PCI-82x Series Cards
User Manual

Multifunction Boards Version 1.4, Nov. 2015

SUPPORTS

This manual relates to the following Boards: PCI-822LU and PCI-826LU.

WARRANTY

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

WARNING

ICP DAS assumes no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

COPYRIGHT

TRADEMARK

Names are used for identification purposes only and may be registered trademarks of their respective companies.

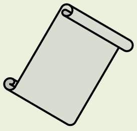
CONTACT US

If you have any questions, feel tree to contact us by email at:
service@icpdas.com, service.icpdas@gmail.com
We will respond to you within 2 working days.

TABLE OF CONTENTS

Packing List

The shipping package includes the following items:

	One of the following multi-function cards: PCI-822LU PCI-826LU	*Note:* *If any of these items is missing or damaged, contact the dealer from whom you purchased the product. Save the shipping materials and carton in case you need to ship or store the product in the future.*
	One CA-4002 D-Sub connector	
	One printed Quick Start Guide	
	One software utility CD	

Related Information

Product Page for the PCI-826LU:

http://www.icpdas.com/root/product/solutions/pc_based_io_board/pci/pci-826lu.html

Product Page for the PCI-822LU:

http://www.icpdas.com/root/product/solutions/pc_based_io_board/pci/pci-822lu.html

Hardware Manual, Datasheet and Quick Start Guide:

More information related to PCI-822/826 Series cards can be found in the \NAPDOS\PCI\PCI-82x\Manual\ folder on the companion CD, or can be downloaded from:
http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pci-82x/manual/

Documentation and Software for the UniDAQ SDK:

More information related to UniDAQ SDK can be found in the \NAPDOS\PCI\UniDAQ\ folder on the companion CD, or can be downloaded from:
http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/

1. Introduction

The PCI-822/826 Series cards provide 32/16-channel 12-bit (16-bit) Analog Input with a sampling rate of up to 250 kSamples/second, and 2-channel 16-bit Analog Output with 32-channel programmable Digital Input/Output.

1.1 Overview

The PCI-822/826 Series are high-performance multifunction cards that provide high-speed Analog and Digital I/O functions. PCI-822/826 Series cards use a Universal PCI interface that supports both the 3.3 V and the 5 V PCI bus. These cards feature a continuous, 250 kSamples/second, 12-bit (16-bit) resolution A/D converter, an 8 k-Sample hardware FIFO, a 2-channel, 16-bit D/A converter, and a 32-channel programmable Digital Input/Output and Digital Output readback. The PCI-822/826 LU provides either 32-channel single-ended or 16-channel differential Analog Input that is jumper selectable, while the PCI-826LU is equipped with a high-speed PGA (Programmable Gain Amplifier) featuring programmable gain controls (1, 2, 4 and 8).

PCI-822/826 Series cards contain an embedded Card ID switch and onboard pull-high/pull-low resistors for the Digital Input, meaning that a unique ID number can be set for each card so that can each card can be individually recognized when two or more PCI-822/826 Series cards are used in a computer at the same time. The pull-high/pull-low resistors allow the status of the Digital Input to be specified, meaning that when the Digital Input channels are disconnected, the status of the Digital Input will remain as either high or low and will not be set to floating.

PCI-822/826 Series cards provide both a programmable software trigger and a pacer trigger, and include an A/D channel scan function called MagicScan. The MagicScan controller eliminates the majority of the effort required to acquire the A/D values, such as selecting the channel, setting the Gain values and the settling time, triggering the ADC, and acquiring the data. Using the built-in MagicScan and the interrupt features, these complex tasks are effectively offloaded from the CPU. Even in channel scan mode, a different Gain code can be used for each channel, and the sampling rate can still achieve a total of 250 kS/s, making the PCI-822/826 Series well-suited to the demands of high-end applications.

1.2 Features

The following is an overview of the general features that apply to both the PCI-822 and PCI-826 Series cards. See Section 1.3 for more detailed specifications.

1. General

- Bus : Universal PCI

2. Analog Input (A/D)

- One 12-bit A/D converter with a maximum sample rate of 250 kSsamples/second (PCI-822LU)
- One 16-bit A/D converter with a maximum sample rate of 250 kSamples/second (PCI-826LU)
- 32 single-ended or 16 differential programmable input channels (PCI-822LU/PCI-826LU)
- Three different A/D trigger methods
- Three different external trigger methods
- Programmable Gain control
- Programmable offset control

3. Analog Output (D/A)

- One D/A converter
- 2-channel 16-bit voltage output
- Voltage Output Range: ± 10 V, ± 5 V, 0 to +10 V, 0 to +5 V

4. Digital Input/Output (DI/O)

- 32-bit programmable DI/O
- High-speed data transfer rate
- Pull-high/low function on Digital Input
- Digital Output readback function

1.3 Specifications

Model	PCI-822LU	PCI-826LU
Analog Input		
Channels	32 single-ended/16 differential	
A/D Converter	12-bit, 8 µs conversion time	16-bit, 8 µs conversion time
Sampling Rate	250 kS/s (Max.)	
FIFO Size	8192 samples	
Over voltage Protection	Continuous ± 35 V_{p-p}	
Input Impedance	10,000 MΩ/4pF	
Trigger Modes	Software, Pacer	
Data Transfer	Polling, Interrupt	
Accuracy	0.1 % of FSR ± 1 LSB @ 25°C, ± 10 V	0.05 % of FSR ± 1 LSB @ 25 °C, ± 10 V
Input Range	Gain: 1, 2, 4, 8 Bipolar (V): ± 10, ± 5, ± 2.5, ± 1.25	
Zero Drift	15 ppm/°C of FSR	
Analog Output		
Channels	2	
Resolution	16-bit	
Accuracy	± 6 LSB	
Output Range	-5 V ~ +5 V , -10 V ~ +10 V 0 ~ +10 V, 0 ~ +5 V	
Output Driving	± 5 mA	
Slew Rate	8.33 V/µs	
Output Impedance	0.1 Ω Max.	
Operating Mode	Software	
Programmable I/O		
Channels	32	
Digital Input		
Compatibility	5 V/TTL	
Input Voltage	Logic 0: 0.8 V (Max.) Logic 1: 2.0 V (Min.)	
Pull High/Low	Yes	
Response Speed	1.0 MHz (Typical)	

Digital Output	
Compatibility	5 V/TTL
Output Voltage	Logic 0: 0.4 V (Max.) Logic 1: 2.4 V (Min.)
Output Capability	Sink: 0.8 mA @ 0.8 V Source: -2.4 mA @ 2.0 V
DO Readback	Yes
Response Speed	1.0 MHz (Typical)
General	
Bus Type	3.3 V/5 V Universal PCI, 32-bit
Data Bus	16-bit
Card ID	Yes (4-bit)
I/O Connector	Female DB37 x 1 20-pin box header x 2
Dimensions (L x W x D)	169 mm x 105 mm x 22 mm
Power Consumption	1 A @ +5 V max.
Operating Temperature	0 ~ 60 °C
Storage Temperature	-20 ~ 70 °C
Humidity	5 ~ 85% RH, Non-condensing

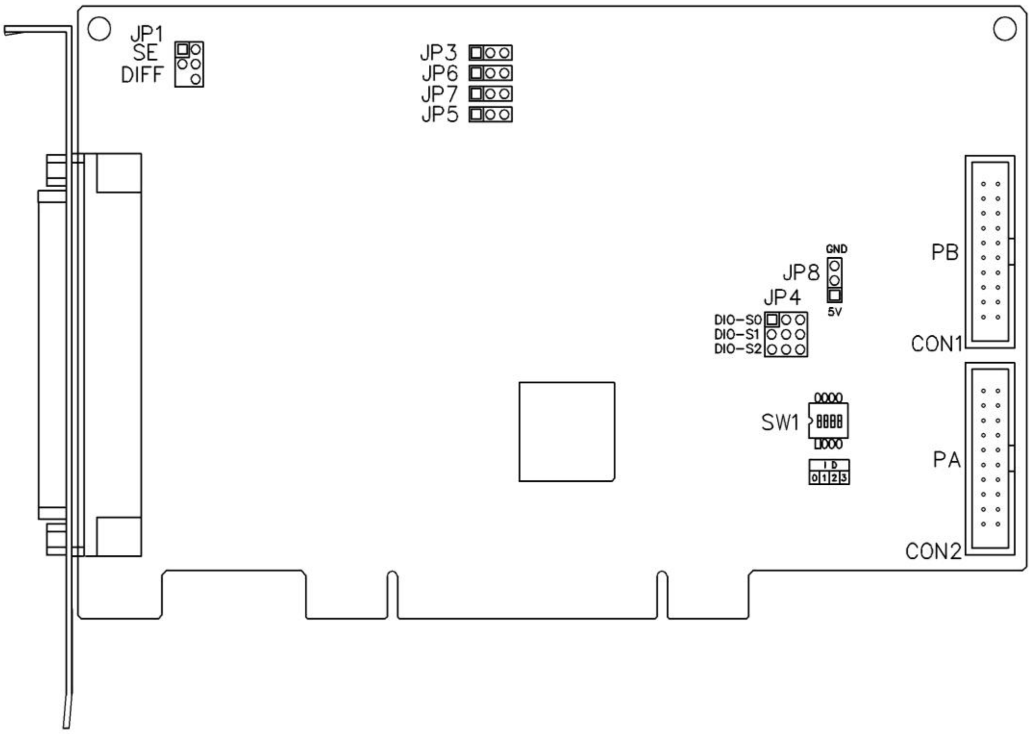
1.4 Applications

- Signal Analysis
- FFT and Frequency Analysis
- Transient Analysis
- Temperature Monitoring
- Vibration Analysis
- Energy Management
- Other industrial and laboratory measurements and control

2 Hardware Configuration

This section provides details of the hardware configuration for the PCI-822/826 Series cards, including the layout of the individual cards, the Jumper positions, and the pin assignments, etc.

2.1 Layout



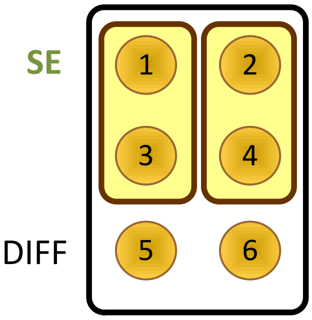
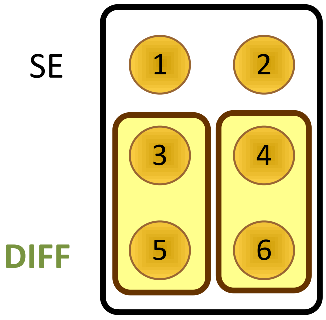
Jumper	Purpose	Comment
JP1	Analog Input Type Configuration	Refer to Section 2.2.1
JP6, JP7	Analog Output Range Configuration	Refer to Section 2.2.2
JP3, JP5	Analog Output Type Configuration	Refer to Section 2.2.3
JP4	Digital Input/Output Mode Configuration	Refer to Section 2.2.4
JP8	Digital Input Pull-high/low Configuration	Refer to Section 2.2.5
SW1	Card ID Configuration	Refer to Section 2.3
CON1	The terminal for Port B	Refer to Section 2.6 for more detailed information about the pin assignments.
CON2	The terminal for Port A	
CON3	The terminal for Analog Input/Output	

2.2 Jumper Settings

PCI-822/826 Series cards include a number of jumpers that can be used to configure features such as the Analog Input/Output Type, the Analog Output Range, and the Digital Input/Output mode, etc. each of which is described in more detail below.

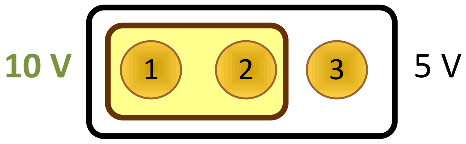
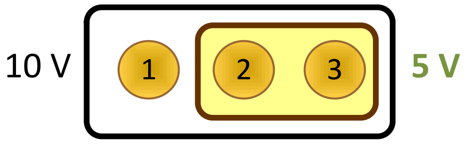
2.2.1 JP1 (Analog Input Type)

Jumper JP1 is used to configure the Analog Input type as either single-ended or differential. To configure single-ended input, connect pin1 to pin 3 and pin 2 to pin 4, as illustrated in the diagram below. The default configuration is single-ended.

Jumper	Single-ended Input (Default)	Differential Input
JP1		

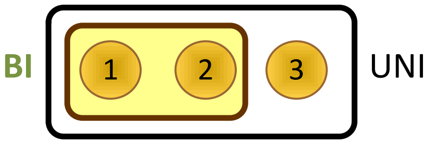
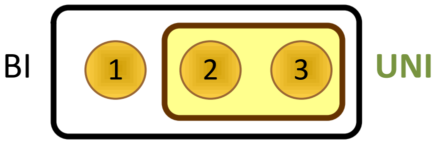
2.2.2 JP6/JP7 (Analog Output Range)

Jumpers JP6 and JP7 are used to configure the output range for Analog Output channel 0 and 1, respectively. Connecting pins 1 and 2 sets the output range to -10 V to +10 V or 0 to +10 V, while connecting pins 2 and 3 sets the output range to -5 V to +5 V or 0 to +5 V. The default configuration is ± 10 V.

Jumper	± 10 V or 0 to +10 V (Default)	± 5 V or 0 to +5 V
JP6 (D/A CH0) **JP7 (D/A CH1)**		

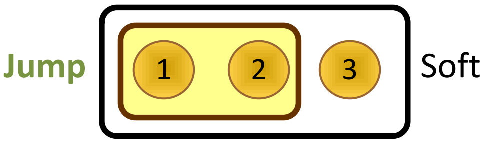
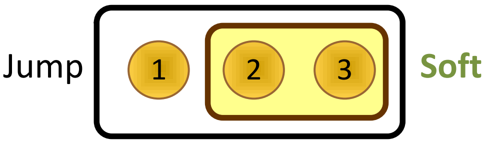
2.2.3 JP3/JP5 (Analog Output Type)

Jumpers JP3 and JP5 are used to select the Analog output type. Connecting pins 1 and 2 sets the output type to Bipolar, while connecting pins 2 and 3 sets the output type to Unipolar. The default configuration is Bipolar.

Jumper	Bipolar (Default)	Unipolar
JP3 (D/A CH0) **JP5 (D/A CH1)**		

2.2.4 JP4 (Digital I/O Mode)

The JP4 Jumper block is used to configure the Digital Input/Output mode as either **Jumper Selectable** or **Software Programmable** Mode. **Connecting pins 1 and 2** of DIO-S0 sets the DI/O mode to **Jumper Selectable Mode**, while **connecting pins 2 and 3** sets the DI/O mode to **Software Programmable Mode**. The default configuration is Jumper Selectable Mode.

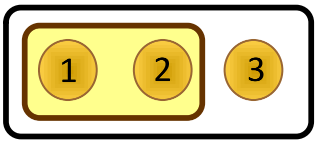
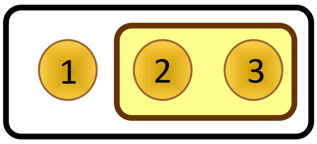
Jumper	Jumper Selectable (Default)	Software Programmable
DIO-S0		

- **Software Programmable Mode:**

Refer to "Section 6.3 Bar1: Digital I/O Registers" for details of how to configure Port A (PA) and Port B (PB) when the jumpers for DIO-S0 is set to Software Programmable Mode. The jumpers for DIO-S1 and DIO-S2 are not used when the DIO-S0 is set to Software Programmable Mode.

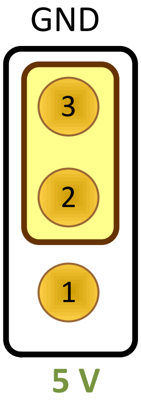
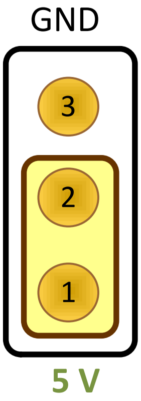
➢ Jumper Selectable Mode:

DIO-S1 (Port A, PA) and DIO-S2 (Port B, PB) are used to configure the I/O ports as either Digital Input (connect pins 1 and 2) or Digital Output (connect pins 2 and 3), when the jumper for DIO-S0 is set to Jumper Selectable Mode. The default configuration is Digital Input.

Jumper	DIO-S0 is Jumper Selectable Mode	
	DI **(Default)**	**DO**
DIO-S1 (Port A) **DIO-S2 (Port B)**		

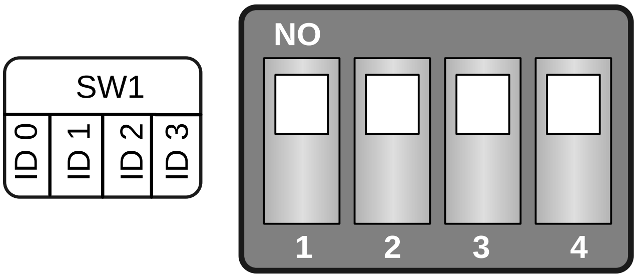
2.2.5 JP8 (Digital Input Pull-high/low)

Jumper JP8 is used to set the Digital Input to either Pull-high or Pull-low, which allows the status of the Digital Input to be predefined instead of remaining floating if the Digital Input channels are disconnected or interrupted. Connecting pins 1 and 2 will set the Digital Input to Pull-high. To set the Digital Input to Pull-low, pins 2 and 3 should be shorted. The default connected is Pull-low.

Jumper	Pull-low (Default)	Pull-high
JP8		

2.3 Card ID Switch

PCI-822/826 Series cards include an onboard Card ID DIP Switch (SW1) that enables the card to be recognized via software if two or more PCI-822/826 Series cards are installed in the same computer. The default Card ID is 0x0 in hexadecimal format. For more detailed information regarding the positions of the SW1 DIP Switch for the different Card ID settings, refer to the table below.



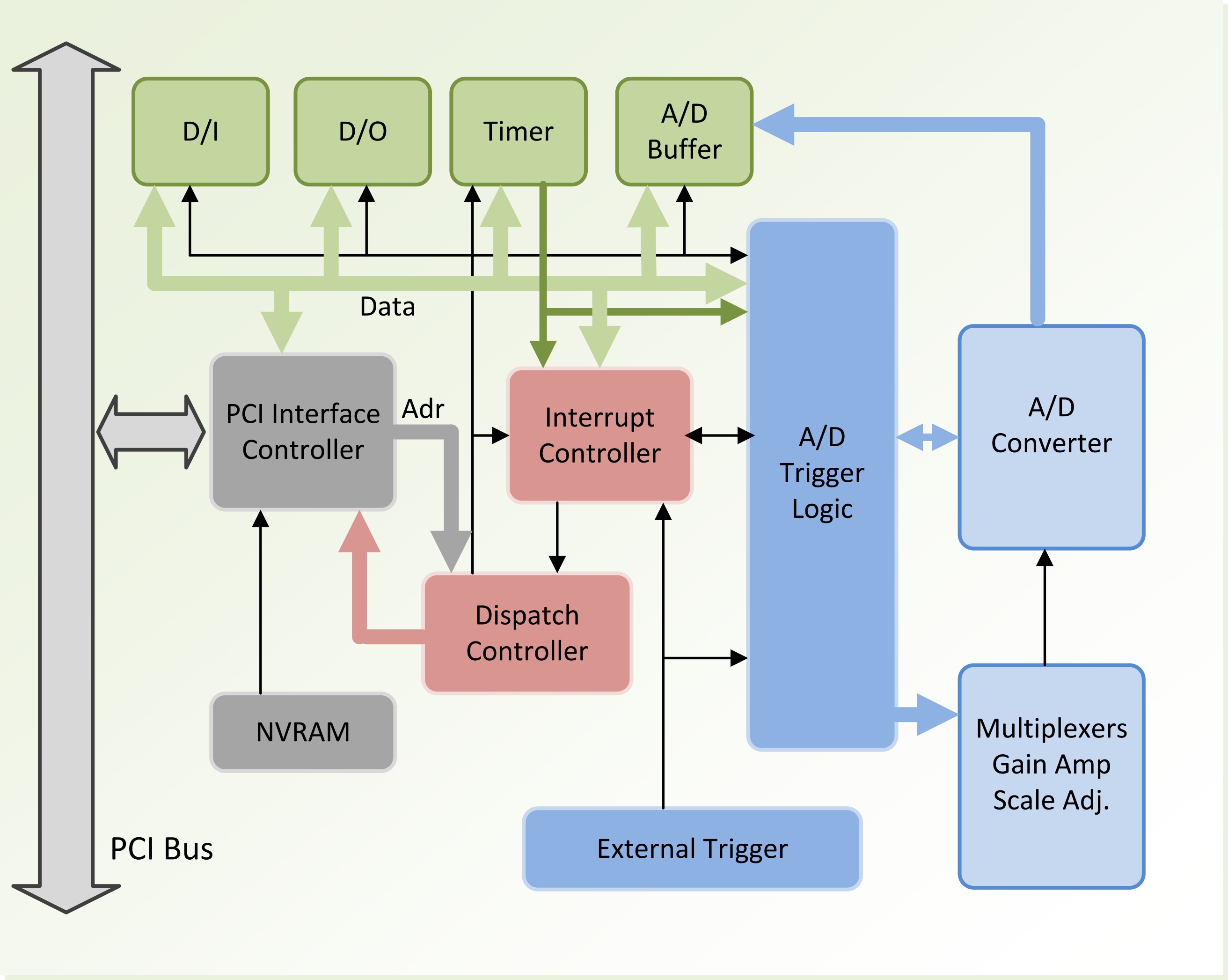
(Default Settings)

Card ID (Hex)	1 ID0	2 ID1	3 ID2	4 ID3
(*) 0x0	ON	ON	ON	ON
0x1	**OFF**	ON	ON	ON
0x2	ON	**OFF**	ON	ON
0x3	**OFF**	**OFF**	ON	ON
0x4	ON	ON	**OFF**	ON
0x5	**OFF**	ON	**OFF**	ON
0x6	ON	**OFF**	**OFF**	ON
0x7	**OFF**	**OFF**	**OFF**	ON
0x8	ON	ON	ON	**OFF**
0x9	**OFF**	ON	ON	**OFF**
0xA	ON	**OFF**	ON	**OFF**
0xB	**OFF**	**OFF**	ON	**OFF**
0xC	ON	ON	**OFF**	**OFF**
0xD	**OFF**	ON	**OFF**	**OFF**
0xE	ON	**OFF**	**OFF**	**OFF**
0xF	**OFF**	**OFF**	**OFF**	**OFF**

(*) Default Settings; ON = 0; OFF = 1

2.4 Block Diagram

The following is an illustration of the System Function Block for PCI-822/826 Series cards:



2.5 Analog Input Signal Connections

PCI-822/826 Series cards can be used to measure either single-ended or differential type Analog Input signals. Although certain signals can be measured using either mode, other signals, however, can only be measured in one mode or the other. The most suitable mode for the measurement must be determined beforehand, and is totally dependent on the prevailing measurement conditions.

In general, there are four different methods that can be used for connecting Analog Input signals, as shown below in **Figures 2.5-1 to 2.5-5**. The connection method illustrated in **Figure 2.5-1** is more suitable for grounding source Analog Input signals. The connection method depicted in **Figure 2.5-3** is used to measure more channels than the method shown in **Figure 2.5-1**, but it is only suitable for large Analog Input signals. The connection method shown in **Figure 2.5-4** is suitable for thermocouple input signals, while the connection method illustrated in **Figure 2.5-5** is suitable for floating source Analog Input signals.

Important Note: For the method illustrated in Figure 2.5-4, the maximum allowable common mode voltage between the Analog Input source and the AGND pin is $70V_{p-p}$, so care must be taken to ensure that the input signal is below this level before continuing. If the common mode voltage is set to above $70V_{p-p}$, the input multiplexer will be permanently damaged.

The easiest approach to selecting the most suitable configuration for the input signal connection is outlined below:

1. **Grounding the source input signal → see Figure 2.5-1**
2. **Thermocouple input signal → see Figure 2.5-4**
3. **Floating source input signal → see Figure 2.5-5**
4. **If Vin > 1 V, and the gain is <=10 and more channels are needed → see Figure 2.5-3**
5. **Current source input signal → see Figure 2.5-6**

If the characteristics of the input signal are unknown or unclear, test the signal using the following procedure to determine the most suitable method:

1. **Measure the signal using the connection illustrated in Figure 2.5-1 and record the result**
2. **Measure the signal using the connection illustrated in Figure 2.5-5 and record the result**
3. **Measure the signal using the connection illustrated in Figure 2.5-3 and record the result**
4. **Compare the three results and select the most suitable connection method**

Figure 2.5-1: Connecting to a grounded source input (correct method)

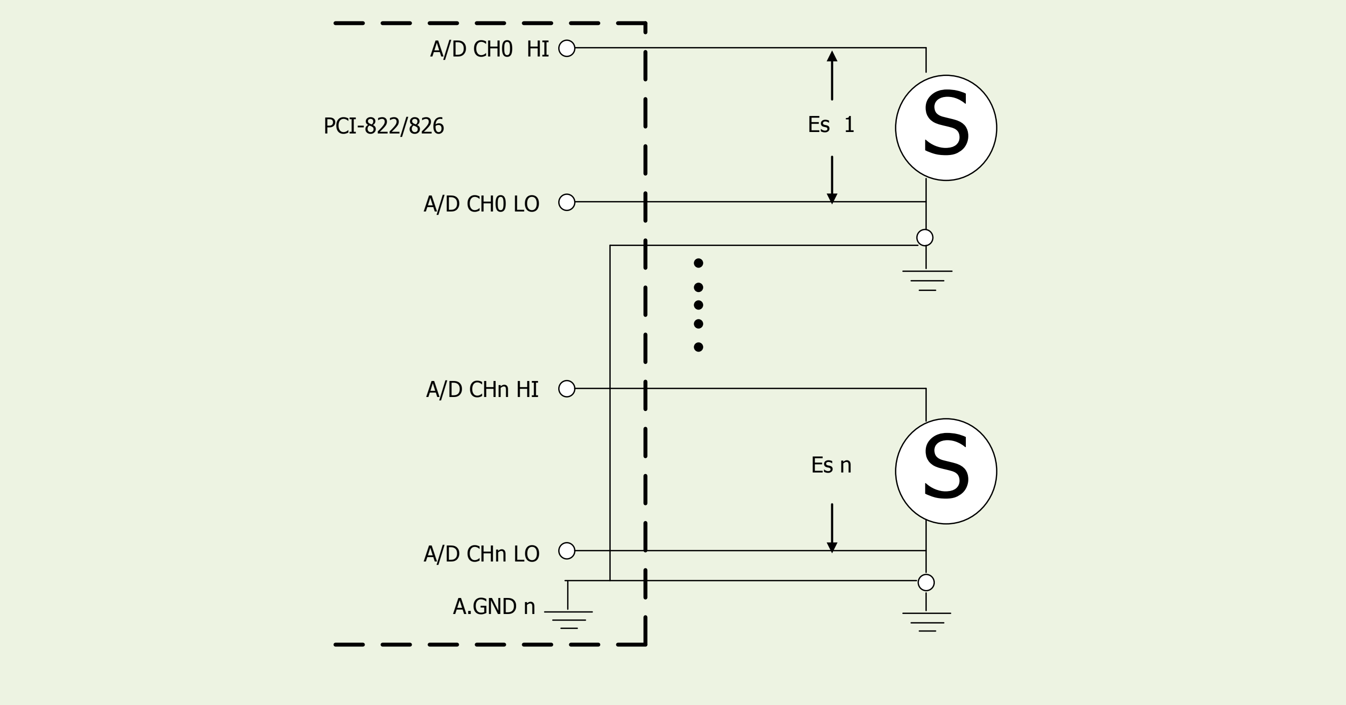


Figure 2.5-2: Connecting to a ground loop input (incorrect method)

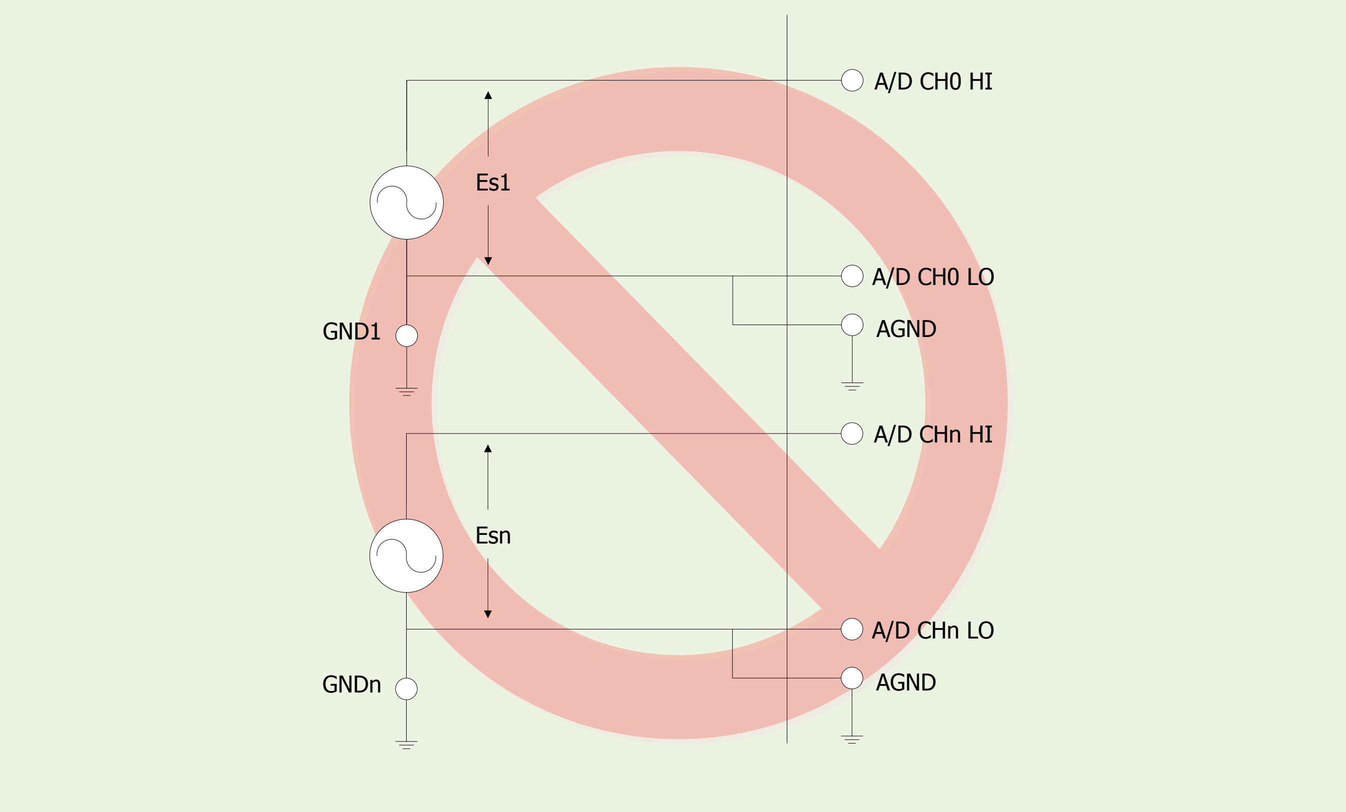


Figure 2.5-3: Connecting to a single-ended input configuration

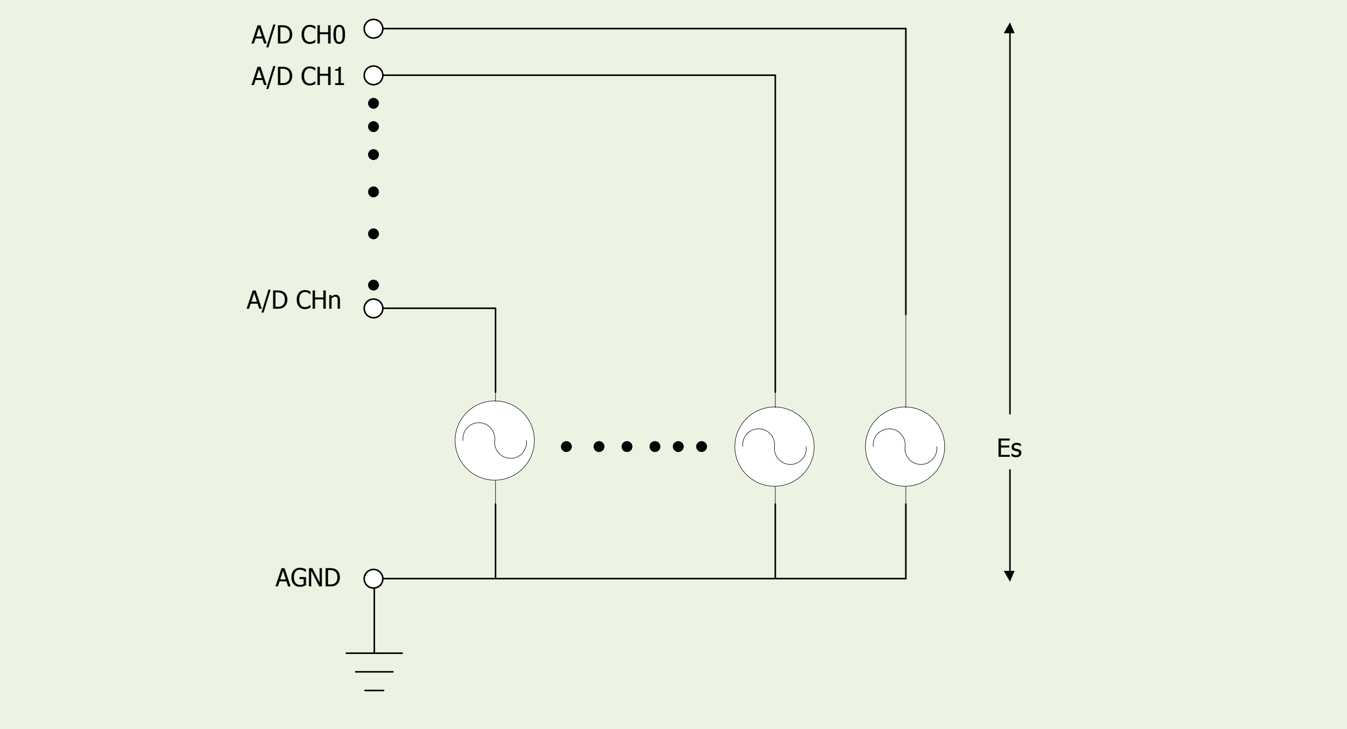
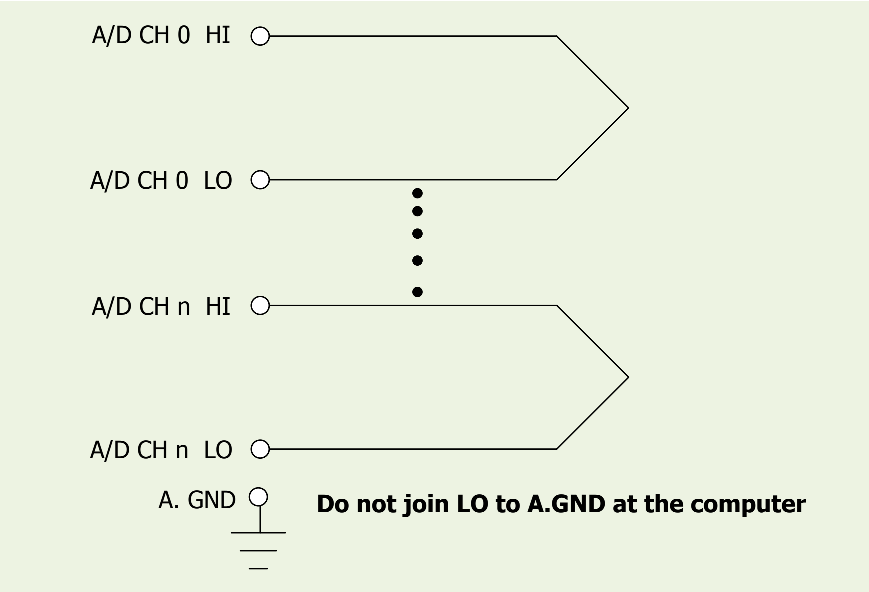


Figure 2.5-4: Connecting to a thermocouple input configuration



Note: If the input is not a thermocouple signal, an oscilloscope should be used to measure the common mode voltage of the V_{in} before connecting to the PCI-822/826, rather than using a voltage meter or multimeter.

CAUTION: The maximum allowable common mode voltage between the Analog Input source and the A.GND pin for the connection shown in Figure 2.5-4, is 70 V_{p-p}. Ensure that the input signal is below this value before making any connections. If the common mode voltage is above 70 V_{p-p}, the input multiplexer will be permanently damaged.

Figure 2.5-5: Connecting to floating source configuration

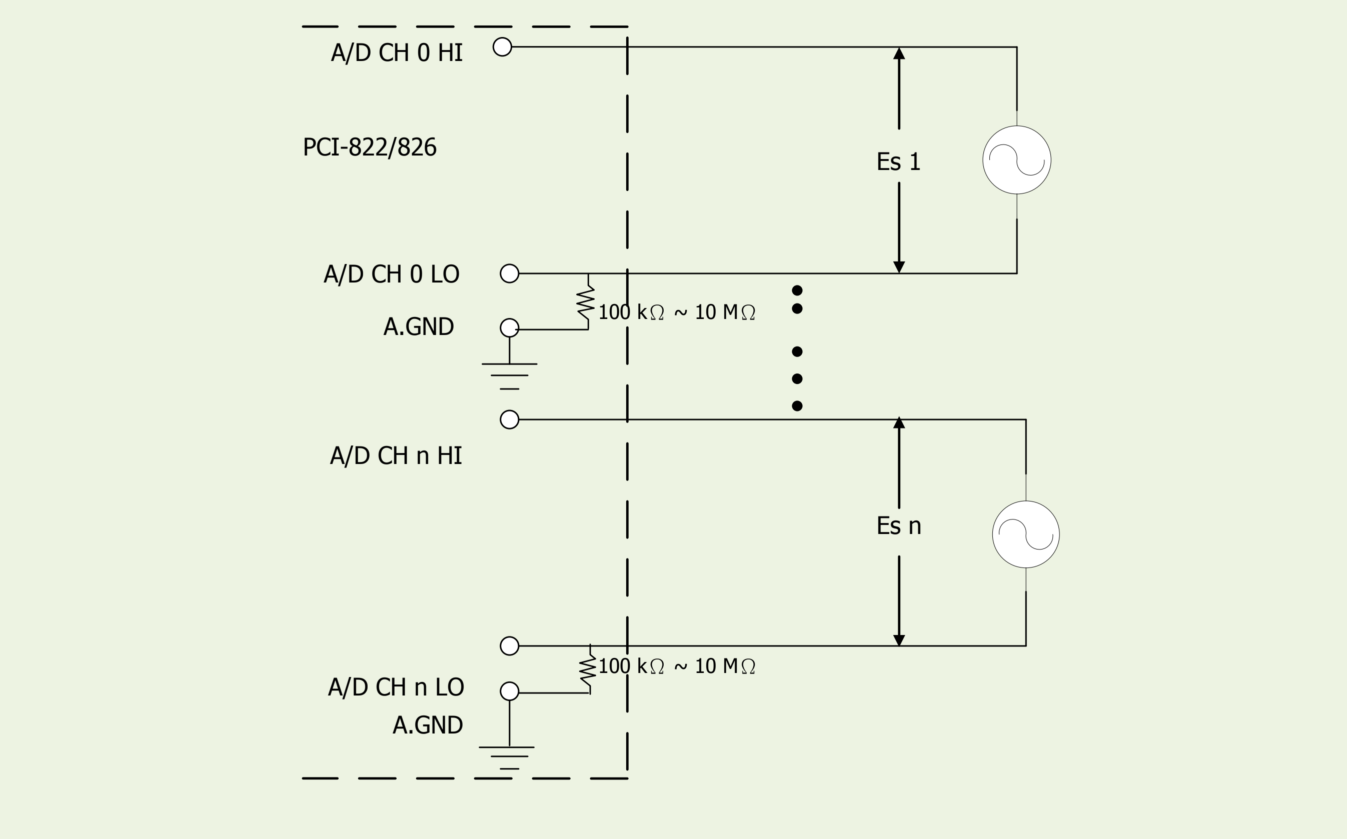
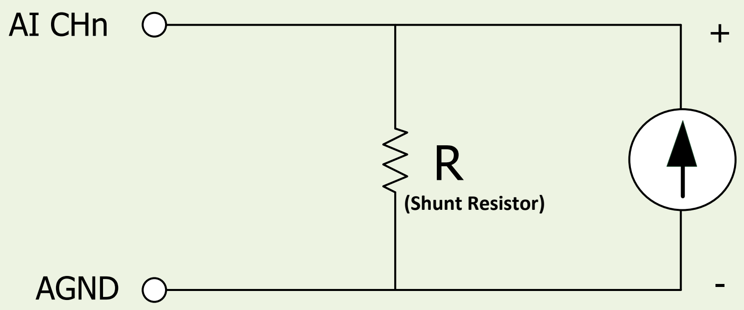


Figure 2.5-6: Connecting to a 4 ~ 20 mA Source



Example: A 20 mA source current through a 125 Ω resistor (e.g. 125 Ω, 0.1% DIP Resistors) between + and – terminals and the board will read a 2.5 V_{DC} voltage. You can use the **I = V/R** (Ohm's law) to calculate what value the source current should have.

Current (I) = Voltage (V) / Resistance (R)

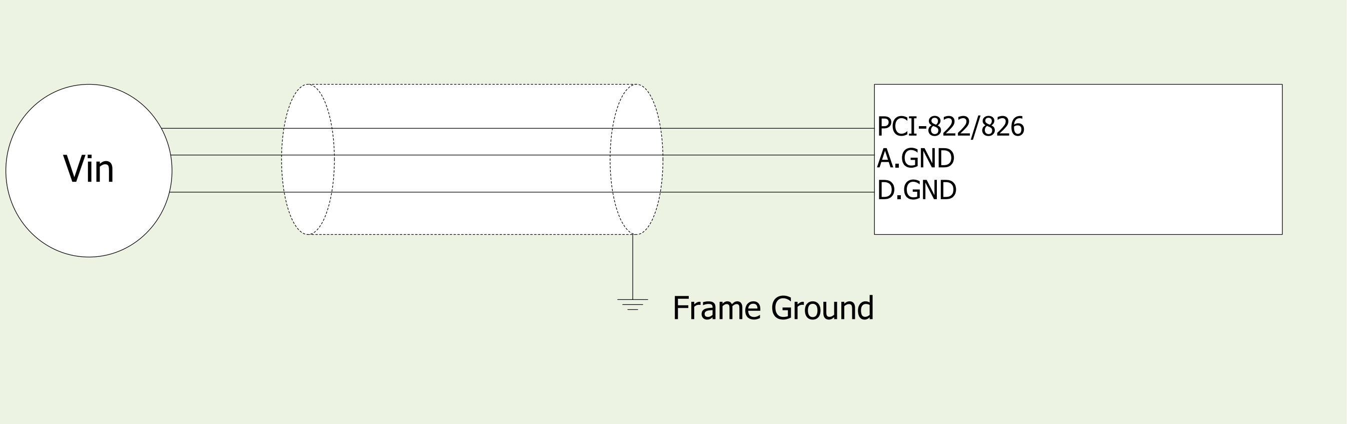
= 2.5 V / 125 Ω

= 0.02 A

= 20 m

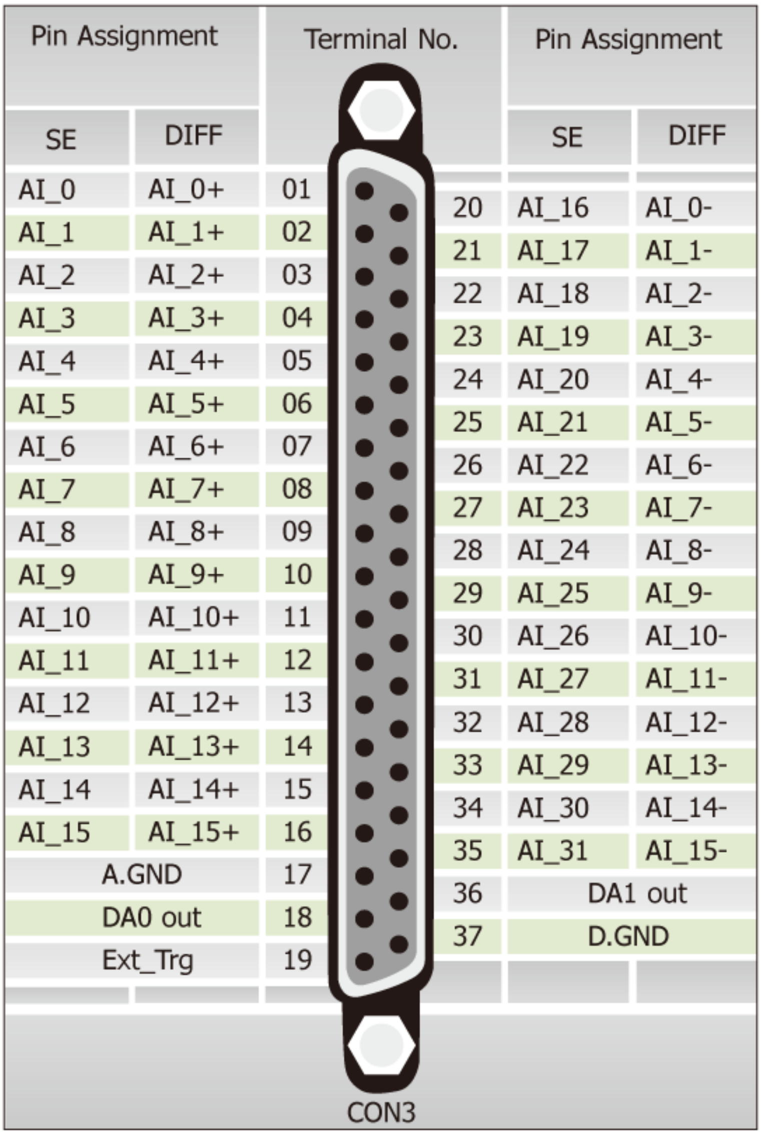
Signal Shielding

- The signal shielding for the connection methods illustrated in **Figures 2.5-1 to 2.5-6** are the same
- Use a single-point connection to the **frame ground, rather than the A.GND or the D.GND pins**



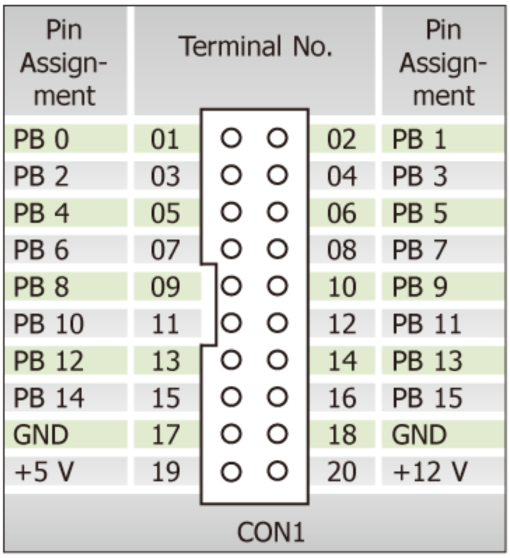
2.6 Pin Assignments

The following is an overview of the pin assignments for PCI-822/826 Series cards. CON1 and CON2 relate to the Digital Input and Digital Output connectors, while CON3 is used for both single-ended and differential input.



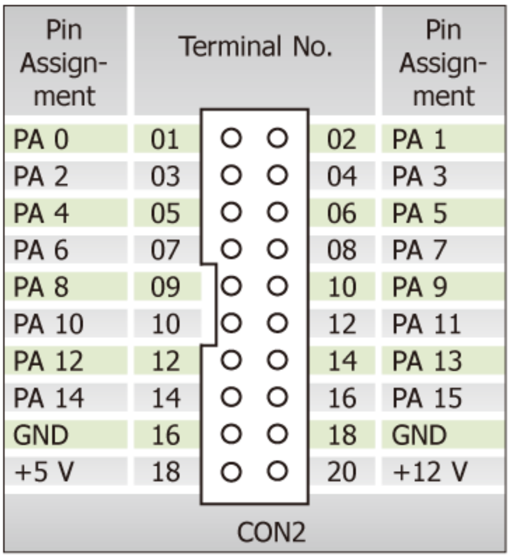
Pin Assignment		Terminal No.	Terminal No.	Pin Assignment	
SE	DIFF			SE	DIFF
AI_0	AI_0+	01	20	AI_16	AI_0-
AI_1	AI_1+	02	21	AI_17	AI_1-
AI_2	AI_2+	03	22	AI_18	AI_2-
AI_3	AI_3+	04	23	AI_19	AI_3-
AI_4	AI_4+	05	24	AI_20	AI_4-
AI_5	AI_5+	06	25	AI_21	AI_5-
AI_6	AI_6+	07	26	AI_22	AI_6-
AI_7	AI_7+	08	27	AI_23	AI_7-
AI_8	AI_8+	09	28	AI_24	AI_8-
AI_9	AI_9+	10	29	AI_25	AI_9-
AI_10	AI_10+	11	30	AI_26	AI_10-
AI_11	AI_11+	12	31	AI_27	AI_11-
AI_12	AI_12+	13	32	AI_28	AI_12-
AI_13	AI_13+	14	33	AI_29	AI_13-
AI_14	AI_14+	15	34	AI_30	AI_14-
AI_15	AI_15+	16	35	AI_31	AI_15-
A.GND		17	36	DA1 out	
DA0 out		18	37	D.GND	
Ext_Trg		19			

CON3



Pin Assign-ment	Terminal No.	Terminal No.	Pin Assign-ment
PB 0	01	02	PB 1
PB 2	03	04	PB 3
PB 4	05	06	PB 5
PB 6	07	08	PB 7
PB 8	09	10	PB 9
PB 10	11	12	PB 11
PB 12	13	14	PB 13
PB 14	15	16	PB 15
GND	17	18	GND
+5 V	19	20	+12 V

CON1



Pin Assign-ment	Terminal No.	Terminal No.	Pin Assign-ment
PA 0	01	02	PA 1
PA 2	03	04	PA 3
PA 4	05	06	PA 5
PA 6	07	08	PA 7
PA 8	09	10	PA 9
PA 10	10	12	PA 11
PA 12	12	14	PA 13
PA 14	14	16	PA 15
GND	16	18	GND
+5 V	18	20	+12 V

CON2

3 Hardware Installation

Note:

As certain operating systems, such as Windows 2000 and Windows XP may require the computer to be restarted after a new driver is installed, it is recommended that the driver is installed first, which will reduce the time required for installation.

To install your PCI-822/826 Series card, follow the procedure described below:

Step 1: Install the driver for the PCI-822/826 Series card on the computer.

For more detailed information about installing the driver for the PCI-822/826 Series card, refer to **Chapter 4 Software Installation.**

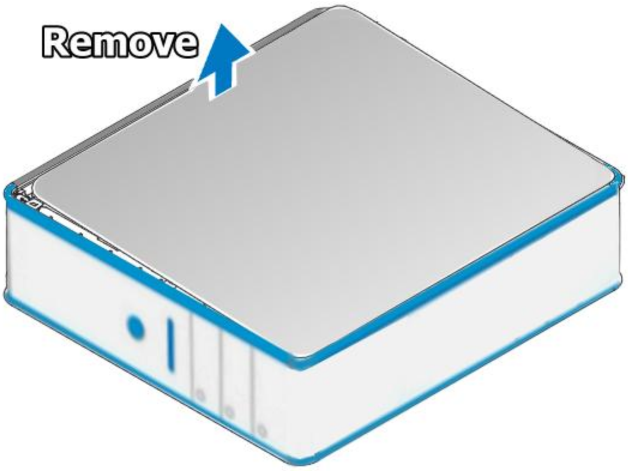
Step 2: Configure the Jumpers and the SW1 DIP Switch on the PCI-822/826 Series card depending on the required configuration.

For more detailed information about the configuring the Jumpers and Card ID (SW1), refer to **Section 2.2 Jumper Settings** and **Section 2.3 Card ID Switch**.

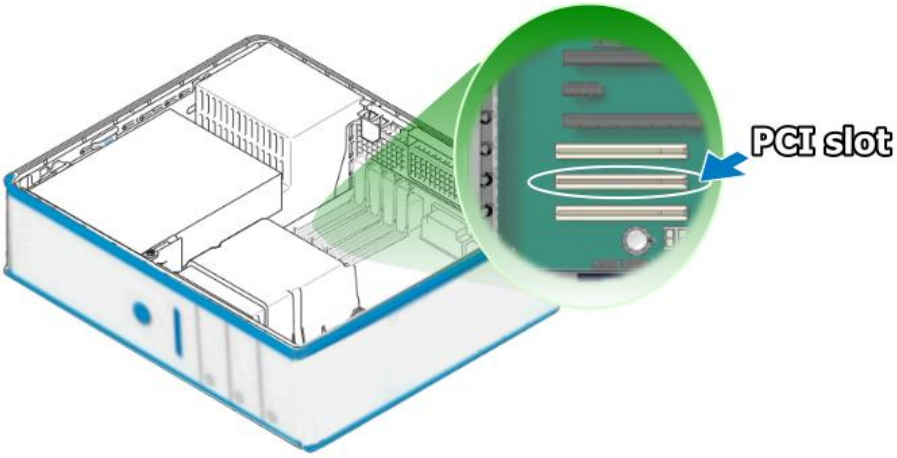
Turn off computer
Turn Off

Step 3: Correctly shut down and power off your computer, and then disconnect the power supply.

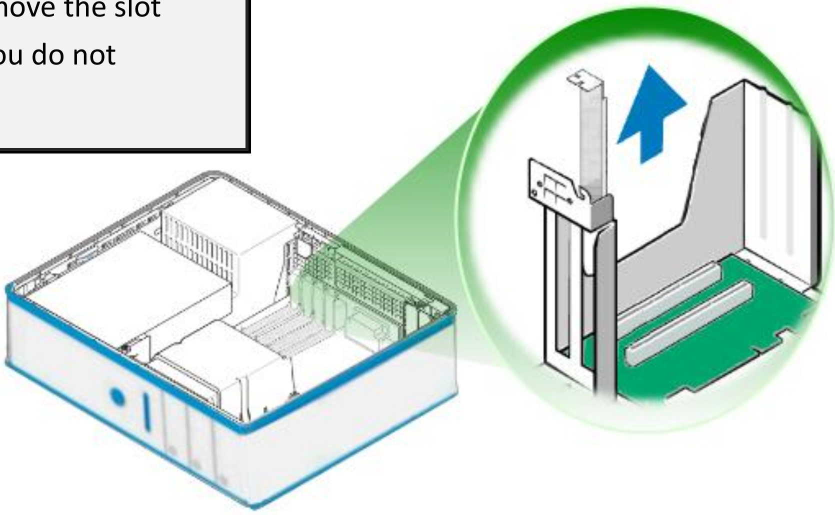
Step 4: Remove the cover from the computer.

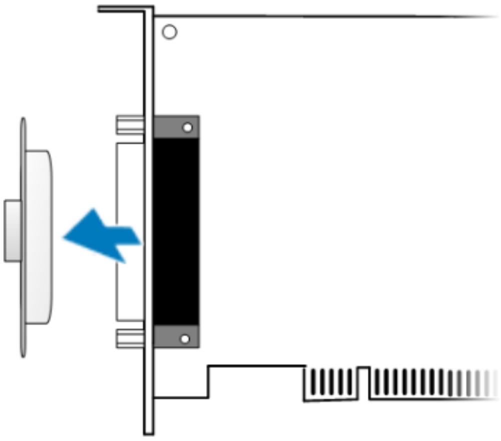


Step 5: Select a vacant PCI slot.



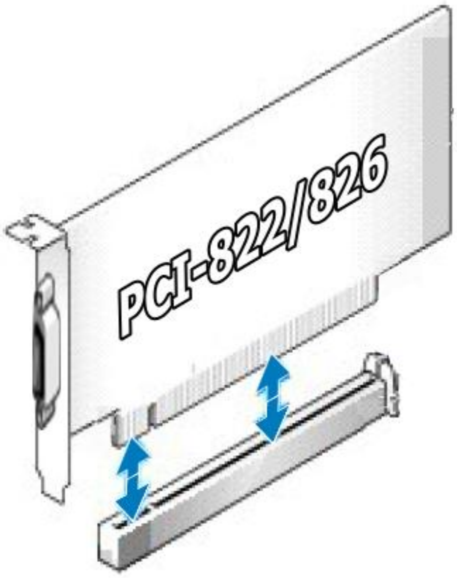
Step 6: Remove the screw holding the cover for the PCI slot in place and then remove the slot cover from the PC. Ensure that you do not misplace the screw.

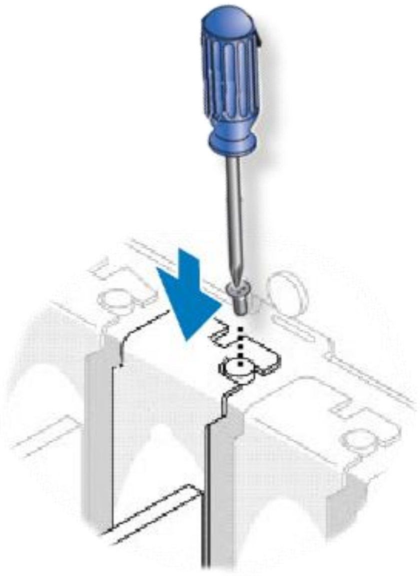




Step 7: Remove the connector cover from the PCI-822/826 Series card.

Step 8: Align the contacts of the PCI card with the open slot on the motherboard and then carefully insert the PCI-822/826 Series card into the PCI slot by gently pushing down on both sides of the card until it slides into the slot.

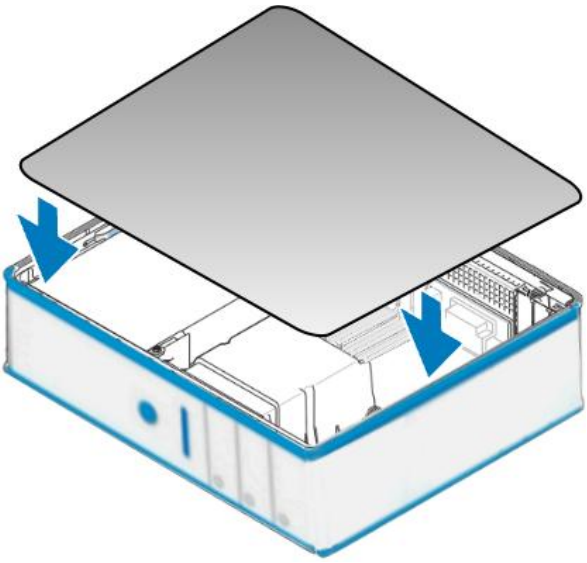




Step 9: Secure the PCI-822/826 Series card in place using the retaining screw that was removed in step 5.

Confirm that the PCI-822/826 series card is correctly mounted in the motherboard.

Step 10: Replace cover for the computer and then reconnect the power supply and any other cables.



Step 11: Switch on the power to the computer.

Once the computer reboots, follow any messages that may be displayed to complete the Plug and Play installation procedure. Refer to **Chapter 4 Software Installation** for more information.

4 Software Installation

This chapter provides a detailed description of the process for installing the driver for the PCI-822/826 Series card as well as how to verify whether the PCI-822/826 Series card was properly installed. PCI-822/826 Series cards can be used on DOS, Linux and Windows 2000 and 32-/64-bit version of Windows XP/2003/Vista/7/8 based systems, and the drivers are fully Plug and Play compliant for easy installation.

4.1 Obtaining the Driver Installation Package

The driver installation package for PCI-822/826 Series cards can be found on the companion CD-ROM, or can be obtained from the ICP DAS FTP web site. Install the appropriate driver for the operating system. The location and website addresses for the installation package are indicated below.

➢ UniDAQ Driver/SDK

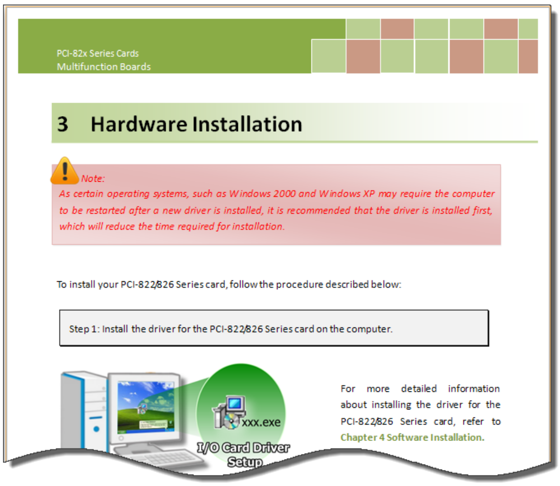
Operating System	**Windows 2000、32/64-bit Windows XP、32/64-bit Windows 2003、32/64-bit Windows Vista、32/64-bit Windows 7、32/64-bit Windows 2008 and 32/64-bit Windows 8**
Driver Name	**UniDAQ Driver/SDK (unidaq_win_setup_xxxx.exe)**
CD-ROM	CD:\\ NAPDOS\PCI\UniDAQ\DLL\Driver\
Website	http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/dll/driver/
Installation Procedure	Please follow the following steps to setup software: **Step 1:** Double click the **UniDAQ_Win_Steupxxx.exe** to setup it. **Step 2:** When the Setup Wizard screen is displayed, click the **Next>** button. **Step 3:** When the Information screen is displayed, click the **Next>** button.

Installation Procedure	**Step 4:** Select the folder where the drivers are to install. The **default path is C:\ICPDAS\UniDAQ**. But if you wish to install the drivers to a different location , click the **"Browse…"** button and select the relevant folder and then click the **Next>** button. **Step 5:** When the Select Components screen is displayed, check PCI-P16R16 series board on the list, then click the **Next>** button. **Step 6:** When the Select Additional Tasks screen is displayed, click the **Next>** button. **Step 7:** When the Download Information screen is displayed, click the **Next>** button. **Step 8:** Select the item **"Yes, restart the computer now"**, press the **Finish** button. System will reboot. For more detailed information about how to install the UniDAQ driver, refer to "Section 2.2 Install UniDAQ Driver DLL" of the UniDAQ Software Manual, which can be found in the \NAPDOS\PCI\UniDAQ\Manual\ folder on the companion CD, or can be downloaded from: http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/manual/

- Linux Driver

OS	Linux Kernel 2.4.x/2.6.x/3.12.x
Driver Name	lxpci.tar.gz
CD-ROM	CD:\\ NAPDOS\Linux\
Website	http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/linux/
Installation Procedure	For more detailed information about how to install the Linux driver, refer to the readme.txt file that can be found in the \NAPDOS\Linux\ folder on the companion CD.

4.2 Plug and Play Driver Installation



PCI-82x Series Cards
Multifunction Boards

3 Hardware Installation

Note:
As certain operating systems, such as Windows 2000 and Windows XP may require the computer to be restarted after a new driver is installed, it is recommended that the driver is installed first, which will reduce the time required for installation.

To install your PCI-822/826 Series card, follow the procedure described below:

Step 1: Install the driver for the PCI-822/826 Series card on the computer.

For more detailed information about installing the driver for the PCI-822/826 Series card, refer to Chapter 4 Software Installation.

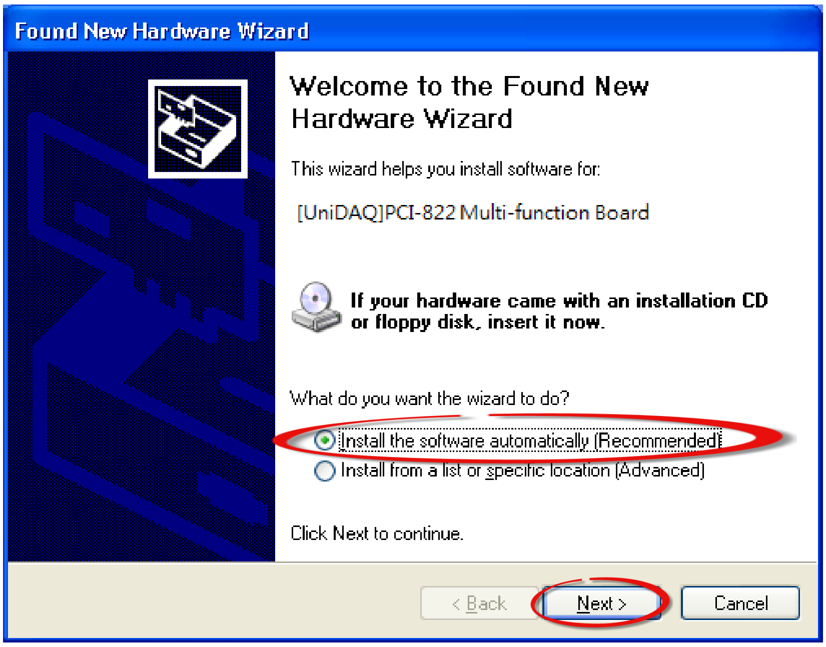
Step 1: Correctly shut down and power off your computer and disconnect the power supply, and then install the PCI-822/826 Series board into the computer.

For detailed information about the hardware installation of the PCI-822/826 Series board, refer to Chapter 3 Hardware Installation.

Step 2: Power on the computer and complete the Plug and Play installation**.**

Note: More recent operating systems, such as Windows 7/8 will automatically detect the new hardware and install the necessary drivers etc., so Steps 3 to 5 can be skipped.

Step 3: Select **"Install the software automatically [Recommended]"** and click the **"Next>"** button.



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install software for:

[UniDAQ]PCI-822 Multi-function Board

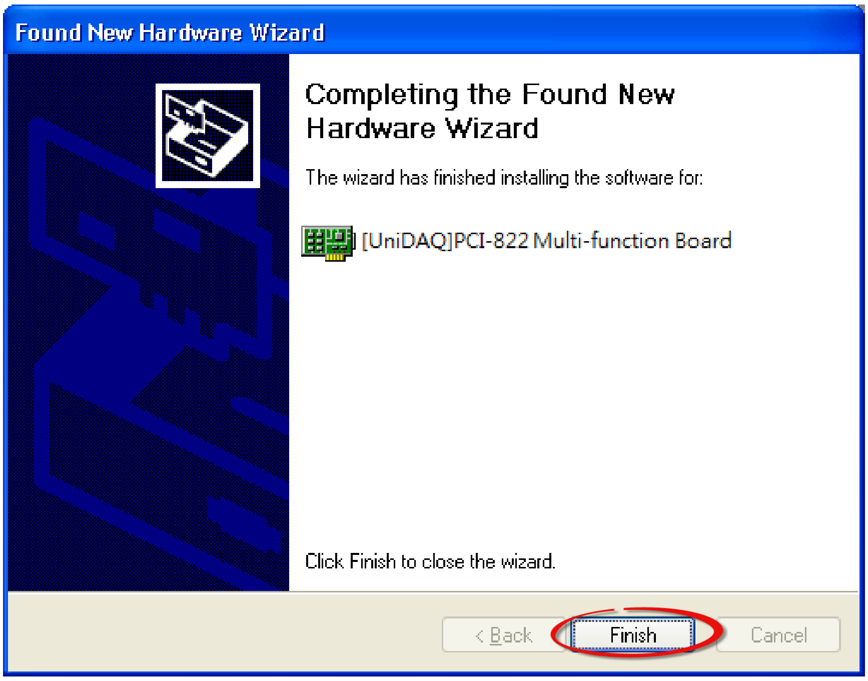
If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

● Install the software automatically (Recommended)
○ Install from a list or specific location (Advanced)

Click Next to continue.

Step 4: Click the **"Finish"** button.



Found New Hardware Wizard

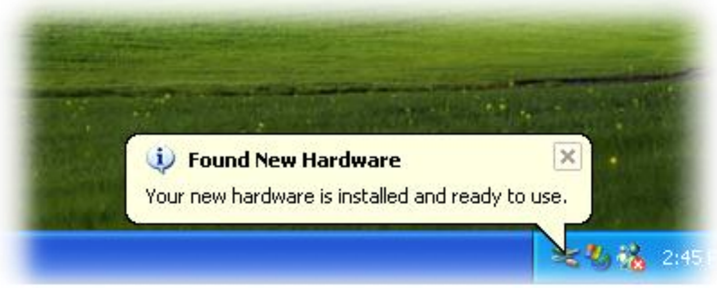
Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

[UniDAQ]PCI-822 Multi-function Board

Click Finish to close the wizard.

Step 5: Windows pops up **"Found New Hardware"** dialog box again.



Found New Hardware

Your new hardware is installed and ready to use.

4.3 Verifying the Installation

To verify that the driver was correctly installed, use the Windows **Device Manager** to view and update the device drivers installed on the computer, and to ensure that the hardware is operating correctly. The following is a description of how access the Device Manager in each of the major versions of Windows. Refer to the appropriate description for the specific operating system to verify the installation.

4.3.1 Accessing Windows Device Manager

■ Windows 95/98/ME

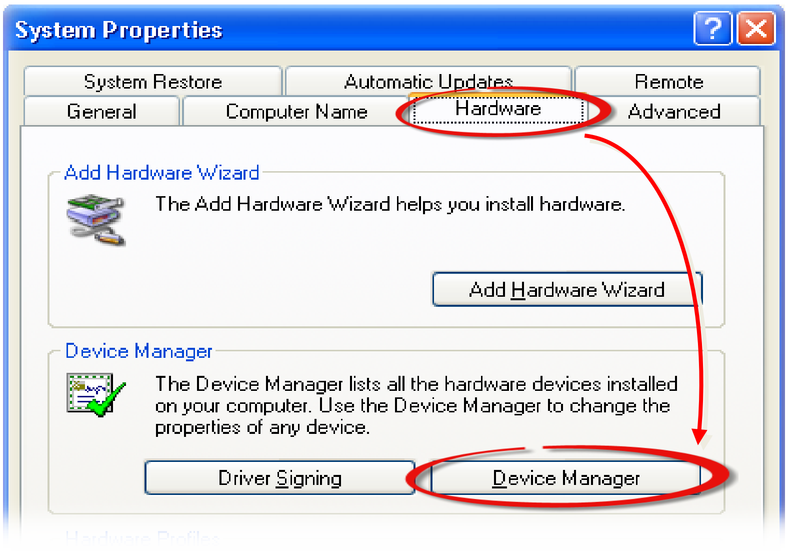
Step 1: Either right-click the **"My Computer"** icon on the desktop and then click **"Properties"**, or open the **"Control Panel"** and double-click the **"System"** icon to open the **System Properties** dialog box.

Step 2: In the **System Properties** dialog box, click the "**Device Manager**" tab.

■ Windows 2000/XP

Step 1: Click the "**Start"** button and then point to "**Settings"** and click "**Control Panel**". Double-click the "**System**" icon to open the **"System Properties"** dialog box.

Step 2: Click the "**Hardware**" tab and then click the "**Device Manager**" button.



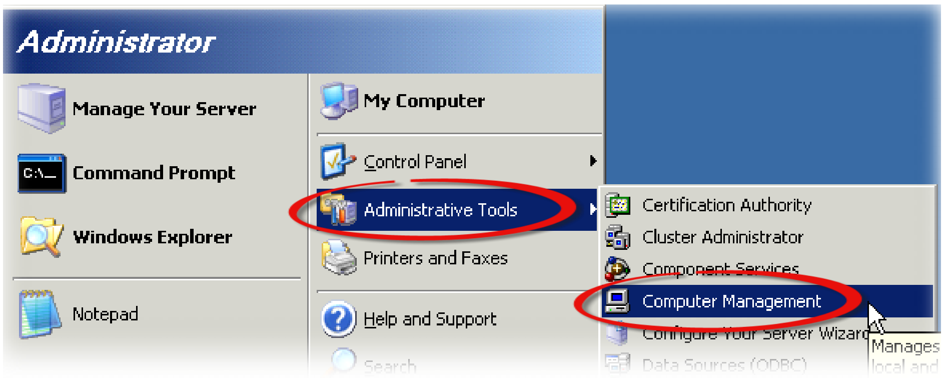
The Add Hardware Wizard helps you install hardware.

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

Windows Server 2003

Step 1: Click the **"Start"** button and point to **"Administrative Tools"**, and then click the **"Computer Management"** option.

Step 2: Expand the **"System Tools"** item in the console tree, click **"Device Manager"**.



Windows Vista/7

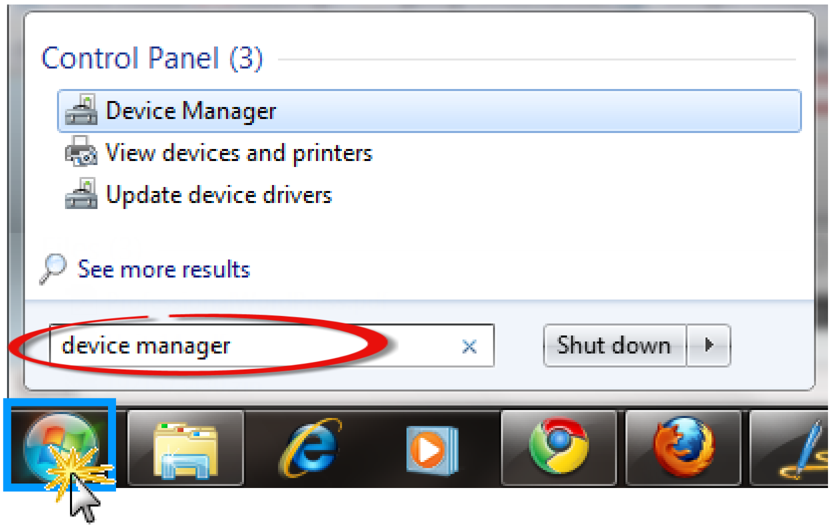
Step 1: Click the **"Start"** button, and then click **"Control Panel"**.

Step 2: Click the **"System and Maintenance"**, and then click **"Device Manager"**.

Alternatively,

Step 1: Click the **"Start"** button.

Step 2: In the **Search field,** type **Device Manager** and then press Enter.



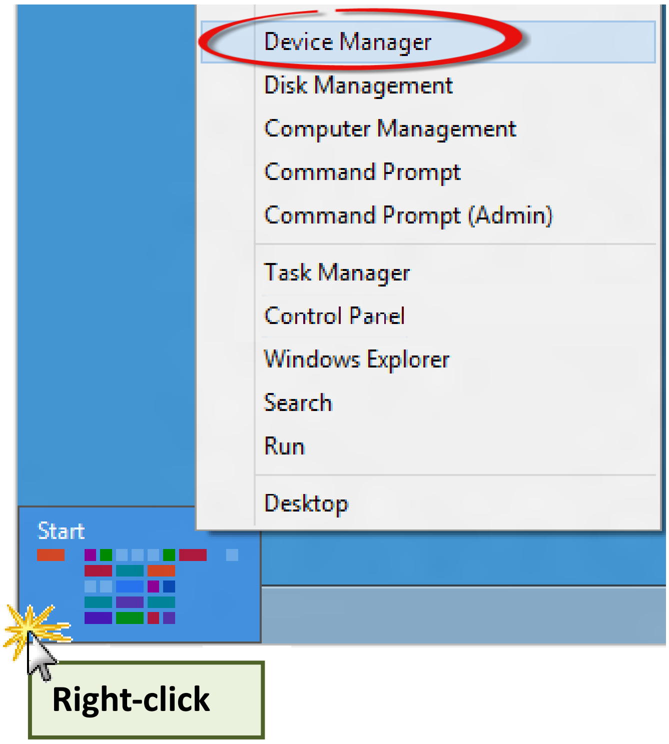
Note that Administrator privileges are required for this operation. If you are prompted for an administrator password or confirmation, enter the password or provide confirmation by clicking the "Yes" button in the User Account Control message.

Microsoft Windows 8

Step 1: To display the **Start screen icon** from the desktop view, hover the mouse cursor over the **bottom-left corner** of screen.

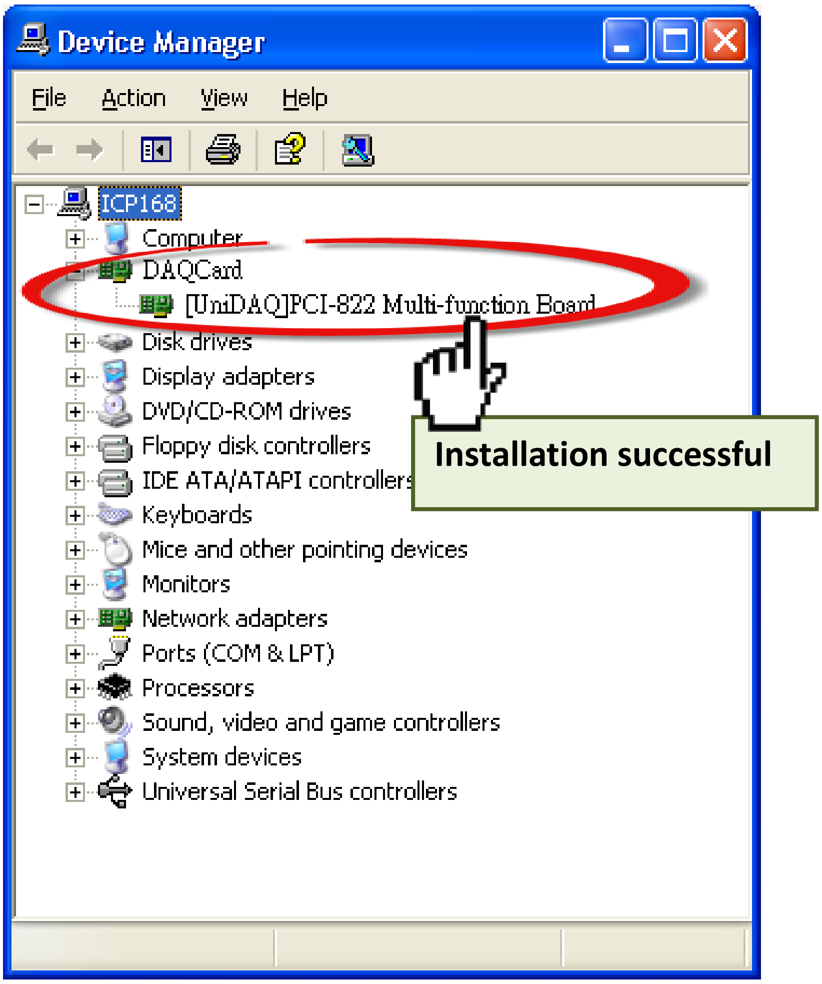
Step 2: **Right-click** the Start screen icon then click "**Device Manager**".

Alternatively, press **[Windows Key] +[X]** to open the Start Menu, and then select Device Manager from the options list.



4.3.2 Check the Installation

Check that the PCI-822/826 Series card is correctly listed in the Device Manager, as illustrated below.



Installation successful

5 Testing the PCI-822/826 Series Card

This chapter provides detailed information about the **"Self-Test"** process, which is used to confirm that PCI-822/826 Series card is operating correctly. Before beginning the **"Self-Test"** process, ensure that both the hardware and driver installation procedures are fully completed. For detailed information about the hardware and driver installation, refer to **Chapter 3 Hardware Installation** and **Chapter 4 Software Installation**.

5.1 Self-Test Wiring

The following is a description of how to configure the wiring in order to perform the "Self-Test" procedures for the Digital Input and Digital Output. Refer to the appropriate descriptions for PCI-822/826 Series cards in Sections 5.1.1 to 5.1.3 for more detailed information.

5.1.1 Digital Input/Output Test Wiring

- Preparing the device:

Before beginning the **"Self-Test"** procedure, ensure that the following items are available:

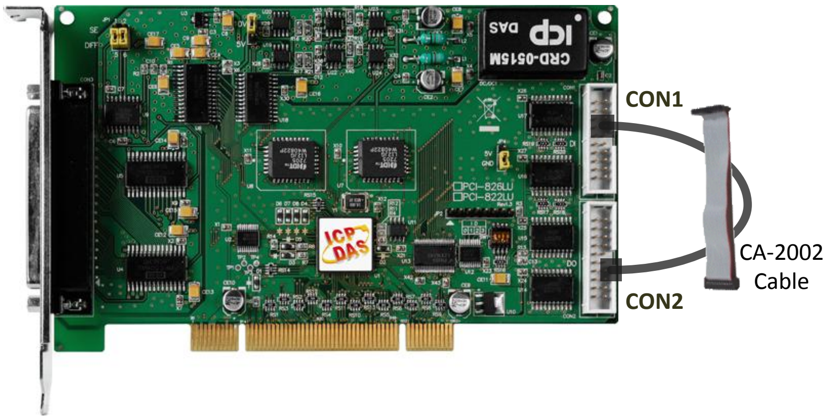
☑ A CA-2002 cable

(Website: http://www.icpdas.com/products/Accessories/cable/cable_selection.htm)

Step 1: Ensure that the **JP4 jumper (DIO-S0)** on the PCI-822/826 card is set to the **"soft" position**. Refer to **Section 2.2.4 JP4 Digital I/O Mode** for more detailed information.

*Note: The direction of the DI/O ports are controlled by software when the **"software Program Mode"** option is selected.*

Step 2: Use the CA-2002 cable to connect the CON1 connector to the CON2 connector on the PCI-822/826 Series card.



5.1.2 Analog Input/Output Test Wiring

- Preparing the device:

Before beginning the **"Self-Test"** procedure, ensure that the following items are available:

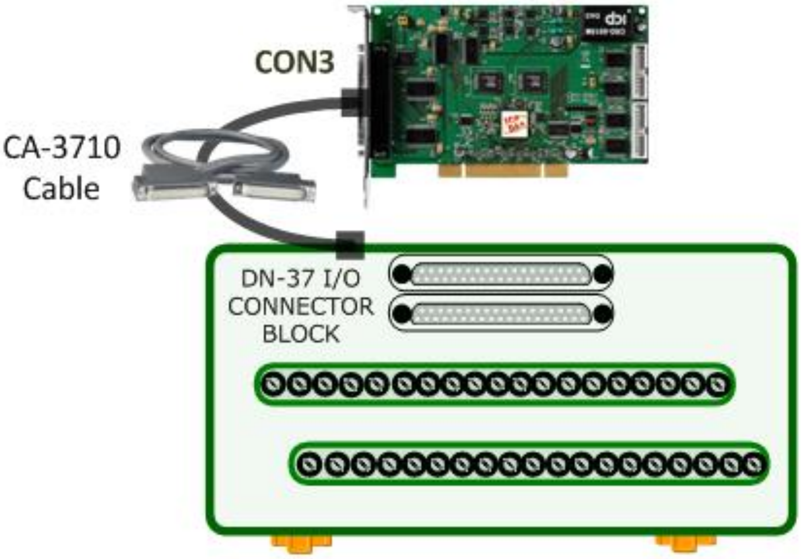
☑ A CA-3710 cable

(Website: http://www.icpdas.com/products/Accessories/cable/cable_selection.htm)

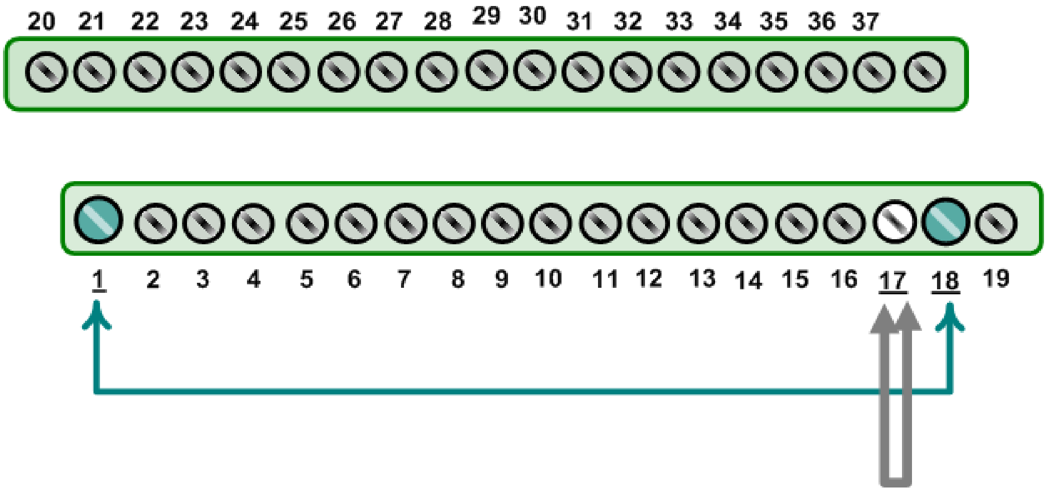
☑ A DN-37 (optional) terminal board

(Website:
http://www.icpdas.com/root/product/solutions/pc_based_io_board/daughter_boards/dn-37.html)

Step 1: Use the DN-37 (optional) to connect the CON3 on the PCI-822/826 card.



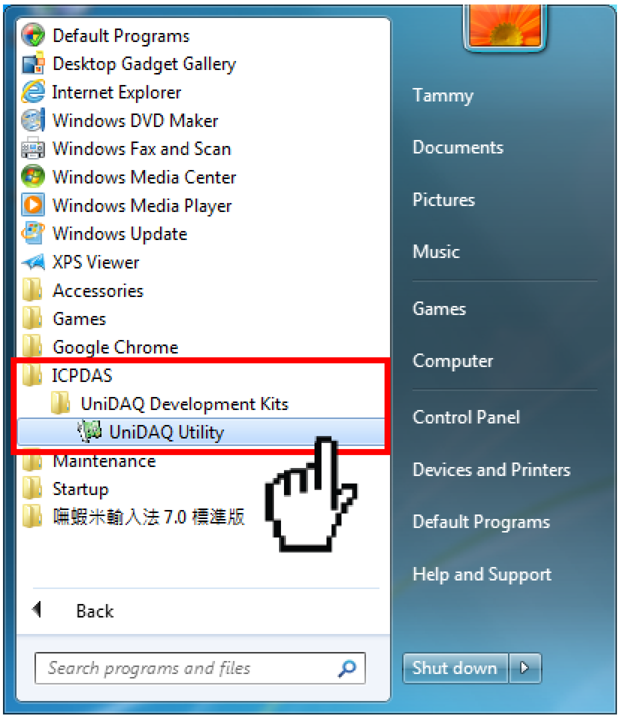
Step 2: Set the **JP1 jumper (Analog Input Type)** on the PCI-822/826 Series card to **Single-ended** mode (see Section 2.2.1 for more details), and then **connect the AO0_Out (Pin18) to the AI0 (Pin1)**, and **connect the A.GND pin (Pin17) to the A.GND pin (Pin17).**



5.2 Execute the Test Program

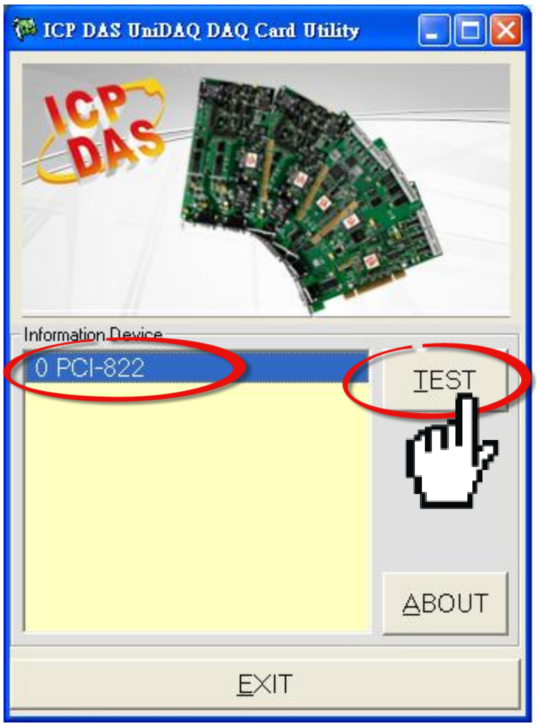
After installation, the UniDAQ Utility will be located in the default folder (C:\ICPDAS\UniDAq\Driver\). Use the procedure described below to perform the "Self-Test".

Step 1: Click the "**Start"** button point to **"All Programs"** and then click the **"ICPDAS"** folder. Point to **"UniDAQ Development Kits"**and then click the **"UniDAQ Utility**" to execute the UniDAQ Utility Program.



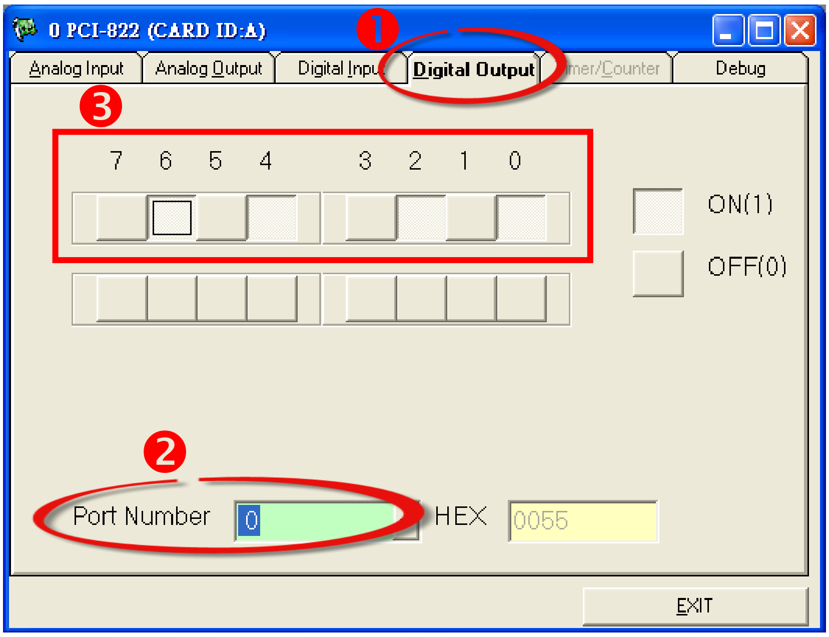
Step 2: Confirm the PCI-822/826 Series card has been successfully installed in the Host system. Note that the device numbers start from 0.

Step 3: Click the **"TEST"** button to start the test.

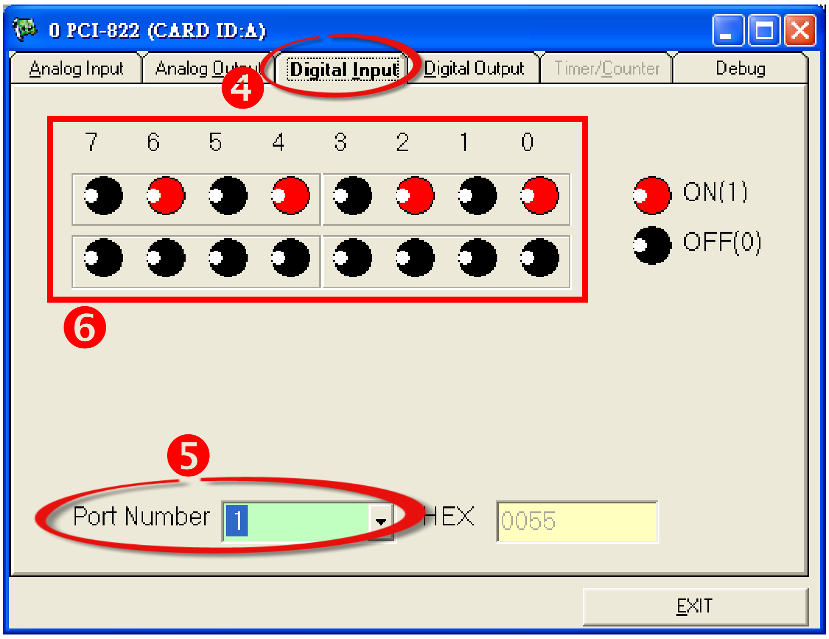


Step 4: Get DIO function test result.

1. Click the **"Digital Output"** tab.
2. Select the **"Port0"** from the **"Port Number"** drop-down options.
3. Check **channels 0, 2, 4 and 6**.



4. Click the **“Digital Input”** tab.
5. Select the **“Port1”** from the **“Port Number”** drop-down options.
6. The corresponding DI become **red** for channel 0, 2, 4, 6 of DO is **ON**.

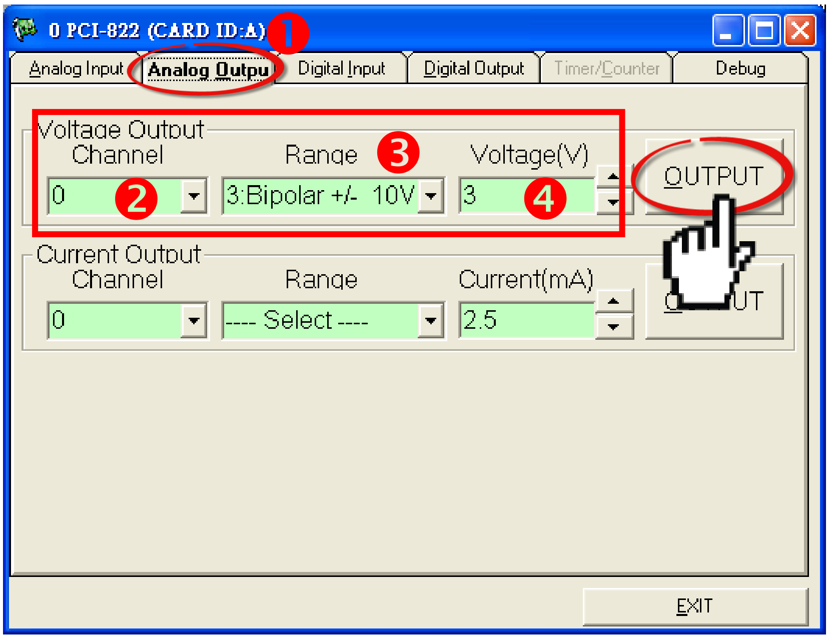


Port Number 1

HEX 0055

Step 3: Check the results of the A/D and D/A functions test.

1. Click the **“Analog Output”** tab.
2. In the Voltage Output section, select the **“0”** from the **“Channel”** drop-down menu.
3. Select **“Bipolar +/- 10 V”** from the **“Range”** drop-down menu.
4. Enter a **“voltage value (e.g., 3)”** in the **“Voltage(V)”** field.
5. Click the **“OUTPUT”** button.

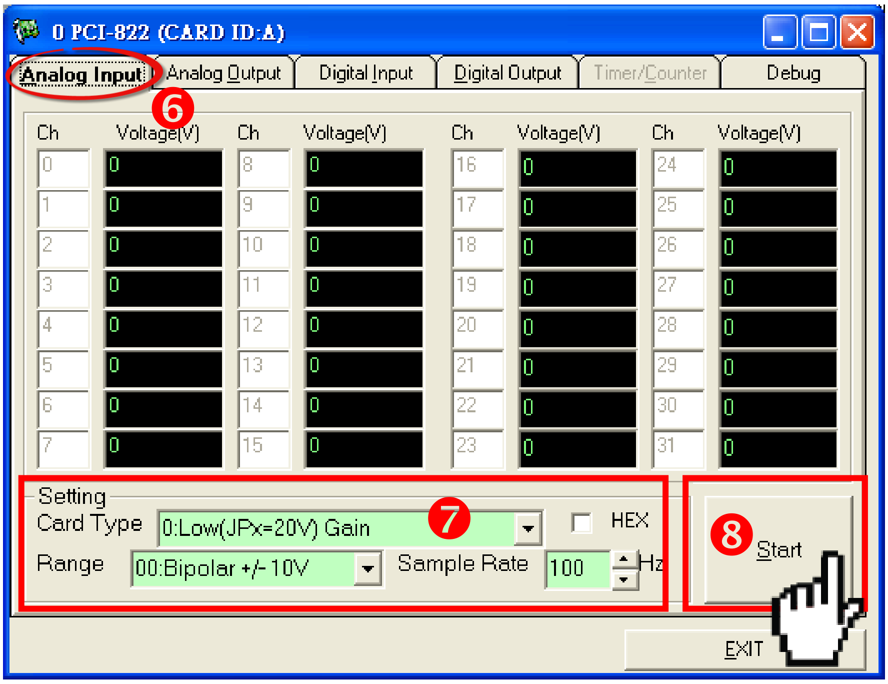


Voltage Output
Channel: 0
Range: 3:Bipolar +/- 10V
Voltage(V): 3

Current Output
Channel: 0
Range: ---- Select ----
Current(mA): 2.5

Note:
The Analog Output Range setting states depend on the JP3 and JP6 setting on the PCI-822/826 card. Refer to Section 2.2.2 JP6/JP7 Analog Output Range and Section 2.2.3 JP3/JP5 Analog Output Type for more detailed information.

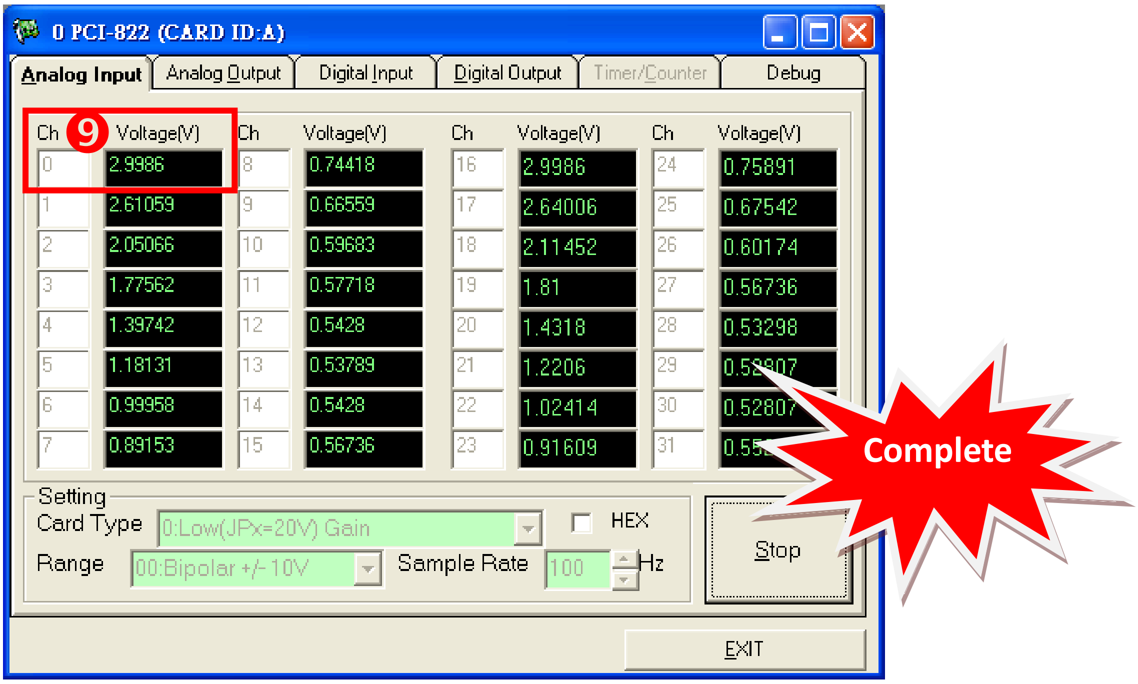
6. Click the **“Analog Input”** tab.
7. Confirm that the configuration settings are correct.
8. Click the **“Start”** button to start the test.



Ch	Voltage(V)	Ch	Voltage(V)	Ch	Voltage(V)	Ch	Voltage(V)
0	0	8	0	16	0	24	0
1	0	9	0	17	0	25	0
2	0	10	0	18	0	26	0
3	0	11	0	19	0	27	0
4	0	12	0	20	0	28	0
5	0	13	0	21	0	29	0
6	0	14	0	22	0	30	0
7	0	15	0	23	0	31	0

Setting
Card Type: 0:Low(JPx=20V) Gain □ HEX
Range: 00:Bipolar +/- 10V Sample Rate: 100 Hz

9. Check that the Analog Input value for Channel 0 is the same as the voltage value entered in Step 3. The values for other channels will be a floating number.



Ch	Voltage(V)	Ch	Voltage(V)	Ch	Voltage(V)	Ch	Voltage(V)
0	2.9986	8	0.74418	16	2.9986	24	0.75891
1	2.61059	9	0.66559	17	2.64006	25	0.67542
2	2.05066	10	0.59683	18	2.11452	26	0.60174
3	1.77562	11	0.57718	19	1.81	27	0.56736
4	1.39742	12	0.5428	20	1.4318	28	0.53298
5	1.18131	13	0.53789	21	1.2206	29	0.52807
6	0.99958	14	0.5428	22	1.02414	30	0.52807
7	0.89153	15	0.56736	23	0.91609	31	0.55

Setting
Card Type: 0:Low(JPx=20V) Gain □ HEX
Range: 00:Bipolar +/- 10V Sample Rate: 100 Hz

6 I/O Control Registers

6.1 Determining the I/O Address

During the power-on stage, the Plug and Play BIOS will assign an appropriate I/O address to each PCI-822/826 Series card installed in the system. Each PCI-822/826 Series card includes four fixed ID numbers that are used to identify the card, and are indicated below:

Model	PCI-822LU	PCI-826LU
Vendor ID	0x10B5	0xE10B5
Device ID	0x3001	0x3001
Sub-Vendor ID	0x2129	0x2129
Sub-Device ID	0x0822	0x0826

The following functions are provided for the PCI-822/826 card:

1. **PCI82x_DriverInit(&wTotalBoards)**
   This function is used to detect how many PCI-822 and/or PCI-826 cards are installed in the system, and also records the I/O resource information for the card(s) in the library. The function is implemented based on the PCI Plug and Play mechanism.

   For example:
   - wTotalBoards = 1 → there is only one PCI-822/826 Series card installed in the system
   - wTotalBoards = 2 → there are two PCI-822/826 Series cards installed in the system

2. PCI82X_GetConfigAddressSpace(wBoardNo, *wBaseAddr, *wBaseDIO, *wBaseDA, *wBaseAD, *wIrqNo, *wModelD, *wCardID)

This function is used to retrieve the I/O resource information for the PCI-822/826 Series cards installed in the system. The application will then be able to directly control all the functions related to the PCI-822/826 Series cards.

- **wBoardNo = 0 to N** → A total of N+1 PCI-822/826 Series cards are installed in the system
- **wBaseAddr, wBaseDIO, wBaseDA, wBaseDA**→ The base address of the PCI-822/826 card
- **wIrq** → The IRQ channel number allocated for the PCI-822/826 card

The following is a sample of the source code:

```
/* Step1: Detect all PCI-822&PCI-826 cards first */
wRetVal=PCI82X_DriverInit(&wTotalBoards);
printf("Threr are %d PCI-822&PCI-826 Cards in this PC\n",wBoards);

/* Step2: Save resources of all PCI-822&PCI-826 cards installed in this PC */
for (i=0; i<wBoards; i++)
    {
      PCI82X_GetConfigAddressSpace(wBoardNo,&wBaseAddr,&wBaseDIO,
                                            &wBaseDA,&wBaseAD,&wIrqNo,&wModelD,&wCardID);

    printf("\nCard%d: wBase=%x, wIrq=%x, wBaseDIO=%x"
    , i,wBaseAddr,wIrq,wBaseDIO);
    wConfigSpace[i][0]=wBaseAddress;    // save all resource of this card
    wConfigSpace[i][1]=wBaseDIO;        // save all resource of this card
    wConfigSpace[i][2]=wBaseDA;         // save all resource of this card
     wConfigSpace[i][3]=wBaseAD;        // save all resource of this card

    }

/* Step3: Control the PCI-822&PCI-826 directly */
   outpw(wBaseDIO+0x0,wDoValue);              // control the D/O states of card_0
   wDiValue=inpw(wBaseDIO+0x0);               // read the D/I states of card_0

   outpw(wBaseDIO+0x0,wDoValue);              // control the D/O states of card_1
   wDiValue=inpw(wBaseDIO+0x0);              // read the D/I states of card_1
```

6.2 I/O Address Mapping

The address of each register can be determined by simply adding the offset value to the base address of the corresponding section. More detailed descriptions of each register can be found in the following sections and also in the software manual. The following is an overview of the registers for PCI-822/826 Series cards and their functions:

Bar No.	Offset	Register Function Description	
		Read	Write
1 (DI/O)	0h	Reads the Digital I/O for PortA	Writes the Digital I/O to PortA
	4h	Reads the Digital I/O for Port B	Writes the Digital I/O to PortB
	8h	Reads from EEPROM	Writes to EEPROM
	Ch	Reads the status of the DI/O Jumper and Card ID	Sets the configuration for Port A and Port B
2 (D/A)	0h	Reads the D/A Data	Writes the D/A Data
	4h	Reads the D/A control settings	Configures the D/A control settings
	8h	N/A	Enables/Disables the D/A Channel
3 (A/D)	0h	Reads the Analog Input configuration	Configures the A/D polling control settings
	4h	Reads the A/D data	Sets the A/D trigger for polling mode
	8h	Reads the Sampling Rate	Sets the Sampling Rate
	Ch	Reads the Magic Scan counter value	Sets the Magic Scan counter value
	10h	Reads the A/D pacer control configuration	Configures the A/D pacer control settings
	14h	Reads the Base Frequency/Magic Scan control configuration	Configures the Magic scan control/Base Frequency settings
	18h	Clears the interrupts	Starts/stops Magic Scan
	1Ch	Reads the status of the interrupts	Configures the interrupt control settings

Note: The length of the register is 16-bit.

6.3 Bar 1: Digital I/O Registers

6.3.1 Read/Write 16-bit Data for Port A/B

- **(Read/Write): wBase+0+0x00 Read/Write 16-bit data for Port A**
- **(Read/Write): wBase+0+0x04 Read/Write 16-bit data for Port B**

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Data	DF	DE	DD	DC	DB	DA	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

This register provides the function for configuring the Digital Input/Output ports on the PCI-822/826 Series card. The I/O ports can be configured as either DI or a DO port, each port can be read/written by accessing its data register. The I/O ports on PCI-822/826 Series cards are 16-bit.

6.3.2 Read/Write I/O Port Selection

- **(Read/Write): wBase+0+0x0C I/O Port Selection**

Bit	1	0
Data	Port B	Port A

This register provides the function for configuring the mode for the Digital Input/Output ports on the PCI-822/826 Series card. Each I/O port can be programmed as either a Digital Input or Digital Output. *Note that all ports are initialized as Digital Input when the Host system is first turned on and that the jumpers for DIO-S2 of JP4 must be set to "Software Programmable Mode". See Section 2.2.4 for more details.*

Port x = 1 → The port is configured used as a Digital Output port
Port x = 0 → The port is configured used as a Digital Input port

6.3.3 Read the Card ID (SW1) and the DI/O Jumper (JP4) Settings

- **(Read)wBase+0x0C Read the Card ID (SW1) and the DI/O Jumper (JP4) Settings**

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Data	x	x	x	x	x	S0	S1	S2	x	x	x	x	ID3	ID2	ID1	ID0

This register is used to read the Card ID (SW1) and the DIO Jumper (JP4) settings on the PCI-822/826 Series card.

The following is an example of how to read the 4-bit Card ID:

```
wCardID = inportb(wBaseDIO+0x0C)&0xF;          /* Read the Card ID
wJumper=(inportb(wBaseDIO+0xC)>>8)&0x7;        /* Reads the DI/O Jumper (JP4) settings
```

DI/O Jumper (JP4) Settings				DIO Port Configuration		
wJumper	S0	S1	S2	JP4 DIO-S0	PA	PB
0x0	0	x	x	Soft	x	x
0x4	1	0	0	Jump	DI	DI
0x5	1	0	1	Jump	DI	DO
0x6	1	1	0	Jump	DO	DI
0x7	1	1	1	Jump	DO	DO

6.4 Bar 2: Analog Output Registers

6.4.1 Read/Write 16-bit D/A Data

- **(Read/Write)wBase+0x0 Read/Write 16-bit D/A Data**

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Data	DF	DE	DD	DC	DB	DA	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Each D/A converter provide 2-channel Analog Output. Refer to Section 6.4.2 below for details of how to select the D/A channel before writing data to the D/A converter.

6.4.2 Read/Write Analog Output Channel Selection

- **(Read/Write)wBase+0x04 Read/Write Analog Output Channel Selection**

Bit	1	0
Data	M1	M0

Before D/A data can be written, a D/A channel must be selected as the output. The following table illustrates how to select the D/A channel:

D/A Channel Selection:

M1	M0	D/A Channel
0	0	Channel 0
1	1	Channel 1

6.4.3 Enable/Disable an Analog Output Channel

- **(Write)wBase+0x08 Enable/Disable an Analog Output Channel**

Bit	1	0
Data	CH1	CH2

D/A channel must be enabled or disabled before the voltage can be output to the channel. The D/A channels are allocated as follows:

CH x = 1 → Enable the channel
CH x = 0 → Disable the channel

The following is an example of how to enable the Analog Output Channel:

```
  //Set the AO Mode
  outpw(wBaseDA+0x04,(WORD)(((wChannel/2)<<2)|(((wChannel%2)<<1)|(wChannel%2))));
  //Write data to the D/A Port
  outpw(wBaseDA +0x00,(WORD)(wValue&0xFFFF));

  //Enable the D/A Port (only for the first D/A Output)
  if((inpw(wBaseDA+0x08)&(0x1<<wChannel))==0)
{
  outpw(wBaseDA+0x08,(WORD)(inpw((WORD)( wBaseDA+0x08))|(0x1<<wChannel)));
}
```

6.5 Bar 3: Analog Input Registers

6.5.1 Read/Write the A/D Polling Configuration

- **(Read/Write)wBase+0x0** **Read/Write the A/D Polling Configuration**

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Data	F0	x	x	x	x	x	x	x	G1	G0	M5	M4	M3	M2	M1	M0

This register is used to read/set the Analog Input configuration for polling mode (software trigger). The configuration parameters include the Gain code, the Mode (single-ended or differential), and the channel number.

Clearing the FIFO (Bit F)

Write a value of 1 to F0 to clear the FIFO Data. The status of the FIFO will then be changed to empty.

Analog Input Gain Control (Polling Mode) (Bits 6 and 7)

Gain Control:
Bits 6 and 7 are used to configure the Gain Control mode for the Analog Input channels, as indicated below.

G1	G0	Gain	AI Range
0	0	0	± 10.00 V
0	1	1	± 5.00 V
1	0	2	± 2.50 V
1	1	3	± 1.25 V

Note: Only the values for G1 and G0 (Bits 6 and 7) are used for the Gain Control code.

Analog Input Channel Control (Polling Mode) (Bits 0 to 5)

Single-ended/Differential Analog Input Selection:
Bits 4 and 5 are used to configure the control (Polling) mode for the Analog Input channels, as indicated below.

M5	M4	Analog Input
0	0	x
0	1	Single-ended Input Channels 0-15
1	0	Single-ended Input Channels 16-31
1	1	Differential Input Channels 0-15

Analog Input Channel Selection:
Bits 0 to 3 are used to configure the input mode for the Analog Input channels, as indicated below.

M3	M2	M1	M0	(M5, M4)		
				(0,1)	(1,0)	(1,1)
0	0	0	0	SE Ch0	SE Ch16	DIFF Ch0
0	0	0	1	SE Ch1	SE Ch17	DIFF Ch1
0	0	1	0	SE Ch2	SE Ch18	DIFF Ch2
0	0	1	1	SE Ch3	SE Ch19	DIFF Ch3
0	1	0	0	SE Ch4	SE Ch20	DIFF Ch4
0	1	0	1	SE Ch5	SE Ch21	DIFF Ch5
0	1	1	0	SE Ch6	SE Ch22	DIFF Ch6
0	1	1	1	SE Ch7	SE Ch23	DIFF Ch7
1	0	0	0	SE Ch8	SE Ch24	DIFF Ch8
1	0	0	1	SE Ch9	SE Ch25	DIFF Ch9
1	0	1	0	SE Ch10	SE Ch26	DIFF Ch10
1	0	1	1	SE Ch11	SE Ch27	DIFF Ch11
1	1	0	0	SE Ch12	SE Ch28	DIFF Ch12
1	1	0	1	SE Ch13	SE Ch29	DIFF Ch13
1	1	1	0	SE Ch14	SE Ch30	DIFF Ch14
1	1	1	1	SE Ch15	SE Ch31	DIFF Ch15

The following is an example of how to enable the Analog Output Channel:

```
outpw(wBaseAD+0x0,0x8032);
//Software trigger mode. Sets the DIFF Input channel 2 and clears the FIFO buffer
```

6.5.2 Write the AI Software Trigger

- **(Write)wBase+0x04 Write the AI Software Trigger**

This register is used to trigger the A/D Conversion and then save the data to the FIFO buffer.

6.5.3 Read the FIFO Data

- **(Read)wBase+0x04 Read the FIFO Data**

This register is used to read Analog Input Data from the FIFO buffer.

6.5.4 Read/Write AI Pacer Sampling Rate

- **(Read/Write)wBase+0x8 Read/Write the AI Pacer Sampling Rate**

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Data	CF	CE	CD	CC	CB	CA	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

This register is used to set the internal pacer clock and then start the Analog Input in Pacer Trigger mode. The base frequency is set using the register offset 14H.The default clock value is 20 MHz.

The Sampling Rate = Base Frequency / Data

The following is an example of how to enable the Analog Output Channel:

```
    outpw(wBase+0x14,0x0001);       //Sets the Base frequency to 8 MHz
    outpw(wBase+0x8,800);           //Pacer Clock = 8 MHz / 800 = 10 KHz
```

6.5.5 Read/Write MagicScan Counter Value

- **(Write/Read)wBase+0x0C Read/Write MagicScan Counter Value**

This register is used to set/read the sampling value for MagicScan mode. The maximum value is 5000.

6.5.6 Write AI Pacer Configuration

- **(Write)wBase+0x10 Write the AI Pacer Configuration**

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Data	F0	S4	S3	S2	S1	S0	x	x	G1	G0	M5	M4	M3	M2	M1	M0

This register is used to set/read the Analog Input configuration for Pacer mode. The configuration parameters include the Gain code, the Mode (single-ended or differential), and the channel number.

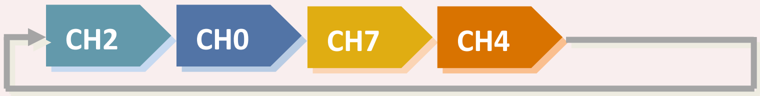
Clearing the FIFO(Bit F)

Write a value of 1 to F0 to clear the FIFO Data. The status of the FIFO will then be changed to empty.

Scan Channel Sequence Control (Bits A to E)

The following is an example of how to control the Scan Channel Sequence:

```
wChannels = 4
outpw( wBase+0x14,(0x0001|(wChannels-1)<<3));     //Set the Base Frequency and Total Scan Channels
outpw(wBase +0x08,100);                         //Set the Sampling Rate
outpw(wBase+0x10,0x8032|(0<<10));               //Set the first channel number to 2,
outpw(wBase+0x10,0x8030|(1<<10));               //Set the second channel number to 0,
outpw(wBase+0x10,0x8037|(2<<10));               //Set the third channel number to 7,
outpw(wBase+0x10,0x8034|(3<<10));               //Set the four channel number to 4,
```



Analog Input Gain Control (Pacer Mode) (Bits 6 and 7)

Gain Control:

Bits 6 and 7 are used to configure the Gain Control mode for the Analog Input channels, as indicated below.

G1	G0	Gain	AI Range
0	0	0	± 10.00 V
0	1	1	± 5.00 V
1	0	2	± 2.50 V
1	1	3	± 1.25 V

Note: Only the values for G1 and G0 (Bits 6 and 7) are used for the gain control code.

Analog Input Channel Control (Pacer Mode) (Bits 0 to 5)

Single-ended/Differential Analog Input Selection:

Bits 4 and 5 are used to configure the control (Pacer) mode for the Analog Input channels, as indicated below.

M5	M4	Analog Input
0	0	x
0	1	Single-ended Input Channel 0-15
1	0	Single-ended Input Channel 16-31
1	1	Differential Input Channel 0-15

Analog Input Channel Selection:

Bits 0 to 3 are used to configure the input mode for the Analog Input channels, as indicated below.

M3	M2	M1	M0	(M5, M4)		
				(0,1)	(1,0)	(1,1)
0	0	0	0	SE Ch0	SE Ch16	DIFF Ch0
0	0	0	1	SE Ch1	SE Ch17	DIFF Ch1
0	0	1	0	SE Ch2	SE Ch18	DIFF Ch2
0	0	1	1	SE Ch3	SE Ch19	DIFF Ch3
0	1	0	0	SE Ch4	SE Ch20	DIFF Ch4
0	1	0	1	SE Ch5	SE Ch21	DIFF Ch5
0	1	1	0	SE Ch6	SE Ch22	DIFF Ch6
0	1	1	1	SE Ch7	SE Ch23	DIFF Ch7

1	0	0	0	SE Ch8	SE Ch24	DIFF Ch8
1	0	0	1	SE Ch9	SE Ch25	DIFF Ch9
1	0	1	0	SE Ch10	SE Ch26	DIFF Ch10
1	0	1	1	SE Ch11	SE Ch27	DIFF Ch11
1	1	0	0	SE Ch12	SE Ch28	DIFF Ch12
1	1	0	1	SE Ch13	SE Ch29	DIFF Ch13
1	1	1	0	SE Ch14	SE Ch30	DIFF Ch14
1	1	1	1	SE Ch15	SE Ch31	DIFF Ch15

6.5.7 Read the Status of the FIFO/JP1/ADC

- **(Read)wBase+0x10 Read the Status of the FIFO/JP1/ADC**

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Data	FF	FH	Fe	A0	D0	x	x	x	x	x	x	x	x	x	x	x

This register is used to read the current status of the FIFO, the JP1 Jumper and the A/D Conversion. Refer to the following table for details:

Data	**Read Value**	
	0	1
D0	JP1 is set to Differential mode	JP1 is set to Single-ended mode
A0	The ADC is Ready	The ADC is Busy
FF	The FIFO is Full	The FIFO isn't Full
FH	The FIFO is Half Full	The FIFO isn't Half Full
FE	The FIFO is Empty	The FIFO isn't Empty

6.5.8 Read/Write the Base Frequency and the MagicScan Control Settings

- **(Read/Write)wBase+0x14 Read/Write the Base Frequency and the MagicScan Control Settings**

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Data	E0	x	x	x	CK3	CK2	CK1	CK0	S4	S3	S2	S1	S0	M2	M1	M0

This register is used to set/read the trigger edge settings, the base frequency, the MagicScan mode and the total number of channels to be scanned.

Note that Bits C to E are reserved.

External Trigger Edge Settings (Bit F)

Bit F used to configure the edge settings for the external trigger source, as indicated below.

E0	Trigger Edge
0	Failing Edge
1	Rising Edge

Base Frequency Settings (Bits 8 to B)

Bits 8 to B are used to configure the Base frequency settings, as indicated below.

CK3	CK2	CK1	CK0	Base Frequency
0	0	0	0	8 MHz
0	0	0	1	4 MHz
0	0	1	0	2 MHz
0	0	1	1	1 MHz
0	1	0	0	500 kHz
0	1	0	1	250 kHz
0	1	1	0	125 kHz
0	1	1	1	62.5 kHz

Total Number of Channels to be Scanned for MagicScan (Bits 3 to 7)

Bits 3 to 7 are used to set the total number of channels to be scanned using the MagicScan function (multi-channel scan). Set “0” to scan a single channel, set “1” to scan two channels. The maximum number of channels that can be scanned is 31.

MagicScan Mode Settings (Bits 0 to 2)

Bits 0 to 2 are used to set the MagicScan mode.

M2	M1	M0	Mode
0	0	0	Polling
0	0	1	Pacer
0	1	0	Pacer and Down Counter
0	1	1	Post Trigger Down Counter
1	0	0	Middle Trigger Down Counter
1	0	1	Pre Trigger Down Counter

6.5.9 Start/Stop MagicScan

- **(Write)wBase+0x18 Start/Stop MagicScan**

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Data	Start	Stop	x	x	x	x	x	x	x	x	x	x	x	x	x	x

This register is used to start or stop the AI Pacer or the MagicScan function, as indicated below.

Bit F	1	Enable Pacer START
	0	Disable Pacer START
Bit E	1	Enable Pacer STOP
	0	Disable Pacer STOP

6.5.10 Clear the Interrupt and External Trigger

- **(Read)wBase+0x18 Clear the Interrupt and External Trigger**

This register is used to clear the interrupt and the external trigger.

6.5.11 Read/Write the Interrupt Control/Status

- **(Read/Write)wBase+0x1C Read/Write the Interrupt Control/Status**

Bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Data	x	x	x	x	x	x	x	x	x	x	x	x	A0	E0	FH	FF

The PCI interrupt control register (1Ch) is used to control the interrupt sent to the system, and supports four events in the ISR.

Interrupt Settings (Bits 0 to 3)

1. Sends the interrupt after the ADC stops

Bit 3	A/D Stop Interrupt
0	Disable
1	Enable

2. Reads the external trigger signal and then sends the interrupt

Bit 2	External Trigger Interrupt
0	Disable
1	Enable

3. Send the interrupt when the FIFO is half full

Bit 1	FIFO-Half Interrupt
0	Disable
1	Enable

4. Sends the interrupt when the FIFO is full

Bit 0	FIFO-Full Interrupt
0	Disable
1	Enable

The following is an example of how to enable the FIFO Half-full and External Trigger Interrupts:

```
    outpw(wBase+0x1C,0x5)    //Enables the FIFO Half-full and External Trigger Interrupts
```

Analog Input Application Case Study

- **Polling (Single Channel + Software Trigger)**

```
//Set the Gain/Channel and clear the FIFO
outpw(wBase+0x00,0x8030);

//Set the Software AI Trigger
outpw(wBase+0x04,0x0000);
```

- **Pacer (Single-Channel + FIFO-Half Full Interrupt + Internal Pacer Trigger)**

```
//Disable the Pacer Trigger
outpw(wBase+0x18,0x4000);

//Disable the Interrupt
outpw(wBase+0x1C,0x0000);

//Set the Gain/Channel and clear the FIFO
outpw(wBase+0x10,0x8030);

//Set the Base Frequency to 8 MHz
outpw(wBase+0x14,0x0001);

//Set the Frequency to 8 MHz/200 = 40 KHz
//Set the Total Number of Scan Channels to 1
outpw(wBase+0x08,0xC8);

//Enable the Interrupt and set to FIFO-Half Full
outpw(wBase+0x1C,0x8002);

//Start the Pacer Trigger
outpw(wBase+0x18,0x8000);
```

- **Channel Scan [Magic Scan] (Multi-channel + FIFO-Half Full Interrupt + Internal Pacer Trigger)**

```
//Disable the Pacer Trigger
outpw(wBase+0x18,0x4000);

//Disable the Interrupt
outpw(wBase+0x1C,0x0000);

//Set the Base Frequency to 8 MHz
//Set the Total Number of scan Channels to 3
outpw( wBase+0x14,(0x0001|(3-1)<<3));

// Set the Frequency to 8 MHz/200 = 40 KHz
outpw(wBase +0x08,0xC8);

//Set the channel number for the first channel to 2
outpw(wBase+0x10,0x8032|(0<<10));

//Set the channel number for the second channel to 0
outpw(wBase+0x10,0x8030|(1<<10));

//Set the channel number for the third channel to 7
outpw(wBase+0x10,0x8037|(2<<10));

// Enable the Interrupt and set it to FIFO-Half Full
outpw(wBase+0x1C,0x8002);

// Start the Pacer Trigger (Start MagicScan)
outpw(wBase+0x18,0x8000); utpw(wBase+0x24,0);
```

- **External Post-Trigger (Multi-channel + External Trigger)**

```
//Disable the Pacer Trigger
outpw(wBase+0x18,0x4000);

//Disable the Interrupt
outpw(wBase+0x1C,0x0000);

//Set the Base Frequency to 8 MHz
//Set the Total Number of Scan Channels to 3
//Set the Trigger Edge to Rising and Enable External Trigger Mode
outpw( wBase+0x14,(0x8003|2<<3));

//Set the Frequency to 8 MHz/200 = 40 KHz
outpw(wBase +0x08,0xC8);

//Set the first channel number to 2
outpw(wBase+0x10,0x8032|(0<<10));

//Set the second channel number to 0
outpw(wBase+0x10,0x8030|(1<<10));

//Set the third channel number to 7
outpw(wBase+0x10,0x8037|(2<<10));
```

7 Calibration

Each PCI-822/826 Series card is already fully calibrated when shipped from the factory, including the calibration coefficients that are stored in the onboard EEPROM, so it is usually unnecessary to calibrate the board again unless the accuracy is lost. The procedure described below provides a method that allows the board to be calibrated for a more precise application of voltages within the system, so that the correct voltages for your field connection can be achieved. This calibration method ensures that the effects of voltage drops caused by IR loss in the cable and/or the connector can be eliminated.

Before beginning calibration, ensure that a precise multi-meter is available. Note that the calibrated values for the Analog Input/Output channels are stored within three words in the address of the EEPROM, as shown in the table below.

The calibration values stored in the EEPROM address are as follows:

EEPROM Address		**D/A Calibration**	
		CH 0	CH 1
Bipolar	-10 V	0	8
	+10 V	1	9
Unipolar	+0 V	2	10
	+10 V	3	11
Bipolar	-5 V	4	12
	+5 V	5	13
Unipolar	+0 V	6	14
	+5 V	7	15

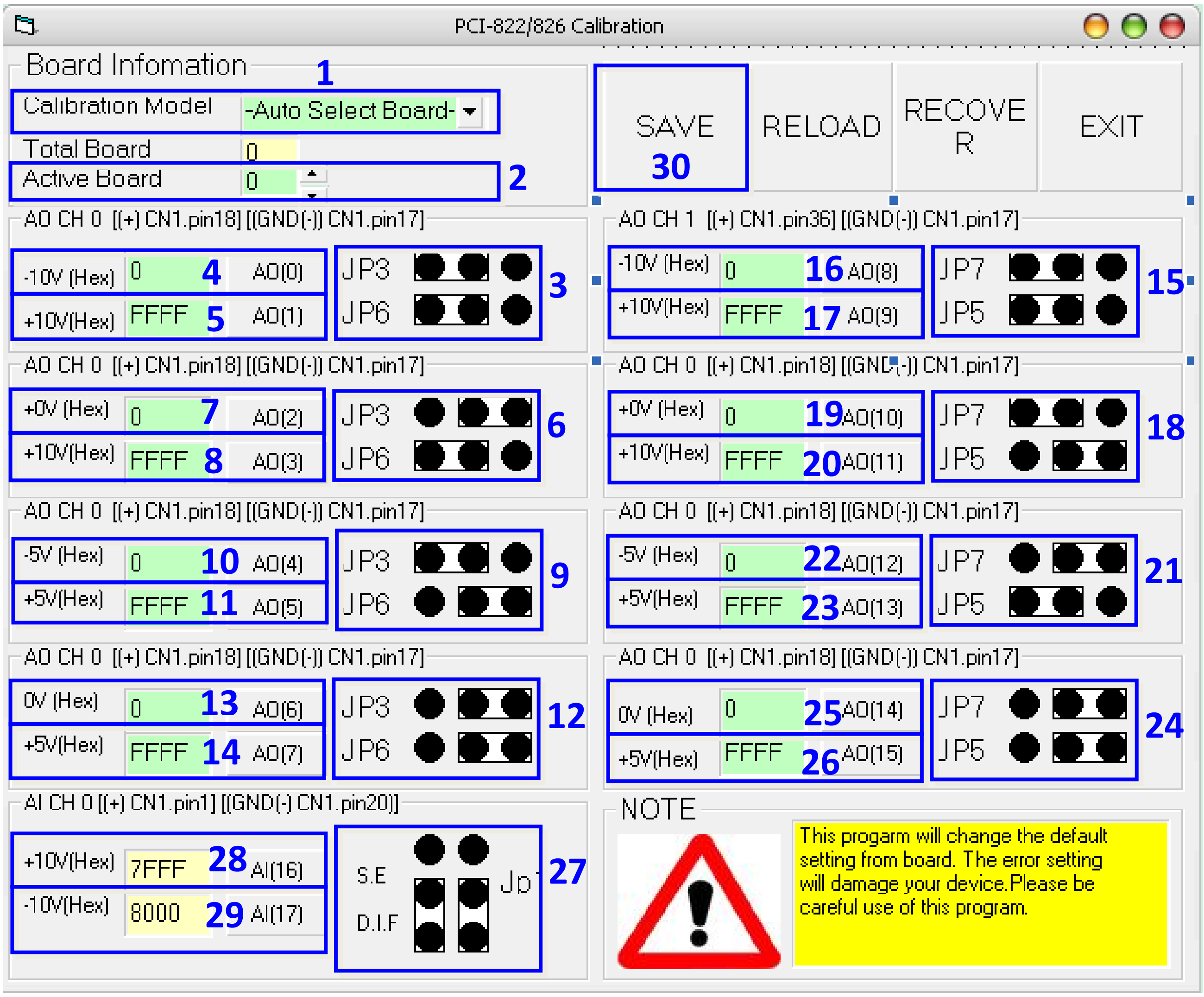
EEPROM Address		**D/A Calibration**
		ADC 0
Bipolar	+10 V	16
	-10 V	17

Warning:

*It is recommended that calibration is **not** performed until the process is fully understood.*

***Note:** Ensure that a stable calibration source is used. An unstable calibration source will cause calibration errors and will affect the accuracy of the data acquisition.*

The following is a demonstration of how to perform the calibration procedure:



Calibration Model: -Auto Select Board-
Total Board: 0
Active Board: 0

AO CH 0 [(+) CN1.pin18] [(GND(-)) CN1.pin17]
-10V (Hex): 0 AO(0)
+10V(Hex): FFFF AO(1)

AO CH 1 [(+) CN1.pin36] [(GND(-)) CN1.pin17]
-10V (Hex): 0 AO(8)
+10V(Hex): FFFF AO(9)

AO CH 0 [(+) CN1.pin18] [(GND(-)) CN1.pin17]
+0V (Hex): 0 AO(2)
+10V(Hex): FFFF AO(3)

AO CH 0 [(+) CN1.pin18] [(GND(-)) CN1.pin17]
+0V (Hex): 0 AO(10)
+10V(Hex): FFFF AO(11)

AO CH 0 [(+) CN1.pin18] [(GND(-)) CN1.pin17]
-5V (Hex): 0 AO(4)
+5V(Hex): FFFF AO(5)

AO CH 0 [(+) CN1.pin18] [(GND(-)) CN1.pin17]
-5V (Hex): 0 AO(12)
+5V(Hex): FFFF AO(13)

AO CH 0 [(+) CN1.pin18] [(GND(-)) CN1.pin17]
0V (Hex): 0 AO(6)
+5V(Hex): FFFF AO(7)

AO CH 0 [(+) CN1.pin18] [(GND(-)) CN1.pin17]
0V (Hex): 0 AO(14)
+5V(Hex): FFFF AO(15)

AI CH 0 [(+) CN1.pin1] [(GND(-) CN1.pin20)]
+10V(Hex): 7FFF AI(16)
-10V(Hex): 8000 AI(17)

This progarm will change the default setting from board. The error setting will damage your device.Please be careful use of this program.

The following pin assignments are used in the calibration process.

1. D/A Ch0 (+): CON3.pin 18
2. A.GND(+): CON3.pin 1
3. D/A Ch1 (+): CON3.pin 36
4. A/D Ch0 (-): CON3.pin 21

To calibrate the PCI-822/826 Series card, follow the procedure outlined below.

Step	Description	Step	Description
1	Select the calibration model for the board	16	(1) Enter the D/A data (2) Click the **A0(8)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **-10 V**
2	Select the number of the active board	17	(1) Enter the D/A data (2) Click the **A0(9)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **10 V**
3	Set jumper JP3 to Bipolar and jumper JP6 to 10 V	18	Set jumper JP5 to Unipolar and jumper JP7 to 10 V
4	(1) Enter the D/A data (2) Click the **A0(0)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **-10 V**	19	(1) Enter the D/A data (2) Click the **A0(10)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **0 V**
5	(1) Enter the D/A data (2) Click the **A0(1)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **10 V**	20	(1) Enter the D/A data (2) Click the **A0(11)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **10 V**
6	Set jumper JP3 to Unipolar and jumper JP6 to 10 V	21	Set jumper JP5 to Bipolar and jumper JP7 to 5 V
7	(1) Enter the D/A data (2) Click the **A0(2)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **0 V**	22	(1) Enter the D/A data (2) Click the **A0(12)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **-5 V**
8	(1) Enter the D/A data (2) Click the **A0(3)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **10 V**	23	(1) Enter the D/A data (2) Click the **A0(13)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **5 V**
9	Set jumper JP3 to Bipolar and jumper JP6 to 5 V	24	Set jumper JP5 to Unipolar and jumper JP7 to 5 V
10	(1) Enter the D/A data (2) Click the **A0(4)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **-5 V**	25	(1) Enter the D/A data (2) Click the **A0(14)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **0 V**
11	(1) Enter the D/A data (2) Click the **A0(5)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **5 V**	26	(1) Enter the D/A data (2) Click the **A0(15)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **5 V**
12	Set jumper JP3 to Unipolar and jumper JP6 to 5 V	27	Set jumper J1 to D.I.F.F.
13	(1) Enter the D/A data (2) Click the **A0(6)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **0 V**	28	Use A/D CH0 to measure the **10 V** Voltage then click the **AI(16)** button
14	(1) Enter the D/A data (2) Click the **A0(7)** button (3) Repeat items 1 and 2 until the D/A output voltage for CH0 is **5 V**	29	Use A/D CH0 to measure the **-10 V** Voltage then click the **AI(17)** button
15	Set jumper JP5 to Bipolar and jumper JP7 to 10 V	30	Click the Save button to complete the calibration

Appendix

A1. DOS Library Function Description

All of the functions provided for PCI-822/826 card are listed in Appendixes A1-2 to A1-5 below. This list of functions is expanded on in the text that follows. However, in order to provide a clear and simplified description for the functions, the attributes of the input and output parameters for each function are indicated as [input] and [output] respectively, as shown in following table. Furthermore, the error codes for all functions supported by the PCI-822/826 Series card are also provided in Appendix A1-1.

Keyword	Parameter must be set before calling the function	Data/values from this parameter is retrieved after calling the function
[Input]	Yes	No
[Output]	No	Yes
[Input, Output]	Yes	Yes

A1-1: Error Code Definitions

Error Code	Error ID	Description
0	NoError	The command was OK
1	DriverHandleError	There was an error opening the device driver
2	DriverCallError	An error occurred while calling the driver functions
3	FindBoardError	The board cannot be found on the system
4	TimeOut	A Timeout occurred
5	ExceedBoardNumber	The board number was invalid. The valid range is 0 to (TotalBoard - 1)
6	NotFoundBoard	The board cannot be detected on the system
7	InvalidChannel	The channel number was invalid
8	AIQueueError	A driver buffer error occurred
9	FIFOError	A FIFO error occurred on the device
10	InvalidEEPBlock	The EEPROM Block was invalid
11	InvalidEEPAddr	The EEPROM Address was invalid
12	InvalidCfgCode	The Gain Code was invalid

A1-2: Driver Functions

PCI82X_DriverInit

This function is used to detect all PCI-822/826 cards installed in the system, and is implemented based on the PCI Plug and Play mechanism. Once all the PCI-822/826 Series cards installed in the system are detected, the resource information will be saved to the library.

- **Syntax:**
  WORD **PCI82X_DriverInit**(WORD *wBoards);

- **Parameters:**

  wBoardNo
  [Output] The number of boards found on the Host PC.

- **Returns:**
  Refer to Appendix A1-1 Error Code Definitions.

PCI82X_DriverClose

This function is used to release the driver resources currently in use by the PCI-822/826 Series card.

- **Syntax:**
  WORD **PCI82X_DriverClose**(void);

- **Parameters:**
  None

- **Returns:**
  Refer to Appenidx A1-1 Error Code Definitions.

PCI82X_GetConfigAddressSpace

This function is used to retrieve the resource information found for all PCI-822/826 Series cards installed in the system. The application can then be used to directly control all of the functions related to the PCI-822/826 Series cards.

- **Syntax:**

```
WORD    PCI82X_GetConfigAddressSpace(WORD wBoardNo,
                                     WORD *wBaseAddr,
                                     WORD *wBaseDIO,
                                     WORD *wBaseDA,
                                     WORD *wBaseAD,
                                     WORD *wIrqNo,
                                     WORD*wModeID,
                                     WORD *wCardID);
```

- **Parameters:**

wBoardNo

[Input] The board number (Base 0)

wBaseAddr

[Output] The section 0 base address for the board

wBaseDIO

[Output] The section 1 base address for the board

wBaseDA

[Output] The section 2 base address for the board

wBaseAD

[Output] The section 3 base address for the board

wIrqNo

[Output] The IRQ number being used by the board *wModeID*

[Output] The Model ID number (Sub-Device ID, see Section 6.1 for details)

0x822 is PCI-822

0x826 is PCI-826

wCardID

[Output] The Card ID number as configured using DIP Switch SW1 (see Section 2.3 for details)

- **Returns:**

Refer to Appendix A1-1 Error Code Definitions.

A1-3: Digital I/O Functions

PCI82X_SetDIOMode32

This function is used to configure the mode to be used for the I/O ports (Port A and Port B) on a specified board.

- **Syntax:**

```
WORD    PCI82X_SetDIOMode32(WORD wBoardNo,
                            WORD wDirection);
```

- **Parameters:**

wBoardNo

[Input] The board number (Base 0)

wDirection

[Input] Configures the mode to be used for Digital I/O ports A and B as either Digital Input or Digital Output, Bit 0 is Port A, and Bit 1 is Port B, as indicated below.

wDirection	Bit 1	Bit 0
0	Port B Input	Port A Input
1	Port B Output	Port A Output

- **Returns:**

Refer to Appendix A1-1 Error Code Definitions.

PCI82X_WriteDO

This function is used to send data to a specific I/O port on a specified board in 16-bit integer format.

- **Syntax:**

```
WORD    PCI82X_WriteDO(WORD wBoardNo,
                       WORD wPortNo,
                       WORD wValue);
```

- **Parameters:**

wBoardNo

[Input] The board number (Base 0)

wPortNo

[Input] The Port number, where 0 denotes Port A, and 1 denotes Port B

wValue

[Input] The data to be sent to the specified I/O port in 16-bit integer format

- **Returns:**

Refer to Appendix A1-1 Error Code Definitions.

PCI82X_ReadDI

This function is used to data from a specific I/O port on a specified board in 16-bit integer format.

- **Syntax:**

```
WORD    PCI82X_ReadDI(WORD wBoardNo,
                      WORD wPortNo,
                      WORD *wValue);
```

- **Parameters:**

wBoardNo

[Input] The board number (Base 0)

wPortNo

[Input] The Port number to be read, where 0 denotes Port A, and 1 denotes Port B

**wValue*

[Output] The data received from the I/O port in 16-bit integer format

- **Returns:**

Refer to Appendix A1-1 Error Code Definitions.

A1-4: Analog Output Functions

PCI82X_WriteAO

This function is used to output the voltage value to a specific D/A channel on a specified board in float format.

- **Syntax:**

```
WORD    PCI82X_WriteAO(WORD wBoardNo,
                       WORD wChannel,
                       WORD wConfig,
                       float fValue);
```

- **Parameters:**

wBoardNo

[Input] The board number (Base 0)

wChannel

[Input] The D/A channel number

wConfig

[Input] Configures the voltage range for the Analog Output, as indicated below.

wConfig	Min.	Max.
0	0 V	+5 V
1	-5 V	+5 V
2	0 V	+10 V
3	-10 V	+10 V

fValue

[Input] The data sent to the D/A channel in float format

- **Returns:**

Refer to Appendix A1-1 Error Code Definitions.

PCI82X_WriteAOH

This function is used to send data to a specific D/A channel on a specified board in 16-bit integer format.

- **Syntax:**

```
WORD    PCI82X_WriteAO(WORD wBoardNo,
                       WORD wChannel,
                       WORD wValue);
```

- **Parameters:**

wBoardNo

[Input] The board number (Base 0)

wChannel

[Input] The D/A channel number

wValue

[Input] The data sent to the D/A channel in 16-bit integer format

- **Returns:**

Refer to Appendix A1-1 Error Code Definitions.

A1-5: Analog Input Functions

PCI82X_PollingAI

This function is used to perform multiple A/D conversions on a specified board using the polling method, and then returns the A/D data as a float value.

- **Syntax:**

```
WORD    PCI82X_PollingAI(WORD wBoardNo,
                         WORD wChannel,
                         WORD wConfigCode,
                         DWORD dwDataCount,
                         float fValue[]);
```

- **Parameters:**

wBoardNo
[Input] The board number (Base 0)

wChannel
[Input] The A/D channel number

wConfigCode
[Input] Configures the voltage range for the Analog Input, as indicated below.

wConfigCode	AI Range
0	± 10.00 V
1	± 5.00 V
2	± 2.50 V
3	± 1.25 V

dwDataCount
[Input] The number of A/D conversions that will be performed

fValue[]
[Output] An array containing the A/D data received from the Analog Input channel in float format

- **Returns:**

Refer to Appendix A1-1 Error Code Definitions.

PCI82X_PollingAIH

This function is used to perform multiple A/D conversions on a specified board using the polling method, and then returns the A/D data in 16-bit integer format.

➢ Syntax:

```
WORD    PCI82X_PollingAIH(WORD wBoardNo,
                          WORD wChannel,
                          WORD wConfigCode,
                          DWORD dwDataCount,
                          WORD wValue[]);
```

➢ Parameters:

wBoardNo

[Input] The board number (Base 0)

wChannel

[Input] The A/D channel number

wConfigCode

[Input] Configures the voltage range for the Analog Input, as indicated below.

wConfigCode	AI Range
0	± 10.00 V
1	± 5.00 V
2	± 2.50 V
3	± 1.25 V

dwDataCount

[Input] The number of A/D conversions that will be performed

wValue[]

[Output] An array containing the A/D data received from the Analog Input channel in 16-bit integer format

➢ Returns:

Refer to Appendix A1-1 Error Code Definitions.

PCI82X_StartAI

This function is used to start pacer trigger operation on a specified board and returns to the caller immediately. The A/D data is stored in the driver buffer and either the PCI82X_GetAIBuffer() or the PCI82X_GetAIBufferH() function must be called to retrieve the A/D data. Refer to the PCI82X_StopAI() function for details of how to stop the pacer trigger operation.

- **Syntax:**

```
WORD    PCI82X_StartAI(WORD wBoardNo,
                       WORD wChannel,
                       WORD wConfig,
                       float fSamplingRate,
                       DWORD dwDataCount);
```

- **Parameters:**

wBoardNo

[Input] The board number (Base 0)

wChannel

[Input] The A/D channel number

wConfig

[Input] Configures the voltage range for the Analog Input, as indicated below.

wConfig	AI Range
0	± 10.00 V
1	± 5.00 V
2	± 2.50 V
3	± 1.25 V

fSamplingRate

[Input] Sets the sampling rate in Hz

dwDataCount

[Input] The Analog Input data value

- **Returns:**

Refer to Appendix A1-1 Error Code Definitions.

PCI82X_StartAIScan

This function is used to start MagicScan operation on a specified board and returns to the caller immediately. The A/D data is stored in the driver buffer and either the PCI82X_GetAIBuffer() or the PCI82X_GetAIBufferH() function must be called to retrieve the A/D data. Refer to the PCI82X_StopAI() function for details of how to stop the MagicScan operation.

- **Syntax:**

```
WORD    PCI82X_StartAIScan(WORD wBoardNo,
                           WORD wChannels,
                           WORD wChannelList[],
                           WORD wConfigList[],
                           float fSamplingRate,
                           DWORD dwDataCountPerChannel);
```

- **Parameters:**

wBoardNo

[Input] The board number (Base 0)

wChannels

[Input] The total number of A/D channels to be scanned

wChannelList[]

[Input] An array containing the scan channel sequence

wConfigList[]

[Input] An arrary containing the voltage range configuration for each Analog Input channel to be scanned, as indicated below.

wConfig	AI Range
0	± 10.00 V
1	± 5.00 V
2	± 2.50 V
3	± 1.25 V

fSamplingRate

[Input] Sets the total sampling rate in Hz

dwDataCountPerChannel

[Input] The A/D data values for every Analog Input channels

Returns:

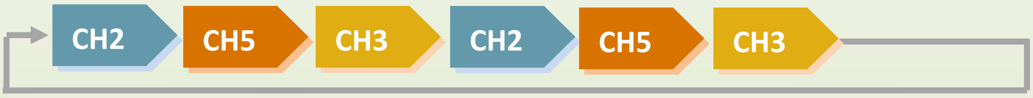
Refer to Appendix A1-1 Error Code Definitions.

Example:

```
wChannels = 3                //Scan 3 channels
wChannelList[0] = 2          //The first channel to be scanned is AI Channel 2
wChannelList[1] = 5          //The second channel to be scanned is AI Channel 5
wChannelList[2] = 3          //The third channel to be scanned is AI Channel 3

wConfigList[0] = 2           //The input range for the first channel is +/- 2.5 V
wConfigList[1] = 0           //The input range for the second channel is +/- 10 V
wConfigList[2] = 1           //The input range for the third channel is +/- 5 V

fSamplingRate = 30000         //The total sampling rate =30 kHz,
                              //Therefore, each channel is (fSamplingRate/wChannels) Hz
                                →30 kHz/3 = 10 kHz
dwDataCountPerChannel = 5
```



Refer to the following table for details of where the data is stored in the driver buffer:

0	CH2	Val0
1	CH5	Val0
2	CH3	Val0
3	CH2	Val1
4	CH5	Val1
5	CH3	Val1

PCI82X_GetAIBuffer

This function is used to retrieve the A/D data from the driver buffer of a specified board in float format.

- **Syntax:**

```
WORD    PCI82X_GetAIBuffer(WORD wBoardNo,
                           DWORD dwDataCount,
                           float fValue[]);
```

- **Parameters:**

wBoardNo
[Input] The Board number (Base 0)

dwDataCount
[Input] The Analog Input data value

fValue[]
[Output] An array containing the A/D data received from the Analog Input channels in float format

- **Returns:**

Refer to Appendix A1-1 Error Code Definitions.

PCI82X_GetAIBufferH

This function is used to retrieve the A/D data from the driver buffer of a specified board in 16-bit integer format.

- **Syntax:**

```
WORD    PCI82X_GetAIBufferH(WORD wBoardNo,
                            DWORD dwDataCount,
                            WORD hValue[]);
```

- **Parameters:**

wBoardNo

[Input] The board number (Base 0)

dwDataCount

[Input] The A/D data value for every Analog Input channel

hValue[]

[Output] An array containing the A/D data received from the Analog Input channels in 16-bit integer format

- **Returns:**

Refer to Appendix A1-1 Error Code Definitions.

PCI82X_StopAI

This function is used to stop all currently active MagicScan or Pacer trigger functions on a specified board.

- **Syntax:**

```
WORD    PCI82X_StopAI(WORD wBoardNo);
```

- **Parameters:**

wBoardNo

[Input] The board number (Base 0)

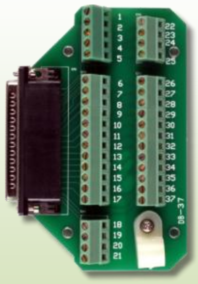
- **Returns:**

Refer to Appendix A1-1 Error Code Definitions.

A2. Daughterboards

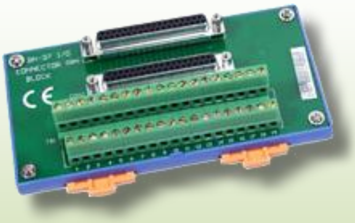
DB-37

The DB-37 is a general-purpose daughterboard for devices that include 37-pin D-sub connectors, and is designed for easy wiring.

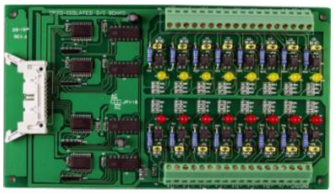


DN-37

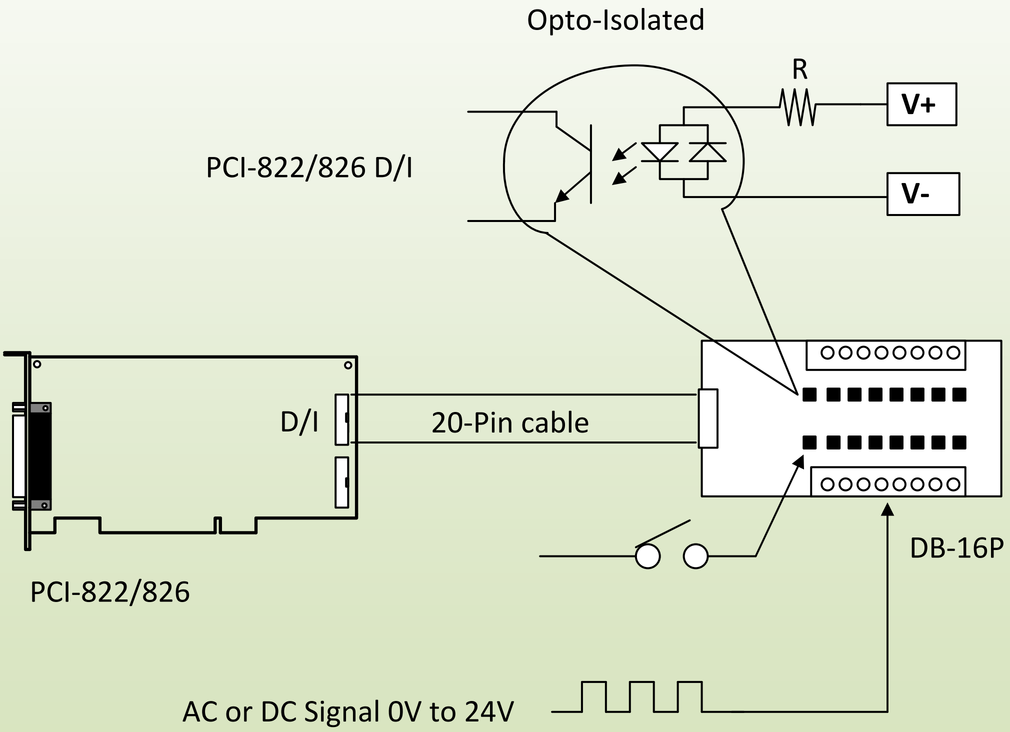
The DN-37 is a general-purpose daughterboard for devices that include 37-pin D-sub connectors. It is DIN-Rail mountable and is designed for easy wiring.



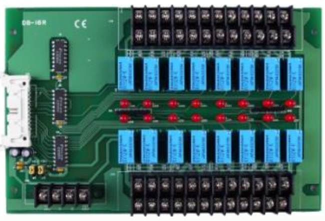
DB-16P Isolated Digital Input Daughterboard



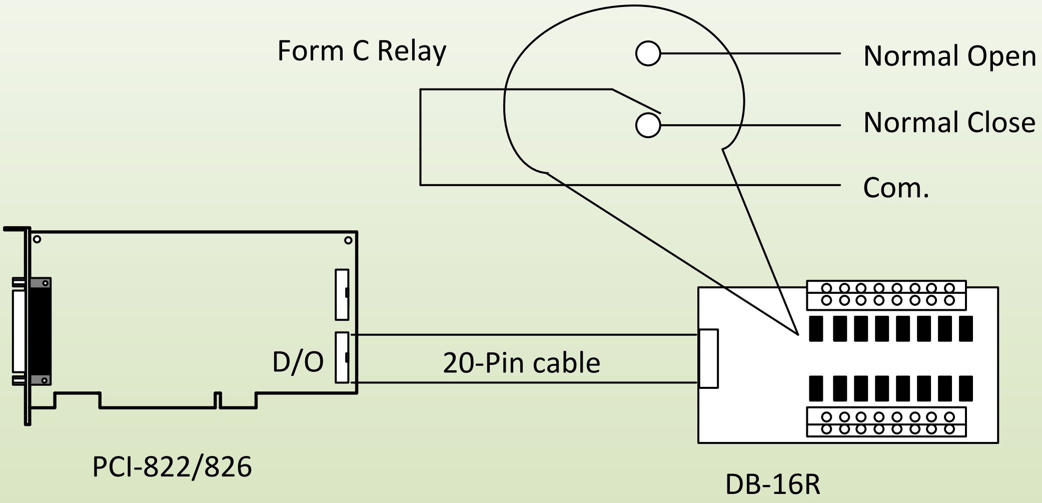
The DB-16P is a 16-channel isolated Digital Input daughterboard with optically isolated inputs that consist of a bi-directional optocoupler with a resistor to allow current sensing. The DB-16P can be used to sense DC signals from TTL levels up to 24 V, or can be used to sense a wide range of AC signals. This board can also be used to isolate the Host system from high common-mode voltages, ground loops and transient voltage spikes that can often occur in harsh industrial environments.



DB-16R Relay Output Daughterboard



The DB-16R is a 16-channel Relay Output daughterboard that provides 16 Form C Relay Output channels which enable the efficient switching of loads via programmable controls. Both the connectors and the functionally are compatible with 785 Series boards an industrial-type terminal block. The Relay channels are powered by applying a 5 V signal to the appropriate channel via a 20-pin flat cable connector. The DB-16R includes 16 LEDs, one for each Relay channel, which are illuminated when the associated relay is activated. To prevent the power supply on the Host system from being overloaded, the DB-16R includes a screw terminal to allow an external power supply to be connected.



Note: The load for the Relay control can be up to 0.5 A @ 110 V_{AC} or 1A @ 24 V_{DC}