



PIO-D48 Series Card User Manual

48-channel DIO board

Version 3.5, Jun. 2018

SUPPORTS

Board includes PIO-D48, PIO-D48U, PIO-D48SU and PEX-D48.

WARRANTY

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

WARNING

ICP DAS assumes no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

COPYRIGHT

Copyright © 2018 by ICP DAS. All rights are reserved.

TRADEMARK

Names are used for identification only and may be registered trademarks of their respective companies.

CONTACT US

If you have any question, please feel to contact us. We will give you quick response within 2 workdays.

Email: service@icpdas.com, service.icpdas@gmail.com

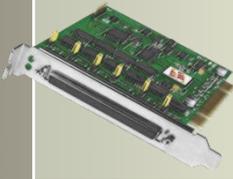


TABLE OF CONTENTS

PACKING LIST 3

1. INTRODUCTION 4

 1.1 FEATURES 5

 1.2 SPECIFICATIONS..... 6

2. HARDWARE CONFIGURATION 7

 2.1 BOARD LAYOUT 7

 2.2 I/O PORT LOCATION 9

 2.3 CARD ID SWITCH..... 10

 2.4 PIN ASSIGNMENTS 11

 2.5 ENABLE I/O OPERATION 12

 2.6 DI/DO ARCHITECTURE..... 13

 2.7 INTERRUPT OPERATION 14

 2.7.1 *Interrupt Block Diagram*..... 15

 2.7.2 *INT_CHAN_0* 16

 2.7.3 *INT_CHAN_1* 17

 2.7.4 *INT_CHAN_2* 18

 2.7.5 *INT_CHAN_3* 19

3. HARDWARE INSTALLATION.....20

4. SOFTWARE INSTALLATION24

 4.1 OBTAINING/INSTALLING THE DRIVER INSTALLER PACKAGE 24

 4.2 PNP DRIVER INSTALLATION 27

 4.3 VERIFYING THE INSTALLATION 29

 4.3.1 *Accessing Windows Device Manager*..... 29

 4.3.2 *Check that the Installation* 32

5. BOARD TESTING33

 5.1 SELF-TEST WIRING 33

 5.1.1 *PIO-D48(U) and PEX-D48* 33

 5.1.2 *PIO-D48SU*..... 34

 5.2 LAUNCH THE TEST PROGRAM..... 35

6. I/O CONTROL REGISTER38

6.1 HOW TO FIND THE I/O ADDRESS 38

6.2 THE ASSIGNMENT OF I/O ADDRESS..... 41

6.3 THE I/O ADDRESS MAP 43

 6.3.1 RESET\ Control Register..... 44

 6.3.2 AUX Status Register..... 44

 6.3.3 INT Mask Control Register..... 45

 6.3.4 Interrupt Polarity Control Register 46

 6.3.5 Read/Write I/O Port 47

 6.3.6 Read/Write 8254..... 49

 6.3.7 Read/Write Clock/Int Control Register 51

 6.3.8 Read Card ID Register..... 51

7. DEMO PROGRAMS52

 7.1 DEMO PROGRAM FOR WINDOWS..... 52

 7.2 DEMO PROGRAM FOR DOS 54

APPENDIX: DAUGHTER BOARD55

 A1. DB-37, DN-37, DN-50 AND DN-100 55

 A2. DB-8125 56

 A3. ADP-37/PCI AND ADP-50/PCI..... 56

 A4. DB-24P AND DB-24PD ISOLATED INPUT BOARD..... 57

 A5. DB-24R AND DB-24RD RELAY BOARD 58

 A6. DAUGHTER BOARDS COMPARISON TABLE 59

Packing List

The shipping package includes the following items:

	One PIO-D48 series card hardware
	One printed Quick Start Guide
	One Software Utility CD

Note:

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

1. Introduction

The PIO-D48U/D48SU and PEX-D48 is the new generation product that ICP DAS provides to meet RoHS compliance requirement and is designed as completely compatible with the PIO-D48. Users can replace the PIO-D48 by the PIO-D48U/PIO-D48SU/PEX-D48 directly without software/driver modification.

The PIO-D48U/D48SU supports 3.3 V/5 V PCI bus, while the PEX-D48 supports PCI Express bus. These cards provide 48 TTL digital I/O lines, and these lines are grouped into six 8-bit bi-direction ports. Every three 8-bit ports are grouped as port A (PA), port B (PB) and port C (PC) on a connector, and the port C can be split into 2 nibble-wide(4-bit) parts. All ports are configured as inputs upon power-up or reset.

The PIO-D48U/D48SU and PEX-D48 also adds a Card ID switch and pull-high/pull-low resistors for DI on board. Users can set Card ID on a board and recognize the board by the ID via software when using two or more PIO-D48U/PIO-D48SU/PEX-D48 cards in one computer. The pull-high/pull-low resistors allow the DI status to be specified when the DI channels are unconnected; the DI status will remain in high or low status other than floating.

These cards support various OS such as Linux, DOS, Windows 98, Windows 2000, Windows NT, 32-/64-bit Windows XP/2003/2008/7/8 and Windows 10. It also provides the DLL and Active X control, and various language sample programs in Turbo C++, Borland c++, Microsoft C++, Visual C++, Borland Delphi, Borland C++ Builder, Visual Basic, C#.NET, Visual Basic.NET and LabVIEW to help users to quickly and easily develop their applications.

1.1 Features

- Support the +5V PCI bus for PIO-D48
- Support the +3.3/+5 V PCI bus for PIO-D48U/D48SU
- Supports PCI Express x 1 for PEX-D48
- 48 channels of Digital Input/Output
- Bi-direction programmable I/O ports under software control
- All I/O lines buffered on the board
- Six 8-bit bi-direction I/O ports
- Connects directly to DB-24PR, DB-24PD, DB-24RD, DB-24PRD, DB-16P8R, DB-24POR, DB-24SSR, DB-24C or any OPTO-22 Compatible daughter boards
- PIO-D48(U)/PEX-D48: One DB37 connector and one 50-pin box headers
- PIO-D48SU: One SCSI II 100-pin connector
- Interrupt source 4-channel
- Card ID function for PIO-D48U/PIO-D48SU/PEX-D48
- One 16-bit event counter and one 32-bit programmable internal timer
- Emulate two industrial-standard 8255 mode 0
- Buffer output for higher driving capability than 8255
- PIO-D48(U)/D48SU:DIO response time is about 1 μ s (1 MHz)
- PEX-D48: DIO response time is about 500 kHz.
- SMD, short card, power saving

1.2 Specifications

Model Name	PIO-D48 (Phased-out)	PIO-D48U	PIO-D48SU	PEX-D48
Programmable Digital I/O				
Channels	48			
Digital Input				
Compatibility	5 V/TTL			
Input Voltage	Logic 0: 0.8 V max. Logic 1: 2.0 V min.			
Response Speed	1 MHz			500 KHz
Digital Output				
Compatibility	5 V/TTL			
Output Voltage	Logic 0: 0.4 V max. Logic 1: 2.4 V min.			
Output Capability	Sink: 64mA @ 0.8 V Source: 32 mA @ 2.0 V			
Response Speed	1 MHz			500 KHz
General				
Bus Type	5 V PCI, 32-bit, 33 MHz	3.3 V/5 V Universal PCI, 32-bit, 33 MHz		PCI Express x1
Data Bus	8-bit			
Card ID	No	Yes(4-bit)		
I/O Connector	Female DB37 x 1 50-pin box header x 1	Female SCSI II 100 pin x 1	Female DB37 x 1 50-pin box header x 1	
Dimensions (L x W x D)	156 mm x 105 mm x 22mm	140 mm x 97 mm x 22 mm	172 mm x 112 mm x 22 mm	
Power Consumption	900 mA @ +5 V			1500 mA @ +3.3 V 0 mA @ +12 V
Operating Temperature	0 ~ 60 °C			
Storage Temperature	-20 ~ 70 °C			
Humidity	5 ~ 85% RH, non-condensing			

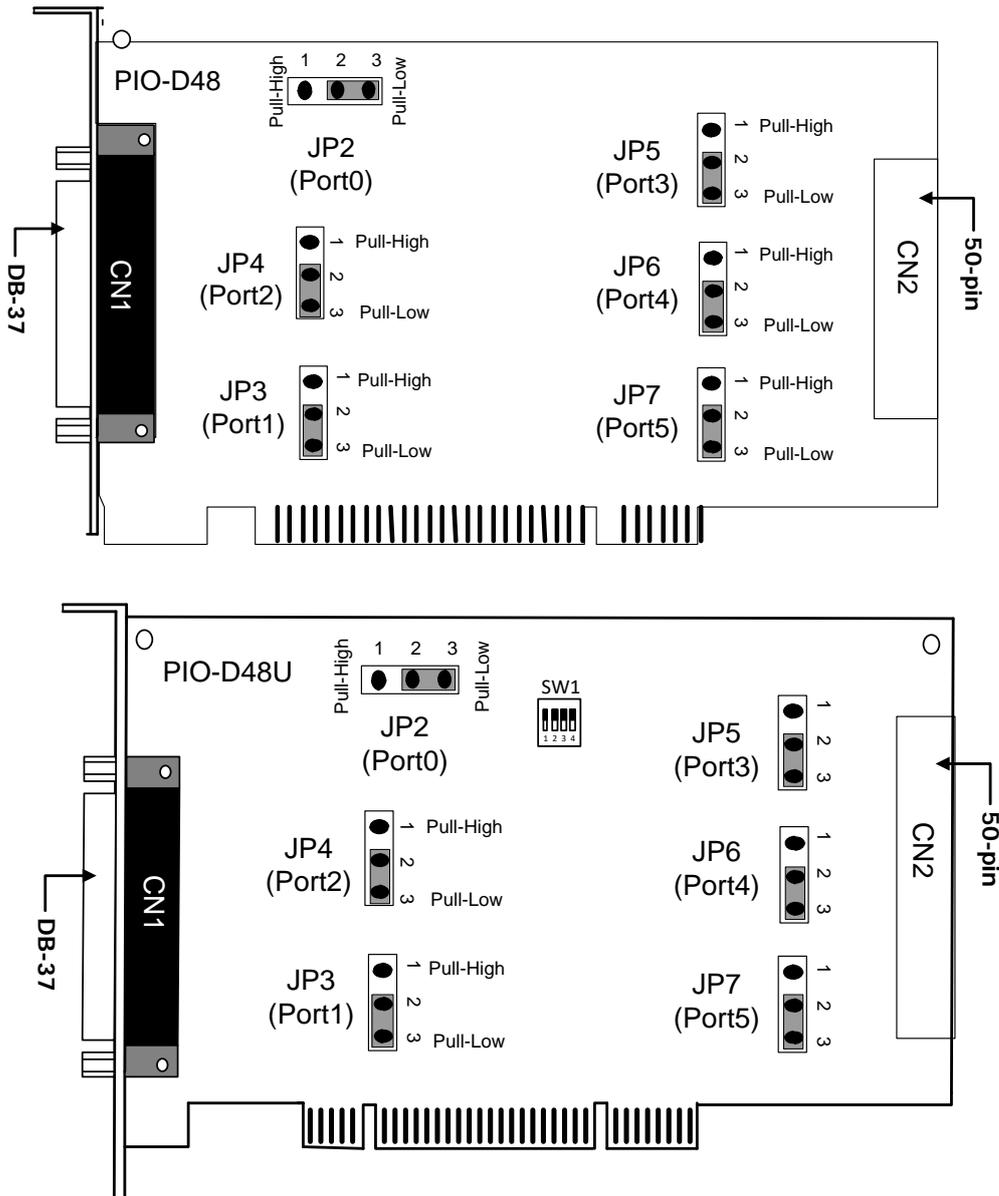
Note:

The I/O speed is depending on I/O card, bus speed, CPU speed and system loading. Any condition changes may cause the I/O speed different.

2. Hardware Configuration

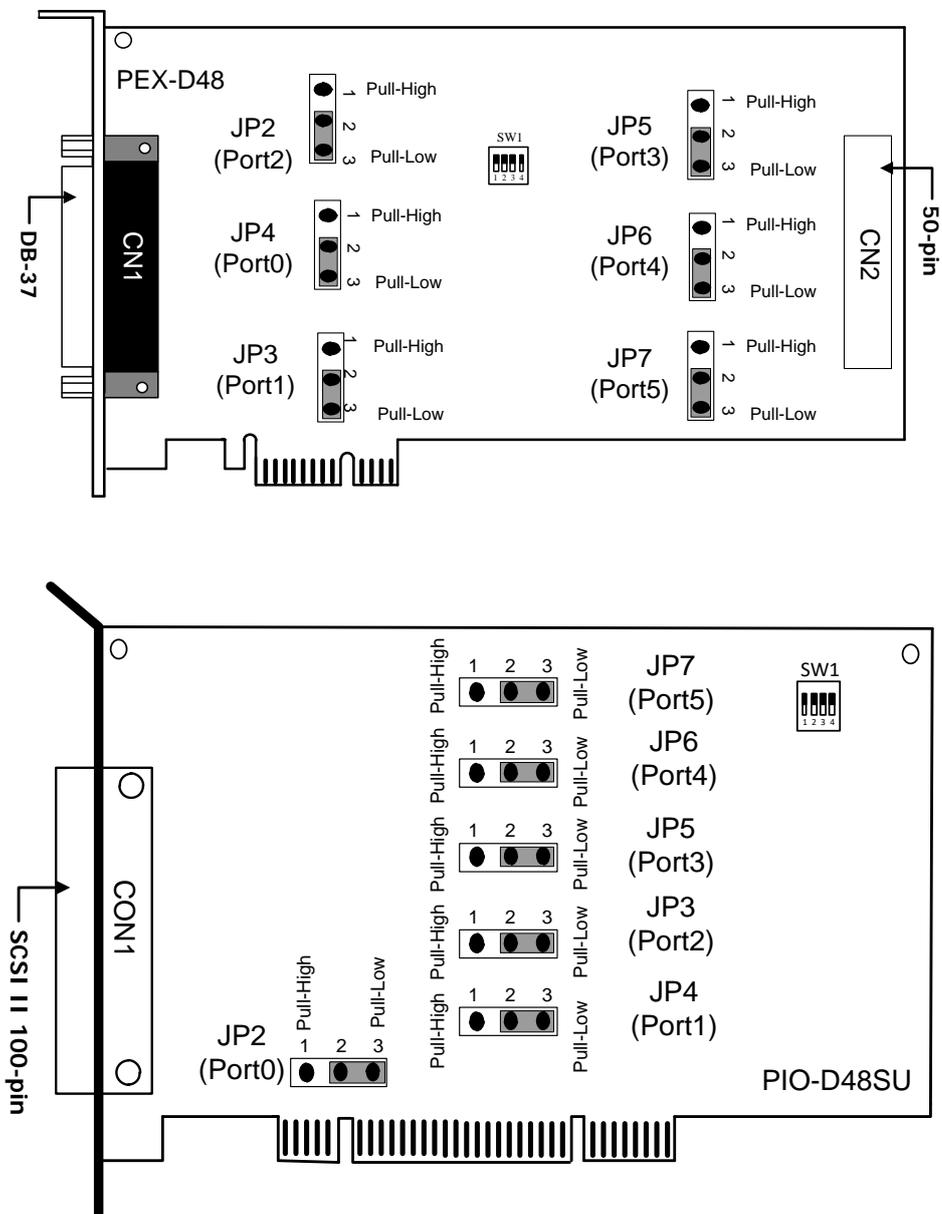
2.1 Board Layout

➤ The board layout of the PIO-D48 and PIO-D48U cards are shown below:



Note: Default Setting: JP2/3/4/5/6/7=2-3 short = Pull-low, refer to [Section 2.2 "I/O Port Location"](#) for more detailed about DI pull-high/low information.

➤ The board layout of the PEX-D48 and PIO-D48SU cards are shown below:



Note: Default Setting: JP2/3/4/5/6/7=2-3 short = Pull-low, refer to [Section 2.2 "I/O Port Location"](#) for more detailed about DI pull-high/low information.

2.2 I/O Port Location

There are six 8-bit I/O ports in the PIO-D48 series cards. Each I/O port can be programmed as a D/I or D/O port. When the PC is first powered-on or reset, all the ports are configured as D/I ports. These D/I ports can be selected to either pull-high or pull-low via placement of the JP2 to JP7 jumpers. These I/O port locations are given as follows:

Connector of <u>PIO-D48/D48U</u>	CN1 (DB37)	CN2 (50-pin)
PA0 ~ PA7	Port0 (pull-high/low by JP2)	Port3 (pull-high/low by JP5)
PB0 ~ PB7	Port1 (pull-high/low by JP3)	Port4 (pull-high/low by JP6)
PC0 ~ PC7	Port2 (pull-high/low by JP4)	Port5 (pull-high/low by JP7)

Connector of <u>PEX-D48</u>	CN1 (DB37)	CN2 (50-pin)
PA0 ~ PA7	Port0 (pull-high/low by JP4)	Port3 (pull-high/low by JP5)
PB0 ~ PB7	Port1 (pull-high/low by JP3)	Port4 (pull-high/low by JP6)
PC0 ~ PC7	Port2 (pull-high/low by JP2)	Port5 (pull-high/low by JP7)

Connector of <u>PIO-D48SU</u>	CON1 (SCSI 100-pin)
PA00 ~ PA07	Port0 (pull-high/low by JP2)
PB00 ~ PB07	Port1 (pull-high/low by JP3)
PC00 ~ PC07	Port2 (pull-high/low by JP4)
PA10 ~ PA17	Port3 (pull-high/low by JP5)
PB10 ~ PB17	Port4 (pull-high/low by JP6)
PC10 ~ PC17	Port5 (pull-high/low by JP7)

Note:

This board is a bi-directional I/O design with default DI mode when power on. Before switching to DO mode, the DI pull-high jumper setting may activate active-high DO devices (e.g., DB-24R / 24PR / 24C), or the pull-low setting may activate active-low DO devices. Please have a appropriate jumper setting depending on the characteristics of your external device.

2.3 Card ID Switch

The PIO-D48U/D48SU and PEX-D48 has a Card ID switch (SW1) with which users can recognize the board by the ID via software when using two or more PIO-D48U/D48SU and PEX-D48 cards in one computer. The default Card ID is 0x0. For detail SW1 Card ID settings, please refer to Table 2.1.

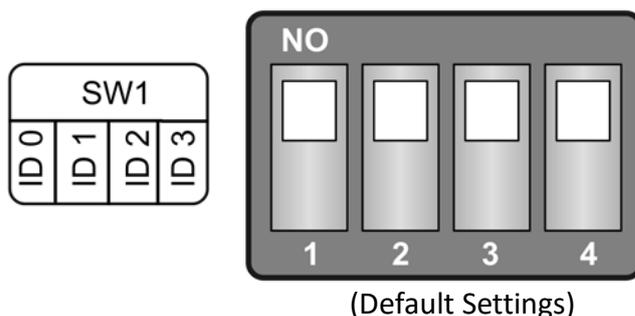


Table 2.1 (*) Default Settings; OFF → 1; ON → 0

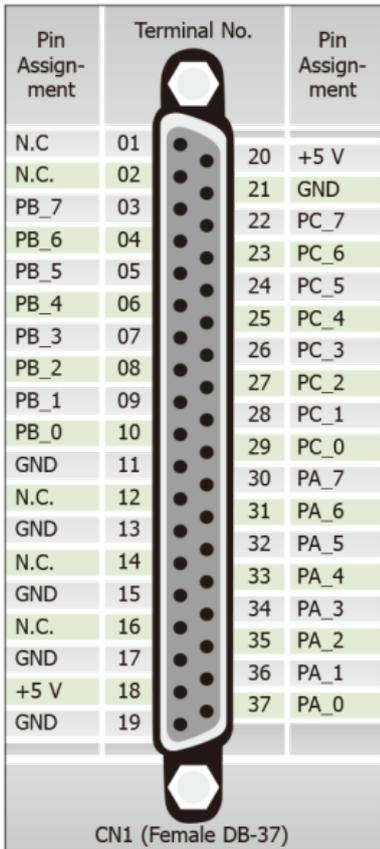
Card ID (Hex)	1 ID0	2 ID1	3 ID2	4 ID3
(*) 0x0	ON	ON	ON	ON
0x1	OFF	ON	ON	ON
0x2	ON	OFF	ON	ON
0x3	OFF	OFF	ON	ON
0x4	ON	ON	OFF	ON
0x5	OFF	ON	OFF	ON
0x6	ON	OFF	OFF	ON
0x7	OFF	OFF	OFF	ON
0x8	ON	ON	ON	OFF
0x9	OFF	ON	ON	OFF
0xA	ON	OFF	ON	OFF
0xB	OFF	OFF	ON	OFF
0xC	ON	ON	OFF	OFF
0xD	OFF	ON	OFF	OFF
0xE	ON	OFF	OFF	OFF
0xF	OFF	OFF	OFF	OFF

2.4 Pin Assignments

The Pin assignments for all connectors on the PIO-D48(U)/D48SU and PEX-D48 are represented in the figure below. All signal sources for each Digital Input or Output channel is TTL compatible.

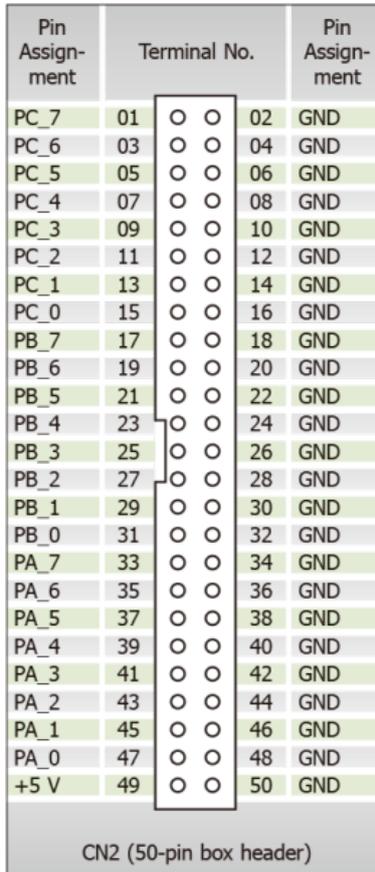
PIO-D48/D48U, PEX-D48

Pin Assignment	Terminal No.	Pin Assignment
N.C	01	20 +5 V
N.C.	02	21 GND
PB_7	03	22 PC_7
PB_6	04	23 PC_6
PB_5	05	24 PC_5
PB_4	06	25 PC_4
PB_3	07	26 PC_3
PB_2	08	27 PC_2
PB_1	09	28 PC_1
PB_0	10	29 PC_0
GND	11	30 PA_7
N.C.	12	31 PA_6
GND	13	32 PA_5
N.C.	14	33 PA_4
GND	15	34 PA_3
N.C.	16	35 PA_2
GND	17	36 PA_1
+5 V	18	37 PA_0
GND	19	



CN1 (Female DB-37)

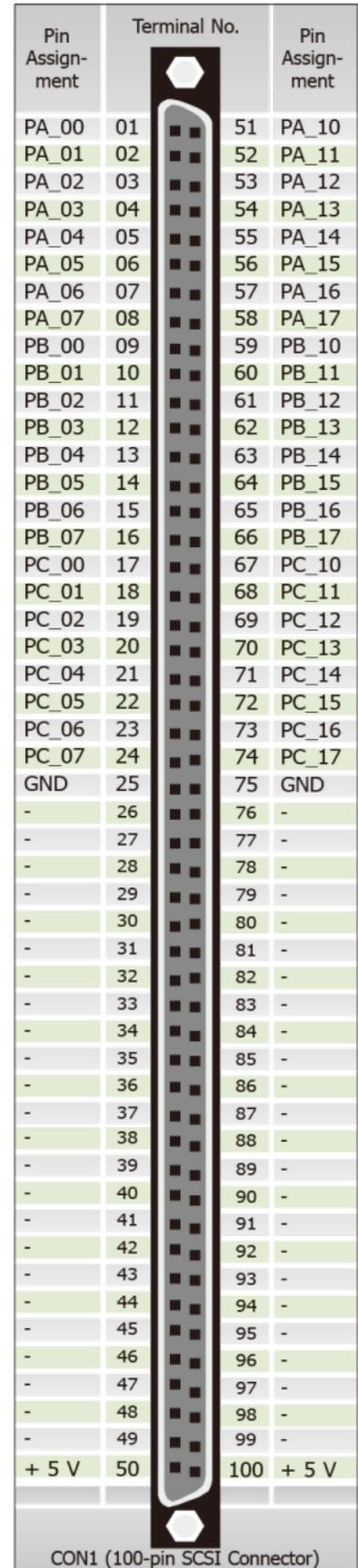
Pin Assignment	Terminal No.	Pin Assignment
PC_7	01	02 GND
PC_6	03	04 GND
PC_5	05	06 GND
PC_4	07	08 GND
PC_3	09	10 GND
PC_2	11	12 GND
PC_1	13	14 GND
PC_0	15	16 GND
PB_7	17	18 GND
PB_6	19	20 GND
PB_5	21	22 GND
PB_4	23	24 GND
PB_3	25	26 GND
PB_2	27	28 GND
PB_1	29	30 GND
PB_0	31	32 GND
PA_7	33	34 GND
PA_6	35	36 GND
PA_5	37	38 GND
PA_4	39	40 GND
PA_3	41	42 GND
PA_2	43	44 GND
PA_1	45	46 GND
PA_0	47	48 GND
+5 V	49	50 GND



CN2 (50-pin box header)

PIO-D48SU

Pin Assignment	Terminal No.	Pin Assignment
PA_00	01	51 PA_10
PA_01	02	52 PA_11
PA_02	03	53 PA_12
PA_03	04	54 PA_13
PA_04	05	55 PA_14
PA_05	06	56 PA_15
PA_06	07	57 PA_16
PA_07	08	58 PA_17
PB_00	09	59 PB_10
PB_01	10	60 PB_11
PB_02	11	61 PB_12
PB_03	12	62 PB_13
PB_04	13	63 PB_14
PB_05	14	64 PB_15
PB_06	15	65 PB_16
PB_07	16	66 PB_17
PC_00	17	67 PC_10
PC_01	18	68 PC_11
PC_02	19	69 PC_12
PC_03	20	70 PC_13
PC_04	21	71 PC_14
PC_05	22	72 PC_15
PC_06	23	73 PC_16
PC_07	24	74 PC_17
GND	25	75 GND
-	26	76 -
-	27	77 -
-	28	78 -
-	29	79 -
-	30	80 -
-	31	81 -
-	32	82 -
-	33	83 -
-	34	84 -
-	35	85 -
-	36	86 -
-	37	87 -
-	38	88 -
-	39	89 -
-	40	90 -
-	41	91 -
-	42	92 -
-	43	93 -
-	44	94 -
-	45	95 -
-	46	96 -
-	47	97 -
-	48	98 -
-	49	99 -
+ 5 V	50	100 + 5 V



CON1 (100-pin SCSI Connector)

2.5 Enable I/O Operation

When the PC is first turned on, all operations involved with digital I/O channels are disabled. Note that the digital I/O channel of each port is enabled or disabled by the RESET\ signal (Refer to [Section 6.3.1 “RESET\ Control Register”](#) for more information related to this). The power-on states are given as follows:

- DI/DO operations for each port are disabled.
- DI/DO ports are all configured as Digital input ports.
- DO latch register outputs are all high impedance. (Refer to [Section 2.6 “DI/DO Architecture”](#))

The user has to perform some initialization before using these Digital I/O ports. The recommended steps are given below:

Step 1: Find the address-mapping for PIO/PISO cards. (Refer to [Section 6.1 “How to Find the I/O Address”](#))

Step 2: Enable all Digital I/O operations. (Refer to [Section 6.3.1 “RESET\ Control Register”](#)).

Step 3: Configure ports to their expected D/I/O state & send their initial values to every D/O port (Refer to [Section 6.3.7 “Read/Write Clock/Int Control Register”](#))

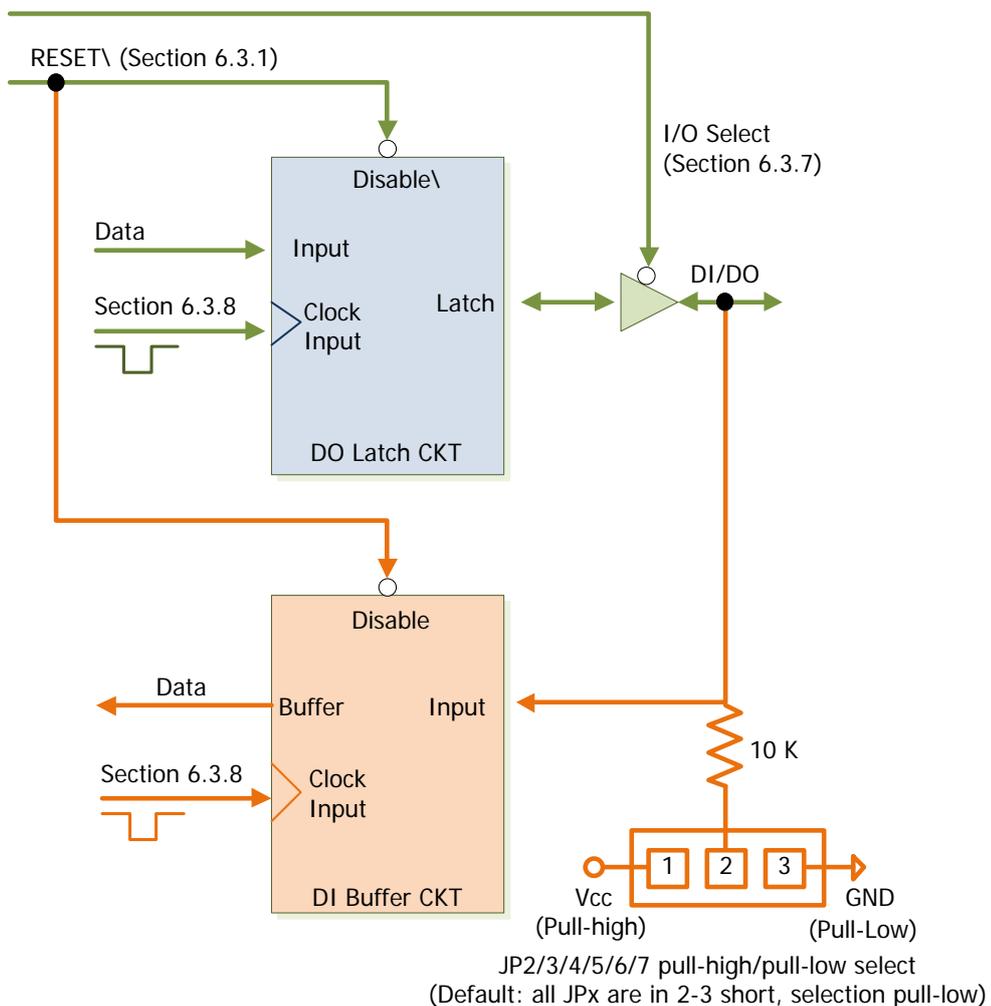
Note:

For more information on the initial procedure for Digital I/O ports, refer to the DIO demo program.

2.6 DI/DO Architecture

The Digital I/O control architecture for the PIO-D48/D48U/D48SU and PEX-D48 are demonstrated in the figure below. The operation method used for the control signal is presented below.

- RESET\ is in the Low-state → all DI/DO operation is disabled
- RESET\ is in the High-state → all DI/DO operation is enabled.
- If DI/DO is configured as a DI port → DI= external input signal.
→DI ports can be configured as either pull-high or pull-low by setting the JP2/3/4/5/6/7 jumpers (shorted 1-2=pull-high; shorted 2-3= pull-low).
- If DI/DO is configured as a DO port → D/I = read back DO.
- If DI/DO is configured as DI port → sending data to a digital input port will only change the DO latch register. The latched data will be output when the port is configured as digital output and is activated right away.



2.7 Interrupt Operation

There are four interrupt sources in the PIO-D48/D48U/D48SU and PEX-D48. These four signals are named INT_CHAN_0, INT_CHAN_1, INT_CHAN_2 and INT_CHAN_3. Their signal sources are given as follows:

- INT_CHAN_0: PC3/PC7 from port-2 (Refer to [Section 2.7.2 “INT_CHAN_0”](#))
- INT_CHAN_1: PC3/PC7 from port-5 (Refer to [Section 2.7.3 “INT_CHAN_1”](#))
- INT_CHAN_2: Cout0 (Refer to [Section 2.7.4 “INT_CHAN_2”](#))
- INT_CHAN_3: Cout2 (Refer to [Section 2.7.5 “INT_CHAN_3”](#))

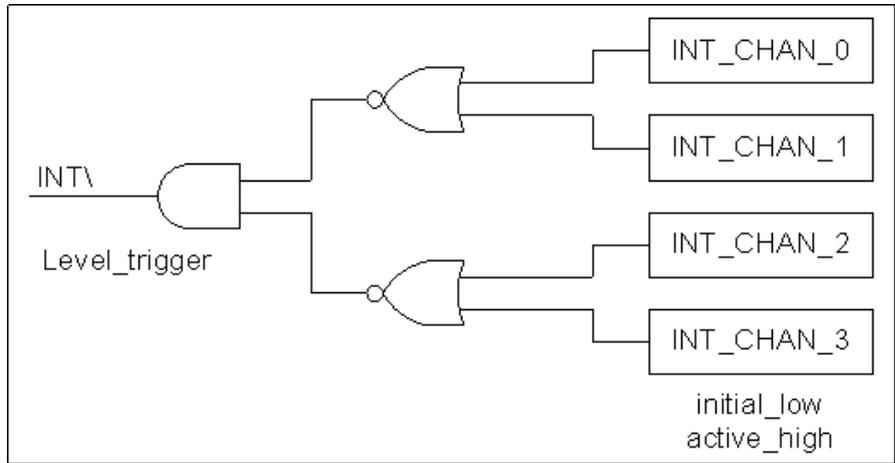
Note that DEMO4.C, DEMO7.C, DEMO8.C, DEMO9.C and DEMO10.C are demo programs for a single interrupt source and DEMO11.C is the demo program for more than one interrupt source in the DOS operating system. If only one interrupt signal source is used, the interrupt service routine does not need to identify the interrupt source. However, if there are more than one interrupt source, the interrupt service routine has to identify the active signals in the following manner:

1. Read the new status of all interrupt signal sources. (refer to [Section 6.3.2 “AUX Status Register”](#))
2. Compare the new status with the old status to identify the active signals.
3. If INT_CHAN_0 is active, service INT_CHAN_0 & non-inverter/inverted the INT_CHAN_0 signal.
4. If INT_CHAN_1 is active, service INT_CHAN_1 & non-inverted/inverted the INT_CHAN_1 signal.
5. If INT_CHAN_2 is active, service INT_CHAN_2 & non-inverted/inverted the INT_CHAN_2 signal.
6. If INT_CHAN_3 is active, service INT_CHAN_3 & non-inverted/inverted the INT_CHAN_3 signal.
7. Update the interrupt status.

Note:

If the interrupt signal is too short, the new status may be the same as the old status. So the interrupt signal must be held active until the interrupt service routine has been executed. This hold time is different for differing operating systems. The hold time can be as short as a micro-second or as long as 1 second. In general, 20 ms is enough for all O.S.

2.7.1 Interrupt Block Diagram



The INT\ interrupt output signals are **level-trigger** and **Active_Low**. If the INT\ generates a low-pulse, the PIO-D48 series cards will interrupt the PC once per occasion. If the INT\ is fixed in low level, the PIO-D48 series cards will interrupt the PC continuously. So the INT_CHAN_0/1/2/3 must be controlled with **pulse_type** signals. **They should normally be fixed in a low level state and generate a high_pulse to interrupt the PC.**

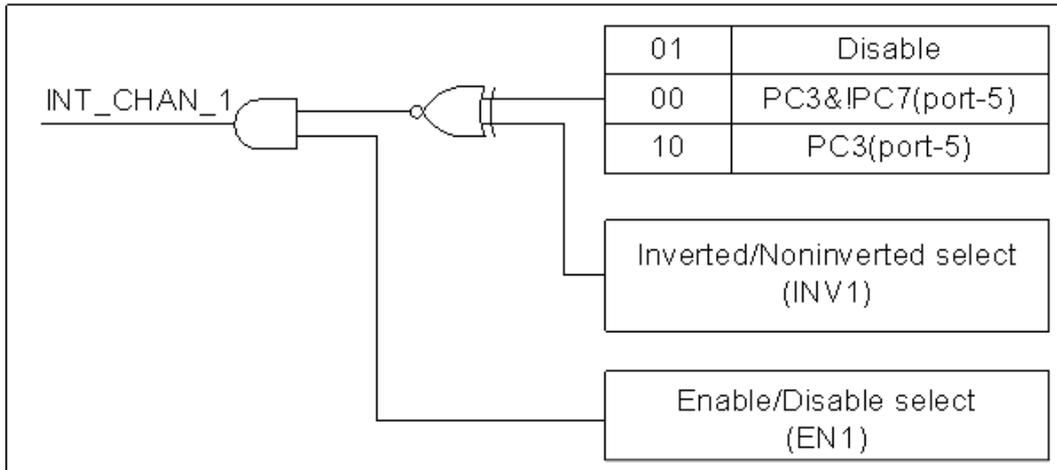
The priority of INT_CHAN_0/1/2/3 is the same. If all these four signals are active at the same time, then INT\ will be active only once per occasion. So the interrupt service routine has to read the status for all interrupt channels for multi-channel interruptions. (Refer to [Section 2.7 “Interrupt Operation”](#) for more information).

DEMO11.C (DOS)	For both INT_CHAN_0 & INT_CHAN_1
----------------	----------------------------------

If only one interrupt source is used, the interrupt service routine doesn't have to read the interrupt source status. Note that DEMO4.C to DEMO10.C is demo programs for a single-channel interruption within the DOS operating system.

DEMO4.C (DOS)	For INT_CHAN_3 only
DEMO7.C (DOS)	For INT_CHAN_2 only
DEMO8.C (DOS)	For INT_CHAN_0 only
DEMO9.C (DOS)	For INT_CHAN_0 only
DEMO10.C (DOS)	For INT_CHAN_1 only

2.7.3 INT_CHAN_1



INT_CHAN_1 should normally be fixed in low level state and generate a high_pulse to interrupt the PC.

INT_CHAN_1 can be equal to **PC3&!PC7** or be **PC3** programmable as is shown below: (Refer to [Section 6.3.7 “Read/Write Clock/Int Control Register”](#))

- CTRL_D5=0, CTRL_D4=1 → INT_CHAN_1= Disabled
- CTRL_D5=1, CTRL_D4=0 → INT_CHAN_1= PC3 of port-5
- CTRL_D5=0, CTRL_D4=0 → INT_CHAN_1= PC3&!PC7 of port-5

EN1 can be used to enable/disable the INT_CHAN_1 as follows: (Refer to [Section 6.3.3 “INT Mask Control Register”](#))

- EN1=0 → INT_CHAN_1= Disabled
- EN1=1 → INT_CHAN_1= Enabled

INV1 can be used to invert/non-invert the PC3 or PC3&!PC7 as follows: (Refer to [Section 6.3.4 “Interrupt Polarity Control Register”](#))

- INV1=0 → INT_CHAN_1=inverted state of (PC3 or PC3&!PC7 of port-5)
- INV1=1 → INT_CHAN_1=non-inverted state of (PC3 or PC3&!PC7 of port-5)

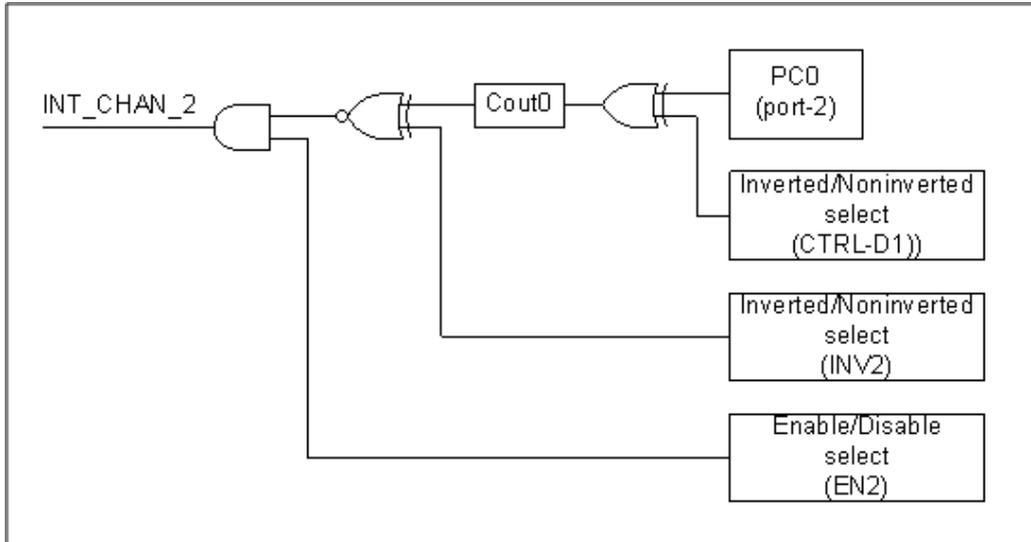
Refer to the following demo program for more information:

DEMO10.C (DOS) For INT_CHAN_1 only (PC3&!PC7 of Port-5)

Note:

Refer to [Section 2.7.2 “INT_CHAN_0” for active high-pulse generation.](#)

2.7.4 INT_CHAN_2



INT_CHAN_2 should normally be fixed in a low-level state and generate a high_pulse to interrupt the PC.

PC0 (Port-2) can be inverted/non-inverted programmable as is shown below: (Refer to [Section 6.3.7 “Read/Write Clock/Int Control Register”](#))

- CTRL_D1=0 → Cin0=PC0 of Port-2
- CTRL_D1=1 → Cin0=!PC0 of Port-2

EN2 can be used to enable/disable the INT_CHAN_2 as follows: (Refer to [Section 6.3.3 “INT Mask Control Register”](#))

- EN2=0 → INT_CHAN_2=disabled
- EN2=1 → INT_CHAN_2=enabled

INV2 can be used to invert/non-invert the Cout0 as follows: (Refer to [Section 6.3.4 “Interrupt Polarity Control Register”](#))

- INV2=0 → INT_CHAN_2=inverted state of (Cout0)
- INV2=1 → INT_CHAN_2=non-inverted state of (Cout0)

Refer to the following demo program for more information:

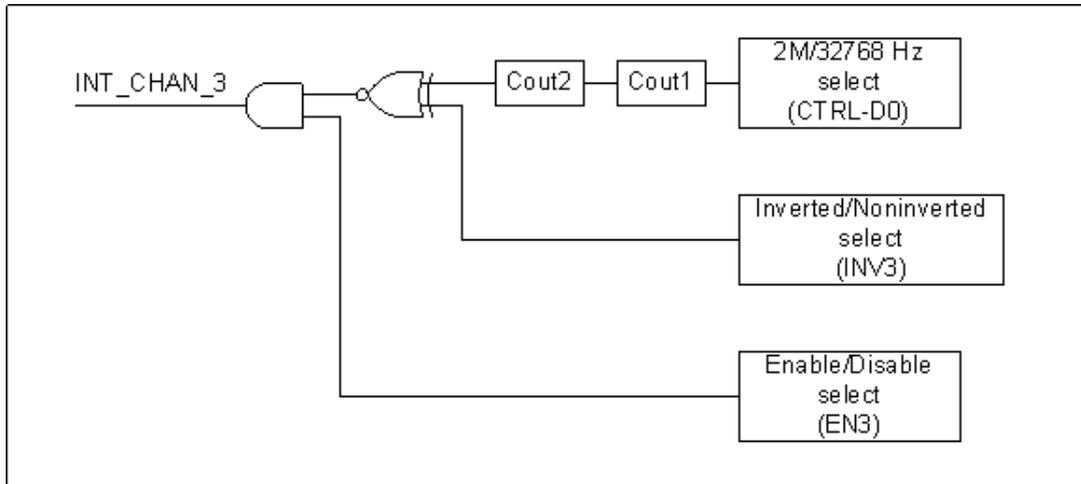
DEMO7.C (DOS)

For INT_CHAN_2 only (Cout0)

Note:

Refer to [Section 2.7.2 “INT_CHAN_0”](#) for active high-pulse generation.

2.7.5 INT_CHAN_3



INT_CHAN_3 should normally be fixed in a low-level state and generate a high_pulse to interrupt the PC.

Cin1 can be 2M/32768Hz programmable as is given below: (Refer to [Section 6.3.7 “Read/Write Clock/Int Control Register”](#))

- CTRL_D0=0 → Cin1=2 M clock source
- CTRL_D0=1 → Cin1=32768 Hz clock source

EN3 can be used to enable/disable the INT_CHAN_3 as follows: (Refer to [Section 6.3.3 “INT Mask Control Register”](#))

- EN3=0 → INT_CHAN_3= Disabled
- EN3=1 → INT_CHAN_3= Enabled

INV3 can be used to invert/non-invert the Cout0 as follows: (Refer to [Section 6.3.4 “Interrupt Polarity Control Register”](#))

- INV2=3 → INT_CHAN_3=invert (Cout2)
- INV2=3 → INT_CHAN_3=non-invert (Cout2)

Refer to the following demo program for more information:

DEMO4.C (DOS)	For INT_CHAN_3 only (Cout2)
---------------	-----------------------------

Note:

Refer to [Section 2.7.2 “INT_CHAN_0”](#) for active high-pulse generation.

3. Hardware Installation

Note:

It is recommended that the driver is installed before installing the hardware as the computer may need to be restarted once the driver is installed in certain operating systems, such as Windows 2000 or Windows XP, etc. Installing the driver first helps reduce the time required for installation and restarting the computer.

To install your PIO-D48 series card, follow the procedure described below:

Step 1: Install the driver for your board on Host computer.



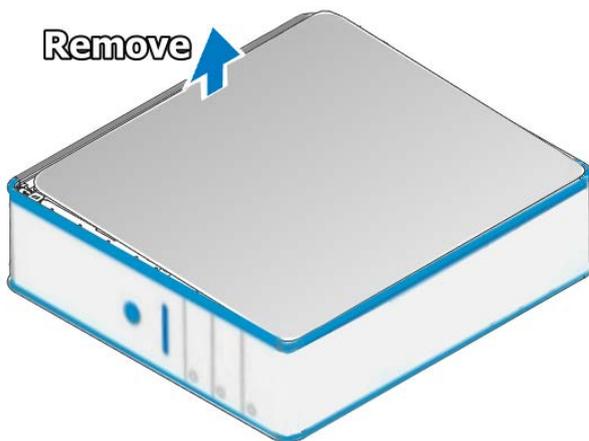
For detailed information about the driver installation, please refer to [Chapter 4 “Software Installation”](#).

Step 2: Configure the Card ID using the DIP Switch (SW1).



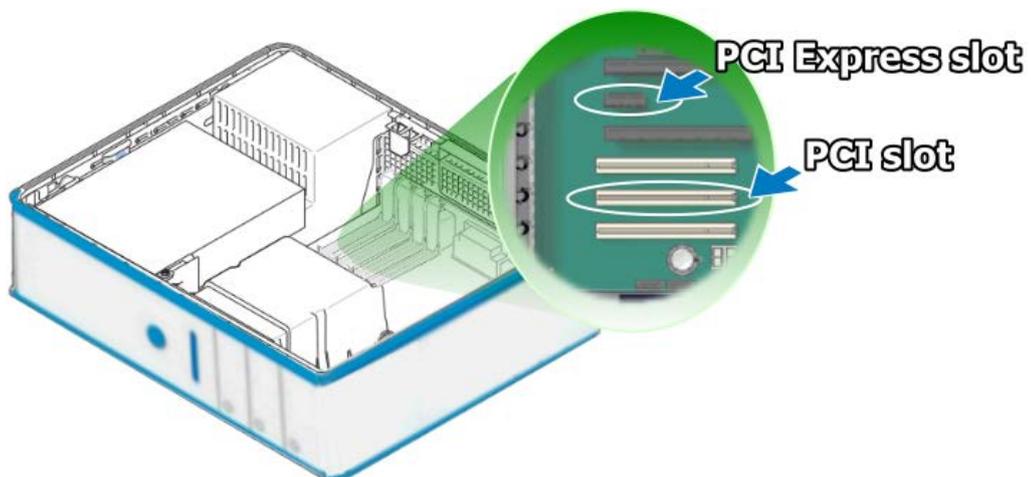
For detailed information about the card ID (SW1), please refer to [Section 2.3 “Car ID Switch”](#).

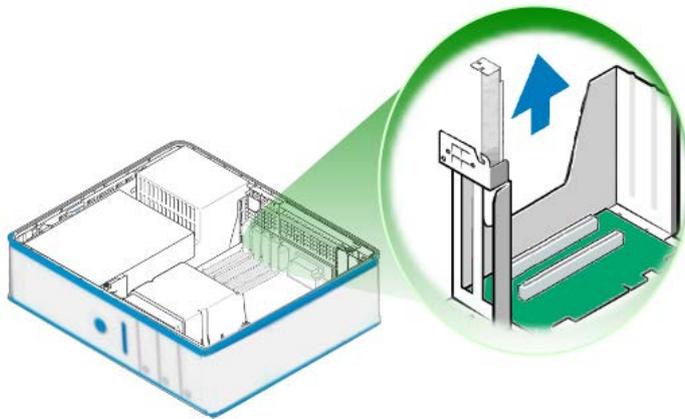
Step 3: Shut down and switch off the power to the computer, and then disconnect the power supply.



Step 4: Remove the cover from the computer.

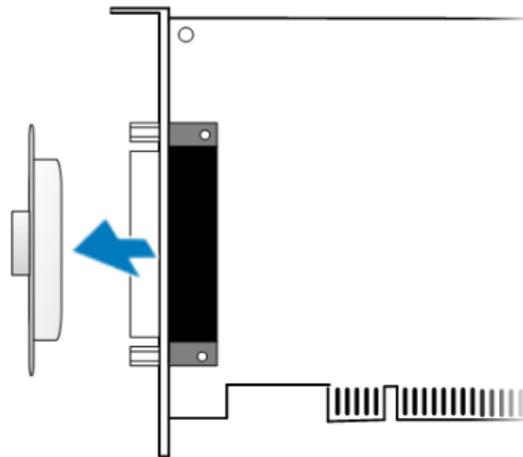
Step 5: Select a vacant PCI/PCI Express slot.



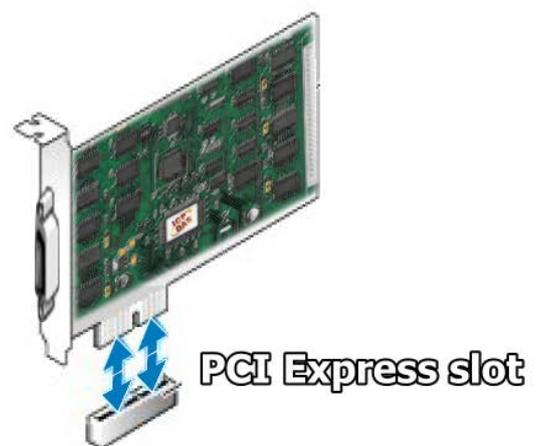
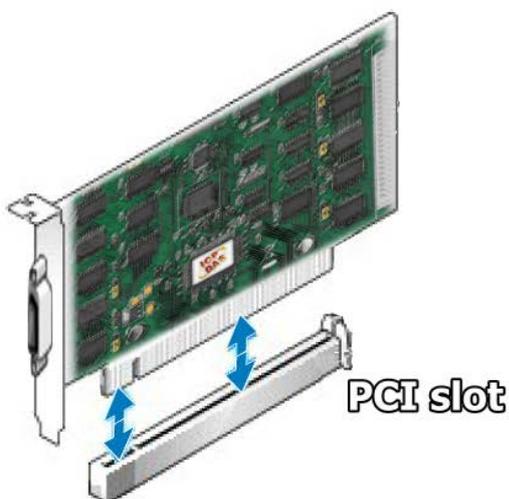


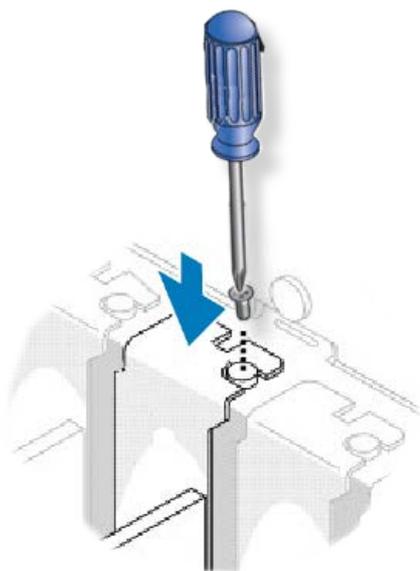
Step 6: Unscrew and remove the PCI slot cover from the computer case.

Step 7: Remove the connector cover from your board.



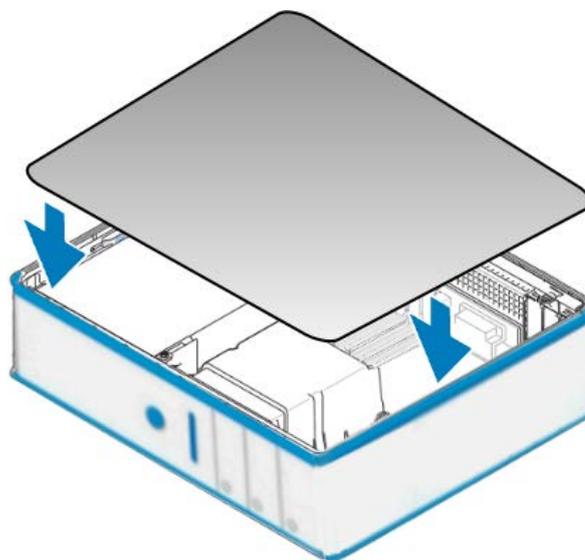
Step 8: Carefully insert your board into the PCI/PCI Express slot by gently pushing down on both sides of the board until it slides into the PCI connector.





Step 9: Confirm that the board is correctly inserted in the motherboard, and then secure your board in place using the retaining screw that was removed in **Step 6**.

Step 10: Replace the covers on the computer.



Step 11: Re-attach any cables, insert the power cord and then switch on the power to the computer.



Once the computer reboots, follow any message prompts that may be displayed to complete the Plug and Play installation procedure. Refer to [Chapter 4 “Software Installation”](#) for more information.

4. Software Installation

This chapter provides a detailed description of the process for installing the driver for the PIO-D48 series board as well as how to verify whether your board was properly installed. PIO-D48 series can be used on DOS, Linux and 32/64-bit versions of Windows XP/2003/2008/7/8/10 based systems, and the drivers are fully Plug and Play compliant for easy installation.

4.1 Obtaining/Installing the Driver Installer Package

The driver installation package for PIO-D48 series board can be found on the companion CD-ROM, or can be obtained from the ICP DAS FTP web site. Install the appropriate driver for your operating system. The location and website addresses for the installation package are indicated below.

➤ **UniDAQ Driver/SDK** (It is recommended to install this driver for new user.)

OS	Windows 2000, 32/64-bit Windows XP, 32/64-bit Windows 2003, 32/64-bit Windows Vista, 32/64-bit Windows 7, 32/64-bit Windows 2008, 32/64-bit Windows 8, 32/64-bit Windows 10
Driver Name	UniDAQ Driver/SDK (unidaq_win_setup_xxxx.exe)
CD-ROM	CD:\NAPDOS\PCI\UniDAQ\DLL\Driver\
Web Site	http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/dll/driver/
Installing Procedure	<p>To install the UniDAQ driver, follow the procedure described below.</p> <p>Step 1: Double-click the UniDAQ_Win_Setupxxx.exe icon to begin the installation process.</p>

**Installation
Procedure**

Step 2: When the “Welcome to the ICP DAS UniDAQ Driver Setup Wizard” screen is displayed, click the “**N**ext>” button to start the installation.

Step 3: On the “Information” screen, verify that the DAQ board is included in the list of supported devices, then click the “**N**ext>” button.

Step 4: On the “Select Destination Location” screen, click the “**N**ext>” button to install the software in the default folder, **C:\ICPDAS\UniDAQ**.

Step 5: On the “Select Components” screen, verify that the DAQ board is in the list of device, and then click the “**N**ext>” button to continue.

Step 6: On the “Select Additional Tasks” screen, click the “**N**ext>” button to continue.

Step 7: On the “Download Information” screen, click the “**N**ext>” button to continue.

Step 8: Once the installation has completed, click “**No, I will restart my computer later**”, and then click the “**F**inish” button.

For more detailed information about how to install the UniDAQ driver, refer to **Section 2.2 “Install UniDAQ Driver DLL”** of the **UniDAQ Software Manual**, which can be found in the [\NAPDOS\PCI\UniDAQ\Manual](#) folder on the companion CD, or can be downloaded from: <http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/manual/>

- **PIO-DIO Series Classic Driver** (Recommended to install this driver for have been used PIO-DIO series boards of regular user)

OS	Windows 95/98/ME, Windows NT, Windows 2000, 32-bit Windows XP, 32-bit Windows 2003, 32-bit Windows Vista, 32-bit Windows 7, 32-bit Windows 8, , 32-bit Windows 10
Driver Name	PIO-DIO Series Classic Driver(PIO_DIO_Win__vxxx.exe)
CD-ROM	CD:\NAPDOS\PCI\PIO-DIO\DLL_OCX\Driver\
Web Site	http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-dio/dll_ocx/driver/
Installing Procedure	<p>Please follow the following steps to setup software:</p> <p>Step 1: Double click the PIO-DIO Series Classic Driver to setup it.</p> <p>Step 2: When the Setup Wizard screen is displayed, click the Next> button.</p> <p>Step 3: Select the folder where the drivers are to install. The default path is C:\DAQPro\PIO-DIO. But if you wish to install the drivers to a different location , click the "Browse..." button and select the relevant folder and then click the Next> button.</p> <p>Step 4: Click the Install button to continue.</p> <p>Step 5: Select the item "No, I will restart my computer later", press the Finish button.</p> <p>For detailed information about how to install the PIO-DIO Classic Driver, refer to the PIO-DIO Series Classic Driver DLL Software, which can be found in the \NAPDOS\PCI\PIO-DIO\Manual\ folder on the companion CD, or can be downloaded from: http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-dio/manual/</p>

4.2 PnP Driver Installation

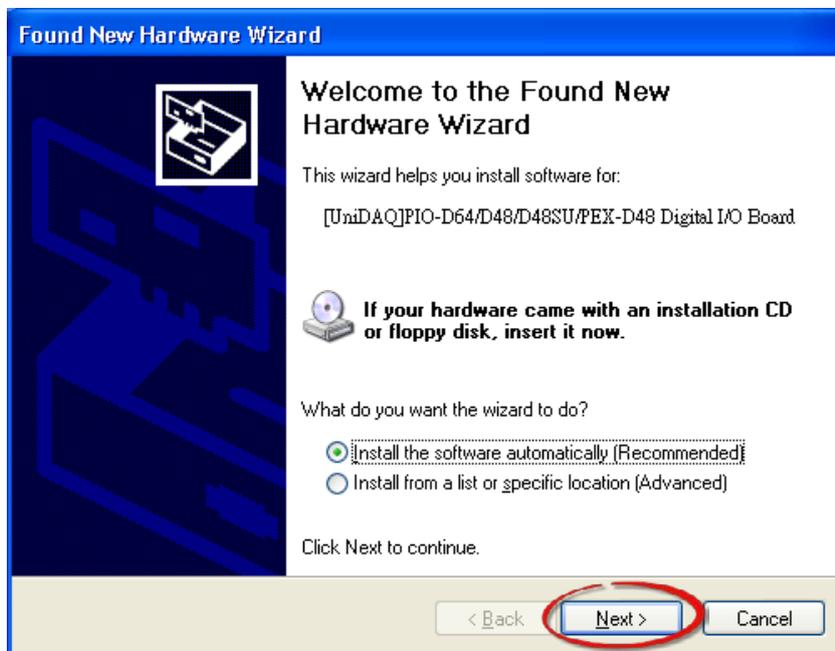
Step 1: Correctly shut down and power off your computer and disconnect the power supply, and then install your board into the computer. For detailed information about the hardware installation of PIO-D48 series board, refer to [Chapter 3 “Hardware Installation”](#).

Step 2: Power on the computer and complete the Plug and Play installation.

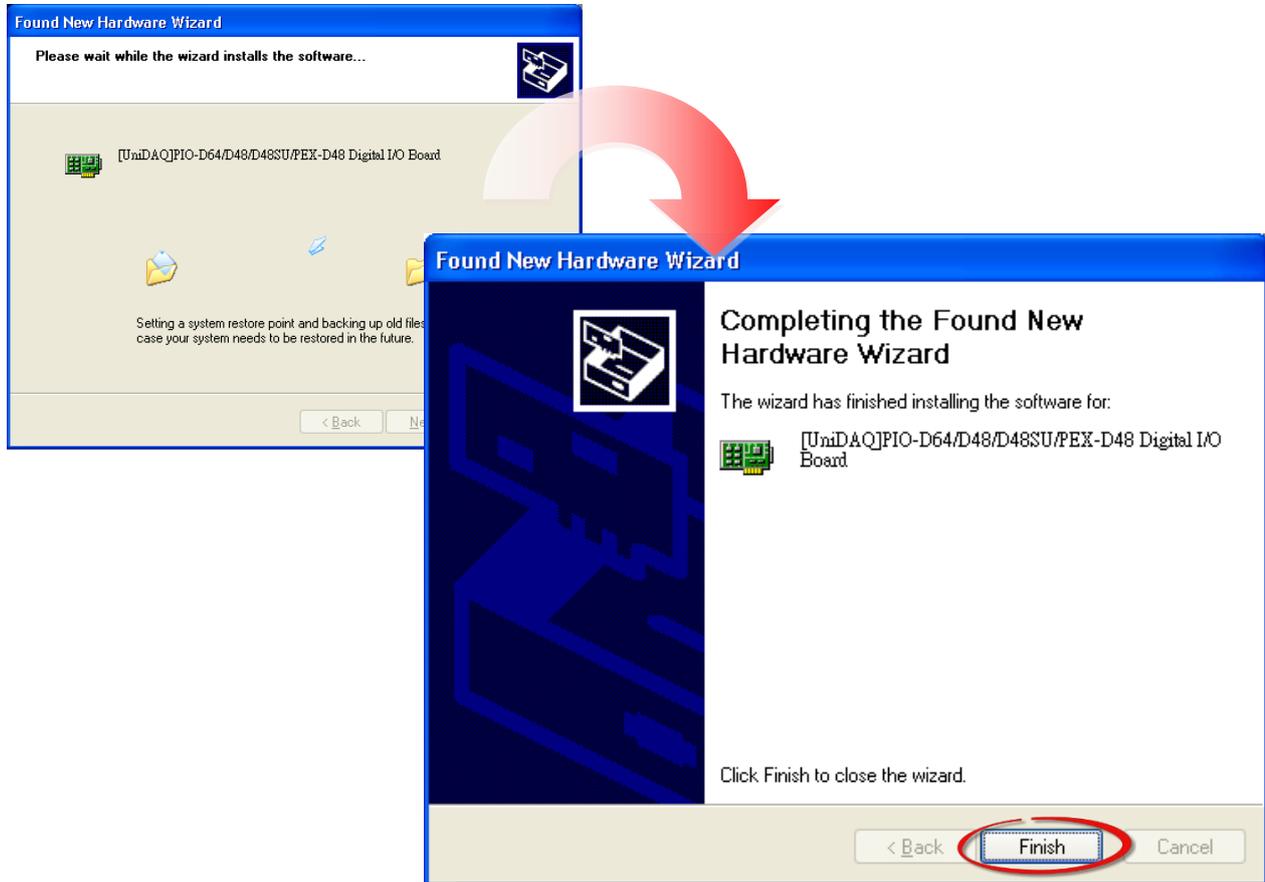
Note:

More recent operating systems, such as Windows 7/8/10 will automatically detect the new hardware and install the necessary drivers etc., so Steps 3 to 5 can be skipped.

Step 3: Select “Install the software automatically [Recommended]” and click the “Next>” button.



Step 4: Click the “Finish” button.



Step 5: Windows pops up “Found New Hardware” dialog box again.



4.3 Verifying the Installation

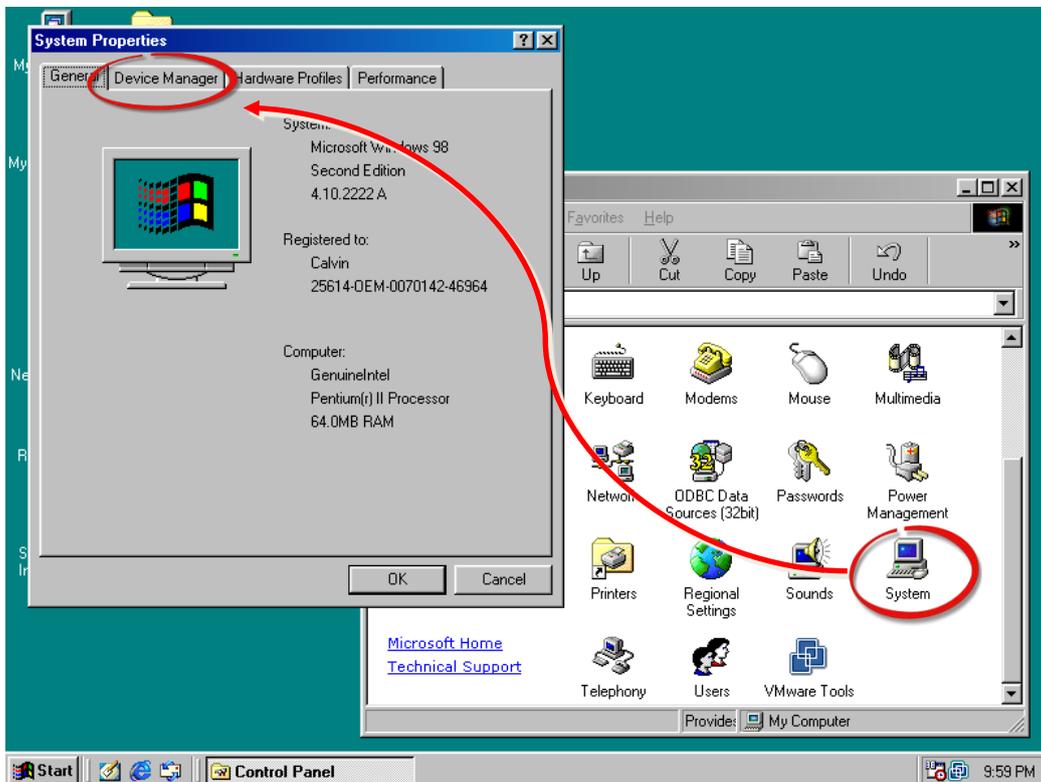
To verify that the driver was correctly installed, use the Windows **Device Manager** to view and update the device drivers installed on the computer, and to ensure that the hardware is operating correctly. The following is a description of how access the Device Manager in each of the major versions of Windows. Refer to the appropriate description for the specific operating system to verify the installation

4.3.1 Accessing Windows Device Manager

➤ **Windows 95/98/ME**

Step 1: Either right-click the **“My Computer”** icon on the desktop and then click **“Properties”**, or open the **“Control Panel”** and double-click the **“System”** icon to open the System Properties dialog box.

Step 2: In the **System Properties** dialog box, click the **“Device Manager”** tab.



➤ **Windows 2000/XP**

Step 1: Click the “**Start**” button and then point to “**Settings**” and click “**Control Panel**”.
Double-click the “**System**” icon to open the “**System Properties**” dialog box.

Step 2: Click the “**Hardware**” tab and then click the “**Device Manager**” button.



➤ **Windows Server 2003**

Step 1: Click the “**Start**” button and point to “**Administrative Tools**”, and then click the “**Computer Management**” option.

Step 2: Expand the “**System Tools**” item in the console tree, and then click “**Device Manager**”.



➤ **Windows 7/10**

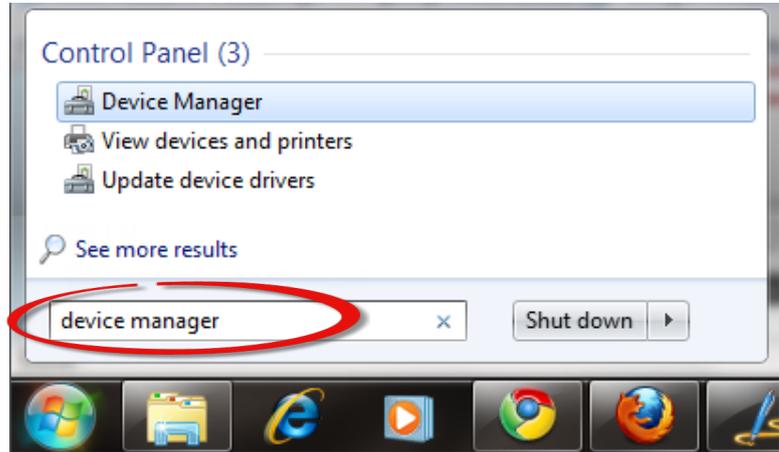
Step 1: Click the “Start” button, and then click “Control Panel”.

Step 2: Click “System and Maintenance”, and then click “Device Manager”.

Alternatively,

Step 1: Click the “Start” button.

Step 2: In the **Search field**, type **Device Manager** and then press Enter.



Note:

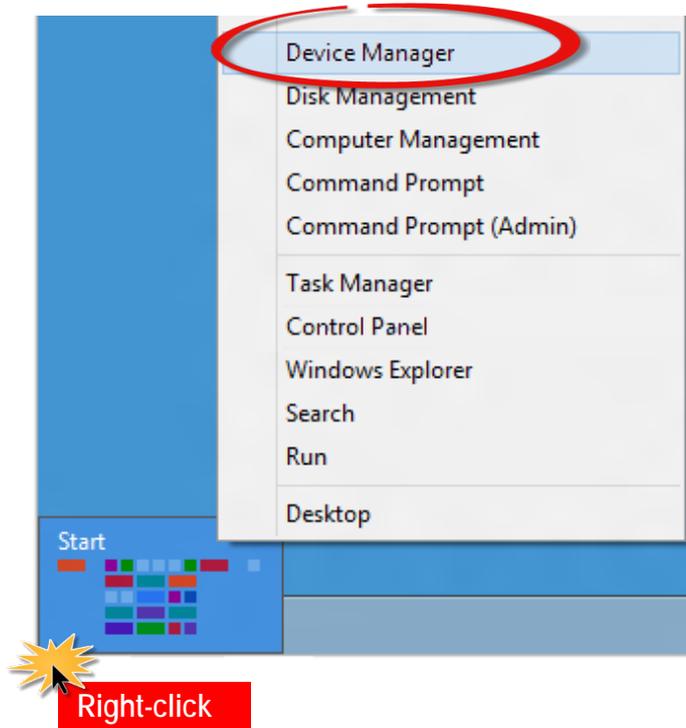
Administrator privileges are required for this operation. If you are prompted for an administrator password or confirmation, enter the password or provide confirmation by clicking the “Yes” button in the User Account Control message.

➤ **Windows 8**

Step 1: To display the **Start screen icon** from the desktop view, hover the mouse cursor over the **bottom-left corner** of screen.

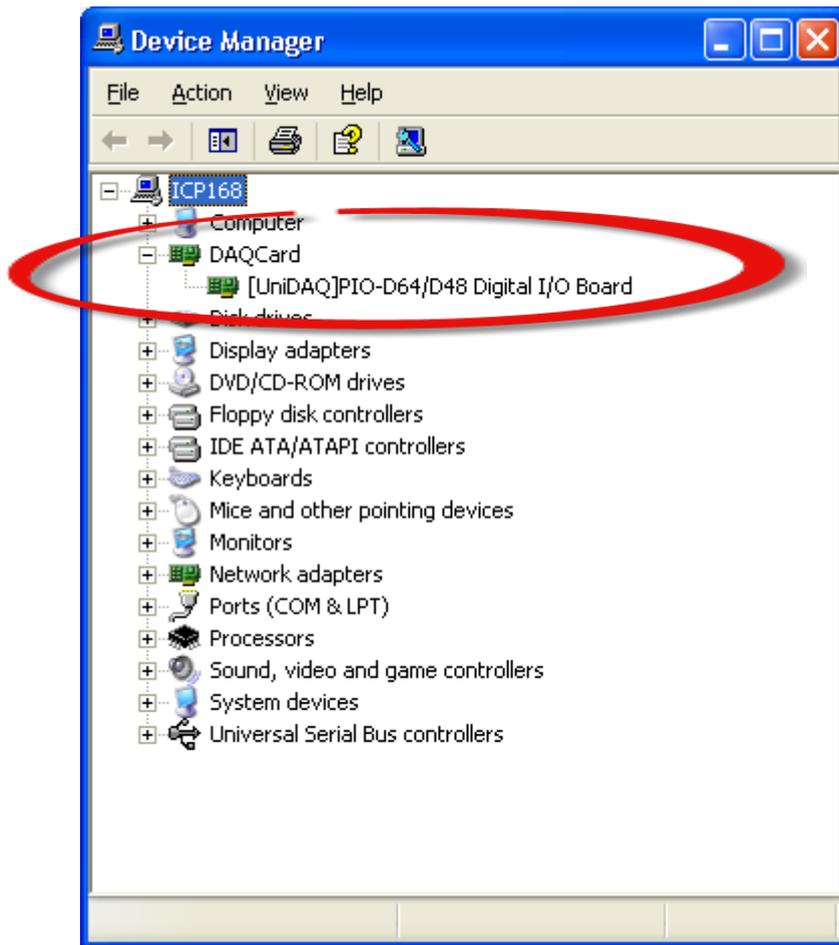
Step 2: **Right-click** the Start screen icon and then click “Device Manager”.

Alternatively, press [**Windows Key**] +[**X**] to open the Start Menu, and then select Device Manager from the options list.



4.3.2 Check that the Installation

Check that the PIO-D48 series board is correctly listed in the **Device Manager** window, as illustrated below.



5. Board Testing

This chapter provides detailed information about the “Self-Test” process, which is used to confirm that the PIO-D48 series board is operating correctly. Before beginning the “Self-Test” process, ensure that both the hardware and driver installation procedures are fully completed. For detailed information about the hardware and driver installation, refer to [Chapter 3 “Hardware Installation”](#) and [Chapter 4 “Software Installation”](#).

5.1 Self-Test Wiring

5.1.1 PIO-D48(U) and PEX-D48

Before beginning the “Self-Test” procedure, ensure that the following items are available:

- A CA-3710 Cable

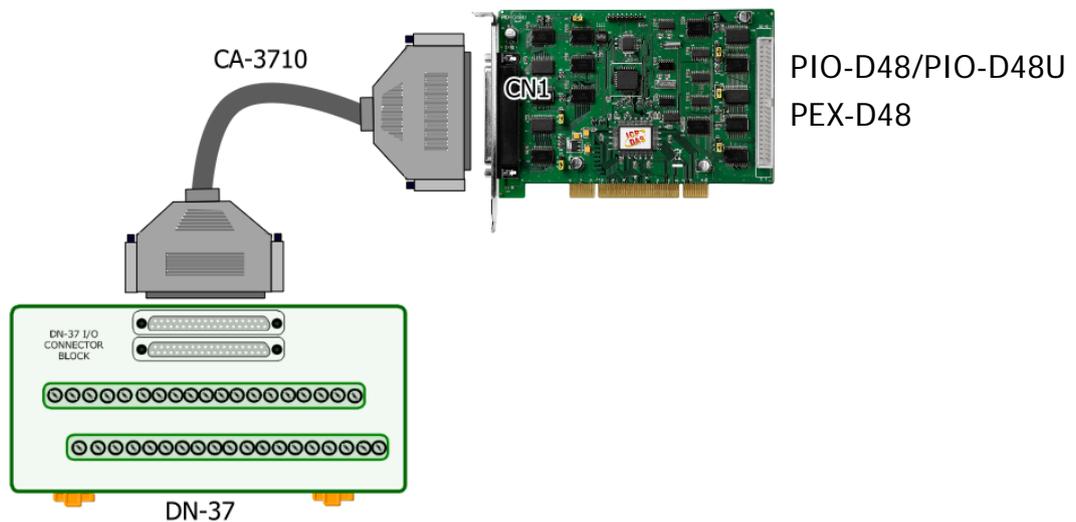
(Optional, Website: http://www.icpdas.com/products/Accessories/cable/cable_selection.htm)

- A DN-37 Terminal Board

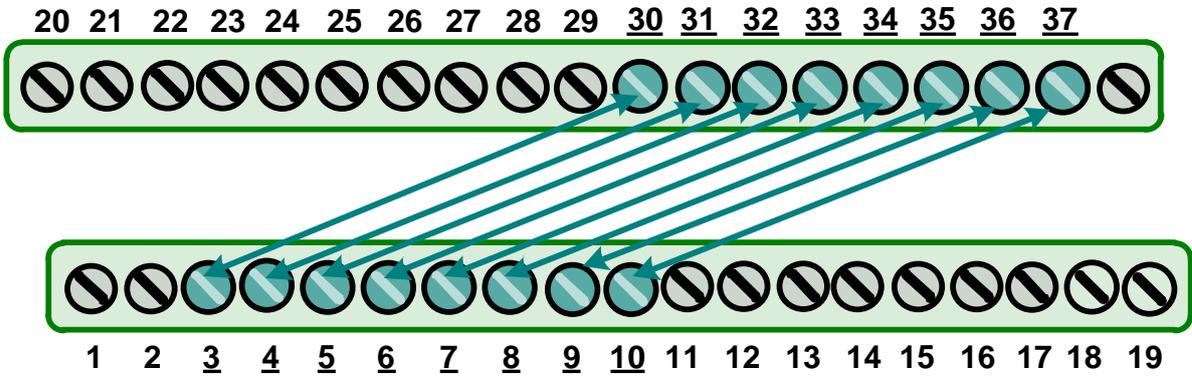
(Optional, Website:

http://www.icpdas.com/root/product/solutions/pc_based_io_board/daughter_boards/dn-37.html)

Step 1: Connect the DN-37 to the CN1 connector on your board using the CA-3710 cable.



Step 2: Connect the Port0 (PA0 to PA7) with Port1 (PB0 to PB7).



5.1.2 PIO-D48SU

Before beginning the “Self-Test” procedure, ensure that the following items are available:

- A CA-SCSI100-15 Cable

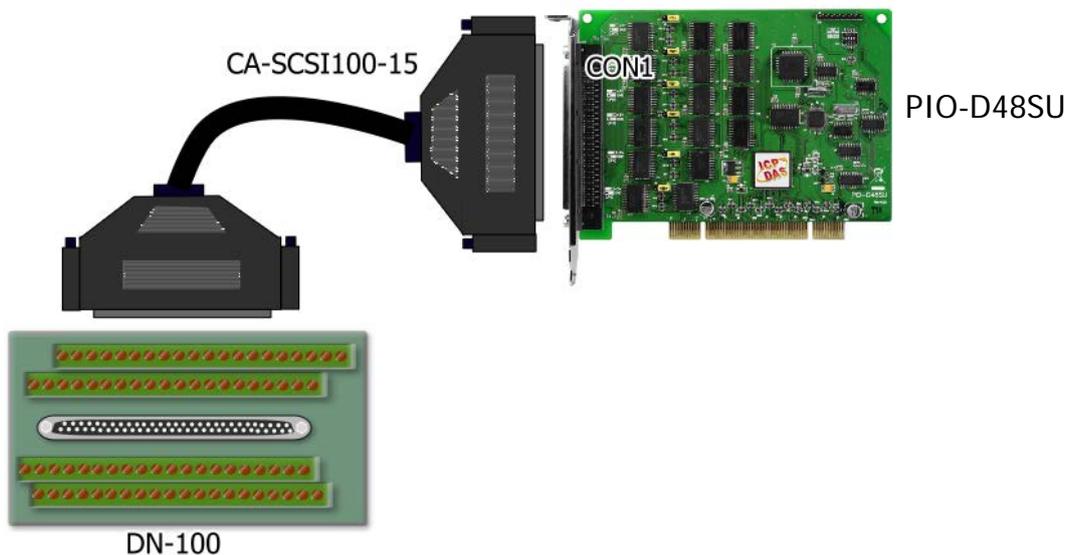
(Optional, Website: http://www.icpdas.com/products/Accessories/cable/cable_selection.htm)

- A DN-100 Terminal Board

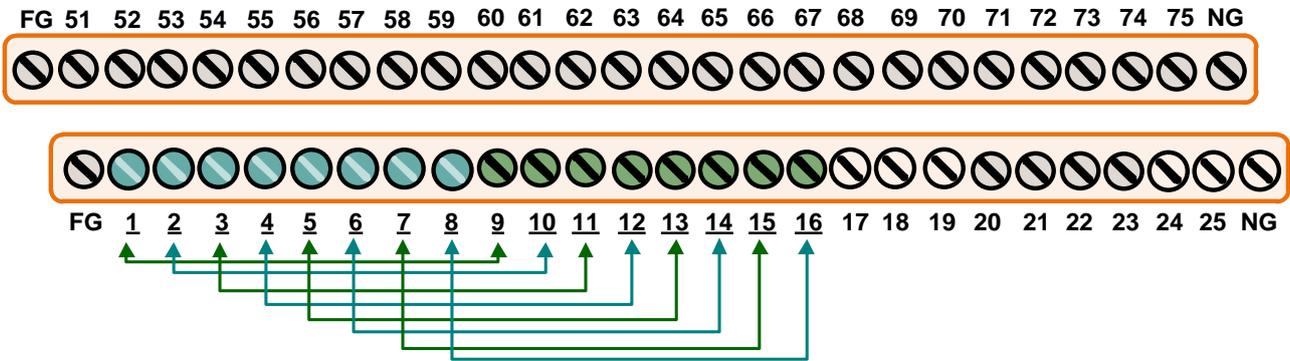
(Optional, Website:

http://www.icpdas.com/root/product/solutions/pc_based_io_board/daughter_boards/dn-100.html)

Step 1: Connect the DN-100 to the CON1 connector on your board using the CA-SCSI100-15 cable.



Step 2: Connect the Port0 (PA00 to PA07) with Port1 (PB00 to PB07).



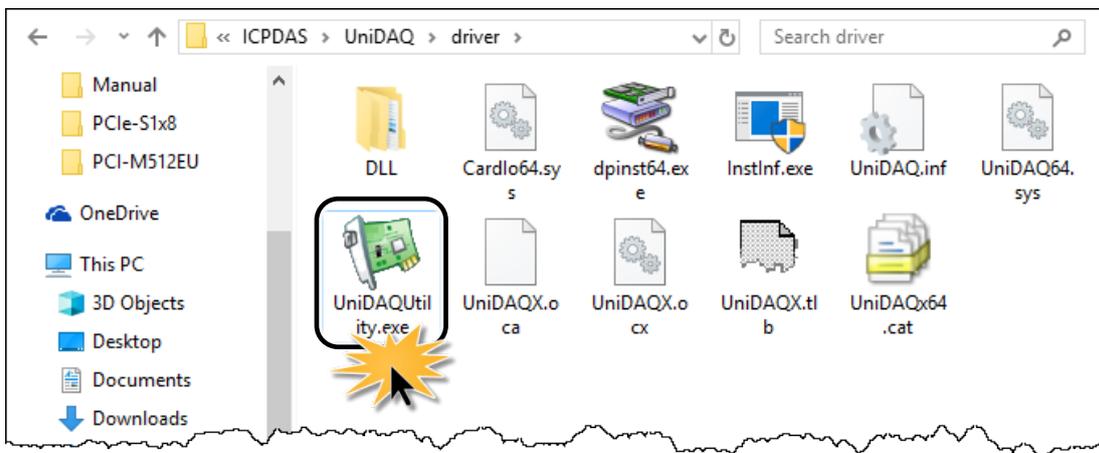
5.2 Launch the Test Program

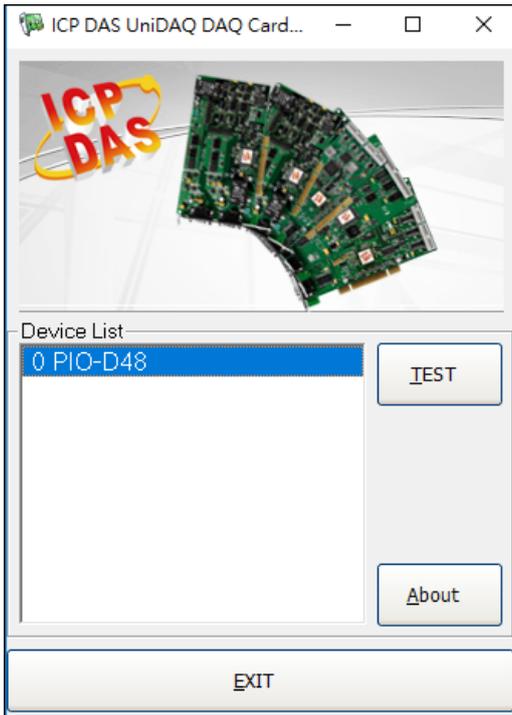
The following example use UniDAQ driver to perform self-test. If you install the PIO-DIO series classic driver, refer to Quick Start Guide of the PIO-D48 series

(<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-dio/manual/quickstart/classic/>)

to execute the self-test.

Step 1: Double-click the **UniDAQ Utility** software. The UniDAQ Utility will be placed in the **default path "C:\ICPDAS\UniDAQ\Driver"** after completing installation.





Step 2: Confirm that your board has been successfully installed in the Host system. **Note that the device number starts from 0.**

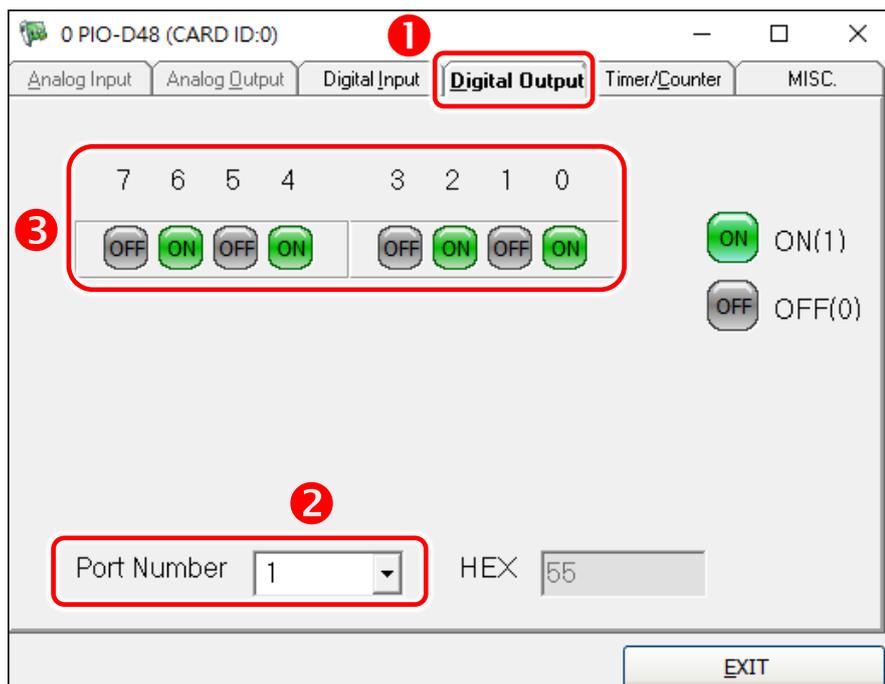
Step 3: Click the “**TEST**” button to start the test.

Note:

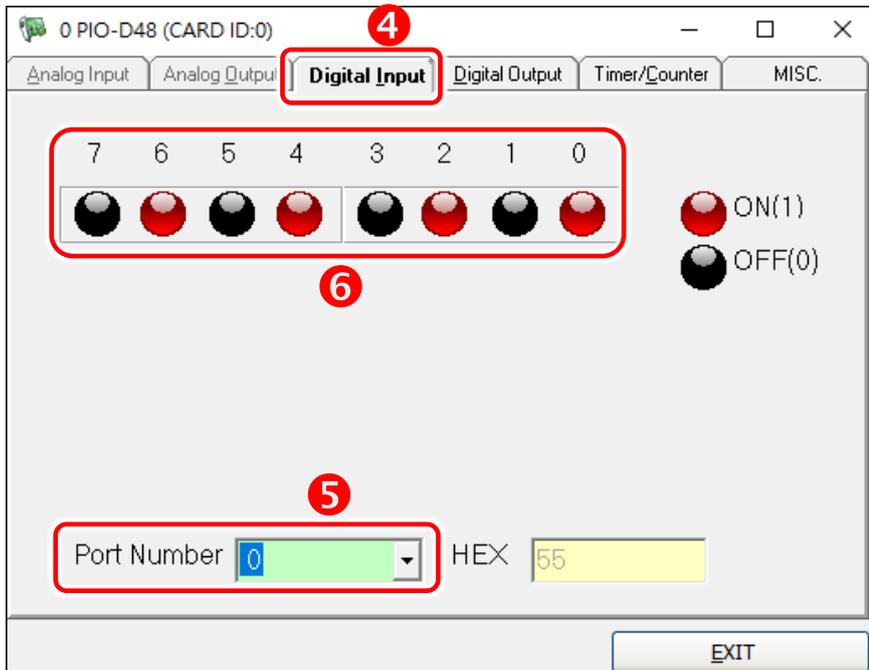
The PEX-D48, PIO-D48SU and PIO-D48U software is fully compatible with the PIO-D48 series software.

Step 4: Check the results of the **Digital Input** and **Digital Output** functions test.

1. Click the “**Digital Output**” tab.
2. Select “**Port 1**” from the “**Port Number**” drop-down menu.
3. Check the checkboxes for **channels 0, 2, 4 and 6**.



4. Click the **“Digital Input”** tab.
5. Select **“Port 0”** from the **“Port Number”** drop-down menu.
6. The DI indicators will turn **red** when the corresponding DO channels 0, 2, 4 and 6 are **ON**.



6. I/O Control Register

6.1 How to Find the I/O Address

During the power-on stage, the Plug and Play BIOS will assign an appropriate I/O address to each PIO-D48 series card installed in the system. Each board includes four fixed ID numbers that are used to identify the board, and are indicated below:

Table 6-1:

PIO-D48(U) (Rev 1.x)		PIO-D48(U)/PEX-D48 (Rev 2.0 or above)	
Vendor ID	0xE159	Vendor ID	0xE159
Device ID	0x0002	Device ID	0x0001
Sub-Vendor ID	0x80	Sub-Vendor ID	0x0080
Sub-Device ID	0x01	Sub-Device ID	0x01
Sub-Aux ID	0x30	Sub-Aux ID	0x30

Table 6-2:

PIO-D48SU (Rev 1.0 or above)	
Vendor ID	0xE159
Device ID	0x0001
Sub-Vendor ID	0x0080
Sub-Device ID	0x0001
Sub-Aux ID	0x0030

We provide all necessary functions as follows:

1. **PIO_DriverInit**(&wBoard, wSubVendor, wSubDevice, wSubAux)
2. **PIO_GetConfigAddressSpace**(wBoardNo,*wBase,*wIrq, *wSubVendor,*wSubDevice, *wSubAux, *wSlotBus, *wSlotDevice)
3. **Show_PIO_PISO**(wSubVendor, wSubDevice, wSubAux)

All functions are defined in PIODIO.H. Refer to [Section 6.3 “The I/O Address Map”](#) for more information. The important driver information is given as follows:

■ **Allocated resource information:**

- **wBase** : BASE address mapping in this PC
- **wIrq**: Allocated IRQ channel number of this board in this PC

■ **PIO/PISO identification information:**

- **wSubVendor**: subVendor ID of this board
- **wSubDevice**: subDevice ID of this board
- **wSubAux**: subAux ID of this board

■ **PC’s physical slot information:**

- **wSlotBus**: The bus number of the slot used by this board.
- **wSlotDevice**: The device number of the slot used by this board.

The PIO_PISO.EXE utility will detect and show all PIO/PISO cards installed in this PC. Refer to [“PIO_PISO.EXE Utility”](#) for more information.

➤ PIO_PISO.EXE Utility

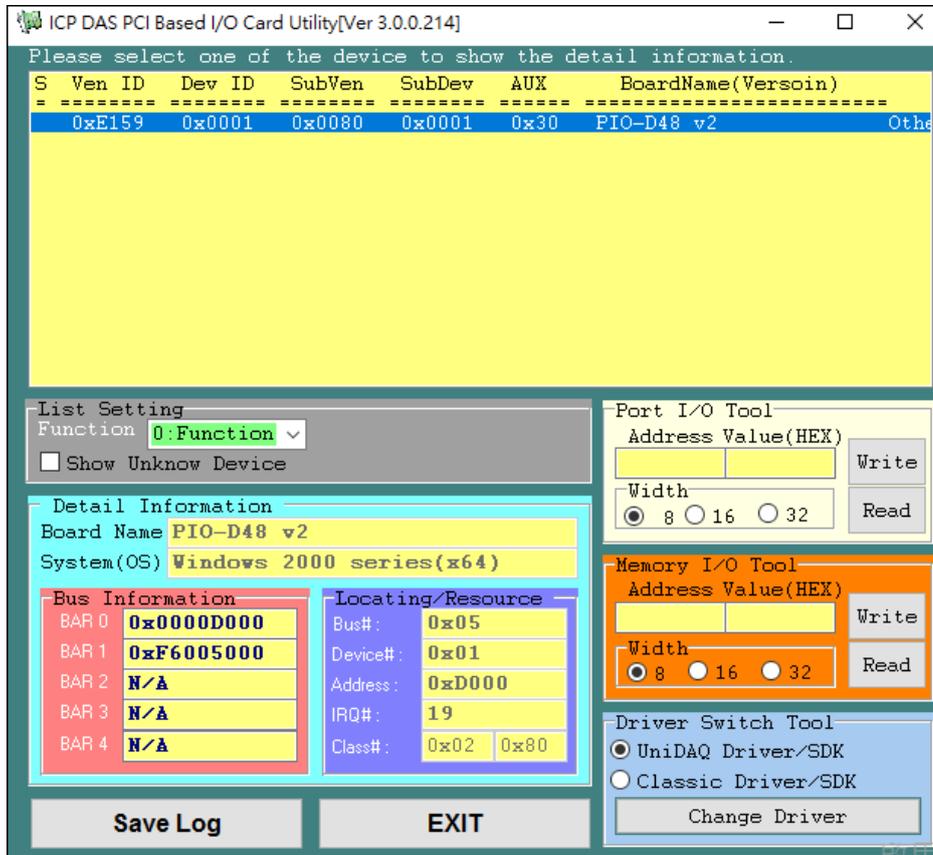
The **PIO_PISO.EXE** utility is valid for all PIO/PISO cards. This program shows all PCI hardware ID regarding the PIO and PISO series DAQ cards. It is useful to test if the card Plug & Play successfully when the computer bootup. If the PIO or PISO series card does not shown in the screen correctly, please try to use another PCI slot and try again.

The user can execute the **PIO_PISO.EXE** to get the following information:

- List all PIO/PISO cards installed in this PC
- List all resources allocated to every PIO/PISO cards
- List the wSlotBus and wSlotDevice for specified PIO/PISO card identification. (refer to [Section 6.2 “The Assignment of I/O Address”](#) for more information about the assignment of I/O Address)

The **PIO_PISO.EXE** utility is located on the CD as below and is useful for all PIO-DIO series boards. (CD:\NAPDOS\PCI\Utility\Win32\PIO_PISO\)

http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/utility/win32/pio_piso/



6.2 The Assignment of I/O Address

Use Card ID for new users.

The Plug and Play BIOS will assign the proper I/O address to a PIO/PISO series card. If there is only one PIO/PISO card, the user can identify the card as card_0. If there are two PIO/PISO cards in the system, it is very difficult to identify which board is card_0. The software driver can support a maximum of 16 cards. Therefore, the user can install 16 PIO/PSIO series cards onto one PC system. The methods used to find and identify card_0 and card_1 is demonstrated below.

The simplest way to identify which card is card_0 is to use wSlotBus and wSlotDevice in the following manner:

- Step 1:** Remove all PIO-D48 series boards from the PC.
- Step 2:** Install one PIO-D48 series onto the PC's PCI_slot1, run PIO_PISO.EXE.
Then record the "wSlotBus1" and "wSlotDevice1" information.
- Step 3:** Remove all PIO-D48 series boards from the PC.
- Step 4:** Install one PIO-D48 series into the PC's PCI_slot2 and run PIO_PISO.EXE.
Then record the "wSlotBus2" and "wSlotDevice2" information.
- Step 5:** Repeat Steps(3) and (4) for every PCI_slot and record all information from "wSlotBus" and "wSlotDevice".

The records may look similar to the table follows:

Table 6-3

PC's PCI Slot	Locating/Resource	
	wSlotBus (Bus#)	wSlotBus (Device#)
Slot_1	0	0x07
Slot_2	0	0x08
Slot_3	0	0x09
Slot_4	0	0x0A
PCI-BRIDGE		
Slot_5	1	0x0A
Slot_6	1	0x08
Slot_7	1	0x09
Slot_8	1	0x07

The above procedure will record all the “wSlotBus” and “wSlotDevice” information on a PC. These values will be mapped to this PC’s physical slot and this mapping will not be changed for any PIO/PISO cards. Therefore, this information can be used to identify the specified PIO/PISO card by following steps:

Step1: Using the “wSlotBus” and “wSlotDevice” information from Table 6-4.

Step2: Enter the board number into PIO_GetConfigAddressSpace(...) function to get the information for a specific card, especially the “wSlotBus” and “wSlotDevice” details.

Step3: Identify the specific PIO/PISO card by comparing the data of the “wSlotBus” and “wSlotDevice” from Step1 and Step2.

Note:

Normally the card installed in slot 0 is card0 and the card installed in slot1 is card1 for PIO/PISO series cards.

6.3 The I/O Address Map

The I/O address of the PIO/PISO series card is automatically assigned by the main board ROM BIOS. The I/O address can also be re-assigned by the user, but it is strongly recommended that the I/O address is not changed by user. The Plug and Play BIOS will assign an appropriate I/O address to each PIO/PISO series card. The I/O addresses of the PIO-D48 series card are as follows, and are based on the base address of each card.

Address	Read	Write
wBase+0	-	RESET\ Control Register
wBase+5	INT mask control register	Same
wBase+7	Aux pin status register	Same
wBase+0x2a	INT polarity control register	Same
wBase+0xc0	Read Port0 (PA)	Write Port0 (PA)
wBase+0xc4	Read Port1 (PB)	Write Port1 (PB)
wBase+0xc8	Read Port2 (PC)	Write Port2 (PC)
wBase+0xcc	-	Write Control Word (Port 0/1/2)
wBase+0xd0	Read Port3 (PA)	Write Port3 (PA)
wBase+0xd4	Read Port4 (PB)	Write Port4 (PB)
wBase+0xd8	Read Port5 (PC)	Write Port5 (PC)
wBase+0xdc	-	Write Control Word (Port 3/4/5)
wBase+0xe0	Read 8254-Counter0	Write 8254-Counter0
wBase+0xe4	Read 8254-Counter1	Write 8254-Counter1
wbase+0xe8	Read 8254-Counter2	Write 8254-Counter2
wBase+0xec	Read 8254 Control Word	Writer 8254 Control Word
wBase+0xf0	Read Clock/Int Control Word	Write Clock/Int Control Word
wBase+0xf4	Read Card ID	-

Note:

Refer to [Section 6.1 "How to Find the I/O Address"](#) for more information about wBase.

6.3.1 RESET\ Control Register

(Read/Write): wBase+0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	RESET\						

When the PC’s power is first turned on, RESET\ signal is in a Low-state. **This will disable all D/I/O operations.** The user has to set the RESET\ signal to a High-state before any D/I/O command applications are initiated.

```

For example:
outputb (wBase,1);      /* RESET\=High → all D/I/O are enable now */
outputb (wBase,0);     /* RESET\=Low → all D/I/O are disable now */
    
```

6.3.2 AUX Status Register

(Read/Write): wBase+7

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AuxID7	AuxID6	AuxID5	AuxID4	IntSate3	IntSate2	IntSate1	IntSate0

IntSate0 = INT_CHAN_0, IntSate1 = INT_CHAN_1, IntSate2 = INT_CHAN_2, IntSate3 = INT_CHAN_3, AuxID7 to 4 = Aux-ID. The IntSate0 to 3 are used as interrupt source. The interrupt service routine has to read this register to identify the interrupt source.

Refer to [Section 2.7 “Interrupt Operation”](#) for more information.

6.3.3 INT Mask Control Register

(Read/Write): wBase+5

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	EN3	EN2	EN1	EN0

EN0=0 → Disable INT_CHAN_0 as an interrupt signal (Default).

EN0=1 → Enable INT_CHAN_0 as an interrupt signal

EN1=0 → disable INT_CHAN_1 as a interrupt signal (Default)

EN1=1 → enable INT_CHAN_1 as a interrupt signal

EN2=0 → disable INT_CHAN_2 as a interrupt signal (Default)

EN2=1 → enable INT_CHAN_2 as a interrupt signal

EN3=0 → disable INT_CHAN_3 as a interrupt signal (Default)

EN3=1 → enable INT_CHAN_3 as a interrupt signal

For example:

```

outportb(wBase+5,0);      /*Disable all interrupt */
outportb(wBase+5,1);      /* Enable interrupt of INT_CHAN_0 */
outportb(wBase+5,2);      /* Enable interrupt of INT_CHAN_1 */
outportb(wBase+5,4);      /* Enable interrupt of INT_CHAN_2 */
outportb(wBase+5,8);      /* Enable interrupt of INT_CHAN_3 */
outportb(wBase+5,0x0f);   /* Enable all four channels of interrupt */
    
```

6.3.4 Interrupt Polarity Control Register

(Read/Write): wBase+0x2A

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	INV3	INV2	INV1	INV0

This register provides a function to control invert or non-invert for the interrupt signal source. A detailed application example is given below.

INV0=0 → select the inverted signal from interrupt_channel_0

INV0=1 → select the non-inverted signal from interrupt_channel_0

INV1= Control interrupt channel_1

INV2= Control interrupt channel_2

INV3= Control interrupt channel_3

For example:

```
/* select the non-inverted input from all 4 channel */
```

```
outportb(wBase+0x2a,0x0f);
```

```
/* select the inverted input from all 4 channel */
```

```
outportb(wBase+0x2a,0x00);
```

```
/* select the inverted input of INT_CHAN_0 */
```

```
/* select the non-inverted input from the others */
```

```
outportb(wBase+0x2a,0x0e);
```

```
/* select the inverted input of INT_CHAN_0& INT_CHAN_1 */
```

```
/* select the non-inverted input from the others */
```

```
outportb(wBase+0x2a,0x0c);
```

Refer to [Section 2.7 “Interrupt Operation”](#) and **DEMO5.C (DOS)** for more information.

6.3.5 Read/Write I/O Port

➤ Control Word

(Read/Write): wBase+0xcc/ 0xdc

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	0	PA	PC-H	0	PB	PC-L

These are six 8-bit I/O ports in the PIO-D48 series. Every I/O port can be programmed to be a DI or a DO port based on the control word settings. All six ports are configured as DI ports when the power is first turned on.

(Read/Write) → wBase+0xcc = Set Input/Output mode for the port-0/1/2

(Read/Write) → wBase+0xdc = Set Input/Output mode for the port-3/4/5

PA/PB/PC-H/PC-L: **1** → Input port; **0** → Output port

PC-H: high nibble of PC

PC-L: Low nibble of PC

➤ Read/Write 8-bit Data

(Read/Write): wBase+0xc0/ 0xc4/ 0xc8/ 0xd0/ 0xd4/ 0xd8

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

(Read/Write)

wBase+0xc0= Port-0 (PA)

wBase+0xc4= Port-1 (PB)

wBase+0xc8= Port-2 (PC)

wBase+0xd0= Port-3 (PA)

wBase+0xd4= Port-4 (PB)

wBase+0xd8= Port-5 (PC)

For example:

```
outportb(wBase+0xcc,0x80);    /* set output mode for the port-0/1/2 */
outportb(wBase+0xc0,V1);     /* write to port_0 (PA) */
outportb(wBase+0xc4,V2);     /* write to port_1 (PB) */
outportb(wBase+0xc8,V3);     /* write to port_2 (PC) */

outportb(wBase+0xdc,0x9B);    /* set input mode for the port-3/4/5 */
V1=inportb(wBase+0xd0);      /* read from port_3 (PA) */
V2=inportb(wBase+0xd4);      /* read from port_4 (PB) */
V3=inportb(wBase+0xd8);      /* read from port_5 (PC) */
```

6.3.6 Read/Write 8254

➤ 8254 Control Word

(Read/Write): wBase+0xec

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

BCD	
0	Binary Count
1	BCD Count

M2	M1	M0	
0	0	0	Interrupt on terminal count
0	0	1	Programmable one-shot
0	1	0	Rate generator
0	1	1	Square-wave generator
1	0	0	software triggered pulse
1	0	1	Hardware triggered pulse

RL1	RL0	
0	0	Counter latch instruction
0	1	Read/write low counter byte only
1	0	Read/write high counter byte only
1	1	Read/write low counter byte first, then high counter

SC1	SC0	
0	0	Counter0
0	1	Counter1
1	0	Counter2
1	1	Readback command

➤ **Read/Write 8-bit data of 8254**

(Read/Write): wBase+0xe0/ 0xe4/ 0xe8

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

(Read/Write): wBase+0xec= 8254 control word

(Read/Write): wBase+0xe0= 8254-counter-0

(Read/Write): wBase+0xe4= 8254-counter-1

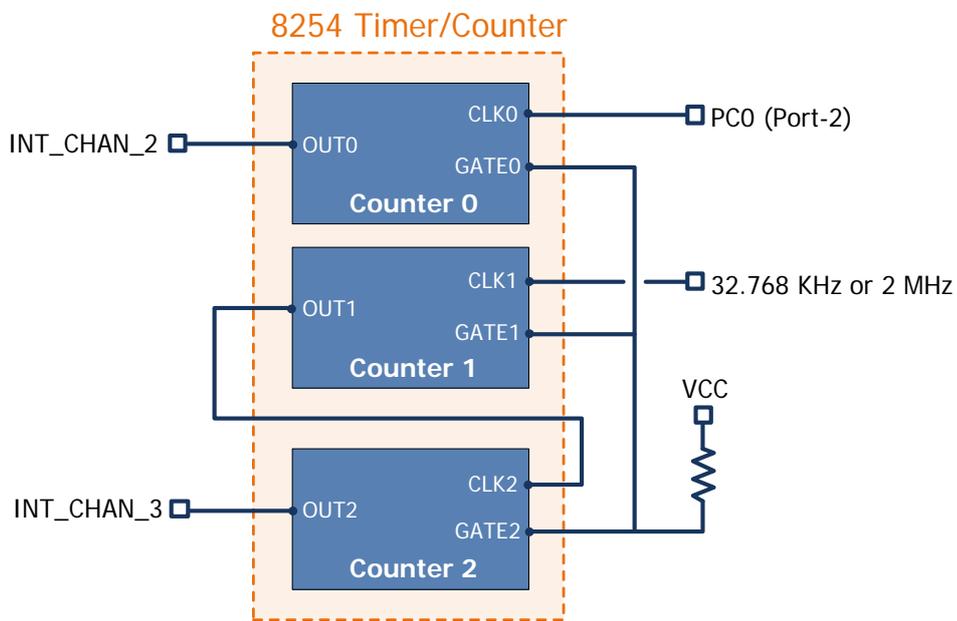
(Read/Write): wBase+0xe8= 8254-counter-2

For example:

```

outputb(wBase+0xec,0x30);    /* Counter0, mode-0 */
outputb(wBase+0xe0,0xff);    /* write to low byte first */
outputb(wBase+0xe0,0xff);    /* write to high byte second */
                                /* Then Counter0 will down count from 0xffff */
    
```

➤ **The configuration of 8254 counter:**



Refer to the following demo programs for more related information:

Int2 Demo	Counter0 (Using interrupt INT_CHAN_2)
Int3 Demo	Counter1 ~ Counter 2 (Using interrupt InT_CHAN_3)

6.3.7 Read/Write Clock/Int Control Register

(Read/Write): wBase+0xf0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	CTRL-D5	CTRL-D4	CTRL-D3	CTRL-D2	CTRL-D1	CTRL-D0

CTRL-D0: timer source CLK1 selection (Refer to [Section 6.3.6 “Read/Write 8254”](#))

0 → 2 MHz

1 → 32.768 KHz

CTRL-D1: invert/non-invert the PC0 of port-2 (Refer to [Section 2.7.4 “INT_CHAN_2”](#))

0 → non-invert

1 → invert

CTRL-D3, CTRL-D2: interrupt source select (Refer to [Section 2.7.2 “INT_CHAN_0”](#))

01 : disable PC3 & !PC7 (of port-2) as interrupt source

10 : INT_CHAN_0=PC3 of port-2

00 : INT_CHAN_0=PC3 & !PC7 of port-2

CTRL-D5, CTRL-D4: interrupt source select (refer to [Section 2.7.3 “INT_CHAN_1”](#))

01 : disable PC3 & !PC7 (of port-5) as interrupt source

10 : INT_CHAN_1=PC3 of port-5

00 : INT_CHAN_1=PC3&!PC7 of port-5

6.3.8 Read Card ID Register

(Read): wBase+0xf4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	ID3	ID2	ID1	ID0

For example:

```
wCardID=inportb(wBase+0xf4); /* Read Card ID */
```

Note: The Card ID function is only supported by the PIO-D48U/D48SU and PEX-D48 (Ver. 1.0 or above)

7. Demo Programs

7.1 Demo Program for Windows

All demo programs will not work properly if the DLL driver has not been installed correctly. During the DLL driver installation process, the install-shields will register the correct kernel driver to the operation system and copy the DLL driver and demo programs to the correct position based on the driver software package you have selected (Win98/Me/NT/2K and 32/64-bit Win XP/2003/7/8/10). Once driver installation is complete, the related demo programs and development library and declaration header files for different development environments will be presented as follows.

➤ Demo Program for PIO-DIO Series Classic Driver

The demo program is contained in:

CD:\NAPDOS\PCI\PIO-DIO\DLL_OCX\Demo\

http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-dio/dll_ocx/demo/

<ul style="list-style-type: none"> ⊕ BCB4 → for Borland C++ Builder 4 PIODIO.H → Header files PIODIO.LIB → Linkage library for BCB only 	<ul style="list-style-type: none"> ⊕ Delphi4 → for Delphi 4 PIODIO.PAS → Declaration files
<ul style="list-style-type: none"> ⊕ VC6 → for Visual C++ 6 PIODIO.H → Header files PIODIO.LIB → Linkage library for VC only 	<ul style="list-style-type: none"> ⊕ VB6 → for Visual Basic 6 PIODIO.BAS → Declaration files
<ul style="list-style-type: none"> ⊕ VB.NET2005 → for VB.NET2005 PIODIO.vb → Visual Basic Source files 	<ul style="list-style-type: none"> ⊕ CSharp2005 → for C#.NET2005 PIODIO.cs → Visual C# Source files

For detailed information about the DLL function of the PIO-D48 series, refer to PIO-DIO DLL Software Manual (CD:\NAPDOS\PCI\PIO-DIOManual\)

➤ Demo Program for UniDAQ SDK Driver

The demo program is contained in:

CD:\NAPDOS\PCI\UniDAQ\DLL\Demo\

<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/dll/demo/>

<ul style="list-style-type: none"> ⊕ BCB6 → for Borland C++ Builder 6 UniDAQ.H → Header files UniDAQ.LIB → Linkage library for BCB only 	<ul style="list-style-type: none"> ⊕ Delphi6 → for Delphi 6 UniDAQ.PAS → Declaration files
<ul style="list-style-type: none"> ⊕ VB6 → for Visual Basic 6 UniDAQ.BAS → Declaration files 	<ul style="list-style-type: none"> ⊕ CSharp2005 → for C#.NET2005 UniDAQ.cs → Visual C# Source files
<ul style="list-style-type: none"> ⊕ VC6 → for Visual C++ 6 UniDAQ.H → Header files UniDAQ.LIB → Linkage library for VC only 	<ul style="list-style-type: none"> ⊕ VB.NET2005 → for VB.NET2005 UniDAQ.vb → Visual Basic Source files
<ul style="list-style-type: none"> ⊕ VC.NET2005 → for VC.NET2005 (32-bit) UniDAQ.H → Header files UniDAQ.LIB → Linkage library for VC only 	<ul style="list-style-type: none"> ⊕ VC.NET2005 → for VC.NET2005 (64-bit) UniDAQ.H → Header files UniDAQ.LIB → Linkage library for VC only

For detailed information about the DLL function and demo program of the UniDAQ, refer to UniDAQ DLL Software Manual (CD:\NAPDOS\PCI\UniDAQ\Manual)

7.2 Demo Program for DOS

The demo program is contained in:

CD:\NAPDOS\PCI\PIO-DIO\DOS\D48\PIOD48\

<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-dio/dos/d48/piod48/>

- ⊕ \TC*. * → for Turbo C 2.xx or above
- ⊕ \MSC*. * → for MSC 5.xx or above
- ⊕ \BC*. * → for BC 3.xx or above

- ⊕ \TC\LIB*. * → for TC Library
- ⊕ \TC\DEMO*. * → for TC demo program
- ⊕ \TC\DIAG*. * → for TC diagnostic program
- ⊕ \TC\LIB\PIO.H → TC Declaration File
- ⊕ \TC\LIB\TCPIO_L.LIB → TC Large Model Library File
- ⊕ \TC\LIB\TCPIO_H.LIB → TC Huge Model Library File

The list of demo programs:

DEMO1: D/O Demo

DEMO2: D/I Demo

DEMO3: D/I/O Demo

DEMO4: Timer Interrupt of INT_CHAN_3

DEMO5: Event Counter (no interrupt) of INT_CHAN_2 (init_HIGH & active_Low)

DEMO6: Event Counter (no interrupt) of INT_CHAN_2 (init_Low & active_HIGH)

DEMO7: Down-Counter (interrupt) of INT_CHAN_2 (init_HIGH & active_Low)

DEMO8: Interrupt demo of INT_CHAN_0 (PC7 of Port2 don't care)

DEMO9: Interrupt demo of INT_CHAN_0 (PC7 of Port2 interrupt is enable)

DEMO10: Interrupt demo of INT_CHAN_1 (PC7 of Port5 interrupt is disable)

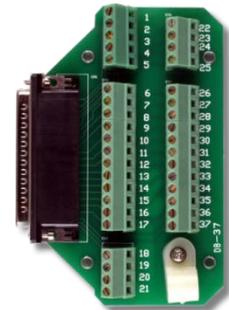
DEMO11: Interrupt demo of INT_CHAN_0 & INT_CHAN_1

For detailed information about the DLL function of the DOS, please refer to PIO-DIO DLL Software Manual (CD:\NAPDOS\PCI\PIO-DIO\Manual)

Appendix: Daughter Board

A1. DB-37, DN-37, DN-50 and DN-100

- **DB-37:** The DB-37 is a general purpose daughter board for D-sub 37 pins. It is designed for easy wire connection via pin-to-pin. Use a 37-pin cable (e.g. CA-3710, etc.) to connect DB-37 to CON1 of the PIO-D48(U)/PEX-D48.



DB-37

- **DN-37 and DN-50:** The DN-37 is a general purpose daughter board for DB-37 pins with DIN-Rail Mountings. The DN-50 is designed for 50-pin flat-cable headers with DIN-Rail mountings. They are also designed for easy wire connection via pin-to-pin.

Use a 37-pin cable (e.g. CA-3710, etc.) to connect to CN1 of the PIO-D48(U) and PEX-D48 by DN-37, and then use a 50-pin cable (e.g. CA-5002, etc.) to connect to CN2 by DN-50.

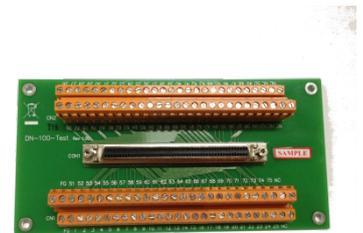


DN-50



DN-37

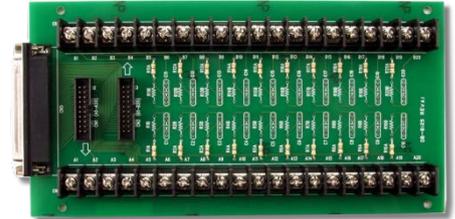
- **DN-100:** The DN-100 is a general purpose daughter board for SCSI II 100 pins. It is designed for easy wire connection via pin-to-pin. Use a 100-pin SCSI II cable (e.g. CA-SCSI100-15 ,etc.) to connect DN-100 to CON1 of the PIO-D48SU.



DN-100

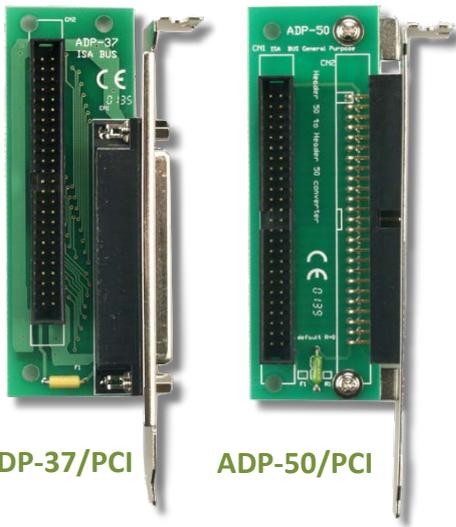
A2. DB-8125

The DB-8125 is a general purpose screw terminal board. It is designed for easy wire connection. The DB-8125 consists of one DB-37 and two 20-pin flat-cable headers. Use a 37-pin cable (e.g. CA-3710, etc.) to connect DB-8125 to CON1 of the PIO-D48(U) and PEX-D48.



DB-8125

A3. ADP-37/PCI and ADP-50/PCI

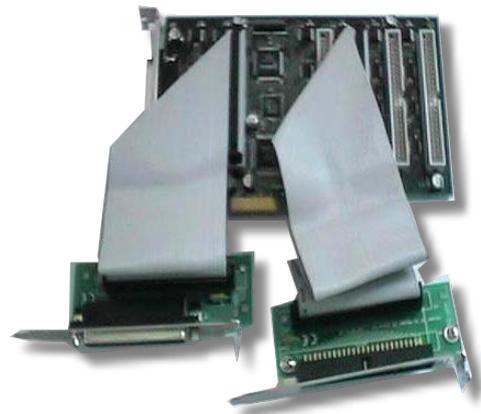


ADP-37/PCI

ADP-50/PCI

The ADP-37/PCI and ADP-50/PCI are extender for the 50-pin header. The one side of the ADP-37/PCI or the ADP-50/PCI can be connected to a 50-pin header. Note: that ADP-37/PCI is 50-pin header to DB-37 extender and ADP-50/PCI is 50-pin header to 50-pin header extender.

The other side can be mounted onto the PC chassis as is depicted by the following:



A4. DB-24P and DB-24PD Isolated Input Board

The DB-24P is a 24-channel isolated digital input daughter board. The optically isolated inputs of the DB-24P consist of a bi-directional optocoupler with a resistor for current sensing. You can use the DB-24P to sense DC signals from TTL levels up to 24 V or use the DB-24P to sense a wide range of AC signals. You can also use this board to isolate the computer from large common-mode voltage, ground loops and transient voltage spikes that often occur in industrial environments, as shown in Figure A4-1. Table A4-1 is the comparison of DB-24P and DB-24PD.

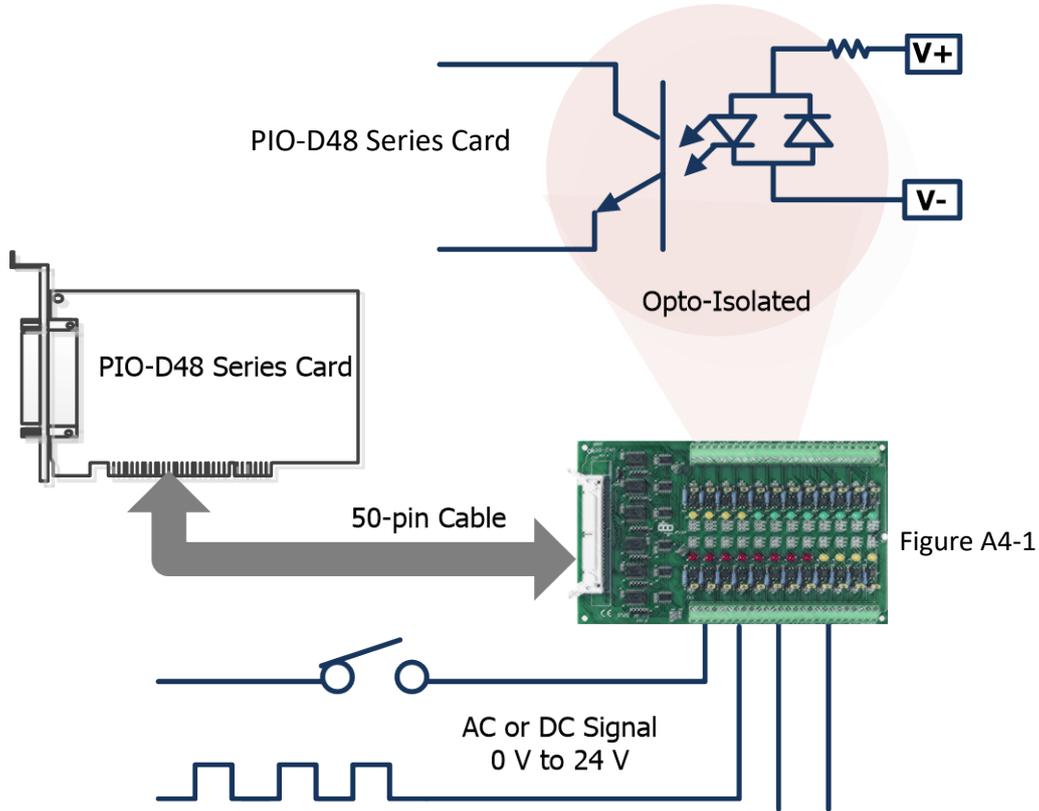
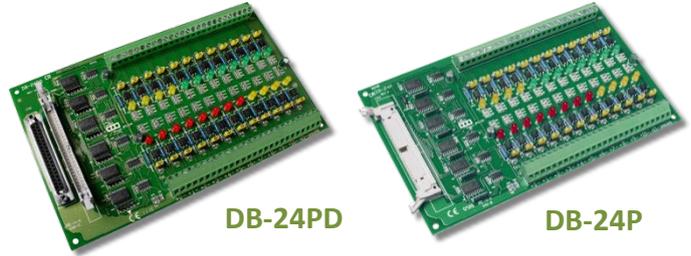


Table A4-1:

	DB-24P	DB-24PD
50-pin Flat-Cable Header	Yes	Yes
D-sub 37-pin Header	No	Yes
Other Specifications	Same	

A5. DB-24R and DB-24RD Relay Board

The DB-24R, 24-channel relay output board, consists of 24 form-C relays for efficiently controlling the switch with the use of an appropriately loaded program. The relays are energized by applying a 12 V/24 V voltage signal to the appropriate relay channel on the 50-pin flat-cable connector. There are 24 enunciator LEDs for each relay channel and the LED light will go on when their associated relay has been activated. The control scheme is illustrated below.

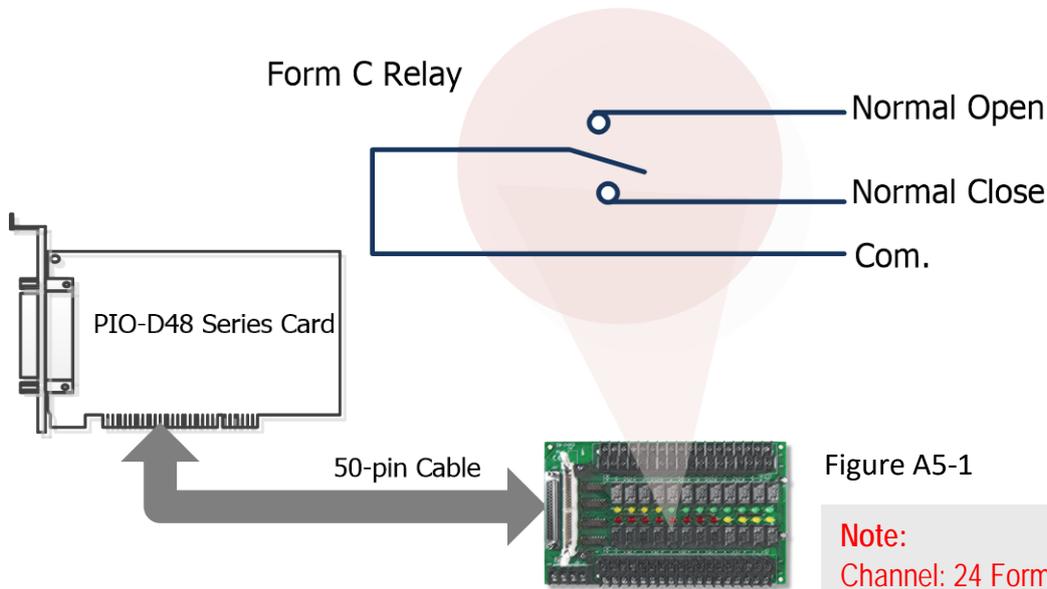
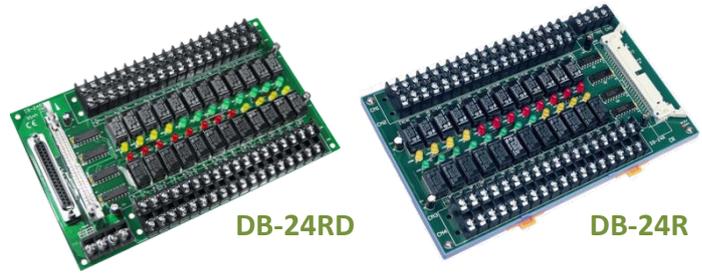


Figure A5-1

Note:
 Channel: 24 Form C Relay
 Relay: Switching up to 0.5 A at 110 V_{AC}
 or 1 A at 24 V_{DC}

Table A5-1:

	DB-24R	DB24RD
50-pin Flat-Cable Header	Yes	Yes
D-sub 37-pin Header	No	Yes
Other Specifications	Same	

Table A5-2:

DB-24R, DB-24RD	24 * Relay (120 V, 0.5 A)
DB-24PR, DB-24PRD	24 * Power Relay (250 V, 5 A)
DB-24POR	24 * PhotoMOS Relay (350 V, 0.1 A)
DB-24SSR	24 * SSR (250 V _{AC} , 4 A)

A6. Daughter Boards Comparison Table

Table A6-1 is the comparison table for the daughter application of PIO/PISO series cards.

Table A6-1:

I/O Card	-	PIO-D48 PIO-D48U PEX-D48	PIO-D48 PIO-D48U PEX-D48	PIO-D48SU
Cable/ Daughter Boards	20-Pin Flat-Cable	50-Pin Flat-Cable	37-Pin D-sub	100-Pin SCSI II
DB-37	NO	NO	Yes	NO
DN-37	NO	NO	Yes	NO
ADP-37/PCI	NO	Yes	Yes	NO
ADP-50/PCI	NO	Yes	NO	NO
DB-24P	NO	Yes	NO	NO
DB-24PD	NO	Yes	Yes	NO
DB-16P8R	NO	Yes	Yes	NO
DB-24R	NO	Yes	NO	NO
DB-24RD	NO	Yes	Yes	NO
DB-24C	Yes	Yes	Yes	NO
DB-24PR	Yes	Yes	NO	NO
DB-24PRD	NO	Yes	Yes	NO
DB-24POR	Yes	Yes	Yes	NO
DB-24SSR	NO	Yes	Yes	NO
DN-100	NO	NO	NO	Yes