

# User's Guide

<http://www.omega.com>  
e-mail: [info@omega.com](mailto:info@omega.com)

**PCI-DAS1200**

# Table of Contents

<b>1.0 INTRODUCTION</b>	1
<b>2.0 INSTALLATION</b>	3
<b>2.1 WINDOWS</b>	3
2.1.1 <i>Install the InstaCAL™ software</i>	3
2.1.2 <i>Launching InstaCAL™</i>	3
2.1.3 <i>Using InstaCal™</i>	4
2.1.4 <i>Testing the Installation</i>	4
<b>2.2 DOS</b>	5
2.2.1 <i>Install the InstaCal™ software</i>	5
2.2.2 <i>Launching InstaCAL™</i>	5
2.2.3 <i>Using InstaCal™</i>	5
2.2.4 <i>TESTING THE INSTALLATION</i>	6
<b>3.0 HARDWARE CONNECTIONS</b>	6
<b>3.1 CONNECTOR PIN DIAGRAM</b>	7
<b>3.2 CONNECTING SIGNALS TO THE PCI-DAS1200</b>	7
<b>4.0 ANALOG CONNECTIONS</b>	8
<b>4.1 ANALOG INPUTS</b>	8
4.1.1 <i>Single-Ended and Differential Inputs</i>	8
4.1.2 <i>System Grounds and Isolation</i>	10
<b>4.2 WIRING CONFIGURATIONS</b>	12
4.2.2 <i>Common Ground / Differential Inputs</i>	13
4.2.3 <i>Common Mode Voltage &lt; +/-10V / Single-Ended Inputs</i>	14
4.2.4 <i>Common Mode Voltage &lt; +/-10V / Differential Inputs</i>	14
4.2.6 <i>Isolated Grounds / Single-Ended Inputs</i>	15
4.2.7 <i>Isolated Grounds / Differential Inputs</i>	15
<b>5.0 PROGRAMMING &amp; APPLICATIONS</b>	16
<b>5.1 PROGRAMMING LANGUAGES</b>	16
<b>5.2 PACKAGED APPLICATIONS PROGRAMS</b>	16
<b>6.0 SELF-CALIBRATION OF THE PCI-DAS1200</b>	17
<b>6.1 CALIBRATION CONFIGURATION</b>	17
<b>7.0 PCI-DAS1200 REGISTER DESCRIPTION</b>	19
<b>7.1 REGISTER OVERVIEW</b>	19
<b>7.2 BADR0</b>	19
<b>7.3 BADR1</b>	19
7.3.1 <i>INTERRUPT / ADC FIFO REGISTER</i>	19
7.3.2 <i>ADC CHANNEL MUX AND CONTROL REGISTER</i>	21
7.3.3 <i>TRIGGER CONTROL/STATUS REGISTER</i>	23

# Table of Contents

7.3.4 CALIBRATION REGISTER .....	25
7.3.5 DAC CONTROL/STATUS REGISTER ( <i>Does not apply to PCI-DAS1200/JR</i> ) .....	26
<b>7.4 BADR2</b> .....	27
7.4.1 ADC DATA REGISTER .....	27
7.4.2 ADC FIFO CLEAR REGISTER .....	27
<b>7.5 BADR3</b> .....	28
7.5.1 ADC PACER CLOCK DATA AND CONTROL REGISTERS .....	28
7.5.2 DIGITAL I/O DATA AND CONTROL REGISTERS .....	29
<b>7.6 BADR4</b> .....	33
7.6.1 DAC0 DATA REGISTER .....	33
7.6.2 DAC1 DATA REGISTER .....	33
<b>8.0 ELECTRICAL SPECIFICATIONS</b> .....	34
ANALOG INPUT SECTION .....	34
ANALOG OUTPUT: .....	35
PARALLEL DIGITAL INPUT / OUTPUT .....	35
COUNTER SECTION .....	36
OTHER SPECIFICATIONS: .....	37

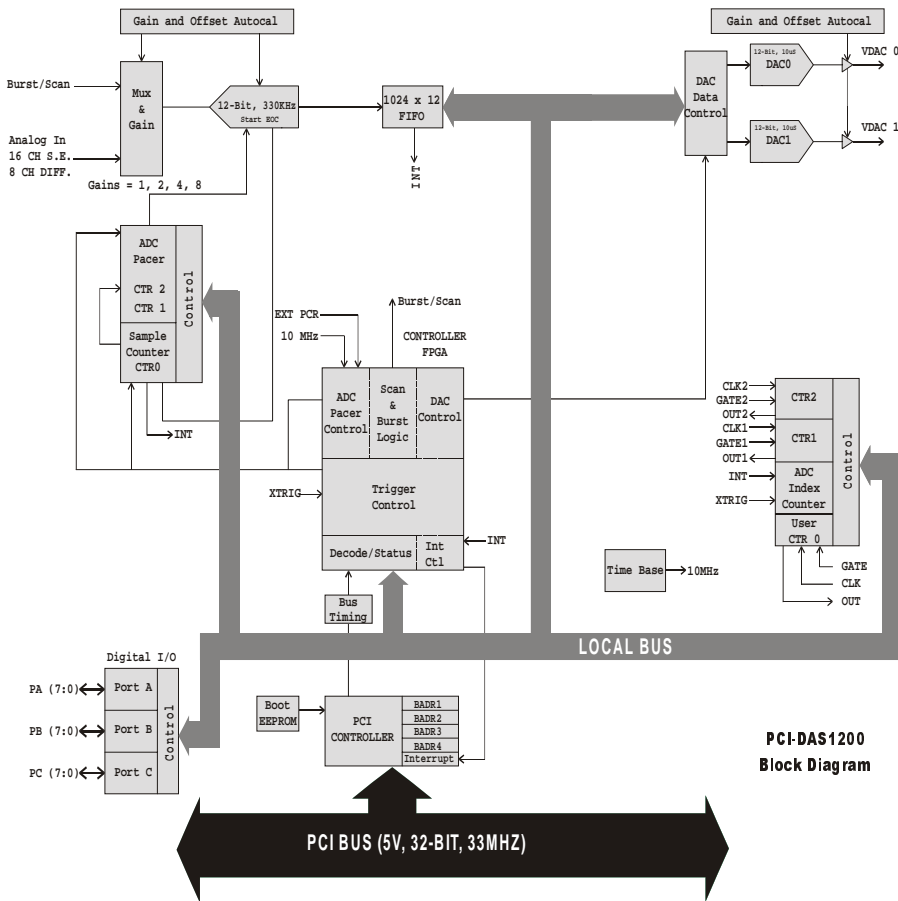
# 1.0 INTRODUCTION

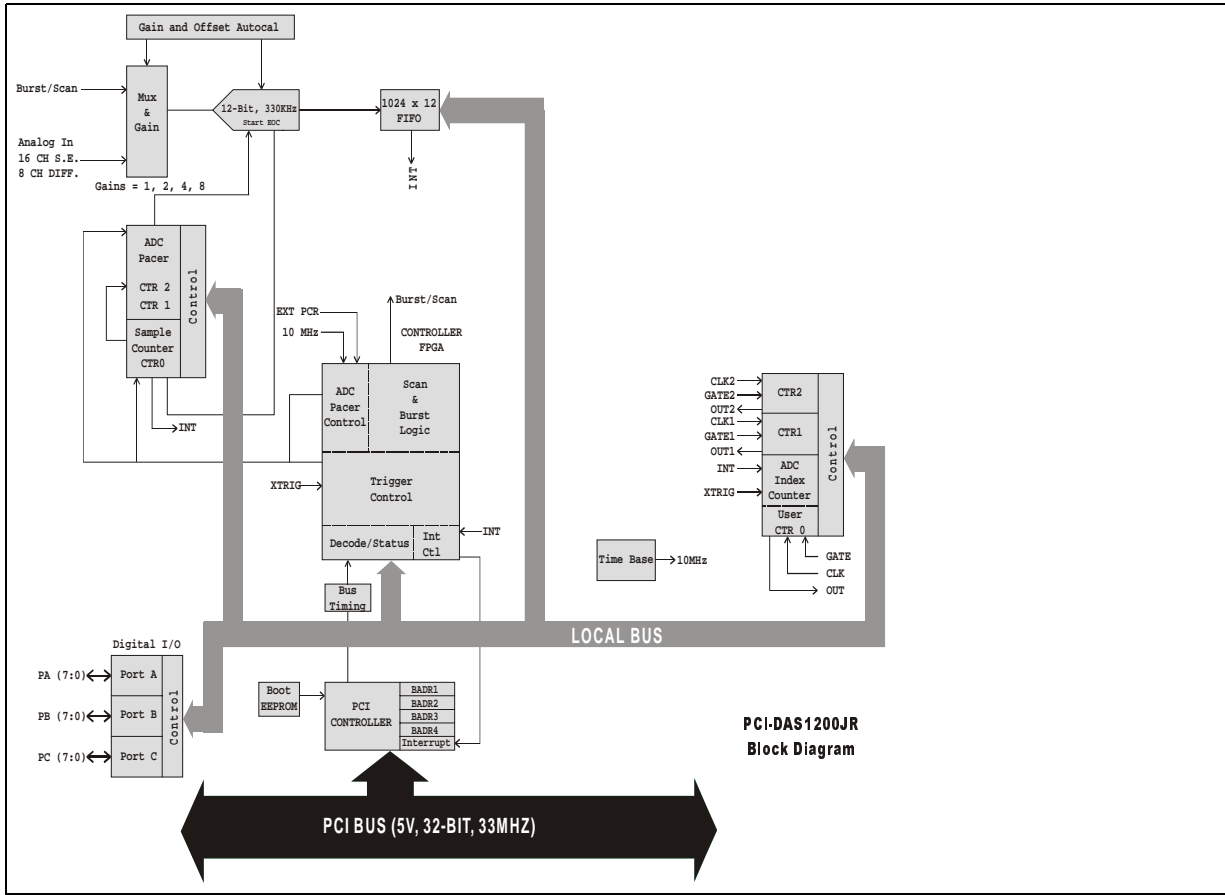
The PCI-DAS1200 and PCI-DAS1200/JR are multifunction measurement and control boards designed to operate in computers with PCI bus accessory slots. The difference between the boards is that the analog output functions are not supported by the /JR board. The architecture of the boards is loosely based on the original CIO-DAS16; the standard of ISA bus data acquisition. Much has changed though, and all of it due to improvements in technology. Surface mount packaging technology and custom ASICs allow a far greater range of control over programmable options, such as calibration, triggering, synchronization, and data transfer.

Even the connector has changed. New, denser connectors allow up to 100 signal lines where 37 was once the standard.

The PCI-DAS1200 and PCI-DAS1200/JR are completely plug-and-play. There are no switches or jumpers on the board. All board addresses are set by your computer's plug-and-play software.

These products are supported by our Universal Library programming library. As an owner, you are entitled to the latest revision of the manual and software. Just call with your current revision numbers handy, and request an update be sent to you.





## 2.0 INSTALLATION

---

The PCI-DAS1200 and PCI-DAS1200/JR boards are easy to use. This quick start procedure will help you setup, install, and test your board quickly and easily. We assume you already know how to open the PC and install expansion boards. If you are unfamiliar or uncomfortable with board installation, please refer to your computer's documentation.

---

### 2.1 WINDOWS 95, 98 & NT

PCI products are completely plug and play. Simply follow the steps shown below to install your PCI hardware.

1. Turn your computer off, unplug it, open it up, and insert the PCI board into any available PCI slot.
2. Close your computer up, plug it back in, and turn it on.
3. Windows 95, 98 will automatically detect the board as it starts up. If the board's installation file is already on the system, it will load without user interaction. If the installation file is not detected, you will be prompted to insert the disk containing it. The required file is on the *InstaCal* disk or CD you received with your board. The appropriate file will then be automatically loaded, and the PCI board will appear in the Device Manager under DAS Components.

#### 2.1.1 Install the *InstaCAL*<sup>TM</sup> software

*InstaCal* is a powerful installation, calibration, and self-test program included with all boards. It is very important that you load and run *InstaCal* prior to trying to use your board. To install *InstaCal*, insert the *InstaCal* disk or CD into an appropriate disk drive in your computer. From the main **Start** menu (located on the lower left of your screen) select **Run**. At this point type the drive letter where the *InstaCal* disk resides followed by **:\*setup***. For example if the *InstaCal* floppy disk is in your "a" drive, from the RUN screen enter **a:\*setup***. (omit the period.)

Alternatively, from the RUN screen, select browse, select the drive where the *InstaCal* disk or CD resides, then select **setup.exe** and click **OK**. The **Add/Remove Programs** utility found in the Control Panel can also be used. Be sure the *InstaCal* floppy disk or CD is in its drive and press **Install...**

The install wizard will now launch, and you will then be prompted for additional information. Follow the instructions and, if possible, accept the defaults. If this is your first installation, we urge you to accept the defaults. It will be easier for us to assist you in the unlikely event of trouble during your system setup and operation.

The installation routines will create all required folders/directories and unpack the various pieces of compressed software. The default location is on your main hard drive in a directory or folder named C:\CB\).

If you have purchased the Universal Library programmers library (highly recommended with the PCI-DAS1602 series), the installation program will install all the software required to run the PCI board as well as the Universal Library.

#### 2.1.2 Launching *InstaCal*

Prior to starting *InstaCal*, reboot your computer so the various changes made to your startup files are active.

Use the *InstaCal* shortcut if you opted to install it.

From Win95, use "Start: Run", type **inscal32** at the prompt, and click **OK**.

Alternatively, you may use Windows Explorer to find **inscal.exe**. It should be on your main hard drive in a directory called C:\CB. (if C:\ is your main hard drive). To launch *InstaCal*, simply double-click on the file **inscal.exe**.

### **2.1.3 Using InstaCal**

*InstaCal* is the Installation, Calibration and Test software supplied with all I/O boards.

If *InstaCal* finds that a new PCI board has been installed and is not in the configuration file (cb.cfg) a dialog will appear listing any PCI boards that *InstaCal* has detected in the system. Each board in the list may be checked or unchecked. Those boards in the list that are checked will be added to the configuration file and appear in the *InstaCal*'s board list view.

*InstaCal* help is available by pressing the F1 function key. Most of *InstaCal* is intuitively obvious, and for that reason there is no user's manual for *InstaCal*.

The base address and IRQ are allocated by the PC's PCI plug & play software, and may not be modified through *InstaCal*. If you have installed ISA bus boards in the past, you are familiar with the need to select a base address and interrupt level. On PCI systems this is not of concern to you. The operating system and installation software automatically selects the base address and IRQ, and ensures that it does not conflict with other installed boards.

Additional board options that are automatically set, may be configured by selecting the Board Configuration menu in *InstaCal*. This option will display the available board configuration parameters as well as the address and IRQ already assigned to the board. All board information is stored in the configuration file CB.CFG. This file is accessed by the Universal Library for programmers. Note also that the Universal Library is the I/O board interface for packaged applications such as Labtech Notebook and HP-VEE, therefore the *InstaCal* settings must be made in order for these and other applications to run.

The board's base address is also stored in the system software. Once *InstaCal* installation software is run, other programming methods such as direct IN and OUT statements can write and read the PCI board registers by reference to the base address and the offset from base address corresponding to the chart of registers located elsewhere in this manual.

But a word of warning is in order here. Direct writes to the addresses simply by reference to the base address of the PCI board I/O registers is not advised. Since the addresses assigned by the PCI plug & play software are not under your control, there is no way to guarantee that your program will run in any other computer. Not only that, when you install new systems or components in your computer, previous base address assignments may be changed, and any particular board may be moved. It is best to use a library such as Universal Library or a program such as HP-VEE to make measurements with your PCI board.

### **2.1.4 Testing the Installation**

After you have run the install program and set your base address with *InstaCal*, it is time to test the installation. The following section describes the *InstaCal* procedure to test that your board is properly installed.

With *InstaCal* running, choose the TEST item on the main menu.

- a. Select the board you just installed from *InstaCal*'s board list view.
- b. Select the option Test/Analog to perform a board loop back test. This test will exercise the various functions of the board by simply connecting one of the board's output signals to one of the board's input channels.
- c. Select an input channel and signal source to test.
- d. Complete the connections shown in the dialog box and verify that the indicated waveform is displayed in the plot window.
- e. The "I/O Test Menu" lists the option "Plot", select it and make the connections as shown to test your card.

---

---

## 2.2 DOS AND/OR WINDOWS 3.X

Most users are now installing boards on systems with Windows 95 or higher operating systems. However, if you wish to install your PCI board in a machine running Windows 3.1 and/or DOS your BIOS will automatically detect the board on boot-up. In this case please proceed to the following section titled “InstaCAL”

*Note: InstaCal and Universal Library support for the PCI-DAS1200/JR is not currently available in DOS or Windows 3.x. If you are going to be using the /JR board with DOS or Windows 3.1, call Technical Support for assistance.*

### 2.2.1 Install the InstaCal software

*InstaCal* installs the same way for Windows 3.x as it does for Windows 95. (i.e., setup.exe). See paragraph 2.1.1. If you are running DOS, call Technical support for special sequences.

### 2.2.2 Launching InstaCal

Prior to starting *InstaCal*, reboot your computer so the various changes made to your startup files are active.

From the DOS prompt you may start *InstaCAL* by simply typing: **instacal** and hitting enter.

From Windows 3.x, use the file manager to find **instacal.exe**. It should be on your main hard drive in a directory called C:\CB. (if C:\ is your main hard drive). To launch *InstaCAL*, simply double click on the file **Instacal.exe**. (You may also launch *InstaCal* via the FILE menu, select **RUN**, type **instacal** and click on **OK**.)

### 2.2.3 Using InstaCal

*InstaCal* is the Installation, Calibration and Test software supplied with all I/O boards. If you have a PCI board installed in a PCI slot in your computer, the first message *InstaCal* displays is:

```
INSTACAL: PCI Card Detection
Card PCI-DAS1200 found in slot # N
Do you wish to add this board to the configuration file?
```

A dialog box “Add PCI Card?” also appears. You should choose **yes**.

Two additional dialog boxes will open. One shows the boards currently installed in your configuration file, the second allows you to choose a board number to assign to the PCI board. If this is your first installation simply hit enter to accept the default of BOARD 0.

If you have other boards already installed, choose a board number not currently in use. *InstaCal* will do the rest of the initial installation of your PCI board selecting addresses and other system resource settings which are not your choice to select. In PCI systems, these settings are assigned by the plug & play system software.

*InstaCal* help is available by pressing the F1 function key. Most of *InstaCal* is intuitively obvious and for that reason there is no user's manual for *InstaCal*.

The base address is allocated by the PCI plug & play procedure and may not be modified. If you have installed ISA bus boards in the past you are familiar with the need to select a base address and interrupt level. On PCI systems this is not of concern to you. It is not up to you to select a base address and ensure that it does not conflict with an installed port. In PCI systems, the operating software and installation software do the selection and checking for you.

The program *InstaCal* selects and sets the base address from the range of available addresses. The address and other information is stored in the configuration file CB.CFG. This file is accessed by the Universal Library for programmers. Note also that the Universal Library is the I/O board interface for packaged applications such as Labtech Notebook and HP-VEE, therefore the *InstaCal* settings must be made in order for these and other applications to run.



The board's base address is also stored in the system software. Once *InstaCal* installation software is run, other programming methods such as direct IN and OUT statements can write and read the PCI board registers by reference to the base address and the offset from base address corresponding to the chart of registers located elsewhere in this manual.

But a word of warning is in order here. Direct writes to the addresses simply by reference to the base address of the PCI board I/O registers is not advised. Since the addresses assigned by the PCI plug & play software are not under your control, there is no way to guarantee that your program will run in any other computer. Not only that, if you install another PCI board in your computer after the original PCI board's addresses have been assigned, those addresses may be moved by the plug & play software when the second board is installed. It is best to use a library such as Universal Library or a program such as HP-VEE to make measurements with your PCI board.

#### **2.2.4 TESTING THE INSTALLATION**

After you have run the install program and set your base address with *InstaCal*, it is time to test the installation. The following section describes the *InstaCal* procedure to test that your board is properly installed.

With *InstaCal* running, choose the TEST item on the main menu.

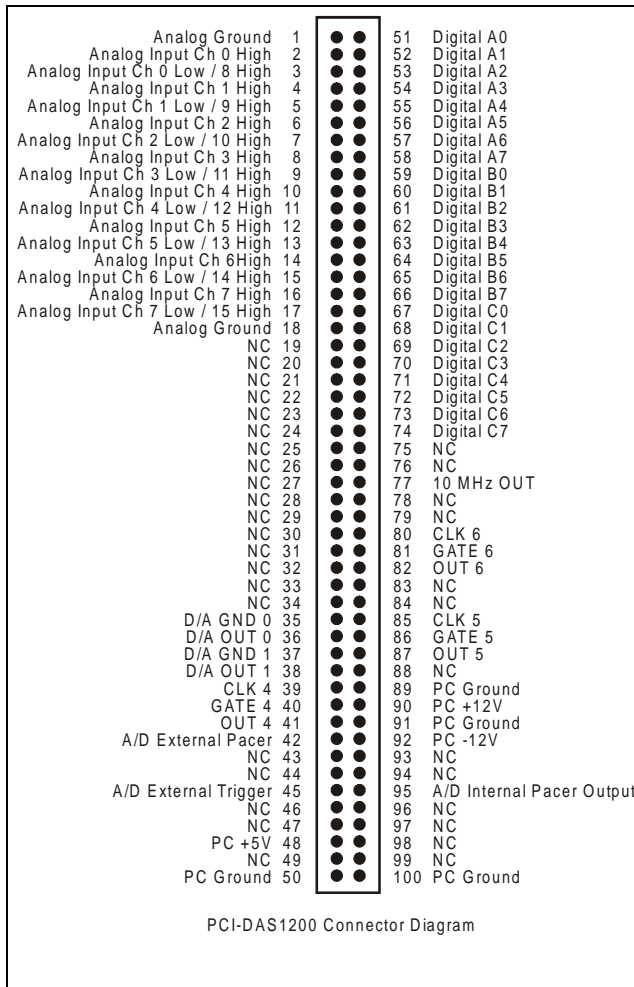
- a. Select the board you just installed
- b. If the choice "Internal Test" is available, then select Internal Test. If not, proceed to e. below.
- c. The internal control registers of the board will then be tested. If this test is successful, your board is installed correctly.
- d. If the Internal Test is completed successfully, you may want to check that the I/O pins are working correctly. To check this select External Test and follow the instruction provided. This will require you to use the shorting wires supplied with the board to connect inputs to outputs for I/O testing.
- e. The "I/O Test Menu" lists the option "Plot", select it and make the connections as shown to test your card.

## 3.0 HARDWARE CONNECTIONS

---

### 3.1 CONNECTOR PIN DIAGRAM

The PCI-DAS1200 and PCI-DAS1200/JR employ a 100-pin I/O connector. Please make accurate notes and pay careful attention to wire connections. In a large system, a misplaced wire may create hours of work ‘fixing’ problems that do not exist. Note that pins 35, 36, 37, and 38 are for analog outputs and are therefore NC (no connection) on the PCI-DAS1200/JR board. (Pin 77 is also NC on the /JR board).



### 3.2 CONNECTING SIGNALS TO THE PCI-DAS1200

The 100-pin connector provides a far greater signal density than the traditional 37-pin D type connector. In exchange for that density comes a far more complex cable and mating connector. The C100-FF-2 cable is a pair of 50-pin ribbon cables. At one end they are joined together with a 100-pin connector. From the 100-pin connector designed to mate with the PCI-DAS1200 connector, the two 50-pin ribbon cables diverge and are terminated at the other end with standard 50-pin header connectors. A CIO-MINI50 screw terminal board (or CIO-MINI50/DST with detachable screw terminals) is the ideal way to terminate real-world signals and route them into the PCI-DAS1200.

## 4.0 ANALOG CONNECTIONS

---

---

### 4.1 ANALOG INPUTS

Analog signal connection is one of the most challenging aspects of applying a data acquisition board. If you are an Analog Electrical Engineer, this section is not for you, but if you are like most PC data acquisition users, the best way to connect your analog inputs may not be obvious. Though complete coverage of this topic is well beyond the scope of this manual, the following section provides some explanations and helpful hints regarding these analog input connections. This section is designed to help you achieve the optimum performance from your PCI-DAS1200 series board.

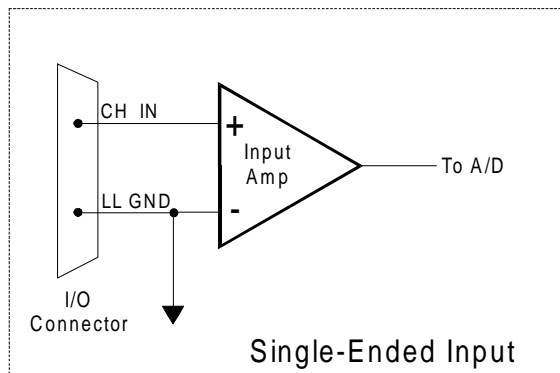
Prior to jumping into actual connection schemes, you should have at least a basic understanding of single-ended/differential inputs and system grounding/isolation. If you are already comfortable with these concepts, you may wish to skip to the next section (on wiring configurations).

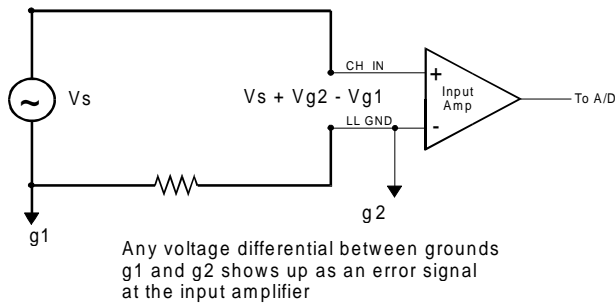
#### ***4.1.1 Single-Ended and Differential Inputs***

The PCI-DAS1200 provides either 8 differential or 16 single-ended input channels. The concepts of single-ended and differential inputs are discussed in the following section.

#### ***Single-Ended Inputs***

A single-ended input measures the voltage between the input signal and ground. In this case, in single-ended mode the PCI-DAS1200 measures the voltage between the input channel and low level ground (LLGND). The single-ended input configuration requires only one physical connection (wire) per channel and allows the PCI-DAS1200 to monitor more channels than the (2-wire) differential configuration using the same connector and onboard multiplexer. However, because the PCI-DAS1200 is measuring the input voltage relative to its own low level ground, single-ended inputs are more susceptible to both EMI (electromagnetic interference) and any ground noise at the signal source. The following diagrams show the single-ended input configuration.

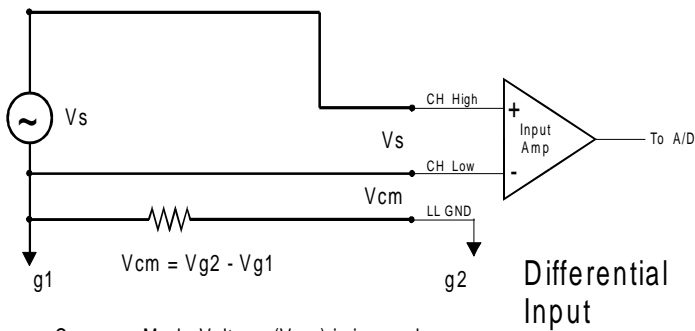
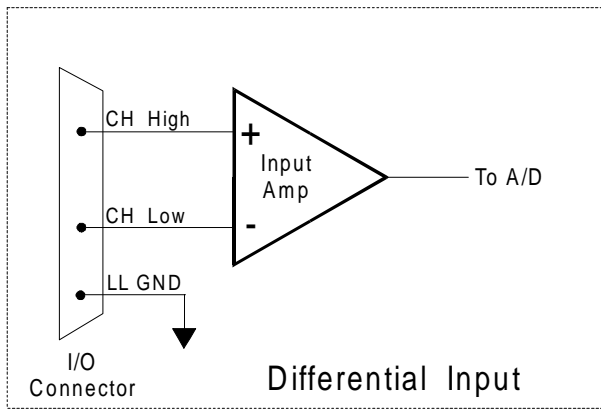




Single-ended input with Common Mode Voltage

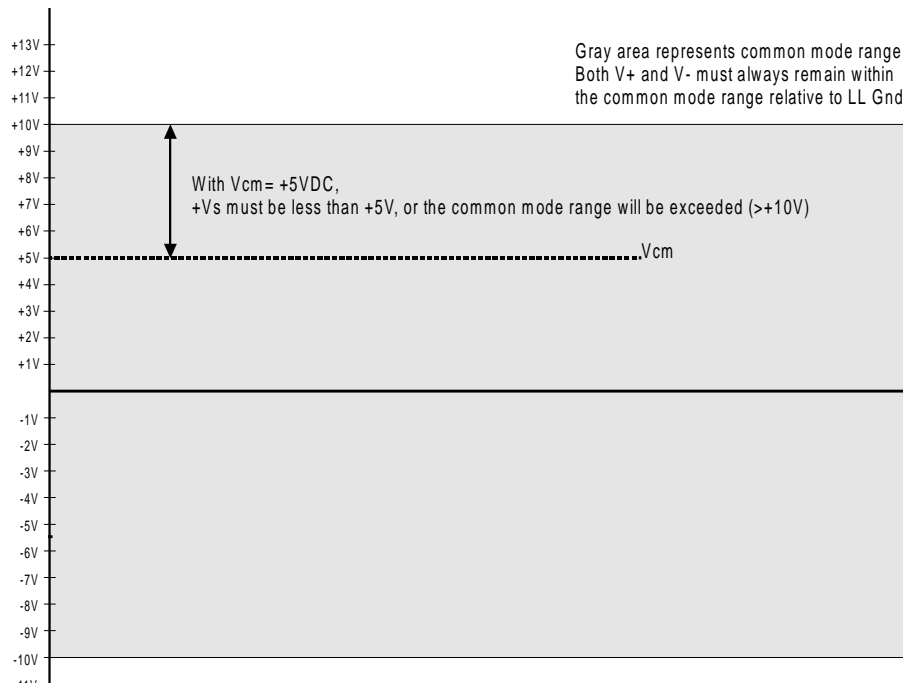
### Differential Inputs

Differential inputs measure the voltage between two distinct input signals. Within a certain range (referred to as the common mode range), the measurement is almost independent of signal source to PCI-DAS1200 ground variations. A differential input is also much more immune to EMI than a single-ended one. Most EMI noise induced in one lead is also induced in the other, the input measures only the difference between the two leads, and the EMI common to both is ignored. This effect is a major reason for twisted pair wire because the twisting ensures that both wires are subject to virtually identical external influence. The diagram below shows a typical differential input configuration.



Common Mode Voltage ( $V_{cm}$ ) is ignored by differential input configuration. However, note that  $V_{cm} + V_s$  must remain within the amplifier's common mode range of  $\pm 10V$

Before moving on to the discussion of grounding and isolation, it is important to explain the concepts of common mode, and common mode range (CM Range). Common mode voltage is depicted in the diagram above as  $V_{cm}$ . Though differential inputs measure the voltage between two signals, without (almost) respect to the either signal's voltages relative to ground, there is a limit to how far away from ground either signal can go. Though the PCI-DAS1200 has differential inputs, it will not measure the difference between 100V and 101V as 1 Volt (in fact the 100V would destroy the board!). This limitation or common mode range is depicted graphically in the following diagram. The PCI-DAS1200 common mode range is +/- 10 Volts. Even in differential mode, no input signal can be measured if it is more than 10V from the board's low level ground (LLGND).



#### 4.1.2 System Grounds and Isolation

There are three scenarios possible when connecting your signal source to your PCI-DAS1200 board.

1. The PCI-DAS1200 and the signal source may have the same (or **common**) ground. This signal source may be connected directly to the PCI-DAS1200.
2. The PCI-DAS1200 and the signal source may have an offset voltage between their grounds (ac and/or dc). This offset is commonly referred to as **common mode voltage**. Depending on the magnitude of this voltage, it may or may not be possible to connect the PCI-DAS1200 directly to your signal source. We will discuss this topic further in a later section.
3. The PCI-DAS1200 and the signal source may already have **isolated grounds**. This signal source may be connected directly to the PCI-DAS1200.

#### Which system do you have?

Try the following experiment. Using a battery powered voltmeter\*, measure the voltage (difference) between the ground signal at your signal source and at your PC. Place one voltmeter probe on the PC ground and the other on the signal source ground. Measure both the ac and dc Voltages.

*\*If you do not have access to a voltmeter, skip this experiment and take a look at the following three sections. You may be able to identify your system type from the descriptions provided.*

If both ac and dc readings are 0.00 volts, you may have a system with common grounds. However, since voltmeters will average out high frequency signals, there is no guarantee. Please refer to the section below titled **Common Grounds**.

If you measure reasonably stable ac and dc voltages, your system has an offset voltage between the grounds category. This offset is referred to as a Common Mode Voltage. Please to read the following warning carefully, then proceed to the section describing **Common Mode** systems.

#### WARNING

*If either the ac or dc voltage is greater than 10 volts, do not connect the PCI-DAS1200 to this signal source. You are beyond the board's usable common mode range and will need to either adjust your grounding system or add special isolation signal conditioning to take useful measurements. A ground offset voltage of more than 30 volts will likely damage the PCI-DAS1200 board and possibly your computer. Note that an offset voltage much greater than 30 volts will not only damage your electronics, but it may also be hazardous to you.*

*This is such an important point, that we will state it again. If the voltage between the ground of your signal source and your PC is greater than 10 volts, your board will not take useful measurements. If this voltage is greater than 30 volts, it will likely cause damage, and may represent a serious shock hazard! In this case you will need to either reconfigure your system to reduce the ground differentials, or purchase and install special electrical isolation signal conditioning.*

If you cannot obtain a reasonably stable dc voltage measurement between the grounds, or the voltage drifts around considerably, the two grounds are most likely isolated. The easiest way to check for isolation is to change your voltmeter to it's ohm scale and measure the resistance between the two grounds. It is recommended that you turn both systems off prior to taking this resistance measurement. If the measured resistance is more than 100 Kohm, it's a fairly safe bet that your system has electrically **isolated grounds**.

#### **Systems with Common Grounds**

In the simplest (but perhaps least likely) case, your signal source will have the same ground as the PCI-DAS1200. This would typically occur when providing power or excitation to your signal source directly from the PCI-DAS1200. There may be other common ground configurations, but it is important to note that any voltage between the PCI-DAS1200 ground and your signal ground is a potential error voltage if you set up your system based on a common ground assumption.

As a safe rule of thumb, if your signal source or sensor is not connected directly to an LLGND pin on your PCI-DAS1200, it's best to assume that you do not have a common ground even if your voltmeter measured 0.0 Volts. Configure your system as if there is ground offset voltage between the source and the PCI-DAS1200. This is especially true if you are using high gains, since ground potentials in the sub millivolt range will be large enough to cause A/D errors, yet will not likely be measured by your handheld voltmeter.

#### **Systems with Common Mode (ground offset) Voltages**

The most frequently encountered grounding scenario involves grounds that are somehow connected, but have ac and/or dc offset voltages between the PCI-DAS1200 and signal source grounds. This offset voltage may be ac, dc, or both and may be caused by a wide array of phenomena including EMI pickup, resistive voltage drops in ground wiring and connections, etc. Ground offset voltage is a more appropriate term to describe this type of system, but since our goal is to keep things simple, and help you make appropriate connections, we'll stick with our somewhat loose usage of the phrase Common Mode.

### Small Common Mode Voltages

If the voltage between the signal source ground and PCI-DAS1200 ground is small, the combination of the ground voltage and input signal will not exceed the PCI-DAS1200's +/-10V common mode range, (*i.e., the voltage between grounds, added to the maximum input voltage, stays within +/-10V*). This input is compatible with the PCI-DAS1200 and the system may be connected without additional signal conditioning. Fortunately, most systems will fall in this category and have a small voltage differential between grounds.

### Large Common Mode Voltages

If the ground differential is large enough, the PCI-DAS1200's +/- 10V common mode range will be exceeded (*i.e. the voltage between PCI-DAS1200 and signal source grounds, added to the maximum input voltage you're trying to measure exceeds +/-10V*). In this case the PCI-DAS1200 cannot be directly connected to the signal source. You will need to change your system grounding configuration or add isolation signal conditioning. (Please look at our ISO-RACK and ISO-5B-series products to add electrical isolation, or give our technical support group a call to discuss other options.)

#### NOTE

*Relying on the earth prong of a 120 Vac for signal ground connections is not advised.. Different ground plugs may have large and potentially even dangerous voltage differentials. Remember that the ground pins on 120 Vac outlets on different sides of the room may only be connected in the basement. This leaves the possibility that the "ground" pins may have a significant voltage differential (especially if the two 120 Vac outlets happen to be on different phases!)*

### PCI-DAS1200 and signal source already have isolated grounds

Some signal sources will already be electrically isolated from the PCI-DAS1200. The diagram below shows a typical isolated ground system. These signal sources are often battery powered, or are fairly expensive pieces of equipment (since isolation is not an inexpensive proposition), isolated ground systems provide excellent performance, but require some extra effort during connections to ensure optimum performance is obtained. Please refer to the following sections for further details.

---

---

## 4.2 WIRING CONFIGURATIONS

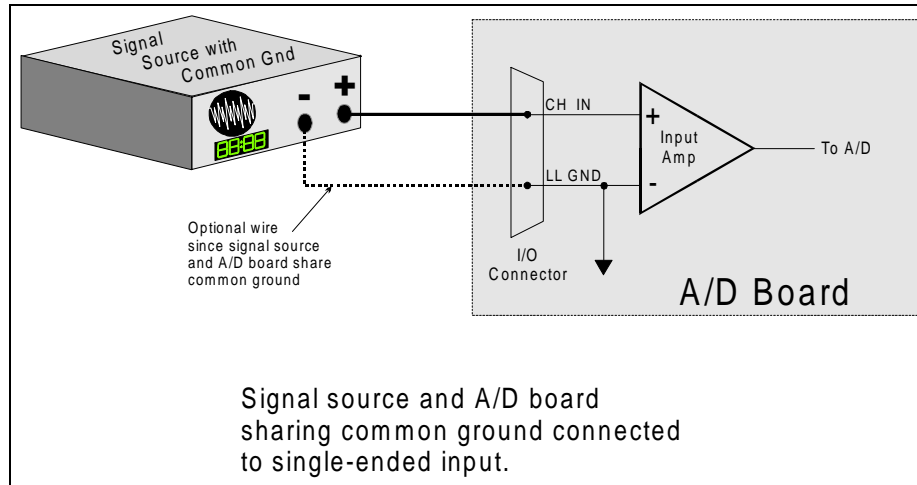
Combining all the grounding and input type possibilities provides us with the following potential connection configurations. The combinations along with our recommendations on usage are shown in the chart below.

Ground Category Our view		Input Configuration
Common Ground	Single-Ended Inputs	Recommended
Common Ground	Differential Inputs	Acceptable
Common Mode Voltage < +/-10V	Single-Ended Inputs	Not Recommended
Common Mode Voltage < +/-10V	Differential Inputs	Recommended
Common Mode Voltage > +/- 10V	Single-Ended Inputs	Unacceptable without adding Isolation
Common Mode Voltage > +/-10V	Differential Inputs	Unacceptable without adding Isolation
Already Isolated Grounds	Single-ended Inputs	Acceptable
Already Isolated Grounds	Differential Inputs	Recommended

The following sections depict recommended input wiring schemes for each of the 8 possible input configuration/grounding combinations.

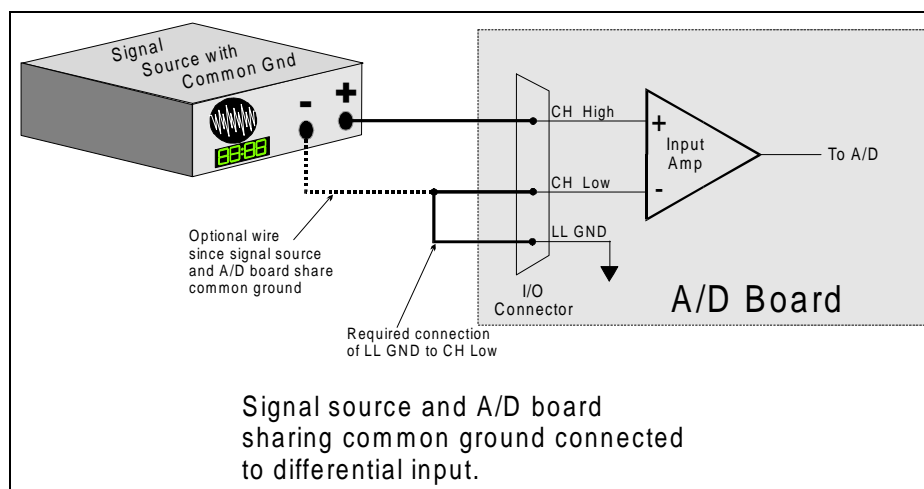
#### 4.2.1 Common Ground / Single-Ended Inputs

Single-ended is the recommended configuration for common ground connections. However, if some of your inputs are common ground and some are not, we recommend you use the differential mode. There is no performance penalty (other than loss of channels) for using a differential input to measure a common ground signal source. However, the reverse is not true. The diagram below shows a recommended connection diagram for a common ground / single-ended input system



#### 4.2.2 Common Ground / Differential Inputs

The use of differential inputs to monitor a signal source with a common ground is an acceptable configuration, though it requires more wiring and offers fewer channels than selecting a single-ended configuration. The diagram below shows the recommended connections in this configuration.



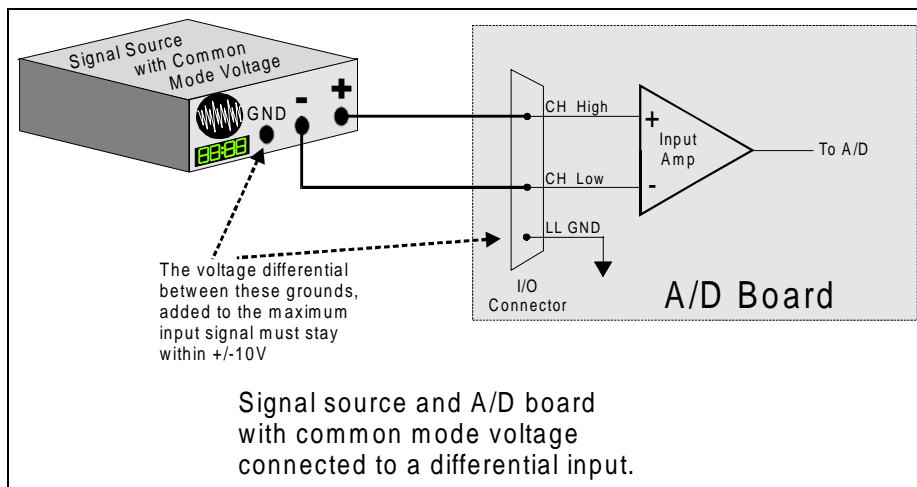


#### 4.2.3 Common Mode Voltage $< \pm 10V$ / Single-Ended Inputs

This is not a recommended configuration. In fact, the phrase “common mode” has no meaning in a single-ended system, and this case would be better described as a system with offset grounds. Anyway, you are welcome to try this configuration, no system damage should occur, and, depending on the overall accuracy you require, you may receive acceptable results.

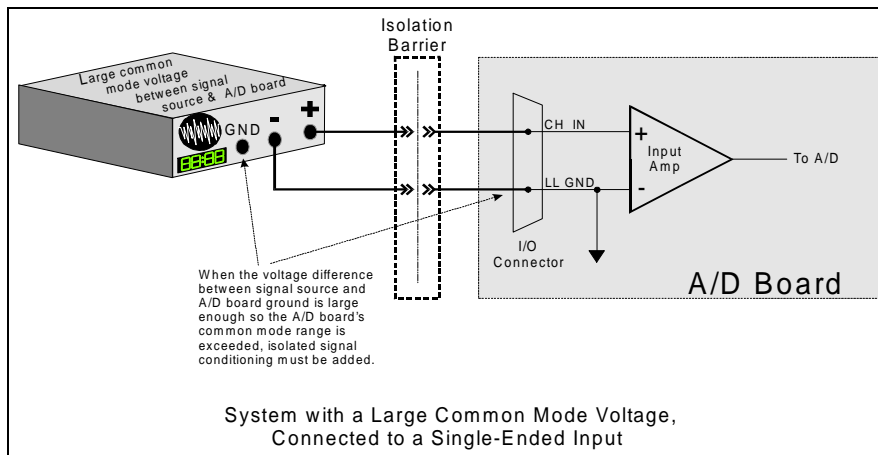
#### 4.2.4 Common Mode Voltage $< \pm 10V$ / Differential Inputs

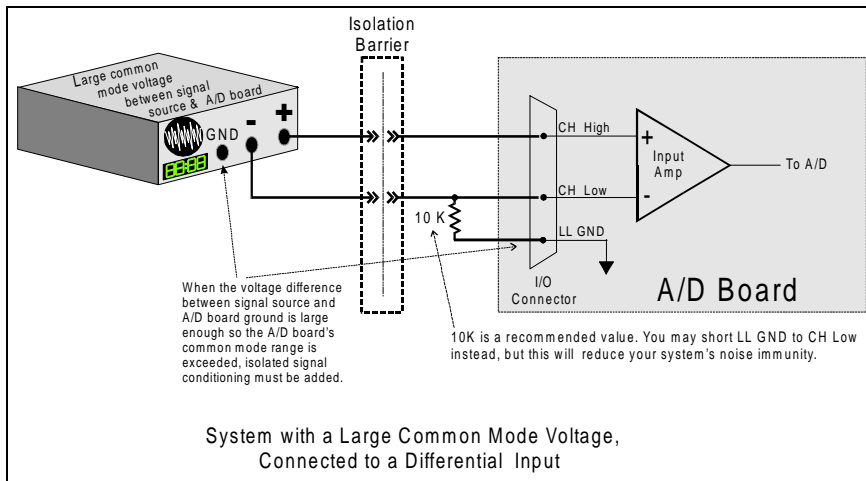
Systems with varying ground potentials should always be monitored in the differential mode. Use care to ensure that the sum of the input signal and the ground differential (referred to as the common mode voltage) does not exceed the common mode range of the A/D board ( $\pm 10V$  on the PCI-DAS1200). The diagram below shows recommended connections in this configuration.



#### 4.2.5 Common Mode Voltage $> \pm 10V$

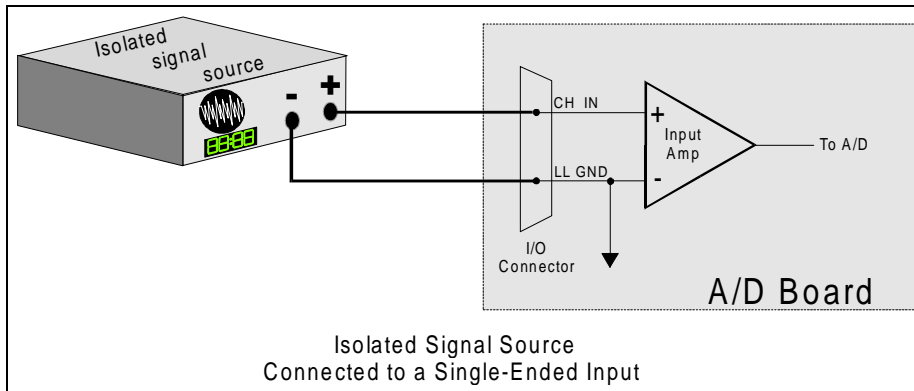
The PCI-DAS1200 will not directly monitor signals with common mode voltages greater than  $\pm 10V$ . You will need to either alter the system ground configuration to reduce the overall common mode voltage, or add isolated signal conditioning between the source and your board.





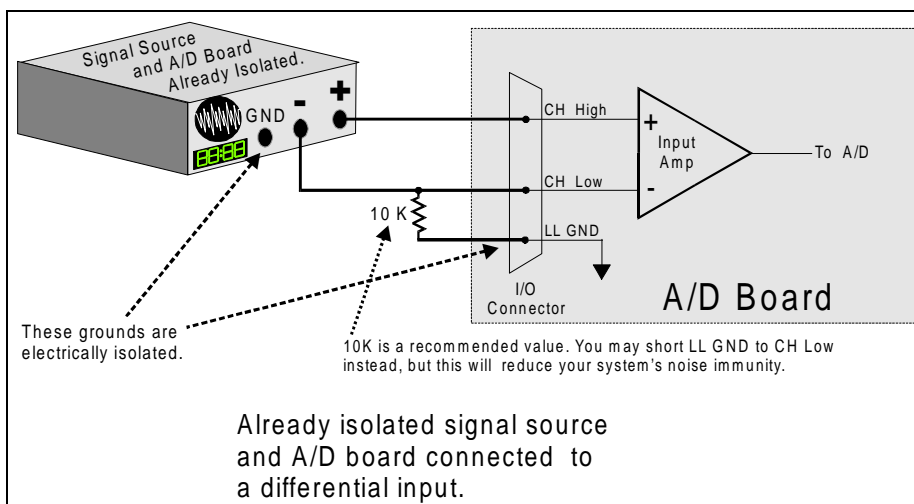
#### 4.2.6 Isolated Grounds / Single-Ended Inputs

Single-ended inputs can be used to monitor isolated inputs, though the use of the differential mode will increase your system's noise immunity. The diagram below shows the recommended connections in this configuration.



#### 4.2.7 Isolated Grounds / Differential Inputs

Optimum performance with isolated signal sources is ensured with the use of the differential input setting. The diagram below shows the recommended connections in this configuration.



## 5.0 Programming & Applications

---

---

Your PCI-DAS1200 is supported by the powerful Universal Library. We strongly recommend that you take advantage of the Universal Library as your software interface. The complexity of the registers required for automatic calibration combined with the Windows 9X and NT's dynamic allocation of addresses and internal resources makes the PCI-DAS1200 series very challenging to program via direct register I/O operations. Direct I/O programming should typically be attempted only by very experienced programmers.

Although the PCI-DAS1200 is part of the larger DAS family, there is no correspondence between register locations of the PCI-DAS1200 and boards in the CIO-DAS16 family. Software written at the register level for the other DAS boards will not work with the PCI-DAS1200.

---

---

### 5.1 PROGRAMMING LANGUAGES

The Universal Library provides complete access to the PCI-DAS1200 functions from a range of programming languages; both DOS and Windows. If you are planning to write programs, or would like to run the example programs for Visual Basic or any other language, please turn now to the Universal Library manual.

The optional VIX Components package may greatly simplify your programming effort. VIX Components is a set of programming tools based on a DLL interface to Windows languages. A set of VBX, OCX, or ActiveX interfaces allows point and click construction of graphical displays, analysis and control structures. Please see the catalog for a complete description of the package.

---

---

### 5.2 PACKAGED APPLICATIONS PROGRAMS

Many packaged application programs, such as DAS Wizard and HP-VEE now have drivers for the PCI-DAS1200. If the package you own does not appear to have drivers for the PCI-DAS1200 please fax or e-mail the package name and the revision number from the install disks. We will research the package for you and advise how to obtain PCI-DAS1200 drivers.

Some application drivers are included with the Universal Library package, but not with the Application package. If you have purchased an application package directly from the software vendor, you may need to purchase our Universal Library and drivers. Please contact us for more information on this topic.

## 6.0 Self-Calibration of the PCI-DAS1200

The PCI-DAS1200 is shipped fully-calibrated from the factory with cal coefficients stored in nvRAM. When using Universal Library at run time, these calibration factors are loaded into system memory and are automatically retrieved each time a different DAC/ADC range is specified. The user has the option to recalibrate with respect to the factory-measured voltage standards at any time by simply selecting the "Calibrate" option in InstaCal. Full calibration typically requires less than two minutes and requires no user intervention.

---

---

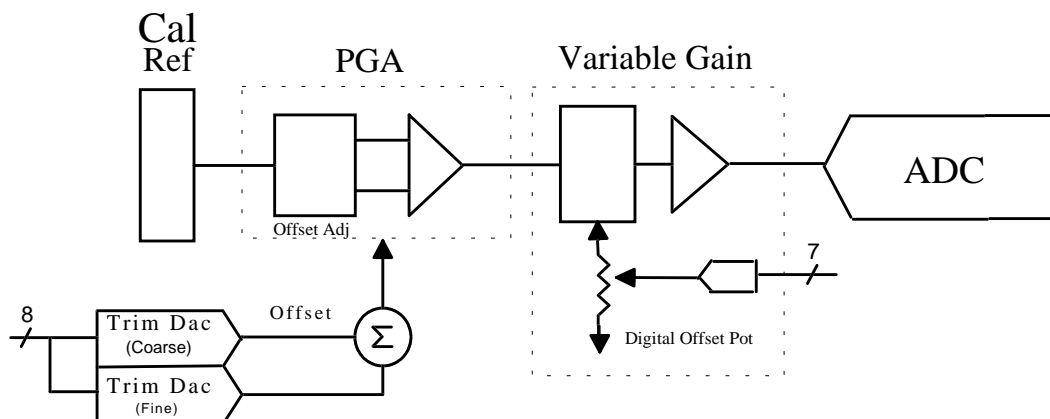
### 6.1 CALIBRATION CONFIGURATION

The PCI-DAS1200 provides self-calibration of the analog source and measurement systems thereby eliminating the need for external equipment and user adjustments. All adjustments are made via 8-bit calibration DACs or 7-bit digital potentiometers referenced to an on-board factory calibrated standard. Calibration factors are stored on the serial nvRAM..

A variety of methods are used to calibrate the different elements on the board. The analog front-end has several *knobs* to turn. Offset calibration is performed in the instrumentation amplifier gain stage. Front-end gain adjustment is performed via a variable attenuator/gain stage.

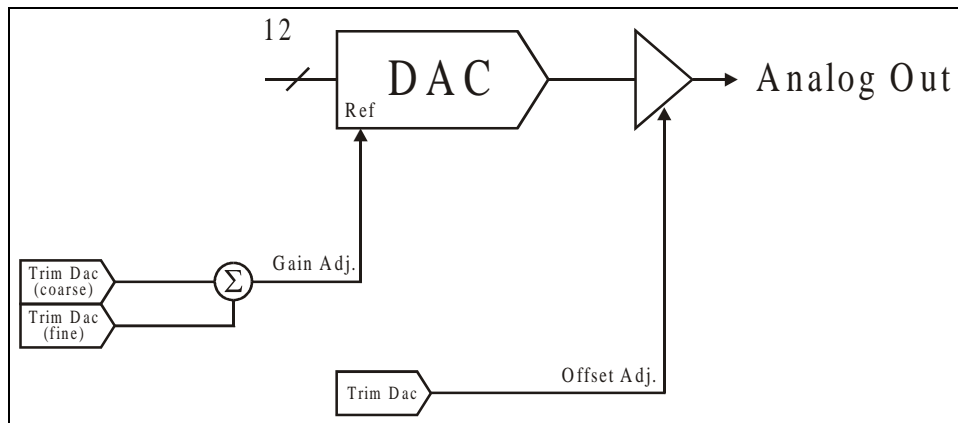
The analog output circuits are calibrated for both gain and offset. Offset adjustments for the analog output are made in the output buffer section. The tuning range of this adjustment allows for max DAC and output buffer offsets. Gain calibration of the analog outputs are performed via DAC reference adjustments.

*Figure 1* below is a block diagram of the analog front-end calibration system:



*Figure 1*

The calibration scheme for the Analog Out (not applicable on the PCI-DAS1200/JR) section is shown in *Figure 2* below. This circuit is duplicated for both DAC0 and DAC1



*Figure 2*

## 7.0 PCI-DAS1200 Register Description

---

---

---

---

### 7.1 REGISTER OVERVIEW

PCI-DAS1200 operation registers are mapped into I/O address space. Unlike ISA bus designs, this board has *several* base addresses, each corresponding to a reserved block of addresses in I/O space. As we mention in our programming chapter, we highly recommend customers use the Universal Library package. Direct register level programming should be attempted only by extremely experienced register level programmers.

Of six Base Address Regions (BADR) available in the PCI 2.1 specification, five are implemented in this design and are summarized as follows:

I/O Region	Function	Operations
BADR0	PCI Controller Operation Registers	32-Bit DWORD
BADR1	General Control/Status Registers	16-Bit WORD
BADR2	ADC Data, FIFO Clear Registers	16-Bit WORD
BADR3	Pacer, Counter/Timer and DIO Registers	8-Bit BYTE
BADR4	DAC Data Registers (not applicable for (JR))	16-Bit WORD

BADR<sub>n</sub> will likely be different on different machines. Assigned by the PCI BIOS, these Base Address values cannot be guaranteed to be the same even on subsequent power-on cycles of the same machine. All software must interrogate BADR0 at run-time with a READ\_CONFIGURATION\_WORD instruction to determine the BADR<sub>n</sub> values. Please see the "1997 AMCC S5933 PCI Controller Data Book" for more information.

---

---

### 7.2 BADR0

BADR0 is reserved for the AMCC S5933 PCI Controller operations. There is no reason to access this region of I/O space for most PCI-DAS1200 users. The installation procedures and Universal Library access all required information in this area. Unless you are writing direct register level software for the PCI-DAS1200, you will not need to be concerned with BADR0 address.

---

---

### 7.3 BADR1

The I/O region defined by BADR1 contains 5 control and status registers for ADC, DAC, interrupt and Autocal operations. This region supports 16-bit WORD operations.

#### 7.3.1 INTERRUPT / ADC FIFO REGISTER

BADR1+ 0: Interrupt Control, ADC status. A read/write register.

WRITE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	ADFLCL	-	-	-	-	-	INTCL	EOACL	-	EOAIE	-	INTE	INT1	INT0

Write operations to this register allow the user to select interrupt sources, enable interrupts, and clear interrupts as well as ADC FIFO flags. The following is a description of the Interrupt/ADC FIFO Register:

**INT[1:0]** General Interrupt Source selection bits.

INT1	INT0	Source
0	0	Not Defined
0	1	End of Channel Scan
1	0	AD FIFO Half Full
1	1	AD FIFO Not Empty

**INTE** Enables interrupt source selected via the INT[1:0] bits.

- 1 = Selected interrupt Enabled
- 0 = Selected interrupt Disabled

**EOAIE** Enables End-of-Acquisition interrupt. Used during FIFO'd ADC operations to indicate that the desired sample size has been gathered.

- 1 = Enable EOA interrupt
- 0 = Disable EOA interrupt

**EOACL** A write-clear to reset EOA interrupt status.

- 1 = Clear EOA interrupt.
- 0 = No effect.

**INTCL** A write-clear to reset **INT[1:0]** selected interrupt status.

- 1 = Clear **INT[1:0]** interrupt
- 0 = No effect.

**ADFLCL** A write-clear to reset latched ADC FIFO Full status.

- 1 = Clear ADC FIFO Full latch.
- 0 = No Effect.

*NOTE:* It is not necessary to reset any write-clear bits after they are set.

READ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	LADFUL	ADNE	ADNEI	ADHFI	EOBI	-	INT	EOAI	-	-	-	-	-	-

Read operations to this register allow the user to check status of the selected interrupts and ADC FIFO flags. The following is a description of Interrupt / ADC FIFO Register Read bits:

**EOAI** Status bit of ADC FIFO End-of-Acquisition interrupt

- 1 = Indicates an EOA interrupt has been latched.
- 0 = Indicates an EOA interrupt has not occurred.

**INT** Status bit of General interrupt selected via **INT[1:0]** bits. This bit indicates that *any* one of these interrupts has occurred.

- 1 = Indicates a General interrupt has been latched.
- 0 = Indicates a General interrupt has not occurred.

- EOBI** Status bit ADC End-of-Burst interrupt. Only valid for ADC Burst Mode enabled.  
 1 = Indicates an EOB interrupt has been latched.  
 0 = Indicates an EOB interrupt has not occurred.
- ADHFI** Status bit of ADC FIFO Half-Full interrupt. Used during REP INSW operations.  
 1 = Indicates an ADC Half-Full interrupt has been latched. FIFO has been filled with more than 255 samples.  
 0 = Indicates an ADC Half-Full interrupt has not occurred. FIFO has not yet exceeded 1/2 of its total capacity.
- ADNEI** Status bit of ADC FIFO Not-Empty interrupt. Used to indicate ADC conversion complete in single conversion applications.  
 1 = Indicates an ADC FIFO Not-Empty interrupt has been latched and that one data word may be read from the FIFO.  
 0 = Indicates an ADC FIFO Not-Empty interrupt has not occurred. FIFO has been cleared, read until empty or ADC conversion still in progress.
- ADNE** Real-time status bit of ADC FIFO Not-Empty status signal.  
 1 = Indicates ADC FIFO has at least one word to be read.  
 0 = Indicates ADC FIFO is empty.
- LADFUL** Status bit of ADC FIFO FULL status. This bit is latched.  
 1 = Indicates the ADC FIFO has *exceeded* full state. Data may have been lost.  
 0 = Indicates non-overflow condition of ADC FIFO.

### 7.3.2 ADC CHANNEL MUX AND CONTROL REGISTER

BADR1 + 2

This register sets channel mux HI/LO limits, ADC gain, offset and pacer source.  
 A Read/Write register.

WRITE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	ADPS1	ADPS0	UNIBIP	SEDIFF	GS1	GS0	CHH8	CHH4	CHH2	CHH1	CHL8	CHL4	CHL2	CHL1

#### CHL8-CHL1, CHH8-CHH1

When these bits are written, the analog input multiplexors are set to the channel specified by CHL8-CHL1. After each conversion, the input multiplexors increment to the next channel, reloading to the "CHL" start channel after the "CHH" stop channel is reached. LO and HI channels are the decode of the 4-bit binary patterns.

**GS[1:0]** These bits determine the ADC range as indicated below.

GS1	GS0	Range
0	0	10V
0	1	5V
1	0	2.5V
1	1	1.25V

**SEDIFF** Selects measurement configuration for the Analog Front-End.  
 1 = Analog Front-End in Single-Ended Mode. This mode supports



up to 16 channels.  
 0 = Analog Front-End in Differential Mode. This mode supports up to 8 channels.

**UNIBIP**

Selects offset configuration for the Analog Front-End.  
 1 = Analog Front-End Unipolar for selected range  
 0 = Analog Front-End Bipolar for selected range.

The following table summarizes all possible Offset/Range configurations:

UNIBIP	GS1	GS0	Input Range	Input Gain	Measurement Resolution
0	0	0	±10 V	1	4.88 mV
0	0	1	± 5 V	2	2.44 mV
0	1	0	±2.5 V	4	1.22 mV
0	1	1	±1.25V	8	610 µV
1	0	0	0-10V	1	2.44 mV
1	0	1	0-5V	2	1.22 mV
1	1	0	0-2.5V	4	610 µV
1	1	1	0-1.25V	8	305 µV

**ADPS[1:0]**

These bits select the ADC Pacer Source. Maximum Internal/External Pacer frequency is 330 kHz.

ADPS1	ADPS0	Pacer Source
0	0	SW Convert
0	1	82C54 Counter/Timer
1	0	External Falling
1	1	External Rising

*Note:* For ADPS[1:0] = 00 case, SW conversions are initiated via a word write to BADR2 + 0. Data is 'don't care.'

**READ**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	EOC	-	-	-	-	-	-	-	-	-	-	-	-	-	-

**EOC**

Real-time, non-latched status of ADC End-of-Conversion signal.  
 1 = ADC DONE  
 0 = ADC BUSY

**7.3.3 TRIGGER CONTROL/STATUS REGISTER**

## BADR1 + 4

This register provides control bits for all ADC trigger modes. A Read/Write register.

### WRITE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	COSRC	FFM0	ARM	-	-	-	XTRCL	PRTRG	BURSTE	TGEN	-	-	TS1	TS0

**TS[1:0]** These bits select one-of-two possible ADC Trigger Sources:

TS1	TS0	Source
0	0	Disabled
0	1	SW Trigger
1	0	External (Digital)
1	1	Not Defined

*Note:* TS[1:0] should be set to 0 while setting up Pacer source and count values.

### TGEN

This bit is used to enable External Trigger function

1 = External rising-edge Digital Trigger enabled.

0 = External Digital Trigger has no effect.

Note that the external trigger requires proper setting of the **TS[1:0]** and **TGEN** bits. Once these bits are set, the next rising edge will start a Paced ADC conversion. Subsequent triggers will have no effect until external trigger flop is cleared (**XTRCL**).

### BURSTE

This bit enables 330 kHz ADC Burst mode. Start/Stop channels are selected via the CHLx, CHHx bits in ADC CTRL/STAT register at BADR1 + 2.

1 = Burst Mode enabled

0 = Burst Mode disabled

### PRTRG

This bit enables ADC Pre-trigger Mode. This bit works with the ARM and FFM0 bits when using Pre-trigger mode.

1 = Enable Pre-trigger Mode

0 = Disable Pre-trigger Mode

### XTRCL

A write-clear to reset the **XTRIG** flip-flop.

1 = Clear **XTRIG** status.

0 = No Effect.

**ARM,  
FFM0**

These bits work in conjunction with **PRTRG** during FIFO'd ADC operations. Direct register level programming is beyond the scope of this manual, and should be attempted only by extremely experienced register level programmers. Call Technical Support for further information.

The table below provides a summary of bit settings and operation.

<b>PRTRG</b>	<b>FFM0</b>	<b>ARM is set...</b>	<b>FIFO Mode</b>	<b>Sample CTR Starts on...</b>
0	0	Via SW when remaining count <1024 ----- Via SW immediately	# Samples >1 FIFO Normal Mode ----- 1/2 FIFO < # Samples < 1 FIFO Normal Mode	ADHF
0	1	Via SW immediately	# Samples <1/2 FIFO Normal Mode	ADC Pacer
1	0	Via SW when remaining count <1024 ----- Via SW immediately	# Samples >1 FIFO Pre-Trigger Mode ----- 1/2 FIFO < # Samples < 1 FIFO Pre-Trigger Mode	ADHF
1	1	Via SW immediately	# Samples <1/2 FIFO, Pre-Trigger Mode	XTRIG

**COSRC**

This bit allows the user to select the clock source for user Counter 0.  
1 = Internal 10MHz oscillator  
0 = External clock source input via *CTROCLK* pin on 100p connector.

**READ**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	INDX_GT	-	-	-	-	XTRIG	-	-	-	-	-	-	-

**XTRIG**

1 = External Trigger flip-flop has been set. This bit is write-cleared.  
0 = External Trigger flip-flop reset. No trigger has been received.

**INDX\_GT**

1 = Pre-trigger index counter has completed its count.  
0 = Pre-trigger index counter has not been gated on or has not yet completed its count.

### 7.3.4 CALIBRATION REGISTER

As mentioned before, direct register level programming should be attempted only by extremely experienced register level programmers. This is true for register-level calibration. If you're not sure, don't attempt it. Call Technical Support for more information.

BADR1 + 6

This register controls all autocal operations. This is a Write-only register.

WRITE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDI	CALEN	CSRC2	CSRC1	CSRC0	-	SEL7376	SEL8800	-	-	-	-	-	-	-	-

**SEL8800** This bit enables the 8-bit trim DACs for the following circuits:

DAC Channel	Cal Function
0	DAC0 Fine Gain
1	DAC0 Coarse Gain
2	DAC0 Offset
3	DAC1 Offset
4	DAC1 Fine Gain
5	DAC1 Coarse Gain
6	ADC Coarse Offset
7	ADC Fine Offset

**SEL7376** This bit latches the 7-bit serial data stream into the AD7376 digital potentiometer (10KOhm). The AD7376 is used for analog front-end gain calibration.

**CSRC[2:0]** These bits select the different calibration sources available to the ADC front end.

CSRC2	CSRC1	CSRC0	Cal Source
0	0	0	AGND
0	0	1	7.0V
0	1	0	3.5V
0	1	1	1.75V
1	0	0	0.875V
1	0	1	8.6mV
1	1	0	VDAC0
1	1	1	VDAC1

**CALEN** This bit is used to enable Cal Mode.  
 1 = Selected Cal Source, **CSRC[2:0]**, is fed into Analog Channel 0.  
 0 = Analog Channel 0 functions as normal input.

**SDI** Serial Data In. This bit is used to set serial address/data stream for the DAC8800 TrimDac and 7376 digital potentiometer. Used in conjunction with **SEL8800** and **SEL7376** bits.

### 7.3.5 DAC CONTROL/STATUS REGISTER (Does not apply to PCI-DAS1200/JR)

BADR1 + 8

This register selects the DAC gain/range and update modes. This is a Write-only register.

WRITE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	DAC1R1	DAC1R0	DAC0R1	DAC0R0	MODE	-	-	-	-	-	DACEN	-

**DACEN** This bit enables the Analog Out features of the board.  
 1 = DAC0/1 enabled.  
 0 = DAC0/1 disabled.

The power-on state of this bit is 0.

**MODE** This bit determines the analog output mode of operation.  
 1 = Both DAC0 and DAC1 updated with data written to DAC0 data register.  
 0 = DACn updated with data written to DACn data register.

The power-on state of this bit is 0.

**DACnR[1:0]** These bits select the independent gains/ranges for either DAC0 or DAC1.  
 n=0 for DAC0 and n=1 for DAC1.

DACnR1	DACnR0	Range	LSB Size
0	0	Bipolar 5V	2.44mV
0	1	Bipolar 10V	4.88mV
1	0	Unipolar 5V	610uV
1	1	Unipolar 10V	1.22mV

---

---

## 7.4 BADR2

The I/O Region defined by BADR2 contains the ADC Data register and the ADC FIFO clear register.

### 7.4.1 ADC DATA REGISTER

BADR2 + 0

ADC Data register.

WRITE

Writing to this register is only valid for SW initiated conversions. The ADC Pacer source must be set to 00 via the ADPS[1:0] bits. A null write to BADR2 + 0 will begin a single conversion. Conversion status may be determined in two ways. The **EOC** bit in BADR1 + 0 may be polled until true or **ADNEI** (the AD FIFO not-empty interrupt) may be used to signal that the ADC conversion is complete and the data word is present in the FIFO.

READ

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

MSB LSB

**AD[11:0]** This register contains the current ADC data word. Data format is dependent upon offset mode:

***Bipolar Mode:*** Offset Binary Coding  
000 h = -FS  
7FFh = Mid-scale (0V)  
FFFh = +FS - 1LSB

***Unipolar Mode:*** Straight Binary Coding  
000 h = -FS (0V)  
7FFh = Mid-scale (+FS/2)  
FFFh = +FS - 1LSB

### 7.4.2 ADC FIFO CLEAR REGISTER

BADR2 + 2

ADC FIFO Clear register. A Write-only register. A write to this address location clears the ADC FIFO. Data is don't care. The ADC FIFO should be cleared before all new ADC operations.

## 7.5 BADR3

The I/O Region defined by BADR3 contains data and control registers for the ADC Pacer, Pre/Post-Trigger Counters, User Counters and Digital I/O bytes. The PCI-DAS1200 has two 8254 counter/timer devices. These are referred to as 8254A and 8254B and are assigned as shown below:

Device	Counter #	Function
8254A	0	ADC Post-Trigger Sample Counter
8254A	1	ADC Pacer Lower Divider
8254A	2	ADC Pacer Upper Divider
8254B	0	User Counter #3 & ADC Pre-Trigger Index Counter
8254B	1	User Counter #4
8254B	2	User Counter #5

All reads/writes to BADR3 are *byte* operations.

### 7.5.1 ADC PACER CLOCK DATA AND CONTROL REGISTERS

#### 8254A COUNTER 0 DATA - ADC POST TRIGGER CONVERSION COUNTER

BADR3 + 0

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Counter 0 is used to stop the acquisition when the desired number of samples have been gathered. It essentially is gated on when a 'residual' number of conversions remain. The main counting of samples is done by the Interrupt Service Routine, which will increment each time by 'packets' equal to 1/2 FIFO. Generally the value loaded into Counter 0 is  $N \bmod 1024$ , where N is the total count, or the post trigger count, since Total count is not known when pre-trigger is active. Counter 0 will be enabled by use of the **ARM** bit (BADR1 + 4) when the next-to-last 1/2-full interrupt is processed. Counter 0 is to operated in Mode 0.

#### 8254A COUNTER 1 DATA - ADC PACER DIVIDER LOWER

BADR3 + 1

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

## 8254A COUNTER 2 DATA - ADC PACER DIVIDER UPPER

BASE + 2

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Counter 1 provides the lower 16 bits of the 32-bit pacer clock divider. Its output is fed to the clock input of Counter 2 which provides the upper 16-bits of the pacer clock divider. The clock input to Counter 1 is a precision 10MHz oscillator source.

Counter 2 output is called the 'Internal Pacer' and can be selected by software to be the ADC Pacer source. Counters 1 & 2 should be configured to operate in 8254 Mode 2.

## ADC 8254 CONTROL REGISTER

BADR3 + 3

WRITE ONLY

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

The control register is used to set the operating Modes of 8254 Counters 0,1 & 2. A counter is configured by writing the correct Mode information to the Control Register followed by count written to the specific Counter Register.

The Counters on the 8254 are 16-bit devices. Since the interface to the 8254 is only 8-bits wide, Count data is written to the Counter Register as two successive bytes. First the low byte is written, then the high byte. The Control Register is 8-bits wide. Further information can be obtained on the 8254 data sheet, available from Intel or Harris.

## 7.5.2 DIGITAL I/O DATA AND CONTROL REGISTERS

The 24 DIO lines on the PCI-DAS1200 are grouped as three byte-wide I/O ports. Port assignment and functionality is identical to that of the industry standard 8255 Peripheral Interface. Please see the Intel or Harris data sheets for more information.

## DIO PORT A DATA

BADR3 + 4

PORT A may be configured as an 8-bit I/O channel.

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0



## DIO PORT B DATA

BADR3 + 5

PORT B may be configured as an 8-bit I/O channel. Its functionality is identical to that of PORT A.

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

## DIO PORT C DATA

BADR3 + 6

PORT C may be configured as an 8-bit port of either input or output, or it may be split into two independent 4-bit ports of input or output. When split into two 4-bit I/O ports, **D[3:0]** make up the lower nibble, **D[7:4]** comprise the upper nibble. Although it may be split, every write to Port C is a byte operation. Unwanted information must be ANDed out during reads and writes must be ORd with current value of the other 4-bit port.

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

## DIO CONTROL REGISTER

BADR3 + 7

The DIO Control register is used configure Ports A,B and C as inputs or outputs. Operation is identical to that of the 8255 in Mode 0.

WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

The following table summarizes the possible I/O Port configurations for the PCI-DAS1200 DIO operatin in MODE 0:

D4	D3	D1	D0	PORT A	PORT C UPPER	PORT B	PORT C LOWER
0	0	0	0	OUT	OUT	OUT	OUT
0	0	0	1	OUT	OUT	OUT	IN
0	0	1	0	OUT	OUT	IN	OUT
0	0	1	1	OUT	OUT	IN	IN
0	1	0	0	OUT	IN	OUT	OUT
0	1	0	1	OUT	IN	OUT	IN
0	1	1	0	OUT	IN	IN	OUT
0	1	1	1	OUT	IN	IN	IN
1	0	0	0	IN	OUT	OUT	OUT
1	0	0	1	IN	OUT	OUT	IN
1	0	1	0	IN	OUT	IN	OUT
1	0	1	1	IN	OUT	IN	IN
1	1	0	0	IN	IN	OUT	OUT
1	1	0	1	IN	IN	OUT	IN
1	1	1	0	IN	IN	IN	OUT
1	1	1	1	IN	IN	IN	IN

### 7.5.3 INDEX and USER COUNTER DATA AND CONTROL REGISTERS

#### 8254B COUNTER 0 DATA—ADC PRE-TRIGGER INDEX COUNTER (or USER COUNTER #4)

BADR3 + 8

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Counter 0 of the 8254B device is a shared resource on the PCI-DAS1200. When not in ADC pre-trigger mode, the clock, gate and output lines of Counter 0 are available to the user at the 100-pin connector as User Counter 4. The Counter 0 clock source is SW selectable via the **C0SRC** bit in BADR1+4.

When in ADC Pre-trigger mode, this counter is used as the ADC Pre-Trigger index counter. This counter serves to mark the boundary between pre- and post-trigger samples when the ADC is operating in Pre-Trigger Mode. The External ADC Trigger flip flop gates Counter 0 on; the ADC FIFO Half-Full signal gates it off. Knowing the desired number of post-trigger samples, software can then calculate how many 1/2 FIFO data packets need to be collected and what corresponding residual sample count needs to be written to BADR3 + 0.

## 8254B COUNTER 1 DATA - USER COUNTER #5

BADR3 + 9

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

The clock, gate and output lines of Counter 1 are available to the user at the 100-pin connector as User Counter 5. The Counter 1 clock source is always external and must be provided by the user. The buffered version of the internal 10-MHz clock available at the user connector may be used as the clock source.

## 8254B COUNTER 2 DATA - USER COUNTER #6

BADR3 + Ah

READ/WRITE

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

The clock, gate and output lines of Counter 2 are available to the user at the 100-pin connector as User Counter 6. The Counter 2 clock source is always external and must be provided by the user. The buffered version of the internal 10-MHz clock available at the user connector may be used as the clock source.

## 8254B CONTROL REGISTER

BADR3 + Bh

WRITE ONLY

7	6	5	4	2	3	1	0
D7	D6	D5	D4	D3	D2	D1	D0

The control register is used to set the operating Modes of 8254B Counters 0,1 & 2. A counter is configured by writing the correct Mode information to the Control Register, then the proper count data must be written to the specific Counter Register.

The Counters on the 8254 are 16-bit devices. Since the interface to the 8254 is only 8-bits wide, Count data is written to the Counter Register as two successive bytes. First the low byte is written, then the high byte. The Control Register is 8-bits wide. Further information can be obtained on the 8254 data sheet, available from Intel or Harris.

---

---

## 7.6 BADR4

(Does not apply to PCI-DAS1200/JR)

The I/O Region defined by BADR4 contains the DAC0 and DAC1 data registers.

### 7.6.1 DAC0 DATA REGISTER

BADR4 + 0

WRITE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	DAC0(11)	DAC0(10)	DAC0(9)	DAC0(8)	DAC0(7)	DAC0(6)	DAC0(5)	DAC0(4)	DAC0(3)	DAC0(2)	DAC0(1)	DAC0(0)
MSB								LSB							

Writing to this register will initiate data conversion on DAC0. If the **MODE** bit in BADR1+8 is set, writes to this register will provide a simultaneous update of both DAC0 and DAC1 with the data written to this register. The data format is dependent upon the offset mode described below:

***Bipolar Mode:*** Offset Binary Coding

000 h = -FS  
7FFh = Mid-scale (0V)  
FFFh = +FS - 1LSB

***Unipolar Mode:*** Straight Binary Coding

000 h = -FS (0V)  
7FFh = Mid-scale (+FS/2)  
FFFh = +FS - 1LSB

### 7.6.2 DAC1 DATA REGISTER

(Does not apply to PCI-DAS1200/JR)

BADR4 + 2

WRITE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	DAC1(11)	DAC1(10)	DAC1(9)	DAC1(8)	DAC1(7)	DAC1(6)	DAC1(5)	DAC1(4)	DAC1(3)	DAC1(2)	DAC1(1)	DAC1(0)
MSB								LSB							

Writing to this register will initiate data conversion on DAC1. If the **MODE** bit in BADR1+8 is set, writes to this register will have no effect

## 8.0 Electrical Specifications

---

(Typical specifications for 25 Deg C unless otherwise specified.)

---

### ANALOG INPUT SECTION

Resolution	12 bits
Programmable ranges	$\pm 10$ V, $\pm 5$ V, $\pm 2.5$ V, $\pm 1.25$ V, 0 to 10 V, 0 - 5 V, 0 to 2.5 V, 0 to 1.25V
A/D pacing	Programmable: internal counter or external source (A/D External Pacer) or software polled
Burstmode	Software selectable option, rate = 3 $\mu$ s
A/D Trigger sources	External digital (A/D External Trigger)
A/D Triggering Modes	
Digital:	Software enabled, rising edge, hardware trigger
Pre-trigger:	Unlimited pre- and post-trigger samples. Total # of samples must be > 512.
Data transfer	From 1024 sample FIFO via REPINSW, interrupt or software polled
Polarity	Unipolar/Bipolar, software selectable
Number of channels	8 differential or 16 single-ended, software selectable
A/D conversion time	3 $\mu$ s
Throughput	330 kHz min
Relative Accuracy	$\pm 1.5$ LSB
Differential Linearity error	$\pm 0.75$ LSB
Integral Linearity error	$\pm 0.5$ LSB typ, $\pm 1.5$ LSB max
Gain Error (relative to calibration reference)	$\pm 0.02\%$ of reading max
No missing codes guaranteed	12 bits
Gain drift (A/D specs)	$\pm 6$ ppm/ $^{\circ}$ C
Zero drift (A/D specs)	$\pm 1$ ppm/ $^{\circ}$ C
Common Mode Range	$\pm 10$ V
CMRR @ 60 Hz	70 dB
Input leakage current	200 nA
Input impedance	10 megohms min
Absolute maximum input voltage	$\pm 35$ V

---

---

## ANALOG OUTPUT:

(Does not apply to PCI-DAS1200/JR)

Resolution	12 bits
Number of channels	2
Output Ranges	$\pm 10$ V, $\pm 5$ V, 0 to 5 V, 0 to 10 V. Each channel independently programmable.
D/A pacing	Software
Data transfer	Programmed I/O.
Offset error	$\pm 600$ $\mu$ V max, all ranges (calibrated)
Gain error	$\pm 0.02\%$ FSR max (calibrated)
Differential nonlinearity	$\pm 1$ LSB max
Integral nonlinearity	$\pm 1$ LSB max
Monotonicity	12 bits
D/A Gain drift	$\pm 2$ ppm/ $^{\circ}$ C max
D/A Bipolar offset drift	$\pm 5$ ppm/ $^{\circ}$ C max
D/A Unipolar offset drift	$\pm 5$ ppm/ $^{\circ}$ C max
Throughput	PC dependent
Settling time (to .01% of 10V step)	4 $\mu$ s typ
Slew Rate	7 V/ $\mu$ s
Current Drive	$\pm 5$ mA min
Output short-circuit duration	25 mA indefinite
Output Coupling	dc
Amp Output Impedance	0.1 ohms max
Miscellaneous	Power up and reset, all DACs cleared to 0 volts, $\pm 200$ mV

---

---

## PARALLEL DIGITAL INPUT / OUTPUT

Digital Type	82C55A
Configuration	2 banks of 8, 2 banks of 4, programmable by bank as input or output
Number of channels	24 I/O
Output High	3.0 volts @ -2.5mA min
Output Low	0.4 volts @ 2.5 mA max
Input High	2.0 volts min, $V_{cc}+0.5$ volts absolute max
Input Low	0.8 volts max, GND-0.5 volts absolute min
Power-up / reset state	Input mode (high impedance)
Interrupts	INTA# - mapped to IRQn via PCI BIOS at boot-time
Interrupt enable	Programmable
Interrupt sources	Residual counter, End-of-channel-scan, AD-FIFO-not-empty, AD-FIFO-half-full

---

---

COUNTER SECTION

**Counter type**  
**Configuration**

**82C54**

Two 82C54 devices. 3 down counters per 82C54, 16 bits each

**82C54A:**

Counter 0 - ADC residual sample counter.

Source: ADC Clock.

Gate: Internal programmable source.

Output: End-of-Acquisition interrupt.

Counter 1 - ADC Pacer Lower Divider

Source: 10 MHz oscillator

Gate: Tied to Counter 2 gate, programmable source.

Output: Chained to Counter 2 Clock.

Counter 2 - ADC Pacer Upper Divider

Source: Counter 1 Output.

Gate: Tied to Counter 1 gate, programmable source.

Output: ADC Pacer clock (if software selected), available at user connector.

**82C54B:**

Counter 0 - Pretrigger Mode

Source: ADC Clock.

Gate: External trigger

Output: End-of-Acquisition interrupt.

Counter 0 - User Counter 4 (when in non-Pretrigger Mode)

Source: User input at 100-pin connector (CLK4) or internal 10 MHz (software selectable)

Gate: User input at 100-pin connector (GATE4).

Output: Available at 100-pin connector (OUT4).

Counter 1 - User Counter 5

Source: User input at 100-pin connector (CLK5).

Gate: User input at 100-pin connector (GATE5).

Output: Available at 100-pin connector (OUT5).

Counter 2 - User Counter 6

Source: User input at 100-pin connector (CLK6).

Gate: User input at 100-pin connector (GATE6).

Output: Available at 100-pin connector (OUT6).

Clock input frequency	10 MHz max
High pulse width (clock input)	30 ns min
Low pulse width (clock input)	50 ns min
Gate width high	50 ns min
Gate width low	50 ns min
Input low voltage	0.8 V max
Input high voltage	2.0 V min
Output low voltage	0.4 V max
Output high voltage	3.0 V min

---

---

OTHER SPECIFICATIONS:

**Power consumption**

Icc: Operating (A/D converting to FIFO)      0.8 A typical, 1.0 A max

**Environmental**

Operating temperature range    0 to 70 °C  
Storage temperature range      -40 to 100 °C  
Humidity                              0 to 90% noncondensing



— FOR YOUR NOTES —

## EC Declaration of Conformity

PCI-DAS1200	High speed analog I/O board for the PCI bus
PCI-DAS1200/JR	High speed analog input board for the PCI bus
<hr/> Part Number	<hr/> Description

to which this declaration relates, meets the essential requirements, is in conformity with, and CE marking has been applied according to the relevant EC Directives listed below using the relevant section of the following EC standards and other normative documents:

**EU EMC Directive 89/336/EEC:** Essential requirements relating to electromagnetic compatibility.

**EU 55022 Class B:** Limits and methods of measurements of radio interference characteristics of information technology equipment.

**EN 50082-1:** EC generic immunity requirements.

**IEC 801-2:** Electrostatic discharge requirements for industrial process measurement and control equipment.

**IEC 801-3:** Radiated electromagnetic field requirements for industrial process measurements and control equipment.

**IEC 801-4:** Electrically fast transients for industrial process measurement and control equipment.

Carl Haapaoja, Director of Quality Assurance

