WinAC

driver for PCIO board

User documentation

V1.2 • September 2010

Applikationen & Tools

Answers for industry.

SIEMENS

Industry Automation and Drives Technologies Service & Support Portal

This article is taken from the Service Portal of Siemens AG, Industry Automation and Drives Technologies. The following link takes you directly to the download page of this document.

http://support.automation.siemens.com/WW/view/en/48354988

If you have any questions concerning this document please e-mail us to the following address:

online-support.automation@siemens.com

SIEMENS

SIMATIC
WinAC driver for PCIO board

Warranty and Liability

Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These application examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice.
If there are any deviations between the recommendations provided in these application examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of Siemens Industry Sector.

Table of Contents

Instruction

Content

This document describes the WinAC driver for the PCIO **PC IO Base 400** (6ES7648-2CE20-0AA0) board for the Microbox PC427B. The driver supports the two modules DIO – Digital In/Out **PC IO MOD Digital 010** (6ES7648-2CE40-0BA0) and AIO – Analog In/Out **PC IO MOD Analog 020** (6ES7648-2CE40-0CA0).

1 Basic information

1.1 Description of the problem

I IA SE has developed a central periphery board **PCIO** for SIMATIC Microbox PC427B.

The base board of the PCIO (including four encoder/counter channels) can handle four extension boards (DIO – digital in/out / AIO – analogue in/out)

This document describes the PCIO driver for WinAC RTX 2008. Thus it is possible to use the PCIO functionality (base board, DIO, AIO) within a PLC program of the WinAC.

1.2 Needed Knowledge

To understand this document the knowledge of the following information is needed:

Table 1-1 Documents needed for understanding

Document
Central PC IO expansion – operating manual edition 02/2007
Windows Automation Center RTX – WinAC RTX 2008 Manual

1.3 Reference system

- SIMATIC Microbox PC 427B (1 GHz, 512 MB RAM, 1 GB Flash) with Windows XP embedded SP2
- one PCIO Board installed with 1xDIO and 1xAIO module
  - PC IO Base 400 (6ES7648-2CE20-0AA0)
  - PC IO MOD Digital 010 (6ES7648-2CE40-0BA0)
  - PC IO MOD Analog 020 (6ES7648-2CE40-0CA0)
    FPGA Firmware Version: 13_h
    Microcontroller Firmware Version: 12_h
- WinAC RTX 2009
- STEP 7 V5.4 + SP5

2 Overview

2.1 Functional range

The following functions of the PCIO are supported by the WinAC driver:

- Read digital input (base board and DIO including interrupts)
- Write digital output of DIO
- Read analogue input of AIO (incl. PT100)
- Write analogue output of AIO
- Read encoder (incl. interrupt)
- Read counter (incl. interrupt)

Theoretically you can install up to three PCIO base boards (the third one only without any module). But for the Microbox PC427B there are only two PCI lines with exclusive interrupt (see Siemens support article http://support.automation.siemens.com/WW/view/en/12981782)

2.2 Version of the PCIO driver

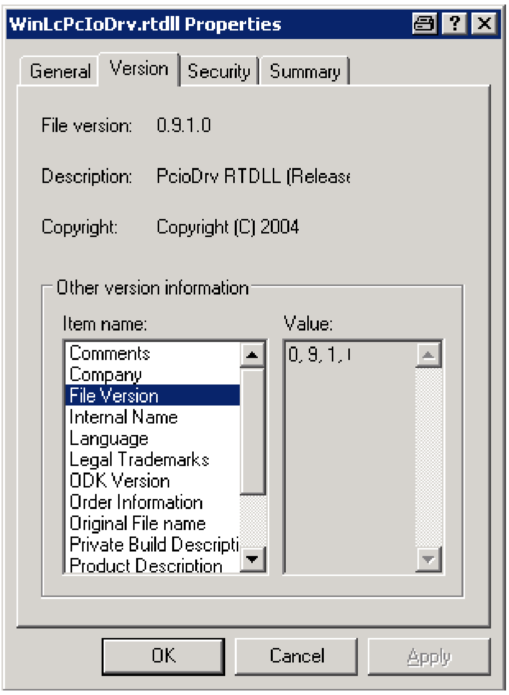
Check driver version with Windows operating system

The registered driver RTDLL is located in the system directory, e.g.

C:\Windows\Rtss\Rtdll

You can identify the version of the driver RTDLL in the file properties (Windows explorer → right click → properties)

Figure 2-1 Version of the driver RTDLL



File version: 0.9.1.0

Description: PcioDrv RTDLL (Releas

Copyright: Copyright (C) 2004

Other version information

Item name:
Comments
Company
File Version
Internal Name
Language
Legal Trademarks
ODK Version
Order Information
Original File name
Private Build Descripti
Product Description

Value:
0, 9, 1, ı

Check driver version in STEP 7 program

In the instance data block of PCIO_INIT it is possible to read the version of the driver RTDLL and the version of the STEP 7 driver function blocks.

C_IF.S7_VERSION	Version of STEP 7 driver function blocks
C_IF.DLL_VERSION	Version of driver RTDLL

3 Installation

3.1 Quickstart

- Check DIP-swithces on PCIO board for correct interrupt lane
- Install PCIO and modules
- Change PCIO from windows device to RTX device
- Check if PCIO owns exclusive interrupt
- Register the driver RTDLL (WinLcPcIoDrv.rtdll) with **setup.bat** (rtssrun /dll WinLcPcIoDrv.rtdll)
- Adapt the STEP 7 example project and use it
- Check Firmware Version of PCIO FPGA and Microcontroller (Instance-DB of PCIO_INIT)

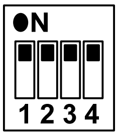
3.2 Installation hardware PCIO in Microbox PC427B

Attention	**Before installation of PCIO you have to check the DIP switches fot the interrupt lane!**

When installing the hardware it is important to get an **exclusive interrupt** for the PCIO. A shared interrupt with another windows device is not supported.

For the installation of the PCIO to the Microbox PC427B you can use the following setting of the DIP-switches for the interrupt lane:

Figure 3-1 Default setting of interrupt DIP on PCIO



Attention	**You have to connect the power supply of the DIO modules (digital in/out) for identification by the PCIO base board.**

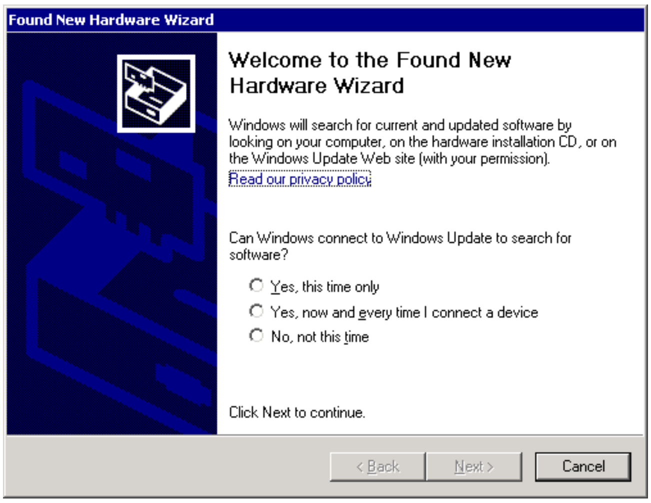
3.3 Install PCIO as RTX device

The PCIO driver is realized as realtime driver for the Windows realtime extension **Ardence RTX** (Realtime eXTension). That's why the PCIO board has to be installed as RTX device.

Cancel Windows Plug and Play manager

Windows Plug and Play Manager recognized the new hardware and tries to install a new driver. This dialogue has to be **canceled.**

Figure 3-2 Windows Plug and Play Manager



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

Read our privacy policy

Can Windows connect to Windows Update to search for software?

○ Yes, this time only

○ Yes, now and every time I connect a device

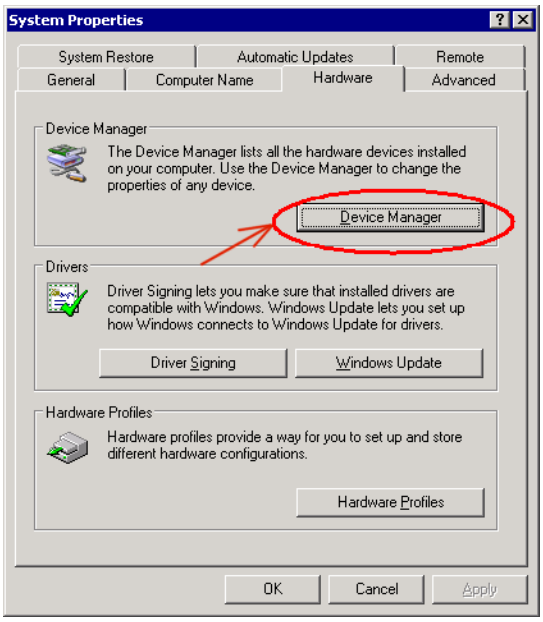
○ No, not this time

Click Next to continue.

Find PCIO in the device manager

The device manager can be started over the system properties.

Figure 3-3 start device manager



System Properties

Device Manager

The Device Manager lists all the hardware devices installed on your computer. Use the Device Manager to change the properties of any device.

Drivers

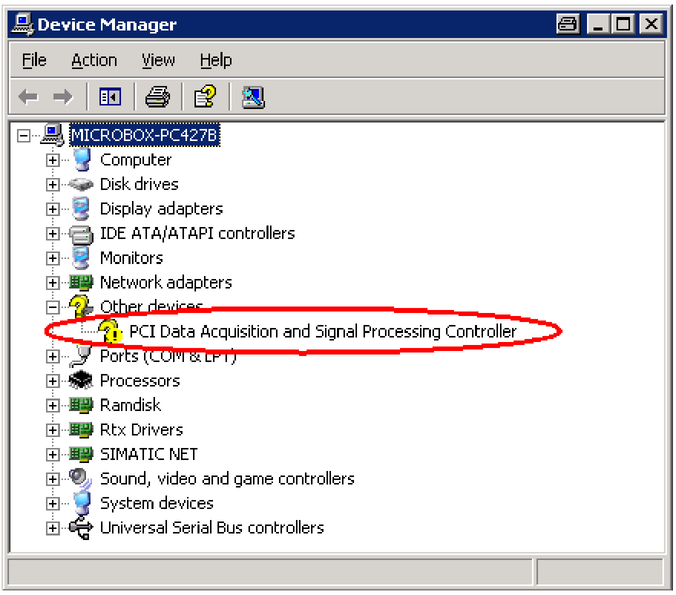
Driver Signing lets you make sure that installed drivers are compatible with Windows. Windows Update lets you set up how Windows connects to Windows Update for drivers.

Hardware Profiles

Hardware profiles provide a way for you to set up and store different hardware configurations.

It should exist only one device with a question mark: „ PCI Data Acquisition and Signal Processing Controller“. This is the PCIO board.

Figure 3-4 unknown device „ PCI Data Acquisition and Signal Processing Controller “

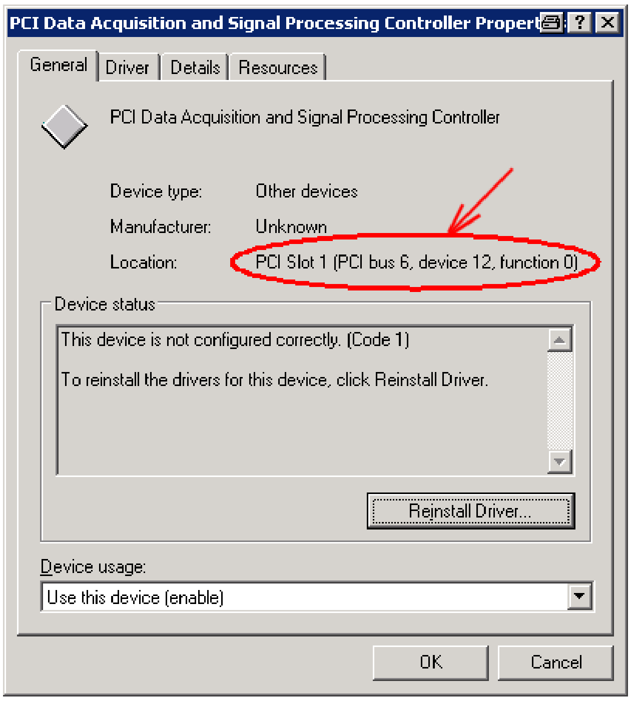


MICROBOX-PC427B
- Computer
- Disk drives
- Display adapters
- IDE ATA/ATAPI controllers
- Monitors
- Network adapters
- Other devices
  - PCI Data Acquisition and Signal Processing Controller
- Ports (COM & LPT)
- Processors
- Ramdisk
- Rtx Drivers
- SIMATIC NET
- Sound, video and game controllers
- System devices
- Universal Serial Bus controllers

Notice the important properties

On the properties you should watch at the slot, bus, device and function.

Figure 3-5 Notice the device properties


PCI Data Acquisition and Signal Processing Controller Propert
General
Driver
Details
Resources
Reinstall Driver...
OK
Cancel

PCI Data Acquisition and Signal Processing Controller

Device type: Other devices

Manufacturer: Unknown

Location: PCI Slot 1 (PCI bus 6, device 12, function 0)

Device status

This device is not configured correctly. (Code 1)

To reinstall the drivers for this device, click Reinstall Driver.

Device usage:

Use this device (enable)

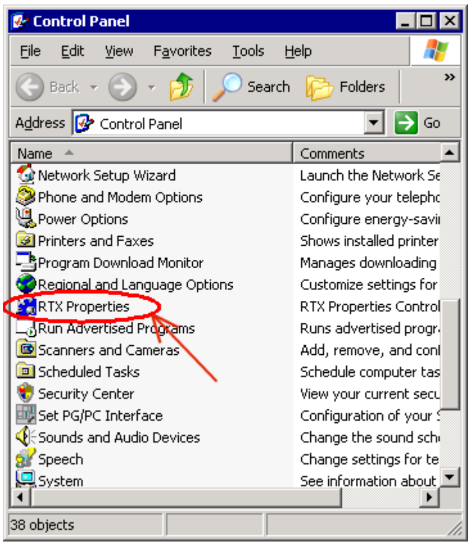
Attention **For the Microbox PC427B only PCI Slot 1 and 2 are valid!**

There is no exclusive interrupt line for PCI Slot 3. Slot 4 is not allowed.

Settings in RTX Properties

Via control panel of the Windows operating system you can reach the RTX Properties.

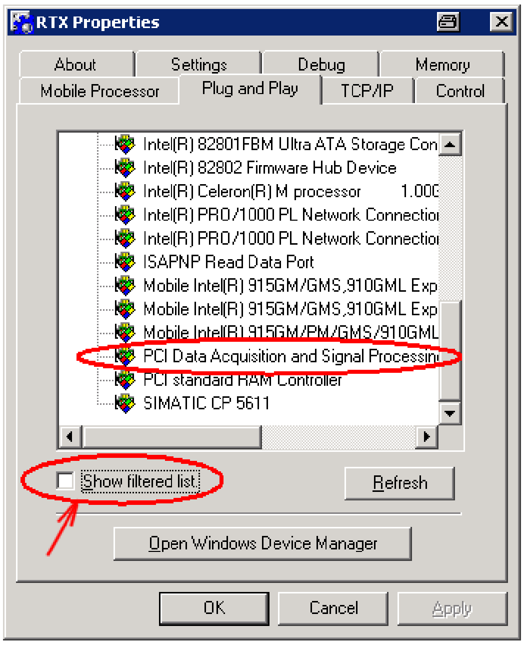
Figure 3-6 RTX Properties in control panel



Name	Comments
Network Setup Wizard	Launch the Network Se
Phone and Modem Options	Configure your telepho
Power Options	Configure energy-savi
Printers and Faxes	Shows installed printer
Program Download Monitor	Manages downloading
Regional and Language Options	Customize settings for
RTX Properties	RTX Properties Control
Run Advertised Programs	Runs advertised progr.
Scanners and Cameras	Add, remove, and conf
Scheduled Tasks	Schedule computer tas
Security Center	View your current secu
Set PG/PC Interface	Configuration of your !
Sounds and Audio Devices	Change the sound sch
Speech	Change settings for te
System	See information about

Select the tab “Plug and Play”. You have to remove the check mark **show filtered list**. Then you can see the PCIO (PCI Data Acquisition and Signal Processing Controller).

Figure 3-7 Plug and Play of the RTX Properties

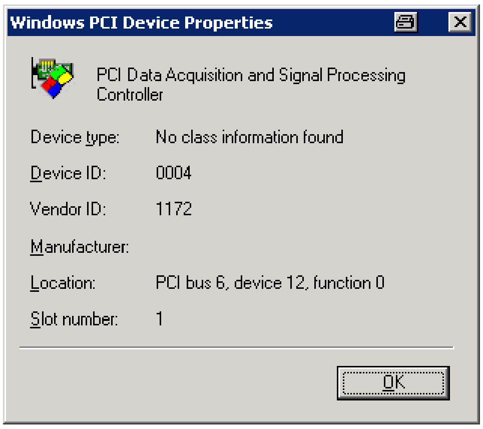


- Intel(R) 82801FBM Ultra ATA Storage Con
- Intel(R) 82802 Firmware Hub Device
- Intel(R) Celeron(R) M processor 1.00G
- Intel(R) PRO/1000 PL Network Connectio
- Intel(R) PRO/1000 PL Network Connectio
- ISAPNP Read Data Port
- Mobile Intel(R) 915GM/GMS,910GML Exp
- Mobile Intel(R) 915GM/GMS,910GML Exp
- Mobile Intel(R) 915GM/PM/GMS/910GML
- PCI Data Acquisition and Signal Processin
- PCI standard RAM Controller
- SIMATIC CP 5611

□ Show filtered list

Via the device properties you can check whether it's the right device (slot, bus, ...).

Figure 3-8 Device Properties in dialogue RTX Properties


Windows PCI Device Properties

PCI Data Acquisition and Signal Processing Controller

Device type: No class information found

Device ID: 0004

Vendor ID: 1172

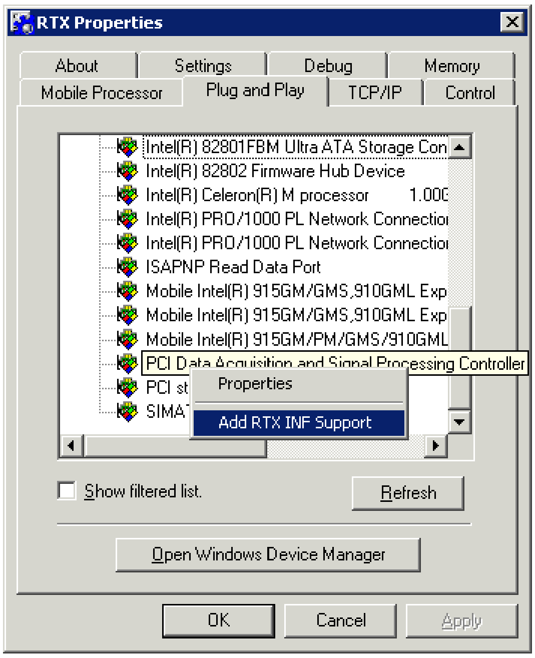
Manufacturer:

Location: PCI bus 6, device 12, function 0

Slot number: 1

Next you choose the device with the right mouse bottom and click on
Add RTX INF support.

Abbildung 3-9 Adding RTX INF support to the PCIO

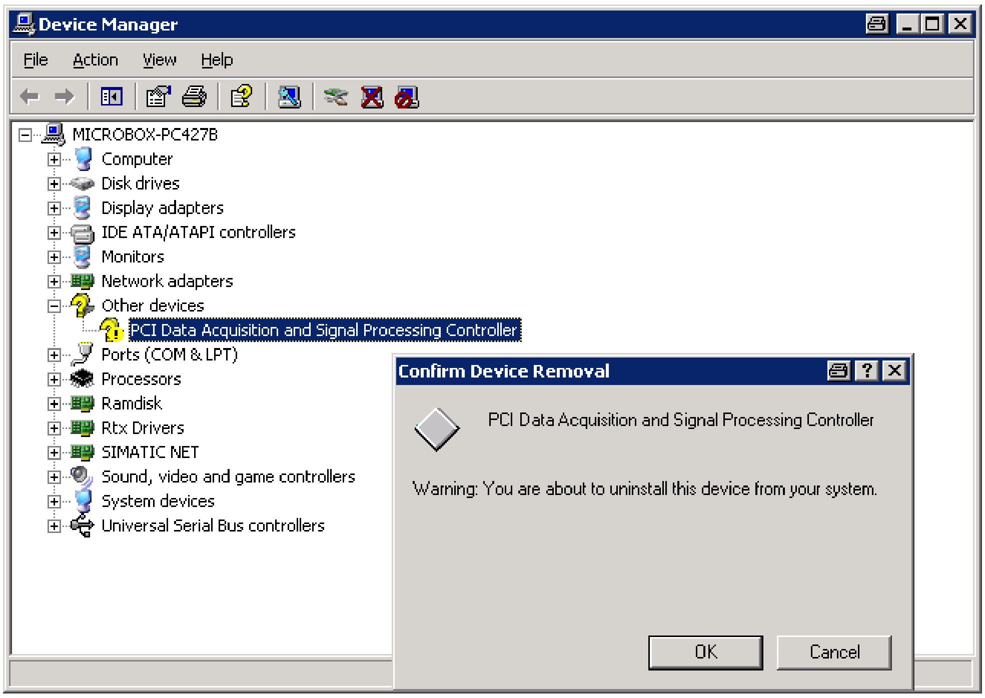

RTX Properties

Intel(R) 82801FBM Ultra ATA Storage Con
Intel(R) 82802 Firmware Hub Device
Intel(R) Celeron(R) M processor 1.00
Intel(R) PRO/1000 PL Network Connectio
Intel(R) PRO/1000 PL Network Connectio
ISAPNP Read Data Port
Mobile Intel(R) 915GM/GMS,910GML Exp
Mobile Intel(R) 915GM/GMS,910GML Exp
Mobile Intel(R) 915GM/PM/GMS/910GML
PCI Data Acquisition and Signal Processing Controller
PCI st
SIMA

Uninstall the device in the Windows device manager

Next you go again to the **Windows device manager** in order to uninstall the PCIO („PCI Data Acquisition and Signal Processing Controller“ with question mark).

Figure 3-10 (Windows) device Uninstallation



PCI Data Acquisition and Signal Processing Controller

Warning: You are about to uninstall this device from your system.

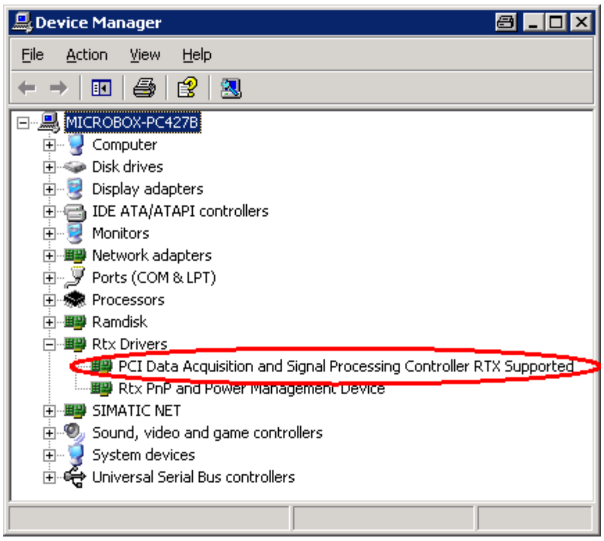
The device disappears firstly from the device manager.

Installation of device as RTX device

Afterwards you choose at the menu **Action → Scan for new Hardware** (This works only, if you have clicked into the main window of the device manager before).

Now the PCIO will be registered as RTX device automatically.

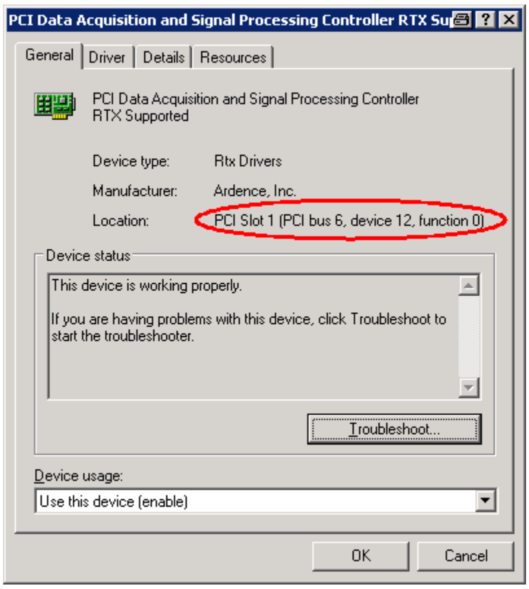
Figure 3-11 PCIO as RTX device



- MICROBOX-PC427B
  - Computer
  - Disk drives
  - Display adapters
  - IDE ATA/ATAPI controllers
  - Monitors
  - Network adapters
  - Ports (COM & LPT)
  - Processors
  - Ramdisk
  - Rtx Drivers
    - PCI Data Acquisition and Signal Processing Controller RTX Supported
    - Rtx PnP and Power Management Device
  - SIMATIC NET
  - Sound, video and game controllers
  - System devices
  - Universal Serial Bus controllers

To be safe you should check with the device properties whether it's the right device.

Figure 3-12 Properties as PCI device


PCI Data Acquisition and Signal Processing Controller RTX Su
General
Driver
Details
Resources
Troubleshoot...
OK
Cancel

PCI Data Acquisition and Signal Processing Controller RTX Supported

Device type: Rtx Drivers

Manufacturer: Ardence, Inc.

Location: PCI Slot 1 (PCI bus 6, device 12, function 0)

Device status

This device is working properly.

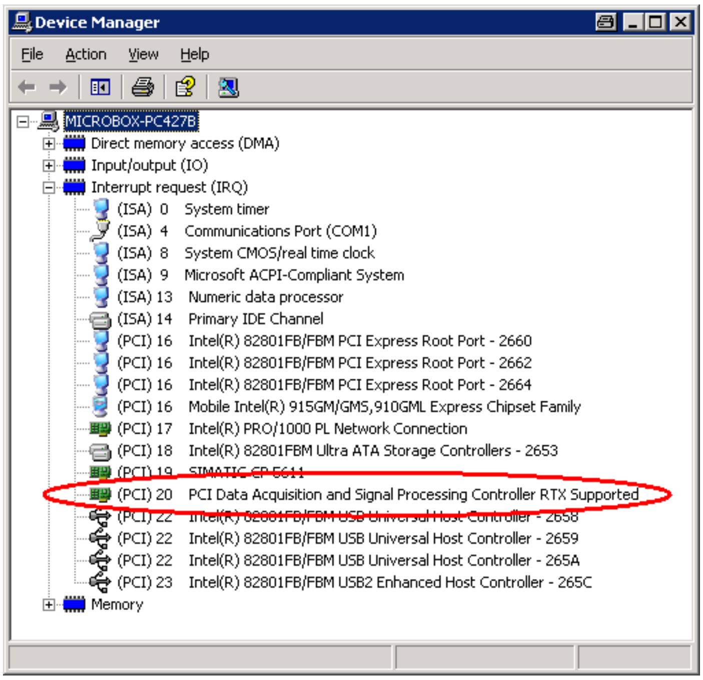
If you are having problems with this device, click Troubleshoot to start the troubleshooter.

Device usage: Use this device (enable)

Check exclusive interrupt

At least you have to check whether the interrupt which is used by the CP1604 is not used by any other device.

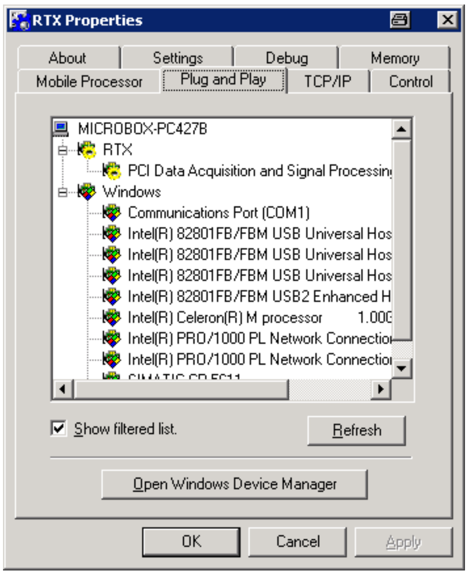
Figure 3-13 Interrupt of PCIO in the Windows device manager



```
MICROBOX-PC427B
  Direct memory access (DMA)
  Input/output (IO)
  Interrupt request (IRQ)
    (ISA)  0   System timer
    (ISA)  4   Communications Port (COM1)
    (ISA)  8   System CMOS/real time clock
    (ISA)  9   Microsoft ACPI-Compliant System
    (ISA) 13   Numeric data processor
    (ISA) 14   Primary IDE Channel
    (PCI) 16   Intel(R) 82801FB/FBM PCI Express Root Port - 2660
    (PCI) 16   Intel(R) 82801FB/FBM PCI Express Root Port - 2662
    (PCI) 16   Intel(R) 82801FB/FBM PCI Express Root Port - 2664
    (PCI) 16   Mobile Intel(R) 915GM/GMS,910GML Express Chipset Family
    (PCI) 17   Intel(R) PRO/1000 PL Network Connection
    (PCI) 18   Intel(R) 82801FBM Ultra ATA Storage Controllers - 2653
    (PCI) 19   SIMATIC CP 5611
    (PCI) 20   PCI Data Acquisition and Signal Processing Controller RTX Supported
    (PCI) 22   Intel(R) 82801FB/FBM USB Universal Host Controller - 2658
    (PCI) 22   Intel(R) 82801FB/FBM USB Universal Host Controller - 2659
    (PCI) 22   Intel(R) 82801FB/FBM USB Universal Host Controller - 265A
    (PCI) 23   Intel(R) 82801FB/FBM USB2 Enhanced Host Controller - 265C
  Memory
```

The interrupt settings can be checked in the **RTX Properties** too.

Figure 3-14 RTX Properties with PCIO as RTX-Device

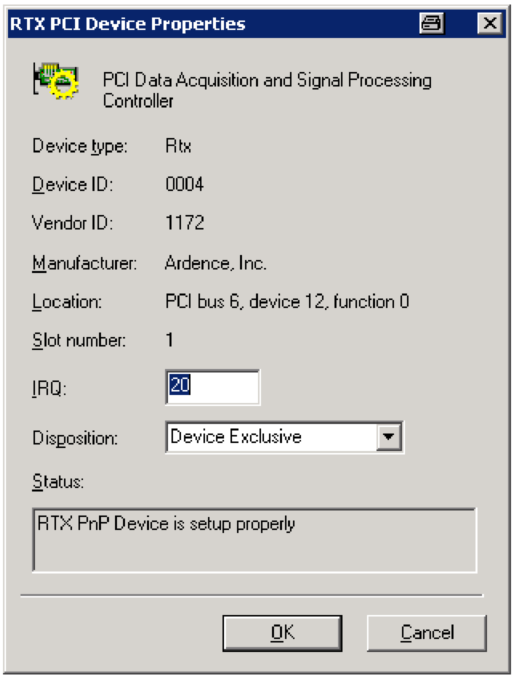


```
MICROBOX-PC427B
  RTX
    PCI Data Acquisition and Signal Processin
  Windows
    Communications Port (COM1)
    Intel(R) 82801FB/FBM USB Universal Hos
    Intel(R) 82801FB/FBM USB Universal Hos
    Intel(R) 82801FB/FBM USB Universal Hos
    Intel(R) 82801FB/FBM USB2 Enhanced H
    Intel(R) Celeron(R) M processor        1.00
    Intel(R) PRO/1000 PL Network Connectio
    Intel(R) PRO/1000 PL Network Connectio
    SIMATIC CP 5611
```

☑ Show filtered list.

Also at this place (right mouse bottom → properties) you can check whether the interrupt is exclusive:

Figure 3-15 Interrupt of PCIO in the RTX Properties



PCI Data Acquisition and Signal Processing Controller

Device type: Rtx

Device ID: 0004

Vendor ID: 1172

Manufacturer: Ardence, Inc.

Location: PCI bus 6, device 12, function 0

Slot number: 1

IRQ: 20

Disposition: Device Exclusive

Status:

RTX PnP Device is setup properly

Note

The check of the exclusive interrupts is only reasonable after this last step because this allocation changes during the installation.

3.3.1 Check PCIO installation with “PcioScan.rtss”

The WinAC PCIO driver includes a tool to check the installed PCIO board including the modules. The tool **PcioScan.rtss** is located in the **\tools** directory. It is a RTX application. On a computer with installed Ardence RTX you can start this application with double click. This is the case on all computers with WinAC RTX.

Figure 3-16 Interrupt of PCIO in the RTX Properties

```
PC IO board <0>
Interrupt 20, Firmware MC 0x12 Firmware FPGA 0x13
  Module 0x01 – DIO
  Module 0x02 – AIO
  Module 0x00 – empty
  Module 0x00 – empty
```

3.4 Installation WinAC driver on runtime system

The installation of the WinAC driver PCIO is limited to the registration of the driver realtime DLL (WinLcPcIoDrv.rtdll) with the batch file **setup.bat** (rtssrun /dll WinLcPcIoDrv.rtdll).

You can check the installation with the command **rtsskill**. At the registered DLLs you must see the WinLcPcioDrv.rtdll.

3.5 Installation WinAC driver on engineering system

On the engineering system only this documentation and the STEP 7 example program is needed. You can copy the needed function blocks and user defined types from this STEP 7 example to you own application.

3.6 Updating firmware PCIO

The PCIO board uses two firmwares: one for the FPGA and the other for microcontroller. Both can be updated by software.

Attention	**In the moment the updating of PCIO firmware is only possible with a bootable floppy disk. Updating with a bootable USB stick does not work!**

Update firmware of FPGA

Use the batch file **fp.bat** from the floppy disk.

Update firmware of microcontroller

Use the batch file **mc.bat** from the floppy disk.

4 Driver supported functionality

The functionality of the PCIO is described in the document “Central PC IO expansion – operating manual”. In this chapter special features of the WinAC driver are explained.

4.1 Interrupts

The PCIO board generates a number of different interrupts. Some interrupts are processed in the driver internally. The most interrupts are transferred to the WinAC (via interrupt OB). The user can configure which interrupts are signalled to the WinAC.

4.2 Digital input / output

One PCIO base board can handle up to four DIO modules (every module 24 DI and 16 DO) plus four digital inputs of the base board.

There are two “very fast” digital inputs on every DIO. These inputs are not accessible separately by the WinAC driver. Of course these “very fast” inputs can be used as inputs for the counter functionality.

The eight interrupt digital inputs of every DIO trigger an interrupt for every changing. If edge detection is needed, it has to be done in the interrupt OB (operation block) in the STEP 7 program.

4.3 Analogue input / output

The conversions are done over a multiplexer. Every “read analogue input” by the WinAC function blocks returns the last converted values of the analogue inputs.

The driver has to process some interrupt handling for analog conversion internally. No processing on STEP 7 side is needed for that.

The PCIO board offers three different modes for digital-analogue and analogue-digital-conversion (block mode, single mode, fast mode). The WinAC driver uses the block mode.

Attention	**For the right function of the analogue inputs/outputs the following firmware version is needed:** **FPGA firmware 13_h** **Microcontroller firmware 12_h**

Operation of analogue output

With every PCIO_WRITE a new block conversion is initiated. With the PCIO_CONFIG the channels for analogue output are selected.

The speed is defined by performance of block conversion of the PCIO:

Picture 4-1 Computing conversion time analogue output

$$T_{\text{ConversionOut}} = \textit{<QuantityChannels>} \text{ x } 100\ \mu s + 100\ \mu s\ ^{*1)} + (0..50\ \mu s)\ ^{*2)}$$

[*1)] This ist he time for initiating the DA conversion.

[*2)] This Jitter based on the clocked processing of the PCIO of 50µs.

Thus even with eight channels you need maximum 950µs conversion time. This is enough for a WinAC cycle time of 1 ms.

Operation of analogue input

When minimum one analogue input channel is selected by PCIO_CONFIG, the block conversion starts. Internally the driver restarts the block conversion continuously. Thus the WinAC gets always updated analogue input values for the selected channels.

The speed is defined by performance of block conversion of the PCIO:

Picture 4-2 Computing conversion time analogue input

$$T_{\text{ConvIn}} = \textit{<CountChannels>} \text{ x } 50\ \mu s + 50\ \mu s\ ^{*1)} + (0..50\ \mu s)\ ^{*2)} + (100\ \mu s)\ ^{*3)}$$

[*1)] This ist he time for initiating the AD conversion.

[*2)] This Jitter based on the clocked processing of the PCIO of 50µs.

[*3)] Only if the PT100 inputs are activated.

Thus even with eight channels you need maximum 500µs conversion time. If using 1 ms cycle time for WinAC you will get new analogue input values for every WinAC cycle.

Operation of PT100

Every AIO module offers four PT100 input channels. Additional the reference values for 100 Ω and 200 Ω are determined. The WinAC PCIO driver provides the raw values. Furthermore the temperature values are provided according to the following formula:

$$T = \frac{266°C \cdot \left(PT100 - ref_{100\Omega}\right)}{ref_{200\Omega} - ref_{100\Omega}}$$

This value is given in [0,1 °C].

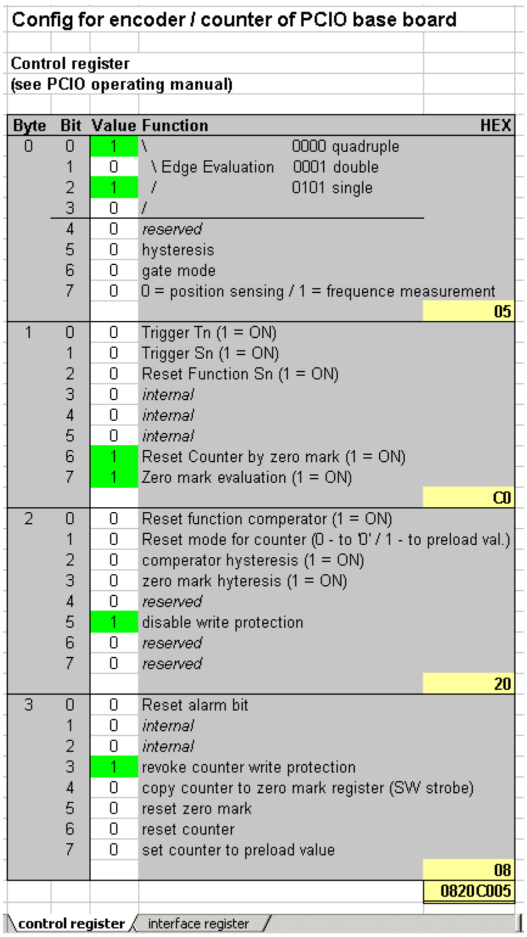
4.4 Counter / Encoder

The counter/encoder functionality of the PCIO is very powerful. The parameterisation is done according to the operating manual of the PCIO by writing the configuration registers. These values have to be set within the configuration function block in the WinAC program. That means for configuring the counter/encoder functionality the PCIO operating manual is the reference.

All interrupts sources of the counter/encoder are signalled to the WinAC if needed.

To do the configuration of the counter/encoder in an easy way an Excel tool **PcioEncoderConfig.xls** is part of the WinAC PCIO driver (directory **tools**\).

Figure 4-3 Configuration of counter/encoder with Excel tool



Config for encoder / counter of PCIO base board

Control register
(see PCIO operating manual)

Byte	Bit	Value	Function	HEX
0	0	1	\ 0000 quadruple	
	1	0	\ Edge Evaluation 0001 double	
	2	1	/ 0101 single	
	3	0	/	
	4	0	*reserved*	
	5	0	hysteresis	
	6	0	gate mode	
	7	0	0 = position sensing / 1 = frequence measurement	
				05
1	0	0	Trigger Tn (1 = ON)	
	1	0	Trigger Sn (1 = ON)	
	2	0	Reset Function Sn (1 = ON)	
	3	0	*internal*	
	4	0	*internal*	
	5	0	*internal*	
	6	1	Reset Counter by zero mark (1 = ON)	
	7	1	Zero mark evaluation (1 = ON)	
				C0
2	0	0	Reset function comperator (1 = ON)	
	1	0	Reset mode for counter (0 - to '0' / 1 - to preload val.)	
	2	0	comperator hysteresis (1 = ON)	
	3	0	zero mark hyteresis (1 = ON)	
	4	0	*reserved*	
	5	1	disable write protection	
	6	0	*reserved*	
	7	0	*reserved*	
				20
3	0	0	Reset alarm bit	
	1	0	*internal*	
	2	0	*internal*	
	3	1	revoke counter write protection	
	4	0	copy counter to zero mark register (SW strobe)	
	5	0	reset zero mark	
	6	0	reset counter	
	7	0	set counter to preload value	
				08
				0820C005

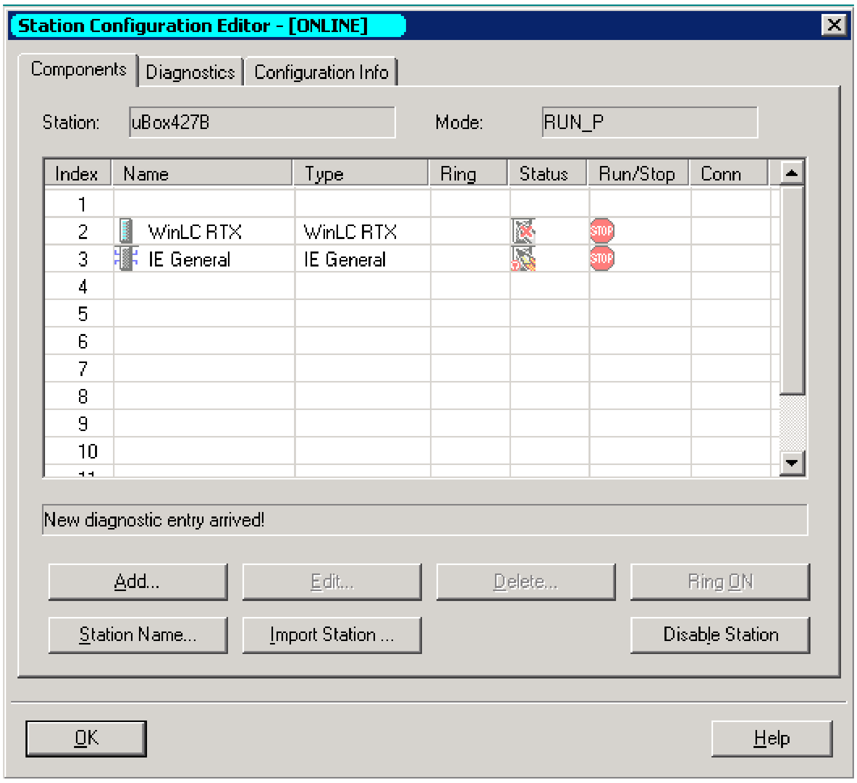
5 SIMATIC projecting for PCIO

5.1 Component Configurator on runtime system

The PCIO board is accessed by the driver directly. That’s why a configuration in the Component Configurator on the runtime system is **not** needed.

If the runtime system is projected over Ethernet, the Component Configurator looks like this:

Figure 5-1 Component Configurator (projecting runtime over Ethernet)



Station: uBox427B

Mode: RUN_P

Index	Name	Type	Ring	Status	Run/Stop	Conn
1						
2	WinLC RTX	WinLC RTX			STOP	
3	IE General	IE General			STOP	
4						
5						
6						
7						
8						
9						
10						

New diagnostic entry arrived!

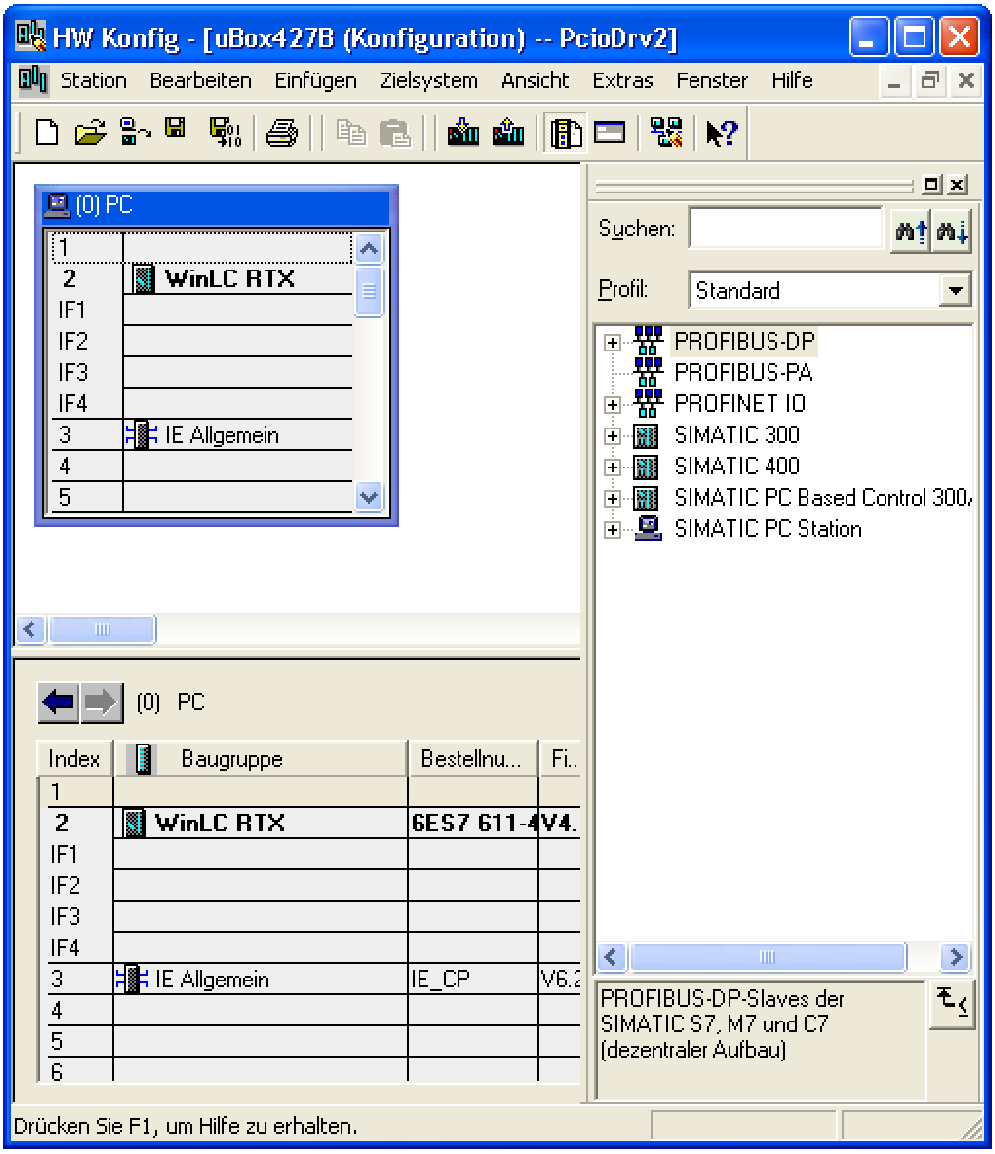
The PCIO board is not part of the Component Configurator on the runtime system.

5.2 SIMATIC Manager HW Config

Because of direct hardware access of the driver there is no entry in the hardware configuration of the SIMATIC Manager for the PCIO card needed.

If the runtime system is projected over Ethernet, the hardware configuration looks like this:

Figure 5-2 HW Config for the WinAC project


HW Konfig - [uBox427B (Konfiguration) -- PcioDrv2]
Station
Bearbeiten
Einfügen
Zielsystem
Ansicht
Extras
Fenster
Hilfe
(0) PC
Suchen:
Profil:
Standard
PROFIBUS-DP
PROFIBUS-PA
PROFINET IO
SIMATIC 300
SIMATIC 400
SIMATIC PC Based Control 300/
SIMATIC PC Station
PROFIBUS-DP-Slaves der SIMATIC S7, M7 und C7 (dezentraler Aufbau)
Drücken Sie F1, um Hilfe zu erhalten.

1	
2	WinLC RTX
IF1	
IF2	
IF3	
IF4	
3	IE Allgemein
4	
5	

(0) PC

Index	Baugruppe	Bestellnu...	Fi..
1			
2	WinLC RTX	6ES7 611-4	V4.
IF1			
IF2			
IF3			
IF4			
3	IE Allgemein	IE_CP	V6.2
4			
5			
6			

6 The STEP 7 user interface

To use the PCIO functionality from the WinAC program there are some function blocks and user defined types in the example project. Of course you can change the numbers of the FBs, DBs and UDTs.

FB167 – PCIO_INIT

FB168 – PCIO_CONFIG

FB169 – PCIO_READ

FB170 – PCIO_WRITE

6.1 Multi instance FBs

The driver FBs are not multi instancable!

Explanation:

The WinAC driver is realised with the WinAC ODK (Open Develeopment Kit). All driver FBs need the reference to the driver RTDLL (ODK handle). The Init-FB distributes this ODK handle to all the instance DBs of the driver FBs and does some initialisation within the instance DBs.

6.2 Initialisation PCIO_INIT

The initialisation function block **PCIO_INIT** has to be called before any other driver FB call.

This FB loads the driver RTDLL. It distributes the information about the ODK handle to the instanced DBs of all driver FBs. The FB reads the information about the installed PCIO boards (obtained by a PCI scan). This data is stored in the instance DB of this FB.

The RTX driver checks the version of the STEP 7 FBs. Only for a matching version the FB call is done without 0.

Table 6-1 Parameters of the FB PCIO_INIT

Parameter	In/Out	Type	Description
DBI_PCIO_CONFIG	In	Block_DB	Instance DB von PCIO_CONFIG
DBI_PCIO_READ	In	Block_DB	Instance DB von PCIO_READ
DBI_PCIO_WRITE	In	Block_DB	Instance DB von PCIO_WRITE
ERROR	Out	BOOL	Error occurred (if 1, the STATUS gives more detailed information about the problem)
STATUS	Out	WORD	Error code

6.2.1 Additional information in the instance DB of PCIO_INIT

The user can obtain additional information in the instance DB of the PCIO_INIT function block:

Table 6-2 Information in the instance -DB of PCIO_INIT

Name	In/Out	Description
C_IF.S7_VERSION	Out	Version of the STEP 7 FBs of the driver
C_IF.DLL_VERSION	In	Version of the driver RTDLL
C_IF.PCIO_INIT_RES	In	State of the initialisation of the driver (e.g. PCI scan)
C_IF.PCIO_HW[0..2]	Out	HW-Info about the recognised PCIO boards (max. 3)

Coding of the RTDLL version

The version of the RTDLL is coded hexadecimal. The last digit of the DWORD is used to label the Debug or Release version.

D – Debug-Version

A – Release-Version

Figure 6-1 Examples for RTDLL version in instance DB of PCIO_INIT

```
"DBI_PCIO_INIT".C_IF.DLL_VERSION   HEX   DW#16#0001000D
                                                  \  /|
                                                   \/ +- Debug
                                                   +---- V 1.0.0.0


"DBI_PCIO_INIT".C_IF.DLL_VERSION   HEX   DW#16#0001100A
                                                  \  /|
                                                   \/ +- Release
                                                   +---- V 1.1.0.0
```

Note The data of the instance DB (e.g. driver version) is valid after an error-free call of PNIO_INIT only!

6.2.2 Check the recognised PCIO configuration

There is the structure **C_IF.PCIO_HW** inside the instance DB of PCIO_INIT: At this structure the information about the recognised PCIO configuration is stored.

Table 6-3 Information about recognised PCIO hardware

Name	Description
PCIO_HW.Interrupt	Interrupt number of this PCIO board
PCIO_HW.VersionFpga	Version of FPGA software
PCIO_HW.VersionFirmware	Version of microcontroller firmware
PCIO_HW.ModuleType0	Module type slot 0
PCIO_HW.ModuleType1	Module type slot 1
PCIO_HW.ModuleType2	Module type slot 2
PCIO_HW.ModuleType3	Module type slot 3

The following module types are supported by the PCIO driver:

Table 6-4 Supported module types

Nummer	Modultyp
00	Empty slot respectively module not recognized
01	DIO – Digital In/Out *1)
02	AIO – Analogue In/Out
Nn	All other values – unknown module type

*1) The DIO module can be recognised only if the 24 V power supply is connected.

6.3 Configuration with PCIO_CONFIG

One call of **PCIO_CONFIG** configures one PCIO board with up to four modules. If using more than one PCIO board this function block has to be called for every PCIO board.

All configuration data is stored in the configuration DB. This DB is a parameter for PCIO_CONFIG:

This function block has to be called before any read / write access of the PCIO board.

There is no separate function block e.g. for changing configuration of encoder functionality. For changing such parameters the function block **PCIO_CONFIG** has to be called again.

Table 6-5 Parameters of FBs PCIO_CONFIG

Parameter	In/Out	Type	Description
PARAM	In	Any	Any-Pointer to configuration DB
ERROR	Out	BOOL	Error occurred (if 1, the STATUS gives more detailed information about the problem)
STATUS	Out	WORD	Error code

6.3.1 Structure of the configuration data for one PCIO board

The configuration DB for **PCIO_CONFIG** includes the complete configuration of one PCIO board including the maximum amount of four extension modules. This configuration data block consists always of fife UDTs:

- UDT_PCIO_CONFIG_BASE
- four UDTs for the modules (DIO / AIO / empty)

The four UDTs must match the hardware configuration of the PCIO board.

Table 6-6 Examples for the configuration DB corresponding to the PCIO hardware

PCIO hardware	Config-DB	
	Name	**Type**
	CFG_BASE	UDT_PCIO_CONFIG_BASE
Slot 0: empty	CFG_SLOT0	UDT_PCIO_CONFIG_EMPTY
Slot 1: empty	CFG_SLOT1	UDT_PCIO_CONFIG_EMPTY
Slot 2: empty	CFG_SLOT2	UDT_PCIO_CONFIG_EMPTY
Slot 3: empty	CFG_SLOT3	UDT_PCIO_CONFIG_EMPTY
	Name	**Type**
	CFG_BASE	UDT_PCIO_CONFIG_BASE
Slot 0: DIO	CFG_SLOT0	UDT_PCIO_CONFIG_DIO
Slot 1: AIO	CFG_SLOT1	UDT_PCIO_CONFIG_AIO
Slot 2: empty	CFG_SLOT2	UDT_PCIO_CONFIG_EMPTY
Slot 3: empty	CFG_SLOT3	UDT_PCIO_CONFIG_EMPTY

If the configuration DB does not match the PCIO hardware, there is an error message like „no DIO module on this slot installed"(see also chapter 8.2 "Special error codes of the PC IO driver" on page 42).

Configuration of base board (including counter/encoder)

The configuration of one base board including counter/encoder is done with the UDT **UDT_PCIO_CONFIG_BASE**.

Table 6-7 Structure of UDT_PCIO_CONFIG_BASE

Name	Type	Description
ParamType	BYTE	Do not change!
Length	INT	Do not change!
BoardNo	INT	PCIO base board number
IRQ_OB_NO	INT	Number of WinAC interrupt OB (52..54)
ENC_IRQ_MAKS	DWORD	interrupt mask for encoder interrupts *1) *2)
ENC_CONFIG	ARRAY [0 .. 3]	
IF_SWITCH_REG	DWORD	encoder interface switching register *1) *3)

Name	Type	Description
PRE_LOAD_REG	DWORD	preload value register [*1)]
CONTROL_REG	DWORD	control register [*1)] [*3)]
UNIVERSAL_REG_1	DWORD	universal register no. 1 [*1)]
USE_UNIVERSAL_REG_1	BOOL	use universal register no. 1 [*1)]

[*1)] See “PCIO operating manual” chapter “main interrupt mask register”

Note

[*2)] A bit value of ‚1’ activates the interrupt.

The bit value ‚0’ deactivates the corresponding interrupt.

[*3)] To do the configuration of the counter/encoder in an easy way an Excel tool **PcioEncoderConfig.xls** is part of the WinAC PCIO driver (directory **tools**\).

If using incremental encoder at all four encoder inputs you can use the following values for the configuration (according the “PC IO operating manual”):

Table 6-8 Configuration of four incremental encoders

Channel	Interface switch register	Control register
0	0x0001 0909	0x0800 8000
1	0x0002 1212	0x0800 8000
2	0x0001 0909	0x0800 8000
3	0x0002 1212	0x0800 8000

In the following example the DI 0 of DIO module on slot 0 should be counted. DI 1 is the gate for this counter, i.e. counter runs only when DI 1 is **high**.

Table 6-9 Configuration of one counter

Channel	Interface switch register	Control register
0	0x0000 001B	0x4020 0080

Configuration of DIO module (Digital In/Out)

The configuration of the DIO module is done by **UDT_PCIO_CONFIG_DIO**.

Table 6-10 Structure of UDT_PCIO_CONFIG_DIO

Name	Type	Description
ParamType	BYTE	Do not change!
Length	INT	Do not change!
DI_IRQ_MASK	BYTE	IRQ-Mask for this DIO (only 8 DI possible IRQ source) [*1)]

[*1)]

Note

A bit value of ‚1’ activates the interrupt.

The bit value ‚0’ deactivates the corresponding interrupt.

Configuration of DIO module (Analogue In/Out)

The configuration of the DIO module is done by **UDT_PCIO_CONFIG_AIO**.

Table 6-11 Structure UDT_PCIO_CONFIG_AIO

Name	Type	Description
ParamType	BYTE	Do not change!
Length	INT	Do not change!
PT100_REQ	BYTE	requested PT100 channels (bit coded) *1)
AI_CHANNEL_SELECT	BYTE	Enable channels for input (bit coded)
AI_RANGES : ARRAY	ARRAY [0 .. 7] OF WORD	input conversion range (+/- 0/5/10V) for input channels *2)
AO_CHANNEL_SELECT	BYTE	Enable channels for output (bit coded)

*1) If the PT100 inputs are activated, it affects the conversion time of the other analogue inputs of this AIO module (see “PCIO operating manual”).

Note If using the PT100 channels also the two reference values have to be activated.

*2) The select the conversion range use the following values:

Tabelle 6-12 Coding of AD range

AD range	code
0 V ... + 5 V	0x00
- 5 V ... + 5 V	0x04
0 V ... +10 V	0x08
-10 V ... +10 V	0x0C

6.4 Read inputs with PCIO_READ

The function block **PCIO_READ** returns the actual inputs of one PCIO board including all modules.

Table 6-13 Parameters of the FB PCIO_READ

Parameter	In/Out	Type	Description
DATA	In	Any	Any pointer to DB with READ data
ERROR	Out	BOOL	Error occurred (if 1, the STATUS gives more detailed information about the problem)
STATUS	Out	WORD	Error code

Attention **The parameter ‚DATA’ may only be connected with a data block (DB) – see example project. A general ANY pointer is not allowed.**

6.4.1 Data structure for reading inputs

To use the same function block for reading the inputs, independent from the PCIO hardware configuration, the read data block always consists of fife UDTs:

- UDT_PCIO_READ_BASE
- four UDTs for the modules (DIO / AIO / empty)

The four UDTs must match the hardware configuration of the PCIO board.

Table 6-14 Examples for structure of read DB according the PCIO hardware

PCIO hardware	Read data DB
  Slot 0: empty Slot 1: empty Slot 2: empty Slot 3: empty	**Name** **Type** CFG_BASE UDT_PCIO_READ_BASE CFG_SLOT0 UDT_PCIO_READ_EMPTY CFG_SLOT1 UDT_PCIO_READ_EMPTY CFG_SLOT2 UDT_PCIO_READ_EMPTY CFG_SLOT3 UDT_PCIO_READ_EMPTY
  Slot 0: DIO Slot 1: AIO Slot 2: empty Slot 3: empty	**Name** **Type** CFG_BASE UDT_PCIO_READ_BASE CFG_SLOT0 UDT_PCIO_READ_DIO CFG_SLOT1 UDT_PCIO_READ_AIO CFG_SLOT2 UDT_PCIO_READ_EMPTY CFG_SLOT3 UDT_PCIO_READ_EMPTY

If the configuration DB does not match the PCIO hardware, there is an error message like „no DIO module on this slot installed“(see also chapter 8.2 “Special error codes of the PC IO driver” on page 42).

Read inputs of base board (incl. Counter/encoder)

Reading inputs of the base board including the counter/encoder is done by **UDT_PCIO_READ_BASE**.

Table 6-15 Structure of UDT_PCIO_READ_BASE

Name	Type	Description
ParamType	BYTE	Do not change!
Length	INT	Do not change!
STATUS	WORD	status of the 'read base'
BoardNo	INT	PCIO base board number
ENC_STATUS	DWORD	encoder status and digital input of base board (4 bits) [*1)]
ENC_VAL	ARRAY [0 .. 3]	
COUNT_VAL	DWORD	counter/encoder register
ZERO_MARK	DWORD	zero mark register
UNIV_0	DWORD	universal register 0
UNIV_1	DWORD	universal register 1

[*1)] see also “PCIO operating manual” chapter “Encoder status register””

Read inputs of DIO module

Reading inputs of one DIO module is done by **UDT_PCIO_READ_DIO**.

Table 6-16 Structure of UDT_PCIO_READ_DIO

Name	Type	Description
ParamType	BYTE	Do not change!
Length	INT	Do not change!
STATUS	WORD	status of the 'read DI' of this DIO module
DIG_IN	DWORD	digital input of this module (24 Bits used)

Read inputs of AIO module

Note Analogue inputs are read only, if they are enabled in the configuration (PCIO_CONFIG).

Reading of inputs of one AIO module is done by **UDT_PCIO_READ_AIO**.

Table 6-17 Structure of UDT_PCIO_READ_AIO

Name	Type	Description
ParamType	BYTE	Do not change!
Length	INT	Do not change!
STATUS	WORD	status of the 'read AI' of this AIO module
AI_VAL	ARRAY [0..7]	
RAW	WORD	Raw value from AIO module
SCALED	WORD	Scaled to S7 range [*1)]
FLOAT	REAL	Floating point value
PT100	ARRAY [0..3]	
	INT	PT100 results [0,1 °C] (linear equation)
PT100_RAW	ARRAY [0..3]	
	WORD	Raw values of PT100
REF_100	WORD	Raw value of 100 Ω
REF_200	WORD	Raw value of 200 Ω

[*1)] The PCIO board uses intern a special scaling (depending from range). The values are transferred to S7 scaling.

Table 6-18 Analogue input with range 0..5 V

Voltage [V]	PCIO raw value	S7 scaling
0	0x000	0x0000
2,5	0x800	0x3600
5	0xFFF	0x6C00

Table 6-19 Analogue input with range 0..10 V

Voltage [V]	PCIO raw value	S7 scaling
0	0x000	0x0000
5	0x800	0x3600
10	0xFFF	0x6C00

Table 6-20 Analogue input with range -5..+5 V

Voltage [V]	PCIO raw value	S7 scaling
-5	0x800	0x9400
-2,5	0xC00	0xCA00
0	0x000	0x0000
2,5	0x400	0x3600
5	0x7FF	0x6C00

Table 6-21 Analogue input with range -10..+10 V

Voltage [V]	PCIO raw value	S7 scaling
-10	0x800	0x9400
-5	0xC00	0xCA00
0	0x000	0x0000
5	0x400	0x3600
10	0x7FF	0x6C00

6.5 Write outputs with PCIO_WRITE

The function block **PCIO_WRITE** writes all outputs of one PCIO board including all modules.

Table 6-22 Parameters of the FB PCIO_READ

Parameter	In/Out	Type	Description
DATA	In	Any	Any-Pointer to DB with WRITE data
ERROR	Out	BOOL	Error occurred (if 1, the STATUS gives more detailed information about the problem)
STATUS	Out	WORD	Error code

Attention **The parameter ‚DATA' may only be connected with a data block (DB) – see example project. A general ANY pointer is not allowed.**

6.5.1 Data structure for writing outputs

To use the same function block for writing the outputs, independent from the PCIO hardware configuration, the write data block always consists of fife UDTs:

- UDT_PCIO_WRITE_BASE
- four UDTs for the modules (DIO / AIO / empty)

The four UDTs must match the hardware configuration of the PCIO board.

Table 6-23 Examples for structure of write DB according the PCIO hardware

PCIO hardware	Write data DB
  Slot 0: empty Slot 1: empty Slot 2: empty Slot 3: empty	**Name** **Type** CFG_BASE UDT_PCIO_WRITE_BASE CFG_SLOT0 UDT_PCIO_WRITE_EMPTY CFG_SLOT1 UDT_PCIO_WRITE_EMPTY CFG_SLOT2 UDT_PCIO_WRITE_EMPTY CFG_SLOT3 UDT_PCIO_WRITE_EMPTY
  Slot 0: DIO Slot 1: AIO Slot 2: empty Slot 3: empty	**Name** **Type** CFG_BASE UDT_PCIO_WRITE_BASE CFG_SLOT0 UDT_PCIO_WRITE_DIO CFG_SLOT1 UDT_PCIO_WRITE_AIO CFG_SLOT2 UDT_PCIO_WRITE_EMPTY CFG_SLOT3 UDT_PCIO_WRITE_EMPTY

If the configuration DB does not match the PCIO hardware, there is an error message like „no DIO module on this slot installed“(see also chapter 8.2 “Special error codes of the PC IO driver” on page 42).

Write outputs of base board (incl. Counter/encoder)

The base board does not own any output values. That’s why the **UDT_PCIO_WRITE_BASE** holds only the PCIO board number (0..2).

Table 6-24 Structure of UDT_PCIO_WRITE_BASE

Name	Type	Description
ParamType	BYTE	Do not change!
Length	INT	Do not change!
BoardNo	INT	PCIO base board number

Write outputs of DIO module

Writing outputs of one DIO module is done by **UDT_PCIO_WRITE_DIO**.

Table 6-25 Structure of UDT_PCIO_WRITE_DIO

Name	Type	In/Out	Description
ParamType	BYTE	Out	Do not change!
Length	INT	Out	Do not change!
STATUS	WORD	In	status of the 'write DO' for this DIO module
DIG_OUT	DWORD	Out	digital input of this module (16 Bits used)
DO_ERROR	DWORD	In	error on DO detected (bit = 1 -> error) [*1)]

[*1)] The DIO module recognises errors at the digital output channels. The error is only signalled as “sum error” four 8 outputs. That’s why the value of DO_ERROR identifies the group of digital outputs only. There is no information about the specific faulty output channel.

Table 6-26 Meaning of DO_ERROR

Wert	Description
0x0000	No error on DO recognised
0x00FF	Error on one or more outputs of 0..7 recognised
0xFF00	Error on one or more outputs of 8..15 recognised
0xFFFF	Error on one or more outputs of 0..7 and of 8..15 recognised

The open load detection only works with output of 0. The detection of the error needs approximately 500 µs i.e. after a change from 1 to 0 an error will be detected in the next PLC cycle.

Write outputs of AIO module

Note Analogue outputs are written only, if they are enabled in the configuration (PCIO_CONFIG).

Writing outputs of one AIO module is done by **UDT_PCIO_WRITE_AIO**.

Table 6-27 Structure of UDT_PCIO_WRITE_AIO

Name	Type	In/Out	Description
ParamType	BYTE	Out	Do not change!
Length	INT	Out	Do not change!
STATUS	WORD	In	status of the 'write AO' of this AIO module
OutValue	ARRAY [0 .. 7] OF WORD	Out	Output values (S7 scaling) [*1)]

[*1)] The PCIO board uses 16 bit values for analogue output (15 bit + 1 bit for sign). Because of compatibility the well known S7 scaling is used in the WinAC interface.

Table 6-28 Scaling analogue output

S7 scaling	PCIO raw value	Voltage [V]
0x9400	0x0000	-10 V
0xCA00	0x4000	- 5 V
0x0000	0x8000	0 V
0x3600	0xC000	5 V
0x6C00	0xFFFF	10 V

6.6 SW interrupt (OB52-OB54)

For every PCIO board there is a parameter “OB number for interrupt call” (PCIO_CONFIG).

To transfer the interrupt data into the WinAC OB the limited amount of local data of the OB is used:

Table 6-29 Meaning of local data of interrupt OB

Name	Type	Addr.	Description
dataType2	Byte	4.0	Error IRQ Bits
dataType1	Byte	5.0	Counter/Encoder: Tn
data1	Word	6.0	Counter/Encoder: Sn, Comp., Overflow, zero mark
data2	DWord	8.0	Actual DI (4 module x 8 DI)

The actual values of all interrupt digital inputs of all four modules are transferred to the interrupt OB. The PCIO board signals an interrupt for every change of these digital inputs. The edge detection has to be done by the user program in the WinAC.

To access all the interrupt sources symbolic the **UDT_PCIO_IRQ_DATA** is prepared in the example project. In the OB52 of the example project the interrupt information is copied to a variable of this type (UDT). After that the information is accessible symbolically.

7 Examples for applications

7.1 Use of the STEP 7 example project

The STEP 7 example project is realised for the following PCIO configuration:

- one PCIO board
- Slot 0: DIO
- Slot 1: AIO
- Slot 2: DIO
- Slot 3: empty

If using another PCIO configuration you have to adapt the example project.

Of course you have to check the name and the IP address of the WinAC in your hardware configuration.

7.1.1 Structure of the STEP 7 example program

OB100 Complete Restart

The driver has to be started (**PCIO_INIT**). Additional within the OB100 the PCIO Board including the installed modules is configured (**PCIO_CONFIG**).

At the end of this OB some internal counters are initialized.

OB1 CYCL_EXEC

In the beginning of OB1 the inputs are read. After that the processing follows. At the end the outputs are written.

In the example project some lines are comments – you can use it for reading and writing in “single shoot” mode.

OB52 (Interrupt)

In the OB52 the information about the interrupt sources are copied to a variable of type **UDT_PCIO_IRQ_DATA**. Thus this information can be accessed symbolically.

In network 3 you see an example for edge detection of the interrupt digital inputs. This is necessary because the PCIO triggers an interrupt for every change of a digital input.

In the following networks the internal counters for the different encoder/counter interrupt sources are processed.

DB200 - configuration

This DB contains the configuration for the PCIO board. According to the PCIO hardware this DB consists of fife UDTs: “Config-Base”, “Config-DIO”, “Config-AIO”, “Config-DIO”, “Config-Empty”.

DB201 – read data from PCIO

This DB is used for storing the read data from PCIO. According to the PCIO hardware this DB consists of fife UDTs: “Read-Base”, “Read-DIO”, “Read-AIO”, “Read-DIO” and “Read-Empty”.

DB202 – write data to PCIO

This DB is used for storing the data for writing to PCIO. According to the PCIO hardware this DB consists of fife UDTs: “Write-Base”, “Write-DIO”, “Write-AIO”, “Write-DIO” and “Write-Empty”.

DB1000 – internal Variables

This DB contains a number of internal values. The example project does not use any flags. That’s why this DB is used.

7.2 Adapt the STEP 7 example to own demands

7.2.1 Other modules used than in the example

When using other modules than in the example project, the following changes have to be done:

- Build up the configuration DB with the right UDTs according the PCIO hardware structure (see chapter 6.3.1 „Structure of the configuration data for one PC IO board“ on page 29)
  In the example project you have to change the DB200. It must contain the matching UDTs according to the hardware configuration of the PCIO.
- Build up the DB for reading inputs with the right UDTs according the PCIO hardware structure (see chapter 6.4.1 „Data structure for reading inputs“ on page 32)
  In the example project you have to change the DB201. It must contain the matching UDTs according to the hardware configuration of the PCIO.
- Build up the DB for writing outputs with the right UDTs according the PCIO hardware structure (see chapter 6.5.1 “Data structure for writing outputs” on page 35)
  In the example project you have to change the DB202. It must contain the matching UDTs according to the hardware configuration of the PCIO.

7.2.2 Using more than one PCIO board

When using more than one PCIO board, the following changes have to be done:

- Use different interrupt lanes for all PCIO boards (maximum three)
  (see chapter 3.2 „Installation hardware PCIO in Microbox PC427B“, S. 11)
- Check if all PCIO boards use own exclusive interrupts
  (see „check exclusive interrupt“ on page 18)
- Call **PCIO_INIT** only **one time**, independent of the number of used PCIO boards.
- Change number of PCIO board in the DBs for the PCIO function blocks (e.g. CFG_BASE.BoardNo = 2)
- The configuration has to be done separately for every PCIO board. That means for every PCIO board a separate configuration DB is needed (see chapter 6.3.1 „Structure of the configuration data for one PC IO board“, S. 29).
  The FB **PCIO_CONFIG** has to be called for every PCIO board one time.
- It is recommended to use a separate DB for reading input / writing output for every PCIO board. The function block **PCIO_READ** respectively **PCIO_WRITE** have to be called for every PCIO board.

7.2.3 Using peripheral area

To simplify the usage of PCIO periphery you can copy the inputs to the peripheral area and pick up the outputs from the peripheral area. Because this solution depends on your concrete configuration it is not included in the sample S7 project.

A STEP 7 project using the peripheral area could look like this:

Table 7-1 STEP 7 program with peripheral area

```
// read input from PCIO
CALL "PCIO_READ", "DBI_PCIO_READ"
    DATA = "DB_READ_0"
    ...

// copy input to peripheral area
L DB_READ_0.RD_DIO_DIG_IN
T ED 0


// now you can use standard IN peripheral variables
U E0.0
U E0.1
...


// work with standard OUT peripheral variables
= A1.5
SET
S A1.0

// copy the output from peripheral area
L AD 0
T DB_WRITE0.WR_DIO_DIG_OUT

// write output to PCIO
CALL "PCIO_WRITE"."DBI_PCIO_WRITE"
    DATA = DB_WRITE_0
    ...
```

8 Error Codes

The WinAC PCIO driver can provide different classes of error messages:

- Code in the FB-output **STATUS** according to WinAC-ODK (see chapter 8.1 in this document)
- Special error codes of the PCIO driver (see chapter 8.2 on page 42 in this document)

8.1 Error codes of WinAC ODK 4.1

The WinAC PCIO driver had been developed with the WinAC ODK (Open Development Kit). The ODK can also generate error codes, which are returned from the **STATUS** of the FBs.

8.1.1 Error Codes of SFB65001 CREA_COM

These error codes can only be returned from FB **PCIO_INIT**

Table 8-1 WinAC ODK error messages for CREA_COM

Error Code	Symbol	Description
0	NO_ERRORS	Success
0x807F	ERROR_INTERNAL	An internal error occurred.
0x8001	E_EXCEPTION	An exception occurred.
0x8102	E_CLSID_FAILED	The call to CLSIDFromProgID failed.
0x8103	E_COINITIALIZE_FAILED	The call to CoInitializeEx failed.
0x8104	E_CREATE_INSTANCE_FAILED	The call to CoCreateInstance failed.
0x8105	E_LOAD_LIBRARY_FAILED	The library failed to load.
0x8106	E_NT_RESPONSE_TIMEOUT	A Windows response timeout occurred.
0x8107	E_INVALID_OB_STATE	Controller is in an invalid state for scheduling an OB.
0x8108	E_INVALID_OB_SCHEDULE	Schedule information for OB is invalid.
0x8109	E_INVALID_INSTANCEID	Instance ID for SFB65001 call is invalid.
0x810A	E_START_ODKPROXY_FAILED	Controller could not load proxy DLL.
0x810B	E_CREATE_SHAREMEM_FAILED	The WinAC controller could not create or initialize shared memory area.
0x810C	E_OPTION_NOT_AVAILABLE	Attempt to access unavailable option ocurred.

8.1.2 Error Codes für SFB65002 EXEC_COM

These error codes can be returned from all FBs.

Table 8-2 WinAC ODK error messages for EXEC_COM

Error Code	Symbol	Description
0	NO_ERRORS	Success
0x807F	ERROR_INTERNAL	An internal error occurred.
0x8001	E_EXCEPTION	An exception occurred.
0x8002	E_NO_VALID_INPUT	Input: the ANY pointer is invalid.
0x8003	E_INPUT_RANGE_INVALID	Input: the ANY pointer range is invalid.
0x8004	E_NO_VALID_OUTPUT	Output: the ANY pointer is invalid.
0x8005	E_OUTPUT_RANGE_INVALID	Output: the ANY pointer range is invalid.
0x8006	E_OUTPUT_OVERFLOW	More bytes were written into the output buffer by the extension object than were allocated.
0x8007	E_NOT_INITIALIZED	ODK system has not been initialized: no previous call to SFB65001 (CREA_COM).
0x8008	E_HANDLE_OUT_OF_RANGE	The supplied handle value does not correspond to a valid extension object.
0x8009	E_INPUT_OVERFLOW	More bytes were written into the input buffer by the extension object than were allocated.

8.2 Special error codes of the PCIO driver

Among the general error bit of the driver FBs there is a special error code in the value of **STATUS** to describe the reason of the problem.

Table 8-3 Error codes of PCIO driver

0x0000 - no error

errors of PCIO driver DLL

0x8501 - detected to many PCIO boards on PCI bus

0x8502 - RtTranslateBusAddress failed

0x8503 - Failure on RtMapMemory

0x8504 - RtAttachInterruptVectorEx failed

0x8505 - byte mirroring not successful

0x8506 - no byte mirroring test processed!

0x8511 - false base board no (0..3)

0x8512 - sensor channel not known for PCIO base

0x8513 - no valid sensor channel number (0..3) for PCIO base board

0x8514 - base address BA unknown

0x8521 - modul number not known for DIO
0x8522 - no valid modul number (0..3) for DIO
0x8523 - too many DIO modules found for one PCIO base board
0x8524 - no DIO moudule on this slot
0x8525 - DIO is just booting (not ready for use yet)

0x8531 - modul number not known for AIO
0x8532 - no valid modul number (0..1) for AIO
0x8533 - slot unknown for AIO module
0x8534 - no valid AO channel number (0..7)
0x8535 - no valid AI channel number (0..7)
0x8536 - no more values to AO convert.
0x8537 - no more value to AI convert.
0x8538 - AO data struct not defined
0x8539 - AI data struct not defined
0x853A - too many AIO modules found for one PCIO base board
0x853B - undefined range for analoge input
0x853C - analog out conversion is bussy

0x8551 - Error RtGetClockTime() for start time
0x8552 - Error RtGetClockTime() for end time
0x8553 - internal table for time stamps is full
0x8554 - undefined ID for time stamp table

errors with WinAC Handling (ODK part)
0x9001 - error using ODK_Read.. function
0x9002 - error using ODK_Write.. function

errors with configuration
0x9011 - false PCIO board number
0x9012 - false number for WinAC IRQ OB (only 52-54 allowed)
0x9013 - false lenght of <config section>
0x9014 - false <read section>
0x9015 - no AIO module on this slot installed
0x9016 - read for to many AIO modules
0x9017 - invalid range for AI channel
0x9018 - no DIO module on this slot installed
0x9019 - slot is not empty

errors with reading from PCIO

0x9021 - false lenght of <read section>

0x9022 - expected <read base section>

0x9023 - false <read section>

0x9023 - false PCIO board number

0x9025 - read for to many AIO modules

0x9026 - no AIO module on this slot installed

0x9027 - got no valid AI data from AIO

0x9028 - no DIO module on this slot installed

0x9029 - got no valid DI data from DIO

0x902A - no DIO module on this slot installed

errors with writing to PCIO

0x9031 - false lenght of <write section>

0x9032 - expected <write base section>

0x9033 - false <write section>

0x9034 - false PCIO board number

0x9035 - read for to many AIO modules

0x9036 - no AIO module on this slot installed

0x9037 - got no valid AI data from AIO

0x9038 - no DIO module on this slot installed

0x9039 - got no valid DI data from DIO

0x903A - no DIO module on this slot installed

0x903B - error on DO detected

other errors

0x9101 - error creating event for signaling PCIO-IRQ to WinAC

0x9102 - multiple creation of IRQ monitor
(perhaps multiple PCIO_INIT calls)

9 Abbreviations

AIO	Analogue In/Out: extension module for PCIO board
DB	Data block
DIO	Digital In/Out: extension module for PCIO board
FB	Function block
OB	Organisation block
PC IO	Personal Computer Input / Output
RTX	Real Time eXtension for Windows
UDT	User defined type (definition of data type in STEP 7)

10 History

Table 10-1

Version	Date	Remark
V 1.0	31.10.07	First version for delivery
V 1.1	02.04.09	Tested with WinAC RTX 2008
V 1.2	02.04.09	Tested with WinAC RTX 2009