Copyright

Acknowledgments

PC-LabCard is a trademark of Advantech Co., Ltd. IBM and PC are trademarks of International Business Machines Corporation. MS-DOS, Windows, Microsoft Visual C++ and Visual BASIC are trademarks of Microsoft Corporation. Intel and Pentium are trademarks of Intel Corporation. Delphi and C++ Builder are trademarks of Inprise Corporation.

CE notification

The PCI-1714, developed by ADVANTECH CO., LTD., has passed the CE test for environmental specifications when shielded cables are used for external wiring. We recommend the use of shielded cables. This kind of cable is available from Advantech. Please contact your local supplier for ordering information.

On-line Technical Support

For technical support and service, please visit our support website at:
http://www.advantech.com/support

Part No. 2003171400 1st Edition
Printed in Taiwan July 2003

(This page is left blank for hard printing.)

Contents

1. Introduction

Thank you for buying the Advantech PCI-1714. The PCI-1714 is a 30MHz Simultaneous 4-CH Analog Input Card for the PCI bus. It is an advanced-performance data acquisition card based on 32-bit PCI Bus architecture. The maximum sampling rate of PCI-1714 is up to 30MHz samples per second, with an emphasis on continuous, non-stop, high-speed, streaming data of A/D samples to host memory.

The following sections of this chapter will provide further information about features of the multifunction cards, a Quick Start for installation, together with some brief information on software and accessories for the PCI-1714 card.

1.1 Features

The Advantech PCI-1714 offers the following main features:

- 32-bit PCI-Bus Mastering DMA data transfer
- 4 A/D converters simultaneously sampling
- 12-bit A/D converter up to 30M samples per second
- 4 single-ended analog input channels
- Programmable gain for each input channel
- On board FIFO memory
- Multiple A/D triggering modes
- Programmable pacer/counter
- Auto calibration

Some of them are highlighted and more detailed as following.

PCI-Bus Mastering Data Transfer

The PCI-1714 supports PCI-Bus mastering DMA for high-speed data transfer. By setting aside a block of memory in the PC, the PCI-1714 performs bus-mastering data transfers without CPU intervention, freeing the CPU to perform other more urgent tasks such as data analysis and graphic manipulation. The function allows users to run all I/O functions simultaneously at full speed without losing data.

Simultaneous Sampling

The PCI-1714 capable of simultaneous sampling uses 4 identical circuitries and ADC for each analog input channel. Where the time relationship between inputs is important, this feature let you sample simultaneously.

Supports S/W, Internal and External Pacer Triggering

The PCI-1714 supports three kinds of trigger modes for A/D conversion: software triggering, internal pacer triggering and external pacer triggering. The software trigger allows users to acquire a sample when it is needed; the internal pacer triggers continuously high-speed data acquisition. The PCI-1714 also accepts external trigger sources, allowing synchronous sampling with external devices.

On-board FIFO Memory

There are 32K samples of FIFO memory on PCI-1714. This is an important feature for faster data transfer and more predictable performance under Windows system.

Auto Calibration

PCI-1714 features software auto calibration. There is no variable resister trimming required. This is convenient for user to calibrate.

Note:

✎ For detailed specifications of the PCI-1714, please refer to Appendix A, Specifications.

1.2 Applications

The following are some of the possible applications of PCI-1714:

- Testing Instrument
- Ultrasound Imaging
- Gamma Camera Imaging
- CCD Camera Imaging
- Video Digitizing

1.3 Installation Guide

Before you install your PCI-1714 card, please make sure you have all the following necessary components:

PCI-1714 DA&C card	
PCI-1714 User's Manual	This manual
Driver software	Advantech DLL drivers (included in the companion CD-ROM)
Wiring cable	PCL-10901-1, BNC-BNC (optional)
Wiring board	ADAM-3909 (optional)
Computer	Personal computer or workstation with a PCI-bus slot (running Windows 2000/95/98/ ME/NT/XP)

Some other optional components are also available for enhanced operation:

Application software	ActiveDAQ, GeniDAQ or other third-party software packages

After you get the necessary components and maybe some of the accessories for enhanced operation of your Multifunction card, you can then begin the Installation procedures. Figure 1-1 on the next page provides a concise flow chart to give users a broad picture of the software and hardware installation procedures:

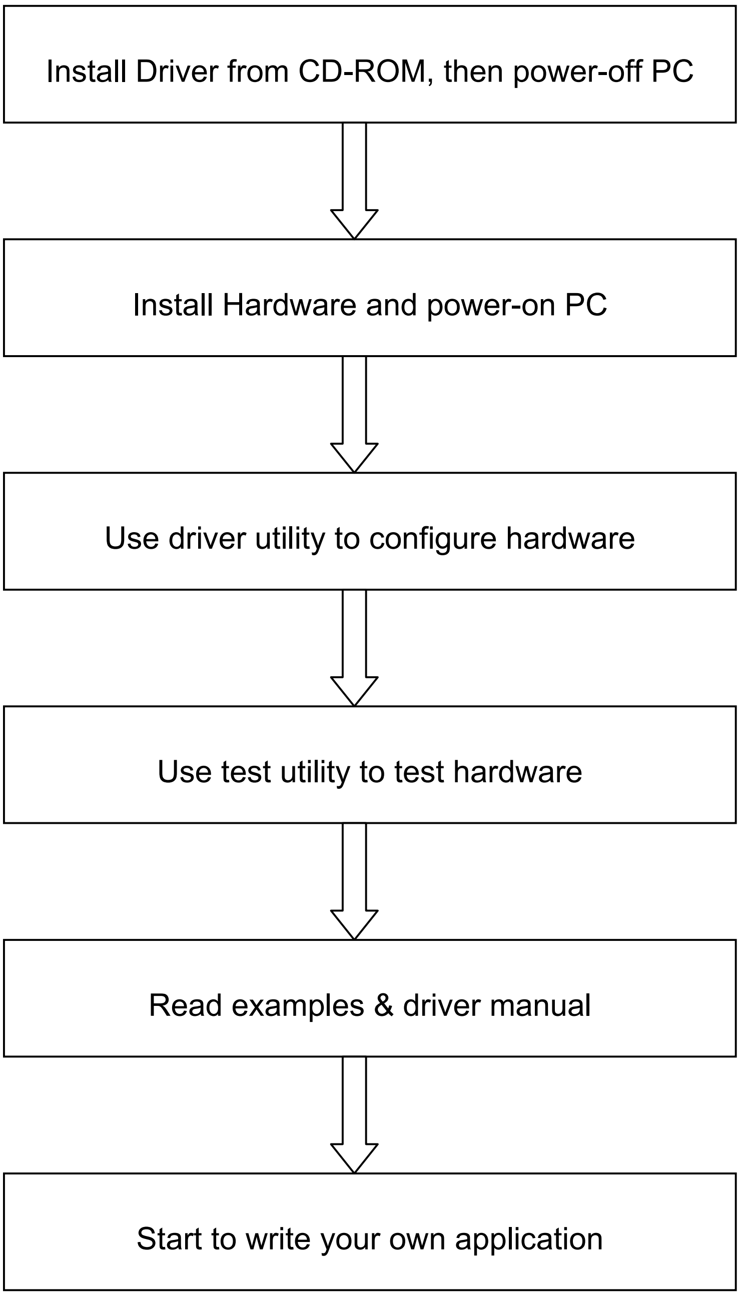


Fig. 1-1 Installation Flow Chart

1.4 Software Overview

Advantech offers a rich set of DLL drivers, third-party driver supports and application software to help fully utilize the functions of your PCI-1714 card:

- Device Drivers (on the companion CD-ROM)
- LabVIEW driver
- Advantech ActiveDAQ
- Advantech GeniDAQ

Programming choices for DA&C cards

You may use Advantech application software such as Advantech Device Drivers. On the other hand, advanced users can use another option for register-level programming, although it is not recommended due to its laborious and time-consuming nature.

Device Drivers

The Advantech Device Drivers software is included on the companion CD-ROM at no extra charge. It also comes with all Advantech DA&C cards. Advantech's device drivers feature a complete I/O function library to help boost your application performance. The Advantech Device Drivers for Windows 2000/95/98/ME/NT/XP works seamlessly with development tools such as Visual C++, Visual Basic, Inprise C++ Builder and Inprise Delphi.

Register-level Programming

Register-level programming is reserved for experienced programmers who find it necessary to write code directly at the level of device registers. Since register-level programming requires much effort and time, we recommend that you use the Advantech Device Drivers instead. However, if register-level programming is necessary, you should refer to the relevant information in *Appendix C, Register Structure and Format*, or to the example codes included on the companion CD-ROM.

1.5 Device Drivers Programming Roadmap

This section will provide you a roadmap to demonstrate how to build an application from scratch using Advantech Device Drivers with your favorite development tools such as Visual C++, Visual Basic, Delphi and C++ Builder. The step-by-step instructions on how to build your own applications using each development tool will be given in the *Device Drivers Manual*. Moreover, a rich set of example source code is also given for your reference.

Programming Tools

Programmers can develop application programs with their favorite development tools:

- ❑ **Visual C++**
- ❑ **Visual Basic**
- ❑ **Delphi**
- ❑ **C++ Builder**

For instructions on how to begin programming works in each development tool, Advantech offers a *Tutorial* Chapter in the *Device Drivers Manual* for your reference. Please refer to the corresponding sections in this chapter of the *Device Drivers Manual* to begin your programming efforts. You can also look at the example source code provided for each programming tool, since they can get you very well oriented.

The *Device Drivers Manual* can be found on the companion CD-ROM. Or if you have already installed the Device Drivers on your system, The *Device Drivers Manual* can be readily accessed through the ***Start*** button:

Start/Programs/Advantech Driver V2.1/Device Driver Manual

The example source codes could be found under the corresponding installation folder such as the default installation path:

\Program Files\Advantech\ADSAPI\Examples

For information about using other function groups or other development tools, please refer to the *Creating Windows 95/NT/2000 Application with Device Drivers* chapter and the *Function Overview* chapter on the *Device Drivers Manual*.

Programming with Device Drivers Function Library

Advantech Device Drivers offers a rich function library to be utilized in various application programs. This function library consists of numerous APIs that support many development tools, such as Visual C++, Visual Basic, Delphi and C++ Builder.

According to their specific functions or services, those APIs can be categorized into several function groups:

Device Function
Analog Input/Output Function
Digital Input/Output Function
Port I/O Function
Counter Function
Temperature Measurement Function
Temperature measurement Function
Alarm Function
Communication port Function
High speed Function
Hardware Function

For the usage and parameters of each function, please refer to the *Function Overview* chapter in the *Device Drivers Manual*.

Troubleshooting Device Drivers Error

Driver functions will return a status code when they are called to perform a certain task for the application. When a function returns a code that is not zero, it means the function has failed to perform its designated function. To troubleshoot the Device Drivers error, you can pass the error code to **DRV_GetErrorMessage** function to return the error message. Or you can refer to the *Device Drivers Error Codes* Appendix in the *Device Drivers Manual* for a detailed listing of the Error Code, Error ID and the Error Message.

1.6 Accessories

Advantech offers a complete set of accessory products to support the PCI-1714 card. These accessories include:

Wiring Cable

PCL-10901-1 The PCL-10901-1 cable is specially designed for PCI-1714 cards to connect to the wiring board, ADAM-3909, for external synchronization signal source, such like an external trigger and/or clock signal.

Source Signal Cable The source signal cable is designed as BNC-BNC or BNC for one side and open-end for the other. The cable links the PCI-1714 cards with the signal source via the BNC connector. There are all four BNC ports on board available for simultaneous signal input.

Wiring Boards

❑ **ADAM-3909** The ADAM-3909 is a DB-9 Wiring Terminal for DIN-rail Mounting. This terminal module can be readily connected to the Advantech PC-Lab cards and allows easy yet reliable access to individual pin connections for the PCI-1714 card.

2. Installation

This chapter gives users a package item checklist, proper instructions about unpacking and step-by-step procedures for both driver and card installation.

2.1 Unpacking

After receiving your PCI-1714 package, please inspect its contents first. The package should contain the following items:

☑ PCI-1714 card
☑ Companion CD-ROM (DLL driver included)
☑ User's Manual

The PCI-1714 card harbors certain electronic components vulnerable to *electrostatic discharge* (ESD). ESD could easily damage the integrated circuits and certain components if preventive measures are not carefully paid attention to.

Before removing the card from the antistatic plastic bag, you should take following precautions to ward off possible ESD damage:

- Touch the metal part of your computer chassis with your hand to discharge static electricity accumulated on your body. Or use a grounding strap.
- Touch the anti-static bag to a metal part of your computer chassis before opening the bag.
- Take hold of the card only by the metal bracket when removing it from the bag.

After taking out the card, first you should:

- Inspect the card for any possible signs of external damage (loose or damaged components, etc.). If the card is visibly damaged, please notify our service department or the local sales representative immediately. Avoid installing a damaged card into your system.

Also, pay extra caution to the following aspects to ensure proper installation:

- Avoid physical contact with materials that could hold static electricity such as plastic, vinyl and Styrofoam.
- Whenever you handle the card, grasp it only by its edges. DO NOT TOUCH the exposed metal pins of the connector or the electronic components.

Note:

- Keep the anti-static bag for future use. You might need the original bag to store the card if you have to remove the card from the PC or transport it elsewhere.

2.2 Driver Installation

We recommend you to install the driver before you install the PCI-1714 card into your system, since this will guarantee a smooth installation process.

The Advantech Device Drivers Setup program for the PCI-1714 card is included on the companion CD-ROM that is shipped with your DA&C card package. Please follow the steps below to install the driver software:

Step 1: Insert the companion CD-ROM into your CD-ROM drive.

Step 2: The Setup program will be launched automatically if you have the autoplay function enabled on your system. When the Setup Program is launched, you’ll see the following Setup Screen.

Note:

✎ If the autoplay function is not enabled on your computer, use Windows Explorer or the Windows ***Run*** command to execute SETUP.EXE on the companion CD-ROM.

Fig. 2-1 The Setup Screen of Advantech Automation Software

Step 3: Select the *Individual Drivers* option.

Step 4: Select the specific device then just follow the installation instructions step by step to complete your device driver setup.

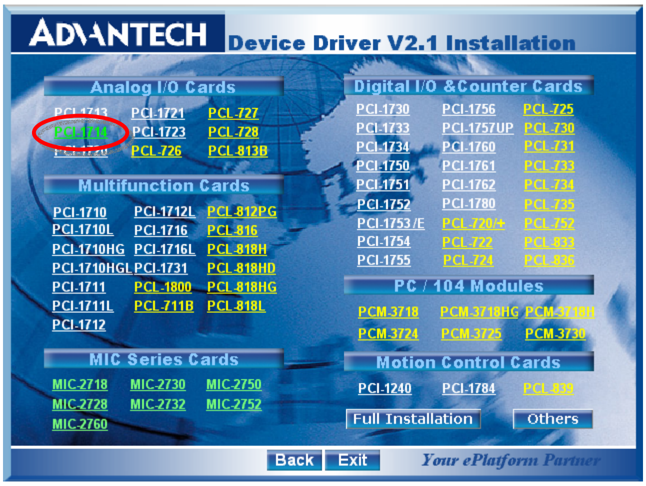


Fig. 2-2 Different options for Driver Setup

For further information on driver-related issues, an online version of *Device Drivers Manual* is available by accessing the following path:

Start/Programs/Advantech Device Drivers V2.1/Device Driver Manual

2.3 Hardware Installation

After the DLL driver installation is completed, you can now go on to install the PCI-1714 card in any PCI slot on your computer. But it is suggested that you should refer to the computer user manual or related documentation if you have any doubt. Please follow the steps below to install the card on your system.

Note:

✎ Make sure you have installed the driver first before you install the card (please refer to *2.2 Driver Installation*)

Step 1: Turn off your computer and unplug the power cord and cables. TURN OFF your computer before installing or removing any components on the computer.

Step 2: Remove the cover of your computer.

Step 3: Remove the slot cover on the back panel of your computer.

Step 4: Touch the metal part on the surface of your computer to neutralize the static electricity that might be on your body.

Step 5: Insert the PCI-1714 card into a PCI slot. Hold the card only by its edges and carefully align it with the slot. Insert the card firmly into place. Use of excessive force must be avoided, otherwise the card might be damaged.

Step 6: Fasten the bracket of the PCI-1714 card on the back panel rail of the computer with screws.

Step 7: Connect appropriate accessories (such as source /sync signal cables, wiring terminals, etc. if necessary) to the PCI-1714 card.

Step 8: Replace the cover of your computer chassis. Re-connect the cables you removed in **Step 2**.

Step 9: Plug in the power cord and turn on the computer.

Note:

✎ In case you installed the card without installing the DLL driver first, Windows 95/98/ME will recognize your card as an “unknown device” after rebooting, and will prompt you to provide the necessary driver. You should ignore the prompting messages (just click the ***Cancel*** button) and set up the driver according to the steps described in *2.2 Driver Installation*.

After the PCI-1714 card is installed, you can verify whether it is properly installed on your system in the *Device Manager*:

1. Access the *Device Manager* through *Control Panel/System/Device Manager*.
2. The *device name* of the PCI-1714 should be listed on the *Device Manager* tab on the System *Property* Page.

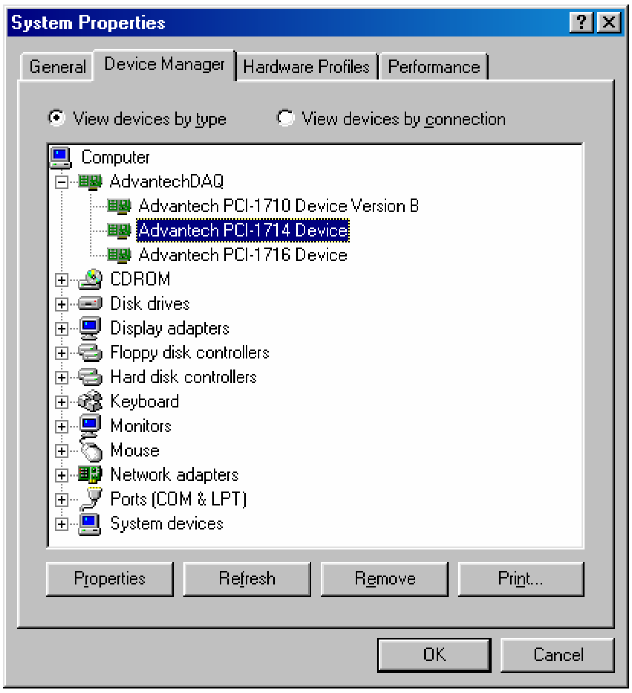


Fig. 2-3 The device name listed in the Device Manager

Note:

✎ If your card is properly installed, you should see the *device name* of your card listed on the *Device Manager* tab. **If you do see your device name listed on it but marked with an exclamation sign "!", it means your card has not been correctly installed.** In this case, remove the card device from the *Device Manager* by selecting its device name and press the ***Remove*** button. Then go through the driver installation process again.

After your card is properly installed on your system, you can now configure your device using the *PCI-1714 Utility* program that has itself already been installed on your system during driver setup. A complete device installation procedure should include *board selection* and *device setup*. After that, you can operate this card through the *operation*. The following sections will guide you through the *board selection*, *device setup* and *operation* of your device.

2.4 Device Setup & Configuration

The *PCI-1714 Utility Program* is a utility that allows you to setup, configure and test your device, and later store your settings on the system registry. These settings will be used when you call the APIs of *Advantech Device Drivers*.

Setting Up the Device

Step 1: To install the I/O device for your card, you must first run the *Device Manager* program (by accessing ***Start/Programs/ Advantech Device Drivers V2.1).***

Step 2: You can then view the device(s) already installed on your system (if any) in the *Installed Devices* list box. Since you haven't installed any device yet, you might see a blank list such as the one below (*Fig. 2-4*).

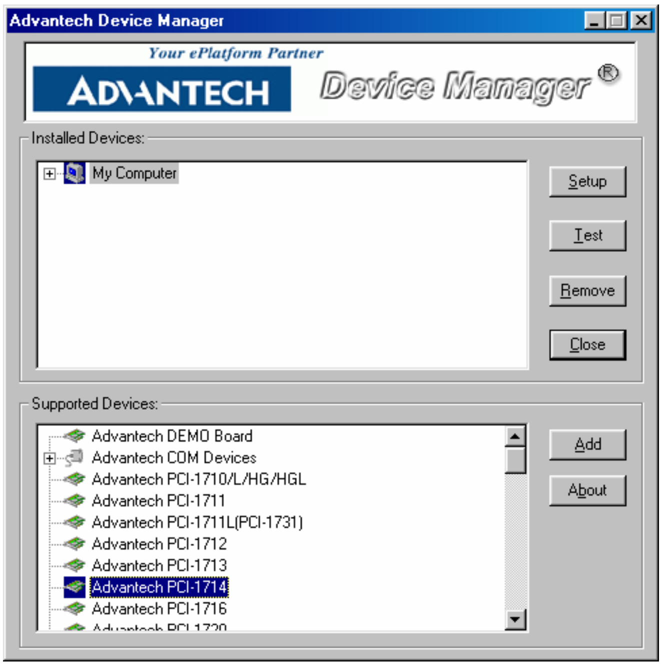


Fig. 2-4 The Device Manager dialog box

Step 3: Scroll down the *Supported Devices* box to find the device that you wish to install, then click the ***Add…*** button to evoke the *Existing unconfigured PCI-1714* dialog box such as one shown in *Fig. 2-5*. The *Existing unconfigured PCI-1714* dialog box lists all the installed devices on your system. Select the device you want to configure from the list box and press the ***OK*** button.

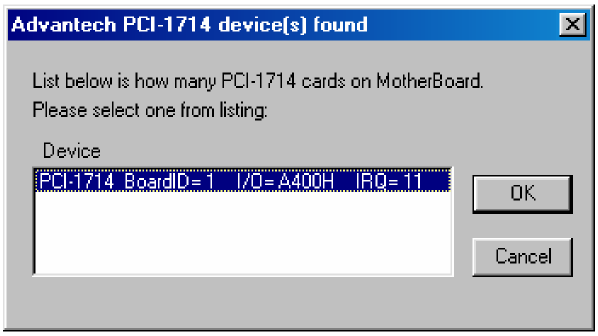


Fig. 2-5 The "Device(s) Found" dialog box

Step 4: After you have finished configuring the device, click ***OK*** and the *device name* will appear in the *Installed Devices* box as the following:

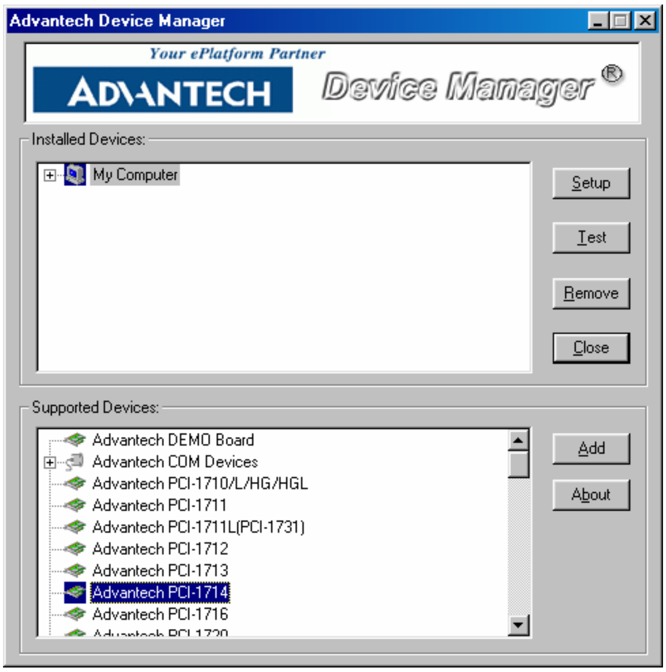


Fig. 2-6 The Device Name appearing on the list of devices box

---

Note:

✎ As we have noted, the *device name* **"001:<PCI-1714 BoardID=7 I/O=c800H>**" begins with a *device number* "000", which is specifically assigned to each card. The *device number* is passed to the driver to specify which device you wish to control.

If you want to test the card device further, go right to the next section on the *Device Testing*.

2.5 Device Testing

Following through the *Setup* and *Configuration* procedures to the last step described in the previous section, you can now proceed to test the device by clicking the ***Test*** button on the *I/O Device Installation* dialog box (Fig. 2-7). A *Device Test* dialog box will appear accordingly:

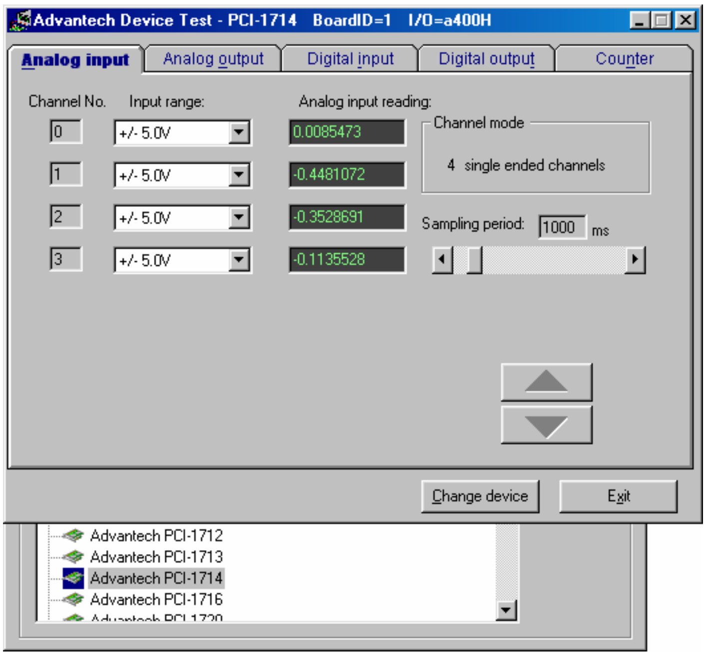


Fig. 2-7 The Device Test dialog box of PCI-1714

On the *Device Test* dialog box, users are free to test various functions of PCI-1714 on the *Analog input* tabs, functions on the other tabs are not supported at this stage.

Note:

✎ You can access the Device Test dialog box either by the previous procedure for the Device Installation Program or simply by accessing Start/Programs/Advantech Driver for 95 and 98 (or for NT) /Test Utility.

Testing Analog Input Function

Click the *Analog Input* tab to bring it up to front of the screen. Select the input range for each channel in the *Input range* drop-down boxes. Configure the sampling rate on the scroll bar. Switch the channels by using the up/down arrow.

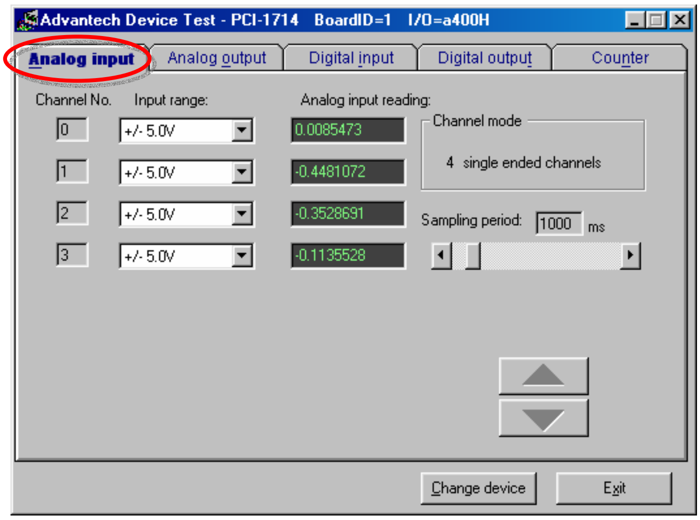


Fig. 2-8 Analog Input tab on the Device Test dialog box

3. Signal Connections

Maintaining proper signal connections is one of the most important factors to ensure that your application system is sending and receiving data correctly. A good signal connection can avoid unnecessary and costly damage to your PC and other hardware devices. This chapter provides useful information about how to connect input and output signals to the PCI-1714 via the I/O connector.

3.1 Overview

Maintaining signal connections is one of the most important factors in ensuring that your application system is sending and receiving data correctly. A good signal connection can avoid unnecessary and costly damage to your PC and other hardware devices. This chapter provides useful information about how to connect input and output signals to the PCI-1714 via the I/O connector.

3.2 Switch and Jumper Settings

The PCI-1714 card has one function switch and five jumper settings.

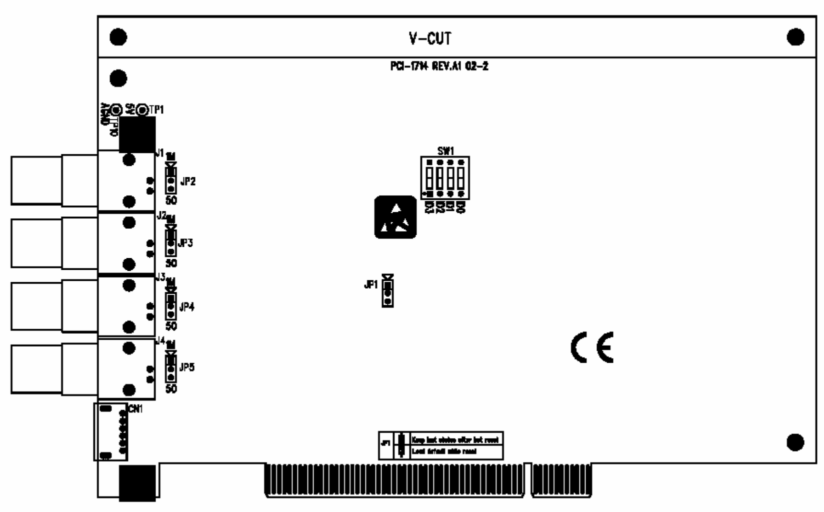


Fig. 3-1 Card connector, jumper and switch locations

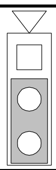
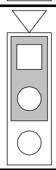
Board ID setting (SW1)

ID3	ID2	ID1	ID0	Board ID
1	1	1	1	0
1	1	1	0	1
1	1	0	1	2
1	1	0	0	3
1	0	1	1	4
1	0	1	0	5
1	0	0	1	6
1	0	0	0	7
0	1	1	1	8
0	1	1	0	9
0	1	0	1	10
0	1	0	0	11
0	0	1	1	12
0	0	1	0	13
0	0	0	1	14
0	0	0	0	15

Note: On: 1, Off: 0

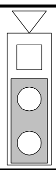
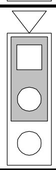
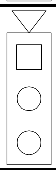
Power on configuration after hot reset (JP1)

To see the table below, User can use the JP1 to set the hot reset type of PCI-1714.

JP1	Power on configuration after hot reset
	Keep the I/O configuration while hot reset.
	Clear the I/O configuration to default while hot reset.

Input terminator select (JP2 to JP5)

To see the table below, User can use the JP2 to JP5 to set input terminator value for each AI channel CH0 to CH3, respectively.

JP2, JP3, JP4, JP5	Input terminator select
	50Ω
	1MΩ
	High impedance

3.3 Signal Connections

CN1 Pin Assignment

Figure 3-2 shows the pin assignments for the PS-2 connector on the PCI-1714 and the DB-9 connector on the cable.

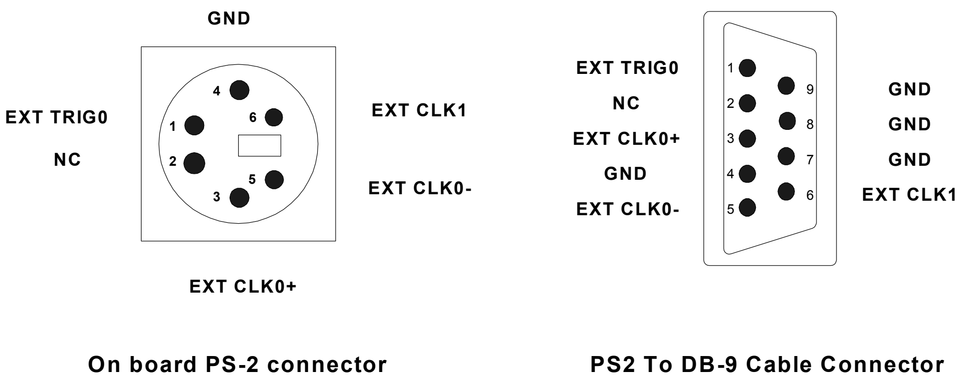


Fig. 3-2 I/O connector pin assignment for the PCI-1714

J1 to J4 BNC are the AI input connectors.

J1 is for AI0, J2 is for AI1, J3 is for AI2 and J4 is for AI3.

4. Principles of Operation

This chapter describes the following features of the PCI-1714 card:

- Analog input ranges and gains
- Analog input acquisition modes
- A/D sample clock sources
- Trigger sources
- Analog Input Data Format

4.1 Analog Input Ranges and Gains

Each channel on the PCI-1714 can measure bipolar analog input signals ranging within ± 10 V FSR, and can be set up with different input ranges respectively. The sampling rate can be up to 30 MS/s.

The PCI-1714 also provides various gain levels that are programmable per channel. *Table 4-1* lists the effective ranges supported by the PCI-1714 using these gains.

Table 4-1 Gains and Analog Input Range

Gain	Analog Input Range
1	**±5 V**
2	**±2.5 V**
5	**±1 V**
10	**±0.5 V**

For each channel, choose the gain level that provides most optimal range that can accommodate the signal range you have to measure. For detailed information, please refer to *Appendix C.4 AI Range Control*.

4.2 Analog Input Acquisition Modes

The PCI-1714 can acquire data in single value, pacer, post-trigger, delay-trigger, about-trigger and pre-trigger acquisition modes. These analog input acquisition modes are described in more details in the following:

Single Value Acquisition Mode

The single value acquisition mode is the simplest way to acquire data. Once the software issues a trigger command, the A/D converter will convert one data, and return it immediately. User can check the A/D FIFO status (Read BASE+10, 12) to make sure if the data is ready to be received. For detailed information, please refer to *Appendix C.8 FIFO Control*, *Appendix C.9 FIFO Status*, *Appendix C.10 FIFO for Progarmmable Flag*.

Pacer Acquisition Mode

Use pacer acquisition mode to acquire data if you want to accurately control the time interval between conversions of individual channels in a scan. A/D conversion clock comes from A/D counter or external clock source on connector. A/D conversion starts when the first clock signal comes in, and will not stop if the clock is still continuously sending into it. Conversion data is put into the A/D FIFO. For high-speed data acquisition, you have to use the DMA data transfer for analog input to prevent data loss.

Post-Trigger Acquisition Mode

Post-trigger allows you to acquire data based on a trigger event. Posttrigger acquisition starts when the PCI-1714 detects the trigger event and stops when the preset number of post-trigger samples has been acquired or when you stop the operation. This trigger mode must work with the DMA data transfer mode enabled. Use post-trigger acquisition mode when you want to acquire data when a post-trigger event occurs. Please specify the following parameters when using software in post-trigger acquisition mode:

- Set to Post-Trigger Acquisition Mode
- The A/D sample clock source and sampling rate
- The trigger source
- The acquired sample number N

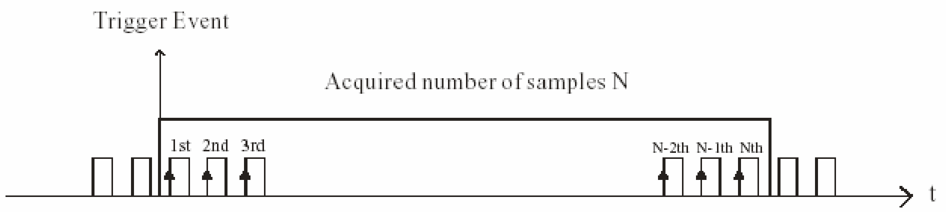


Fig. 4-1 Post-Trigger Acquisition Mode

Delay Trigger Acquisition Mode

In delay trigger mode, data acquisition will be activated after a preset delay number of sample has been taken after the trigger event. The delay number of sample ranges from 2 to 65535 as defined in DMA counter.

Delay-trigger acquisition starts when the PCI-1714 detects the trigger event and stops when the specified number of A/D samples has been acquired or when you stop the operation. This triggering mode must work with the DMA data transfer mode enabled Please specify the following parameters when using software in delay trigger mode:

- Set to Delay-Trigger Acquisition Mode
- The sample clock source and sampling rate
- The trigger source
- The acquired sample number N
- The sample number M delays after the delay-trigger event happened

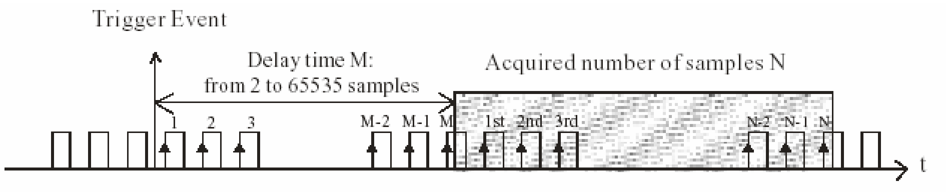


Fig. 4-2 Delay-Trigger Acquisition Mode

About Trigger Acquisition Mode

Use about-trigger acquisition mode when you want to acquire data both before and after a specific trigger event occurs. This operation is equivalent to doing both a pre-trigger and a post-trigger acquisition. When using software, please specify the following parameters, when using software in About-Trigger acquisition mode:

- Set to About-Trigger Acquisition Mode
- The sample clock source and sample rate
- The trigger source
- The total acquired sample number N
- The specific sample number M before the trigger event. The range of preset sample number is from 2 to 65536 samples.

In about-trigger mode, users must first designate the size of the allocated memory and the amount of samples to be snatched after the trigger event happens. The about-trigger acquisition starts when the first clock signal comes in. Once a trigger event happens, the on-going data acquisition will continue until the designated amount of samples have been reached. When the PCI-1714 detects the selected about trigger event, the card keeps acquiring the preset number of samples, and kept the total number of samples on the FIFO.

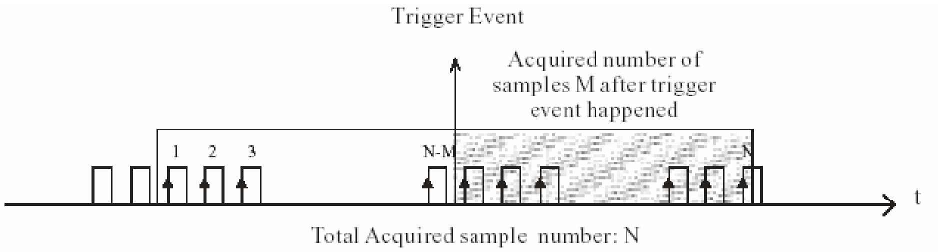


Fig. 4-3 About-Trigger Acquisition Mode

Pre-Trigger Acquisition Mode

Pre-Trigger mode is a particular application of about-trigger mode. Use pre-trigger acquisition mode when you want to acquire data before a specific trigger event occurs. Pre-trigger acquisition starts when you start the operation and stops when the trigger event happens. Then the specific number of samples will be reversed in the FIFO before the pre-trigger event occurred. Please specify the following parameters, when using software in Pre-trigger acquisition mode:

- Set to Pre-Trigger Acquisition Mode
- The sample clock source and sample rate
- The trigger source
- Assume the total acquired sample number is N, then set the total sample number to be N+2.

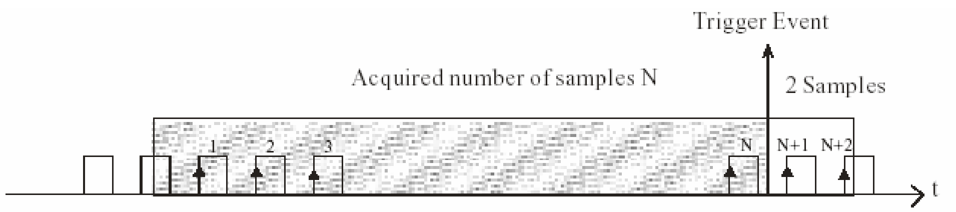


Fig. 4-4 Pre-Trigger Acquisition Mode

4.3 A/D Sample Clock Sources

The PCI-1714 can adopt both internal and external clock sources for pacer, post-trigger, delay-trigger, about-trigger acquisition modes:

- Internal A/D sample clock with 8-bit Counter
- External A/D sample clock that is connected to either the EXT-CLK0 (the differential clock source) or the EXT_CLK1 (the single ended clock source) on the ADAM-3909 screw terminal board.

The internal and both external A/D sample clocks are described in more details as the following.

Internal A/D Sample Clock

The internal A/D sample clock uses a 60 MHz time base. Conversions start on the rising edge of the counter output. You can use software to specify the clock source as internal and the sampling frequency to pace the operation. The minimum frequency is 234375 S/s, the maximum frequency is 30 MS/s. According to the sampling theory (Nyquist Theorem), you must specify a frequency that is at least twice as fast as the input's highest frequency component to achieve a valid sampling. For example, to accurately sample a 300 kHz signal, you have to specify sampling frequency of at least 600 kHz. This consideration can avoid an error condition often know as aliasing, in which high frequency input components appear erroneously as lower frequencies when sampling.

External A/D Sample Clock 0

The external sample clock 0 is a sine wave signal source which is converted to a TTL signal inside the PCI-1714. This signal is AC coupled. The input impedance of external clock 0 is 50 ohms and the input level is 2 volts peak-to-peak.

Please note that the frequency of the external clock is the system clock. The maximum A/D clock frequency is half of the system clock.

External A/D Sample Clock 1

The external sample clock 1 is a digital clock. The input impedance is 50 ohms and the input level should be 2.4V~5V into the 50-ohm load. This signal is DC coupled.

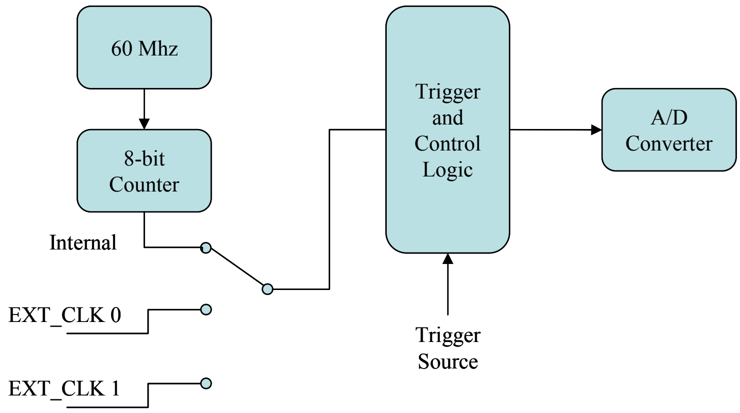


Fig 4-5 PCI-1714 Sample Clock Sources

4.4 Trigger Sources

The PCI-1714 supports the following trigger sources for post-, delay-, about- and pre-trigger acquisition modes:

- Software trigger,
- External digital (TTL) trigger, and
- Analog threshold trigger.

With PCI-1714, user can define the type of trigger source as rising-edge or falling-edge. These following sections describe these trigger sources in more detail.

Software Trigger

A software trigger event occurs when you start the analog input operation (the computer issues a write to the board to begin acquisitions). When you write the value to analog input trigger flag TRGF on Write BASE+Eh to produce either a rising-edge or falling-edge trigger, depending upon the trigger source type you choose. This edge will then act as an A/D trigger event. For detailed information, please refer to *Appendix C.7 Trigger Mode and Source*.

External Digital (TTL) Trigger

For analog input operations, an external digital trigger event occurs when the PCI-1714 detects either a rising or falling edge on the External A/D TTL trigger input signal from screw terminal EXT_TRIG on the ADAM-3909 screw terminal board. The trigger signal is TTL-compatible.

Analog Threshold Trigger

For analog input operations, an analog trigger event occurs when the PCI-1714 detects a transition from above a threshold level to below a threshold level (falling edge), or a transition from below a threshold level to above a threshold level (rising edge). User should connect the analog signals from the external device to one of the four BNC source connectors. Which one of the four sources is selected as the trigger source can be defined or identified by writing to or reading from the flags from TS0 to TS2 of Write/Read BASE+Eh.

On the PCI-1714, the analog trigger threshold voltage level is set using a dedicated 8-bit DAC; you can write or read the flags from AT0 to AT7 on Write/Read BASE+24h to define or identify the analog trigger threshold voltage level. Please also refer to the *Appendix C.14 Analog Trigger Threshold Voltage* for more details.

4.5 Analog Input Data Format

Table 4-2 Analog Input Data Format

A/D Code		Mapping Voltage
Hex.	Dec.	
000h	0d	-FS
7FFh	2047d	-1 LSB
800h	2048d	0
FFFh	2095d	FS-1 LSB
1LSB		FS/2048

Table 4-3 The corresponding Full Scale values for various Input Voltage Ranges

Gain	Range	FS
1	±5	5
2	±2.5	2.5
5	±1	1
10	±0.5	0.5

(This page is left blank for hard printing.)

5. Calibration

This chapter offers you a brief guide to the calibration procedure. The PCI-1714 has been well calibrated at the factory for initial use. Users are not necessary to calibrate the PCI-1714 in normal conditions. However, if some other conditions that the users have to calibrate the PCI-1714, then they can follow the procedure listed below to perform the necessary calibration.

To perform an effective calibration, the user has to prepare a standard 4-1/2 digits resolution, stable and low-noise DC voltage source. It is important that the accuracy of the device will depend on the accuracy of the DC source.

Calibration Procedure

Step 1: Click the ***Setup*** button on the *Advantech Device Manager* window (*Fig.5-1*) to launch the *PCI-1714 Device Setting* window (*Fig.5-2*).

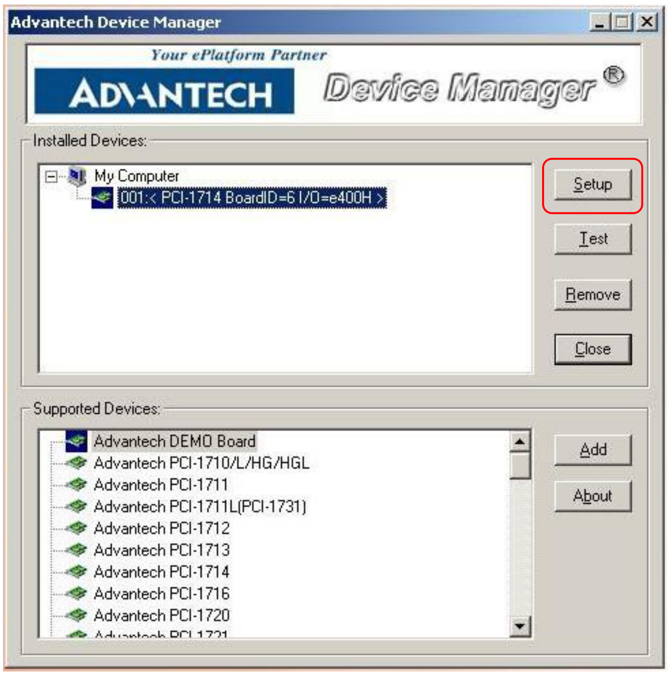


Fig.5-1 Click the Setup button to launch the Device Setting

Step 2: Select the input range of the channel which you want to calibrate.

Step 3: Click the ***Calibration*** button to start the calibration process. The *Calibration Wizard* window will pop up.

Note:

✎ Each calibration process can calibrate only one channel and one input range at a time.

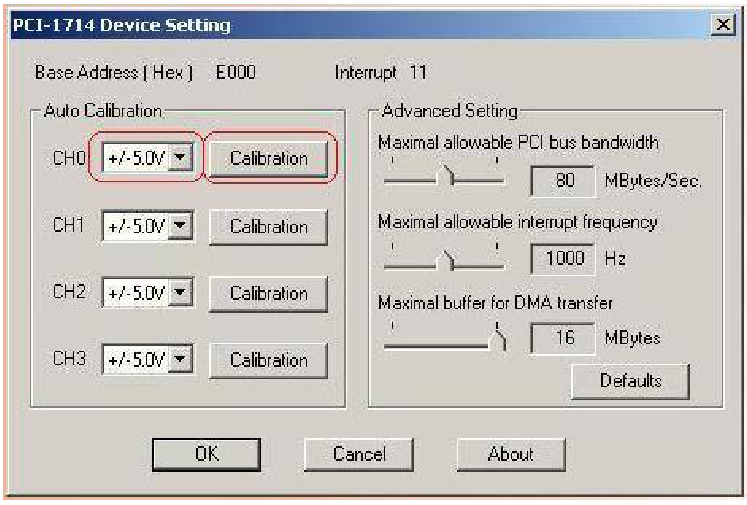


Fig.5-2 Click the Calibration button to launch the calibration

Step 4: Follow the instruction of *Calibration Wizard* to input a correct DC voltage as a reference and click the ***Next*** button to proceed to the next step.

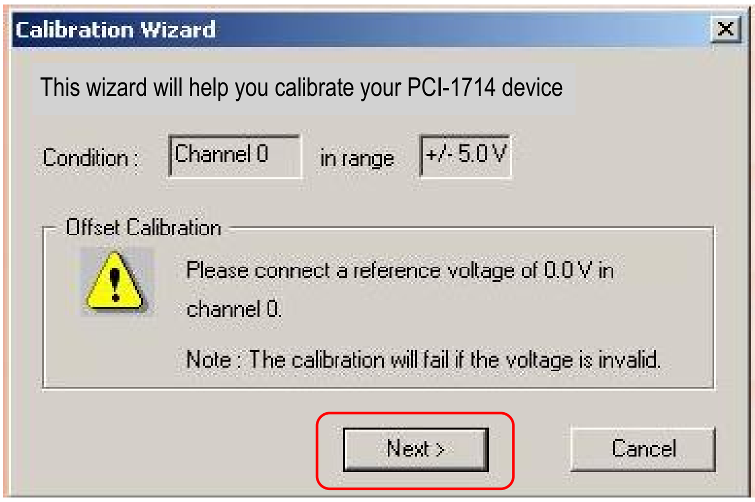


Fig.5-3 The start-up window of offset calibration

Step 5: Click the ***Start*** button to start the *Offset Calibration*. Note that the *Status* will indicate *Unknown* as default at the beginning.

Fig.5-4 The adjustment process of offset calibration

Step 6: If the reference DC voltage source and the wiring are both correct, the calibration will proceed automatically after the ***Start*** button is clicked. When the offset calibration is completed, the *Status* will indicate *Succeeded*, then click the ***Next*** button to proceed to the next step

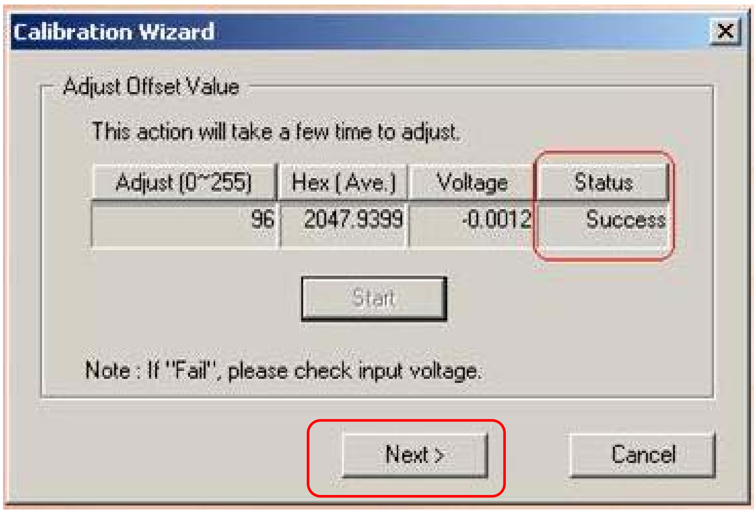


Fig.5-5 Offset calibration succeeded

Step 6a: Once the *Status* indicates *Failed*, please check if both the wiring and the input voltage are correct. When finished checking, click the ***Start*** button again to restart the procedure, or click the ***Cancel*** button to stop the calibration.

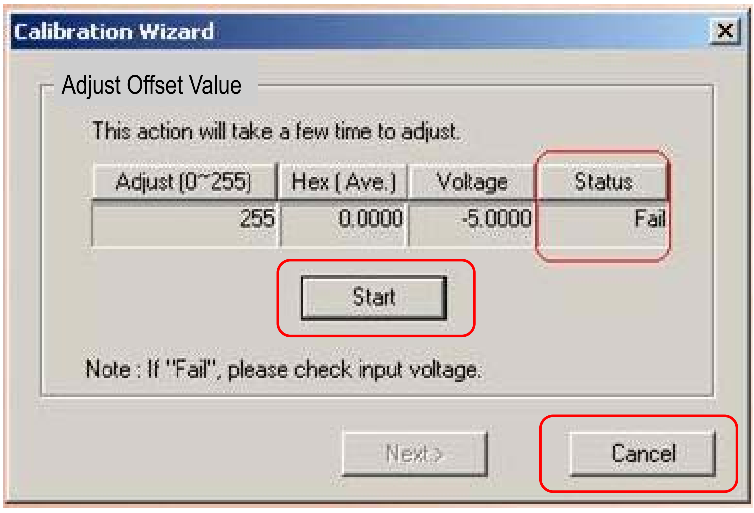


Fig.5-6 Offset calibration failed

Step 7: If the offset calibration is completed, then it is going to proceed to the *Gain Calibration*. The steps of gain calibration are quite similar to those of the offset calibration. Follow the instructions of the *Calibration Wizard* to input a correct DC voltage and click the ***Next*** button to proceed.,

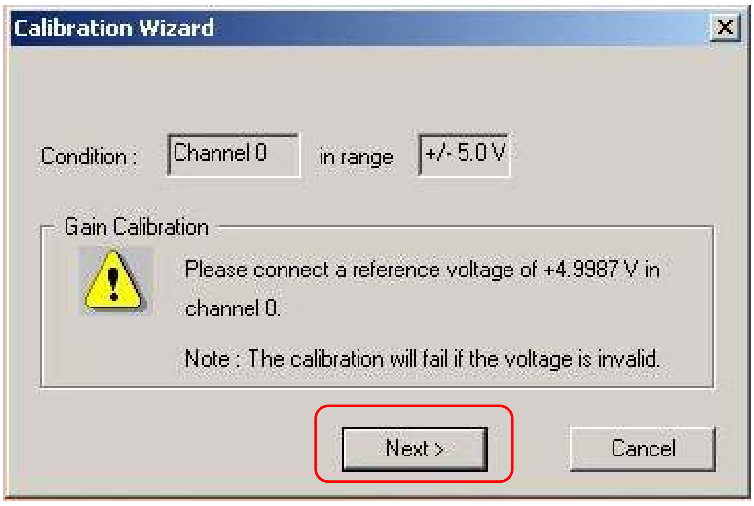


Fig.5-7 The start-up window of offset calibration

Step 8: Click the ***Start*** button to start gain calibration. Note that the *Status* will indicate *Unknown* as default at the beginning.

Fig.5-8 The adjustment process of gain calibration

Step 9: When the gain calibration is completed then click the ***Next*** button to proceed.

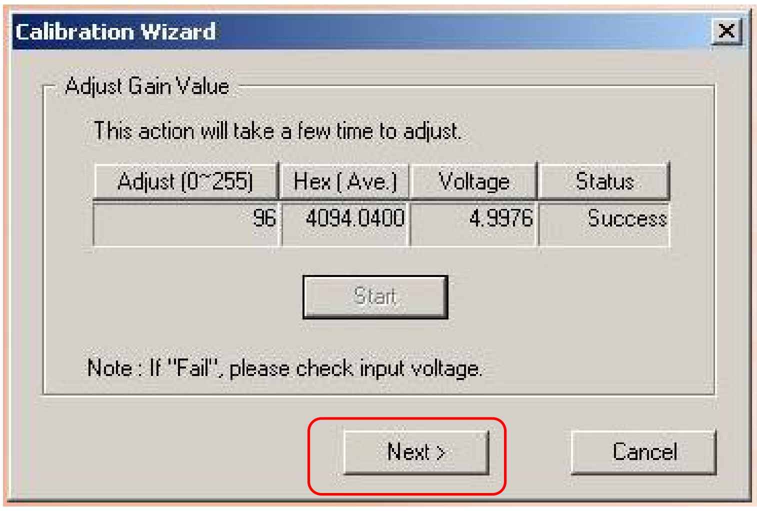


Fig.5-9 Gain calibration succeeded

Step 9a: Once the *Status* indicates *Failed*, please check if both the wiring and the input voltage are correct. When finished checking, click the ***Start*** button again to restart the procedure, or click the ***Cancel*** button to stop the calibration.

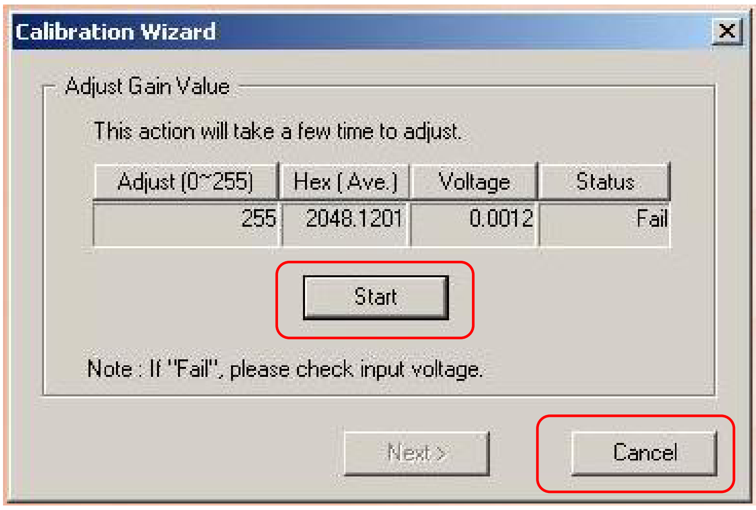


Fig.5-10 Gain calibration failed

Step 10: When the current channel is calibrated, click the ***Finish*** button to end the procedure. You can proceed to **Step 3** to select another channel for calibration, and repeat from **Step 4** to **Step 9**, until the rest of the channels are all calibrated one after one.

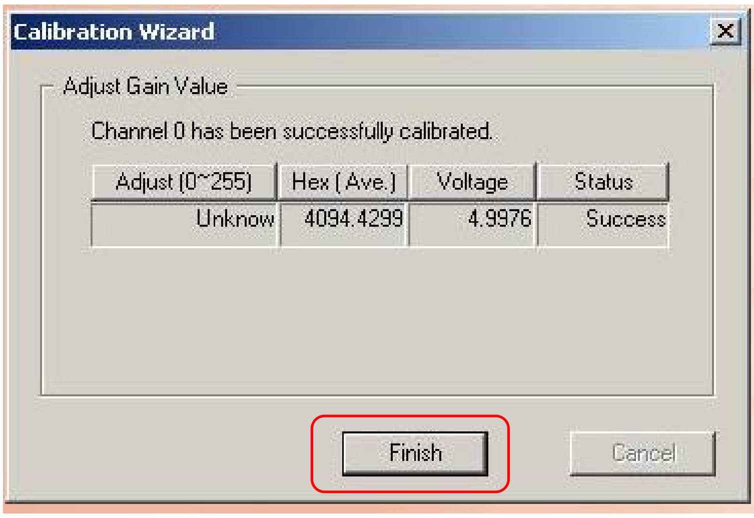


Fig.5-11 Calibration procedure completed

(This page is left blank for hard printing.)

Appendix A. Specifications

Analog Input:

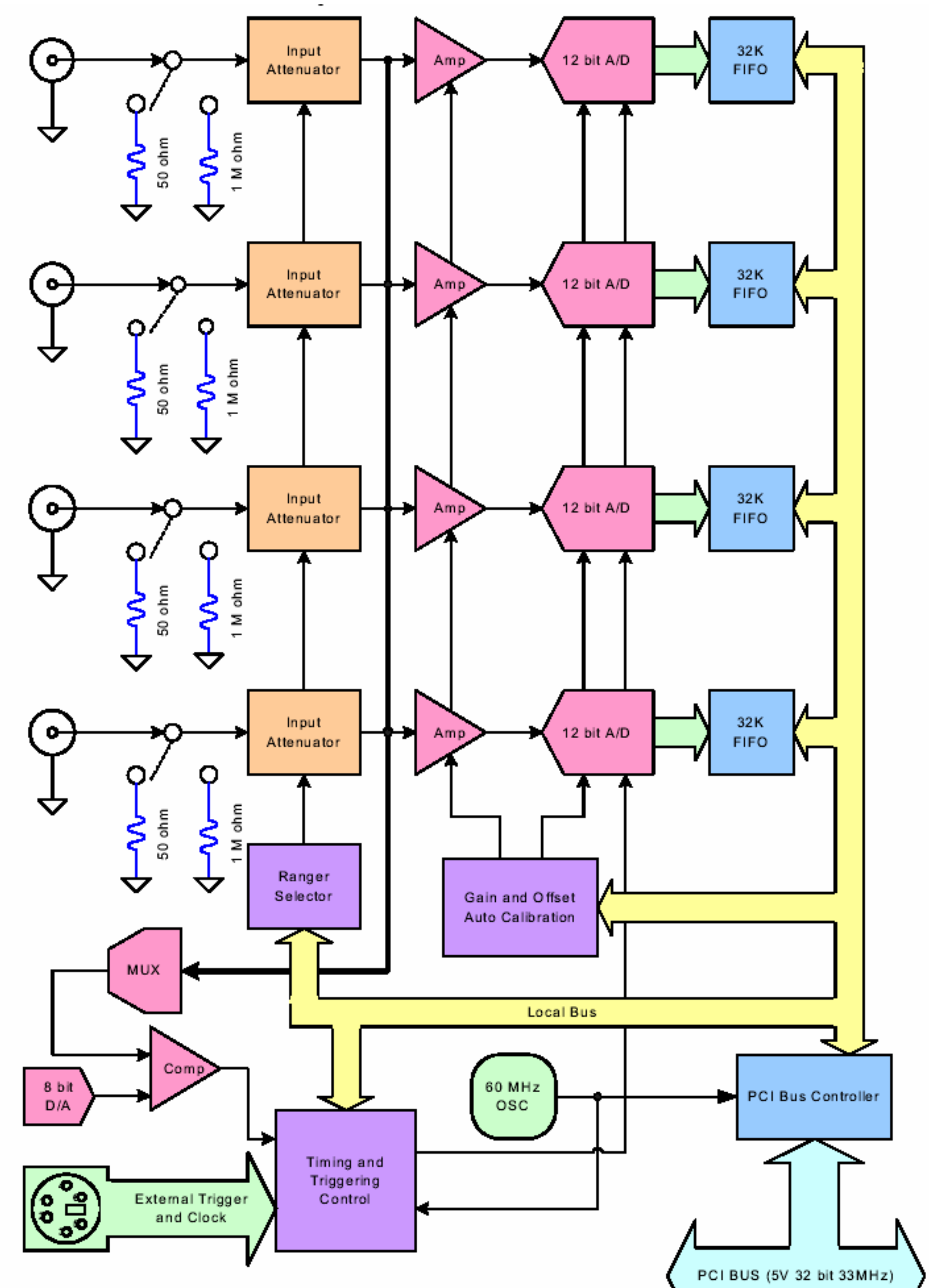
Channels	4 single-ended analog input channels				
Resolution	12-bit	FIFO Size		32K locations	
Max. Sampling Rate[1]	Up to 30MHz[1]				
Conversion Time	500 ns	Common mode voltage		±11 V max. (operational)	
Input range and Gain List	Gain	1	2	5	10
	Range	±5V	±2.5V	±1V	±0.5V
Drift	Gain	1	2	5	10
	Zero (μV/℃)	±30	±30	±30	±30
	Gain (ppm//℃)	±30	±30	±30	±30
Small Signal Bandwidth for PGA	Gain	1	2	5	10
	Bandwidth (-3dB)	7MHz	7MHz	7MHz	7MHz
Max. Input voltage	±15 V		Input Surge Protect	30 Vp-p	
Input Impedance	50Ω/1MΩ/∞ jumper selectable 100pF				
Trigger Mode	Software, pacer, post-trigger, pre-trigger, delay-trigger, about-trigger				
Accuracy	DC	DNLE: ±1LSB (No Missing Codes:12 Bits Guaranteed)			
		INLE: ±2LSB			
		Offset error	Adjustable to ±1LSB		
		Gain error	Adjustable to ±1LSB		
	AC	SINAD: S/(N+D): 68 dB ENOB: 11bits THD: -75 dB			
External TTL Trigger Input	Logic level	Low: 0.8 V max. High: 2.0V min.			
	Input impedance	50 ohms			
	Input coupled	DC			
External sin wave Trigger Input	Logic level	5.0V peak to peak			
	Input impedance	50 ohms			
	Input coupled	AC			
External Analog Trigger Input	Range	By analog input range			
	Resolution	8-bit			

[1] 30 MHz is only for FIFO depth (32K). Continuous acquisition depends on platform performance.

General:

I/O Connector Type	4 BNC connector for AI 1 PS2 connector for ext. clock and trigger	
Dimensions	137 mm x 107 mm (5.4" x 4.2")	
Power Consumption	Typical	+5 V @ 850 mA ; +12 V @ 600 mA
	Max.	+5 V @ 1 A ; +12 V @ 700m A
Temperature	Operation	0~+70℃(32~158℉)
	Storage	-20~+85℃(-4~185℉)
Relative Humidity	5~95%RH non-condensing (refer to *IEC 68-2-3*)	
Certification	CE certified	

Appendix B. Block Diagram



PCI-1714 Block Diagram

(This page is left blank for hard printing.)

Appendix C. Register Structure and Format

C.1 Overview

The PCI-1714 is delivered with an easy-to-use 32-bit DLL driver for user programming under the Windows 2000/95/98/NT/ME/XP operating system. We advise users to program the PCI-1714 using the 32-bit DLL driver provided by Advantech to avoid the complexity of low-level programming by register.

The most important consideration in programming the PCI-1714 at register level is to understand the function of the card's registers. The information in the following sections is provided only for users who would like to do their own low-level programming.

C.2 Register format

The register format is the basis to control the PCI-1714.

There are some rules for programmer’s reference:

1. All registers are 32-bit format. Please use the DWORD command in your own software.
2. Some registers are used only for write or read.
3. Some registers can support write and read back, they usually use the same name.
4. Some registers could write any value to complete a command.
5. In general, read only register is called status register, write only register is called control register.
6. Some registers are very similar, usually denote as a group. For example, A4, A3, A2, A1, A0 usually denote as A4: A0.
7. In this document, 1Fh means hexadecimal number 1F.

Table C-1 shows the function of each register of the PCI-1714 or driver and its address relative to the card's base address.

Table C-1 PCI-1714 register format (Part 1)

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0h	W	**AI Channel 0 Single Value Acquisition**															
	R	**AI Channel 0 Data**															
		TRG F	OV	G1	G0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
2h	W	**AI Channel 1 Single Value Acquisition**															
	R	**AI Channel 1 Data**															
		TRG F	OV	G1	G0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
4h	W	**AI Channel 2 Single Value Acquisition**															
	R	**AI Channel 2 Data**															
		TRG F	OV	G1	G0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
6h	W	**AI Channel 3 Single Value Acquisition**															
	R	**AI Channel 3 Data**															
		TRG F	OV	G1	G0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
8h	W	**AI Range Control Register**															
										CH3_G1	CH3_G0	CH2_G1	CH2_G0	CH1_G1	CH1_G0	CH0_G1	CH0_G0
	R																
										CH3_G1	CH3_G0	CH2_G1	CH2_G0	CH1_G1	CH1_G0	CH0_G1	CH0_G0
Ah	W	**A/D Converter Enable Register**															
														CH3	CH2	CH1	CH0
	R																
														CH3	CH2	CH1	CH0
Ch	W	**Clock Source and Divider Register**															
								CKS1	CKS0	DIV7	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
	R																
								CKS1	CKS0	DIV7	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
Eh	W	**Trigger Mode and Source Register**															
		TRG F	DMA _TCF							TSE	TS2	TS1	TS0		TM2	TM1	TM0
	R																
		TRG F	DMA _TCF							TSE	TS2	TS1	TS0		TM2	TM1	TM0

Table C-1 PCI-1714 register format (Part 2)

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
10h	W	**FIFO Control Register**															
								FRST 1	FCLR 1							FRST 0	FCLR 0
	R	**FIFO Status Register**															
				FIFO 1_AF	FIFO 1_AE		FIFO 1_FF	FIFO 1_HF	FIFO 1_EF			FIFO 0_AF	FIFO 0_AE		FIFO 0_FF	FIFO 0_HF	FIFO 0_EF
12h	W	**FIFO Control Register**															
								FRST 3	FCLR 3							FRST 2	FCLR 2
	R	**FIFO Status Register**															
				FIFO 3_AF	FIFO 3_AE		FIFO 3_FF	FIFO 3_HF	FIFO 3_EF			FIFO 2_AF	FIFO 2_AE		FIFO 2_FF	FIFO 2_HF	FIFO 2_EF
14h	W	**FIFO 0 Programmable Flag Register**															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
	R	**FIFO 0 Programmable Flag Register**															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
16h	W	**FIFO 1 Programmable Flag Register**															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
	R	**FIFO 1 Programmable Flag Register**															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
18h	W	**FIFO 2 Programmable Flag Register**															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
	R	**FIFO 2 Programmable Flag Register**															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
1Ah	W	**FIFO 3 Programmable Flag Register**															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
	R	**FIFO 3 Programmable Flag Register**															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
1Ch	W	**DMA Counter Register**															
		CN15	CN14	CN13	CN12	CN11	CN10	CN9	CN8	CN7	CN6	CN5	CN4	CN3	CN2	CN1	CN0
	R																
		CN15	CN14	CN13	CN12	CN11	CN10	CN9	CN8	CN7	CN6	CN5	CN4	CN3	CN2	CN1	CN0
1Eh	W	**Rest DMA Counter**															
	R																

Table C-1 PCI-1714 register format (Part 3)

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
20h	W	**Interrupt Control Register**															
		INTE							DMA_TC	FIFO3_AF	FIFO3_HF	FIFO2_AF	FIFO2_HF	FIFO1_AF	FIFO1_HF	FIFO0_AF	FIFO0_HF
	R																
		INTF							INTF8	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	INTF0
22h	W	**Clear Interrupt**															
	R	**N/A**															
24h	W	**Analog Trigger Threshold voltage Register**															
										AT7	AT6	AT5	AT4	AT3	AT2	AT1	AT0
	R	**Analog Trigger Threshold voltage Register**															
										AT7	AT6	AT5	AT4	AT3	AT2	AT1	AT0
26h	W	**N/A**															
	R	**N/A**															
28h	W	**Calibration Command Register**															
				CG1	CG0	X	CM2	CM1	CM0	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0
	R																
				CG1	CG0	CBUSY	CM2	CM1	CM0	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0
2Ah	W																
	R																
2Ch	W	**Board ID**															
	R																
														BID3	BID2	BID1	BID0
2Eh	W																
	R																

Table C-1 PCI-1714 register format (Part 4)

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
30h	W	**Reset start read channel to CH0**															
	R	**AD Channel n DATA**															
		TRGF	OV	G1	G0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
32h	W	**N/A**															
	R	**AD Channel n+1 DATA**															
		TRGF	OV	G1	G0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
34h	W	**DMA Request selector**															
																	DS0
	R																

C.3 A/D Single Value Acquisition- Write BASE+0, 2, 4, 6

In single value acquisition mode (SW trigger), the A/D converter will convert one sample when you write to the register Write **BASE+0, 2, 4, 6** with any value. User can check the A/D FIFO status **(FIFOn_FE)** to make sure if the data is ready to be received.

Table C-2 Register for Single Value Acquisition

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0h	W	**AI Channel 0 Single Value Acquisition**															
	R	**AI Channel 0 Data**															
		TRGF	OV	G1	G0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
2h	W	**AI Channel 1 Single Value Acquisition**															
	R	**AI Channel 1 Data**															
		TRGF	OV	G1	G0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
4h	W	**AI Channel 2 Single Value Acquisition**															
	R	**AI Channel 2 Data**															
		TRGF	OV	G1	G0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
6h	W	**AI Channel 3 Single Value Acquisition**															
	R	**AI Channel 3 Data**															
		TRGF	OV	G1	G0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

AD11: AD0 12 bits Data of A/D Conversion

AD0 the least significant bit (LSB) of A/D data.

AD11 the most significant bit (MSB) of A/D data.

G1: G0 Range code

These 2 bits indicate the input range of the data.

G1	G0	Input range
0	0	-5 to +5V
0	1	-2.5 to +2.5V
1	0	-1 to +1V
1	1	-0.5 to +0.5V

OV **Over range flag**

This bit indicates whether the input voltage is over range or not. Read 1 means over range.

TRGF **Trigger Flag (For about trigger use only)**

The trigger flag indicates whether a trigger event has happened during A/D conversion process.

C.4 AI Range Control- Write/Read BASE+8

Table C-3 Register for Analog Input Range Control

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8h	W	AI Range Control Register															
										CH3_G1	CH3_G0	CH2_G1	CH2_G0	CH1_G1	CH1_G0	CH0_G1	CH0_G0
	R																
										CH3_G1	CH3_G0	CH2_G1	CH2_G0	CH1_G1	CH1_G0	CH0_G1	CH0_G0

Analog Input Range Selector

These registers are used to select the analog input range for each channel.

CHn_G1	CHn_G0	Input range
0	0	-5 to +5 V
0	1	-2.5 to +2.5 V
1	0	-1 to +1 V
1	1	-0.5 to +0.5 V

(n = 0~3)

C.5 A/D Converter Enable- Write/Read BASE+A

Table C-4 Register for A/D Converter Enable

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ah	W	A/D Converter Enable Register															
														CH3	CH2	CH1	CH0
	R																
														CH3	CH2	CH1	CH0

CH3, CH2, CH1, CH0 A/D converter Enable bit

These bits are control the A/D converter working. Write 0 will disable the A/D, 1 enable. They could be read back for check.

C.6 Clock Source and Divider- Write/Read BASE+C

Table C-5 Register for Clock Source and Divider

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ch	W	Clock Source and Divider Register															
								CKS1	CKS0	DIV7	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
	R																
								CKS1	CKS0	DIV7	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0

DIV7: DIV0 Clock Divider

When select the internal clock source (60MHz) the clock will pre-divide by the clock divider. The divider is 8-bit wide, so it could divide from 2 to 256.

DIV7: DIV0	Divide value
00h	N/A
01h	divide by 2
02h	divide by 3
.	.
FEh	divide by 255
FFh	divide by 256

CKS1: CKS0 Clock Source selector

These 2 bits select the clock source feed to the A/D converters.

CKS1	CKS0	Clock source
0	0	Internal clock 60MHz
0	1	External clock 0
1	0	External clock 1
1	1	Off

C.7 Trigger Mode and Source- Write/Read BASE+E

Table C-6 Register for Trigger Mode and Source

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Eh	W	Trigger Mode and Source Register															
		TRG F	DMA _TCF							TSE	TS2	TS1	TS0		TM2	TM1	TM0
	R																
		TRG F	DMA _TCF							TSE	TS2	TS1	TS0		TM2	TM1	TM0

TM2: TM0 Trigger Mode selector

There are 5 trigger modes for PCI-1714. Please refer to the operation theorem for more information.

TM2	TM1	TM0	Meaning
0	0	0	Single value acquisition mode (SW trigger)
0	0	1	Pacer acquisition mode
0	1	0	Post-trigger acquisition mode
0	1	1	Delay-trigger acquisition mode
1	0	0	About-trigger acquisition mode
1	0	1	N/A
1	1	0	N/A
1	1	1	N/A

TS2: TS0 **Trigger Source selector**

TS2	TS1	TS0	Meaning
0	0	0	Analog input CH0
0	0	1	Analog input CH1
0	1	0	Analog input CH2
0	1	1	Analog input CH3
1	0	0	Digital trigger input
1	0	1	N/A
1	1	0	N/A
1	1	1	N/A

TSE **Trigger Edge selector:**

0 Rising edge trigger

1 Falling edge trigger

DMA_TCF **DMA counter terminal count flag**

0 DMA counter is not terminal count

1 DMA counter is terminal count

TRGF **Trigger flag**

0 Trigger not occurred

1 Trigger occurred

C.8 FIFO Control- Write BASE+10,12

Table C-7 Register for FIFO Control

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
10h	W	FIFO Control Register															
								FRST 1	FCLR 1							FRST 0	FCLR 0
12h	W	FIFO Control Register															
								FRST 3	FCLR 3							FRST 2	FCLR 2

FCLRn (n = 0~3) **FIFO Clear register**

Write 1 to this bit to clear FIFO data.

FRSTn (n = 0~3) **FIFO Reset register**

Write 1 to this bit to clear FIFO data and reset the AE and AF flag position to 7FH.

C.9 FIFO Status- Read BASE+10,12

Table C-8 Register for FIFO Status

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
10h	R	FIFO Status Register															
				FIFO 1_AF	FIFO 1_AE		FIFO 1_FF	FIFO 1_HF	FIFO 1_EF			FIFO 0_AF	FIFO 0_AE		FIFO 0_FF	FIFO 0_HF	FIFO 0_EF
12h	R	FIFO Status Register															
				FIFO 3_AF	FIFO 3_AE		FIFO 3_FF	FIFO 3_HF	FIFO 3_EF			FIFO 2_AF	FIFO 2_AE		FIFO 2_FF	FIFO 2_HF	FIFO 2_EF

FIFOn_EF (n = 0~3) **FIFO Empty Flag**

1 FIFO is empty

0 FIFO is not empty

FIFOn_HF (n = 0~3) **FIFO Half full Flag**

1 FIFO is half full

0 FIFO is not half full

FIFOn_FF (n = 0~3) **FIFO Full Flag**

1 FIFO is full

0 FIFO is not full

FIFOn_AE (n = 0~3) **FIFO Almost Empty flag**

1 FIFO is almost empty

0 FIFO is not almost empty

FIFOn_AF (n = 0~3) **FIFO Almost Full flag**

1 FIFO is almost full

0 FIFO is not almost full

C.10 FIFO for Programmable Flag-Write/Read BASE+14,16,18,1A

Table C-9 Register for FIFO Programmable Flag

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
14h	W	FIFO 0 Programmable Flag Register															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
	R	FIFO 0 Programmable Flag Register															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
16h	W	FIFO 1 Programmable Flag Register															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
	R	FIFO 1 Programmable Flag Register															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
18h	W	FIFO 2 Programmable Flag Register															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
	R	FIFO 2 Programmable Flag Register															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
1Ah	W	FIFO 3 Programmable Flag Register															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
	R	FIFO 3 Programmable Flag Register															
			PF14	PF13	FP12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0

PF14: PF0 **FIFO n Programmable Flag Register (n = 0 ~3)**

The FIFO on PCI-1714 is very powerful. It allow user to define the indicate flag in any depth. There are two flags could be defined: **FIFO Almost Empty flag** and **FIFO Almost Full flag.** To define these flags must follow the procedure:

First write is the Almost Empty flag offset count from the empty.

Second write is the Almost Full flag offset count from the full. Read procedure is the same as write. Once set the offset, the value will keep until FIFO reset.

C.11 DMA Counter- Write/Read BASE+1C, Write BASE+1E

Table C-10 Register for DMA Counter

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1Ch	W	DMA Counter Register															
		CN15	CN14	CN13	CN12	CN11	CN10	CN9	CN8	CN7	CN6	CN5	CN4	CN3	CN2	CN1	CN0
	R																
		CN15	CN14	CN13	CN12	CN11	CN10	CN9	CN8	CN7	CN6	CN5	CN4	CN3	CN2	CN1	CN0
1Eh	W	Rest DMA Counter															

CN15: CN0 DMA counter register:

DMA counter is a16-bit counter designed for **ABOUT** and **DELAY** trigger mode only. Set the counter value for about trigger data counts after the trigger event. Also the value for delay trigger data counts after the trigger event.

Rest DMA Counter

Before start the DMA counter, write the **BASE + 1Eh** to reset the DMA counter.

C.12 Interrupt Control/Flag- Write/Read BASE+20

Table C-11 Register for Interrupt Control/Flag

<table>
<tr><th colspan="2" rowspan="2">Base Address + HEX</th><th colspan="16">PCI-1714 Register Format</th></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td rowspan="4">20h</td><td rowspan="2">W</td><td colspan="16">Interrupt Control Register</td></tr>
<tr><td>INTE</td><td></td><td></td><td></td><td></td><td></td><td></td><td>DMA _TC</td><td>FIFO 3_AF</td><td>FIFO 3_HF</td><td>FIFO 2_AF</td><td>FIFO 2_HF</td><td>FIFO 1_AF</td><td>FIFO 1_HF</td><td>FIFO 0_AF</td><td>FIFO 0_HF</td></tr>
<tr><td rowspan="2">R</td><td colspan="16">Interrupt Flag</td></tr>
<tr><td>INTF</td><td></td><td></td><td></td><td></td><td></td><td></td><td>INTF 8</td><td>INTF 7</td><td>INTF 6</td><td>INTF 5</td><td>INTF 4</td><td>INTF 3</td><td>INTF 2</td><td>INTF 1</td><td>INTF 0</td></tr>
</table>

Interrupt Control Register

PCI-1714 provides 9 sources to generate the interrupt. Write 1 to enable the interrupt, write 0 to disable. The INTE is control the total interrupt.

FIFO0_HF FIFO 0 Half Full
FIFO0_AF FIFO 0 Almost Full
FIFO1_HF FIFO 1 Half Full
FIFO1_AF FIFO 1 Almost Full
FIFO2_HF FIFO 2 Half Full
FIFO2_AF FIFO 2 Almost Full
FIFO3_HF FIFO 3 Half Full
FIFO3_AF FIFO 3 Almost Full
DMA_TC DMA counter Terminal Count
INTE Total Interrupt Enable

Interrupt Flag

These bits correspond to the same bit number of the interrupt control register to indicate which interrupt occurred. Read 1 means interrupt occurred.

INTF0 FIFO 0 Half Full interrupt flag
INTF1 FIFO 0 Almost Full interrupt flag
INTF2 FIFO 1 Half Full interrupt flag
INTF3 FIFO 1 Almost Full interrupt flag
INTF4 FIFO 2 Half Full interrupt flag
INTF5 FIFO 2 Almost Full interrupt flag

INTF6 FIFO 3 Half Full interrupt flag

INTF7 FIFO 3 Almost Full interrupt flag

INTF8 DMA counter Terminal Count interrupt flag

INTF Total Interrupt flag

C.13 Clear Interrupt- Write BASE+22

Table C-12 Register for Clear Interrupt

Base Address + HEX		**PCI-1714 Register Format**															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
22h	W	**Clear Interrupt**															

Clear Interrupt

Write any value to this address will clear interrupt. It will clear all flags to 0 if there is no any interrupt in coming.

C.14 Analog Trigger Threshold Voltage- Write/Read BASE+24

Table C-13 Register for Analog Trigger Threshold Voltage

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24h	W	Analog Trigger Threshold voltage Register															
										AT7	AT6	AT5	AT4	AT3	AT2	AT1	AT0
	R	Analog Trigger Threshold voltage Register															
										AT7	AT6	AT5	AT4	AT3	AT2	AT1	AT0

AT7: AT0 Analog Trigger Threshold voltage Register

These registers set the analog trigger threshold voltage level.

AT7: AT0	±0.5V	±1V	±2.5V	±5V
FFh	0.496	0.992	2.48	4.96
FEh	0.492	0.984	2.46	4.92
.	.	.	.	.
81h	0.004	0.008	0.02	0.04
80h	0	0	0	0
79h	-0.004	-0.008	-0.02	-0.04
.	.	.	.	.
01h	-0.496	-0.992	-2.48	-4.96
00h	-0.5	-1	-2.5	-5

C.15 Calibration Command- Write/Read BASE+28

Table C-14 Register for Calibration Command

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
28h	W	Calibration Command Register															
				CG1	CG0	X	CM2	CM1	CM0	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0
	R																
				CG1	CG0	CBUSY	CM2	CM1	CM0	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0

CD7: CD0 Calibration data

The value is form 00h to FFh.

CM2: CM0 Calibration Command Register

CM2	CM1	CM0	Meaning
0	0	0	Analog input CH0 offset adjustment
0	0	1	Analog input CH0 gain adjustment
0	1	0	Analog input CH1 offset adjustment
0	1	1	Analog input CH1 gain adjustment
1	0	0	Analog input CH2 offset adjustment
1	0	1	Analog input CH2 gain adjustment
1	1	0	Analog input CH3 offset adjustment
1	1	1	Analog input CH4 gain adjustment

G1: G0 Calibration range code

G1	G0	Input range
0	0	-5 to +5 V
0	1	-2.5 to +2.5 V
1	0	-1 to +1 V
1	1	-0.5 to +0.5 V

CBUSY Calibration command busy flag

This bit indicates the calibration command is complete and ready for next command input.

C.16 Board ID- Read BASE+2C

Table C-15 Register for Board ID

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2Ch	R	Board ID															
														BID3	BID2	BID1	BID0

BID3: BID0 Board ID

Board ID selector value is from 0 to 15. Please refer to board ID switch setting.

C.17 Reset DMA Start Channel to CH0- Write BASE+30

Table C-16 Register for Reset DMA Start Channel to CH0

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
30h	W	Reset DMA start channel to CH0															

Reset DMA start channel to CH0

Write any value to **BASE+30h** to reset DMA transfer data from CH0. Before start DMA transfer, user has to reset the start channel to CH0. This only for four channels DMA data transfer.

C.18 AD Channel n DATA- Read BASE+30,32

Table C-17 Register for AD Channel n DATA

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
30h	R	*AD Channel n DATA*															
		TRGF	OV	G1	G0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
32h	R	AD Channel n+1 DATA															
		TRGF	OV	G1	G0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

AD Channel n DATA (n = 0 or 2)

BASE + 30, 32 are for four channels DMA data transfer. Data transfer will alternate from CH0 + CH1 to CH2 + CH3 automatically. The format is the same as **BASE + 0, 2 or BASE + 4, 6**. CH0 + CH1 is first 32-bit, CH2 + CH3 is the second and CH0 + CH1, …and so on. User only want to transfer CH0 + CH1, please use **BASE + 0, 2**, transfer CH2 + CH3, please use **BASE + 4, 6**. About DMA data transfer, please refer to PCI9056 datasheet.

DMA data transfer support 1, 2 or 4 channels data acquisition.
For 1 channel data acquisition, only channel 0 or 2 is acceptable. For 2 channels data acquisition, only channel 0,1 or 2,3 is acceptable.

The DMA data transfer to memory format are list as below:

1. One channel CH0:

Memory Address	D31 D16	D15 D0
N	CH0 data 1	CH0 data 0
N+1	CH0 data 3	CH0 data 2
N+2	CH0 data 5	CH0 data 4
N+3	CH0 data 7	CH0 data 6
:	:	:

2. Two channel CH0 + CH1

Memory Address	D31 D16	D15 D0
N	CH1 data 0	CH0 data 0
N+1	CH1 data 1	CH0 data 1
N+2	CH1 data 2	CH0 data 2
N+3	CH1 data 3	CH0 data 3
:	:	:

3. Four channel CH0 + CH1 + CH2 + CH3

Memory Address	D31 D16	D15 D0
N	CH1 data 0	CH0 data 0
N+1	CH3 data 0	CH2 data 0
N+2	CH1 data 1	CH0 data 1
N+3	CH3 data 1	CH2 data 1
:	:	:

C.19 DMA Request Selector- Write BASE+34

Table C-18 Register for DMA Request Selector

Base Address + HEX		PCI-1714 Register Format															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
34h	W	DMA Request selector															
																	DS0

DS0 **DMA Request selector**

This bit select the DMA request (hardware signal DREQ), user could use FIFO 0 flag or FIFO 2 flag to generate DREQ.

0 FIFO 0 flag

1 FIFO 2 flag