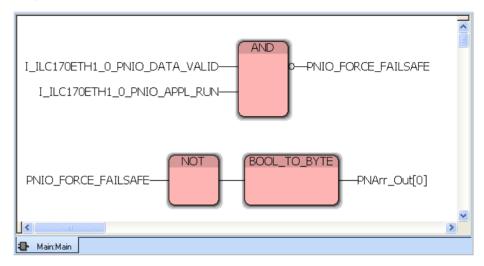


Figure 3-24 Inserting and linking variables

- Select the "0" array in the byte array by writing the field "[0]" after the "PNArr_Out" variable.
- Then compile the project and save it.

3.4 Online configuration

3.4.1 Preparing the PC for communication

• For configuration and parameterization assign an appropriate IP address for your PC within the 192.168.0.x address area.

In this example the PC receives the address 192.168.0.10.

Internet Protocol (TCP/IP) Prope	erties 🛛 🛛 🔀				
General					
You can get IP settings assigned autor this capability. Otherwise, you need to the appropriate IP settings.					
Obtain an IP address automatical	ly 🔤				
Ose the following IP address: —					
IP address:	192.168.0.10				
S <u>u</u> bnet mask:	255 . 255 . 255 . 0				
Default gateway:	· · ·				
○ O <u>b</u> tain DNS server address autor	natically				
── Use the following DNS server add	dresses:				
Preferred DNS server:					
<u>A</u> lternate DNS server:	· · ·				
Ad <u>v</u> anced					
	OK Cancel				

Figure 3-25 Assigning an IP address

• Select the network card of your PC that is to be used for communication in the "Tools/PROFINET..." menu of PC WorX.

PROFINET	
Communication	
Ethernet Network Board	Generic Marvell Yukon 88E8053 based Ethernet Controller - SecuRem 😪
DCP Timeout	Please choose Marvell Yukon 88E8055 PCI-E Gigabit Ethernet Controller - SecuRemote M Generic Marvell Yukon 88E8053 based Ethernet Controller - SecuRemote M Check Point Virtual Network Adapter For SecureClient - SecuRemote Minip
	OK Cancel Accept

Figure 3-26 Selecting the network card

Now the PC is ready for communications within the subnet.

3.4.2 Configuring the ILC 170 ETH 2TX

Assigning IP settings

To set the IP address in PC WorX proceed as described below:

- Open your project "ILC170_Device".
- Establish an Ethernet connection between your PC and the controller.
- In the PC WorX menu bar, select the "Extras... BootP/SNMP/TFTP-Configuration..." menu.

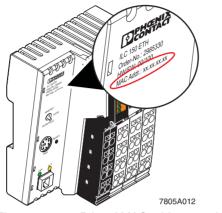
🖾 PC WORX -	
File Edit Yiew Project Build Online	Extras 2
🗋 🖗 🔒 🕄 🍇 🕌 🖬 🦌	Pagelayout Editor
2 2 2 2 2 Q D (#	Create OPC Visualization File
Bus Structure	Reset DA-Addresses
	PROFINET Configuration
	BootP/SNMP/TFTP-Configuration
	Check Source Storages
	Shortcuts
	Options

- Figure 3-27 Selecting "Extras/BootP/SNMP/TFTP-Configuration..."
- Activate the "BootP server active" checkbox.

BootP Settings	<
BootP	
OK Cancel Accept Help]

Figure 3-28 Activate BootP server

- Switch to the bus configuration workspace, see Figure 3-30
- Select the controller node.
- Select the "IP Settings" tab in the "Device Details" window.
- Enter the MAC address of the controller. It is printed on the device and starts with 00.A0.45.





Printed MAC address on the ILC 150 ETH controller

🗄 Bus Structure 🕴 📮 🔺 🔟	Device Details	🕂 🔻 🔀
🖃 🍰 ILC170_Device	ILC 170 ETH 2TX 192.168.0.7 \IP Settings\	
E ILC 170 ETH 2TX 192.168.0.7		
R Resource	Name Value	
R STD_RES ILC170_35	Phoenix Contact	
🔄 🚖 # INTERBUS 0 . 0	Designation ILC 170 ETH 2TX	
	Functional description Inline Controller for Ethernet Networks	s With 8
	Device type PLC	
	Device family ILC1xx	
	Crder number 2916532	
	□ Revision 01/3.50	
🗄 Device Catalog 🛛 🕴 🔺 🗖	🗅 Station Name	
E Festo	Device Name	
Phoenix Contact	Module Equipment ID	
Universal	DNS/PROFINET Device Name ILC170ETH1	
	MAC Address 00-A0-45-18-8A-A8	
	P IP Address 192.168.0.7	
	D Subnetmask 255.255.255.0	
	Default Gateway	
AI	💘 IP Settings 📕 Extended Settings 🔏 Communication 🔠 CPU Service Editor 🌲 Bus	s interfaci < 🗲

Figure 3-30 Entering the IP address

- Perform a cold restart for the controller.
- To do this, switch the supply voltage off and then on again after around 2 seconds.

The controller is assigned the IP address, which is specified in the project for the controller (here: 192.168.0.7). The following message appears in the message window in the "Bus Configurator" tab.



Figure 3-31 Message window

The IP address will now be permanently stored on the controller Flash memory.

Switching on the PROFINET IO device function



The following applies to the devices: ILC 170/330/350/370/390 PN / RFC 470 PN-3TX

By default upon delivery the PROFINET IO device function is switched off for every controller.

- Switch to the "Extended Settings" tab.
- Select the "IO device status" item in the device details under "Network Settings".
- Under "Settings", select "activated" in the pull-down menu.

🛿 Device Details 🕴 🔻 💆						
ILC 170 ETH 2TX 192.168.0.7 \Extended Settings\						
Network Settings Ethernet SNMP agent IO device status PROFINET Device	Settings deactivated deactivated Diffine					
IP Settings Extended Settings	🔏 Communication 📲 CPU Service Editor 🛛 🦨 Bus interfaces 📄 Data sheet 👘					

Figure 3-32 Device function activated

- Click on "Transmit".
- In the "Settings Communication Path" dialog confirm with "OK" the suggested IP address or the one you have selected for your application.

Settings Communication Path				
Ethernet (192.168.0.7)	~			
<u>OK</u> <u>C</u> ancel				

Figure 3-33 Setting the communication path

Device Details		🕂 🔻 🔀
LC 170 ETH 2TX 192.168.0.7 \Extended Set	ttings\ Settings activated Read Send Service executed successfully!	
TP Settings Extended Settings	🖌 Communication 🔠 CPU Service Editor 🗣 Bus interfaces 🖹 🛙	Data sheet

Successful execution of the service will be displayed in the status window.

Figure 3-34 Status window

To transfer the network settings you have to reset the IO controller.

• Select the "Ethernet" item in the Device Details window under "Network Settings".

1

•

The device name in the higher-level project (ILC 170 ETH device) must match the device name of the lower-level project (ILC 170 ETH).

In the "Activate Network Settings" area click the "Reset Control System" button.

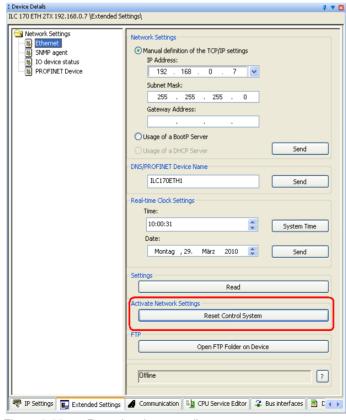


Figure 3-35 Resetting the controller

• In the "Settings Communication Path" dialog confirm with "OK" the suggested IP address or the one you have selected for your application.

Settings Communication Path				
Ethernet (192.168.0.7)	~			
<u> </u>				

Figure 3-36 Setting the communication path

Successful execution of the service will be displayed in the status window.

	Activate Network Settings
	Reset Control System
	FTP Open FTP Folder on Device
	Service executed successfully!
IP Settings Extended Settings	🔏 Communication 🖺 🛔 CPU Service Editor 🛛 🦨 Bus interfaces 🗋 🖻 C 🔹

Figure 3-37 Status window

The input/output data ranges available for the ILC 170 ETH 2TX as PROFINET IO device are displayed under "Network Settings" -> "PROFINET Device".

Device Details					÷ 🔻 🖬
ILC 170 ETH 2TX 192.168.0.7 \Extended Settings\					
Network Settings Ethernet SIMP agent D device status PROFINET Device	1	Name Input range Output range	Value 256 bytes 256 bytes	• •	
I Reference IP Settings Reference IP Setting		ommunication 📳	CPU Service Ed	ditor	r 🧣 Bus interfaces 📄 Data sheet

Figure 3-38 Input/output data ranges

To set the task to update the I/Os, select the device resource in the Bus Structure window.
Set the update task to "DEFAULT".

🚦 Bus Structure 🛛 📮 🔻 🔀	🚦 Device Details 📮 🔻			
🖃 🌆 ILC170_Device	STD_RES ILC170_35 \Resource\			
□	Name	Value		
R STD_RES ILC170_35	Configuration	STD_CNF		
🔄 🚖 # INTERBUS 0 . 0	Resource name	STD_RES		
	PLC type	eCLR		
	Processor type	ILC170_35		
	I/O Update by Task	<default></default>		
	Resource			

Figure 3-39 Setting the update task

3.4.3 Configuring the ILC 330 PN

To configure the ILC 330 PN controller, proceed as described in Section "Configuring the ILC 170 ETH 2TX" on page 3-19.

Assigning IP settings

Open the higher-level project "ILC330_Controller" and proceed as described in Section "Configuring the ILC 170 ETH 2TX" on page 3-19.

Please not the following modifications:

- Enter the MAC address of the ILC 330 PN controller.
- Assign the IP address 192.168.0.2.

To use the PROFINET device functions, the following conditions apply for the "ILC330_Controller" project.

ILC 330 PN settings as PROFINET controller:

IP address:	192.168.0.2
Subnet mask	255.255.255.0
PROFINET device name:	ILC330PN1
ILC 170 ETH 2TX settings	as PROFINET IO device:
•	
IP address:	192.168.0.7
IP address: Subnet mask	

1

Please make sure that the same PROFINET device name of the ILC 170 ETH (here: ILC170ETH1) is used in the lower-level project as in the higher-level project for the ILC 170 ETH as a device (here: ILC170ETH1).

3.4.4 Observe startup behavior

Starting up the controller is the easiest way to check whether

- The controller is correctly parameterized
- The IO devices have the right name
- There are double names or double IP addresses in the system.

Compile the ILC330_Controller project with the bus configuration. There will be a warning message if there is no application program. You can ignore this message.

Make sure that the controller has the IP address that was set in the project. Start the project control dialog via the menu bar.

If the message "Timeout" appears after 10 seconds, the project and device addresses do not match. It is also possible that the IP address of the computer has not been set correctly.

The controller can be reset from the project control dialog. The existing project will be deleted. Start the download and perform a cold reset. Afterwards the BF LEDs must go out on all devices.

To access the network status from the program, the following system variables are mapped in the global variables of the programming environment. Activate the "Debug On" operating mode and the values of these variables will be displayed.

Global variable	Description
PNIO_CONFIG_STATUS_ACTIVE	Connection to these devices is being established or has been completed.
PNIO_CONFIG_STATUS_READY	The connection establishment to the devices has been completed.

3.4.5 Checking the program start of the higher-level project

When the program is started correctly, the following screen will be shown in the Debug mode:

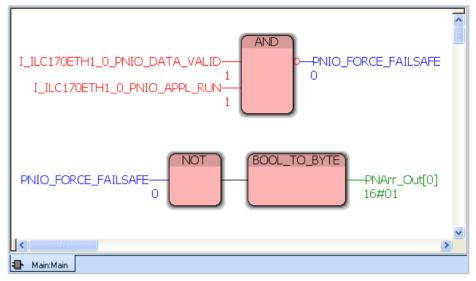
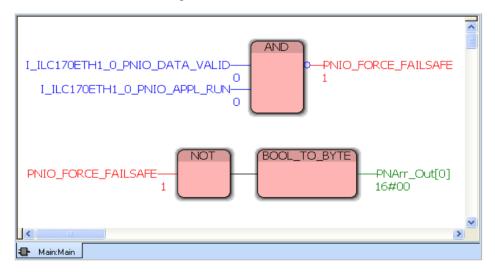


Figure 3-40 Program status

The PNIO_FORCE_FAILSAVE variable is in the FALSE state, thus communication is ensured and the outputs are set according to the process data.

If you remove the voltage connector of the ILC 170 2TX or change the device to the "Stop Mode", the status of PNIO_FORCE_FAILSAFE will change to TRUE. All outputs will be set to "0" and the value "1" is no longer transferred to the device.





3.4.6 Checking the program start of the lower-level project

The behavior described before can also be observed in the ILC170_Device project.

- Please open the lower-level project of the ILC 170 ETH 2TX.
- Then open the "Data_Acknowledge" POU and activate the Debug mode.

The following screen appears:

TRUE ONBOARD_OUTPUT_BITO := FALSE;							
2 3 16#01 if BYTE TO INT(PND S1S1 INPUTS[0]) = 1							
A 16#01 11 BYTE_TO_INT(PND_S1S1_INPOTS[0]) = 1							
5		then					
6	TRUE	ONBOARD_OUTPUT_BITO := TRUE;					
7							
8 9	en	d_if;					
Ē.	-						
	_					_	
Data_Ackn							
_		Value	Default value	Туре	Instance		
Data_Ackn	S1_INPUTS		Default value	PND_10_256	Instance STD_CNF.STD_F		
Data_Ackn	S1_INPUTS	Value	Default value		Instance		
Variable	S1_INPUTS		Default value	PND_10_256	Instance STD_CNF.STD_F		
Variable	S1_INPUTS	16#01	Default value	PND_IO_256 BYTE	Instance STD_CNF.STD_R STD_CNF.STD_F		
Variable	S1_INPUTS	16#01 16#00	Default value	PND_IO_256 BYTE BYTE	Instance STD_CNF.STD_R STD_CNF.STD_R STD_CNF.STD_R		
Data_Ackn	S1_INPUTS	16#01 16#00 16#00 16#00		PND_IO_256 BYTE BYTE BYTE	Instance STD_CNF.STD_F STD_CNF.STD_F STD_CNF.STD_F STD_CNF.STD_F		

Figure 3-42 Program status active

The value 1 is in array [0] of the PND_S1S1_INPUTS. The ONBOARD_OUTPUT_BIT0 variable is TRUE and the LED is ON.

1	FALSE	ONBOARD_OU	TPUT_BITO	:= FALSE;	~			
2								
4								
5		then						
6	FALSE	ON	IBOARD_OUTH	PUT_BITO :=	TRUE;			
8		end_if;						
9					~			
Ľ					>			
	Data_Ackn							
	Variable	Value	Default value	Туре	Instance			
	PND_S1S1_INPUT	IS		PND_10_256	STD_CNF.STD_R			
		16#00		BYTE	STD_CNF.STD_R			
3	[1]	16#00		BYTE	STD_CNF.STD_R			
P	[2]	16#00		BYTE	STD_CNF.STD_R			
МЧ	[3]	16#00		BYTE	STD_CNF.STD_R			
Vato								

Figure 3-43 Program is stopped

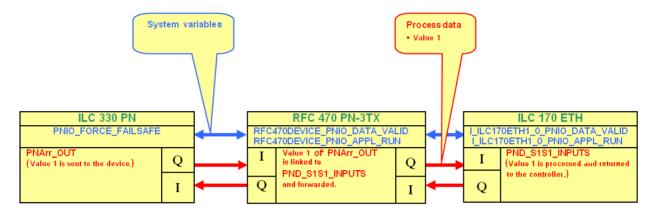
When communication is interrupted by removing the voltage connector of the ILC 170 device or by changing into the "Stop Mode" through the ILC 330 PN, the value is set to "0".

i

If you need more detailed information, call the Diag+ diagnostic tool from PC WorX under View-> Diag+. Here you connect explicitly to a controller and receive further information.

4 Description of a typical application (devices in several networks)

In the following application all devices are in several networks, see also the topology example on page 2-2.



The following devices are used for this application:

Device	Order No.	IP address
ILC 330 PN as master	2988191	192.168.1.3
RFC 470 PN-3TX as master	2916600	192.168.0.5
RFC 470 PN-3TX as device	2916600	192.168.1.5
ILC 170 ETH 2TX as device	2916532	192.168.0.7
FL SWITCH SMCS 4TX (optional)	2989093	-
Laptop (higher-level network 1)		192.168.1.10
Laptop (lower-level network 2)		192.168.0.10

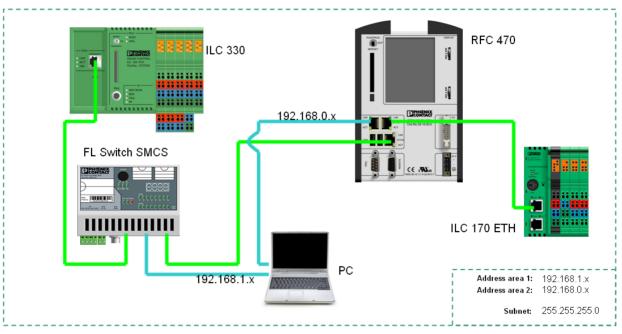


Figure 4-1 Typical application, devices in several networks

In this example, a project is created on the lower-level controller (ILC 170 ETH 2TX) by requesting the status variables of PROFINET communication (PND_S1S1). For this purpose, a function block is created in structured text that sets the value "true" on the ONBOARD_OUTPUT_BIT0 system variable. The LED is ON when the ILC 330 PN sends the value "1". This example uses two networks, the RFC 470 PN-3TX links the process data between ILC 330 PN and ILC 170 ETH 2TX. The program is identical with the first example application.

In the example, a function block is used for logical ANDing. The PNIO_DATA_VALID and PNIO_APPL_RUN variables (both system variables) of the RFC map the status of the inputs to which the PNIO_FORCE_FAILSAFE system variable is connected.

The PNIO_DATA_VALID system variable indicates for each PROFINET IO device whether the connection to this PROFINET IO device was established successfully. Only if this bit is set does the PROFINET IO device supply valid data and all other process values are active.

A negated result is linked to the PNIO_FORCE_FAILSAFE variable. The PROFINET system is stable when the system variable PNIO_FORCE_FAILSAFE = 0. All outputs are set according to the process data If PNIO_FORCE_FAILSAFE = 1 (at least one PNIO_DATA_VALID variable set to 0), the safe state "0" is output for all PROFINET IO device outputs.

Assign the value 1 to the PNArr_OUT[0] variable (user variable). This is done via the negated status of the PNIO_FORCE_FAILSAFE system variable. The value 1 is converted in the BYTE data type, since the PROFINET process data (PND_IO_256) are assigned as ARRAY OF BYTE data type for the variable.

4.1 Offline configuration

4.1.1 Lower-level 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 control system are now prepared.

 Select the "ILC 170 ETH Rev. >01/3.50" control system and confirm your selection with "OK".

New Project	
General CP 3xx CPX FC xxx ILC 1xx ILC 2xx ILC 3xx ()	OK Cancel
ILC 155 ETH Rev. > ILC 170 ETH Rev. > 01/3.50 01/3.00	<u>₽</u>
ILC 170 ETH Rev. > 01/3.50	

Figure 4-2 Selecting the controller

- Select the "File, Save Project As/Zip Project As..." command.
- Enter a project name (here: ILC170_Device) and save the project.

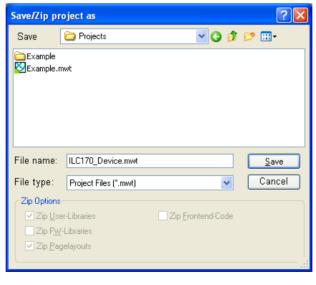


Figure 4-3 Save project

UM EN PROFINET CTRL DEV

<u> </u>	e			
Ihe	tollo	wina	window	onens.

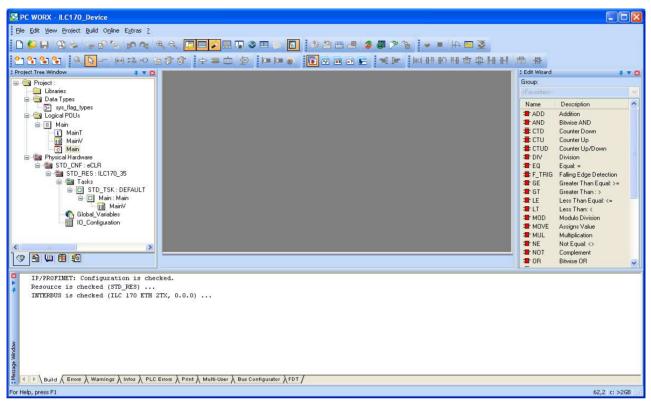


Figure 4-4 PC WorX start screen

- Right-click on Logical POUs.
- Insert the function block.

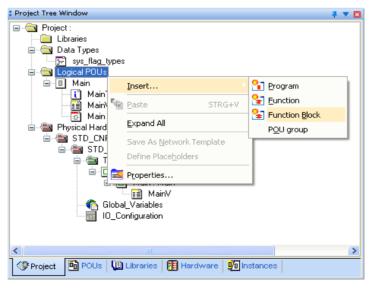


Figure 4-5 Inserting the function block

- Select the ST (Structured Text) language.
- Name the block "Data_Acknowledge".

Insert		
<u>N</u> ame: Data_Acknowlegde		OK Cancel
Type Program Function Function Block Action Transition	Language ○ IL ○ <u>ST</u> I ○ <u>S</u> FC ○ <u>F</u> BD ○ L <u>D</u> ○ FFL <u>D</u> ○ <u>M</u> SFC	Laricer
O Step O <u>W</u> orksheet	 ○ VAR ○ Data Types ○ Description 	Mode Insert Append
Datatyge of return value:	▼ Pr <u>o</u> cessor	type:
<independent></independent>	<independent< td=""><td>dent> 💌</td></independent<>	dent> 💌

Figure 4-6 Selecting the programming language and naming the function block

• Open the worksheet by double-clicking on "Data_Acknowledge".

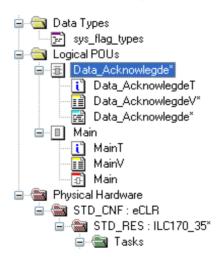


Figure 4-7 Opening the worksheet

• Insert the following program to your worksheet.

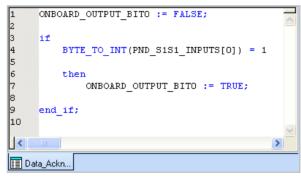


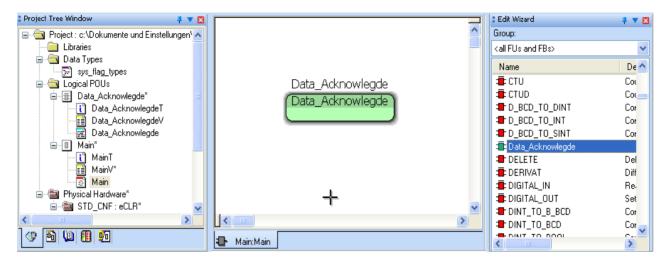
Figure 4-8 Inserting the program

The ONBOARD-OUTPUT_BIT0 system variable and the PROFINET IO device status variable PND_S1S1_INPUTS for the process data can be found under the **Global_Variables**.

• Select the process data length of 256 bytes (PND_IO_256) for the data exchange between master and device.

: Project Tree Window 📮 🔻 🛽	3	Name	Тур	Verwendung	Beschreibung	_
🖃 🔄 Project :		ONBOARD_INPUT_BIT2	BOOL	VAR GLOBAL	Local input IN3	%M0
- Dibraries		ONBOARD_INPUT_BIT3	BOOL	VAR_GLOBAL	Local input IN4	%M0
😑 🔄 Data Types			BOOL	-		%MC
sys_flag_types		ONBOARD_INPUT_BIT4		VAR_GLOBAL	Local input IN5	
		ONBOARD_INPUT_BITS	BOOL	VAR_GLOBAL	Local input IN6	%M0
Data_Acknowlegde*		ONBOARD_INPUT_BIT6	BOOL	VAR_GLOBAL	Local input IN7	%M0
i) Data_AcknowlegdeT		ONBOARD_INPUT_BIT7	BOOL	VAR_GLOBAL	Local input IN8	%MC
Data_AcknowlegdeV*		ONBOARD_OUTPUT_BIT0	BOOL	VAR_GLOBAL	Local output OUT1)%M
Data_AcknowlegdeV*		ONBOARD_OUTPUT_BIT1	BOOL	VAR_GLOBAL	Local output OUT2	%M
		ONBOARD_OUTPUT_BIT2	BOOL	VAR_GLOBAL	Local output OUT3	%M
		RTC_DAY	INT	VAR_GLOBAL	Sust	%M
î` MainT ∎ MainV ⊡ Main			INT	VAD	, stem time (month)	%M
MainV		RTC_YEAR		GLOBAL	System time (year)	%M
		PND_S1S1_PLC_RUN	BOOL	VAR_GLOBAL	Status of the higher-level control system	%IXI
🖻 📾 Physical Hardware		PND_S1S1_VALID_DATA	BOOL	VAR_GLOBAL	IO Controller has established the connection	%IXI
STD_CNF: eCLR		PND_S1S1_OUTPUT_STAT	BOOL	VAR_GLOBAL	IOP status of the higher-level control system	%IXI
😑 📾 STD_RES : ILC170_35*		PND S1S1 INPUT STATUS	BOOL	VAR_GLOBAL	IOC status of the higher-level control system	%IXI
🗐 🖼 Tasks		PND_S1S1_DATA_LENGTH	WORD	VAR GLOBAL	Process data length	%M
⊡ STD_TSK : DEFAULT		PND_S1S1_OUTPUTS	PND_IO_256	VAR GLOBAL	Output process data	%QE
😑 🔲 Main : Main		PND_S1S1_INPUTS	PND_IO_256	VAR_GLOBAL	Input process data	%IBI =
MainV		IB DEVICE PARAM ACTIV	BOOL	VAR GLOBAL	Interbus device configuration activated	%M0
Global_Variables		IB DEVICE PARAM READY	BOOL	VAR GLOBAL	Interbus device configuration completed	%M0
IO_Configuration		IB_DEVICE_PARAM_ERROR	BOOL	VAR GLOBAL	Interbus device configuration error	%MC 🗸
	<)		>
) (* B) (1) (1) (1)		Data_Ackn 🔢 Global_Vari				

Figure 4-9 Selecting the process data



Afterwards insert the created function block in the "Main" worksheet using drag & drop.



- Compile the project and save it.
- Close the project.

•

4.1.2 RFC 470 PN-3TX higher-level/lower-level 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 control system are now prepared.
- Select the "RFC 470 PN-3TX Rev. > 00/4.6F/3.50" control system and confirm your selection with "OK".

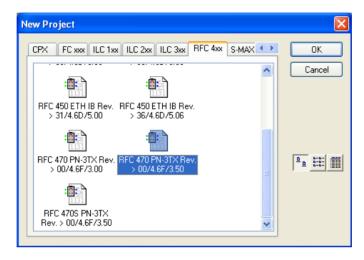


Figure 4-11 Selecting the controller

- Select the "File, Save Project As/Zip Project As..." command.
- Enter a project name (here: RFC470_Controller_Device and save the project.

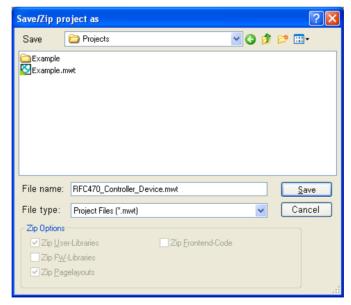


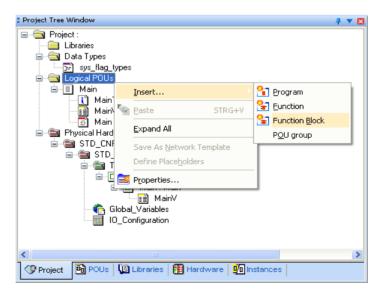
Figure 4-12 Save project

S PC WORX - RFC470 Controller Device	
Elle Edit View Project Build Opline Extras 2	
<u>▼</u> ■A = • <i>K</i> < ₩ ♦ 8 8 8 8 8 8 8 8 8 8	
21212121 《文十 金沙女的学校 中華語 物 百百多 副田田田市 美国 医子宫 化化合金化合金	12 H2H
: Project Tree Window 🕴 🔻 🖸	🗄 Edit Wizard 📮 🔻 🖾
Fright : c \Dokumente und Einstellungen\pyt Fight : c \Dokumente und Einstellungen\pyt	Group:
by syn ling lypes	Name
a 🔤 Logical POUs B-10 Main	ADS ACOS
— Ū MainT	📲 ADD
	ADD_T_T
⊡ main ⊟ ∰ Physical Hardware	ADDN_10_ID
e 🎒 STD_CNF:/PC_40	:#: AR_MGT
e 🚔 STD_RES : RFC470PN	TASIN
G ISTD_TSK: DEFAULT	ASL ASR
🖼 🔲 Main: Main	📲 ATAN
	B_BCD_TO_DINT B_BCD_TO_INT
🗕 📶 10_Configuration	B BCD_TO_SINT
	BCD_TO_DINT
	BIT_TEST
	BOOL_TO_BYTE BOOL_TO_DINT
	BOOL_TO_DWORD
	< >
-	
8	
- week	
Configurator (FDT /	
For Help, press F1	63,0 c: >2GB

The following window opens:

Figure 4-13 PC WorX start screen

- Right-click on Logical POUs.
- Insert the function block.





- Select the ST (Structured Text) language.
- Name the block "Data_Acknowledge".

Insert		
<u>N</u> ame: Data_Acknowlegde		OK Cancel
Type Program Function Function Block Action Transition Step Worksheet	Language	<u>Help</u> Use <u>R</u> eserve Mode ○ Insert ⊙ Append
Datatyge of return value: PLC type: (independent)	Pr <u>o</u> cessor	type:

Figure 4-15 Selecting the programming language and naming the function block

• Open the worksheet by double-clicking on "Data_Acknowledge".

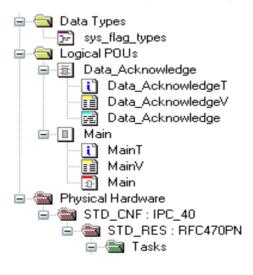


Figure 4-16 Opening the worksheet

Insert the following program to your worksheet.

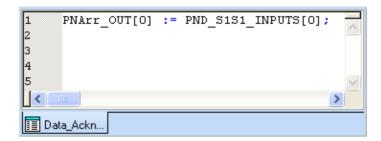


Figure 4-17 Inserting the program

•

The PNArr_OUT[0] variable is linked with the PROFINET device status variable PND_S1S1_INPUTS, so that the ILC 170 device can call the status of the ONBOARD_OTPUT_BIT0 system variable.

Select the maximum process data length of 256 bytes (PND_IO_256) for the data exchange between ILC 330 PN, RFC 470 PN-3TX and ILC 170 ETH 2TX.

The RFC 470 PN-3TX can transmit up to 512 bytes of data, however, the process data length is adapted to the ILC 170 ETH 2TX. It can transmit up to 256 bytes.

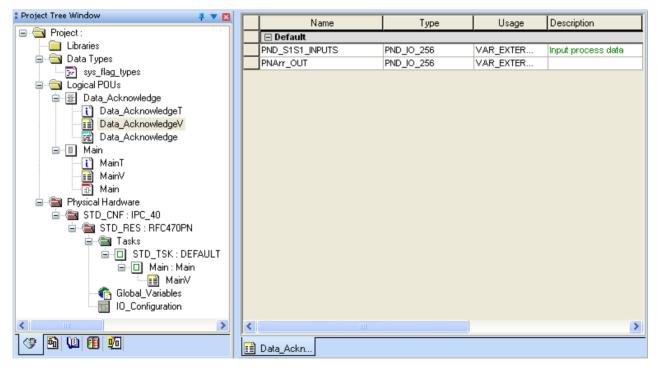
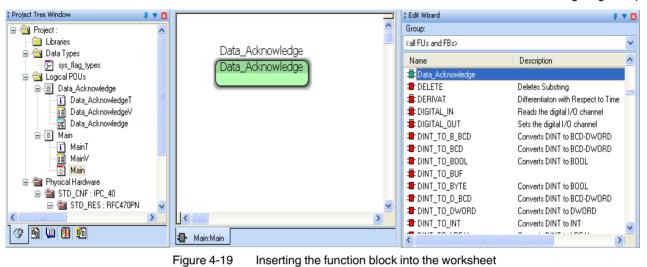


Figure 4-18 Creating the variables

UM EN PROFINET CTRL DEV

•



Afterwards insert the created function block in the "Main" worksheet using drag & drop.

Compile the project and save it.

Assigning process data

- Switch to the process data assignment workspace.
- In the top left window, "Symbols/Variables", select the "System Variables" program.
- Highlight the PROFINET IO device in the top right window.
- Highlight the I256 process data item in the bottom right window.
 - Highlight the PNArr_OUT variable in the bottom left window.
 - Enable the context menu on the variable and select the "Connect" command.

Process Data Assignment									🕂 🔻 🖾
Symbols/Variables STD_CNF: IPC_40 STD_RES: RFC470PN Default System Variables C STD_TSK: DEFAULT Main : Main				RFC470_Controller_Device RFC470_Controller_Device Resource RFC Resource RSTD_RES RFC470 RFC170 ETH 2TX 0 @ILC 170 ETH 2TX 0 @ILC 170 ETH 2TX 0 @ILC 170 ETH 2TX Unconnected	IPN (DEVICE) ilc1 TH 2TX (DEVI	CE)	7		
, Symbol/Variable	Data Type	Process Data Item	Ē	Device	Process Da	ata Item	I/Q		Data Type
PNArr OUT	PND IO 256			0 @ILC 170 ETH 2TX (DEVICE)	~PNIO DA	TA STATE	I		BYTE
-				0 @ILC 170 ETH 2TX (DEVICE)	PNIO_IS_P	RIMARY	I		BOOL
				0 @ILC 170 ETH 2TX (DEVICE)	PNIO_DATA	A_VALID	I		BOOL
				0 @ILC 170 ETH 2TX (DEVICE)	PNIO_APPL		I		BOOL
				0 @ILC 170 ETH 2TX (DEVICE)	PNIO_NO_I	DIAG	I		BOOL
				1 EA256: ILC 170 ETH 2TX (DEVICE)	Q256		I		Byte_256
				1 EA256: ILC 170 ETH 2TX (DEVICE)	1256		Q		Byte_256
						Select all		Ctrl+A	
						Search		Ctrl+F	
						Filter		•	
						✓ Color-coded v	iew		
						Connect		Ctrl+B	
						Disconnect		Ctrl+Y	
						Create Variab	le	Ctrl+E	
						Search Variab	le	Ctrl+O	
<		3	>	<		Search Cross	Reference		>
) <u>·</u>				1			, <u> </u>

Figure 4-20 Linking process data

• Compile, save, and close the project.

4.1.3 Higher-level 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 control system are now prepared.
- Select the "ILC 330 PN Rev. > 01/4.6F/3.50" control system and confirm your selection with "OK".

Save/Zip pr	oject as		? 🗙
Save:	🚞 Projects	O Ø	⊳ 🛄 🔁
ि Example M Example.r	nwt		
File name:	ILC330_Controller.mwt		<u>Save</u>
File type:	Project Files (*.mwt)	~	Cancel
Zip Options			
Zip Use	er-Libraries -Libraries	Zip <u>F</u> rontend-Code	
Zip Pag			

Figure 4-21 Selecting the controller

- Select the "File, Save Project As/Zip Project As..." command.
- Enter a project name (here: ILC330_Controller and save the project.

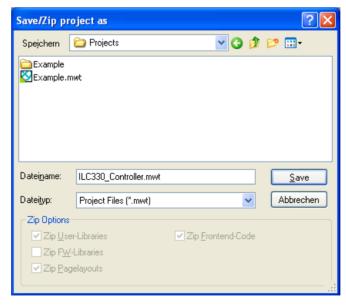
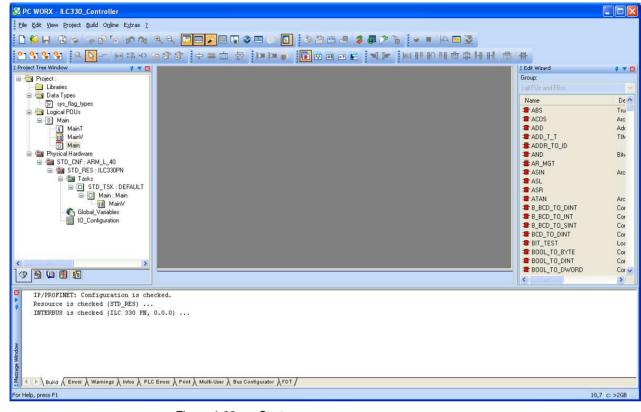


Figure 4-22 Save project

Description of a typical application (devices in several networks)



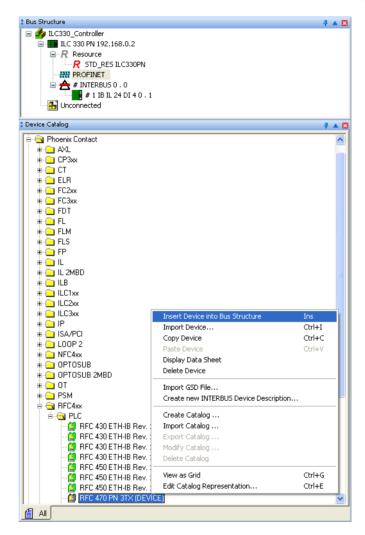
The following window opens:

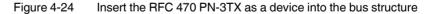
Figure 4-23 Start screen

Integrating the RFC 470 PN-3TX as PROFINET IO device

The following section describes how you integrate the RFC 470 PN-3TX as PROFINET device in the "ILC330_Controller" project.

- Change to the bus structure. To do this, click on the "Bus Structure" icon in the toolbar.
- Insert the RFC 470 PN-3TX as a device into the bus structure (right click).





Description of a typical application (devices in several networks)

The PROFINET device inserted will be displayed in the Bus Structure workspace. The IP address is created depending on the IO controller address.

E Bus Structure 🕴 🔺 🗖	🛛 🔤 🔤 Device Details					
🖃 🍰 ILC330_Controller	R	FC 47	0 PN 3TX (DEVICE) RFC470PN1 192.168.1.5 \PROF	INET Settings\		
🖨 🎫 ILC 330 PN 192.168.1.2	ШG					
R Resource			Name	Value		
R STD_RES ILC330PN		ß	Vendor	Phoenix Contact		
PROFINET		ß	VendorID	0×00B0		
🖮 🛄 RFC 470 PN 3TX (DEVICE) RFC470PN1 192.168.1.5		ß	Designation	RFC 470 PN 3TX (DEVICE)		
0 @RFC 470 PN 3TX (DEVICE)		ß	DeviceID	0x0032		
1 EA512: RFC 470 PN 3TX (DEVICE)		ß	Functional description			
🖻 🚖 # INTERBUS 0 . 0		ß	Device type	PLC		
# 1 IB IL 24 DI 4 0 . 1		ß	Device family	RFC4xx		
		ß	Order number	2916600		
		ß	Revision	XX		
		6	DNS/PROFINET Device Name	RFC470PN1		
Device Catalog	4 H	ß	Station Name			
		ß	Device Name			
r = - Eesto		ß	Module Equipment ID			
🖶 🚞 Phoenix Contact		ß	IP Address	192.168.1.5		
🖮 🦳 Universal		В	Subnetmask	255.255.255.0		
		В	Default Gateway			
		В	Realtime class	RT		
		В	Reduction ratio input	16 ms		
		В	Reduction ratio output	16 ms		
		ß	Faulty telegrams until connection is aborted	12		
		В	Monitoring Time Inputs (ms)	192		
		В	Monitoring Time Outputs (ms)	192		
		В	Operation in case of configuration differences	no		
		ß	Log connection state	yes		
		ß	Drive BF	yes		
		ß	Node ID	221		
	11			🖉 Dus interferenz 🔊 Data sheet		
		-≪ P	ROFINET Settings III PROFINET Stationnames	- Dus interfaces Data sheet		
Figure 4.25 The PEC	• 1	70	DN 3TV integrated as a DD	OFINET IO dovice in the bus		

Figure 4-25 The RFC 470 PN-3TX integrated as a PROFINET IO device in the bus structure

The process data of the PROFINET device will be displayed in the Device Details workspace of the "Process Data" tab.

🔋 Bus Structure 📮 🔺 🔟	📔 Device Details 📮 🔻 🚺						
🖃 🎝 ILC330_Controller	1 E/	A512: RFC 470 PN 3TX (DE	VICE) \F	Process Data\			
🖬 🔜 ILC 330 PN 192.168.1.2							
R Resource		Process Data		Byte.Bit	I/Q	Function Text	Terminal equi
R STD_RES ILC330PN	Ī	Q512		0.0	I		
PROFINET		D I512		0.0	Q		
😑 🛄 RFC 470 PN 3TX (DEVICE) RFC470PN1 192.168.1.5							
- 🔞 0 @RFC 470 PN 3TX (DEVICE)							
1 EA512: RFC 470 PN 3TX (DEVICE)							
🖻 🚖 # INTERBUS 0 . 0							
# 1 IB IL 24 DI 4 0 . 1							
: Module Catalog 🕴 🔺 🕅							
EA128: RFC 470 PN 3TX (DEVICE)							
(EA256: RFC 470 PN 3TX (DEVICE)							
EA32: RFC 470 PN 3TX (DEVICE)							
EA512: RFC 470 PN 3TX (DEVICE)							
EA64: RFC 470 PN 3TX (DEVICE)	1						>
				1.000			
		Module Settings	rocess [Data 🖹 Data	sheet		

Figure 4-26 Process data of the PROFINET IO device

- Replace the EA512 I/O module with the EA256 I/O module of the RFC. As a device the lower-level ILC 170 ETH 2TX can transmit up to 256 bytes.
- Delete the EA512 I/O module (right click).
- Drag the EA256 I/O module in the bus structure (left click).

The RFC 470 PN-3TX is now available as PROFINET IO device in the "ILC330_Controller" PC WorX project.

• Switch to the IEC programming



and open the "Main" worksheet.

- Add the mapped function blocks.
- Create the following variables at the links as specified.
- Negate the output at the AND block.

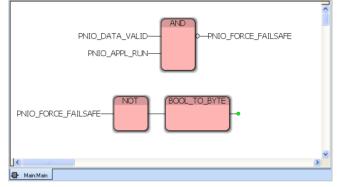


Figure 4-27 Adding function blocks

• Save the modified main program.

Assigning process data

For the system variables for displaying the status of a PROFINET IO device, the process data is generated automatically.

- Switch to the process data assignment workspace.
- In the top left window, "Symbols/Variables", select the program (here: Main : Main).
- Highlight the PROFINET IO device in the top right window.
- Highlight the PNIO_DATA_VALID variable in the bottom right window.
- Highlight the PNIO_DATA_VALID variable in the bottom left window.
- Enable the context menu on the variable and select the "Connect" command.
- Proceed in the same way for the PNIO_APPL_RUN variable.

Process Data Assignment							🕂 🔻 🖡		
Symbols/Variables STD_CNF: ARM_L_40 STD_RES: ILC330PN STD_RES: ILC330PN System Variables ID_STD_TSK: DEFAUL ID_STD_TSK: DEFAUL ID_Default ID_Default Auto	bols/Variables TD_CNF: ARM 1_40 TD_CNF: ARM 1_40 TD_Default TD_Default TD_System Variables TD_TSK: DEFAULT TD_Main: Main								
Symbol/Variable	Data Type	Process Data Item		Device	Process Data Item	I/Q	Data Type		
PNIO_APPL_RUN	BOOL			0 @RFC 470 PN 3TX (DEVICE)	~PNIO_DATA_STATE	I	BYTE		
PNIO_DATA_VALID	BOOL			0 @RFC 470 PN 3TX (DEVICE)	PNIO_IS_PRIMARY	I	BOOL		
				0 @RFC 470 PN 3TX (DEVICE)	PNIO_DATA_VALID		BOOI		
				0 @RFC 470 PN 3TX (DEVICE) 0 @RFC 470 PN 3TX (DEVICE)	PNIO_APPL_RUN PNIO NO DIAG	Select all	Ctrl+A		
				1 EA256: RFC 470 PN 31X (DEVICE)		Search	Ctrl+F		
				1 EA256: RFC 470 PN 3TX (DEVICE)	1256	Filter	•		
				· · · · · · · · · · · · · · · · · · ·		✓ Color-coded view			
						Connect	Ctrl+B		
						Disconnect	Ctrl+Y		
						Create Variable	Ctrl+E		
						Search Variable	Ctrl+O		
						Search Cross Reference			
<			>	<					

Figure 4-28 Linking variables

 In addition, create the "PNArr_Out" variable with the "PND_IO_256" data type as "VAR-EXTERNAL".

	Name	Туре	Usage	Description					
	🖃 Default								
	PNIO_FORCE_FAILSAFE	BOOL	VAR_EXTER	All PROFINET devices are prompted to set their					
	PNIO_APPL_RUN	BOOL	VAR_EXTER						
	PNIO_DATA_VALID	BOOL	VAR_EXTER						
	PNArr_Out	PND_IO_256	VAR_EXTER						
<									
-	🖶 Main:Main 🔢 MainV:Main								

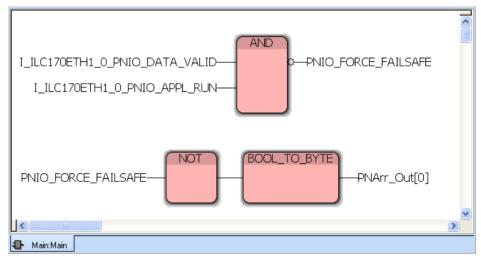
Figure 4-29 Creating the "PNArr_Out" variable

• Connect the "PNArr_Out" variable to the "I256" process data item of the RFC 470 PN-3TX device.

The total available data width of 256 bytes was selected in this example. You can change it later in the online configuration.

Process Data Assignment							∓ ▼ E
Symbols/Variables STD_CNF: ARM_L_40 STD_RES: ILC330PN System Variables STD_TSK: DEFAult System Variables STD_TSK: DEFAult Structure Auto							
Symbol/Variable PNArr_Out	Data Type PND_IO_256	Process Data Item	0	wice DRFC 470 PN 3TX (DEVICE		I/Q I	Data Type BYTE BOOL
			0	PRFC 470 PN 3TX (DEVICE PRFC 470 PN 3TX (DEVICE PRFC 470 PN 3TX (DEVICE PRFC 470 PN 3TX (DEVICE	E) PNIO_DATA_VALID E) PNIO_APPL_RUN	I I I I	BOOL BOOL BOOL BOOL
				A256: RFC 470 PN 3TX (E A256: RFC 470 PN 3TX (E	EVICE) 1256	I	Byte_256 Byte_256
					Select all Search	Ctrl+A Ctrl+F	
					Filter Color-coded view	•	
					Connect Disconnect	Ctrl+B Ctrl+Y	
					Create Variable Search Variable	Ctrl+E Ctrl+Q	
<			> <		Search Cross Referenc	e Ctrl+R	>

Figure 4-30 Connecting the "PNArr_Out" variable to the process data



Switch to IEC programming and link the variables as shown in the figure below.

Figure 4-31 Inserting and linking variables

- Select the "0" array in the byte array by writing the field "[0]" after the "PNArr_Out" variable.
- Then compile the project and save it.

4.2 Online configuration

4.2.1 Preparing the PC for communication

• For configuration and parameterization assign an appropriate IP address for your PC within the 192.168.0.x address area. In this example the PC receives the address 192.168.0.10.

Internet Protocol (TCP/IP) Prope	rties 🛛 🛛 🛛 🔀			
General				
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.				
◯ <u>O</u> btain an IP address automatically				
Use the following IP address: —				
<u>I</u> P address:	192.168.0.10			
S <u>u</u> bnet mask:	255 . 255 . 255 . 0			
<u>D</u> efault gateway:	· · ·			
○ 0 <u>b</u> tain DNS server address autor	natically			
O Use the following DNS server addresses:				
Preferred DNS server:				
<u>A</u> lternate DNS server:				
Ad <u>v</u> anced				
OK Cancel				

Figure 4-32 Assigning an IP address

 Select the network card of your PC that is to be used for communication in the "Tools/PROFINET..." menu of PC WorX.

PROFINET	
Communication	
Ethernet Network Board	Generic Marvell Yukon 88E8053 based Ethernet Controller - SecuRem 🗸
Please choose DCP Timeout Marvell Yukon 88E8055 PCI-E Gigabit Ethernet Controller - SecuRemote Generic Marvell Yukon 88E8053 based Ethernet Controller - SecuRemote Check Point Virtual Network Adapter For SecureClient - SecuRemote Min	
	OK Cancel Accept

Figure 4-33 Selecting the network card

Now the PC is ready for communications within the subnet.

i

Set the address 192.168.1.10 for the higher-level network with the ILC 330 PN as a master and the RFC 470 PN-3TX as a device. Set the address 192.168.0.10 for the lower-level network with the RFC 470 PN-3TX as a master and the ILC 170 ETH 2TX as a device.

4.2.2 Configuring the ILC 170 ETH 2TX

Assigning IP settings

To set the IP address in PC WorX proceed as described below:

- Open your project "ILC170_Device".
- Establish an Ethernet connection between your PC and the controller.
- In the PC WorX menu bar, select the "Extras... BootP/SNMP/TFTP-Configuration..." menu.

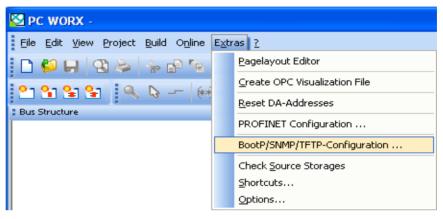


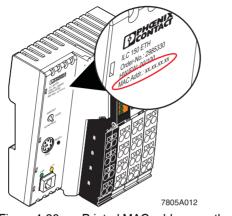
Figure 4-34 Selecting "Extras/BootP/SNMP/TFTP-Configuration..."

Activate the "BootP server active" checkbox.

BootP Settings	
BootP	
I	BootP Server active
	OK Cancel Accept Help

Figure 4-35 Activate BootP server

- Switch to the bus configuration workspace, see Figure 4-37
- Select the controller node.
- Select the "IP Settings" tab in the "Device Details" window.
- Enter the MAC address of the controller. It is printed on the device and starts with 00.A0.45.





Printed MAC address on the ILC 150 ETH controller

🗄 Bus Structure 🕴 🔺 🗖	Device Details	÷ 🔻 🛛
	ILC 170 ETH 2TX 192.168.0.7 \IP Settings\	
□	Name	Value
R STD_RES ILC170_35	🗅 Vendor	Phoenix Contact
🔄 📥 # INTERBUS 0 . 0	Designation	ILC 170 ETH 2TX
	Functional description	Inline Controller for Ethernet Networks With 8
	Device type	PLC
	Device family	ILC1xx
	D Order number	2916532
	🗅 Revision	01/3.50
🕽 Device Catalog 🛛 📮 🔺 🔀	🗅 Station Name	
E-G Festo	Device Name	
B- Phoenix Contact	Module Equipment ID	
	DNS/PROFINET Device Name	ILC170ETH1
	MAC Address	00-A0-45-18-8A-A8
	IP Address	192.168.0.7
	🗅 Subnetmask	255.255.255.0
	Default Gateway	
AI AI	🤫 IP Settings 🔄 Extended Settings 🛛 🖉 Comm	unication 🖺 CPU Service Editor 🚑 Bus interface < 🗲

Figure 4-37 Entering the IP address

- Perform a cold restart for the controller.
- To do this, switch the supply voltage off and then on again after around 2 seconds.

The controller is assigned the IP address, which is specified in the project for the controller (here: 192.168.0.7). The following message appears in the message window in the "Bus Configurator" tab.



The IP address will now be permanently stored on the controller Flash memory.

4.2.3 Configuring the RFC 470 PN-3TX

By default upon delivery the diagnostic display has the following status:

CONTROL	RDY	
Projectname: 		
INTERBUS MASTER	RDY	
PROFINET CONTROL	WAIT	
		
PROFINET DEVICE	WAIT	*
-		

Figure 4-39 Diagnostic display

The initial assignment of the IP settings can generally always be carried out using the diagnostic display.

When using the PC WorX software, the initial assignment of the IP settings can be carried out with BootP or using the COM1 serial interface.

- If the Remote Field Controller already has IP settings that are valid in your network, you can modify the IP settings via the network using PC WorX.
- Set the RFC 470 PN-3TX to the mapped IP address 192.168.0.5. It can be accessed in the network after a restart of the device.

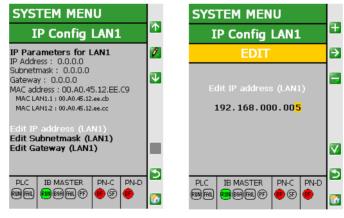


Figure 4-40 LAN1 configuration display

The procedure for assigning the IP settings is essentially the same for the LAN1 (LAN1.1/LAN1.2) and LAN 2 interfaces. The following describes the assignment of the IP settings at the LAN1 interface as an example. The LAN1.1/LAN1.2 interfaces are switched internally. Thus, both ports can be accessed using the IP settings defined.

For the LAN2 interface proceed as in the previous example, however, set the IP address to 192.168.1.5. The RFC 470 PN-3TX communicates with this address as a device.

NOTE:

The IP address of your PC must be in the same subnet as the LAN1 or LAN2 interface of the RFC 470 PN-3TX. Only then is communication for configuration of the ILC 170 ETH 2TX possible.

In this case the modification was done via the LAN1 interface (192.168.0.x subnet).

4.2.4 Configuring the ILC 330 PN

Assigning IP settings

- When assigning the IP settings for the ILC 330 PN, please proceed in the same way as for the ILC 170 ETH 2TX, see "Configuring the ILC 170 ETH 2TX" on page 4-22.
- Open the "ILC330_Controller" project.

Please not the following modifications:

Connect the network cable of your PC to the switch. Now you have established a connection from the PC to the ILC 330 PN.

- BootP server is active
- Enter the MAC address of the ILC 330 PN controller.
- Assign the IP address 192.168.1.2.

Switching on the IO PROFINET device function



The following applies to the devices: ILC 170/330/350/370/390 PN / RFC 470 PN-3TX

By default upon delivery the PROFINET device function is switched off for every controller. To switch it on, start your existing project (in the example here: "ILC170_Device") in PC WorX and activate the PROFINET device function as follows:

- Switch to the "Extended Settings" tab.
- Select the "IO device status" item in the device details under "Network Settings".
- Under "Settings", select "activated" in the pull-down menu.

Device Details ILC 170 ETH 2TX 192,168,0,7 \Extended Set	tings)
Network Settings Ethernet SIMP agent B IO device status PROFINET Device	Settings deactivated deactivated Diffine
Reference of the settings Provided Settings Reference of the settings	🔏 Communication 🔠 CPU Service Editor 🗣 Bus interfaces 🖹 Data sheet

Figure 4-41 Device function activated

- Click on "Transmit".
- In the "Settings Communication Path" dialog confirm with "OK" the suggested IP address or the one you have selected for your application.

Settings Communication Path	
Ethernet (192.168.0.7)	~
<u> </u>	

Figure 4-42 Setting the communication path

Successful execution of the service will be displayed in the status window.

Network Settings	Settings	
B Ethernet SNMP agent IO device status PROFINET Device	activated	~
PROFINET Device	Read Send	
	Service executed successfully!	
	Service executed successfully!	
	Gervice executed successfully!	
	Service executed successfully!	
	Service executed successfully!	
	Service executed successfully!	

Figure 4-43 Status window

To transfer the network settings you have to reset the IO controller.

Description of a typical application (devices in several networks)

• Sele • The de name c

Select the "Ethernet" item in the Device Details window under "Network Settings".

The device name in the higher-level project (ILC 170 ETH device) must match the device name of the lower-level project (ILC 170 ETH).

• In the "Activate Network Settings" area click the "Reset Control System" button.

Device Details .C 170 ETH 2TX 192.168.0.7 \Exten	ded Settings\
Network Settings SMMP agent SMMP usent ProFINET Device	Network Settings Manual definition of the TCP/IP settings IP Address:
	O Usage of a BootP Server O Usage of a DHCP Server Send
	Real-time Clock Settings Time: 10:00:31 System Time Date:
	Settings
	Activate Network Settings Reset Control System FTP
	Open FTP Folder on Device
IP Settings E, Extended Set	ttings 👩 Communication 📲 CPU Service Editor 🗣 Bus interfaces 📄 C 🕻

Figure 4-44 Resetting the controller

In the "Settings Communication Path" dialog confirm with "OK" the suggested IP address or the one you have selected for your application.

Settings Communication Path	×
Ethernet (192.168.0.7)	~
<u>D</u> K <u>C</u> ancel	

Figure 4-45 Setting the communication path

	Activate Network Settings
	Reset Control System
	FTP
	Open FTP Folder on Device
	Service executed successfully!
P Settings Extended Settings	🔏 Communication 🗓 CPU Service Editor 🦨 Bus interfaces 🖹 C 🔹

Successful execution of the service will be displayed in the status window.

Figure 4-46 Status window

The input/output data ranges available for the ILC 170 ETH 2TX as PROFINET IO device are displayed under "Network Settings" -> "PROFINET Device".

ILC 170 ETH 2TX 192.168.0.7 \Extended Settings\
Image: State in the state
Image: State in the state
Image: SNMP agent 2 Output range 256 bytes Image: SNMP agent 2 0 0
ID device status ID device status Image: PROFINET Device
ID device status PROFINET Device
PROFINET Device
-
🔫 IP Settings 📴 Extended Settings 🖌 Communication 📲 CPU Service Editor 🖓 Bus interfaces 📄 Data sheet

Figure 4-47 Input/output data ranges

Setting the update task

To set the update task, select the device resource in the Bus Structure window.

• Set the update task to "DEFAULT".

🚦 Bus Structure 🛛 📮 💌 🔀	Device Details	+ 🔻 🖬
🖃 🌆 ILC170_Device	STD_RES ILC170_35 \Resource\	
□	Name	Value
R STD_RES ILC170_35		STD_CNF
🔄 📥 # INTERBUS 0 . 0	Resource name	STD_RES
	PLC type	eCLR
	Processor type	ILC170_35
	I/O Update by Task	<default></default>
	Resource	

Figure 4-48 Setting the update task

4.2.5 Observe startup behavior

To use the PROFINET device functions, the following conditions apply for the "ILC330_Controller" project.

Higher-level controller:	ILC 330 PN
Controller settings:	
IP address:	192.168.1.2
Subnet mask:	255.255.255.0
PROFINET device name:	ILC330PN1

RFC 470 PN-3TX settings as a PROFINET IO device

IP address:	192.168.1.5
Subnet mask:	255.255.255.0
PROFINET device name:	RFC470PN1

To use the PROFINET device functions, the following conditions apply for the "RFC470_Device" project.

RFC 470 PN-3TX settings as a PROFINET IO controller

IP address:	192.168.0.5
Subnet mask:	255.255.255.0
PROFINET device name:	RFC470PN1

ILC 170 ETH 2TX settings as PROFINET IO device:

IP address:	192.168.0.7
Subnet mask:	255.255.255.0
PROFINET device name:	ILC170ETH1

Please make sure that the same PROFINET device name of the RFC 470 PN-3TX IO controller (here: RFC470PN1) is used in the lower-level project as in the higher-level project for the RFC 470 PN-3TX as a device (here: RFC140PN1).

Starting up the controller is the easiest way to check whether

- The controller is correctly parameterized
- The I/O devices have the right name
- There are double names or double IP addresses in the system.

Make sure that the controller has the IP address that was set in the project. Start the project control dialog via the menu bar.

If the message "Timeout" appears after 10 seconds, the project and device addresses do not match. It is also possible that the IP address of the computer has not been set correctly.

The controller can be reset from the project control dialog. The existing project will be deleted. Start the download and perform a cold reset. Afterwards the BF LEDs must go out on all devices.

To access the network status from the program, the following system variables are mapped in the global variables of the programming environment. Activate the "Debug On" operating mode and the values of these variables will be displayed.

Global variable	Description
PNIO_CONFIG_STATUS_ACTIVE	Connection to these devices is being established or has been completed.
PNIO_CONFIG_STATUS_READY	The connection establishment to the devices has been completed.

4.2.6 Checking the program start of the higher-level project

• Open the "ILC330_Controller" project.

When the program is started correctly, the following screen will be shown in the Debug mode:

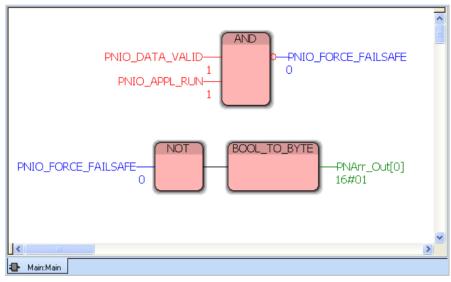
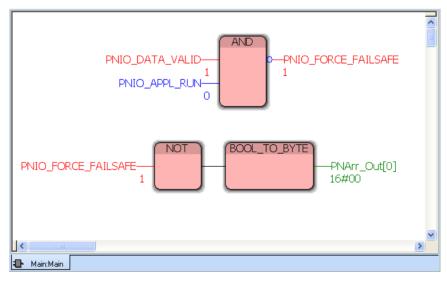
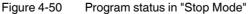


Figure 4-49 Program status

The PNIO_FORCE_FAILSAVE variable is in the FALSE state, thus communication is ensured and the outputs are set according to the process data.

If you remove the voltage connector of the RFC 470 PN-3TX or change the device to the "Stop Mode", the status of PNIO_FORCE_FAILSAFE will change to TRUE. All outputs will be set to "0" and the value "1" is no longer transferred to the device.





4.2.7 Checking the program start of the lower-level project

The behavior described before can also be observed in the ILC170_Device project. Please note that the RFC 470 PN-3TX is operating as a master and as a slave at the same time. It acts as the link between the ILC 330 PN controller and the ILC 170 2TX controller.

- Please open the lower-level project of the ILC 170 ETH 2TX.
- Then open the "Data_Acknowledge" POU and activate the Debug mode. The following screen appears:

1 FALSE ON	BOARD_OU	TPUT_BITO	:= FALSE;	ĺ
	BYTE_TO	_INT (PND_S	SIS1_INPUTS	[0]) = 1
5	then			
6 FALSE	ON	IBOARD_OUTH	PUT_BITO :=	TRUE;
8 en	d_if;			
				>
Data_Ackn				
Data_Ackn	Value	Default value	Туре	Instance
	Value	Default value	Type PND_I0_256	Instance
Variable	Value	Default value		
Variable PND_S1S1_INPUTS []		Default value	PND_10_256	STD_CNF.STD_R
Variable PND_S1S1_INPUTS []	16#00	Default value	PND_IO_256 BYTE	STD_CNF.STD_R STD_CNF.STD_R
Variable PND_S1S1_INPUTS [0] [1]	16#00 16#00	Default value	PND_IO_256 BYTE BYTE	STD_CNF.STD_R STD_CNF.STD_R STD_CNF.STD_R

Figure 4-51 Program status active

The value 1 is in array [0] of the PND_S1S1_INPUTS. The ONBOARD_OUTPUT_BIT0 variable is TRUE and the Q1 LED on the ILC 170 2TX is ON. Now switch the PROFINET IO controller (ILC 330 PN and/or RFC 470 PN-3TX) to stop. Communications is terminated and the value is set to 0. The LED goes out as well, because the ONBOARD_INPUT_BIT0 variable is reset to FALSE.

1	TRUE O	NBOARD_O	JTPUT_BITO	:= FALSE;	~
2					
3	16#01 <mark>i</mark>	f BYTE_TO	D_INT (PND_S	51S1_INPUTS	([0]) = 1
4					
5		then			
6	TRUE	or	iboard_outi	PUT_BITO :=	TRUE;
7					
8	e	nd_if;			
9					~
					>
Data_Ac	,				<u></u>
	skn	Value	Default value	Туре	Instance
Data_Ac	skn		Default value	Type PND_10_256	
Data_Ac	skn		Default value		Instance
Data_Ac	ckn S1S1_INPUTS [0]	6	Default value	PND_10_256	Instance STD_CNF.STD_R
Data_Ac	skn 	16#01	Default value	PND_IO_256 BYTE	Instance STD_CNF.STD_R STD_CNF.STD_R
Data_Ac	xn S1S1_INPUTS [0] [1] [2]	16#01 16#00	Default value	PND_IO_256 BYTE BYTE	Instance STD_CNF.STD_R STD_CNF.STD_R STD_CNF.STD_R
Data_Ac	xn S1S1_INPUTS [0] [1] [2]	16#01 16#00 16#00 16#00		PND_IO_256 BYTE BYTE BYTE	Instance STD_CNF.STD_R STD_CNF.STD_R STD_CNF.STD_R STD_CNF.STD_R

Figure 4-52 Program is stopped

When communication is interrupted by removing the voltage connector of the ILC 170 device, a BF error appears on the RFC display at the PROFINET controller.



If you need more detailed information, call the Diag+ diagnostic tool from PC WorX under View-> Diag+. Here you connect explicitly to a controller and receive further information.