NuDAQ
PCI-9812/10
20MHz Simultaneous 4-CH
Analog Input Card
Users' Guide

Recycled Paper

Manual Rev. 2.40: April 11, 2003

Part No. 50-11116-201

Trademarks

Getting service from ADLINK

Customer Satisfaction is the most important priority for ADLINK Tech Inc. If you need any help or service, please contact us.

ADLINK Technology Inc.			
Web Site	http://www.adlinktech.com		
Sales & Service	Service@adlinktech.com		
Technical Support	NuDAQ + USBDAQ + PXI	nudaq@adlinktech.com	
	Automation	automation@adlinktech.com	
	NuIPC	nuipc@adlinktech.com	
	NuPRO / EBC	nupro@adlinktech.com	
TEL	+886-2-82265877	FAX	+886-2-82265717
Address	9F, No. 166, Jian Yi Road, Chungho City, Taipei, 235 Taiwan.		

Please email or FAX us of your detailed information for a prompt, satisfactory and constant service.

Detailed Company Information			
Company/Organization			
Contact Person			
E-mail Address			
Address			
Country			
TEL		FAX	
Web Site			
Questions			
Product Model			
Environment to Use	OS: Computer Brand: M/B: CPU: Chipset: BIOS: Video Card: Network Interface Card: Other:		
Detail Description			
Suggestions to ADLINK			

CONTENTS

Outline of Chapters

This manual is designed to help you use the PCI-9812/10. The manual describes how to modify various settings on the PCI-9812/10 card to meet your requirements. It is divided into eight chapters:

Chapter 1, **"Introduction",** gives an overview of the product features, applications, and specifications.

Chapter 2, **"Installation",** describes how to install the PCI-9812/10. In addition, the layout of PCI-9812/10 is shown.

Chapter 3, **"Signal Connection",** describes the connectors' pin assignment and how to connect the outside signal and devices with the PCI-9812/10.

Chapter 4, **"Registers Structure & Format",** describes the details of register format and structure of the PCI-9812/10, this information is very important for the programmers who want to control the hardware by low-level programming.

Chapter 5, **"Operation Theorem",** describes how to operate the PCI-9812/10. The A/D functions are introduced. Also, some programming concepts are specified.

Chapter 6, **"C/C++ Software Library",** describes high-level programming interface in C/C++ language. It helps programmer to control PCI-9812/10 in high-level language style.

Chapter 7, **"Calibration",** describes how to calibrate the PCI-9812/10 for accurate measurement.

Chapter 8, **"Software Utility",** describes how to run the utility program included in the software CD.

1

Introduction

PCI-9812/10 is an advanced-performance data acquisition card based on 32-bit PCI Bus architecture. The maximum sampling rate of PCI-9812/10 is up to 20M samples per second, with an emphasis on continuous, non-stop, high-speed, and streaming of A/D samples to host memory. The high performance design and state-of-the-art technology make this card ideal for DSP, FFT, digital filtering, and image processing applications.

1.1 Features

PCI-9812 PCI Bus Advanced Data Acquisition Card is designed with the following advanced features:

- 32-bit PCI-Bus, Bus Mastering DMA data transfer
- 12-bit (9812) or 10-bit (9810) analog input resolution
- On-board 32K words (samples) A/D FIFO memory
- Up to 20MHz A/D sampling rate
- 4 single-ended analog input channels
- Bipolar input signals
- 4 A/D converters simultaneously sampling
- Five A/D trigger modes: software trigger, pre-trigger, Post-trigger, middle trigger, and delay trigger

1.2 Applications

- IF and BASEBAND Digitization
- Ultrasound Imaging
- Gamma Cameras
- Test Instrument
- CCD Imaging
- Video Digitizing

1.3 Specifications

Analog Input (A/D)

- Converters: B.B. ADS800 series
- Input Channels: 4 single-ended
- Resolution: 12-bit (9812), 10-bit (9810)
- Input Range: Bipolar: ±1V, or ±5V by soldering selection
- Over Voltage Protection:
  - ✓ Bipolar ± 2V, or ±10V regarding the input range
- Max. Sampling Rate: 20 MHz samples/sec

Note: For single channel enabled, the maximum sampling rate is 20 MHz. For two channels enabled, the 20 MHz sampling rate can be reached only when the number of samples accessed for each channel is smaller than 16K. For four channels enabled, the 20 MHz sampling rate can be reached only when the number of samples accessed for each channel is smaller than 8K. Please refer to section 5.5 for more detail information about the sampling rate and data length limitation.

- Accuracy: Gain Error ±1.5% at 25°C
- Input Impedance of Analog Input: (soldering selectable)
  - ✓ 50Ω (±1V and ±5V)
  - ✓ 1.25KΩ (±5V only)
  - ✓ 15MΩ (±1V only)

- Dynamic Characteristic:
  - ✓ Differential Linearity Error:
    - – ±0.4 LSB (Typ.) ±1.0 LSB (Max.) at 25°C
  - ✓ Integral Linearity Error:
    - – ±1.9 LSB at 25°C
- A/D Clock Sources:
  - ✓ Internal clock, Continuous external digital clock and Continuous external sine wave.
- Input Impedance of External Clock Source: 50Ω
- Trigger Sources:
  - ✓ Software, Analog threshold comparator using internal D/A to set trigger level, and External digital trigger
- Trigger Modes:
  - ✓ Software-trigger, Pre-trigger, Post-trigger, Middle-trigger, and Delay-trigger
- AD Data Transfer Method: DMA (Bus mastering)

Digital Input

- Numbers of channel:
  - ✓ 3 TTL compatible inputs with 10K ohms pull down resistor
- Input Voltage:
  - ✓ Low: Min. 0V; Max. 0.8V
  - ✓ High: Min. +2.0V Max. 5.5V
  - ✓ Input Load:
  - ✓ Low: ±1μA @0V
  - ✓ 0.5mA@ 5V
  - ✓ High: +2.7V min.@20mA max.

General Specifications

- Connectors: 5 BNC-type, one 10-pin header
- Operating Temperature: 0° C ~ 40° C
- Storage Temperature: -20° C ~ 80° C
- Humidity: 5 ~ 85%, non-condensing
- Power Consumption: +5 V @ 2.5 A (maximum)
- Dimension: 101mm(H) X 173mm(L)

1.4 Software Support

ADLINK provides versatile software drivers and packages for users' different approach to building a system. ADLINK not only provides programming libraries such as DLL for most Windows based systems, but also provide drivers for many software package such as LabVIEW®, HP VEETM, DASYLabTM, InTouchTM, InControlTM, ISaGRAFTM, and so on.

All software options are included in the ADLINK CD. Non-free software drivers are protected with licensing codes. Without the software code, you can install and run the demo version for two hours for trial/demonstration purposes. Please contact ADLINK dealers to purchase the formal license.

1.4.1 Programming Library

For customers who are writing their own programs, we provide function libraries for many different operating systems, including:

DOS Library: Borland C/C++ and Microsoft C++, the functions descriptions are included in this user's guide.

Windows 95 DLL: For VB, VC++, Delphi, BC5, the functions descriptions are included in this user's guide.

PCIS-DASK: Include device drivers and DLL for **Windows 98**, **Windows NT** and **Windows 2000**. DLL is binary compatible across Windows 98, Windows NT and Windows 2000. That means all applications developed with PCIS-DASK are compatible across Windows 98, Windows NT and Windows 2000. The developing environment can be VB, VC++, Delphi, BC5, or any Windows programming language that allows calls to a DLL. The user's guide and function reference manual of PCIS-DASK are in the CD. (\\Manual_PDF\Software\PCIS-DASK)

PCIS-DASK/X: Include device drivers and shared library for Linux. The developing environment can be Gnu C/C++ or any programming language that allows linking to a shared library. The user's guide and function reference manual of PCIS-DASK/X are in the CD. (\Manual_PDF\Software\PCIS-DASK-X.)

The above software drivers are shipped with the board. Please refer to the "***Software Installation Guide***" to install these drivers.

1.4.2 PCIS-LVIEW: LabVIEW® Driver

PCIS-LVIEW contains the VIs, which are used to interface with NI's PCIS-LVIEW contains the VIs, which is used to interface with NI's LabVIEW® software package. The PCIS-LVIEW supports Windows 95/98/NT/2000. The LabVIEW® drivers is shipped free with the board. You can install and use them without a license. For more information about PCIS-LVIEW, please refer to the user's guide in the CD. (\\Manual_PDF\Software\PCIS-LVIEW).

1.4.3 PCIS-VEE: HP-VEE Driver

The PCIS-VEE includes the user objects, which are used to interface with HP VEE software package. PCIS-VEE supports Windows 95/98/NT. The HP-VEE drivers are free shipped with the board. You can install and use them without license. For detail information about PCIS-VEE, please refer to the user's guide in the CD. (\\Manual_PDF\Software\PCIS-VEE)

1.4.4 DAQBench™: ActiveX Controls

We suggest the customers who are familiar with ActiveX controls and VB/VC++ programming use the DAQBench™ ActiveX Control components library for developing applications. The DAQBench™ is designed under Windows NT/98. For more detailed information about DAQBench, please refer to the user's guide in the CD. (\\Manual_PDF\Software\DAQBench\DAQBench Manual.PDF)

1.4.5 DASYLab™ PRO

DASYLab is an easy-to-use software package, which provides easy-setup instrument functions such as FFT analysis. Please contact us to get DASYLab PRO, which include DASYLab and ADLINK hardware drivers.

2

Installation

This chapter describes how to install the PCI-9812/10. Firstly, please read the contents in the package and unpacking information and follow these instructions carefully.

The PCI-9812/10 will perform an automatic configuration of the IRQ and I/O port address. There is no need to set any configuration, as you would use in an ISA form factor DAS card. For system reliability, it is necessary to manually assign some critical settings for analog input and output, because these settings will not be changed after your data acquisition system configuration is decided. The settings will allow your system to perform reliably & safely (the configuration cannot be changed using the software package only) when your system is running.

Please follow the steps to install the PCI-9812/10 products.

- Check what you have (section 2.1)
- Unpacking (section 2.2)
- Check the PCB (section 2.3)
- Install the hardware (section 2.4)

2.1 What You Have

In addition to this *User's Guide*, the package includes the following items:

- PCI-9812/10 Enhanced Multi-function Data Acquisition Card
- 5 BNC terminators
- ADLINK All-in-one CD
- Software Installation Guide

If any of these items is missing or damaged, contact the dealer from whom you purchased the product for replacement. Save the shipping materials and carton in case you want to ship or store the product in the future.

2.2 Unpacking

Your PCI-9812/10 card contains electro-static sensitive components that can be easily be damaged by static electricity.

Therefore, the card should be handled on a grounded anti-static mat. The operator should be wearing an anti-static wristband, grounded at the same point as the anti-static mat.

Inspect the card module carton for obvious damages. Shipping and handling may cause damage to your module. Be sure there are no shipping and handling damages on the modules carton before continuing.

After opening the card module carton, extract the system module and place it only on a grounded anti-static surface with component side up.

Again, inspect the module for damages. Press down on all the socketed IC's to make sure that they are properly seated. Do this only with the module place on a firm flat surface.

Note: DO NOT ATTEMPT TO INSTALL A DAMAGED BOARD IN THE COMPUTER.

You are now ready to install your card.

2.3 PCI-9812/10's Layout

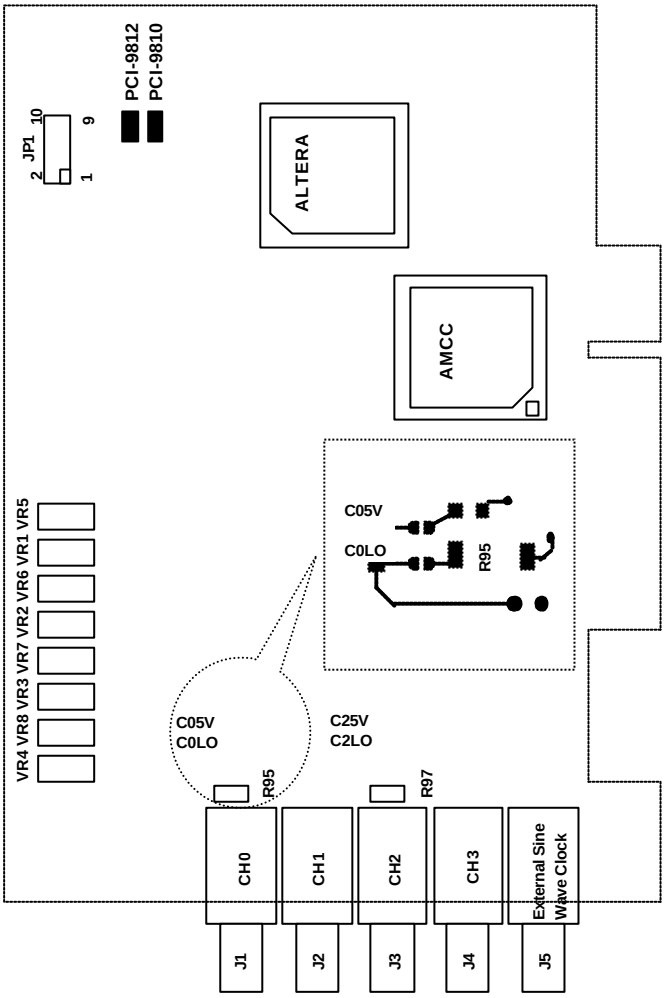


Figure 1. PCB Layout of the PCI-9812/10

2.4 Hardware Installation Outline

PCI configuration

The PCI cards (or CompactPCI cards) are equipped with plug and play PCI controller, it can request base addresses and interrupt according to PCI standard. The system BIOS will install the system resource based on the PCI cards' configuration registers and system parameters (which are set by system BIOS). Interrupt assignment and memory usage (I/O port locations) of the PCI cards can be assigned by system BIOS only. These system resource assignments are done on a board-by-board basis. It is not suggested to assign the system resource by any other methods.

PCI slot selection

The PCI card can be inserted to any PCI slot without any configuration for system resource. Please note that the PCI system board and slot must provide bus-mastering capability to operate this board well.

Installation Procedures

1. Turn off your computer
2. Turn off all accessories (printer, modem, monitor, etc.) connected to your computer.
3. Remove the cover from your computer.
4. Setup jumpers on the PCI or CompactPCI card.
5. Select a 32-bit PCI slot. PCI slot are short than ISA or EISA slots, and are usually white or ivory.
6. Before handling the PCI cards, discharge any static buildup on your body by touching the metal case of the computer. Hold the edge and do not touch the components.
7. Position the board into the PCI slot you selected.
8. Secure the card in place at the rear panel of the system.

2.5 Device Installation for Windows Systems

Once Windows 95/98/2000 has started, the Plug and Play function of Windows system will find the new NuDAQ/NuIPC cards. If this is the first time to install NuDAQ/NuIPC cards in your Windows system, you will be informed to input the device information source. Please refer to the "*Software Installation Guide*" for the steps of installing the device.

3

Signal Connection

This chapter describes the connector of the PCI-9812/10, the signal connection between the PCI-9812/10 and external devices, and the switch setting for different applications.

3.1 Connectors

The PCI-9812/10 connects to external devices through five BNC connectors and one 10-pin dual-in-line header. Fig. 2 shows the location of these connectors.

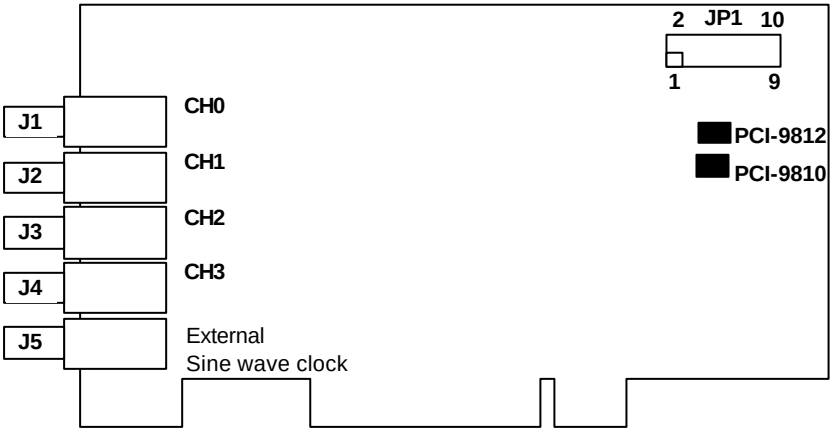


Figure 2. Location of connectors

J1: The J1 BNC connector is used for the input signal of channel 0 A/D converter.

J2: The J2 BNC connector is used for the input signal of channel 1 A/D converter.

J3: The J3 BNC connector is used for the input signal of channel 2 A/D converter.

J4: The J4 BNC connector is used for the input signal of channel 3 A/D converter.

J5: The J5 BNC connector is used for the input signal of external clock 0.

JP1: The 10-pin connector is used for digital input signal, including 1 digital clock, 1 digital trigger and 3 digital inputs.

The pin-out of JP1 is listed below:

Pin	Signal	Pin	Signal
1	External Clock Input 1	6	Ground
2	Ground	7	Digital Input 1
3	External Digital Trigger Input	8	Ground
4	Ground	9	Digital Input 2
5	Digital Input 0	10	Ground

Note: If JP1 is connected to a 9-pin D-type connector through a ribbon cable, the pin-out of the D-type connector is changed to:

Pin	Signal	Pin	Signal
1	External Clock Input 1	6	Ground
2	Ground	7	Digital Input 1
3	External Digital Trigger Input	8	Ground
4	Ground	9	Digital Input 2
5	Digital Input 0	10	N/A

3.2 Analog Input Impedance Setting

This section describes the characteristics of the different inputs of the PCI-9812/10.

3.2.1 Analog Input

PCI-9812/10 has four analog input channels which are connected through connectors J1 ~ J4. The input impedance and input amplitude range can be changed through soldering the gap switches on board (refer to PCI-9812/10's layout). A solder gap switch consists of two copper pads, the switch can be turned on by soldering these two pads together. All the four channels have the same way to configure their input characteristics, and only channel 0 is discussed here. There are 2 solder gap switches, named C0LO (channel 0 low impedance) and C05V (channel 0 5V input), to setup the input characteristics of channel 0. (Please refer to fig. 2.1 in section 2.3)

C0LO	C05V	Input Impedance	Input Range
Open	Open	High (~15M Ohm)	±1V
Open	Close	1.25K Ohm	± 5V
Close	Open	Low (50 Ohm)	± 1V (default)
Close	Close	Low (50 Ohm)	± 5V

CAUTION:

1) When the input channel is configured as a high impedance input, **DO NOT** leave the input connector unconnected. The input connector must be connected to a low impedance signal source to provide a return path for the input bias current. That is because that the maximum input bias current of the OPAMP in the input stage could be up to 35uA. That means if the input is left unconnected, then the OPAMP will be in an abnormal working environment that leads to saturation in the output stage. Although we have used a current-limiting resistor to protect the ADC, the large current brought by the saturation will damage the ADC.

2) If users want to use the high impedance 15Mohms for the applications, and the output impedance of the signal sources are really high, that will lead to the offset problem. Since the input bias current could be up to 35uA, then the voltage drop introduced by the output impedance with the bias current would be in the range of several volts. Even by adjusting the variable resistor, you could not eliminate that large offset voltage.

Note: 75-ohm input impedance can be achieved by:

Replace R95 by a 75-ohm resistor and close C0LO.

Place a T-connector with a 75-ohm terminator on J1 and open C0LO.

The corresponding switches and resistors of other channels are shown below:

Channel	Switches		Resistor
Channel 0	C0LO	C05V	R95
Channel 1	C1LO	C15V	R96
Channel 2	C2LO	C25V	R97
Channel 3	C3LO	C35V	R98

3.2.2 External Clock 0

The external clock 0 (J5) is a sine wave signal, which is converted to a TTL signal inside the PCI-9812/10. 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 External Clock's frequency is the system clock. The maximum A/D clock frequency is half of the system clock. When using the external sine clock for PCI-9812, users should note that the frequency of the sine clock must be **above 300 KHz**, otherwise the sine clock will be converted into a digital clock with a long rise time. When the rise time of a clock signal is too long, the CPLD may work unpredictably, meanwhile the real sample clock fed into the ADC will not be continuous, that is why users will see the strange sampled waveform when using slower sine wave clock. If slower sampling rate is necessary when using the PCI-9812/10, users could feed a sine wave clock that has the frequency higher than 300 KHz, and use the clock divisor to obtain a slower sampling rate.

3.2.3 External Clock 1

The external clock 1 (JP1 pin 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.

3.2.4 Digital Input

PCI-9812/10 has four digital inputs: one external digital trigger (JP1 pin3) and three general-purpose digital inputs (JP1 pin5, 7, and 9). These inputs are TTL compatible with 10K-ohm pull-down resistors.

4

Registers

The detailed descriptions of the register format and structure of the PCI-9812/10 are specified in this chapter. This information is useful for the programmer who wants to handle the card using low-level programming.

4.1 I/O Port Address

The PCI-9812/10 functions as 32-bit PCI target device to any master on the PCI bus. It supports burst transfer to memory space by using 32-bit data. So, both data read and write will be based on 32-bit data transfer. Table 1 shows the I/O address of each register with respect to the base address. The function of each register is also shown.

I/O Address	Read	Write
Base + 0	----------	ADC Channel Enable Reg.
Base + 4	----------	ADC Clock Divisor Reg.
Base + 8	----------	Trigger Mode Reg.
Base + C	----------	Trigger Level Reg.
Base + 10	----------	Trigger Source Reg.
Base + 14	----------	Post Trigger Counter Reg.
Base + 18	FIFO Control & Status Reg	FIFO Control & Status Reg.
Base + 1C	----------	Acquisition Enable Reg.
Base + 20	----------	Clock Source Register

Table 1. I/O Address

4.2 ADC Channel Enable Register

The PCI-9812/10 has 4 analog input channels, named CH0, CH1, CH2, and CH3. CH0 ~ CH3 can be enabled or disabled by bit 0 ~ 3 of the ADC channel enable register.

Address: BASE + 0

Attribute: write only

Data Format:

Bit	7	6	5	4	3	2	1	0
BASE+0	---	---	---	---	CH3EN	CH2EN	CH1EN	CH0EN
BASE+1	---	---	---	---	---	---	---	---
BASE+2	---	---	---	---	---	---	---	---
BASE+3	---	---	---	---	---	---	---	---

Bit 31->4 -- don't care

Bit 3 -- CH3EN

Bit 2 -- CH2EN

Bit 1 -- CH1EN

Bit 0 -- CH0EN

Legal combinations (refer to section 5.5) of these four bits are:

0000 -- no channel is enabled

0001 -- only CH0 is enabled

0011 -- CH0 and CH1 are enabled

1111 -- all channels are enabled

4.3 ADC Clock Divisor Register

Feeding the ADC source clock to a clock frequency divider generates the ADC sampling clock. The output of the frequency divider becomes the sampling clock. The frequency of the ADC sampling clock is:

Frequency of source clock / ADC clock divisor

Address: BASE + 04h

Attribute: write only

Data Format:

Bit	7	6	5	4	3	2	1	0
Base + 4	DIV7	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
Base + 5	DIV15	DIV14	DIV13	DIV12	DIV11	DIV10	DIV9	DIV8
Base + 6	---	---	---	---	---	---	---	---
Base + 7	---	---	---	---	---	---	---	---

DIV15..0: The AD clock frequency devisor

---: Don't care

Note: The minimum value of this register is 2, and the DIV0 is hardwired to 0.

4.4 Trigger Mode Register

The PCI-9812/10 has five trigger modes: software trigger, post trigger, pre-trigger, middle trigger and delay trigger. The trigger mode register is used to specify which trigger mode is currently used.

Address: BASE + 08h

Attribute: write only

Data Format:

Bit	7	6	5	4	3	2	1	0
Base + 8	---	---	---	---	---	TRGMOD2	TRGMOD1	TRGMOD0
Base + 9	---	---	---	---	---	---	---	---
Base + A	---	---	---	---	---	---	---	---
Base + B	---	---	---	---	---	---	---	---

TRGMOD2..0: Trigger mode

--- : don't care

The 5 trigger modes are list as the following table:

TRGMOD2	TRGMOD1	TRGMOD0	Trigger Mode
0	0	0	Software trigger
0	0	1	Post trigger
0	1	0	Pre-trigger
0	1	1	Delay trigger
1	0	0	Middle trigger

Note: All the other values of this register are illegal, and the PCI-9812/10 will not acquire data if illegal value is set.

4.5 Trigger Level Register

The trigger condition of the PCI-9812/10 includes trigger level and trigger slope. This register sets the trigger level, and the trigger source register described in the next paragraph sets the trigger slope.

Address: BASE + 0ch

Attribute: write only

Data Format:

Bit	7	6	5	4	3	2	1	0
BASE+Ch	TRGLVL7	TRGLVL6	TRGLVL5	TRGLVL4	TRGLVL3	TRGLVL2	TRGLVL1	TRGLVL0
BASE+Dh	---	---	---	---	---	---	---	---
BASE+Eh	---	---	---	---	---	---	---	---
BASE+Fh	---	---	---	---	---	---	---	---

TRGLVL7..0: trigger level

---: Don't care.

The relationship between the 8-bit trigger level and the trigger voltage is:

TRGLVL7..0(bit 7..0)	trigger voltage(±1V)	trigger voltage(±5V)
0xFF	0.992V	4.96V
0xFE	0.984V	4.92V
|	|	|
0x81	0.008V	0.04V
0x80	0.000V	0.00V
0x7F	-0.008V	-0.04V
|	|	|
0x01	-0.992V	-4.96V
0x00	-1.000V	-5.00V

4.6 Trigger Source Register

PCI-9812/10 supports five trigger sources. They are CH0, CH1, CH2, CH3 and external digital trigger.

Address: BASE + 10h

Attribute: write only

Data Format:

Bit	7	6	5	4	3	2	1	0
Base + 10	--	--	--	--	TRGSLP	TRGSRC2	TRGSRC1	TRGSRC0
Base + 11	--	--	--	--	---	---	---	---
Base + 12	--	--	--	--	---	---	---	---
Base + 13	--	--	--	--	---	---	---	---

TRGSLP: trigger slope.

0: positive slope trigger

1: negative slope trigger

TRGSRC2 -> TRGSRC0: trigger source.

TRIGSRC2	TRIGSRC2	TRIGSRC2	trigger source
0	0	0	CH0
0	0	1	CH1
0	1	0	CH2
0	1	1	CH3
1	X	X	EX_DIG_trigger

When the external digital trigger is selected, the positive slope trigger equals to rising edge trigger, and the negative slope trigger equals to falling edge trigger, and the value of trigger level register is meaningless.

4.7 Post Trigger Counter Register

The post trigger counter is a 16-bit down counter. The counter is pre-loaded with the value in post trigger counter register and it will count down on the rising edge of ADC sampling clock after the trigger condition is met. When the count reaches 0, the counter stops. The counter is used to control the delay time in delay trigger mode and to control the post trigger sampling count in middle trigger mode.

Address: BASE + 14h

Attribute: write only

Data Format:

Bit	7	6	5	4	3	2	1	0
BASE+Ch	PSTCN 7	PSTCN 6	PSTCN 5	PSTCN 4	PSTCN 3	PSTCN 2	PSTCN 1	PSTCN0
BASE+Dh	PSTCN15	PSTCN14	PSTCN13	PSTCN12	PSTCN11	PSTCN10	PSTCN9	PSTCN8
BASE+Eh	---	---	---	---	---	---	---	---
BASE+Fh	---	---	---	---	---	---	---	---

PSTCNT15..0: This value is pre-loaded to the post trigger counter when the post trigger counter register is written.

---: Don't card

4.8 FIFO Status Register

This register is used to monitor some status of the PCI-9812/10.

Address: BASE + 18h

Attribute: read

Data Format:

Bit	7	6	5	4	3	2	1	0
BASE+18h	---	---	ACQ	TD	PTCO	FIFOOR	FIFOHF	FIFOIR
BASE+19h	---	---	---	---	---	---	---	---
BASE+1Ah	---	---	---	---	---	---	---	---
BASE+1Bh	---	---	---	---	---	---	---	---

Bit 0 -- FIFOIR, FIFO input ready flag**.**

0: The FIFO is not ready for input, which means the FIFO is full

1: The FIFO is ready for input (not full).

Bit 1 -- FIFOHF, FIFO half full flag.

0: The FIFO is not half full yet.

1: The FIFO is at least half full.

Bit 2 -- FIFOOR, FIFO output ready flag

0: The FIFO is not ready for output, which means the FIFO is empty.

1: The FIFO is ready for output (not empty).

Bit 3 -- PTC0, post trigger counter is 0

0: The post trigger counter is not 0.

1: The post trigger counter reaches 0.

Bit 4 -- TD, trigger detection flag

0: The trigger condition is not met yet no trigger is detected.

1: Trigger is detected.

Bit 5 -- ACQ, acquisition flag

0: The PCI-9812/10 is not acquiring data. Maybe the card is disabled or the card is waiting for trigger.

1: The PCI-9812/10 is acquiring data.

Bit 6..31 -- don't card

4.9 FIFO Control Register

This register is used to control the on-board FIFO memory.

Address: BASE + 18h

Attribute: write

Data Format:

Bit	7	6	5	4	3	2	1	0
BASE+18h	---	---	---	---	---	---	CLRTRG	CLRFIFO
BASE+19h	---	---	---	---	---	---	---	---
BASE+1Ah	---	---	---	---	---	---	---	---
BASE+1Bh	---	---	---	---	---	---	---	---

Bit 0 -- CLRFIFO, clear the on-board FIFO

When a "1" is written to this bit, the entire on-board FIFO is cleared.

Bit 1 -- CLRTRG, clear trigger detection flag

When a "1" is written to this bit, the trigger detection bit is cleared.

Bit 2..31 -- don't care

4.10 Acquisition Enable Register

The register enables or disables the ADC acquisition.

Address: BASE + 1ch

Attribute: write only

Data Format:

: Bit	7	6	5	4	3	2	1	0
BASE+18h	---	---	---	---	---	---	---	ACQEN
BASE+19h	---	---	---	---	---	---	---	---
BASE+1Ah	---	---	---	---	---	---	---	---
BASE+1Bh	---	---	---	---	---	---	---	---

Bit 31..1 -- don't care

Bit 0 -- ACQEN, acquisition enable

When a "1" is written to this bit, the PCI-9812/10 is ready to sample data. When a "0" is written, the PCI-9812/10 is disabled.

4.11 Clock Source Register

The register is used to select the system clock source.

Address: BASE + 20h

Attribute: write only

Data Format:

Bit	7	6	5	4	3	2	1	0
BASE+18h	---	---	---	---	---	CLKSRC1	CLKSRC0	Freq_Sel
BASE+19h	---	---	---	---	---	---	---	---
BASE+1Ah	---	---	---	---	---	---	---	---
BASE+1Bh	---	---	---	---	---	---	---	---

Bit 31..3 -- don't care

Bit 2..1 -- CLKSRC1..0, ADC clock source

Bit 0: --- Freq_Sel, Frequency selection.

Freq_Sel:

1: Frequency of A/D clock source is higher than PCI clock frequency (33 MHz)

0: Frequency of A/D clock source is lower than PCI clock frequency (33 MHz)

CLKSRC2	CLKSRC1	Selected clock source
0	0	Internal clock (40 MHz)
0	1	External sine wave clock
1	0	External digital clock
1	1	Illegal

Note: When external clock is selected, this external clock is also divided by the frequency divider as mentioned before, so the frequency of the external clock should be at least twice as the desired sampling frequency.

4.12 High Level Programming

To operate the PCI-9812/10, you can by-pass the detailed register structures and control your PCI-9812/10 card directly via the high-level Application-Programming-Interface (API). The software Libraries, including DOS Library for Borland C++ and DLL driver for Win-95, are included in the ADLINK CD. For more detailed information, please refer to Chapter 6 “C/C++ Software Library”.

4.13 Low Level Programming

To operate the PCI-9812/10, users do not need to understand how to write a hardware dependent low-level program. Because it is more complex to control the PCI controller and the information is not described in this manual. We do not recommend users to program its applications based on low-level programming. The PCI controller used in the PCI-9812 is AMCC-S5933. For more s5933 PCI controller information, please visit the web site: www.amcc.com.

5

Operation Theory

The operation theorem of the functions on PCI-9812/10 card is described in this chapter. The functions include A/D conversion and digital input. The operation theorem will help you to understand, operate and program the PCI-9812/10.

5.1 A/D Conversion Procedure

Before programming the PCI-9812/10 to perform the A/D conversions, you have to understand the following concepts:

- A/D conversion procedure
- A/D signal source control
- A/D trigger source control
- A/D clock control
- A/D data transfer mode
- A/D data format

When using an A/D converter, users should know about the properties of the signal to be measured at first. Users can decide which channels to use and connect the signals to the PCI-9812/10. In addition, users should define and control the A/D signal sources, including the A/D channels, A/D gains, and A/D signal types. Please refer to section 5.2 for A/D signal source control.

After deciding the A/D signal source, users must decide how to trigger the A/D conversion and define/control the trigger source. The A/D converter will start to convert the signal to a digital value when a trigger condition is met. The PCI-9812/10 provides five trigger modes please refer to section 5.3

The A/D clock is controlled by internal clock or external clock source. The detailed operation of the A/D clock source is described in section 5.4.

At the end of an A/D conversion, the A/D data is buffered in a FIFO. The total FIFO size on PCI-9812/10 is 32K samples . This buffer size is relative to the highest data transfer rate. The A/D data should be transferred into PC's memory for further processing. The PCI-9812/10 uses DMA to transfer the A/D data to host memory. Please refer to section 5.5.

To process A/D data, programmer should know about the A/D data format. Please refer to section 5.6.

5.2 A/D Signal Source Control

To control the A/D signal source, three concepts should be understood; they are signal type, signal channel and signal range.

Signal Type

The A/D signal sources of PCI-9812/10 are single ended (SE).

Numbers of Channel

There are four channels for SE mode. The ADC Channel Enable Register controls the channel number. Please refer to section 4.2.

Signal Range and Input impedance

The proper signal range is important for data acquisition. The available signal input ranges for 9812/10 are ±5V or ±1V, which are set by soldering the copper pads on the PCB. The input impedance for high-speed applications should also be considered. The selectable input impedance values are 50 Ohm, 1.25 K, 15M ohm. Please refer to section 3.2 for details.

5.3 A/D Trigger Source Control

Performing the trigger acquisition in PCI-9812/10, the following items have to be specified before DMA operation starts:

Clock source: refer to section 5.4

Clock rate: refer to section 5.4

Trigger source: refer to section 5.3.1.

Trigger level: The trigger event occurs when the trigger signal crosses the specified trigger voltage. Please refer to section 4.5 for the relationship between the 8-bit trigger level and the trigger voltage.

The trigger is detected while the trigger event occurs. For Post-trigger and Middle-trigger, the data acquisition is performed after the trigger event; however, the time when the AD conversion starts is 350 ns slower than the time when the trigger is detected. This 350ns delay will have minor effect for high-speed data acquisition.

Trigger polarity: trigger slope '0' value means positive trigger and '1' value means negative trigger.

Trigger mode: refer to section 5.3.2.

5.3.1 Trigger Sources

Internal Trigger

An internal trigger is a software trigger. The trigger event occurs when you call _9812_AD_DMA_Start( ) function to start the operation.

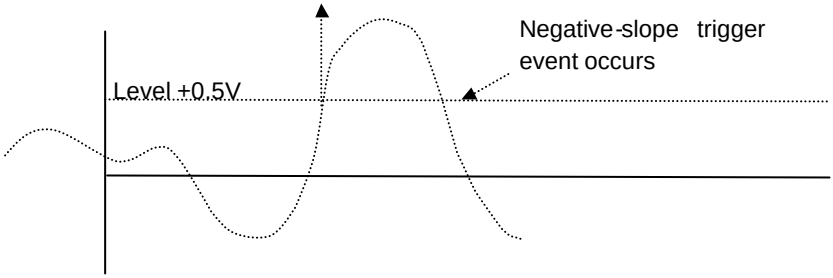
External Analog Trigger

You can use the signal on any analog input channel (CH0, CH1, CH2, or CH3) as the trigger signal for external analog trigger. The trigger conditions for analog triggers are described as follows:

Positive-slope trigger - The trigger event occurs the first time the trigger signal (analog input signal) changes from a voltage that is lower than the specified trigger level to a voltage that is higher than the specified trigger level.

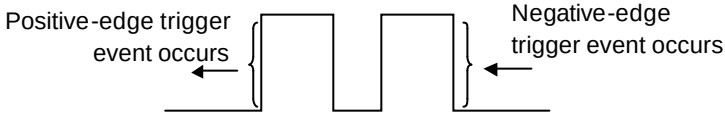
Negative-slope trigger - the trigger event occurs the first time the trigger signal (analog input signal) changes from a voltage that is higher then the specified trigger level to a voltage that is lower than the specified trigger level.

Positive-slope trigger event occurs



External Digital Trigger

An external digital trigger occurs when a rising edge or a falling edge is detected on the digital signal connected to pin3 of JP1 for external digital trigger.



5.3.2 Simultaneous Trigger for Multiple Cards

When multiple PCI-9812 cards are used in one system, the trigger sources of every card can be connected together to provide the function of simultaneous trigger for multiple cards. Please note that simultaneous trigger is not equivalent to simultaneous A/D conversion. The theoretical time difference between the samples on different card will be ½ of the clock period. Refer to section 5.4 for more information about A/D conversion clock control.

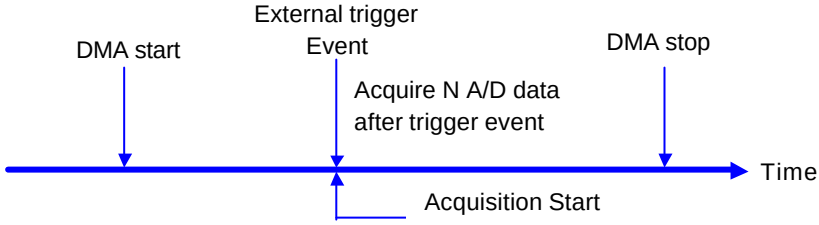
5.3.3 Trigger Modes

Software-Trigger Acquisition

This trigger mode does not need any external trigger source. The trigger event occurs when you call _9812_AD_DMA_Start( ) function to start the operation.

Post-Trigger Acquisition

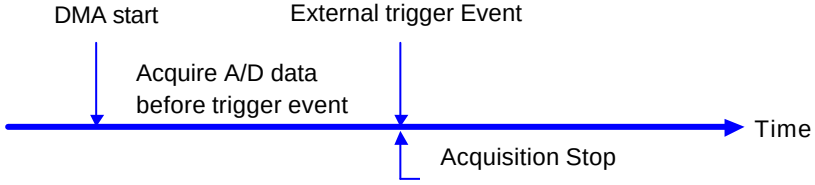
Use post-trigger acquisition in applications when you want to collect data after a specified trigger event. The trigger can either be an external analog trigger or digital trigger.



Pre-Trigger Acquisition

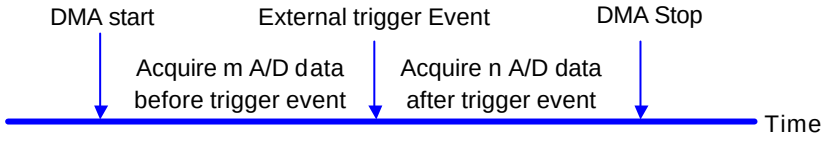
Use pre-trigger acquisition in applications where you want to collect data before a specified trigger event. The trigger can either be an external analog trigger or digital trigger.

The data acquisition starts when DMA operation starts. The operation stops when the external trigger event occurs. If the external trigger occurs before the specified count of data (specified by _9812_AD_DMA_Start( ) function, refer to section 6.2.3) is read, the number of retrieved data will be fewer than the specified count. However if the external trigger occurs after the specified count of data is read, the program only samples the specified count of data.



Middle-Trigger Acquisition

Use middle-trigger acquisitions in application where you want to collect data before and after a specified trigger event. Acquiring data before a trigger event occurs for Middle trigger might not get the specified count of data, just like the pre-trigger mode.

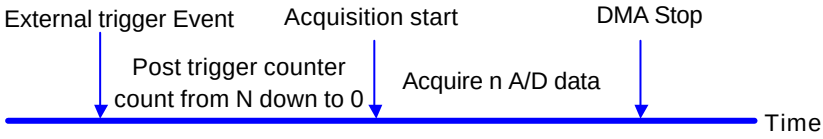


The desired number of samples after trigger event is pre-loaded in post trigger counter register and will count down on the rising edge of ADC sampling clock after the trigger condition is met. When the count reaches 0, the counter stops. The trigger can either be an external analog trigger or digital trigger.

Delay Trigger Acquisition

Use delay trigger acquisition in application that you want to delay the data collection after the occurrence of a specified trigger event. The delay time is controlled by the value, which is pre-loaded in the post trigger counter register. Then the counter counts down on the rising edge of ADC sampling clock after the trigger condition was met.

When the count reaches 0, the counter stops and PCI-9812/10 starts to acquire data.



5.4 A/D Clock Source Control

The AD clock source determines how the board regulates the timing of conversions when you are acquiring multiple samples from a single channel or from a group of multiple channels. The A/D clock sources on the PCI-9812 must use pacer clock but not single shot, because the A/D converters are in a pipelined structure, which needs 8 conversion clock to complete the conversion of digital data.

5.4.1 A/D Clock Sources

The A/D converters operate under the paced mode, which uses pacer clock for A/D conversion at a fixed rate. PCI-9812/10 supports three clock sources for analog input conversion:

- Internal A/D pacer clock (default);
- External sine wave clock;
- External square clock.

The description of these three clock sources is in the following.

5.4.2 Internal Pacer Clock

An on-board timer / counter is used as the internal A/D pacer clock. The frequency of the pacer is software controllable. The maximum pacer signal rate is 40Mz/2=20MHz, that is also the maximum sampling rate of PCI-9812/10. Note that 40MHz is the on-board clock. Feeding the clock source into a frequency divider generates the ADC sampling frequency. The following formula determines the ADC sampling frequency:

Sampling Rate = Frequency of Source Clock / ADC Clock Divisor

Note that the ADC Clock Divisor = 2,4,6,8,10… 65534 (maximum)

5.4.3 External Pacer Clock

Users could connect an external pacer clock (sine or square wave) to the EXTCLK1 (pin 1) on JP1. Because users can handle the external signal with outside devices, the conversion rate of this mode is more flexible than the previous mode. When external clock is selected, the frequency divider as mentioned also divides this external clock. Therefore the frequency of the external clock should be at least twice as the desired sampling frequency. The formula is shown as following:

Sampling Rate = Frequency of Source Clock / ADC Clock Divisor

Note: 1.The clock divider must be an even number, that is, the ADC Clock Divisor = 2, 4, 6, 8, 10… 65534 (maximum), The minimum divider value is 2.Please refer to section 6.2.6 and section 6.2.7 to set the clock source and frequency divider.

2.Because of the pipelined architecture of the ADC, the first AD sample takes several clocks to convert; therefore, the external clock must be continuous for correct AD operation.

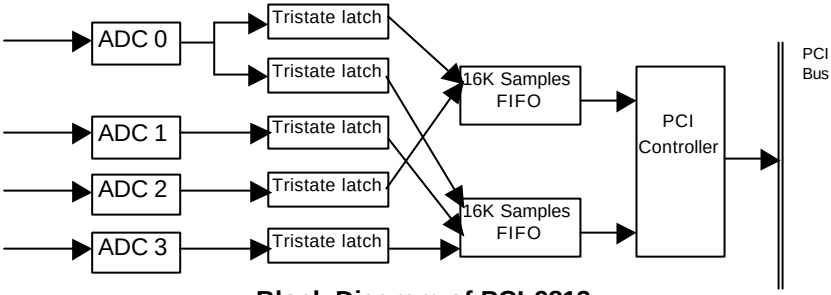
5.4.4 Multiple Cards Operation

When multiple cards are used in one system, 4-channels on one card can achieve simultaneous conversion because of the same internal clock source. However, the channels between two cards cannot be synchronized because the clock sources on different cards come from different sources. Even when the same external clock source is applied to all cards, the A/D conversion time is still possibly asynchronous because an on-board clock divider (divided by 2) is used. Therefore, when the same external clock source is applied to multiple cards, the time difference of the sampling clocks between two cards will be half of the sampling clock period. The A/D clock cannot synchronize the multiple cards.

5.5 A/D Data Transfer

5.5.1 AD Data Transfer

For acquiring the AD data, there are several function blocks on the PCI-9812 board. Even when the maximum sampling rate is specified up to 20MHz, however, there are some limitations due to the high total data throughput. Users should refer to the following block diagram to understand how the analog signal is converted to digital form and transferred to PC host memory. The data transfer rate limitation and the bottleneck will also be introduced in this section.



Block Diagram of PCI-9812

5.5.2 Simultaneous Sampling of 4 AD Channels

The PCI-9812/10 is equipped with 4 AD converters supporting maximum 20MHz simultaneous sampling rate. The high speed and simple use allow the PCI-9812/9810 to serve many applications, such as image digitizing, medical applications, vibration testing equipment and RF or baseband signal digitization.

For one channel application, you can only select channel 0, and the total FIFO length is 32K samples. For simultaneous two-channel application, you have to use channel 0 and channel 1 to optimize the FIFO usage. The on-board circuit does not allow you to use channel 0 and 2 or channel 2 and 3 at the same time. Please note that for simultaneous sampling applications, the hardware only support 2 or 4 channels but no 3-channel data acquisition.

5.5.3 Total Data Throughput

When 4 channels start at the same time, the total data throughput from the AD converter to the on-board FIFO memory will be

Sampling Rate x number of channels x 2 bytes/channel.

Therefore, the maximum total data throughput is 160M bytes /sec.

160MB/s = 20MHz ´ 4 channels ´ 2 bytes/channel.

This extremely high data rate is beyond the 32bit/33MHz PCI-bus bandwidth. Therefore, two 16K words (samples) FIFO are designed for buffering the data.

In are a total, 32K word (32K samples) FIFO is on-board. When 4 channels are used, the FIFO size is 8K samples per channel. When only two channels (#0 and #1) are used, the FIFO size is 16K samples per channel. When only channel 0 is used, the FIFO size is 32K samples.

Users will have to calculate the total data throughput for their applications, because this value will have relation to the total data length you can continuously acquire.

5.5.4 Maximum Acquiring Data Length

The burst PCI bandwidth is 132MB /sec. However, the effective sustained data rate is usually less than 100MB/s. This value may be lower when many PCI add-on devices are used at the same time. If the total AD data throughput is lower than the PCI bandwidth, then the AD data can be put into the host memory through Bus-mastering DMA, and the total acquiring data length could be up-to 64M bytes (32M samples) which is the limitation of the PCI controller. If the total AD data throughput is higher than the PCI bandwidth, then the maximum data length will be 16K or 8K samples, and that is the limitation from the size of on-board FIFO.

5.5.5 Bus-mastering Data Transfer

PCI bus-mastering DMA is necessary for high speed DAQ in order to utilize the maximum PCI bandwidth. The bus-mastering controller, which is built-in into the AMCC-5933 PCI controller ASIC, controls the PCI bus when it becomes the master of the bus. Bus mastering reduces the size of on-board memory and the CPU loading because data is directly transferred to the computer's memory without host CPU intervention.

Bus-mastering DMA provides the fastest data transfer rate on PCI-bus. Once the analog input operation starts, control returns to your program. The hardware temporarily stores the acquired data in the onboard A/D FIFO and then transfers the data to a user-defined DMA buffer memory in the computer. Please note that even when the acquiring data length is less than the FIFO, the AD data will not be kept in the FIFO, but will be directly transferred to host memory by bus-mastering DMA.

The DMA transfer mode is very complex to program. We recommend using a high-level program library to manipulate this card. If you want to program the software, which can handle the DMA bus master data transfer, please refer to more information about the PCI controller (www.amcc.com).

5.5.6 Host Memory Operation

Because of DMA data transfer, the AD data cannot be simultaneously processed when the data is transferred. Therefore, user will have to process the AD data after the completion of one DMA cycle. Please note that if the total data throughput in your application is relatively high (>20MB/s), you with have to consider that the processing of the AD data needs time and also consumes the CPU computation power. There is limitation. For example, you cannot continuously acquire data at the rate of 20MB/s, but CPU can only process the data at the rate of 10MB/s. In this case, the FIFO will be full finally and the data acquisition will be discontinuous.

Storing the data in host memory into hard disk or other storage device should be also considered. The burst data rate of current HDD technology could be 90 up to 80MB/s. However, practically the HDD effective bandwidth is usually less than 10MB/s, especially when the HDD seek time is longer, for example, 20 ms, the FIFO is already full and the acquired data could not be continuous.

Another limitation comes from the OS and the host memory size. Under DOS environment, the maximum memory size you can allocate is less than 640K except that you use extended memory managing software. Under Windows-95 or NT, it is quite hard to get a continuously large size of memory such as 64M bytes, you have to keep the memory “clean” , then there may be a chance to allocate large size of memory. The PCI bus-mastering DMA controller of PCI-9812 needs continuous memory to store the AD data.

5.5.7 Summary

So far we have discussed the AD data transferring, the maximum sampling rate and the maximum continuous data length at the sample time. Here we summarize to get the following steps:

1. Calculate the total data throughput;
2. Check if the total data throughput is higher than the effective PCI bandwidth in your system;
3. if it is higher, then the maximum data length will be limited to 16K samples for 2 channels and 8K samples for 4 channels;
4. If it is lower, then the maximum total data size is 64M bytes, which comes from the limitation of host memory;
5. For continuously acquiring the AD data, that is, the data length is infinite; you can use the "double-buffered" software. However, you still need to calculate the CPU computation bandwidth to check if the continuity is possible.

5.6 AD Data Format

The A/D data of 12-bit PCI-9812 is on the 12 MSB of the 16-bit A/D data. The 4 LSB of the 16-bit A/D data must be truncated by software (please refer to section 6.2.3). The relationship between the real signal voltage and the sampled value is shown in the following table:

A/D Data (Hex)	Decimal Value	V (Volts, -1V~1V)	V (Volts, -5V~5V)
7FF 0	+32752	+1.0000	+5.0000
400 0	+16384	+0.5002	+2.5010
001 0	+16	+0.0005	+0.0025
000 0	0	0.0000	0.0000
FFF 0	-16	0.0005	-0.0025
C00 0	-16384	-0.5002	-2.5010
801 0	-32752	-1.0000	-5.0000
800 0	-32768	-1.0049	-5.0024

The A/D data of 10-bit PCI-9810 is on the 10 MSB of the A/D data. The 6 LSB of the 16-bit A/D data must be truncated by software. The relationship between the real signal voltage and the sampled value is shown in the following table:

A/D Data (Hex)	Decimal Value	V (Volts, -1V~1V)	V (Volts, -5V~5V)
7FC 0	+32704	+1.0000	+5.0000
400 0	+16384	+0.5002	+2.5010
0040	+64	+0.0005	+0.0025
000 0	0	0.0000	0.0000
FFC 0	-64	-0.0005	-0.0025
C00 0	-16384	-0.5002	-2.5010
804 0	-32704	-1.0000	-5.0000
800 0	-32768	-1.0020	-5.0098

The formula between the A/D data and the analog value is

Voltage = AD_data x (1/K) x (Gain)

Where *Gain* and *K* are constants.

For analog input range **-1V~1V,** *Gain* =1; for analog input range **-5V~5V,** *Gain* =5.

For PCI-9812, K=2047x16=32752;

For PCI-9810, K=511x64=32704.

6

C/C++ Library

This chapter describes the software library for operating this card. Only the functions in DOS library and Windows 95 DLL are described. Please refer to the PCIS-DASK function reference manual, which included in ADLINK CD, for the descriptions of the Windows 98/NT/2000 DLL functions.

The function prototypes and some useful constants are defined in the header files LIB directory (DOS) and INCLUDE directory (Windows 95). For Windows 95 DLL, the developing environment can be Visual Basic 4.0 or above, Visual C/C++ 4.0 or above, Borland C++ 5.0 or above, Borland Delphi 2.x (32-bit) or above, or any Windows programming language that allows calls to a DLL. It provides the C/C++, VB, and Delphi include files.

6.1 Libraries Installation

Please refer to the "***Software Installation Guide***" for the detail information about how to install the software libraries for DOS, or Windows 95 DLL, or PCIS-DASK for Windows 98/NT/2000.

The device drivers and DLL functions of Windows 98/NT/2000 are included in the PCIS-DASK. Please refer the PCIS-DASK user's guide and function reference, which included in the ADLINK CD, for detailed programming information.

6.2 Programming Guide

6.2.1 Naming Convention

The functions of the NuDAQ PCI cards or NuIPC CompactPCI cards' software driver are using full-names to represent the functions' real meaning. The naming convention rules are:

In DOS Environment :

_{hardware_model}_{action_name}. **e.g. _9812_Initial().**

All functions in PCI-9812 driver are with 9812 as {hardware_model}. But they can be used by PCI-9812, PCI-9810.

In order to recognize the difference between DOS library and Windows 95 library, a capital "**W**" is put on the head of each function name of the Windows 95 DLL driver. e.g. **W_9812_Initial().**

6.2.2 Data Types

We defined some data type in Pci_9812.h (DOS) and Acl_pci.h (Windows 95). These data types are used by NuDAQ Cards' library. We suggest you to use these data types in your application programs. The following table shows the data type names and their range.

Type Name	Description	Range
U8	8-bit ASCII character	0 to 255
I16	16-bit signed integer	-32768 to 32767
U16	16-bit unsigned integer	0 to 65535
I32	32-bit signed integer	-2147483648 to 2147483647
U32	32-bit single-precision floating-point	0 to 4294967295
F32	32-bit single-precision floating-point	-3.402823E38 to 3.402823E38
F64	64-bit double-precision floating-point	-1.797683134862315E308 to 1.797683134862315E309
Boolean	Boolean logic value	TRUE, FALSE

6.3 _9812_Initial

@ Description

This function is used to initialize PCI-9812/10. Every PCI-9812/10 has to be initialized by this function before calling other functions.

@ Syntax

C/C++ (DOS)

```
int _9812_Initial (int card_number, U16
            *op_base_address,U16  *pt_base_address,
            U16 *irq_no, U16 *pci_master)
```

C/C++ (Windows 95)

```
int W_9812_Initial (int card_number, U16
            *op_base_address,U16  *pt_base_address,
            U16 *irq_no, U16 *pci_master)
```

Visual Basic (Windows 95)

```
W_9812_Initial (ByVal card_number As Long,
            op_base_address As Integer,
            pt_base_address As Integer, irq_no As
            Integer, pci_master As Integer) As
            Long
```

@ Argument

card_number: the card number of PCI-9812/10 to be initialized, totally 10 cards can be initialized, the valid card numbers are 0~9.

op_base_address: the physical location of S5933 operation Registers in I/O space.

pt_base_address: the physical location of add-on registers in pass-through I/O space

irq_no : the interrupt IRQ level of your PCI-9812 card, this available IRQ value is automatically assigned by system BIOS.

pci_master : BIOS enables or disables bus mastering in PCI Command Register

@ Return Code

```
PCICardNumErr
PCIBiosNotExist
PCIBaseAddrErr
NoError
```

6.4 _9812_Close

@ Description

This function is used to close a previously initialized 9812 card.

@ Syntax

C/C++ (DOS)

```
int _9812_Close (int card_number)
```

C/C++ (Windows 95)

```
int W_9812_Close (int card_number)
```

Visual Basic (Windows 95)

```
W_9812_Close (ByVal card_number As Long) As Long
```

@ Argument

card_number: the card number of PCI-9812 to be closed, the valid card numbers are 0~9.

@ Return Code

```
PCICardNumErr
PCICardNotInit
NoError
```

6.5 _9812_AD_DMA_Start

@ Description

This function will start an operation of A/D conversion N times with DMA data transfer. It will take place in the background, which will not stop until the Nth conversion has been completed or until your program executes _9182_AD_DMA_Stop to stop the operation. After executing this function, it is necessary to check the status of the operation by using the function _9812_AD_DMA_Status.

@ Syntax

C/C++ (DOS)

```
int _9812_AD_DMA_Start (int card_number, int
            ch_cnt, U32 *buff, U32 count)
```

C/C++ (Windows 95)

```
int W_9812_AD_DMA_Start(int card_number, int
            ch_cnt, HANDLE memID, U32 count)
```

Visual Basic (Windows 95)

```
W_9812_AD_DMA_Start (ByVal card_number As Long,
            ByVal ch_cnt As Long, ByVal handle As
            Long, ByVal count As Long) As Long
```

*@ **Argument***

card_number: the card number of PCI-9812

ch_cnt: number of A/D channel enabled. The valid values are:
- 0: no channel is enabled
- 1: only channel 0 is enabled
- 2: channel 0, 1 are enabled, and the sequence of channel scan is 0, 1, 0, 1, ……..
- 4: all channels are enabled, and the sequence of channel scan is 0, 1, 2, 3, 0, 1, 2, 3, …….

buff (DOS): the start address of the memory buffer to store the A/D data. The buffer size must be larger than the number of A/D conversion. This memory should be in double word alignment. The resolution of A/D data is 12-bit (for 9812) and 10-bit (for 9810). Please refer to section 5.6 for the A/D data format. The buffer format is:

DATA 1	DATA 2	DATA 3	DATA 4	.....	DATA N-1	DATA N
16-bit	16-bit	16-bit	16-bit	16-bit	16-bit	16-bit

Each 16-bit data:
D11 D10 D9 ..... D1 D0 b3 b2 b1 b0
where D11, D10, ... , D0 : A/D converted data (9812) or
D11, D10, ... , D2 : A/D converted data (9810).
b2, b1, b0 : digital input data from channel DI2, DI1, DI0.
b3: trigger detection flag,
0: no trigger is detected
1: trigger is detected

memID (Win-95): the memory ID of the allocated system DMA memory. In Windows 95 environment, before calling **W_9812_AD_DMA_Start**, **W_9812_Alloc_DMA_Mem** must be called to allocate a contiguous DMA memory. **W_9812_Alloc_DMA_Mem** will return a memory ID for identifying the allocated DMA memory, as well as the linear address of the DMA memory for user to access the data. The format of the A/D data is the same as DOS buffer (*buff* argument).

count: the number of A/D conversion.

@ Return Code

```
PCICardNumErr,   PCICardNotInit
InvalidDMACnt,   BufNotDWordAlign
DMATransferNotAllowed, NoError
```

6.6 _9812_AD_DMA_Status

@ Description

Since the _9812_AD_DMA_Start is executed on background, you can issue the function _ 9812_AD_DMA_Status to check its operation status.

@ Syntax

C/C++ (DOS)

```
int _9812_AD_DMA_Status(int card_number, int
            *count, int *status, U32 *start_idx)
```

C/C++ (Windows 95)

```
int W_9812_AD_DMA_Status(int card_number, int
            *count, int *status, U32 *start_idx)
```

Visual Basic (Windows 95)

```
W_9812_AD_DMA_Status (ByVal card_number As Long,
            count As Long, status As Long,
            start_idx As Long) As Long
```

@ Argument

card_number: the card number of PCI-9812/10 to be selected
count: Current transferred data amount of DMA
status: status of DMA data transfer
0: DMA_done
1: DMA_continue
2: DMA_wait_trig
3: DMA_wait_delay
start_idx: The index where the data start from in user's buffer, i.e the sequence of read data is: buff[start_idx], buff[start_idx+1], _, buff[0], buff[1],_.,buff[start_idx-1].

@ Return Code

```
PCICardNumErr
PCICardNotInit
NoError
```

6.7 _9812_AD_DMA_Stop

@ Description

This function is used to stop the DMA data transfer. After executing this function, the _9812_AD_DMA_Start function stops. The function returns the number of data that has been transferred, no matter if the A/D DMA data transfer is stopped by this function or by the DMA terminal count ISR.

@ Syntax

C/C++ (DOS)

```
int _9812_AD_DMA_Stop(int card_number, U32
          *count)
```

C/C++ (Windows 95)

```
int W_9812_AD_DMA_Stop(int card_number, U32
          *count)
```

Visual Basic (Windows 95)

```
W_9812_AD_DMA_Stop (ByVal card_number As Long,
           count As Long) As Long
```

@ Argument

card_number: the card number of PCI-9812 to be selected
count: the number of A/D data that has been transferred.

@ Return Code

```
PCICardNumErr
PCICardNotInit
NoError
```

6.8 _9812_Set_Clk_Src

@ Description

This function is used to specify the ADC clock source.

@ Syntax

C/C++ (DOS)

```
int _9812_Set_Clk_Src (int card_number, int
          clk_src, int ftpci)
```

C/C++ (Windows 95)

```
int W_9812_Set_Clk_Src (int card_number, int
          clk_src, int ftpci)
```

Visual Basic (Windows 95)

```
W_9812_Set_Clk_Src (ByVal card_number As Long,
          ByVal clk_src As Long, ByVal ftpci As
          Long) As Long
```

@ Argument

card_number: the card number of PCI-9812/10 to be selected

clk_src : the ADC clock source, the valid values are as follows:
0: INT_CLK: internal clock
1: SIN_CLK: external sin wave clock
2: SQR_CLK: external square clock

ftpci : Frequency selection.
AD2_GT_PCI: the frequency of A/D clock source is higher than PCI clock frequency.
AD2_LT_PCI: the frequency of A/D clock source is lower than PCI clock frequency.

@ Return Code

```
PCICardNumErr
PCICardNotInit
InvalidClkSrc, NoError
```

6.9 _9812_Set_Clk_Rate

@ Description

This function is used to specify the clock divider for ADC clock. The valid number of the clock divider is 2,4,6,8 …. 65534.

@ Syntax

C/C++ (DOS)

```
int _9812_Set_Clk_Rate (int card_number, U16
          clk_div)
```

C/C++ (Windows 95)

```
int W_9812_Set_Clk_Rate (int card_number, U16
          clk_div)
```

Visual Basic (Windows 95)

```
W_9812_Set_Clk_Rate (ByVal card_number As Long,
          ByVal clk_div As Integer) As Long
```

@ *Argument*

card_number: the card number of PCI-9812 to be selected
clk_div: the ADC clock divisor, this value must be an even number and the minimum value is 2.

@ *Return Code*

```
PCICardNumErr
PCICardNotInit
InvalidClkDiv, NoError
```

6.10 _9812_Set_Trig

@ *Description*

This function is used to set up a trigger. The function specifies the trigger mode, trigger level (voltage), trigger source, trigger slope and post trigger count. Please refer to section 4.4~4.7 for the detailed description of trigger setting.

@ *Syntax*

C/C++ (DOS)

```
int _9812_Set_Trig (int card_number, int
            trig_mode, int trig_src, int trig_pol,
            int trig_lvl, U16 post_trig_cnt)
```

C/C++ (Windows 95)

```
int W_9812_Set_Trig (int card_number, int
            trig_mode, int trig_src, int trig_pol,
            int trig_lvl, U16 post_trig_cnt)
```

Visual Basic (Windows 95)

```
W_9812_Set_Trig (ByVal card_number As Long,
            ByVal trig_mode As Long, ByVal
            trig_src As Long, ByVal trig_pol As
            Long, ByVal trig_lvl As Long, ByVal
            post_trig_cnt As Integer) As Long
```

@ *Argument*

card_number: the card number of PCI-9812 to be selected
trig_mode: selected trigger mode. The valid values are the following:
SOFT_TRIG: Software trigger
POST_TRIG: Post trigger
PRE_TRIG :Pre-trigger
DLY_TRIG : Delay trigger

MID_TRIG : Middle trigger

trig_src: selected trigger source, the valid trigger sources are:
CH0_TRIG : Channel 0
CH1_TRIG : Channel 1
CH2_TRIG : Channel 2
CH3_TRIG : Channel 3
AUX_TRIG : External digital trigger

trig_pol: trigger slope.
0: positive slope trigger
1: negative slope trigger

trig_lvl: trigger level. The relationship between the 8-bit trigger level and the trigger voltage is shown in section 4.5.

post_trig_cnt: The post trigger count. This value is preloaded to the post trigger counter when the post trigger counter register is written. It will count down on the rising edge of ADC sampling clock after the trigger condition is met. When the count reaches 0, the counter stops. The counter is used to control the delay time in delay trigger mode and to control the post trigger sampling count in middle trigger mode.

@ Return Code

```
PCICardNumErr
PCICardNotInit
InvalidClkDiv
NoError
```

6.11 W_9812_Alloc_DMA_Mem

@ Description

Contact Windows 95 system to allocate a block of contiguous memory for DMA transfer. This function is only available in Windows 95 version.

@ Syntax

C/C++ (Windows 95)

```
int W_9812_Alloc_DMA_Mem (U32 buf_size, HANDLE
          *memID, U32 *linearAddr)
```

Visual Basic (Windows 95)

```
W_9812_Alloc_DMA_Mem (ByVal buf_size As Long,
          memID As Long, linearAddr As Long) As
          Long
```

@ *Argument*

buf_size: Bytes to allocate. Please be careful, the unit of this argument is BYTE, not SAMPLE.

memID: If the memory allocation is successful, driver returns the ID of that memory in this argument. Use this memory ID in **W_9812_AD_DMA_Start** function call.

linearAddr: The linear address of the allocated DMA memory. You can use this linear address as a pointer in C/C++ to access the DMA data.

@ *Return Code*

```
NoError
AllocDMAMemFailed
```

6.12 W_9812_Free_DMA_Mem

@ *Description*

Release a system DMA memory under Windows 95 environment. This function is only available in Windows 95 version.

@ *Syntax*

C/C++ (Windows 95)

```
int W_9812_Free_DMA_Mem (HANDLE memID)
```

Visual Basic (Windows 95)

```
W_9812_Free_DMA_Mem (ByVal memID As Long) As
        Long
```

@ *Argument*

memID: The memory ID of the system DMA memory to deallocate.

@ *Return Code*

```
NoError
```

6.13 W_9812_Get_Sample

@ *Description*

For the language without pointer support such as Visual Basic, programmer can use this function to access the data in DMA buffer. This function is only available in Windows 95 version.

@ Syntax

C/C++ (Windows 95)

```
int W_9812_Get_Sample (U32 linearAddr, U32 index,
          I16 *ai_data)
```

Visual Basic (Windows 95)

```
W_9812_Get_Sample (ByVal linearAddr As Long,
          ByVal idx As Long, ai_data As Integer)
          As Long
```

@ Argument

linearAddr: The linear address of the allocated DMA memory.

index: The index of the sample. The first sample is with index 0.

ai_data: Returns the samples retrieved.

@ Return Code

`NoError`

7

Calibration

In data acquisition, calibrating your measurement devices to maintain the accuracy is very important. Users can calibrate the analog input and analog output channels under the users' operating environment to optimize the accuracy. This chapter will guide you to calibrate your PCI-9812/10 to an accurate condition.

7.1 What you need

Before calibrating your PCI-9812/10 card, you should prepare some equipment for the process:

- Calibration utility: Once the program is executed, it will guide you to do the calibration. This program is included in your delivered package.
- A voltage calibrator or a very stable and noise free DC voltage generator.

7.2 VR Assignment

There are eight variable resistors (VR) on the PCI-9812/10 board to allow you to make accurate adjustments on A/D channels. The function of each VR is specified in Table 2

VR1	A/D channel 0 offset adjustment
VR2	A/D channel 1 offset adjustment
VR3	A/D channel 2 offset adjustment
VR4	A/D channel 3 offset adjustment
VR5	A/D channel 0 full scale adjustment
VR6	A/D channel 1 full scale adjustment
VR7	A/D channel 2 full scale adjustment
VR8	A/D channel 3 full scale adjustment

Table 2. Functions of VRs

7.3 A/D Calibration

7.3.1 AD Calibration for Channel 0

1. Apply a +1V input signal to A/D channel 0, and trim the ***VR5*** to obtain the range of the average reading of channel 0 is within 2046.6±0.1 (9812) or 510.9±0.1 (9810).
2. Apply a +0V input signal to A/D channel 0, and trim the ***VR1*** to obtain the range of the average reading of channel 0 is within ±0.2.
3. Repeat step 1 and step 2, adjust ***VR5*** and ***VR1***.

7.3.2 AD Calibration for Channel 1,2,3

You could calibrate other channels with the same procedure in section 7.3.1 while trimming the corresponding VRs. Refer the following table to adjust the correct VR.

Channel Number	Full Scale Adjustment VR	Offset Adjustment VR
0	VR5	VR1
1	VR6	VR2
2	VR7	VR3
3	VR8	VR4

A calibration utility is provided in the software CD included in the product package. The detailed calibration procedures and description can be found in the utility. Users only need to run the software calibration utility and follow the procedures. You will get the accurate measurement data.

When used for the first time, the factory before shipping already calibrates the PCI-9812/10. So, users need not calibrate the PCI-9812/10 when you get it.

8

Software Utility

The software CD provides a utility program, 9812util.exe. This program has three functions, System Configuration, Calibration, and Functional Testing. This utility is designed as menu-driven based windowing style, that is, it provides not only the text messages for operating guidance, but also graphics to instruct you how to set hardware configuration correctly. This utility is described in the following sections.

8.1 Running 9812util.exe

After finishing the DOS installation, you can execute the utility by typing as follows (assume your utility is located in \ADLINK\DOS\9812\Util directory):

C> cd \ADLINK\DOS\9812\Util

C> 9812UTIL

The following diagram will be displayed on the screen. The message at the bottom of each window guides you how to select an item, go to the next step and change the default settings.

```
                ****** PCI-9812/10 Utility  Rev. 1.0 ******

Copyright © 1995-1997, ADLink Technology Inc.  All rights reserved.
```

```
                <F1> : Configuration.

                <F2> : Calibration.

                <F3> : Function testing.

                <Esc>: Quit.
```

```
>>> Select function key F1 ~ F3, or press <Esc> to quit. <<<
```

8.2 System Configuration

These functions will guides you to configure the PCI-9812/10 card, and set the right hardware configuration. The configuration window shows the setting items that you have to set before using the PCI-9812/10 card.

The following diagram will be displayed on the screen as you choose the Configuration function from main menu.

```
        ****** Configuration of PCI9812/10 ******
```

```
        <1> Card Type                   9812

        <2> ADC Trigger Source          CH0

        <3> Timer Clock Source          Internal

        <6> AD Input Range              Bipolar(-1V~1V)
```

```
>>> <Up/Down>: Select Item,  <PgUp/PgDn>: Change Setting <<<
```

8.3 Calibration

This function will guide you on how to calibrate the PCI-9812/10. The calibration program serves as a useful test for the PCI-9812/10's A/D,D/A and DIO functions and can aid in troubleshooting if problems arise.

Note: For an environment with frequent large changes in temperature and vibration, a recalibration interval, 3 months, is recommended. For laboratory conditions, 6 months to 1 year is acceptable

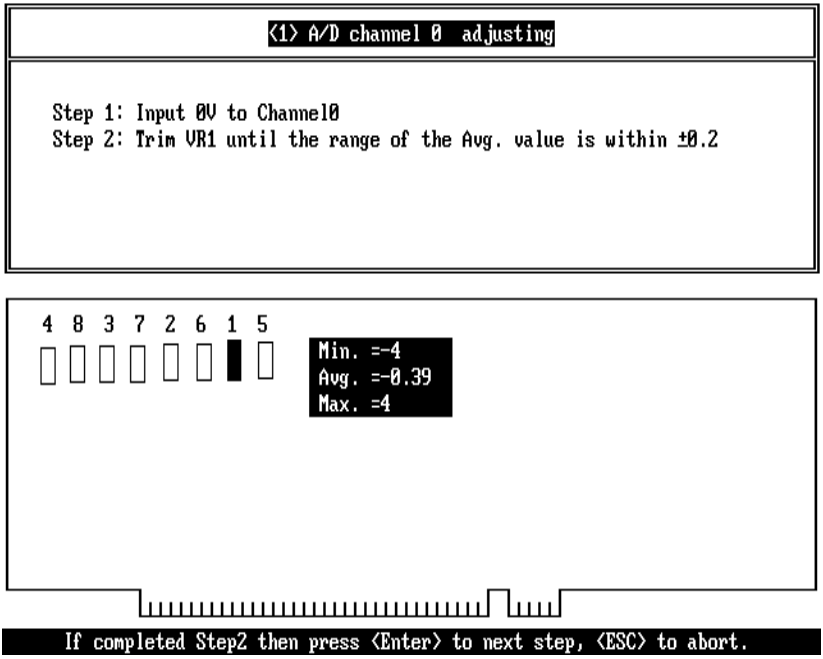
When you choose the calibration function from the main menu list, The calibration item menu is displayed on the screen. After you select one of the calibration item from the menu, a calibration window will appeor. The upper window shows the detailed procedures which you have to follow when proceeding the calibration. The instructions will tell you how to calibrate each item step by step. The bottom window shows the layout of PCI-9812/10. In addition, the proper Variable Resister (VR) will blink to indicate the related VR needs to be adjusted for the current calibration step.

```
****** PCI-9812 Calibration ******
```

```
<1> A/D channel 0 adjusting

<2> A/D channel 1 adjusting

<3> A/D channel 2 adjusting

<4> A/D channel 3 adjusting

<Esc> Quit
```

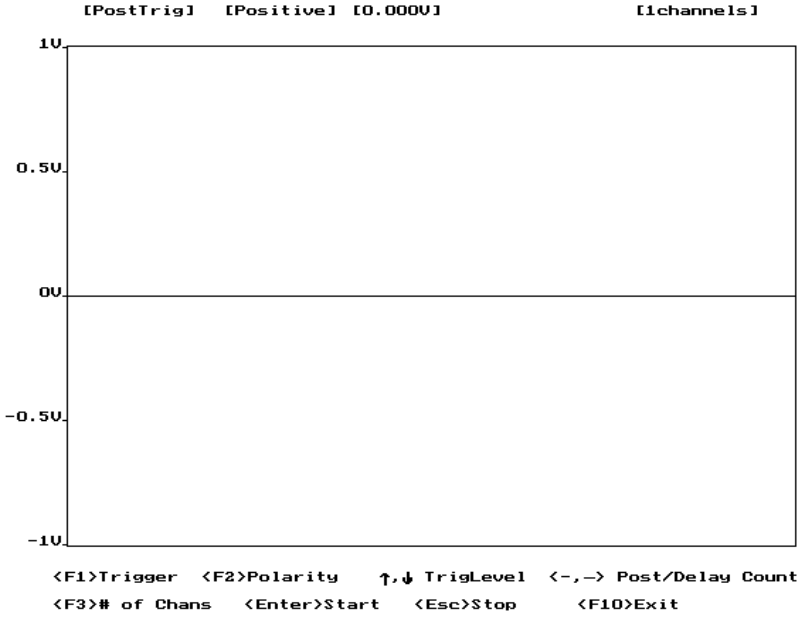
```
Select 1 to 4 or <Esc> to quit calibration.
```

For example, if you select 3, the following figure will be displayed on the screen:

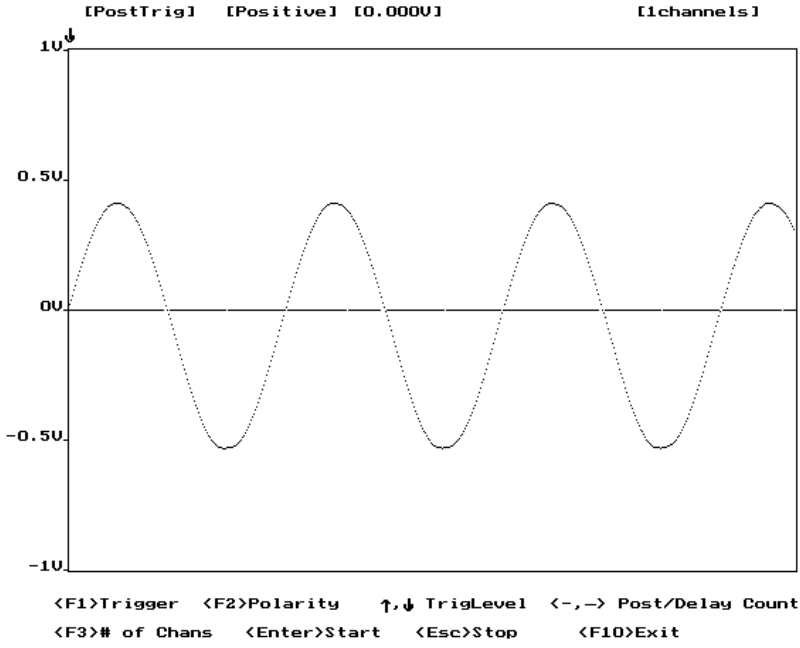


8.4 Functional Testing

This function is used to test the functions of PCI-9812/10 A/D. When you choose testing function from the main menu list, a function testing test window is displayed on the screen. The following is the figure of function testing window.



The message shown at the bottom indicates how to change the setting of trigger mode, trigger signal polarity, trigger level, channel number and post trigger count (for middle trigger and delay trigger). After you finishing the setting mentioned above, push “Enter” key to start performing the testing function. With this function, you can test and view the different effect of various trigger modes. In addition, an arrow shown on the screen indicates the trigger position. If the trigger source is also an enabled A/D channel, you can easily view the result of changing trigger levels. The following figure is a snapshot of the post-trigger testing.



Warranty Policy

Thank you for choosing ADLINK. To understand your rights and enjoy all the after-sales services we offer, please read the following carefully.

1. Before using ADLINK's products, please read the user manual and follow the instructions carefully. When sending in damaged products for repair, please attach an RMA application form.

2. All ADLINK products come with a two-year guarantee, free of repair charge.

    - The warranty period starts from the product's shipment date from ADLINK's factory
    - Peripherals and third-party products not manufactured by ADLINK will be covered by the original manufacturers' warranty
    - End users requiring maintenance services should contact their local dealers. Local warranty conditions will depend on the local dealers

3. Our repair service does not cover the two-year guarantee, if damages are cause by the following:

    a. Damage caused by not following instructions in the user menus.

    b. Damage caused by carelessness on the users' part during product transportation.

    c. Damage caused by fire, earthquakes, floods, lightening, pollution and incorrect usage of voltage transformers.

    d. Damage caused by unsuitable storage environments with high temperatures, high humidity or volatile chemicals.

    e. Damage caused by leakage of battery fluid when changing batteries.

    f. Damages from improper repair by unauthorized technicians.

    g. Products with altered and damaged serial numbers are not entitled to our service.

    h. Other categories not protected under our guarantees.

4. Customers are responsible for the fees regarding transportation of damaged products to our company or to the sales office.

5. To ensure the speed and quality of product repair, please download an RMA application form from our company website www.adlinktech.com. Damaged products with RMA forms attached receive priority.

For further questions, please contact our FAE staff.

ADLINK: service@adlinktech.com

Test & Measurement Product Segment: NuDAQ@adlinktech.com

Automation Product Segment: Automation@adlinktech.com

Computer & Communication Product Segment: NuPRO@adlinktech.com; NuIPC@adlinktech.com