



# **PCI-DIO48(S) Series Product Manual**

**MANUAL NUMBER : 00650-529-1B**



## 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](http://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](http://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

<b>Chapter 1: Introduction .....</b>	<b>1-1</b>
<b>Chapter 2: Installation .....</b>	<b>2-1</b>
<b>Chapter 3: Option Selection .....</b>	<b>3-1</b>
<b>Chapter 4: Address Selection .....</b>	<b>4-1</b>
<b>Chapter 5: Software .....</b>	<b>5-1</b>
<b>Chapter 6: Programming .....</b>	<b>6-1</b>
<b>Chapter 7: Connector Pin Assignments .....</b>	<b>7-1</b>
<b>Appendix A: Programmable Peripheral Interface Data Sheets .....</b>	<b>A-1</b>

## List of Figures

Figure 1-1: PCI-DIO48(S) and PCI-DIO48(S)S Block Diagram .....	1-4
Figure 3-1: PPCI-DIO48(S) and PCI-DIO48(S)S Option Selection Map .....	3-2

## List of Tables

Table 6-1: Address Assignment Table .....	6-1
Table 6-2: Control Register Bit Assignment .....	6-2
Table 6-3: Change-of-state-interrupt-enable Register .....	6-6
Table 7-1: Connector Pin Assignments .....	7-1

**Revision 1B**

**December 2000**

This page intentionally left blank

# Chapter 1: Introduction

## Features

- 48 Bits of Digital Input/Output.
- Interrupt Generation on Input Change of State. (Model "48S")
- Change-of-state Interrupt Software Enabled in Six 8-Input Ports.(Model "48S")
- All 48 I/O Lines Buffered on the Board.
- I/O Buffers Can Be Enabled/Disabled under Program Control.
- Four and Eight Bit Ports Independently Selectable for I/O.
- Pull-Ups on I/O Lines.
- +5V Supply Available to the User.
- Compatible with Industry Standard I/O Racks like Gordos, Opto-22, Potter & Brumfield, Western Reserve Controls, etc.

## Applications

- Automatic Test Systems.
- Laboratory Automation.
- Robotics.
- Machine Control.
- Security Systems, Energy Management.
- Relay Monitoring and Control.
- Parallel Data Transfer to PC.
- Sensing Switch Closures or TTL, DTL, CMOS Logic.
- Driving Indicator Lights or Recorders.

These cards support 48 bits of parallel digital input/output on the PCI bus. They can be programmed to accept inputs or to provide outputs on two groups of three 8-bit ports. Further, in each group, one of the ports can be further divided into two four-bit nibbles.

The feature that distinguishes the "48S" model from the "48" card is that the state of all inputs can be monitored and, if any one or more bits change state, a latched interrupt request can be generated. Thus, it is not necessary to use software to continuously poll the inputs to detect a change of state. The change-of-state interrupt is disabled/enabled by a software write to an interrupt-enable register. Six bits in that register each control an eight-input port at one of two type 8255-5 Programmable Peripheral Interface chips. The change-of-state interrupt latch can be cleared by a software write.

Also, bit C3 at each 24-bit port can be used as an external interrupt to the computer if the IEN jumpers are installed. When bit C3 goes high (edge triggering), an interrupt is requested. Interrupts from the ports are OR'ed together and OR'ed with the change-of-state interrupt. Interrupt levels are assigned by the system.



## PCI-DIO48(S)(S) Manual

The PCI-DIO48(S)S card was designed for industrial applications and can be installed in 7", or longer, PCI slots of IBM or compatible computers. Each I/O line is buffered and capable of sourcing 15 mA or sinking 24mA (64 mA on request). The card contains two Programmable Peripheral Interface chips type 8255-5 (PPI) to provide computer interface to 48 I/O lines. Each PPI provides three 8-bit ports A, B, and C. Each 8-bit port can be software configured to function as either inputs or output latches. Port C can also be configured as four inputs and four output latches. Pull-ups on the card assure that there are no erroneous outputs at power up until the card is initialized by system software.

Tristate I/O line buffers (74LS245) are configured automatically by hardware logic for input or output use according to direction assignment from a control register in the PPI. Further, if a jumper is properly placed on the card, the tristate buffers may be enabled/disabled under program control. (See the Option Selection section to follow.)

I/O wiring connections are via 50-pin headers on the board. Two flat I/O cables connect PCI-DIO48(S)S to termination panels. Also, this provides compatibility with OPTO-22, Gordos, Potter & Brumfield, et al module mounting racks. Every second conductor of the flat cables is grounded to minimize crosstalk between signals in the cables. If needed for external circuits, +5 VDC power is available on each I/O connector pin 49. If you use this power, we recommend that you include a 1A fast-blow fuse in your circuits in order to avoid possible damage to the host computer or cable in the event of a malfunction in those external circuits.

The PCI-DIO48(S)S occupies sixteen bytes of I/O address space. The base address is selected by the system. An illustrated setup program is provided on diskette with the PCI-DIO48(S)S card. Interactive displays show locations and proper settings jumpers to set up the interrupt enable function. Also, sample programs in Turbo-C and Turbo-Pascal are presented in the Software section of this manual.

## Specification

---

### Digital Inputs (TTL Compatible)

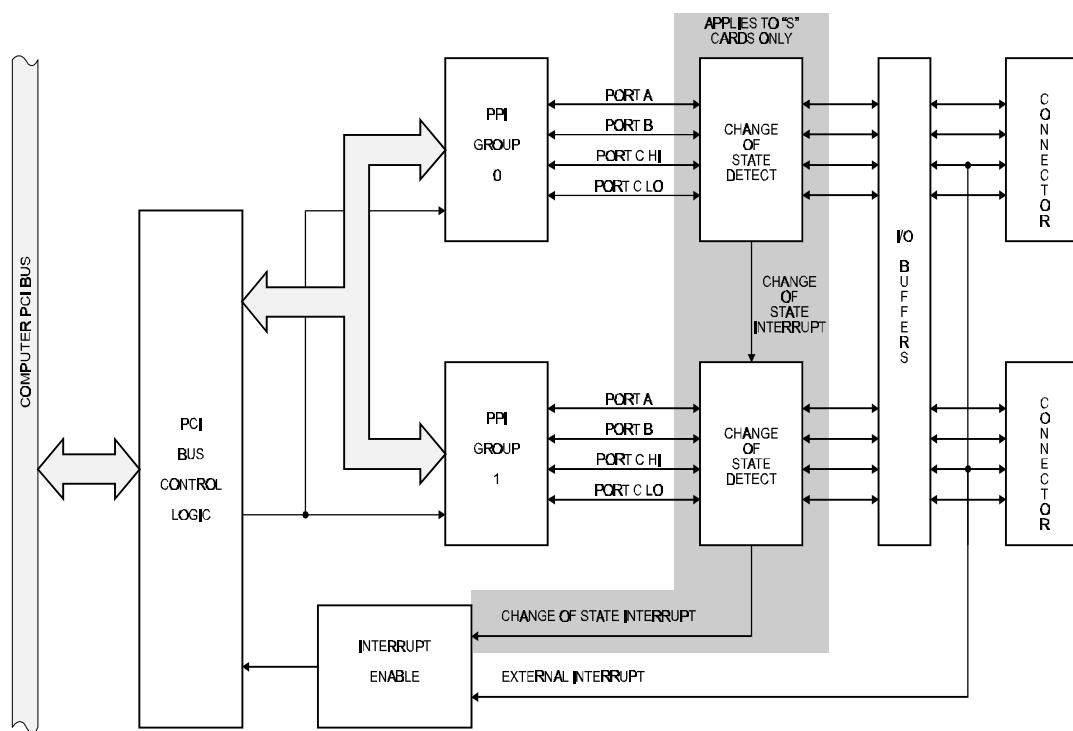
- Logic High: 2.0 to 5.0 VDC.
- Logic Low: -0.5 to +0.8 VDC.
- Input Load (Hi): 20 uA.
- Input Load (Lo): -200 uA.

### Digital Outputs

- Logic High: 2.5 VDC min., source 15 mA.
- Logic Low: 0.5 VDC max., sink 24 mA.  
(64 mA optional)
- Power Output: +5 VDC from computer bus (ext. 1A fast-blow fuse recommended).
- Power Required: +5 VDC at 250 mA typical.
- Size: 6.9" Long.

### Environmental

- Operating Temperature: 0 °C. to 60 °C.
- Storage Temperature: -50 °C. to +120 °C.
- Humidity: 0 to 90% RH, non-condensing.



**Figure 1-1: PCI-DIO48(S) and PCI-DIO48(S)S 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**

---

The PCI-DIO48(S)(S) card 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 the PCI-DIO48(S)(S). The setup program does not set the options. These must be set manually by jumpers on the card.

### **To Install the Card**

1. Turn OFF computer power.
2. Remove the computer cover.
3. Install jumpers from either Chapter 3, Option Selection of this manual or the suggestions of our SETUP.EXE software program.
4. Loosen the nuts on the strain relief bar and swing the top end free.
5. Install the card in an available PCI-bus slot. You may need to remove a backplate first.
6. Thread the I/O cables, one at a time, through the cutout in the mounting bracket and plug them into the headers on the card. Port 0 is the one closest to the mounting bracket.
7. Smooth the cables as close as practicable to the card and, while holding them close to the card surface, swing the strain relief bar into position and tighten the nuts.
8. 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.
9. Replace the computer cover and turn the computer ON.
10. 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 Windows95 (or any other PNP-compliant Operating System) should set the CMOS option to OS. Systems running under DOS, WindowsNT 3.51, 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.

If you are using Windows95, your operating system should detect the new hardware and prompt you for the installation disk. The simplest option is to insert the diskette provided into the A drive and allow the operating system to look for the files that it wants.

Alternatively, if you have already installed the software to your computer, you can specify the installation directory instead of allowing Windows95 to look around. Either option will work acceptably. Please note that even though Windows95 looks for and finds software to install, no driver is actually installed, used, or needed. This step merely tells Windows95 where to place the card in the registry of devices. You can find the installed card in Device Manager, under the class Data Acquisition. Use this method to determine the card's base address as PCIFind.EXE may or may not work under your specific installation of Windows95.

If you are not using Windows95, or if your installation of Windows95 does not detect the new hardware you installed, run PCIFind.EXE to determine the base address that the computer assigned. PCIFind uses the Vendor ID and Device ID to search for your card, then reads the base address and IRQ assigned. If you want to determine the base address and IRQ assignment yourself, the Vendor ID code is 494F (ASCII for "I/O") and the Device ID codes are:

PCI-DIO48(S): 0C60  
PCI-DIO48(S)S: 0E60

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 re-check PCIFind or Device Manager if the hardware configuration is changed,

## **Input/Output Connections**

---

To ensure that there is minimum susceptibility to EMI and minimum radiation, it is important that the card mounting bracket be properly screwed into place and that there be a positive chassis ground. Also, proper EMI cabling techniques (cable connect to chassis ground at the aperture, shielded twisted-pair wiring, etc) be used for the input/output wiring.



## Chapter 3: Option Selection

Refer to the setup programs on the CD provided with the card. Also, refer to Figure 1-1, PCI-DIO48(S) Block Diagram and Figure 3-1, PCI-DIO48(S) Option Selection Map when reading this section of the manual.

External Interrupts are accepted on the I/O connector pin 9 (bit C3) for each group. The Interrupt signal is positive true. External Interrupts are enabled if the IEN0 (for Group 0) and IEN1 (for Group 1) jumper is installed. Interrupts are directed to an available IRQ level by the system.

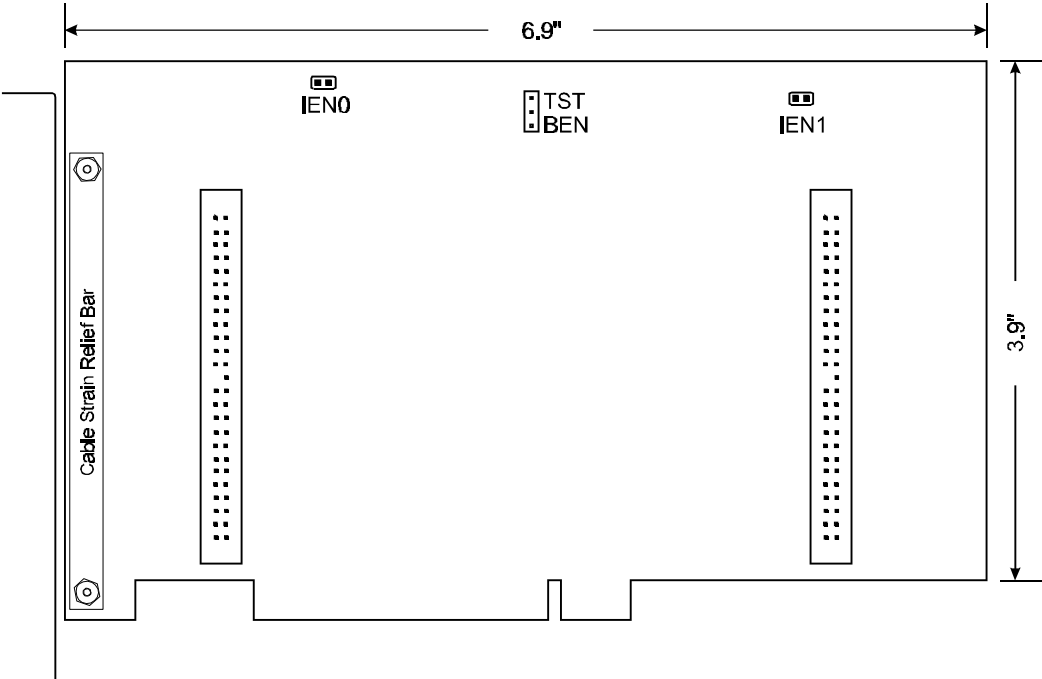
A means of enabling or disabling the 74LS245 input/output buffers under program control is provided at the jumper position labeled TST/BEN. When the jumper is in the BEN (Buffer Enable) position, the I/O buffers are always enabled. When the jumper is in the TST (Tristate) position, enabled/disabled state is controlled by a control register. (See the programming section of this manual for a description.)

An LED, CR1, at the top left of the card to assist you in program development. Each time an interrupt is generated, the LED will illuminate and remain ON until the interrupt is reset. If there is an immediate reset of the interrupt, it is likely that the LED will not remain ON long enough to be observed.

### **Note**

A jumper must be installed in either the TST or the BEN position for the card to function.

The foregoing are the only manual setups necessary to use either the PCI-DIO48(S) or the PCI-DIO48(S)S Input/Output selection and the change-of-state Interrupt Disable/Enable is done via software by writing to a control register in each PPI as described in Chapter 6, Programming.



**Figure 3-1:** PCI-DIO48(S) and PCI-DIO48(S)S Option Selection Map

## Chapter 4: Address Selection

These cards use one address space, occupying sixteen consecutive register locations.

PCI architecture is inherently plug-and-play in nature. This means that the BIOS or Operating System determines the resources assigned to PCI cards rather than you 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 of the ICS Advent cards detected on the PCI bus, the addresses assigned to each function on each of the cards, and the respective IRQs allotted.

Alternatively, some operating systems (Windows95/98/2000) 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 cards are 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.

PCIFind uses the Vendor ID and Device ID to search for your card, then reads the base address and IRQ assigned. If you want to determine these yourself, the Vendor ID is 494F (ASCII for "I/O") and the Device IDs are:

PCI-DIO48(S): 0C60  
PCI-DIO48(S)S: 0E60

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

## Chapter 5: Software

Several programs are provided to support these Digital I/O cards and, also, to help you develop your application's software. Most of these programs are described on page 1-1 of this manual. The following paragraphs describe the IRQCOS driver, Win32COS driver (applicable for PCI-DIO48(S)S only), setup program, and the VisualBASIC utility program

### IRQCOS Driver

---

#### Description

IRQCOS.VXD is a Virtual Device Driver, or VxD, for Windows 95/98. IRQCOS.SYS is a Device Driver for Windows NT 4.0. Functionally, they are identical, and for the purposes of this manual the term IRQCOS Driver will be used to refer to the appropriate file for your operating system. Together with Win32COS.DLL, the IRQCOS Driver allows a program to easily respond to hardware change-of-state interrupt requests, or IRQs, and read data off the card following an IRQ.

#### Installation

To install IRQCOS.VXD in Windows 95/98, copy it to the directory C:\Windows\System. If your hard drive has a letter other than C, substitute the appropriate letter. It is not necessary to reboot your system prior to using IRQCOS.VXD.

To install IRQCOS.SYS in Windows NT, copy it to the directory C:\WINNT\System32\Drivers. If your hard drive has a letter other than C, substitute the appropriate letter. It is not necessary to reboot your system prior to using IRQCOS.SYS.

### Win32COS.DLL

---

#### Description

Win32COS.DLL is a Dynamic Link Library, or DLL, for Windows 95/98 and NT. It provides a simple interface to the IRQCOS Driver, allowing a program to easily respond to hardware change-of-state interrupts and read data off the card after an IRQ occurs. Two functions are exported by Win32COS.DLL: InitCOSDriver, and GetCOSData. To use these functions in a program, they must first be imported into that program. The simplest way to import these functions in Pascal is to include the Win32COS unit in your Uses clause, or, in C, to include Win32COS.H. Additionally, in C, the library file CBCOS.LIB (for C++ Builder) or VCCOS.LIB (for Visual C++) must be linked to the program. This is accomplished by adding it to the project or editing the makefile. If you are not using the Win32COS header file and wish to import the functions manually, the necessary lines in Pascal would be:

```
function InitCOSDriver(BaseAddress: DWORD; IRQ: BYTE; BusType: SmallInt;  
    BusNumber: BYTE): ByteBool; cdecl; external 'Win32COS.dll';  
function GetCOSData(Buf: Pointer): ByteBool; cdecl; external 'Win32COS.dll';
```

To import the functions manually in C, the necessary lines would be:

```
extern "C" __declspec(dllimport) unsigned char InitCOSDriver(unsigned long  
BaseAddress, unsigned char IRQ, short BusType, unsigned char BusNumber);  
extern "C" __declspec(dllimport) unsigned char GetCOSData(void *Buf);
```

### **InitCOSDriver**

This function performs the initialization required to respond to change-of-state IRQs. It must be called before GetCOSData can be called, although calling GetCOSData first will only cause it to return a value of FALSE. The function's parameters are the card's Base Address, the IRQ level to monitor, the card's bus type, and the card's bus number. The last two parameters, BusType and BusNumber, are only used in Windows NT. The header files Win32COS.PAS (for Pascal) and Win32COS.H (for C) contain the enumerated constants that can be passed in the BusType parameter. The two possible values are PCIBus and Isa. For Windows 95/98, the IRQCOS Driver ignores these parameters and their values can be set to zero. The return result is TRUE if initialization was completed successfully, FALSE if not.

### **GetCOSData**


This function suspends the thread that it was called from and waits for an IRQ to occur on the IRQ level that was passed to InitCOSDriver. GetCOSData will return immediately, however, if an IRQ occurred anytime before the GetCOSData call, as long as InitCOSDriver was previously called. Otherwise, GetCOSData will not return until an IRQ occurs. If the program needs to continue running while waiting for GetCOSData, a separate thread should be created from which to call GetCOSData. Immediately upon detecting an IRQ, GetCOSData reads 48 bits of data off the card and returns the data in a pointer supplied by the calling application. This pointer must be at least six bytes and must be allocated by the calling application. For cards that contain less than 48 bits of data, the lowest bits are valid and the extraneous data should be discarded. For example, a card that contains 24 bits of data will return the valid data in the lowest 3 bytes of the memory block pointed to by the supplied pointer, while the upper 3 bytes will be invalid. If multiple interrupts occur before any GetCOSData call, the data read after the most recent IRQ will be returned. The return result is TRUE if an IRQ was detected on the given IRQ level and data was successfully read, FALSE if InitCOSDriver was not called before calling GetCOSData.

## SETUP.EXE

---

This program is supplied in the root or base directory as a tool for you to use in configuring jumpers on the card. It is menu-driven and provides pictures of the card on the computer monitor. You make simple keystrokes to select functions. The picture on the monitor then changes to show how the jumper should be placed to effect your choices.

The setup program is a stand-alone program that can be run at any time. It does not require that the card be plugged into the computer for any part of the setup. The program is self-explanatory with operation instructions and on-line help.

To run this program, at the DOS prompt, enter SETUP.EXE followed by .

## VisualBASIC Utility Driver

---

Extensions to the VisualBASIC 3.0 language are also included on the diskette provided with your card. (For later versions of VisualBASIC, use the ACCES32 driver and samples.) The extensions are in a directory named VBACCES. These extensions are in the form of a .DLL, a .GBL, and a VisualBASIC sample. Together these files allow you to access the port and main memory space in a fashion similar to BASIC, QuickBASIC, Pascal, C/C++, Assembly, and most other standard languages.

To use these files in a VisualBASIC program, you must create a .MAK file (File | New Project) similar to the sample provided (or else, modify your existing project file) and include the .GBL file (File | Add File). Once this has been done, VisualBASIC will be enhanced with the addition of the following functions.

### InPortb

Function: Reads a byte from a hardware port. Due to limitations of VisualBASIC, the number is returned in an integer.

Declaration: `function InPortb(byval address as integer) as integer`

### InPort

Function: Reads an integer from a hardware port. This function returns the 16-bit value obtained from reading the low byte from address and the high byte from address+1.

Declaration: `function InPort(byval address as integer) as integer`

### OutPortb

Function: Writes the lower eight bits of value to the hardware port at address. This function returns the value output.

Declaration: `function OutPortb(byval address as integer, byval value as integer) as integer`

### **OutPort**

Function: Writes all 16 bits of value to the hardware port at address. This function returns the value output.

Declaration: `function OutPort(byval address as integer, byval value as integer) as integer`

### **Peek**

Function: Reads a byte from main memory (DRAM).

Declaration: `function Peek(byval segment as integer, byval offset as integer) as integer`

### **Poke**

Function: Writes the lower eight bits of value to segment:offset.

Declaration: `function Poke(byval segment as integer, byval offset as integer, byval value as integer) as integer`

Note that in all of the above functions, an inherent limitation of BASIC in general and VisualBASIC in particular makes the values sent less intuitive. All integers in BASIC are signed numbers, wherein data are stored in two's complement form. All bit patterns must be converted to-and-from this two's complement form if meaningful display is required. Otherwise, values returned from the InPortb function will be -128 to 127, rather than 0 to 255. An alternative is to perform all assignments in hexadecimal, rather than decimal form.

Before the program will execute, the .GBL file must be modified to include the path to the VBACCES.DLL as appropriate for your system. Merely replace the statement "VBACCES.DLL" with "drive:path\VBACCES.DLL".

As an alternative to changing the source code, you can copy the VBACCES.DLL file into your Windows directory. This will allow multiple programs to find the same .DLL without having to know where it is located. Just leave off all references to a path in the .GBL file as shown in the sample.

## Chapter 6: Programming

These cards are I/O-mapped devices that are easily configured from any language and any language can easily perform digital I/O through the card's ports. This is especially true if the form of the data is byte or word wide. All references to the I/O ports would be in absolute port addressing. However, a table could be used to convert the byte or word data ports to a logical reference.

### Developing Your Application Software

---

If you wish to gain a better understanding of the programs on diskette, then the information in the following paragraphs will be of interest to you. Refer to the data sheets and 8255-5 specification in Appendix A.

A total of 16 register locations are used by the PCI-DIO48(S)(S). The PPIs are addressed consecutively with Address bits A3 through A0 as follows:

Address	Port Assignment	Operation
Base Address	PA Group 0	Read/Write
Base Address +1	PB Group 0	Read/Write
Base Address +2	PC Group 0	Read Write
Base Address +3	Control Group 0	Write Only
Base Address +4	PA Group 1	Read/Write
Base Address +5	PB Group 1	Read/Write
Base Address +6	PC Group 1	Read/Write
Base Address +7	Control Group 1	Write Only
Base Address +8	Enable/DisableBuffer,Grp0	Write Only
Base Address +9	Enable/DisableBuffer,Grp1	Write Only
Base Address +B	Enable Chg-of-St. Interrupt	Write Only
Base Address +F	Clear Chg-of-St. Interrupt	Write Only

**Table 6-1:** Address Assignment Table

These cards use two 8255-5 PPIs to provide a total of 48 bits input/output capability. The cards are designed to use each of these PPIs in Mode 0 wherein:

- There are two 8-bit groups (A and B) and two 4-bit groups (C Hi and C Lo).
- Any group can be configured as an input or an output.
- Outputs are latched.
- Inputs are not latched.



Each PPI contains a Control Register. This write-only, 8-bit register is used to set the mode and direction of the groups. At Power-Up or Reset, all I/O lines are set as inputs. Each PPI should be configured during initialization by writing to the Control Registers even if the groups are only going to be used as inputs. Output buffers are automatically set by hardware according to the Control Register states. Note that Control Registers are located at base address +3 and base address +7. Bit assignments in each of these Control Registers are as follows:

Bit	Assignment	Code
D0	Port C Lo (C0-C3)	1=Input, 0=Output
D1	Port B	1=Input, 0=Output
D2	Mode Select	1=Mode 1, 0=Mode 0
D3	Port C Hi (C4-C7)	1=Input, 0=Output
D4	Port A	1=Input, 0=Output
D5,D6	Mode Select	00=Mode 0, 01=Mode 1, 1x=Mode 2
D7	Mode Set Flag	1=Active

**Table 6-2:** Control Register Bit Assignment

## Note

Mode 1 cannot be used by these cards without modification. Thus, bits D2, D5, and D6 should always be set to "0". If your card has been modified to operate in Mode 1, then there will be an Addendum page in the front of this manual. These cards cannot be used in Mode 2 of the PPI.

## Note

In Mode 0, do not use the control register byte for the individual bit control feature. The hardware uses the I/O bits to control buffer direction on this card. The control register should only be used for setting up input and output of the ports and enabling the buffer.

These cards provide a means to enable/disable the tristate I/O buffers under program control. If the TST/BEN jumper on the card is installed in the BEN position, the I/O buffers are permanently enabled. However, if that jumper is in the TST position, enable/disable of the buffers is software controlled via the control register as follows:

- a. The card is initialized in the receive mode by the computer reset command.
- b. When bit D7 of the Control Register is set high, direction of the three groups of the associated PPI chip as well as the mode can be set. For example, a write to Base Address +3 with data bit D7 high programs port direction at Group 0 ports A, B, and C. If, for example, hex 80 is sent to Base Address +3, the Port 0 PPI will be configured in mode 0 with Groups A, B, and C as outputs.

At the same time, data bit D7 is also latched in a buffer controller for the associated PPI chip. A high state disables the buffers and, thus, all four buffers will be put in the tristate mode; i.e. disabled.

- c. Now, if any of the groups are to be set as outputs, you may set the values to the respective group with the outputs still in the tristate condition. (If all groups are to be set as inputs, this step is not necessary.)
- d. If data bit D7 is low when the control byte is written, ONLY the associated buffer controller is addressed. If, for example, a control byte of hex 80 has been sent as previously described, and the data to be output are correct, and it is now desired to open the three groups, then it is necessary to send a control byte of hex 00 to base address +3 to enable the port 0 buffers. When you do this, the buffers will be enabled.

### Note

Note that all data bits except D7 must be the same for the two control bytes

Those buffers will now remain enabled until another control byte with data bit D7 high is sent to base address +3.

Similarly, the Port 1 groups can be enabled/disabled via the control register at base address +7. The following program fragment in C language illustrates the foregoing:

```
const BASE_ADDRESS 0x300;
outportb(BASE_ADDRESS +3, 0x89);    /*This instruction sets the mode to Mode 0,
                                     ports A and B as output, and port C as input.
                                     Since bit D7 is high, the output buffers are set
                                     to tristate condition. See item b. above.*/

outportb(BASE_ADDRESS,0);
outportb(BASE_ADDRESS+1,0);          /*These instructions set the initial state of ports
                                     A and B to all zeroes. Port C is not set because
                                     it is configured as an input. See item c.
                                     above.*/

outportb(BASE_ADDRESS +3, 0x09);     /*Enable the tristate output buffers by using the
                                     same control byte used to configure the PPI, but
                                     now set bit D7 low. See item d. above.*/
```

Port A	= Input
Port B	= Output
Port C Hi	= Input
Port C Lo	= Output

[illegible]

```
10 BASEADDR=&H2D0
20 OUT BASEADDR+3,&H98
```

To read the inputs at Port A and the upper nybble of Port C:

```
30 X=INP(BASEADDR)      'Read Port A
40 Y=INP(BASEADDR+2)/16  'Read Port C Hi
```

To set outputs high (1) at Port B and the lower nybble of Port C:

```
50 OUT BASEADDR+1,&HFF  'Turn on all Port B bits
60 OUT BASEADDR+2,&HF    'Turn on all bits of Port C
                           lower nybble
```

## **Enabling/Disabling I/O Buffers**

---

When using the tristate mode (Jumper in the TST position), the method to disable the I/O buffers involved writing a control word to the Control Register at Base Address +3 and Base Address +7. This control word was required to have bit D7 (the most significant bit) set. That meant that the PPI translated it as an "active mode set" and reset the output data latches to "zero" on all output ports and the output buffers were disabled. However, if the buffers are to be enabled at a later time, the output latches will be in a "zero" state. For example, if all the outputs were 1's, they will now be 0's and the output buffers will be disabled. This problem can be resolved as follows.

Two computer I/O bus addresses are available that permit you to enable or disable the I/O buffers at will, without programming the PPI mode. Buffers for Port 0 bits are enabled/disabled at Base Address +8 and buffers for Port 1 bits are enabled/disabled at Base Address +9. To enable the buffers and to set outputs to the desired state, you can write to the Control Register with bit D7 low. If you wish to subsequently disable the buffers, you can write to the Control Register with bit D7 high. In this way you can enable/disable the output buffers without programming the PPI mode.

### **Note**

When writing a command byte to these cards while the TST jumper is installed, the PPI output buffers are disabled. Thus, when you desire to change the mode, you must first set the new mode and then enable the buffers. Enabling the buffers can be done at either Base Address +3 (or +7) or Base Address +8 (or +9).

## Change-of-State Interrupts

---

At Power-up or Reset, a latch disables all IRQ sources on the card. In order to properly disable/enable interrupts, you must program the Change-Of-State Interrupt Enable Register first. To program this Change-of-State-Interrupt-Enable Register, write to it at Base Address+B. Data bits D0 through D5 control ports A, B, and C of the 8255 PPIs as shown in Table 4. Any access of Base+B will enable the non-COS IRQ associated with port C bit 3.

Bit	Port Controlled
D0	Group 0, Port A
D1	Group 0, Port B
D2	Group 0, Port C
D3	Group 1, Port A
D4	Group 1, Port B
D5	Group 1, Port C

**Table 6-3:** Change-of-state-interrupt-enable Register

Writing a "one" disables the port; writing a "zero" enables it. When IRQs occur the interrupt state is latched. To clear the latch, write anything to Base Address+F.

## Chapter 7: Connector Pin Assignments

Two 50-pin headers are provided on these cards: one for each 24-bit I/O group. The mating connector is an AMP type 1-746285-0 or equivalent. Connector pin assignments are listed below. Notice that every second line is grounded to minimize crosstalk between signals.

Assignment		Pin		Assignment	Pin
Port C Hi	PC7	1		Ground	2
Port C Hi	PC6	3		"	4
Port C Hi	PC5	5		"	6
Port C Hi	PC4	7		"	8
Port C Lo	PC3*	9		Ground	10
Port C Lo	PC2	11		"	12
Port C Lo	PC1	13		"	14
Port C Lo	PC0	15		"	16
Port B	PB7	17		Ground	18
Port B	PB6	19		"	20
Port B	PB5	21		"	22
Port B	PB4	23		"	24
Port B	PB3	25		"	26
Port B	PB2	27		"	28
Port B	PB1	29		"	30
Port B	PB0	31		"	32
Port A	PA7	33		Ground	34
Port A	PA6	35		"	36
Port A	PA5	37		"	38
Port A	PA4	39		"	40
Port A	PA3	41		"	42
Port A	PA2	43		"	44
Port A	PA1	45		"	46
Port A	PA0	47		"	48
+5 VDC		49		Ground	50

\* This line is an I/O port and also a User Interrupt.

**Table 7-1:** Connector Pin Assignments

# **Appendix A: Programmable Peripheral Interface Data Sheets**

The data sheets in this Appendix are provided to help your understanding of the 8255-5 PPI which is made by a number of companies. These sheets are reprinted with permission of Mitsubishi Electric Corp. ©1987.

The information, diagrams, and all other data included are believed to be correct and reliable. However, no responsibility is assumed by Mitsubishi Electric Corporation for their use, nor for any infringements of patents or other rights belonging to third parties which may result from their use. Values shown on these data sheets are subject to change for product improvement.

# *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:**

PPCI-DIO48(S) and PCI-DIO48(S)S

**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. The CE mark was first applied in 2000.

**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.

Mr. Jim Jameson  
President & Chief Executive Officer



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