

<u>Introduction</u>	<b>1</b>
<u>Quick start for Windows</u>	<b>2</b>
<u>Quick start for Linux</u>	<b>3</b>
<u>Hardware configuration in the TIA Portal</u>	<b>4</b>
<u>Hardware configuration of the application examples</u>	<b>5</b>

## Legal information

### Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

#### **DANGER**

indicates that death or severe personal injury **will** result if proper precautions are not taken.

#### **WARNING**

indicates that death or severe personal injury **may** result if proper precautions are not taken.

#### **CAUTION**

indicates that minor personal injury can result if proper precautions are not taken.

#### **NOTICE**

indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

### Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

### Proper use of Siemens products

Note the following:

#### **WARNING**

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

### Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

### Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

# Table of contents

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
1.1	Overview .....	4
1.2	Security information .....	8
<b>2</b>	<b>Quick start for Windows .....</b>	<b>9</b>
2.1	Installing WinPcap Developer's Pack .....	9
2.2	Installing WinPcap .....	10
2.3	Installing Visual Studio Express 2012.....	10
2.4	Disabling network protocols that are not utilized .....	11
2.5	Disabling the PNIO adapter of the PG/PC interface.....	12
2.6	Start application example .....	13
<b>3</b>	<b>Quick start for Linux .....</b>	<b>16</b>
3.1	Installing Linux Debian.....	16
3.2	Enabling admin rights for the user .....	18
3.3	Installing real-time kernel.....	19
3.4	Installing other required packages.....	20
3.5	Using the PN Device driver.....	21
3.5.1	Unloading other network drivers .....	22
3.5.2	Loading the PN Device Driver .....	22
3.5.3	Unloading the PN Device Driver .....	23
3.5.4	Loading the PN Device Driver automatically .....	24
3.5.5	Compiling the PN Device Driver .....	26
3.5.6	Determining errors during loading of the PN Device Driver.....	27
3.6	Installation and setup of Eclipse IDE .....	28
3.7	Start application example .....	30
<b>4</b>	<b>Hardware configuration in the TIA Portal .....</b>	<b>33</b>
4.1	Installing the Hardware Support Package for PN Driver V1.1 .....	33
4.2	Generating an XML configuration file .....	36
<b>5</b>	<b>Hardware configuration of the application examples .....</b>	<b>38</b>

# Introduction

## 1.1 Overview

### Purpose of this documentation

These compact instructions aim to provide you with a quick introduction to using the PN Driver.

### Target group of the documentation

These compact instructions are intended for software developers and should help them to create an executable user program in a very short amount of time. This requires the following basic knowledge:

- Programming experience with C/C++
- User expertise in the Windows and LINUX operating systems
- Experience with PROFINET IO systems
- Basic knowledge of the configuring software STEP 7, TIA Portal
- General knowledge of automation technology

### What is the PN Driver?

The PN Driver is a PROFINET IO controller development kit for standard Ethernet ports.

### What are the requirements?

In addition to the PN Driver-software, you need a PC with an Ethernet port. Windows or Linux can be installed as operating system. For Linux, you must use the Intel Ethernet adapters I210 ("Springville") or 82574L ("Hartwell").

If the PN Driver-software is operated under **Windows**, the WinPcap driver must be installed. To create your own user programs based on the IO-Base API, we recommend using a development environment, such as Microsoft Visual Studio 2012.

If the PN Driver-software is operated under **Linux**, the supplied PNDevDriver must be installed. We recommend using the Eclipse development environment under Linux.

Even though a hardware configuration can also be created without an existing engineering system, we recommend using the TIA Portal engineering system (no license required).

## **New functions in PN Driver V1.1**

- Linux support
- Hardware configuration with STEP 7 (TIA Portal) V13 SP1 and higher + HSP0074
- Configuration control for IO devices
- Configuration control for IO systems
- Optional IO devices
- Flexible topology
- Multiple use IO systems
- Shared device
- Diagnostics service
- PROFINET IO functions V2.3:
  - RT advanced startup
  - IEEE-compliant LLDP (Link Layer Discovery Protocol)
- Plug-in PROFINET interface submodules for Linux and Windows versions
- Support of prioritized startup for the Linux version

---

### **Note**

Regarding support of prioritized startup for the Linux version, note the restriction that the neighboring PROFINET device in the topology must not support prioritized startup.

---

## Application examples

The PN Driver software package includes a set of application examples which offer a quick start into programming your own applications. The application examples also provide an overview of the new functions of the PN Driver V1.1.

---

### Note

You will receive the source code of the PN Driver V1.1 from us as well as the source code of the application examples for modification and editing only in connection with SIMATIC. You are not permitted to use the source code of the PN Driver and the application examples without SIMATIC or to give it to third parties.

The application examples are non-binding and do not claim to be complete regarding configuration and equipment as well as all eventualities. The application examples do not represent customer-specific solutions but are merely to serve as an aid for typical tasks. You are responsible for proper operation of the described products. These application examples do not release you from the obligation of a safe approach during application, installation, operation and maintenance. By using these application examples, you accept the fact that we are not liable for any damages beyond the scope of the described liability provision. We reserve the right to make changes to these application examples at any time without prior notice. In case of deviations between the proposals in this application example and other Siemens publications, such as catalogs, the content of the other documentation has priority.

We are not liable for information contained in this document.

Our liability for damages resulting from using the examples, notes, programs, configuration and performance data, etc. described in this application example regardless of the legal background is excluded unless required by law, e.g. in cases of willful misconduct, gross negligence, personal injury or death, failure to achieve guaranteed characteristics, fraudulent concealment of a defect or in case of breach of fundamental contractual obligations. The damages in case of a breach of fundamental contractual obligations, however, is limited to the foreseeable damages typical of the contract, unless intention or gross negligence are present or in case of mandatory liability due to injuries to life, body or health. A change in the burden of proof to your disadvantage is not associated with this provision.

Distribution or reproduction of these application examples or excerpts thereof is not permitted unless expressly granted by the Siemens AG.

---

## Technical specifications

PN Driver V1.1	Windows	Linux
Max. number of PROFINET devices	16	128
Max. number of Fast Startup devices	0	8
Supported send cycles	32 ms/ 4 frames per 32 ms	1, 2, 4 ms/ 13 frames per 1 ms
Data consistency submodule	254 bytes	254 bytes
Max. number of input addresses	8192	8192
Max. number of output addresses	8192	8192
Max. data record length	32768 bytes	32768 bytes

## Conventions

The terms "PROFINET Driver for controller" and "PN Driver" are used synonymously in this manual.

Please also observe notes marked as follows:

---

### Note

A note contains important information on the product described in the documentation, on the handling of the product, or on the section of the documentation to which particular attention should be paid.

---

## Overview of the supplied documentation

Documentation	Most important contents
PROFINET IO-Base user programming interface manual	This document describes the IO-Base API which represents the interface for creating your own user programs.
PROFINET Driver for Controller Engineering Interface programming and operating manual	If an engineering system is not available, hardware configuration can also take place by creating an XML configuration file. This document describes how you can create a hardware configuration based on an XML file.
How to Port PN Driver V1.1 parameter manual	This document describes how to port the PN Driver software to another operating system or to a different hardware platform.

## Additional support

If you have questions regarding the described PN Driver that are not addressed in the documentation, please contact your local representative at the Siemens office nearest you.

Please send questions, comments and suggestions regarding this manual in writing to the specified e-mail address.

In addition, you will find general information, current product information, FAQs and downloads that can be useful on the Internet (<http://www.siemens.com/comdec>).

### Technical contact information for Germany/worldwide

Siemens AG	Phone: +49 911 750 2080
ComDeC ( <a href="http://www.siemens.com/comdec">http://www.siemens.com/comdec</a> )	Phone: +49 911 750 4384
	Phone: +49 911 750 2078
	Fax: +49 911 750 2100
	E-mail: ( <a href="mailto:ComDeC@siemens.com">mailto:ComDeC@siemens.com</a> )
Office address:	Postal address:
Breslauer Str. 5	P.O. Box 2355
90766 Fürth, Germany	90713 Fürth, Germany

### Technical contact information for the U.S.

PROFI Interface Center	Phone: +1 (423) 262-2576
( <a href="http://www.profiinterfacecenter.com">http://www.profiinterfacecenter.com</a> )	Fax: +1 (678) 297-7289
One Internet Plaza	E-mail: ( <a href="mailto:PIC.industry@siemens.com">mailto:PIC.industry@siemens.com</a> )
Johnson City, TN 37604	

### Technical contact information for China

PROFI Interface Center China	Fax: +86-10-6476 4725
7, Wangjing Zhonghuan Nanlu	E-mail: ( <a href="mailto:Profinet.cn@siemens.com">mailto:Profinet.cn@siemens.com</a> )
100102 Beijing	

## 1.2 Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. You can find more information about industrial security on the Internet (<http://www.siemens.com/industrialsecurity>).

To stay informed about product updates as they occur, sign up for a product-specific newsletter. You can find more information on the Internet (<http://support.automation.siemens.com>).



## Quick start for Windows

### Overview

This section describes the most important steps required for commissioning the PN Driver under Windows 7. PN Driver V1.1 was not tested with other Windows versions.

### First step

1. Copy the "/src/" folder on the CD to a path on your Windows PC for which you have write permission.

Then execute the steps described in the following sections.

## 2.1 Installing WinPcap Developer's Pack

### Procedure

1. Download the WinPcap Developer's Pack from the website (<http://www.winpcap.org/devel.htm>).
2. Create a folder called "driver" under the path "[..]\src\".
3. Extract the downloaded zip file and copy the folder "WpdPack" to the folder "[..]\src\driver\".

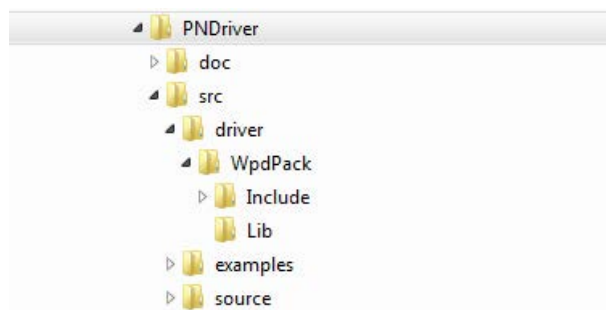


Figure 2-1 PN Driver file structure

The test application uses the files "wpcap.lib" and "packet.lib" to access the local network adapters.

PN Driver V1.1 was tested with version 4.1.2 of the developer's pack.

## 2.2 Installing WinPcap

### Procedure

1. Download the WinPcap driver for Windows from the WinPcap website (<http://www.winpcap.org/>).
2. Perform the installation.

PN Driver V1.1 was tested with version 4.1.3 of the WinPcap driver for Windows.

## 2.3 Installing Visual Studio Express 2012

The Microsoft Visual Studio Express 2012 development environment lets you run the test application provided on the CD under Windows or you can develop your own application on PN Driver.

### Procedure

1. Download Microsoft Visual Studio Express 2012 from the Microsoft-website (<http://www.microsoft.com/en-US/download/details.aspx?id=34673>).
2. Perform the installation.

## 2.4 Disabling network protocols that are not utilized

PN Driver only uses the Ethernet network adapter. It is important that all other network protocols (TCP/IP, File and Printer Sharing, Client for Microsoft Networks, etc.) are disabled.

### Procedure

1. Make sure that all network protocols except the Ethernet network adapter are disabled; if not, disable them.

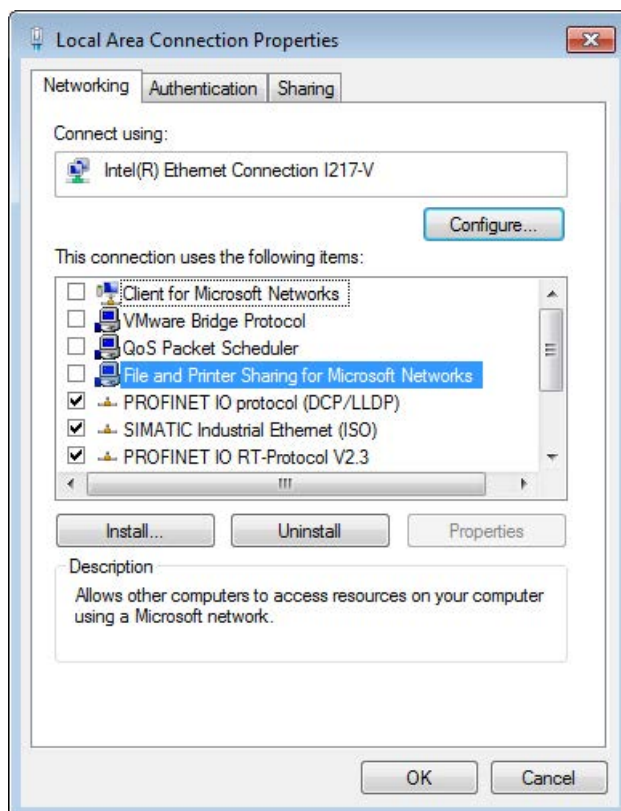


Figure 2-2 Disabling network protocols

## 2.5 Disabling the PNIO adapter of the PG/PC interface

If a copy of a configuration software such as STEP 7 Professional or TIA Portal is installed on your computer, the PNIO adapters of the PG/PC interface must be disabled.

### Procedure

1. Ensure that all PNIO adapters of the PG/PC interface are disabled; disable them if necessary.

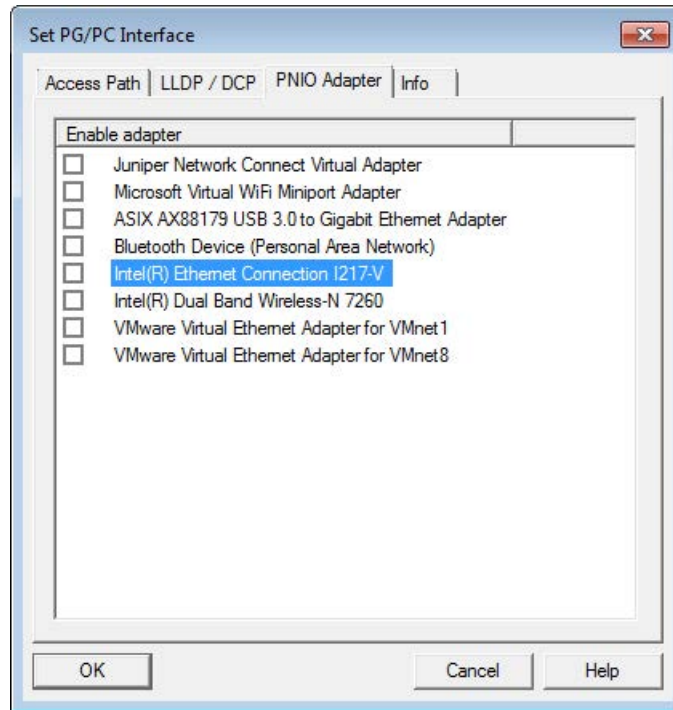


Figure 2-3 Disabling PNIO adapters of the PG/PC interface

## 2.6 Start application example

The enclosed CD includes six application examples. The execution of the "test\_app" application example is described below as an example.

### Requirement

The following paths are important for executing the application example:

- Microsoft Visual Studio 2012 project folder:  
"[.]\src\examples\test\_app\win32\PNDRIVER\_Glob\_Build.sln"
- Source files: "[.]\src\examples\test\_app\src\"
- Example XML configuration file: "[.]\src\examples\test\_app\win32\Station\_1.PN  
Driver\_1.PNDriverConfiguration.XML"

### Procedure

1. Start Visual Studio Express 2012 and open the "Solution" on the CD under the path "[.]\src\examples\test\_app\win32\PNDRIVER\_Glob\_Build.sln". Two projects are loaded in this step: "pnedriver\_w32" and "test\_app".

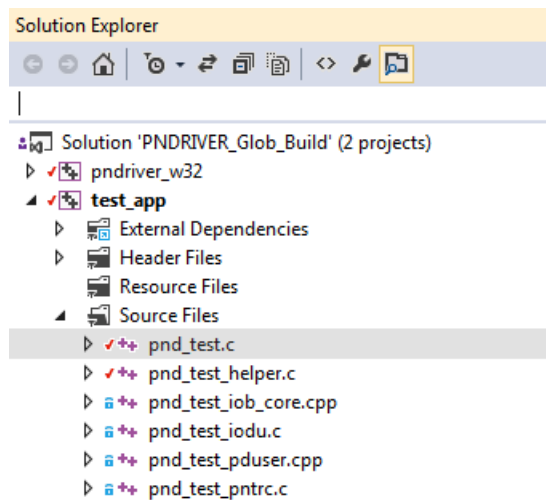


Figure 2-4 Solution Explorer

Before the PN Driver can be started or debugging can take place, you must ensure that the application can find the related XML configuration file. A prepared XML configuration file is available in the directory of the application examples. If you want to control a different HW configuration with this application, you must copy the configuration file to the working directory. See section Hardware configuration of the application examples (Page 38).

2. Under the "test\_app" project, open the file "pnd\_test.c".

- Set a breakpoint on the "int main()" function and start the program.  
When you start the program for the first time, the projects are initially compiled.

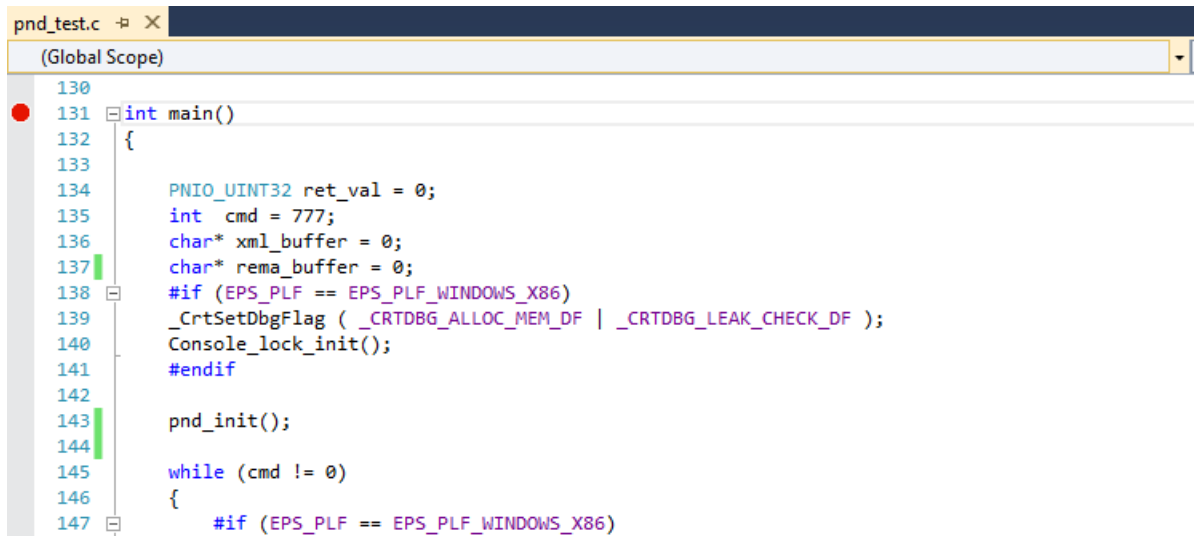


Figure 2-5 Set breakpoint

If the program is executed further, the functions that can be executed by the test application are listed.

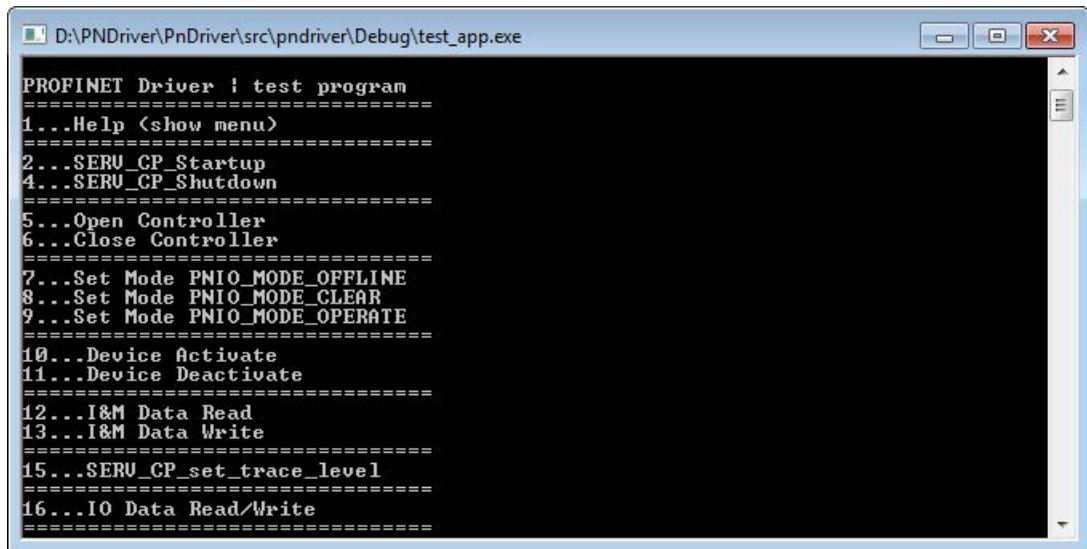
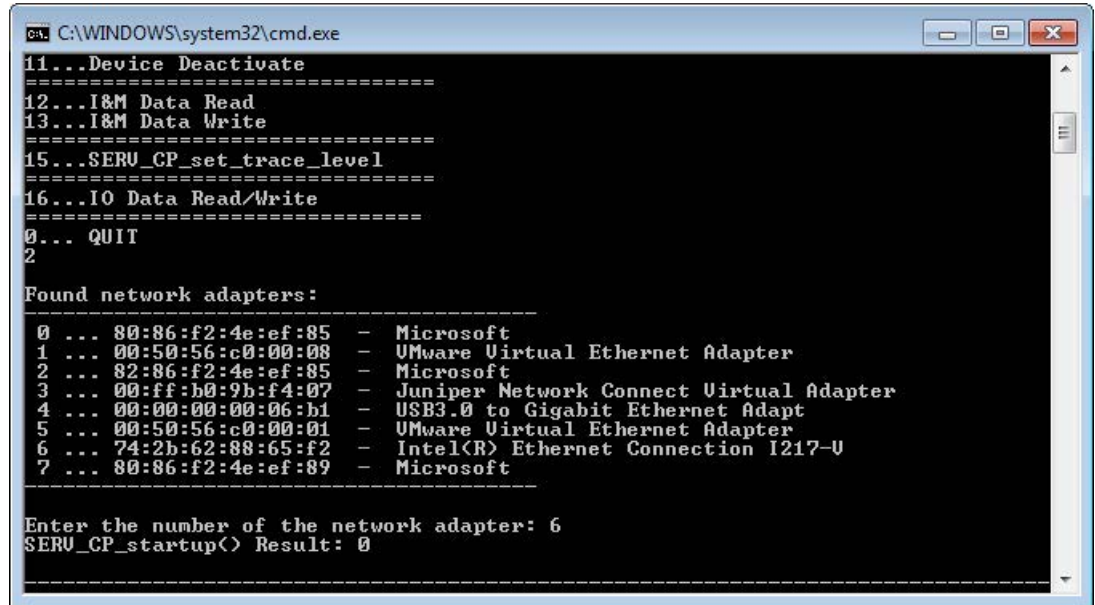


Figure 2-6 Menu of the test application

4. Enter "2" to call the start function. If the WinPcap driver was installed correctly, a list with the network adapters is created. Select the number of the network adapter that you want to use for the PROFINET communication. When the HW configuration was downloaded without errors and the PN Driver is started, "0" is output as return value.



```
C:\WINDOWS\system32\cmd.exe
11...Device Deactivate
=====
12...I&M Data Read
13...I&M Data Write
=====
15...SERV_CP_set_trace_level
=====
16...IO Data Read/Write
=====
0... QUIT
2
Found network adapters:
=====
0 ... 80:86:f2:4e:ef:85 - Microsoft
1 ... 00:50:56:c0:00:08 - VMware Virtual Ethernet Adapter
2 ... 82:86:f2:4e:ef:85 - Microsoft
3 ... 00:ff:b0:9b:f4:07 - Juniper Network Connect Virtual Adapter
4 ... 00:00:00:00:06:b1 - USB3.0 to Gigabit Ethernet Adapt
5 ... 00:50:56:c0:00:01 - VMware Virtual Ethernet Adapter
6 ... 74:2b:62:88:65:f2 - Intel(R) Ethernet Connection I217-U
7 ... 80:86:f2:4e:ef:89 - Microsoft
=====
Enter the number of the network adapter: 6
SERV_CP_startup(>) Result: 0
=====
```

Figure 2-7 List of network adapters

### Note

For an application developed for PN Driver V1.0 to run under PN Driver V1.1, you must make the following changes in the source code:

1. When starting the PN Driver, insert the new function "SERV\_CP\_init" before calling the function "SERV\_CP\_startup".
2. Update the function "SERV\_CP\_startup" according to the specification in the manual "PROFINET IO-Base user programming interface".
3. When shutting down the PN Driver, call the function "SERV\_CP\_undo\_init" after calling the function "SERV\_CP\_shutdown".

Details on the respective functions are available in the manual "PROFINET IO-Base user programming interface".

## Quick start for Linux

### Overview

This section describes the most important steps required for commissioning the PN Driver under Linux.

The following statements refer to Linux Debian 7.6 with kernel V3.2 including RT patch as well as GRUB as bootloader.

### 3.1 Installing Linux Debian

Linux Debian images are available on the Debian website (<https://www.debian.org/>) and can be installed with a USB stick. For installation, you can copy the \*.iso image to a USB stick (for example, with UNetbootin).

The USB stick must be formatted in FAT32 and should be blank.



## Procedure

1. Install Linux Debian.
  - You can use the default settings during installation.
  - User name and password as well as the Superuser(Root) password must be adapted to comply with your security regulations.

As soon as the installation is complete and the PC was restarted, the login window is displayed.

2. Enter the previously specified user name and the specified password.

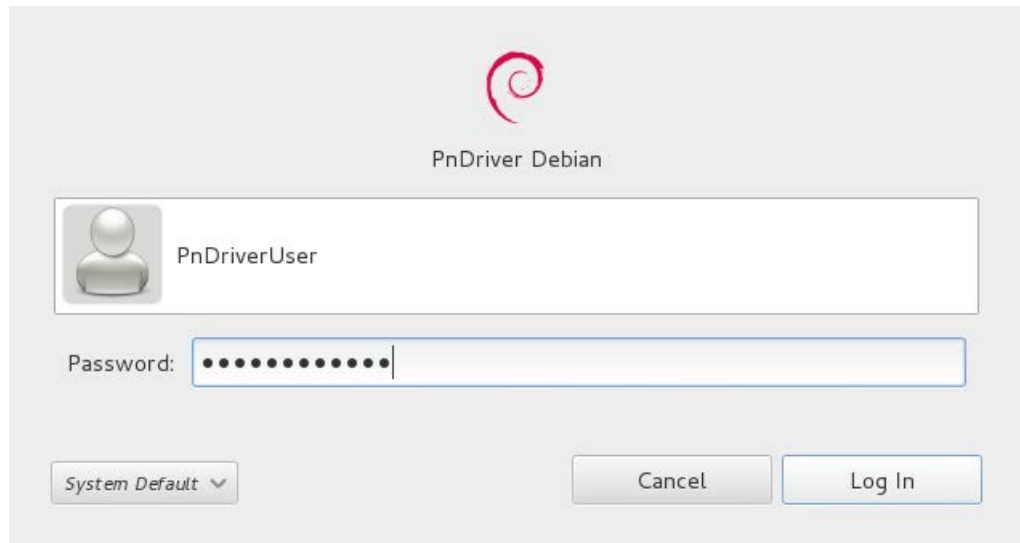


Figure 3-1 Linux Debian login

## 3.2 Enabling admin rights for the user

You need superuser rights for the following steps. To get these rights, proceed as follows.

### Procedure

1. In the terminal program ("Applications > Terminal"), enter "su".
2. Confirm the password prompt with the superuser password specified during installation.
3. Add your user to the superuser group ("sudo") by entering "`adduser PnDriverUser sudo`". ("PnDriverUser" is the user name used in the example).

A screenshot of a Linux terminal window titled "PnDriverUser@PnDriver Debian: ~". The terminal shows the following sequence of commands and output: 

```
PnDriverUser@PnDriver Debian:~$ su
Password:
root@PnDriver Debian:/home/PnDriverUser# adduser PnDriverUser sudo
Adding user `PnDriverUser' to group `sudo' ...
Adding user PnDriverUser to group sudo
Done.
root@PnDriver Debian:/home/PnDriverUser# exit
exit
PnDriverUser@PnDriver Debian:~$
```

Figure 3-2 Enabling admin rights

4. Log off with the current user and log on again for the changes to become effective.

## 3.3 Installing real-time kernel

### Procedure

1. To ensure the real-time capability of the Linux system, download and install the real-time kernel from the repository:

- for 32-bit systems:

```
sudo apt-get install linux-image-3.2.0-4-rt-686
```

- for 64-bit systems:

```
sudo apt-get install linux-image-3.2.0-4-rt-amd64
```

2. When the CPU is not active, parts of it are switched to idle mode. This conserves power for the PC but it takes longer for the CPU until it can compute actively. This wake up mode slows down cyclic tasks with short cycle times (ms and  $\mu$ s). You therefore change the scheduling mechanism of the real-time kernel by adding the boot parameter "idle=poll" below the kernel you use in the file "/boot/grub/grub.cfg".

- Example:

```
linux /boot/vmlinuz-3.2.0-4-rt-686-pae root=UUID=xxx ro quiet idle=poll
```

If no desirable kernel with real-time support ("rt") is available, you must download the source code files of the required kernel, edit them with a Realtime preemption (RT-Preempt) patch and compile the kernel yourself. You can download the kernel and the available Realtime patches at kernel.org (<ftp://ftp.kernel.org/pub/linux/kernel>).

## 3.4 Installing other required packages

### Requirements

You may need the following to execute the instructions listed below:

#### Proxy server

1. If a proxy server is required, enter the following in the terminal program:

```
sudo nano /etc/apt/apt.conf.d/80proxy
```

2. Enter the respective proxy in the file:

```
Acquire::http::Proxy "..."; //(..." = Proxy)
```

3. Exit the input with Ctrl + X (Exit).
4. Save the file with "y" (Yes).

#### Sources for updates

1. To store the sources for the updates, open the following file in the terminal program:

```
sudo nano /etc/apt/sources.list
```

2. There comment out "cdrom".

3. Comment in the following:

```
deb http://ftp.debian.org/debian/ wheezy-updates main contrib
deb-src http://ftp.debian.org/debian/ wheezy-updates main contrib
```

4. Add:

```
deb http://ftp.debian.org/debian/ wheezy main contrib
deb-src http://ftp.debian.org/debian/ wheezy main contrib
```

5. Save the list. Then enter the following in the terminal program:

```
sudo find /var/lib/apt -type f -exec rm{} \+
sudo apt-get update
sudo apt-get dist-upgrade
```

## Procedure

1. To install the required Linux header files, enter the following in the terminal program:

```
sudo apt-get install linux-headers-3.2.0-4-rt-686
```

Make sure that the headers match the currently used kernel. You can read out the kernel version with the command "uname -a".

2. To build the PN Driver application, you must install "gcc", "make" and additional packages. To do so, enter the following in the terminal program:

```
sudo apt-get install build-essential
```

3. Install the following packages to get the debugging option:

```
sudo apt-get install gdb gdbserver gdb-multiarch
```

4. The PN Driver is a 32-bit application. If you have a 64-bit Linux Debian installation but want to run 32-bit applications, you must add the i386 architecture to the package manager and update the package database. To do so, enter the following in the terminal program:

```
sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get install libc6-dev-i386
sudo apt-get install g++-multilib
```

Next you must install the matching C++ standard library:

```
sudo apt-get install libstdc++6:i386
```

You also need the g++ compiler for 32-bit applications:

```
sudo apt-get install g++-multiarch
```

5. Install the Secure Shell (SSH) if the Linux system is to be accessed by a Windows system with PuTTY (or similar). To do so, enter the following in the terminal program:

```
sudo apt-get install ssh
```

You will find a good description of the SSH setup on the Internet.

## 3.5 Using the PN Device driver

### Overview

To use the network adapters under Linux, the PN Driver requires the PN Device Driver (PNDevDriver). It is already available as compiled kernel object for Intel processors in a 32-bit and a 64-bit version (\src\source\pndevdrv\PnDev\_Drv\bin\Linux\\*.ko). If these modules do not suit your architecture, you must compile the modules yourself as described below.

---

#### Note

When we talk about "32-bit" or "64-bit" in this section, we are always referring to the installed operating system and **not** the application. This means you use a 64-bit kernel module for a 64-bit operating system even when a 32-bit application is being executed.

---

### 3.5.1 Unloading other network drivers

During the loading, the PNDevDriver is looking for all supported cards. These also include the Intel chips I210 and 82574L which may be installed on your PC by default. In order to use these network interfaces, the driver for these network cards must be unloaded beforehand.

#### Procedure

1. Enter the following in the terminal program:

- for Intel I210:  
`sudo modprobe -r igb`
- for Intel 82574L:  
`sudo modprobe -r e1000e`

The drivers are now temporarily disabled and do not interfere when loading the PNDevDriver. However, at the next restart they will automatically be installed again. A description of how you can prevent loading of the drivers during restart is available in the section Loading the PN Device Driver automatically (Page 24).

### 3.5.2 Loading the PN Device Driver

#### Procedure

1. To load the PN Device Driver temporarily (until the next restart), use the following command in the same folder in which the "PnDevDrv\_\*.ko" module is located:
  - for 32-bit systems:  
`sudo insmod PnDevDrv_32.ko`
  - for 64-bit systems:  
`sudo insmod PnDevDrv_64.ko`
2. If you do not want your PNDevDriver to use all supported cards of your system, you can start the module with a parameter for it to use the PNDevDriver only for specific cards:  
`sudo insmod PnDevDrv_64.ko PciLoc="3/0/0","4/0/0"`

When starting the driver in this example, the PNDevDriver is **only** used for the network adapters at the PCI location 3,0,0 and 4,0,0. You can determine the PCI location with "lspci". This prevents that a network card installed by default is used by the PNDevDriver.

### 3.5.3 Unloading the PN Device Driver

If the application you have developed crashes, we recommend that you reload the PN Device Driver.

#### Procedure

1. Enter the following in the terminal program:
  - for 32-bit systems:  
`sudo rmmmod PnDevDrv_32`
  - for 64-bit systems:  
`sudo rmmmod PnDevDrv_64`
2. Then you can load the PN Device Driver once again (see section Loading the PN Device Driver (Page 22)).

The error message "Error: Module PnDevDrv\_xx is in use" indicates that the driver is being used by another application in the system. This application must be closed first.

### 3.5.4 Loading the PN Device Driver automatically

Linux supports automatic loading of drivers during booting. For this, you must do the following:

#### Procedure

1. Copy the kernel module `"*.ko"` to a folder that is definitely available during system start, e.g. `"/usr/local/"`. (In case of encrypted home drives, `"/home/<Username>/..."` might not be readable.)
2. For Linux to find the driver during booting, a symbolic link to the kernel module must be created:

```
sudo ln -s /usr/local/PnDevDrv_32.ko /lib/modules/3.2.0-4-rt-amd64
```

The input `"3.2.0-4-rt-amd64"` in the example depends on the installed real-time kernel. Make sure that the path information is correct.

3. Then you must add the driver to the list of modules to be loaded (`"/etc/modules"`):

```
gksu gedit /etc/modules
```

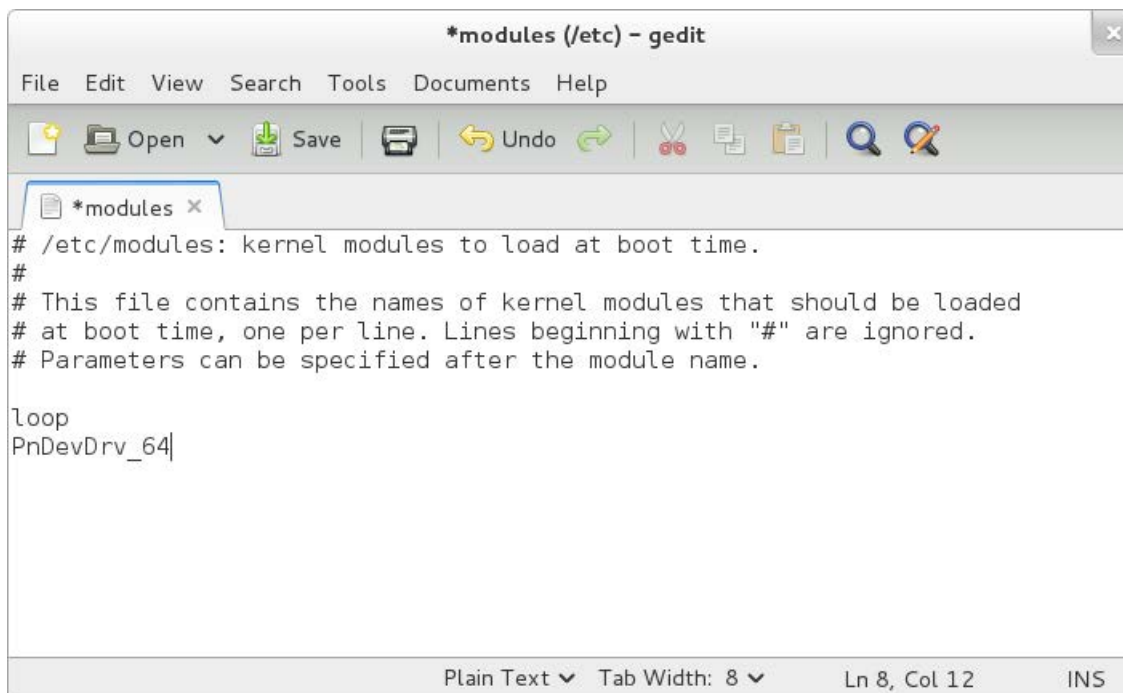


Figure 3-3 Loading the PN Device Driver automatically

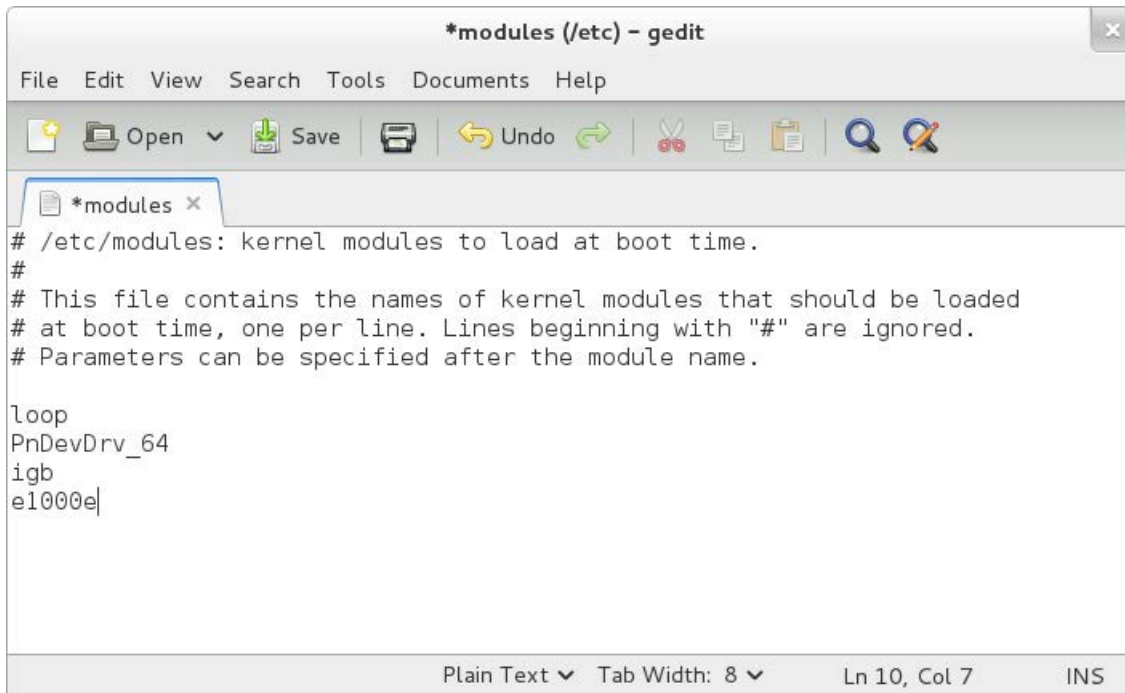
The list has been expanded by the `PnDevDrv_64`. The extension `".ko"` must not be specified. You can also add the addition `"PciLoc="3/0/0"` in this case.

4. For the `PNDevDriver` to be able to use the Intel network cards of type `I210` and `82574L`, these must be "blacklisted" during booting. To do so, create a file `"/etc/modprobe.d/blacklist-pndriver.conf"` with the following content:

```
blacklist igb
blacklist e1000e
```



5. If these drivers are to be used after starting the PNDevDriver, they must also be added to the list in "/etc/modules" but **after** the PnDevDrv\_64.



```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

loop
PnDevDrv_64
igb
e1000e
```

Figure 3-4 Loading other drivers automatically

6. Linux updates its internal start routines with the following two commands so that the driver can be loaded during the next restart:

```
sudo depmod -a
sudo update-initramfs -u
```

After a restart, the PNDevDriver is now loaded during each boot process.

### 3.5.5 Compiling the PN Device Driver

If the supplied kernel modules do not work, you must compile the kernel module again. To do so, you need the directories "src/source/pndevdrv" and "src/source/pnio/src/common/".

#### Procedure

1. In the first line of your Makefile "src/source/pndevdrv/PnDev\_Driver/bin/Linux/Makefile", adapt the path to your environment; the path must point to the folder that includes "/pndevdrv" and "/pnio", e.g. `VIEW:= /usr/src`
2. When you have installed all the build tools mentioned above (see section Installing other required packages (Page 20)), you can now generate the PN Device Driver. To do so, enter the following:
  - in 32-bit systems:  
`sudo make modules`
  - in 64-bit systems:  
`sudo make modules IS_64=1`

The result is your own "\*.ko" file which can then be loaded.

Keep in mind that a "\_32.ko" file can only be compiled and loaded on a 32-bit Linux Debian; a "\_64.ko" file can only be used with 64-bit systems.

3. In addition to the actual driver, a shared object file must be generated; this file corresponds to a DLL in Windows and is the link between application and driver. You can generate this shared object file with the "Makefile\_SharedObject" in "src/source/pndevdrv/PnDev\_Driver/bin/Linux":
  - in 32-bit systems:  
`make -f Makefile_SharedObject PnDev_DriverU32.so CCFLAGS=-DMAP_32BIT=0`
  - in 64-bit systems:  
`make -f Makefile_SharedObject all`

As a result, the two shared object files (\*.so) for 32-bit and 64-bit applications are generated and stored in the same folder.

### 3.5.6 Determining errors during loading of the PN Device Driver

If loading of the PNDevDriver ("insmod") were to fail, you can output the last system messages and try to determine what caused the error.

#### Procedure

1. To do so, enter:

```
dmesg
```

Possible causes of error are:

- Insufficient memory for mapping (for 32-bit systems)
  - Error message during loading: "Permission denied!" - `dmesg` indicates that "vmap" could not be executed
  - Remedy: Start Linux 32-bit systems with the boot parameter "vmalloc=xxxM".
- Other drivers loaded for the network adapters (igb, e1000e)
- Driver is already loaded
- Insufficient coherent memory available (for DMA transfer).
  - Remedy: Load driver during booting when memory is not yet very fragmented.
- Path to "\*.ko" file incorrect (with loading during boot process)

## 3.6 Installation and setup of Eclipse IDE

The Eclipse development environment is suited for starting and debugging the PN Driver under Linux.

### Procedure

1. Under Linux Debian, you can install Eclipse with the package manager "apt-get". To do so, enter:  

```
sudo apt-get install eclipse-cdt
```
2. Start Eclipse for debugging with superuser (root) rights:  

```
sudo eclipse
```
3. After the installation, you can load the test application supplied with the PN Driver in Eclipse with **File > Import > General > Existing Projects into Workspace**.
4. In the **Import** dialog as root directory, navigate to the CD directory "src/examples". Eclipse now displays all existing projects.

5. Select the two projects "pndriver\_32\_native" and "test\_app" for the test application and click "Finish".

Once these projects are imported into Eclipse, you can use the test application for debugging. All settings required for the build process are already implemented.

6. Compile the two projects with **Project > Build All**. The compilation process uses a makefile that is available in the directory "src/examples/test\_app/linux32/build".

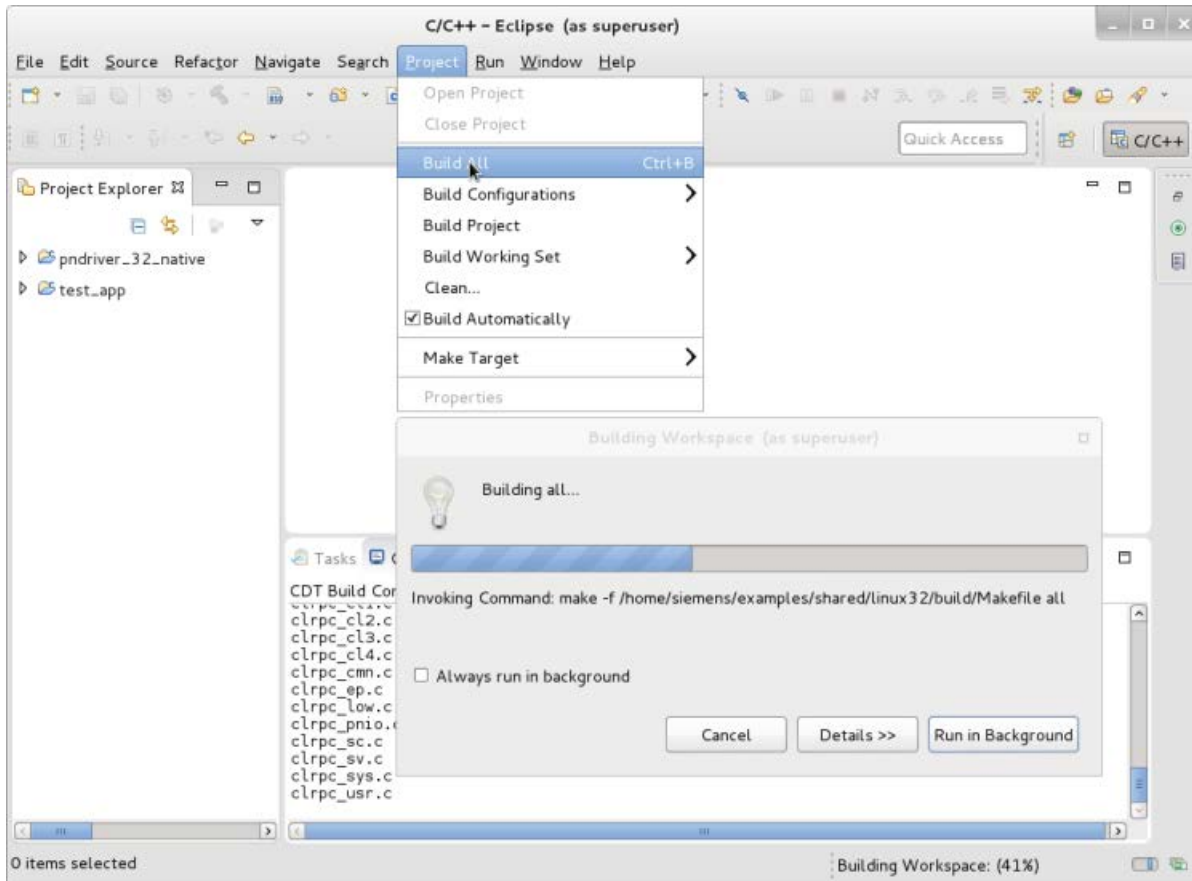


Figure 3-5 Eclipse - Build All

During debugging you must ensure that the starting conditions regarding the XML file and the shared object file are observed. Both files must be located in the same directory as the executed PN Driver. See section Start application example (Page 30).

## 3.7 Start application example

### Requirements

Before you start the PN Driver or run debugging, you must ensure that the application can locate the corresponding XML file as well as the shared object file of the PN Device Driver. Both files must be stored in the same directory as the executable PN Driver file. The directory for the test application is "src/examples/test\_app/linux32".

- This directory already includes a prepared test XML file.
- For communication between the PN Driver and the PNDevDriver, the associated shared object file "PnDev\_DriverU32.so" must be located. This file must also be stored in the same directory and is available in the directory "src/source/pndevdrv/PnDev\_Driver/bin/Linux/" for the supplied and compiled PN Device Driver.
- Because the PN Driver changes the thread priority in the real-time kernel, the application must be started with superuser permissions. For debugging, this is also true for the Eclipse development environment from which the application is started.

---

### Note

When you start the PN Driver application in Eclipse in debugging mode, the application may no longer respond. Exit Eclipse and restart it.

To avoid errors, start the PN Driver application in Eclipse in Run mode.

---

## Procedure

1. You can now create a breakpoint with Eclipse in the "main()" function ("src/examples/test\_app/src/pnd\_test.c") which stops the program at the beginning during debugging.
2. For debugging, right-click the "test\_app" project and select **Debug As > 1 Local C/C++ Application**.

---

### Note

To detect crashes in the application, a time-out of 2 s is executed when you start the PN Driver. You can disable this timeout, which disrupts debugging, in the file "src/source/pnd/src/agent/pnd\_agent.cpp", function "pnd\_agent\_startup", when calling "eps\_open(xxx, xxx, 2);". To do so, replace the last parameter (timeout period in s, here: "2") with "0" (infinite waiting).

---

3. After starting the PN Driver (e.g. `sudo ./PnDriver` or with Eclipse), it will show an overview of all detected and supported Ethernet adapters. By entering "2" (for "SERV\_CP\_startup"), you can select a card that is to be used by the PN Driver.

## 3.7 Start application example

```

sudo ./PNDriver

EPS PnDevDrv (P02.10.00.00_00.01.01.01) install done

EPS PnDevDrv Found PN Boards:
-----
Bus  3, Dev  0, Fct  0 - Hartwell I82574
-----

-----
PROFINET Driver | test program
=====
1...Help (show menu)
=====
2...SERV_CP_Startup
4...SERV_CP_Shutdown
=====
5...Open Controller
6...Close Controller
=====
7...Set Mode PNIO_MODE_OFFLINE
8...Set Mode PNIO_MODE_CLEAR
9...Set Mode PNIO_MODE_OPERATE
=====
10...Device Activate
11...Device Deactivate
=====
12...I&M Data Read
13...I&M Data Write
=====
15...SERV_CP_set_trace_level
=====
16...IO Data Read/Write
=====
0... QUIT
2

PnDevDrv found network adapters:
-----
 0 ... Bus  3, Dev  0, Fct  0 - I82574 board
-----

Enter the number of the network adapter: 0

```

Figure 3-6 Selecting a network adapter



# Hardware configuration in the TIA Portal

## Overview

This section describes the basic steps for generating the required XML configuration files with the TIA Portal.

STEP 7 (TIA Portal) as of V13 SP1 is used as a basis. You also need the Hardware Support Package for the PN Driver V1.1 (HSP0074).

## 4.1 Installing the Hardware Support Package for PN Driver V1.1

### Requirements

- You have created and opened a new project.
- The project view of the TIA Portal is open.

## Procedure

1. Click "Support Packages" under "Options".  
The "Detailed information" dialog opens.

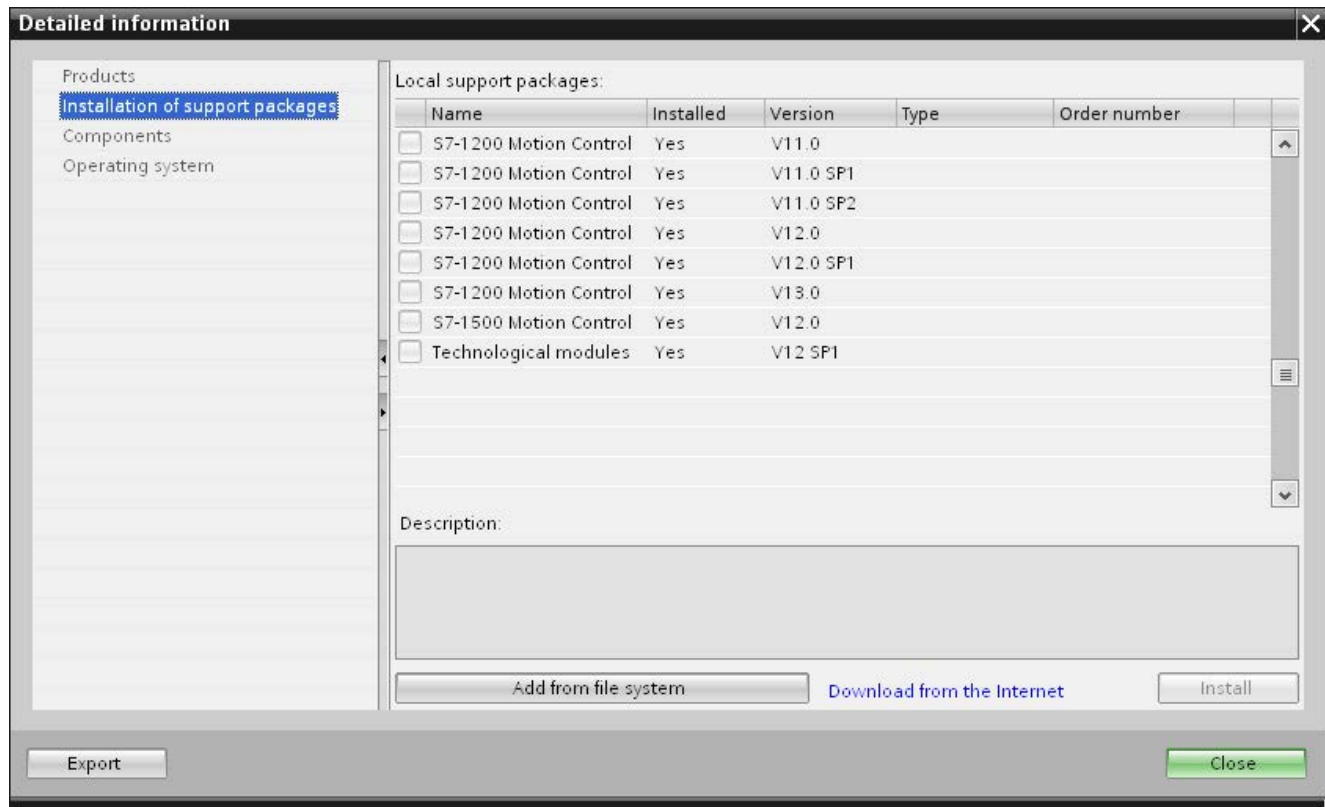


Figure 4-1 Installing support packages

2. You have the following options:
  - If the support package for the PN Driver V1.1 (HSP0074) is already on your computer, you can add it to the "Local Support Packages" list by selecting "Add from file system".
  - If the support package for the PN Driver V1.1 (HSP0074) is not yet on your computer, download it with "Download from the Internet" at the "Service & Support" website. Then you can add it to the "Local Support Packages" list from the file system.
3. Select the "HSP\_V13SP1\_0074\_001\_Other\_PNDriver\_1.1.isp13" support package.
4. Click "Install".
5. Follow the instructions on the screen.

Result: The PN Driver is displayed in the hardware catalog under "Communications Modules" – "PROFINET/Ethernet" – "PN Driver" – "6ES7 195-3AA00-0YA0".



Figure 4-2 PN Driver in the hardware catalog

## 4.2 Generating an XML configuration file

### Requirements

The network view of the TIA Portal is open.

### Procedure

1. Create the PN Driver. You have the following options:
  - Drag the PN Driver to the network view from the hardware catalog.
  - Copy and paste the PN Driver into the network view.
  - Double-click the PN Driver entry in the hardware catalog.
2. Select the "PN Driver Station" in the network view and switch to the device view.

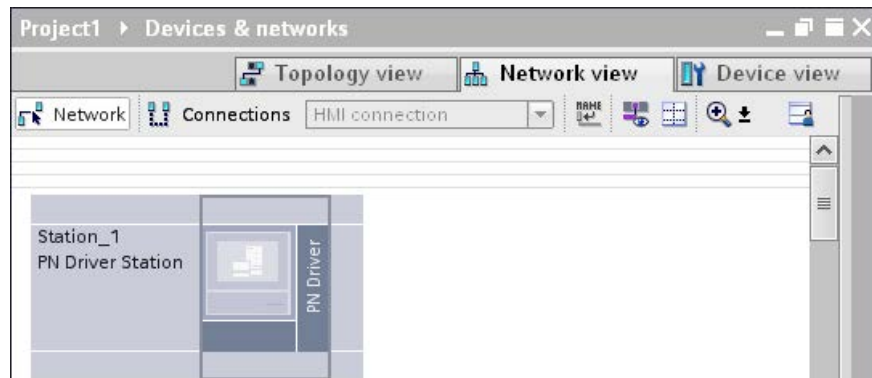


Figure 4-3 PN Driver station in the network view

3. Select the desired interface submodule (for Linux or Windows) in the hardware catalog under "Communications Modules" – "PROFINET/Ethernet" – "Interface Submodules".

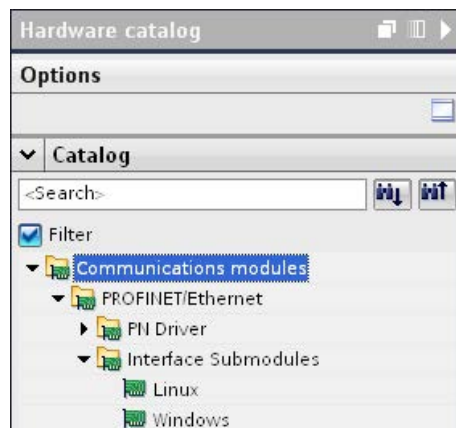


Figure 4-4 Interface submodule in the hardware catalog

4. Perform your hardware configuration as usual.

5. Compile the entire project.  
A consistency check is run when the configuration is compiled. Inconsistencies are displayed as messages in the Inspector window under "Info".
6. When your configuration is consistent, the TIA Portal automatically generates the XML configuration file for the PN Driver.  
The storage path of the XML configuration file and other messages are displayed in the Inspector window under "Info".



Figure 4-5 Storage path of the XML configuration file in the Inspector window

7. Double-click the message "Configuration XML file has been generated under..." to open the Windows Explorer.
8. Integrate the generated XML configuration file in your system.

## Hardware configuration of the application examples

An XML file which includes the hardware configuration is stored for each application example in the respective folder.

### Multiple use IO systems and PNIO diagnostics

The application examples on the CD for **multiple use IO systems** (in the folder "`src\examples\multiple_use_io_systems`") as well as for **PNIO Diagnostics** (in the folder "`src\examples\pnio_ctrl_diag`") all use one shared hardware configuration as shown in the figure below.

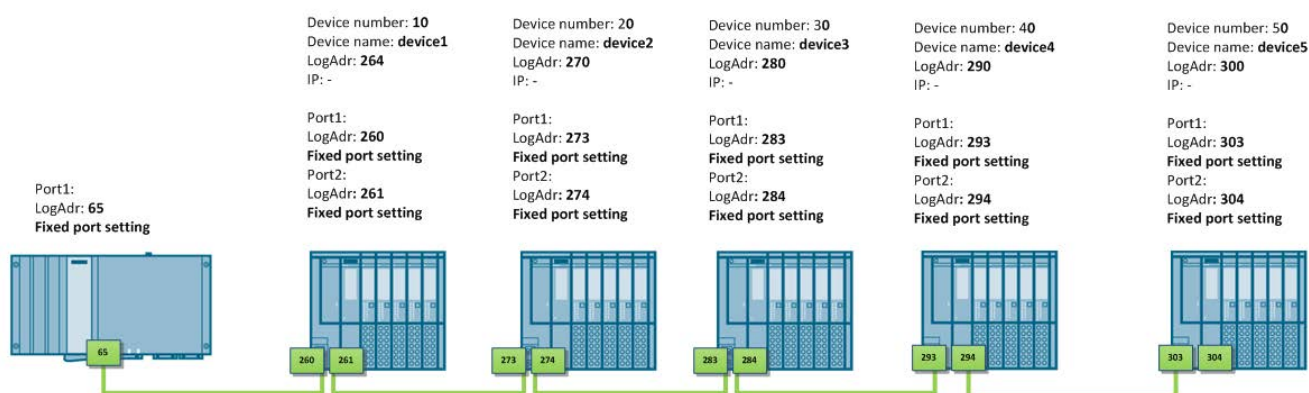


Figure 5-1 Hardware configuration of multiple use IO systems and PNIO diagnostics

The table below lists the included hardware components.

Table 5- 1 Hardware components of the application examples

Component	Number	Article number	Firmware
PNDriver	1		V1.1
SIMATIC ET 200SP, PROFINET interface module IM 155-6 PN ST	5	6ES7155-6AA00-0BN0	V3.1
Light-colored BaseUnit (BU...D)	5	6ES7193-6BP20-0DA0	
Dark-colored BaseUnit (BU...B)	5	6ES7193-6BP20-0BA0	
DI 8x24VDC ST	5	6ES7131-6BF00-0BA0	
DQ 8x24VDC/0.5A ST	5	6ES7132-6BF00-0BA0	
Server module	5	6ES7193-6PA00-0AA0	

## Configuration control for IO systems

The application examples for **configuration control for IO systems** (optional IO devices, flexible topology; in the folder "`\\src\\examples\\configuration_control_of_io_systems`") use the same hardware components; however, you must work with a changed hardware configuration in this case because the application cases use different settings.

The CD includes one application example for optional IO devices and one application example for flexible topology. Each application example has its own hardware configuration and therefore a separate XML file.

### Optional IO devices

The application example for **optional IO devices** uses the hardware configuration shown in the figure below.

Note that the IO device with the device name "device1" is not physically included in the hardware configuration. The created application example is to inform the PN Driver with "PNIO\_reconfig\_iosystem" about this state to commission the plant.

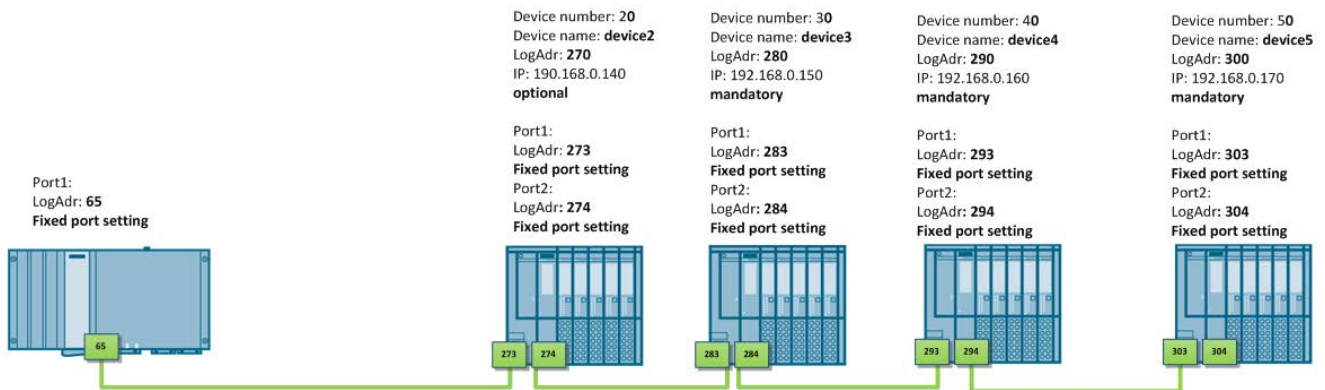


Figure 5-2 Physical hardware configuration of optional IO devices

## Flexible topology

The application example for **flexible topology** is based on the physical hardware configuration shown in the figure below.

Note that the order of the IO devices has changed in the physical configuration. The created application example is to inform the PN Driver about this state to commission the plant.

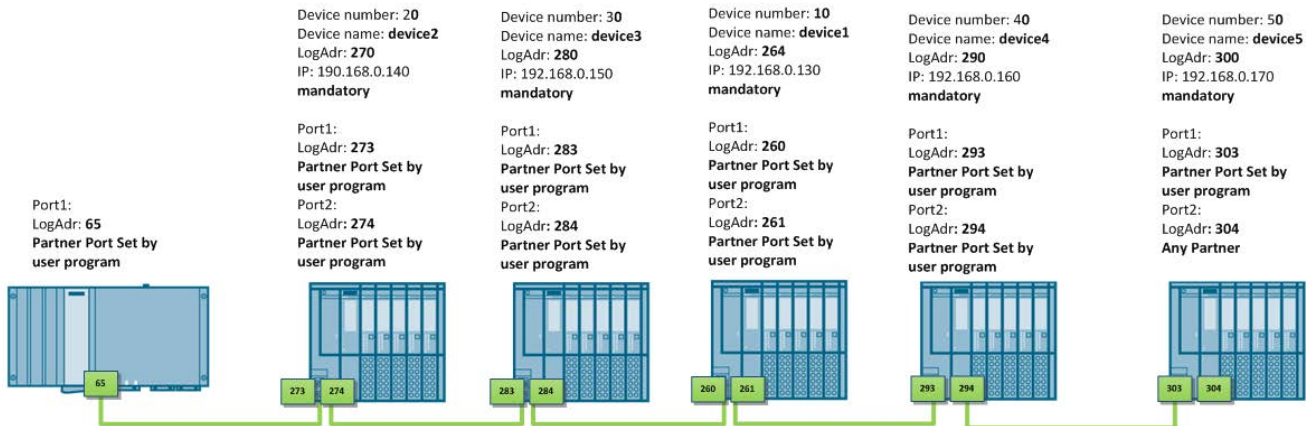


Figure 5-3 Physical hardware configuration of flexible topology

## Configuration control for IO devices

The application example for **Configuration control for IO devices** (in the folder "src\examples\option\_handling") uses a different hardware configuration. The required components are listed in the table below.

Table 5- 2 Hardware components of the configuration control for IO devices

Component	Description	Number	Article number	Firmware
SIMATIC ET 200SP, PROFINET interface module IM 155-6 PN ST		1	6ES7155-6AA00-0BN0	V3.1
Light-colored BaseUnit (BU...D)		2	6ES7193-6BP20-0DA0	
Dark-colored BaseUnit (BU...B)		3	6ES7193-6BP20-0BA0	
DI 8x24VDC ST	M1	1	6ES7131-6BF00-0AB0	
DI 8x24VDC ST	M2	1	6ES7131-6BF00-0AB0	
DQ 4x24VDC/2A ST	M3	1	6ES7132-6BD20-0BA0	
DQ 8x24VDC/0.5A ST	M4	1	6ES7132-6BF00-0BA0	
BU cover	BC	1	6ES7133-6CV15-1AM0	
Server module	SM	1	6ES7 193-6PA00-0AA0	



## References

- Additional information on the use of the configuration control can be found on the Internet (<https://support.industry.siemens.com/cs/ww/en/view/29430270>).
- Additional information on multiple use IO systems and on configuration control for IO systems is available in the function manual PROFINET with STEP 7 V13 (<https://support.industry.siemens.com/cs/ww/en/view/49948856>).

## Opening archived TIA projects in the TIA Portal

1. To open an archived TIA project, select the menu **Project > Retrieve**.

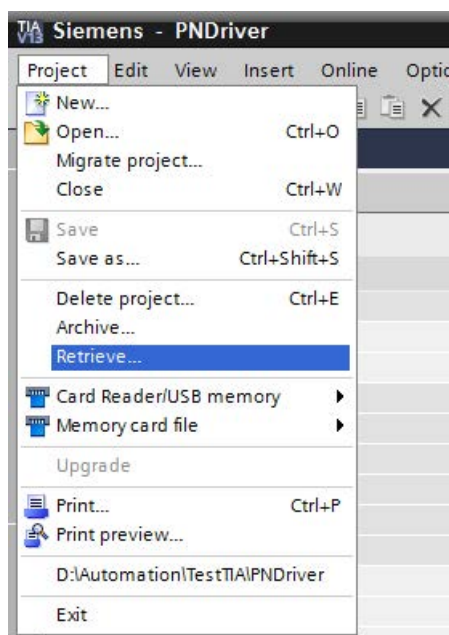


Figure 5-4 TIA Portal - Retrieve

2. In the next window, you first select the archive file of the required project followed by the target directory to which the project should be extracted.