



PCI-AI/1216(M) Series Product Manual

MANUAL NUMBER : 00650-520-1B



FOREWORD

This product manual provides information to install, operate and or program the referenced product(s) manufactured or distributed by ICS Advent. The following pages contain information regarding the warranty and repair policies.

Check our Web site (www.icsadvent.com) for technical information, manuals, and BIOS updates. Technical assistance is also available at: **800-480-0044** (U.S.) or **858-677-0877** (international).

Customer Comments: If you experience any problems with this manual or just want to provide some feedback, please send us a message using the online "Service Form" on our Web site (www.icsadvent.com) under "Support." Detail any errors you find. We will correct the errors or problems as soon as possible and post the revised manual in our online Support Library. Thank you.

The information in this document is provided for *reference* only. ICS Advent does not assume any liability arising from the application or use of the information or products described herein. This document may contain or reference information and products protected by copyrights or patents and does not convey any license under the patent rights of ICS Advent, nor the rights of others.

Copyright © 1995, a California Corporation, 6260 Sequence Drive, San Diego, CA 92121-4371. ICS Advent is a Registered Trademark of ICS Advent. All trademarks and registered trademarks are the property of their respective owners. All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

Guarantee

A thirty day money-back guarantee is provided on all **standard** products sold. **Special order products** are covered by our Limited Warranty, *however they may not be returned for refund or credit. EPROMs, RAM, Flash EPROMs or other forms of solid electronic media are not returnable for credit - but for replacement only. Extended Warranty available. Consult factory.*

Refunds

In order to receive a refund on a product for the purchase price, the product must not have been damaged by the customer or by the common carrier chosen by the customer to return the goods and the product must be returned complete (meaning all manuals, software, cables, etc.) within 30 days of receipt and in an as-new and resalable condition. The **Return Procedure** must be followed to assure prompt refund.

Restocking Charges

Product returned *after* 30 days, and *before* 60 days, of the purchase will be subject to a **minimum** 20% restocking charge and charges for any damaged or missing parts. Products not returned within 60 days of purchase, or products which are not in an as-new and re-saleable condition, are not eligible for credit return and will be returned to the customer.

Limited Warranty

Effective April 1, 1998, all products carry a 2-year limited warranty. Within two years of purchase, ICS Advent will repair or replace, at our option, any defective product. ICS Advent will service the warranty for all standard catalog products for the first two years from the date of shipment. Please note: The 2-year warranty may not apply to special promotion items. Please consult the factory for warranty verification.

The limited warranty is void if the product has been subjected to alteration, neglect, misuse, or abuse; if any repairs have been attempted by anyone other than ICS Advent or its authorized agent; or if the failure is caused by accident, acts of God, or other causes beyond the control of ICS Advent or the manufacturer. Neglect, misuse, and abuse shall include any installation, operation, or maintenance of the product other than in accordance with the user's manual.

No agent, dealer, distributor, service company, or other party is authorized to change, modify, or extend the terms of this Limited Warranty in any manner whatsoever. ICS Advent reserves the right to make changes or improvements in any product without incurring any obligation to similarly alter products previously purchased.



Shipments not in compliance with this Guarantee and Limited Warranty Return Policy will not be accepted by ICS Advent.

Mission Critical Applied Computing Solutions

Return Procedure

For any Guarantee or Limited Warranty return, please contact ICS Advent's Customer Service at **800-480-0044** (U.S.) or **858-677-0877** (international) and obtain a Return Material Authorization (RMA) Number. All product(s) returned to ICS Advent for service or credit **must** be accompanied by a Return Material Authorization (RMA) Number. Freight on all returned items **must** be prepaid by the customer who is responsible for any loss or damage caused by common carrier in transit. Returns for Warranty **must** include a Failure Report for each unit, by serial number(s), as well as a copy of the original invoice showing date of purchase.

To reduce risk of damage, returns of product must be in an ICS Advent shipping container. If the original container has been lost or damaged, new shipping containers may be obtained from ICS Advent Customer Service at a nominal cost.

ICS Advent owns all parts removed from repaired products. ICS Advent uses new and reconditioned parts made by various manufacturers in performing warranty repairs and building replacement products. If ICS Advent repairs or replaces a product, its warranty term is not extended.

ICS Advent will normally return your replacement or repaired items via Second Day Air. Overnight delivery or delivery via other carriers is available at an additional charge.

Limitation of Liability

In no event shall ICS Advent be liable for any defect in hardware, software, loss, or inadequacy of data of any kind, or for any direct, indirect, incidental, or consequential damages in connection with or arising out of the performance or use of any product furnished hereunder. ICS Advent liability shall in no event exceed the purchase price of the product purchased hereunder. The foregoing limitation of liability shall be equally applicable to any service provided by ICS Advent or its authorized agent.

Some *Sales Items* and *Customized Systems* are **not** subject to the guarantee and limited warranty. However in these instances, any deviations will be disclosed prior to sales and noted in the original invoice. ***ICS Advent reserves the right to refuse returns or credits on software or special order items.***

Advisories

Three types of advisories are used throughout the manual to stress important points or warn of potential hazards to the user or the system. They are the Note, the Caution, and the Warning. Following is an example of each type of advisory:

Note: The note is used to present special instruction, or to provide extra information which may help to simplify the use of the product.



CAUTION!



A Caution is used to alert you to a situation which if ignored may cause injury or damage equipment.



WARNING!



A Warning is used to alert you of a situation which if ignored will cause serious injury.

Cautions and Warnings are accented with triangular symbols. The exclamation symbol is used in all cautions and warnings to help alert you to the important instructions. The lightning flash symbol is used on the left hand side of a caution or a warning if the advisory relates to the presence of voltage which may be of sufficient magnitude to cause electrical shock.

Use caution when servicing any electrical component. We have tried to identify the areas which may pose a Caution or Warning condition in this manual; however, ICS Advent does not claim to have covered all situations which might require the use of a Caution or Warning.

You must refer to the documentation for any component you install into a computer system to ensure proper precautions and procedures are followed.

Table of Contents

Chapter 1: Introduction	1-1
Chapter 2: Installation	2-1
Chapter 3: Option Selection	3-1
Chapter 4: Address Selection	4-1
Chapter 5: Programming	5-1
Chapter 6: Programmable Interval Timer	6-1
Chapter 7: Calibration	7-1
Chapter 8: Connector Pin Assignments	8-1

List of Figures

Figure 1-1: PCI-AI/1216 Block Diagram	1-5
Figure 3-1: PCI-AI/1216 Option Selection Map	3-1

List of Tables

Table 5-1: Register addresses on PCI-AI/1216(hex)	5-1
Table 8-1: Connector Pin Assignments	8-1

Revision 1B

December 2000

This page intentionally left blank

Chapter 1: Introduction

The PCI-AI/1216 and PCI-AI/1216(M) are multifunction high-speed analog-to-digital converter cards for use in 5-volt PCI bus computer slots. With these cards installed the computer can be used as a precision data acquisition and control system or as a signal analysis instrument.

Description

The card accepts up to eight differential or 16 single-ended analog input channels. Inputs are protected against over-voltage conditions up to ± 35 volts and typically survive static discharges beyond 4000 volts. The channel input configuration is software selectable. When using the card in differential mode the common mode rejection ratio is a minimum 80dB at 45KHz and the common mode voltage rejection capability is ± 11 V.

The input signals are amplified by an onboard instrumentation amplifier to provide voltage ranges of 10, 5, 1.25-6.25, and 1.25-3.75 volts unipolar and ± 10 , ± 5 , ± 2.5 and ± 1.25 volts bipolar. In addition, you can read 4-20 mA current inputs by manually installing jumpers to utilize onboard resistors. In this case, the current input is converted to an analog voltage range of 1.25-6.25V with full 12-bit resolution and there can be up to eight inputs. Each channel must be dedicated to either voltage or current.

These cards contain an industry standard 12-bit successive-approximation analog-to-digital converter (A/D) with a sample and hold amplifier input. Under ideal conditions throughput of over 100,000 conversions per second is possible.

Model PCI-AI/1216(M) has all of the foregoing plus a 2K samples FIFO data buffer and a 2K words point list buffer. These buffers give capability for full-speed counter-driven background data-acquisition with no computer interaction until the data FIFO needs draining. This allows 100,000KHz data to be taken with little processor overhead.

A/D conversions may be initiated in any of three ways: (a) software command, (b) an on-board programmable timer, or (c) direct external trigger. In turn, data may be transferred to the computer by any of two software selectable methods: (a) polling for end-of-conversion (EOC), or (b) an EOC interrupt. In the PCI-AI/1216(M) a third method of data transfer is available, using the onboard storage FIFO.

Input System Expansion

These cards can be used with up to 16 external AIM-16P analog input expansion cards. Each AIM-16P card provides capability for 16 differential inputs and thus can be used to provide a maximum of 256 analog inputs per combination of analog input expansion cards and a single PCI-AI/1216 or PCI-AI/1216(M). The initial AIM-16Ps are connected to the card by a special cable adapter and ribbon cable. Additional AIM-16P units are connected by daisy-chaining with ribbon cables. In the normal system configuration each AIM-16P is connected to a single-ended input of the PCI-AI/1216. However, if interconnection cable noise becomes problematic or if long

cables are required, then the AIM-16P cards can be connected to the differential inputs of the card. Using the differential connection allows better monitoring of remote locations because noise will be reduced by the common mode noise rejection capability of the card. (Note: In this configuration, the maximum number of inputs that can be handled is 128.)

The control register at base address+3 provides channel and gain control bits for use with submultiplexers like the AIM-16. SEL3-SEL0 are typically used for channel, and G2-G0 are used for gain. If no submultiplexer is being used, it is a good idea to use SEL3-SEL0 as ID tags. These values will comprise the top four bits of the data returned from the a/d converter.

Discrete Digital I/O

Four bits of TTL/CMOS compatible digital I/O capability are provided. The four bits of digital output are available with LSTTL logic levels and have high current drive capability of 500mA per line.

Counter/Timer

The cards contain a type 8254 counter/timer that has three 16-bit programmable down counters. Counter/Timer 0 is enabled by a pin on the connector (CTR0) and uses an external clock of up to 10 MHz as selected by user software. Counter/Timers 1 and 2 are concatenated to form a 32-bit counter and receive 1 MHz clock inputs from an on-board crystal-controlled oscillator.

The counter/timers can be set up for event counting, frequency or period measurements, and pulse or wave form generation. Also, Counter/Timers 1 and 2 can be programmed to initiate A/D conversions. A driver with source code is provided to facilitate use of the counter/timer chip in data acquisition applications.

Interrupts

These cards have interrupt capability within the PCI bus. The interrupt for the card can be software enabled and can be initiated by the completion of an A/D conversion, or in the PCI-AI/1216(M) version, by various FIFO status conditions.

Transferring Data into the Computer

These cards have been designed to provide high data throughput using the PCI bus. Direct I/O transfers provide the easiest means of transferring the data to memory. This type of transfer allows data to be sent at rates of over 100,000 samples/second.

Also, program transfers are subject to disruption by other interrupt processes in the computer. Use of real time triggering of the A/D assures synchronized sampling that is unaffected by other computer operations. That capability is essential in applications such as signal analysis, fast Fourier transform, and vibration and transient analysis where high data rates must be sustained for short intervals of time. The FIFO on the A version of the card allows the card to acquire data from a list of channels using a point-list FIFO and to store the data in the onboard data FIFO, eliminating most timing concerns.

Reference Voltage and Power Required

A +10V (± 0.25) reference voltage is available from the A/D reference source for external use. This reference output can source up to 200 mA of current. The PCI-AI/1216(M) requires only +5VDC and ± 12 VDC from the computer power supply.

Specifications

A/D Specification

- Type: Successive approximation.
- Resolution: 12 binary bits.
- Conversion Time: 8 microseconds max., 5.7 microseconds typical.
- Integral Linearity Error: ± 0.45 LSB maximum.
- Differential Non-Linearity: No missing codes.
- Monotonicity: Guaranteed over operating temperature range.
- Linearity: ± 1 bit.
- Zero Drift: ± 10 ppm/ $^{\circ}$ C maximum.
- Gain Drift: ± 45 ppm/ $^{\circ}$ C maximum.
- Trigger Source: Software selectable, external trigger, programmable timer, or program command.

Sample and Hold Amplifier

- Acquisition Time: 1 microsecond to 0.01% typical for full scale step function input.
- Aperture Uncertainty: 0.3 nanosecond typical.

Reference Voltage Output

- Voltage: +10VDC ± 0.25 VDC.
- Temp. Coefficient: ± 30 ppm/ $^{\circ}$ C.
- Load Drive: ± 200 mA maximum.

Digital I/O

Inputs

- Logic high: 2.0 to 5.0 VDC at 20 uA maximum at 2.7V.
- Logic low: -0.5 to +0.8 VDC at -0.4mA maximum.

Output

- Driver is ULN2003A, monolithic high-voltage, high-current Darlington transistor array. The collector-current rating of each Darlington pair is 500mA. Total of approximately 1.3A maximum for the four output lines at 25 °C .Applications include relay drivers, hammer drivers, lamp drivers, display drivers (LED and gas discharge), and line drivers.

Programmable Timer

- Type: 82C54-2 programmable interval timer.
- Counters: Three 16-bit down counters, two permanently concatenated with 1MHz clock as programmable timer.
- Output Drive: 2.2mA at 0.45V (5 LSTTL loads).
- Input Gate: TTL/DTL/CMOS compatible.
- Clock Frequency: DC to 10MHz.
- Active Count Edge: Negative edge.
- Min Clock Pulse Width: 30nS high/50nS low.
- Timer Range: 2.5 MHz to <1 pulse/hr.

Environmental

- Operating Temp: 0 °C. to 50 °C.
- Storage Temp: -20 °C. to +70 °C.
- Humidity: 0 to 90% RH, non-condensing.
- Power Required:
 - +5VDC: 600 mA typical.
 - +12VDC: 650 mA typical.
 - 12VDC: 650 mA typical.

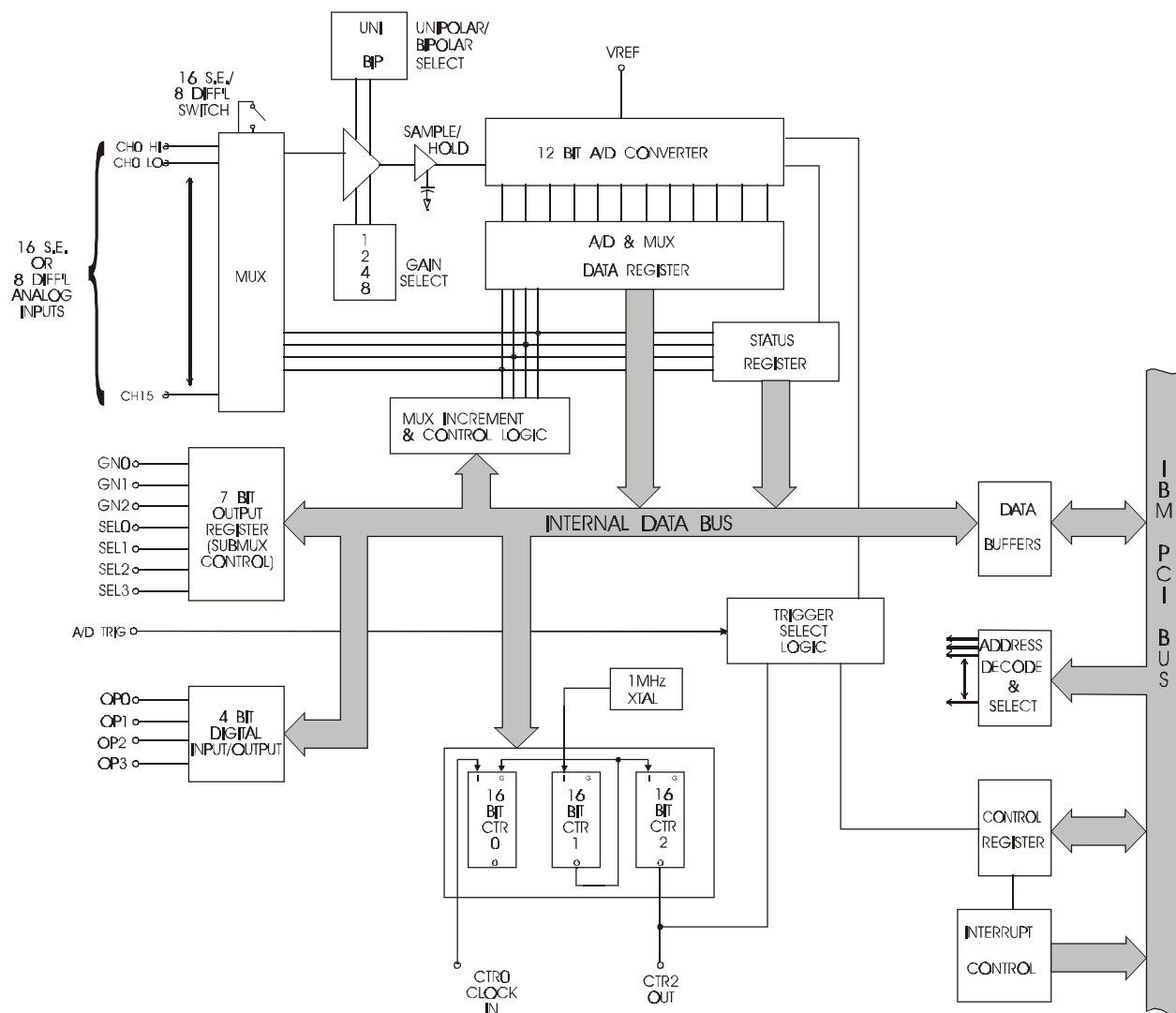


Figure 1-1: PCI-AI/1216 Block Diagram

Chapter 2: Installation

The software provided with this card is contained on either one CD or multiple diskettes and must be installed onto your hard disk prior to use. To do this, perform the following steps as appropriate for your software format and operating system. Substitute the appropriate drive letter for your CD-ROM or disk drive where you see d: or a: respectively in the examples below.

CD Installation

DOS/WIN3.x

1. Place the CD into your CD-ROM drive.
2. Type `D: Enter` to change the active drive to the CD-ROM drive.
3. Type `INSTALL Enter` to run the install program.
4. Follow the on-screen prompts to install the software for this card.

WIN95/98/NT

1. Place the CD into your CD-ROM drive.
2. The CD should automatically run the install program after 30 seconds. If the install program does not run, click START | RUN and type d:install, click OK or press `Enter`.
3. Follow the on-screen prompts to install the software for this card.

3.5-Inch Diskette Installation

As with any software package, you should make backup copies for everyday use and store your original master diskettes in a safe location. The easiest way to make a backup copy is to use the DOS DISKCOPY utility.

In a single-drive system, the command is:

```
D I S K C O P Y  A :  A : Enter
```

You will need to swap disks as requested by the system.

In a two-disk system, the command is:

```
D I S K C O P Y  A :  B : Enter
```

This will copy the contents of the master disk in drive A to the backup disk in drive B.

To copy the files on the master diskette to your hard disk, perform the following steps.

1. Place the master diskette into a floppy drive.
2. Change the active drive to the drive that has the diskette installed. For example, if the diskette is in drive A, type **A :** **Enter**.
3. Type **I N S T A L L** **Enter** and follow the on-screen prompts.

Directories Created on the Hard Disk

The installation process will create several directories on your hard disk. If you accept the installation defaults, the following structure will exist.

[CARDNAME]

Root or base directory containing the SETUP.EXE setup program used to help you configure jumpers and calibrate the card.

DOS\PSAMPLES: A subdirectory of [CARDNAME] that contains Pascal samples.

DOS\CSAMPLES: A subdirectory of [CARDNAME] that contains "C" samples.

Win32\language: Subdirectories containing samples for Win95/98 and NT.

WinRisc.exe

A Windows dumb-terminal type communication program designed for RS422/485 operation. Used primarily with Remote Data Acquisition Pods and our RS422/485 serial communication product line. Can be used to say hello to an installed modem.

ACCES32

This directory contains the Windows 95/98/NT driver used to provide access to the hardware registers when writing 32-bit Windows software. Several samples are provided in a variety of languages to demonstrate how to use this driver. The DLL provides four functions (InPortB, OutPortB, InPort, and OutPort) to access the hardware.

This directory also contains the device driver for Windows NT, ACCESNT.SYS. This device driver provides register-level hardware access in Windows NT. Two methods of using the driver are available, through ACCES32.DLL (recommended) and through the DeviceIOControl handles provided by ACCESNT.SYS (slightly faster).

SAMPLES

Samples for using ACCES32.DLL are provided in this directory. Using this DLL not only makes the hardware programming easier (MUCH easier), but also one source file can be used for both Windows 95/98 and WindowsNT. One executable can run under both operating systems and still have full access to the hardware registers. The DLL is used exactly like any other DLL, so it is compatible with any language capable of using 32-bit DLLs. Consult the manuals provided with your language's compiler for information on using DLLs in your specific environment.

VBACCES

This directory contains sixteen-bit DLL drivers for use with VisualBASIC 3.0 and Windows 3.1 only. These drivers provide four functions, similar to the ACCES32.DLL. However, this DLL is only compatible with 16-bit executables. Migration from 16-bit to 32-bit is simplified because of the similarity between VBACCES and ACCES32.

PCI

This directory contains PCI-bus specific programs and information. If you are not using a PCI card, this directory will not be installed.

SOURCE

A utility program is provided with source code you can use to determine allocated resources at run-time from your own programs in DOS.

PCIFind.exe

A utility for DOS and Windows to determine what base addresses and IRQs are allocated to installed PCI cards. This program runs two versions, depending on the operating system. Windows 95/98/NT displays a GUI interface, and modifies the registry. When run from DOS or Windows3.x, a text interface is used. For information about the format of the registry key, consult the card-specific samples provided with the hardware. In Windows NT, NTioPCI.SYS runs each time the computer is booted, thereby refreshing the registry as PCI hardware is added or removed. In Windows 95/98/NT PCIFind.EXE places itself in the boot-sequence of the OS to refresh the registry on each power-up.

This program also provides some COM configuration when used with PCI COM ports. Specifically, it will configure compatible COM cards for IRQ sharing and multiple port issues.

WIN32IRQ

This directory provides a generic interface for IRQ handling in Windows 95/98/NT. Source code is provided for the driver, greatly simplifying the creation of custom drivers for specific needs. Samples are provided to demonstrate the use of the generic driver. Note that the use of IRQs in near-real-time data acquisition programs requires multi-threaded application programming techniques and must be considered an intermediate to advanced programming topic. Delphi, C++ Builder, and Visual C++ samples are provided.

Findbase.exe

DOS utility to determine an available base address for ISA bus , non-Plug-n-Play cards. Run this program once, before the hardware is installed in the computer, to determine an available address to give the card. Once the address has been determined, run the setup program provided with the hardware to see instructions on setting the address switch and various option selections.

Poly.exe

A generic utility to convert a table of data into an nth order polynomial. Useful for calculating linearization polynomial coefficients for thermocouples and other non-linear sensors.

Risc.bat

A batch file demonstrating the command line parameters of RISCTerm.exe.

RISCTerm.exe

A dumb-terminal type communication program designed for RS422/485 operation. Used primarily with Remote Data Acquisition Pods and our RS422/485 serial communication product line. Can be used to say hello to an installed modem. RISCTerm stands for Really Incredibly Simple Communications TERMinal.

Installing the Card

These cards can be installed in a five-volt PCI slot of an IBM or compatible computer. Before installing the card, carefully read the Option Selection section of this manual and configure the card according to your requirements. Finally, our SETUP.EXE program will lead you through the process of setting the options on your card. The setup program does not set the options; the options must be set manually by jumpers on the card.

To Install the Card

1. Set switches and jumpers from either the Option Selection section of this manual or from the suggestions of SETUP.EXE.
2. Turn OFF computer power.
3. Remove the computer cover.
4. Install the card in an available PCI bus slot. (You may need to remove a backplate first.)
5. Inspect for proper fit of the card and tighten screws. Make sure that the card mounting bracket is properly screwed into place and that there is a positive chassis ground.
6. Replace the computer cover and turn ON the computer.
7. Enter the CMOS setup program of your system and verify that the PCI plug-and-play option is set appropriately for your system. Systems running Windows 95/98 (or any other PNP-compliant Operating System) should set the CMOS option to OS. Systems running under DOS, Windows NT 3.51/4.0, Windows 3.1, or any other non-PNP-compliant Operating System should set the PNP CMOS option to BIOS or Motherboard. Save the option and continue booting the system. (Note: This step can be skipped unless difficulties using the card arise.)

If you are using Windows 95/98, your operating system should detect the new hardware and prompt you for the installation disk. Insert the provided CD into a drive and allow the operating system to look for the files it wants. Please note that even though Windows 95/98 looks for and appears to find software to install, in general no driver is actually installed, used, or needed. This step merely tells Windows 95/98 where to place the card in the registry of devices. You can find the installed card in Device Manager under the class Data Acquisition.

The base address assigned by BIOS or the operating system can change each time new hardware is installed into or removed from the computer. Please recheck PCIFind or Device Manager if the hardware configuration is changed. Software you write can automatically determine the base address of the card using a variety of methods depending on the operating system. In DOS, the PCISOURCE directory shows the BIOS calls used to determine the address and IRQ assigned to installed PCI devices. In Windows, the Windows sample programs demonstrate querying the registry entries (created by PCIFind and NTIOPCI.SYS during boot-up) to determine this same information.

Chapter 3: Option Selection

To operate the cards in the current mode install a jumper in the positions shown at CH0 through CH7 for each channel, 0 to 7, that you will be using as a current input. Only differential operation is supported when using current mode inputs. For best results using 4-20mA inputs, the 1.25-6.25 volt range is recommended.

The CLAMP option provides a diode clamp on the Digital I/O lines, useful when switching inductive (coil) loads. Place a jumper in the CLAMP position if inductive loads will be used on DIO3-DIO0.

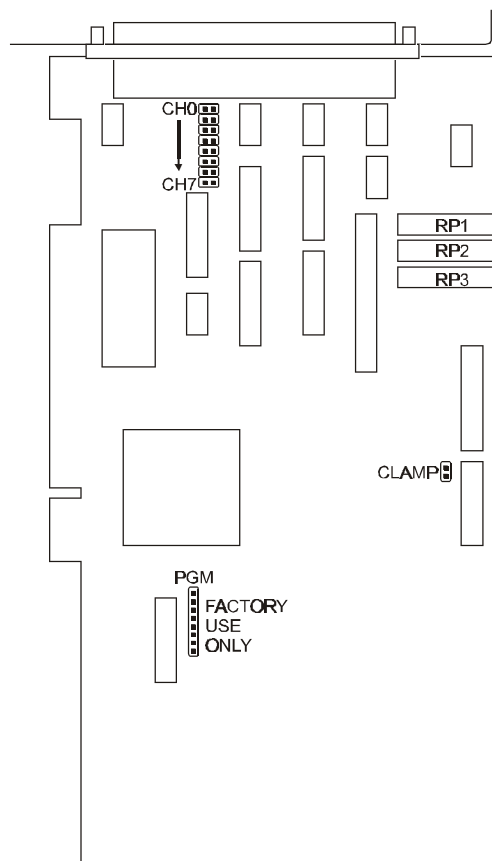


Figure 3-1: PCI-AI/1216 Option Selection Map

Chapter 4: Address Selection

These cards use I/O addresses offset from the selected base address assigned by the PCI bus. The address spaces are defined in the programming section of this manual.

PCI architecture is Plug-and-Play. This means that the BIOS or Operating System determines the resources assigned to PCI cards rather than the user selecting those resources with switches or jumpers. As a result, you cannot set or change the card's base address or IRQ level. You can only determine what the system has assigned.

To determine the base address that has been assigned, run the PCIFind.EXE utility program. This utility will display a list of all the ICS Advent cards detected on the PCI bus, the addresses assigned to each function on each of the cards, and the respective IRQs.

Alternatively, some operating systems (Windows 95/98/2000) can be queried to determine which resources were assigned. In these operating systems, you can use either PCIFind or the Device Manager utility from the System Properties Applet of the control panel. The card is installed in the Data Acquisition class of the Device Manager list. Selecting the card, clicking Properties, and then selecting the Resources tab will display a list of the resources allocated to the card.

The PCI bus supports 64K of I/O address space, so your card's addresses may be located anywhere in the 0000h to FFFFh range.

PCIFind uses the Vendor ID and Device ID to search for your card, then reads the base address and IRQ. If you want to determine the base address and IRQ without using PCIFind, use the following information:

The Vendor ID for these cards is 494F. (ASCII for "IO")

The Device ID for the PCI-AI/1216 is ACA8h.

The Device ID for the PCI-AI/1216(M) is ACA9h.

An example of how to locate PCI card resources is provided with in the PCI\SOURCE directory, under your installation directory. This code runs in DOS, and uses the PCI defined interrupt BIOS calls to query the PCI bus for card specific information. You will need the Device ID and Vendor ID listed above to use this code.

Chapter 5: Programming

This section of the manual is divided into two parts. One part deals with programming the PCI-AI/1216 and the second part deals with programming the PCI-AI/1216(M).

Programming PCI-AI/1216

The following table shows the register definition map for the PCI-AI/1216. Please refer to this table as you read the subsequent descriptions of the registers. See the PCI-AI/1216(M) section for details on that card.

Offset	Write Function	Read Function
0*	Start Conversion	A/D Conversion Data, 8 Least Significant Bits
1	unused	A/D Conversion Data, 4 Most Significant Bits
2*	A/D Control (Channel and Range)	A/D Control Readback
3	Digital Output / Sub-Mux Control	Digital Output Readback
4*	Option Control	Option Status, Clears Interrupt
5	Digital Output (general purpose 4 bits)	Digital Input (general purpose 4 bits)
6	unused	unused
7	unused	unused
8	8254 Counter / Timer 0	8254 Counter / Timer 0
9	8254 Counter / Timer 1	8254 Counter / Timer 1
A	8254 Counter / Timer 2	8254 Counter / Timer 2
B	8254 Counter / Timer Control	8254 Counter / Timer Status

* Offsets marked with this symbol have slightly different functionality in the PCI-AI/1216(M) card. See the next section for information on these differences.

Table 5-1: Register addresses on PCI-AI/1216(hex)

Register Descriptions

Offset 0 Write Start Conversion

This address is written to if you wish to start a conversion programmatically. Any value written to this address will cause the A/D to perform a conversion on the currently selected channel at the currently selected range. Use the A/D Control register at Offset 2 to select the channel and range. Conversions may also be started via an External Start Conversion signal (pin 4, see Offset 4) or via Counters (see Offset 4).

Offset 1 Write Unused

This address is unused, writes are ignored.

Offset 0 Read A/D Data (Low byte)**Offset 1 Read A/D Data (High nybble + SEL readback)**

These addresses contain the results of the most recent conversion. Data is presented in a two's complement form. Although the bytes can be read individually, coding is simplified by treating the data as a single 16-bit wide register at Offset 0.

The following diagram shows the bit positions of the data returned:

Read as 16-bit integer from Offset 0															
Read as byte from Offset 1								Read as byte from Offset 0							
sel3	sel2	sel1	sel0	Db	Da	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Read a single 16-bit integer and mask off the high-order 4 bits to obtain a value. See the programming guide at the end of this chapter for more information.

Sel3 through Sel0 are a readback of the state of the Sel3 through Sel0 bits as they were when the conversion was started. See Offset 3 for an explanation of the Sel bits.

**Offset 2 Write A/D Control Setup Channel, Range, and
Single-Ended or Differential Mode.**

Offset 2 Read A/D Control Readback

A write to this address selects the channel and range that the A/D will next convert. You must write to this register each time you wish to change the channel or range. If your program is only reading one channel and range, this register would be setup only once, during initialization. Normally, when converting more than one channel, or at more than one range, this register must be programmed before starting each conversion. Please note that a small settle time is necessary between setting this register and starting a conversion. This delay allows the channel multiplexer, the gain amplifiers, and the sample-and-hold circuitry time to settle to the new channel/range selected before the sample/hold circuit enters HOLD mode. The maximum required delay is approximately 7 microseconds.

The following diagram defines bit functions in the register

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
MA3	MA2	MA1	MA0	DIFF	R2	R1	R0

MA3, MA2, MA1, MA0

These 4 bits form a binary number from 0 to 15. This is the channel number on the A/D board you wish to setup for conversion. MA3 should be cleared to zero if using Differential mode.

DIFF

This is a control bit for Single-Ended or Differential mode. In Single-Ended mode, the card has 16 channels, each converted in reference to analog ground. In Differential mode, the card has only 8 channels, and the conversion returns the voltage difference from the high to the low channel. When the bit is cleared (0), Single-Ended mode is active; a set bit (1) indicates Differential mode.

R2, R1, R0

These bits select the range at which to convert the data. The following table lists the ranges and the bit combinations for each range.

R2	R1	R0	Range	Range
0	0	0	0	bipolar 10 volts, 20 volt span -10 to +10V
0	0	1	1	bipolar 5 volts, 10 volt span -5 to +5V
0	1	0	2	bipolar 2.5 volts, 5 volt span -2.5 to +2.5V
0	1	1	3	bipolar 1.25 volts, 2.5 volt span -1.25 to +1.25V
1	0	0	4	unipolar 10 volts, 10 volt span 0 to 10V
1	0	1	5	unipolar 5 volts, 5 volt span 0 to 5V
1	1	0	6	unipolar 1.25 to 3.75 volts, 2.5 volt span
1	1	1	7	unipolar 1.25 to 6.25 volts*, 5 volt span

*The 1.25-6.25 volt range is designed for use with the Current Mode jumper on each channel. Selecting this range for those channels that have been placed in current mode results in a 4-20mA range, with full 12-bit precision.

Reading this offset provides a readback of the current state of the register.

Refer to the programming guide at the end of this chapter for more information on using this register.

Offset 3 Write

Digital Output / Sub Mux Control

Offset 3 Read

Readback

This address provides seven bits of digital output. These bits can be used for any purpose, but are generally used to control the channel and range of an attached submultiplexer board, such as the AIM-16 or LVDT-8. The following table describes the register's bits and the names generally associated with Sub-Mux control bits.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SEL3	SEL2	SEL1	SEL0	X	G2	G1	G0

The bit X is unused.

The other seven bits can be used as general purpose output bits or as Sub-Mux control. When used to control a submultiplexer, the SEL bits normally control Sub-Mux channel, and the G bits control GAIN or Range, but the exact purpose of all these bits depends on the submultiplexer used. Please refer to your submultiplexer manual for more information. The SEL3-0 bits will be returned as the top four bits when the data from the A/D converter is read to the computer. They may be used to tag the data, if they are not being used for other purposes.

Offset 4 Write Option Control

This register is used to control various features on the card.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
XSCE					EI		CTR

CTR set to 1 causes start conversions to occur each time counter 2 (used in Mode 2) counts down to zero. The EI bit is Enable Interrupt. If the bit is set, each end-of-conversion generates an IRQ. Bit 7, XSCE, is External Start Conversion Enable. Setting this bit allows pin 4 on the connector to start conversions. All other bits in this register must be cleared to zero for compatibility with future options. Refer to the programming guide at the end of this chapter for more information on using this feature.

Offset 4 Read Status, Clear Interrupt

This register is read to determine the status of various features on the card. Primarily it is used to determine if a conversion has finished.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
BUSY							EXT

The BUSY bit indicates the conversion is in process. A zero means the A/D is busy, a one means a conversion is not in progress. This bit is one of the most important when not using interrupt modes. By reading this bit you can determine when the A/D conversion has finished and know when to take the data.

The EXT bit works like a digital input bit. It shows the current state of the External Start A/D pin on the connector (pin 4). A/D conversions can be started by pulling this pin high if XSCE is set to one in Offset 4. Unused status bits are reserved for expansion and should be ignored.

Offset 5 Write Digital Output, External Start Conversion Control
Offset 5 Read Digital Input

Four bits of general purpose I/O are provided. The least significant four bits of this register set or read the four general purpose digital I/O bits.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
X	X	X	X	DIO3	DIO2	DIO1	DIO0

Offset 8, 9, A and B 8254 Counter / Timer

Refer to the 8254 section, Chapter 6, for information on programming this device.

Programming PCI-AI/1216(M)

The following table shows the registers that are different in the A version of the card. Please refer to this table as you read the subsequent descriptions of the registers. See the previous PCI-AI/1216 section for details on the other registers.

Offset	Write Function	Read Function
0	Start Conversion	A/D Conversion Data, 8 Least Significant Bits
1	unused	A/D Conversion Data, 4 Most Significant Bits, ID
2	A/D Control (Channel and Range)	A/D Control Readback
3	Digital Output / Sub-Mux Control	Digital Output/Sub-Mux Control Readback
4	Option Control	Option Status

Register Descriptions

Registers that are not shown here have exactly the same functionality as described for the base model card.

Offset 0 Read A/D Data (Low byte)
Offset 1 Read A/D Data (High nybble, plus Sel Readback)

These addresses allow you to read the data stored in the data FIFO. The data is read oldest to newest. The format is the same as in the base model card described in the section above. The values of SEL3-0 are placed in the four most significant bits of Offset 1 and may be used as a data tag.

Since the data is contained in both of these registers, it is a good practice to read Offset 0 and Offset 1 with a single command to assure that the two data registers stay properly synchronized.

Offset 2&3 Write**A/D Control****Setup Point-List****Offset 2&3 Read****A/D Control****Readback**

This address (base+2) must be written to using a 16-bit WORD command. (out dx,ax) Doing so loads the point-list FIFO. Repeated writes to this address are used to load an entire point-list into the FIFO for later use. The format of the WORD is identical to the combined format of Offset 2 and Offset 3 from the base card. (Note: it is very useful to use SEL3-SEL0 as markers, as long as external muxes are not being used, to track which channel data is from. When no external mux is being used, writing the same value to SEL3-SEL0 as MA3-MA0 causes the data FIFO to appear to hold the channel number (MA3-0), which enables easier tracking of data and channel number.) After loading up to 2048 WORDS, you must read a WORD from this address before starting conversions.

Offset 4 Write**Option Control**

This register is used to control various features on the card.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
XSCE	CCF			CF	EIFH	EIFE	CTR

The CTR bit has the same functionality as before, but its usefulness is greatly enhanced due to the presence of a point-list FIFO. CCF and CF are Clear Channel FIFO and Clear FIFO (data) respectively. When a 1 is written to these bits, the selected FIFO is reset, effectively clearing its contents. EIFH set to 1 enables interrupts to occur when the data FIFO becomes Half Full. This is the interrupt most often used in background-task-based data acquisition. The EIFE bit set to 1 enables interrupts on Error. That is, anytime the FIFO is read from when it is already empty, or when data is lost because the FIFO is full when an A/D conversion finishes.

Offset 4 Read**Status**

This register is read to determine the status of various features on the card, primarily the state of the point-list and data FIFOs.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
BUSY	CFF	CFH	CF0	FF	FH	F0	EXT

BUSY and EXT have the same functionality as in the base model card.

CFF, CFH, CF0, FF, FH, and F0 indicate when LOW that the respective status is true. (C)FF is FIFO Full, (C)FH is FIFO Half Full, and (C)F0 means FIFO Empty.

Programming Guide

A/D Data Acquisition

Taking data from an A/D can seem confusing at first, but it all boils down to a few simple steps.

Set Channel and Range	Offset 2 (and 3 if using SubMux) Write
Start Conversion	Offset 0 Write
Wait for End of Conversion	Offset 4 Read
Read Data	Offset 0 Read (16-bit integer)
Repeat	

To set the channel, you must build the control register from its component parts. It consists of the channel, the range, and the differential or single-ended control bit. The channel is in the upper nybble, the SE/DIFF bit is next, then the range value. From the table in the programming section, you can pick the range number for the desired range, and in most programs the channel number will come from a FOR loop. Assuming the channel number is in CH, the range is in RNG, and the SE/DIFF bit is in SEDIFF, you can build a control byte with:

```
control = (CH << 4) + (SEDIFF << 3) + RNG;
```

This seems fairly complex, but it is made easier if you take into account that most programs will acquire data from a single range and usually at a single SE/DIFF setting. These two values, then, can be hard-coded into the program, and the code becomes:

```
control = (ch << 4) + 1; // for Single-ended,  $\pm 5$  VDC operation
```

That is all there is for polled mode A/D conversions. The data comes back in counts, and must be converted to engineering units to be useful. The data is returned in two's complement form, but resides in only 12 bits. To get a true two's complement integer, you must sign-extend the data. The following line reads the data and converts it to the simplest form:

```
data = ((int)inport(base + 0) << 4) >> 4; //C or C++  
data := (portw[base + 0] shl 4) DIV 16; {Pascal}
```

The variable "data" now holds a number from -2048 to 2047 for bipolar ranges, 0 to 4095 for unipolar. A simple $Y = mX + b$ equation (where X is the data) would convert this to engineering units.

Counter Driven Data Acquisition

Several things change when using timed A/D start (CTR mode).

Configure Counters	Offset 8-B Writes
Set Channel and Range	Offset 2 (and 3 if using SubMux) Write
Enable Counter Mode	Offset 4 Write
*loop	
Wait for Start of Conversion	Offset 4 Read
Set next channel and range	Offset 2 (and 3 if using SubMux) Write
Wait for End of Conversion	Offset 4 Read
Read Data	Offset 0 Read (16-bit integer)
Repeat at *loop	

Configuring the counters consists of setting up the timing for conversions using the 8254 control and load registers described in the 8254 section, Chapter 6, of this manual. Counters 1 and 2 should both be set for mode 2 operation, and the combined countdown duration of these two counters will be the delay between conversions.

When programming channels, you must not go faster than the counters' start conversion pulses. "Wait for Start of Conversion" paces your program to the counter. "Set next channel and range" solves the need for a settle-delay between set channel and start conversion. Because only the counter knows when it is going to start a conversion, programming the channel at this step ensures as long a delay as possible is provided.

Interrupt Based or Background Data Acquisition

Interrupt mode adds wrinkles of it's own.

Main program:

Install ISR	Motherboard resources
Initialize interrupts	Motherboard resources
Configure Counters	Offsets 8 through B
Set Channel and Range	Offset 2 Write
Enable Counter & IRQ Mode	Offset 4 Write
Wait until done.	Other code can go here.

ISR:

Clear Interrupt	Offset 4 Read
Read Data	Offset 0 Read
Set Channel and Range	Offset 2 Write
Send End-Of-Interrupt	Motherboard resources

Using interrupt mode requires an understanding of IRQ programming practices, such as the PIC IRQ Mask register and the IRQ vector table. Once these are grasped, however, the benefits are well worth the learning curve. IRQ mode is a great way to perform non-data gathering tasks while conversions are taking place. The background task (the ISR) takes data and stores it, while the foreground task can perform other functions (such as display, calculations, or acquisition from some other device).

FIFO Based Data Acquisition

Set Channel and Range FIFO	Offset 2 Write 16-bit WORDS
	Offset 2 Read
Configure Counters	Offset 8-B Writes
Enable Counter Mode	Offset 4 Write
*loop	
Wait for FIFO Half full	Offset 4 Read
Read Data until FIFO empty	Offset 0 Read (16-bit integer), Offset 4 Read

Loop at *loop until you have as much data as you want

This mode can be used to easily test point-list setup and general FIFO functionality, but generally won't be used in real-world environments. IRQ based FIFO handling is so efficient that most applications will use that mode instead. However, many similarities exist and this mode is worthwhile as a experimental or learning aid, as debugging this mode is easier than interrupt debugging.

Set Channel and Range FIFO consists of a predetermined set of WORD writes to Offset 2, setting the channel, range, and sub-mux channel and gain if used. The upper nybble (SEL3-SEL0) is returned in the upper nybble of the A/D conversion data, and is useful as a marker if no mux is to be used. If you are allowing a user to dynamically create the point-list, the Channel FIFO status bits in the Offset 4 register can warn you when the FIFO is full.

Interrupt-Based FIFO Mode

Main program:

Install ISR	
Initialize interrupts	
Configure Counters	Offset 8-B Writes
Set Channel and Range FIFO	Offset 2 Write 16-bit WORDS
	Offset 2 Read
Enable Counter Mode	Offset 4 Write
Enable IRQ on FIFO Half Full	(Same write as above)
Wait until all the data is taken.	Other code can go here.

Clear Interrupt	Offset 4 Read
Read Data until FIFO is empty (or read ½ of FIFO size)	
Send End-Of-Interrupt	

Be aware that the card will continually interrupt on the enabled IRQ conditions while those conditions are "true". This can cause the appearance of "locking-up" a system if the IRQ handler does not remove the condition before exiting. In the case of FIFO based acquisition, this generally means reading enough data from the data FIFO that it is no longer more than half-full.

Chapter 6: Programmable Interval Timer

The PCI-AI/1216(M) contains a type 8254 programmable counter/timer which allows you to implement such functions as a Real-Time Clock, Event Counter, Digital One-Shot, Programmable Rate Generator, Square-Wave Generator, Binary Rate Multiplier, Complex Wave Generator, and/or a Motor Controller. The 8254 is a flexible but powerful device that consists of three independent, 16-bit, presetable, down counters. Each counter can be programmed to any count between 1 or 2 and 65,535 in binary format, depending on the mode chosen.

On the PCI-AI/1216(M) these three counters are designated Counter #0, Counter #1, and Counter #2. Counter #0 is un-dedicated, with the gate, output and clock connections fully accessible via the I/O connector. Counter #0 is enabled by a discrete input and uses either the internal 1MHz clock or an external clock of up to 10MHz. Counters #1 and #2 are cascaded together to form a 32-bit counter. This dual counter can be enabled (gated) by program control and is clocked by a 1MHz precision crystal-controlled internal source.

Operational Modes

The 8254 modes of operation are described in the following paragraphs to familiarize you with the versatility and power of this device. For those interested in more detailed information, a full description of the 8254 programmable interval timer can be found in the Intel (or equivalent manufacturers) data sheets. The following conventions apply for use in describing operation of the 8254 :

Clock:	A positive pulse into the counter's clock input.
Trigger:	A rising edge input to the counter's gate input.
Counter Loading:	Programming of a binary count into the counter.

Mode 0: Pulse on Terminal Count

After the counter is loaded, the output is set low and will remain low until the counter decrements to zero. The output then goes high and remains high until a new count is loaded into the counter. A trigger enables the counter to start decrementing. This mode is commonly used for event counting with Counter #0.

Mode 1: Retriggerable One-shot

The output goes low on the clock pulse following a trigger to begin the one-shot pulse and goes high when the counter reaches zero. Additional triggers result in reloading the count and starting the cycle over. If a trigger occurs before the counter decrements to zero, a new count is loaded. Thus, this forms a re-triggerable one-shot. In mode 1, a low output pulse is provided with a period equal to the counter count-down time.

Mode 2: Rate Generator

This mode provides a divide-by-N capability where N is the count loaded into the counter. When triggered, the counter output goes low for one clock period after N counts, reloads the initial count, and the cycle starts over. This mode is periodic, the same sequence is repeated indefinitely until the gate input is brought low. This mode is used on the PCI-AI/1216(M) card in Counters #1 and #2 to generate periodic A/D start commands. This mode also works well as an alternative to mode 0 for event counting.

Mode 3: Square Wave Generator

This mode operates periodically like mode 2. The output is high for half of the count and low for the other half. If the count is even, then the output is a symmetrical square wave. If the count is odd, then the output is high for $(N+1)/2$ counts and low for $(N-1)/2$ counts. Periodic triggering or frequency synthesis are two possible applications for this mode. Note that in this mode, to achieve the square wave, the counter decrements by two for the total loaded count, then reloads and decrements by two for the second part of the wave form.

Mode 4: Software Triggered Strobe

This mode sets the output high and, when the count is loaded, the counter begins to count down. When the counter reaches zero, the output will go low for one input period. The counter must be reloaded to repeat the cycle. A low gate input will inhibit the counter. This mode can be used to provide a delayed software trigger for initiating A/D conversions.

Mode 5: Hardware Triggered Strobe

In this mode, the counter will start counting after the rising edge of the trigger input and will go low for one clock period when the terminal count is reached. The counter is retriggerable. The output will not go low until the full count after the rising edge of the trigger.

Programming

On the PCI-AI/1216(M), the 8254 counters occupy the following addresses:

- Base Address + 8: Read/Write Counter #0
- Base Address + 9: Read/Write Counter #1
- Base Address + A: Read/Write Counter #2
- Base Address + B: Write to Counter Control register

The counters are programmed by writing a control byte into a counter control register at Base Address + F. The control byte specifies the counter to be programmed, the counter mode, the type of read/write operation, and the modulus. The control byte format is as follows:

Base + B Write: Counter Control Byte

B7	B6	B5	B4	B3	B2	B1	B0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC0-SC1: These bits select the counter that the control byte is destined for.

SC1	SC0	Function
0	0	Program Counter #0
0	1	Program Counter #1
1	0	Program Counter #2
1	1	Read/Write Cmd.*

* See section on Reading and Loading the Counters.

RW0-RW1: These bits select the read/write mode of the selected counter.

RW1	RW0	Counter Read/Write Function
0	0	Counter Latch Command
0	1	Read/Write LS Byte
1	0	Read/Write MS Byte
1	1	Read/Write LS Byte, then MS Byte

M0-M2: These bits set the operational mode of the selected counter.

Mode	M2	M1	M0
0	0	0	0
1	0	0	1
2	X	1	0
3	X	1	1
4	1	0	0
5	1	0	1

BCD: Set the selected counter to count in binary (B0 = 0) or BCD (B0 = 1).

Reading and Loading the Counters

If you attempt to read the counters on the fly when there is a high input frequency, you will most likely get erroneous data. This is partly caused by carries rippling through the counter during the read operation. Also, the low and high bytes are read sequentially rather than simultaneously and, thus, it is possible that carries will be propagated from the low to the high byte during the read cycle.

To circumvent these problems, you can perform a counter-latch operation in advance of the read cycle. To do this, load the RW1 and RW2 bits with zeroes. This instantly latches the count of the selected counter(selected via the SC1 and SC0 bits) in a 16-bit hold register. (An alternative method of latching counter(s) which has an additional advantage of operating simultaneously on several counters is by use of a readback command to be discussed later.) A subsequent read operation on the selected counter returns the held value. Latching is the best way to read a counter on the fly without disturbing the counting process. You can only rely on directly read counter data if the counting process is suspended while reading, by bringing the gate low, or by halting the input pulses.

For each counter you must specify in advance the type of read or write operation that you intend to perform. You have a choice of loading/reading (a) the high byte of the count, or (b) the low byte of the count, or (c) the low byte followed by the high byte. This last is of the most general use and is selected for each counter by setting the RW1 and RW0 bits to ones. Of course, subsequent read/load operations must be performed in pairs in this sequence or the sequencing flip-flop in the 8254 chip will get out of step.

Base + B Read: Counter Control Byte

B7	B6	B5	B4	B3	B2	B1	B0
1	1	CNT	STA	C2	C1	C0	0

CNT: When is 0, latches the counters selected by bits C0-C2.

STA: When is 0, returns the status byte of counters selected by C0-C2.

C0, C1, C2: When high, select a particular counter for readback. C0 selects Counter 0, C1 selects Counter 1, and C2 selects Counter 2.

You can perform two types of operations with the Counter Control Byte. When CNT=0, the counters selected by C0 through C2 are latched simultaneously. When STA=0, the counter status byte is read when the counter I/O location is accessed. The counter status byte provides information about the current output state of the selected counter and its configuration. The status byte returned if STA=0 is:

B7	B6	B5	B4	B3	B2	B1	B0
OUT	NC	RW1	RW2	M2	M1	M0	BCD

OUT: Current state of counter output pin.
 NC: Null count. This indicates when the last count loaded into the counter register has actually been loaded into the counter itself. The exact time of load depends on the configuration selected. Until the count is loaded into the counter itself, it cannot be read.
 RW1 & RW0: Read/Write command.
 M2, M1, M0: Counter mode.
 BCD: BCD = 0 is binary mode, otherwise counter is in BCD mode.

If both STA and CNT bits in the counter status byte are set low and the RW1 and RW0 bits have both been previously set high in the counter control register (thus selecting two-byte reads), then reading a selected counter address location will yield:

1st Read: Status byte
 2nd Read: Low byte of latched data
 3rd Read: High byte of latched data

After any latching operation of a counter, the contents of its hold register must be read before any subsequent latches of that counter will have any effect. If a status latch command is issued before the hold register is read, then the first read will read the status, not the latched value. In this case, the latched value may be read after reading the status.

Programming Examples

Generating a Square Wave Output

To program Counters #1 and #2 for a 1 KHz output you need to divide the 1MHz crystal oscillator input by 1,000. To obtain a symmetrical waveform, the divisor loaded into the counter should be an even number. If it is an odd number, then one half of the waveform would be one input clock pulse period longer than the other. A convenient divisor to use in counter #1 is 10 and counter #2 is 100 (because $10 \times 100 = 1,000$).

```

outportb(BASEADDRESS + B, 0x76); /* counter #1 to square wave mode */
outportb(BASEADDRESS + B, 0xb6); /* counter #2 to square wave mode */
outportb(BASEADDRESS + 9,10); /* write lower byte, counter #1 */
outportb(BASEADDRESS + 9,0); /* write upper byte, counter #1 */
outportb(BASEADDRESS + A,100); /* write lower byte, counter #2 */
outportb(BASEADDRESS + A,0); /* write upper byte, counter #2 */

```

Using Counter #0 as a Pulse Counter

Since the counters are "down" counters the resulting count is subtracted from the starting value to determine the number of pulses. This example starts with a value of 65,535:

```
outportb(BASEADDRESS + B,0x30);    /* counter #0, mode 0 */
outportb(BASEADDRESS + 8,0xff);    /* counter #0 low load byte */
outportb(BASEADDRESS + 8,0xff);    /* counter #0 high load byte */
```

Measuring Frequency and Period

The two previous sections show how to count pulses and generate output frequencies. It is also possible to measure frequency by raising the gate input of Counter 0 for a known time interval and counting the number of clock pulses accumulated for that interval. The gating signal can be derived from Counters #1 and #2 operating in a square wave mode.

Counter #0 can also be used to measure pulse width or half period of a periodic signal. The signal should be applied to the gate input of Counter #0 and a known frequency (such as the 1MHz crystal controlled oscillator) applied to the Counter #0 clock input. During the interval when the gate input is low, Counter #0 is loaded with a full count of 65,535. When the gate input goes high, the counter begins decrementing until the gate input goes back low at the end of the pulse. The counter is then read and the change in count is a linear function of the duration of the gate input signal. Longer pulse durations can be measured if Counters #1 and #2 are used as the input clock source for Counter #0, or by using an external clock source.

Generating Time Delays

There are four methods of using Counter #0 to generate programmable time delays.

Pulse on Terminal Count

After loading, the counter output goes low. Counting is enabled when the gate goes high. The counter output will remain low until the count reaches zero, at which time the counter output goes high. The output will remain high until the counter is reloaded by a programmed command. If the gate goes low during countdown, counting will be disabled as long as the gate input is low.

Programmable One-Shot

The counter need only be loaded once. The time delay is initiated when the gate input goes high. At this point the counter output goes low. If the gate input goes low, counting continues but a new cycle will be initiated if the gate input goes high again before the timeout delay has expired; i.e., is re-triggerable. At the end of the timeout, the counter reaches zero and the counter output goes high. That output will remain high until re-triggered by the gate input.

Software Triggered Strobe

This is similar to Pulse-on-Terminal-Count except that, after loading, the output goes high and only goes low for one clock period upon timeout. Thus, a negative strobe pulse is generated a programmed duration after the counter is loaded.

Hardware Triggered Strobe

This is similar to Programmable-One-Shot except that when the counter is triggered by the gate going high, the counter output immediately goes high, then goes low for one clock period at timeout, producing a negative-going strobe pulse. The timeout is re-triggerable; i.e., a new cycle will commence if the gate goes high before a current cycle has timed out.

Generating Interrupts with the Counter/Timer

The PCI-AI/1216(M) architecture allows you to directly generate an interrupt upon an ADC end of conversion or the timeout of Counter #2. This allows you to either read the converted data or use the counter while using external start of conversions.

Note also that it is possible to trigger the A/D externally or by a programmed write to an I/O port and invoke an interrupt at the end of A/D conversion in the same way.

Chapter 7: Calibration

The cards are calibrated through a software routine contained in the setup program. Once the card is setup, run the calibration sequence as shown in the program and make the calibration adjustments on RP1, RP2, and RP3 on the board.

Chapter 8: Connector Pin Assignments

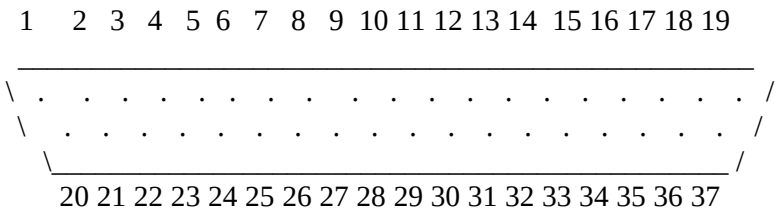
The analog inputs are accessible via a male 37-pin D type connector that extends through the back of the computer case and a DB37S solder cup plug may be used to make connections.

Pin	Name	Function
1	VDD	+12VDC Power from the Computer Bus
2	CLK0	Counter 0 Clock Input
3	G0	LSBit of AIM-16P Gain Select Bits
4	EXT	External Start Conversion Source Pin
5	G1	Bit 1 of AIM-16P Gain Select Bits
6	G2	MSBit of AIM-16P Gain Select Bits
7	SEL0	Multiplexer Channel Selection Bit LSBit
8	SEL1	Multiplexer Channel Selection Bit
9	SEL2	Multiplexer Channel Selection Bit
10	SEL3	Multiplexer Channel Selection MSBit
11	Ground	Low Level Analog Common Ground
12	GATE1	Counter/Timer Gate Pulse
13	OUT2	Counter/Timer 2 Counter Output
14	DI/O 0	Digital I/O LSB
15	DI/O 1	Digital I/O Bit
16	DI/O 2	Digital I/O Bit
17	IN8-	Chl 7 Analog Low Input (Diff'l) Chl 15 Analog High Input (SE)
18	Ground	Low Level Analog Common Ground
19	VREF	+10VDC Reference Voltage
20	VSS	-12VDC
21	IN7-	Chl 6 Analog Low Input (Diff'l) Chl 14 Analog High Input (SE)
22	IN6-	Chl 5 Analog Low Input (Diff'l) Chl 13 Analog High Input (SE)
23	IN5-	Chl 4 Analog Low Input (Diff'l) Chl 12 Analog High Input (SE)
24	IN4-	Chl 3 Analog Low Input (Diff'l) Chl 11 Analog High Input (SE)
25	IN3-	Chl 2 Analog Low Input (Diff'l) Chl 10 Analog High Input (SE)
26	IN2-	Chl 1 Analog Low Input (Diff'l) Chl 9 Analog High Input (SE)
27	IN1-	Chl 0 Analog Low Input (Diff'l) Chl 8 Analog High Input (SE)
28	DI/O 3	Digital I/O MSB
29	+5VDC	+5VDC Power from Computer Bus
30	IN8	Chl 7 Analog High Input
31	IN7	Chl 6 Analog High Input
32	IN6	Chl 5 Analog High Input
33	IN5	Chl 4 Analog High Input
34	IN4	Chl 3 Analog High Input
35	IN3	Chl 2 Analog High Input
36	IN2	Chl 1 Analog High Input
37	IN1	Chl 0 Analog High Input

Table 8-1: Connector Pin Assignments

Note

The figure below shows how pins are numbered on D type connectors.



Declaration of Conformity

Information Technology Equipment



6260 Sequence Drive
San Diego, CA 92121-4371
(800) 523-2320 / (858) 677-0877

The product(s) covered by this declaration:

PCI-AI/1216 and PCI-AI/1216M

The European Union directives covered by this declaration:

EMC Directive 89/336/EEC and Low Voltage Directive 73/23/EEC

The basis on which conformity is declared:

EN 50081-1:1992 Emissions, Generic Requirements

-EN 55022 Limits and Methods of Measurement of Radio Disturbance Characteristics of Information Technology Equipment

EN 50082-1:1992 Immunity, Generic Requirements

-EN61000-4-2:1995 Electrostatic Discharge (ESD) Immunity
-EN61000-4-3:1995 Radiated RF Field Immunity
-EN61000-4-4:1995 EFT Immunity for AC and I/O Lines

EN 60950:1992 Safety of Information Technology Equipment

The technical documentation required to demonstrate this product meets the requirements of the EMC Directive and the Low Voltage Directive has been compiled by ICS Advent and is available for inspection by the relevant enforcement authorities.

Attention

The attention of the specifier, purchaser, installer, or user is drawn to special measures and limitations for use which must be observed when the product is taken into service to maintain compliance with the above directives. Details of these special measures and limitations are in the product manual.

A handwritten signature in black ink, appearing to read 'Jim Jameson'.

Mr. Jim Jameson
President & Chief Executive Officer



ICS Advent Europe
Ben Turner Industrial Road
Oving Road
Chichester, West Sussex
PO19 4ET, UK