NuDAQ PCI-9812/10

20MHz Simultaneous 4-CH

Analog Input Card

User's Manual

Manual Rev. 3.00

Revision Date: March 10, 2005

Part No: 51-11116-202

Recycled Paper

Advance Technologies; Automate the World.

The information in this document is subject to change without prior notice in order to improve reliability, design, and function and does not represent a commitment on the part of the manufacturer.

In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.

Trademarks

NuDAQ, PCI-9812, DAQBench, PCIS-DASK are registered trademarks of ADLINK Technology Inc. Other product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

Getting Service from ADLINK

Customer Satisfaction is top priority for ADLINK Technology Inc. Please contact us should you require any service or assistance.

ADLINK TECHNOLOGY INC.

Web Site: http://www.adlinktech.com

Sales & Service: Service@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 this completed service form for prompt and satisfactory service.

Company Information	
Company/Organization	
Contact Person	
E-mail Address	
Address	
Country	
TEL	FAX:
Web Site	
Product Information	
Product Model	
Environment	OS: M/B: CPU: Chipset: BIOS:

Please give a detailed description of the problem(s):

Table of Contents

List of Tables

List of Figures

1 Introduction

PCI-9812/10 is an advanced performance data acquisition card based on the 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 makes 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
- Onboard 32K words (samples) A/D FIFO memory
- Up to 20MHz A/D sampling rate
- Four single-ended analog input channels
- Bipolar input signals
- Four 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: four 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: 20MHz samples/sec

Note: For single channel enabled, the maximum sampling rate is 20MHz. For two channels enabled, the 20MHz sampling rate can be reached only when the number of samples accessed for each channel is smaller than 16K. For four channels enabled, the 20MHz 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 information on 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

- Number of channels:
  - Three TTL compatible inputs with 10KΩ pull down resistor
- Input Voltage:
  - Low: Min. 0V; Max. 0.8V
  - High: Min. +2.0V Max. 5.5V

  Input Load:
  - Low: ±1uA @0V
  - 0.5mA@ 5V
  - High: +2.7V min.@20mA max.

General Specifications

- Connectors: 5 BNC-type, one 10-pin header
- Operating Temperature: 0°C to 40°C
- Storage Temperature: -20°C to 80°C
- Humidity: 5 to 85%, non-condensing
- Power Consumption: +5V @ 2.5A (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 packages such as LabVIEW®, HP VEETM, DASYLabTM, InTouchTM, InControlTM, and ISaGRAFTM.

All software options are included in the ADLINK CD. Some software drivers are protected by licensing codes. Without the software code, the demo version can still be installed and used for two hours for trial/demonstration purposes. Please contact ADLINK dealers to purchase the formal license.

Programming Library

For customers 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, and 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 software drivers above are shipped with the board. Please refer to the “Software Installation Guide” to install these drivers.

PCIS-LVIEW: LabVIEW® Driver

PCIS-LVIEW contains the VIs which are used to interface with NI’s PCIS-LVIEW, which contains the VIs, which are used to interface with NI’s LabVIEW® software package. PCIS-LVIEW supports Windows 95/98/NT/2000. The LabVIEW® driver is shipped free with the board. These can be installed and used without a license. For more information on PCIS-LVIEW, please refer to the user’s guide in the CD (\\Manual_PDF\Software\PCIS-LVIEW).

PCIS-VEE: HP-VEE Driver

The PCIS-VEE includes 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. These can be installed and used without license. For more information on PCIS-VEE, please refer to the user’s guide in the CD (\\Manual_PDF\Software\PCIS-VEE).

DAQBenchTM: ActiveX Controls

For customers who are familiar with ActiveX controls and VB/VC++ programming, it is recommended that they use the DAQBenchTM ActiveX Control components library for developing applications. The DAQBenchTM is designed under Windows NT/98. For more information on DAQBench, please refer to the user’s guide in the CD (\\Manual_PDF\Software\DAQBench\DAQBench Manual.PDF).

DASYLabTM PRO

DASYLab is an easy-to-use software package, providing easy-setup instrument functions such as FFT analysis. Please contact ADLINK for DASYLab PRO, which includes DASYLab and ADLINK hardware drivers.

2 Installation

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, as these settings will not be changed after your data acquisition system configuration is decided. The configuration cannot be changed using only the software package when your system is running.

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

- Go through the checklist (section 2.1)
- Unpacking (section 2.2)
- Check the PCB (section 2.3)
- Install the hardware (section 2.4)

2.1 Checklist

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

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

If any of these items are missing or damaged, contact the dealer from whom this product was purchased. Save the shipping materials and carton for future shipping or storage.

2.2 Unpacking

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

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 they are properly seated. Do this only with the module placed on a flat firm surface.

Note:	Do not attempt to install a damaged board onto the computer.

You are now ready to install your card.

2.3 PCI-9812/10's Layout

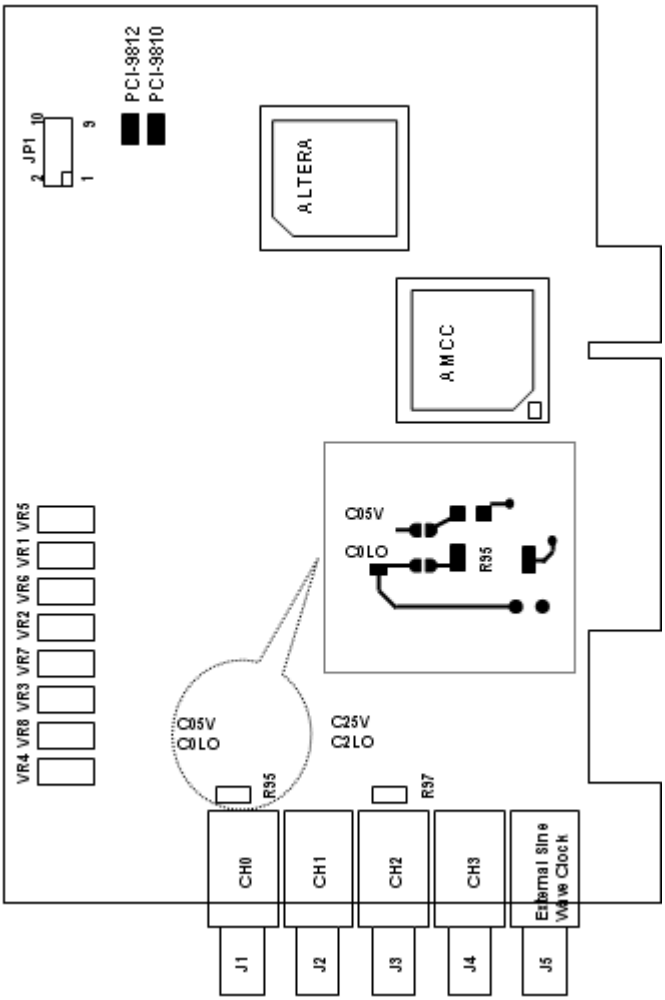


Figure 2-1: PCB Layout of the PCI-9812/10

2.4 Hardware Installation Outline

PCI configuration

The PCI cards (or CompactPCI cards) are equipped with a 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 recommended 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 Instructions

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 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 installing NuDAQ/NuIPC cards in your Windows system, you will be informed to enter the device information source. Please refer to the “Software Installation Guide” for instructions on installing the device.

3 Signal Connection

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

3.1 Connectors

PCI-9812/10 connects to external devices through five BNC connectors and one 10-pin dual-in-line header. Figure 3-1 shows the location of these connectors.

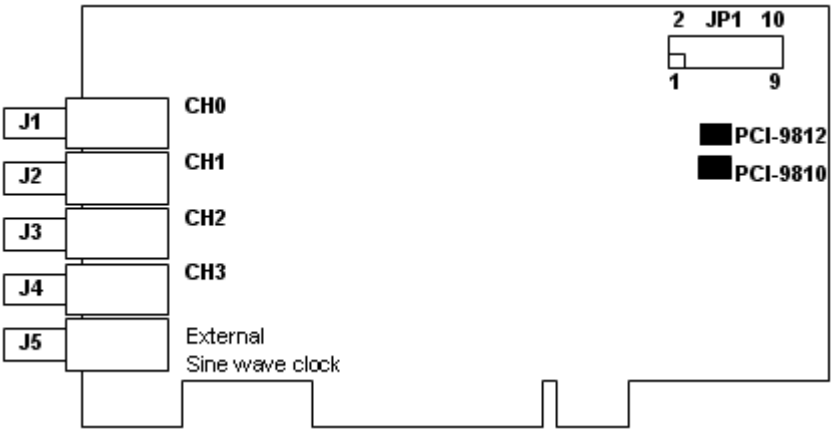


Figure 3-1: Location of connectors

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 one digital clock, one digital trigger and three digital inputs.

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

Table 3-1: Pin-out of JP1

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:

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

Table 3-2: Pin-out of JP1 connected to 9-pin D-type connector

3.2 Analog Input Impedance Setting

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

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. As all four channels use the same method to configure their input characteristics, only channel 0 is discussed here. There are two 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 figure 2-1).

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

Table 3-3: Analog Input

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, as the maximum input bias current of the OPAMP in the input stage is 35uA. Hence if the input is left unconnected, the OPAMP will be in an abnormal working environment leading to saturation in the output stage. Although a current-limiting resistor is used to protect the ADC, the large current brought by the saturation would damage the ADC.
2. Offset problems will result if one uses high impedance (~15MOhms) with signal sources having high output impedance. The high output impedance and the input bias current of up to 35uA introduces a voltage drop ranging several volts. Adjusting the variable resistor will not eliminate this large offset voltage.

Note: 75 ohm input impedance can be achieved by:

- Replacing R95 with a 75 ohm resistor and close C0LO.
- Placing 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

Table 3-4: Switches and Resistors

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 300KHz**, 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, explaining why users would see the strange sampled waveform when using a slower sine wave clock. If slower sampling rate is necessary when using 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.

External Clock 1

The external clock 1 (JP1 pin 1) is a digital clock. The input impedance is 50 ohms and the input level into the 50 ohm load should be between 2.4V and 5V. This signal is DC coupled.

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

Descriptions of the register format and structure of the PCI-9812/10 are specified in this chapter. This information is useful for programmers who want to handle the card using low-level programming.

4.1 I/O Port Address

PCI-9812/10 functions as a 32-bit PCI target device to any master on the PCI bus. It supports burst transfer to memory space by using 32-bit data therefore, both data read and write will be based on 32-bit data transfer. Table 4-1 lists the I/O address of each register with respect to the base address as well as the function of each register.

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 4-1: I/O Address

4.2 ADC Channel Enable Register

PCI-9812/10 has four analog input channels - 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 -- any value

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

---: Any value

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

--- : Any value

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

Table 4-2: Five Trigger Modes

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

4.5 Trigger Level Register

The trigger condition of 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

---: Any value

Relationship between the 8-bit trigger level and the trigger voltage:

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

Table 4-3: Relationship between 8-bit trigger level and trigger voltage

4.6 Trigger Source Register

PCI-9812/10 supports five trigger sources - 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, 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 the 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 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.

---: Any value

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, indicating 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, indicating 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: 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 -- Any value

4.9 FIFO Control Register

This register is used to control the onboard 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 onboard FIFO

When a "1" is written to this bit, the entire onboard 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 -- Any value

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 -- Any value

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 previously mentioned, hence 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 card can be controlled directly via the high-level Application-Programming-Interface (API), hence bypassing the detailed register structures. The software Libraries, including DOS Library for Borland C++ and DLL driver for Win-95, are included in the ADLINK CD. For further information, please refer to Chapter 6 “C/C++ Software Library”.

4.13 Low Level Programming

Users are not required to write any hardware dependent low-level programs to operate PCI-9812/10. Because it is more complex to control the PCI controller and the information is not described in this manual, ADLINK does not recommend its users to program its applications based on low-level programming. The PCI controller used in the PCI-9812 is AMCC-S5933. For further information on the s5933 PCI controller, please visit www.amcc.com.

5 Operation Theory

The operation theorem of the functions on the PCI-9812/10 card is described in this chapter. The functions include A/D conversion and digital input. The operation theorem would assist the user 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, the user must first 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 be familiar with the properties of the signal to be measured first. Users can decide which channels to use and connect the signals to

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 an internal or external clock source. The 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. 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, programmers should be familiar with 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, the following three concepts should be understood: 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, 1.25K, 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

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 350ns 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 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:

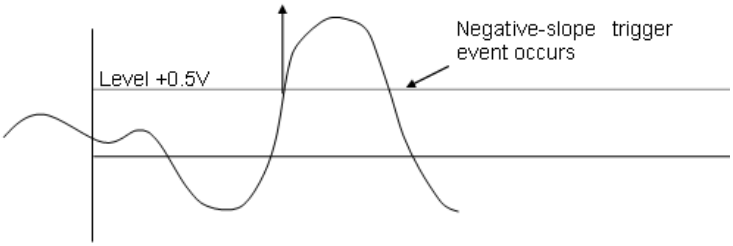
Positive-slope trigger

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.

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.

Trigger Modes

Software-Trigger Acquisition

This trigger mode does not require any external trigger source. The trigger event occurs when the _9812_AD_DMA_Start( ) function is called to start the operation.

Post-Trigger Acquisition

Use post-trigger acquisition in applications where data needs to be collected after a specified trigger event. The trigger can either be an external analog trigger or digital trigger.

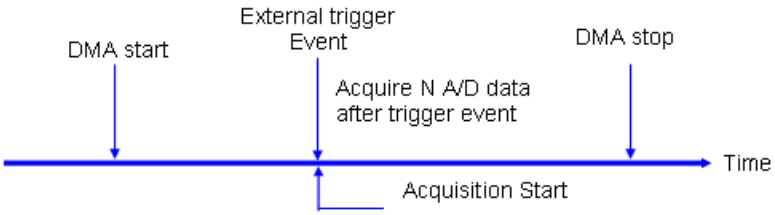


Figure 5-1: Post-Trigger Acquisition

Pre-Trigger Acquisition

Use pre-trigger acquisition 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 is read (specified by the _9812_AD_DMA_Start( ) function, please refer to section 6.2), the amount of retrieved data would be less 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.

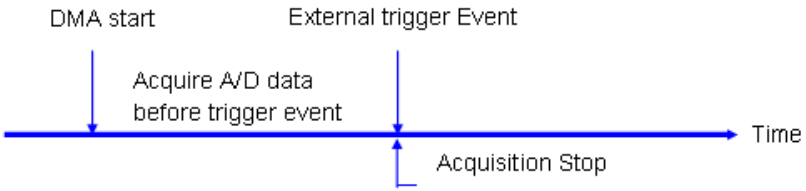


Figure 5-2: Pre-Trigger Acquisition

Middle-Trigger Acquisition

Use middle-trigger acquisition to collect data before and after a specified trigger event. The amount of data acquired before a trigger event occurs when using Middle trigger may not equal to the specified count of data, just like the pre-trigger mode.

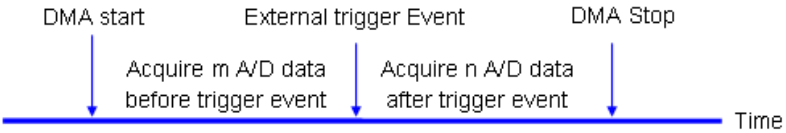


Figure 5-3: Middle-Trigger Acquisition

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 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. The counter then 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.

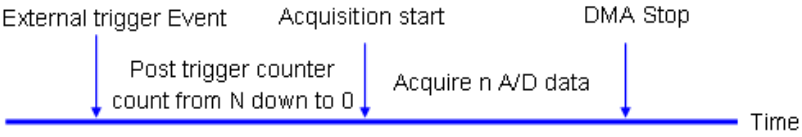


Figure 5-4: Delay Trigger Acquisition

5.4 A/D Clock Source Control

The AD clock source determines how the board regulates the timing of conversions when 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 a pacer clock but not single shot as the A/D converters are in a pipelined structure, which require eight conversion clocks to complete the conversion of digital data.

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.

These three clock sources are described below:

Internal Pacer Clock

An onboard 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 onboard

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)

External Pacer Clock

Users can connect an external pacer clock to the EXTCLK1 (pin 1) on JP1 (for square wave) or Ext. Sine wave clock (for sine wave). 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 the sampling frequency. The formula is shown below:

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 (max.), with the minimum divider value being 2. Please refer to section 6.2 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.

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 onboard 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

AD Data Transfer

For acquiring AD data, there are several function blocks on the PCI-9812 board. Even when the maximum sampling rate is specified up to 20MHz, 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.

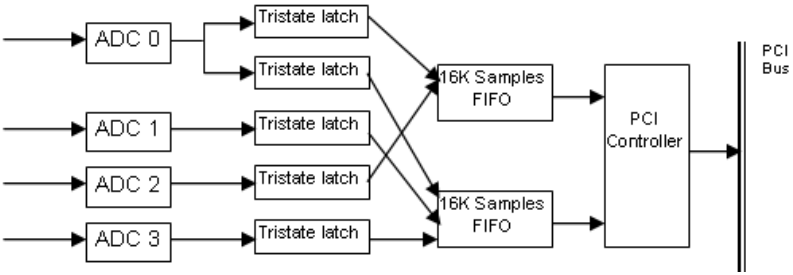


Figure 5-5: Block Diagram of PCI-9812

Simultaneous Sampling of four AD Channels

The PCI-9812/10 is equipped with four 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 applications, only channel 0 can be selected with the total FIFO length being 32K samples. For simultaneous two-channel applications, use channel 0 and channel 1 to optimize the FIFO usage. The onboard circuit does not allow the user to use channel 0 and 2 or channel 2 and 3 simultaneously. Please note for simultaneous sampling applications, the hardware only supports 2 or 4 channels but not 3-channel data acquisitions.

Total Data Throughput

When four channels start simultaneously, the total data throughput from the AD converter to the onboard FIFO memory would be:

Sampling Rate x number of channels x 2 bytes/channel

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

160MB/s = 20MHz x 4 channels x 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.

There is a total of 32K word (32K samples) when FIFO is onboard. When four 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 would need to calculate the total data throughput for their applications, as this value would be related to the total data length that can be continuously acquired.

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 simultaneously. If the total AD data throughput is lower than the PCI bandwidth, the AD data can be put into the host memory through Bus-mastering DMA, and the total acquiring data length could be up-to 64MB (32M samples) which is the maximum of the PCI controller. If the total AD data throughput is higher than the PCI bandwidth,

the maximum data length would be either 16K or 8K samples, which is the maximum length from the size of onboard FIFO.

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 onboard 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 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. ADLINK recommends using a high-level program library to manipulate this card. If the user wishes to program the software that can handle the DMA bus master data transfer, please refer to www.amcc.com for further information on the PCI Controller.

Host Memory Operation

Because of DMA data transfer, the AD data cannot be simultaneously processed when the data is transferred. Hence, the user should process the AD data after the completion of one DMA cycle. If the total data throughput in your application is relatively high (>20MB/s), the processing time of AD data as well as the CPU computation power consumption would need to be considered. For example, if the CPU can only process the data at the rate of 10MB/s, and the user wishes to continuously acquire data at the rate of 20MB/s; FIFO would eventually become full, and data acquisition become discontinuous.

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

Another limitation comes from the OS and host memory size. Under DOS environment, the maximum memory size that can be allocated is less than 640K except that you use extended memory managing software. Under Windows-95 or NT, it is relatively difficult to get a continuously large memory size such as 64MB. It is possible to allocate a large memory size by keeping the memory 'clean'. The PCI bus-mastering DMA controller of PCI-9812 needs continuous memory to store the AD data.

Summary

So far the AD data transferring, the maximum sampling rate, and the maximum continuous data length at the sample time have been discussed. Below is a summary of the steps discused:

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 two channels and 8K samples for four channels;
4. If it is lower, then the maximum total data size is 64M bytes, which comes from the limitation of host memory;
5. To acquire the AD data continuously (ie the data length is infinite); the 'double-buffered' software can be used. However, the CPU computation bandwidth still needs to be calculated 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). 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 to 1V)	V (Volts, -5V to 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 to 1V, Gain =1

For analog input range -5V to 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 in the ADLINK CD for 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 further information on the installation of software libraries for DOS, 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 to the PCIS-DASK user's guide and function reference in the ADLINK CD for programming information.

6.2 Programming Guide

Naming Convention

The functions of the NuDAQ PCI cards and NuIPC CompactPCI cards' software driver use the following naming convention rules:

In DOS Environment:

`_{hardware_model}_{action_name}.`

E.g. **`_9812_Initial().`**

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

To differentiate between DOS libraries and Windows 95 libraries, a capital "W" is placed in the beginning of each function name of the Windows 95 DLL driver. e.g. W_9812_Initial().

Data Types

Some data types were defined in Pci_9812.h (DOS) and Acl_pci.h (Windows 95). These data types are used by the NuDAQ Cards' library. It is recommended that these data types are used in your application programs. The following table lists 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

Table 6-1: Data Types

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. There are 10 cards with valid card numbers ranging from 0 to 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 IRQ value is automatically assigned by the 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 to 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, 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 conversions. 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 _9812_AD_DMA_Start is executed in the background, the function _9812_AD_DMA_Status can be used 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 amount of transferred data 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 starts from is in the 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, whether 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 amount 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 should be in the range of 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 sections 4.4 to 4.7 for further details on 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 as follows:

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: 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

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

@ 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. note that 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. This linear address can be used as a pointer in C/C++ to access the DMA data.

@ Return Code

NoError

AllocDMAMemFailed

6.12 W_9812_Free_DMA_Mem

@ Description

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

@ 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 languages without pointer support such as Visual Basic, programmers can use this function to access the data in DMA buffer. This function is only available in Windows 95.

@ 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 describes the calibration process for PCI-9812/10.

7.1 What you need

The following equipment would be required for the process:

- Calibration utility: The program would guide the user through the calibration process once executed. This program is included in the 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 that allows the user to make accurate adjustments on A/D channels. The function of each VR is specified in Table 7-1:

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 7-1: Functions of VRs

7.3 A/D Calibration

AD Calibration for Channel 0

1. Apply a +1V input signal to A/D channel 0, and trim the VR5 until the average reading of channel 0 is within the range of 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 until the average reading of channel 0 is within the range of ±0.2.
3. Repeat steps 1 and 2, adjust VR5 and VR1.

AD Calibration for Channels 1,2,3

The other channels can be calibrated using the steps shown in section 7.3 by trimming the corresponding VRs. Please refer to the following table when calibrating other channels.

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

Table 7-2: AD Calibration for Channels 1,2,3

A calibration utility is provided in the software CD, which is included in the package. Details of the calibration procedures and description can be found in this utility. Simply follow the instructions in the software calibration utility. The measurement data can also be found within.

Users do not need to calibrate the PCI-9812/10 when using for the first time as ADLINK has already calibrated it before shipping.

8 Software Utility

A utility program 9812util.exe is included in the software CD. 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 the user on setting the hardware configuration correctly. This utility is described in the sections below.

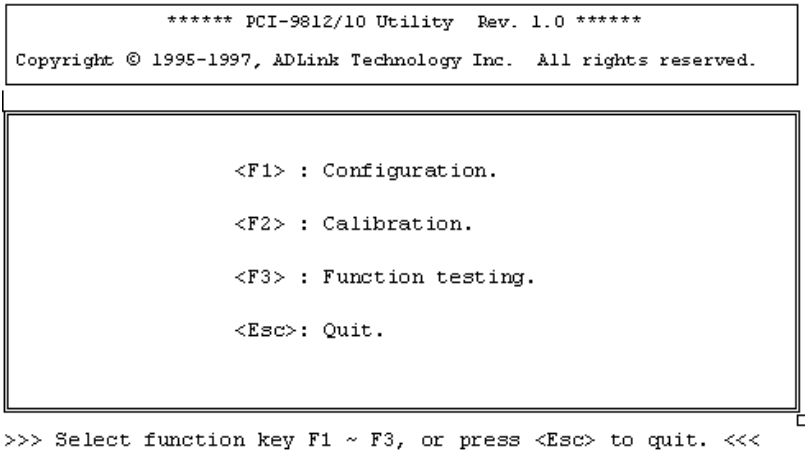
8.1 Running 9812util.exe

After finishing the DOS installation, the utility can be executed by typing the following (assuming your utility is located in \ADLINK\DOS\9812\Util directory):

C> cd \ADLINK\DOS\9812\Util

C> 9812UTIL

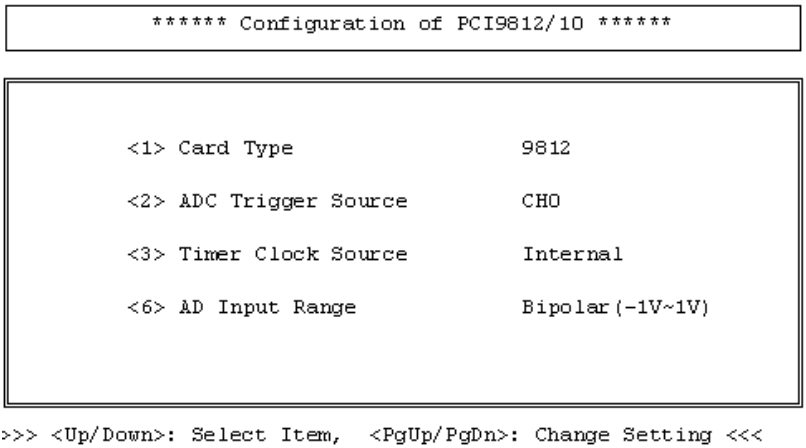
The following diagram will be displayed on the screen. Follow the instructions at the bottom of each window to change the default settings.



8.2 System Configuration

Use the functions in System Configuration to configure your PCI-9812/10 card.

The following diagram will be displayed on the screen when the Configuration function is chosen from main menu.



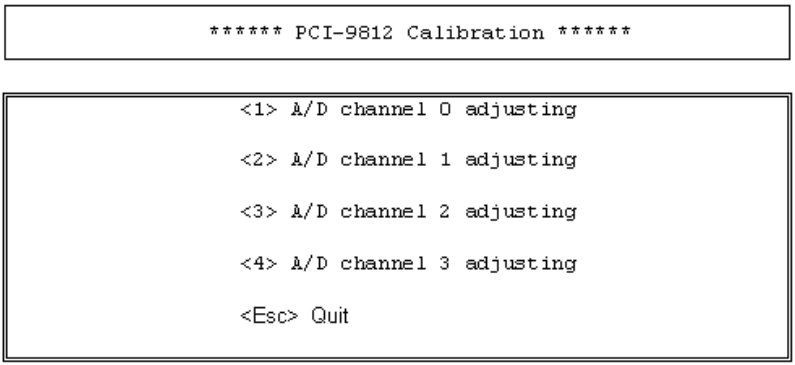
8.3 Calibration

This function takes the user through the calibration process of 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 be a useful tool when troubleshooting problems.

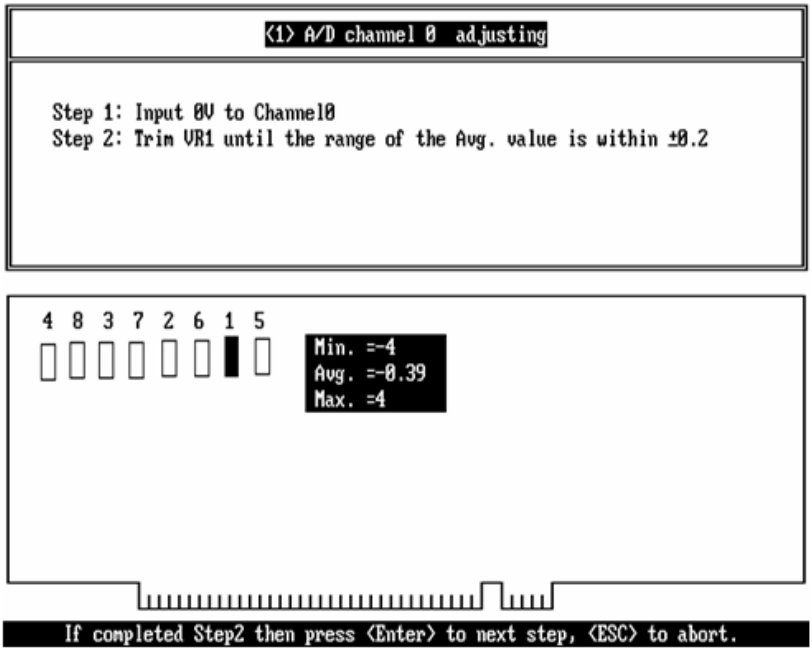
Note:	For an environment with frequent large fluctuations in temperature and vibration, a recalibration interval of 3 months is recommended. For laboratory conditions, 6 months to 1 year is acceptable.

When choosing the calibration function from the main menu list, the calibration item menu is displayed on the screen. After

selecting one of the calibration items from the menu, a calibration window will appear. The upper window shows the step-by-step calibration instructions. The lower 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.

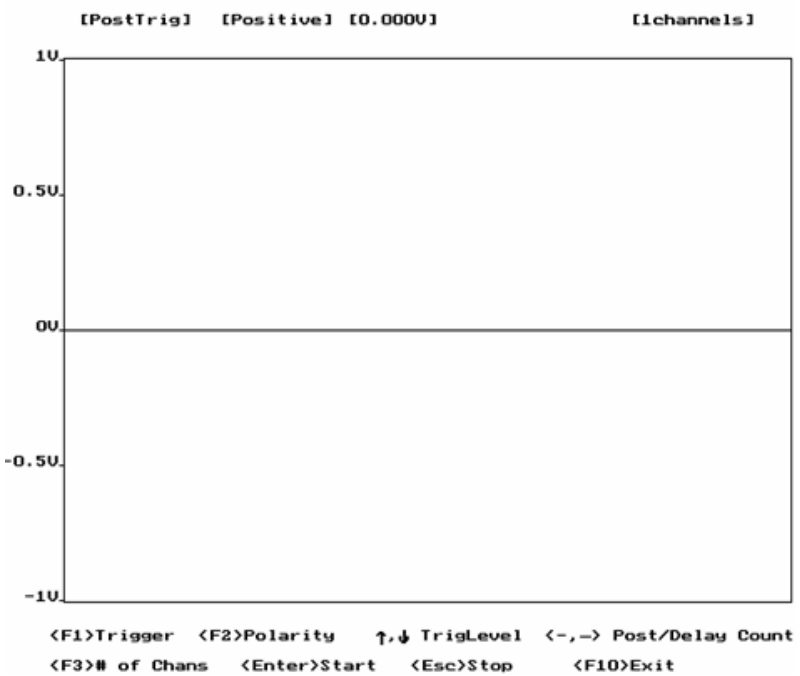


For example, if you select 3, the following window will be displayed on the screen:

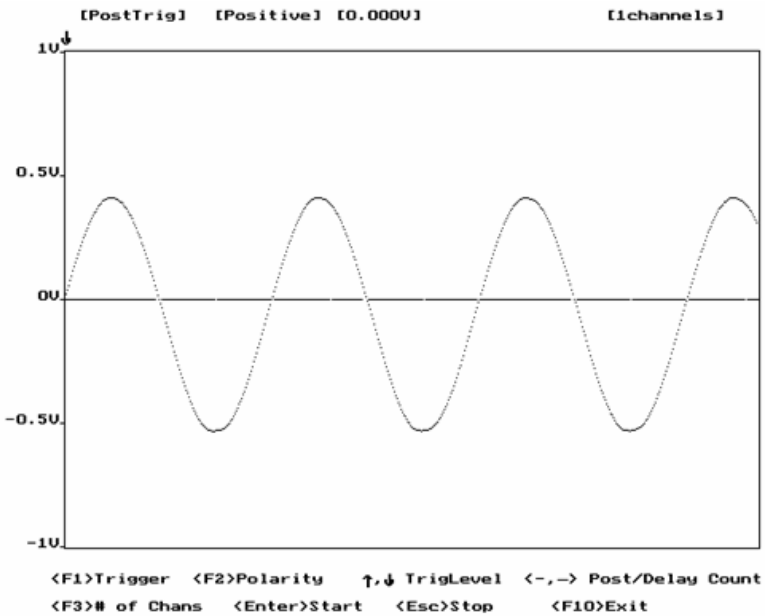


8.4 Functional Testing

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



The setting of trigger mode, trigger signal polarity, trigger level, channel number and post trigger count (for middle trigger and delay trigger) can be adjusted or changed by using the keys indicated. After finishing adjusting these settings, press “Enter” to start performing the test 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 diagram is a snapshot of the post-trigger testing.


[PostTrig] [Positive] [0.000V] [1channels]
1V
0.5V
0V
-0.5V
-1V
<F1>Trigger <F2>Polarity ↑,↓ TrigLevel <-,→ Post/Delay Count
<F3># of Chans <Enter>Start <Esc>Stop <F10>Exit

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 exactly. When sending in damaged products for repair, please attach an RMA application form which can be downloaded from: http://rma.adlinktech.com/policy/.
2. All ADLINK products come with a two-year guarantee:
   - 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.
   - For products containing storage devices (hard drives, flash cards, etc.), please back up your data before sending them for repair. ADLINK is not responsible for loss of data.
   - Please ensure the use of properly licensed software with our systems. ADLINK does not condone the use of pirated software and will not service systems using such software. ADLINK will not be held legally responsible for products shipped with unlicensed software installed by the user.
   - For general repairs, please do not include peripheral accessories. If peripherals need to be included, be certain to specify which items you sent on the RMA Request & Confirmation Form. ADLINK is not responsible for items not listed on the RMA Request & Confirmation Form.

3. Our repair service is not covered by ADLINK's two-year guarantee in the following situations:
   - Damage caused by not following instructions in the user's manual.
   - Damage caused by carelessness on the user's part during product transportation.
   - Damage caused by fire, earthquakes, floods, lightening, pollution, other acts of God, and/or incorrect usage of voltage transformers.
   - Damage caused by unsuitable storage environments (i.e. high temperatures, high humidity, or volatile chemicals).
   - Damage caused by leakage of battery fluid during or after change of batteries by customer/user.
   - Damage from improper repair by unauthorized technicians.
   - Products with altered and/or damaged serial numbers are not entitled to our service.
   - Other categories not protected under our warranty.
4. Customers are responsible for shipping costs to transport damaged products to our company or sales office.
5. To ensure the speed and quality of product repair, please download an RMA application form from our company website: http://rma.adlinktech.com/policy. Damaged products with attached RMA forms receive priority.

If you have any further questions, please email our FAE staff: service@adlinktech.com.