EX-94632
32 channel DIO

Printed Aug. 2002 Rev 1.0

Table of Contents

Chapter 1 Introduction

1.1 Introduction

The EX-94632 is 32-CH high-density digital input and/or output product. This I/O card fully implements the PCI local bus specification Rev 2.1. All bus relative configurations, such as base memory and interrupt assignment, are automatically controlled by BIOS software.

1.2 Features

The EX-94632 Isolated digital I/O card provide the following advanced features:

- 32 digital Input or output channels
- Output status read back
- High output driving capability, 25mA sink current on output
- External interrupt signal on DI channels (dual channels)
- 37-pin D-type connector (Pin compatible to EX-9837)(see page 32)

1.3 Applications

- Laboratory and Industrial automation
- Watchdog timer
- Frequency counter and generator
- Low level pulse generator
- Parallel data transfer
- Driving indicator LEDs

1.4 Specifications

- **Optical Isolated Input Channel**

  Numbers of I/O channel: 32 digital I/O lines

  Program mode: Four ports, each port can be programmed to input or output

- **Input Signal**

  Logic high voltage: 2.0 to 5.25 V

  Logic low voltage: 0.0 to 0.80 V

  High level input current: 20 uA

  Low level input current: -0.2 mA

- **Output Signal**

  Logic high voltage: 2.4 V minimums.

  Logic low voltage: 0.4 V maximum

  High level output current: 15 mA maximum (source)

  Low level output current: 24 mA maximum (sink)

  Driving capability: 15 LS TTL

- **Interrupt Sources**

  Channel 0 and channel 8 of digital input

- **General Specifications**

  Connector: 37-pin D-type connector

  Operating temperature: 0°C ~ 60°C

  Storage temperature: -20°C ~ 80°C

  Humidity: 5 ~ 95%, non-condensing

  Power Consumption: +5V 530 mA typical

  Dimension: 170mm(W) x102mm (H)

1.5 Software Supporting

Topsccc provides versatile software drivers and packages for users' different approach to built-up a system. We not only provide programming library such as DLL for many Windows systems, but also provide drivers for many software package such as LabVIEW™ ,Intouch™ and so on. All the software options are included in the provided CD.

1.6 Programming Library

The provided CD includes the function libraries for many different operating systems, including:

- **DOS Library:** BorlandC/C++ and Microsoft C++, the functionsDescriptions are included in this user's guide.
- **Windows 98/2000/NT/Me/XP DLL:** For VB, VC++, BC5, the functionsDescriptions are included in this user's guide.
- Windows 98/2000/NT/Me/XP ActiveX: For Windows's applications
- **LabVIEW ® Driver:** Contains the VIs, which are used to interface with NI's LabVIEW ® software package. Supporting Windows 95/98/NT/2000. The LabVIEW ® drivers are free shipped with the board.
- **InTouch Driver:** Contains the InTouch driver which support the Windows 98/2000/NT/XP. The The InTouch ® drivers are free shipped with the board.

Chapter 2 Installation

This chapter describes how to install the EX-94632 card. Please follow the follow steps to install the EX-94632 card.

2.1 What You Have

In addition to this *User's Manua*l, the package includes the following items:

- EX-94632 board
- Driver/utilities CD
- This user's manual

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 want to ship or store the product in the future

2.2 Unpacking

Your EX-94632 card contains sensitive electronic components that can be easily damaged by static electricity. The operator should be wearing an anti-static wristband, grounded at the same point as the anti-static mat. Inspect the card module carton for obvious damage. Shipping and handling may cause damage to your module. Be sure there are no shipping and handing damages on the module before processing.

After opening the card module carton, extract the system module and place it only on a grounded anti-static surface component side up. Again inspect the module for damage. Press down on all the socketed IC's to make sure that they are properly seated. Do this only with the module place on a firm flat surface.

2.3 Hardware Installation Outline

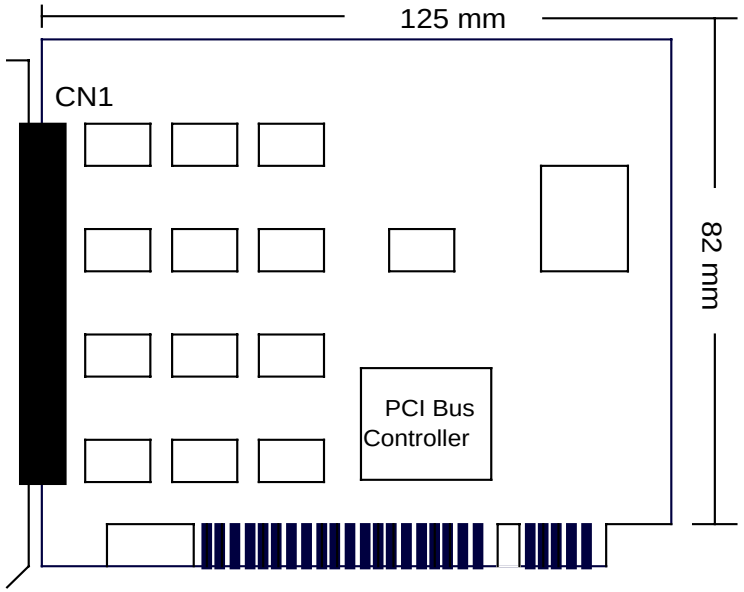
- **PCI configuration**

  The PCI cards are equipped with plug and play PCI controller, it can request base addresses and interrupt according to PCI standard. The system BIOS will install the system resource based on the PCI cards' configuration registers and system parameters (which are set by system BIOS). Interrupt assignment and memory usage (I/O port locations) of the PCI cards can be assigned by system BIOS only. These system resource assignments are done on a board-by-board basis. It is not suggested to assign the system resource by any other methods.

- **PCI slot selection**

  The PCI card can be inserted to any PCI slot without any configuration for system resource.

2.4 PCB Layout



Where

CN1: Digital input/output connector

2.5 Installation Procedures

1. Turn off your computer.
2. Turn off all accessories (printer, modem, monitor, etc.) connected to your computer.
3. Remove the cover from your computer.
4. Setup jumpers on the card.
5. Before handling the PCI cards, discharge any static buildup on your body by touching the metal case of the computer. Hold the edge and do not touch the components.
6. Position the board into the PCI slot you selected.
7. Secure the card in place at the rear panel of the system.

2.6 Device Installation for Windows Systems

Once Windows 95/98/2000 has started, the Plug and Play function of Windows system will find the new Expert cards. If this is the first time to install Expert cards in your Windows system, you will be informed to input the device information source.

2.7 Connector Pin Assignment of EX-94632

The pin assignment of the 37-pins D-type connector is a signal connector, EX-94264's pin assignment is as shown in Figure 2.7

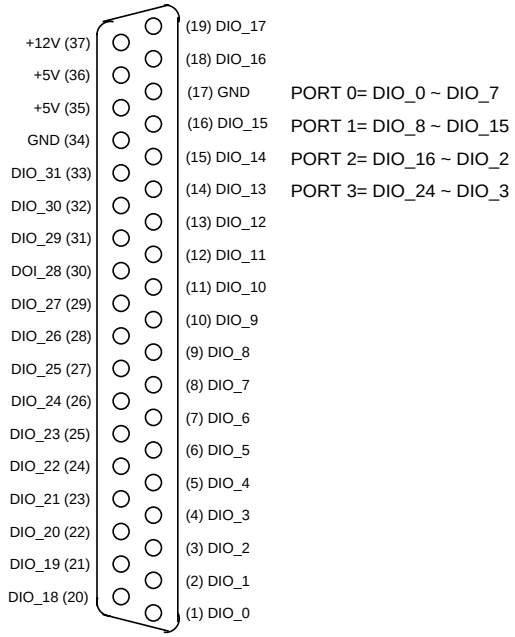


Figure 2.7 Pin Assignment of EX-94632 connector CN1

Legend:

- **DIO_n:** Digital input /output channel #n
- **GND**: Ground r eturn path of input and output channels
- **+5V**: +5VDC output (200 mA max.)
- **+12V**: +12VDC output (100mA max.)

Chapter 3
Registers Format

This information is quite useful for the programmers who wish to handle the card by low-level programming. However, we suggest user have to understand more about the PCI interface then start any low-level programming. In addition, the contents of this chapter can help users understand how to use software driver to manipulate this card.

3.1 PCI PnP Registers

There are two types of registers: PCI Configuration Registers (PCR) and Peripheral Interface Bus (PIB). The PCR, which is compliant to the PCI-bus specifications, is initialized and controlled by the plug & play (PnP) PCI BIOS..

The PCI bus controller Tiger 100/320 is provided by Tigerjet Network Inc. (www.tjnet.com). For more detailed information of PIB, please visit Tigerjet technology's web site to download relative information. It is not necessary for users to understand the details of the PIB if you use the software library. The PCI PnP BIOS assigns the base address of the PIB. The assigned address is located at offset 14h of PIB .

The EX-94264 board registers are in 32-bit width. But only lowest byte (bit0~bit7) is used. The users can access these registers by only 32-bit I/O or 8-bit I/O instructions. The following sections show the address map, including descriptions and their offset addresses relative to the base address.

3.2 Digital Input/Output Register Address Map

There are 32 digital input /output channels on EX-94632, each bit of based address is corresponding to a signal on the digital input or output channel.

3.2.1 PCI controller register address map

- **Reset control register**

  The EX-94632 is in inactive state when the system power on, and should be activated by set bit o of this register to "1" state

  Address: Base + 000H

  Attribute: Write only

  Value: 01

- **Aux port direction control register**

  Address: Base + 002H

  Attribute: Write only

  Value: 7FH

- **Interrupt mask control register**

  Address: Base + 005H

  Attribute: Write only

  Value: 80H =enable PCI INT A#

  00=disable PCI INT #A

3.2.2 Interrupt status register

Address: Base + 0D0H

Attribute: Write/read

Value:

Write 00: Clear interrupts status register

Read: Read interrupt status register

bit 0=1Interrupted by DIO_0

bit 1=1Interrupted by DIO_8

3.2.3 Interrupt and I/O direction control registers

Address: Base + 003H

Attribute: Write only

Value: bit7=0 Always be 0

bit 6=1 / 0Disable / Enable INT from DIO-0 and DIO-8 pins

bit 5=1 / 0Set positive / negative edge interrupt from DIO-8

bit 4=1 / 0Set positive / negative edge interrupt from DIO-0

bit 3=1 / 0Set port 3 to input / output mode

bit 2=1 / 0Set port 2 to input / output mode

bit 1=1 / 0Set port 1 to input / output mode

bit 0=1 / 0Set port 0 to input / output mode

3.2.4 Digital I/O data register

Digital I/O channels of the EX-94632 occupy four data read/write address. each bit of based address is corresponding to a signal on the digital input channel.

Address	Port	Bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Base+0C0H	0	DIO_7	DIO_6	DIO_5	DIO_5	DIO_3	DIO_2	DIO_1	DIO_0
Base+0C4H	1	DIO_15	DIO_14	DIO_13	DIO_12	DIO_11	DIO_10	DIO_9	DIO_8
Base+0C8H	2	DIO_23	DIO_22	DIO_21	DIO_20	DIO_19	DIO_18	DIO_17	DIO_16
Base+0CCH	3	DIO_31	DIO_30	DIO_29	DIO_28	DIO_27	DIO_26	DIO_25	DIO_24

Where

BasePort+0C0H :Data register of Port 0(R/W)

BasePort+0C4H :Data register of Port 1(R/W)

BasePort+0C8H :Data register of Port 2(R/W)

BasePort+0CCH :Data register of Port 3(R/W)

Chapter 4
Operation Theorem

4.1 Digital Input Channels

Each digital input is a TTL structure. The input voltage range form 0V to 5V and input pull-up resister is 10K ohms. The connection between outside signal and EX-94632 digital inputs is shown in Fig 4.1.

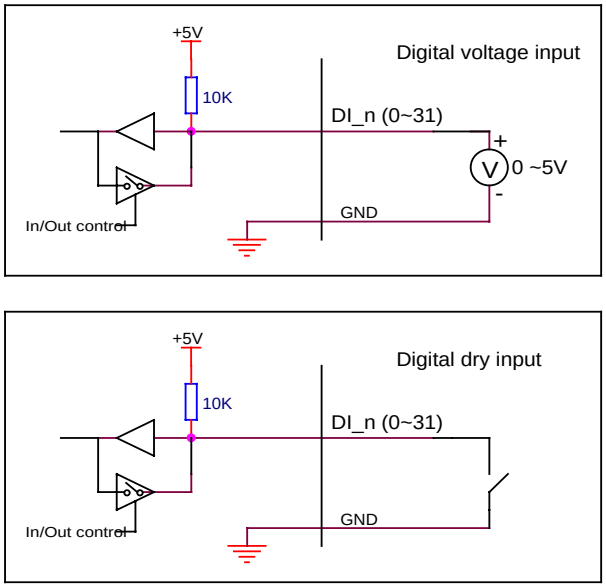


Figure 4.1 digital inputs of EX-94232

4.2 Digital Output Channels

On EX-94632, each port can be programmed to output port by setting Base + 0x03 register (See page 12). Each output channel is TTL compatible with sink current 25mA max. The connection between outside loading and EX-94632 outputs is shown in Fig 4.2

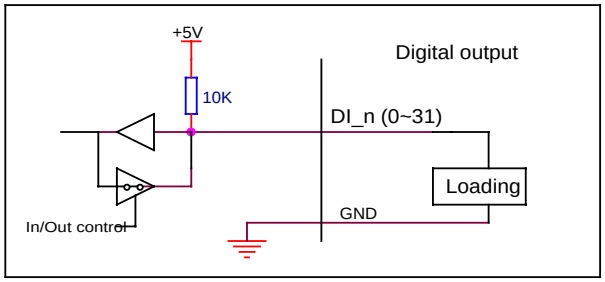


Figure 4.2 digital output of EX-94632

4.3 Edge Change Detection

The ECD (Edge Change Detection) detection circuit is used to detect the edge of level change. In the EX-94632, the detection circuit is applied to 2 input channels (DIO_0 and DIO_8). If channel is programmed to be positive edge or negative edge interrupt mode, the ECD detection circuit generate an interrupt request, when the signal inputs are changed from low to high level or high to low level respectively

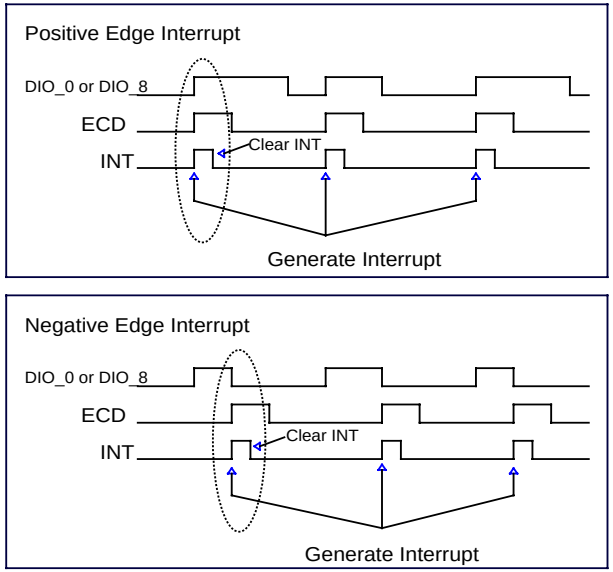


Figure 4.3 debounce block diagram of EX-94632

Chapter 5 Libraries

This chapter describes the software library for operating this card. Only the functions in DOS library and Windows 95 DLL are described. Please refer to the PCIDAQ function reference manual, which included in Topsccc CD for the descriptions of the Windows 98/NT/2000 DLL functions.

5.1 Libraries Installation

The device drivers and DLL functions of Windows 98/NT/2000 are included in the PCIDAQ. The Topsccc CD also includes the detail examples and readme files

5.2 How to use the Functions in PCIDAQ.DLL

- **VC++6.0:**
  - @ Add file '../Include/PCIDAQ.H' in your project
  - @ In link page of menu project| setting, add '../LIB/PCIDAQ.LIB' in the blank of Objects/Library Modules
  - @ Add this sentence "#include '../Include/PCIDAQ.H' " to the head of your main file.
- **Visual BASIC:**
  1. Add file '../Include/Declare.bas' in your project.
- **Delphi:**
  1. Add file '../Include/Declare.pas' in your project
  - @ Add this sentence "uses Declare;" in the head of your unit.pas
- **C++ Builder:**
  1. Add file '../Include/PCIDAQ.H' and '../Lib/PCIDAQ_CB.lib' to your project
  - @ Add this sentence "#include '../Include/PCIDAQ.H' " to head of your main file.

Note: For more information, please refer to program in directory '../Example/'

5.3 Summary of function calls

Function	Description	Page
W_4632_Open	Initial EX-94632 card before using	18
W_4632_Version	Get version number of PCIDAQ.DLL	20
W_4632_GetBusSlot	Get PCI bus and slot number occupied by EX-94632	20
W_4632_Close	Close EX-94632 card before terminating program	22
W_4632_Set_DIOMode	Set port direction (input or output)	23
W_4632_Read_Di	Read digital input port data (8-bit)	24
W_4632_Write_Do	Write data (8-bit) to digital output port	25
W_4632_Read_Do	Read back current value of digital output port	26
W_4632_Set_Do_Bit	Set a bit of port to high	27
W_4632_Reset_Do_Bit	Reset a bit of port to low	28
W_4632_Clear_IntStatus	Clear interrupt status	29
W_4632_IntDisable	Disable digital input interrupt	31
W_4632_IntEnable	Enable interrupt by input(DIO_0~ DIO_8)	30

5.4 W_4632_Open

Description:

Because the EX-94632 is PCI bus architecture and meets the plug and play design, the IRQ and base_address (pass-through address) are assigned by system BIOS directly. EX-94632 cards have to be initialized by this function before calling other functions.

Syntax:

C/C++(Dos)

```
WORD D_4232_Open (WORD cardNo);
```

C/C++ (Windows)

```
WORD W_4632_Open (WORD *ExistedCards);
```

Visual BASIC (Windows)

```
Function W_4632_Open (ByRef ExistedCards As Long) As Long
```

Delphi

```
Function W_4632_Open (var ExistedCards:Integer):Integer;
```

Argument:

CardNo: card number (1,2,3,4) (for DOS only)

existCards: The number of installed EX-94632 cards. (for Windows only)

This return value shows how many EX-94632 cards are installed in your system.

Return Code:

Error code (Please refer to PCIDAQ.H)

5.5 W_4632_Version

Description:

PCIDAQ.DLL driver drives the EX-94632. This function returns the version of PCIDAQ.DLL driver

Syntax:

C/C++(DOS)

```
void D_4232_Version(char *sversion);
```

C/C++ (Windows)

```
Int W_4632_Version (void);
```

Visual BASIC (Windows)

```
W_4632_Version () As Long
```

Delphi

```
W_4632_Version ():Integer;
```

Argument:

sversion: Return the PCIDAQ.DLL driver version string (DOS only)

Return Code:

The version of PCIDAQ.DLL in integer data format (Windows only)

5.6 W_4632_GetBusSlot

Description:

Get the PCI bus and slot number of the card

Syntax:

C/C++(DOS)

```
WORD D_4232_GetBusSlot (WORD cardNo, WORD *bus,WORD *slot);
```

C/C++ (Windows)

```
WORD W_4632_GetBusSlot (WORDcardNo, WORD *bus,WORD *slot);
```

Visual BASIC (Windows)

```
Function W_4632_GetBusSlot (ByValcardNo As Long,
         ByRef bus As Long, ByRef slot As Long) As Long
```

Delphi

```
Function W_4632_GetBusSlot (cardNo:Integer;var bus:Integer;
         var slot:Integer):Integer;
```

Argument:

cardNo: Card number (1,2,3,4)

bus: Return PCI bus Number

slot : Return PCI slot Number of the bus

Return Code:

Error code (Please refer to PCIDAQ.H)

5.7 W_4632_Close

Description:

The IRQ and base_address of EX-94632 (pass-through address) are assigned by system BIOS directly. This function should be called to release all system resource before terminate application program

Syntax:

C/C++(Dos)

```
WORD D_4232_Close (WORD cardNo);
```

C/C++ (Windows)

```
Void W_4632_Close (void);
```

Visual BASIC (Windows)

```
Function W_4632_Close ()
```

Delphi

```
Function W_4632_Close ();
```

Argument:

cardNo: Card number (1,2,3,4)

Return Code:

None

5.8 W_4632_Set_DIOMode

Description:

Set port #0~port #3 is output port or input port

Syntax:

C/C++ (DOS)

```
WORD D_4632_Set_DIOMode (WORDcardNo, BYTE DIO_Direction);
```

C/C++ (Windows)

```
WORD W_4632_Set_DIOMode (WORDcardNo,BYTE DIO_Direction);
```

Visual BASIC (Windows)

```
Function W_4632_Set_DIOMode (ByValcardNo As Long,
         ByVal DIO_Direction As Byte) As Long
```

Delphi

```
Function W_4632_Set_DIOMode (cardNo:Integer;
         DIO_Direction:Integer):Integer;
```

Argument:

cardNo: Card number to select borad (1,2,3,4),It's set by jumper on card

DIO_Direction: Set Port0 to Port3 is Input or output

Bit0=1/0port0 input/output mode

Bit1=1/0port1 input/output mode

Bit2=1/0port2 input/output mode

Bit3=1/0port3 input/output mode

Return Code:

Error code (Please refer to PCIDAQ.H)

5.9 W_4632_Read_Di

Description:

This function is used to read data from digital input port. You can get 8-bit input data from EX-94632 by calling this function.

Syntax:

C/C++ (DOS)

```
WORD D_4632_Read_Di (WORDcardNo,WORDportNo,WORD *DiData);
```

C/C++ (Windows)

```
WORD W_4632_Read_Di (WORDcardNo,WORDportNo,WORD *DiData);
```

Visual BASIC (Windows)

```
Function W_4632_Read_Di (ByValcardNo As Long,
         ByValportNo As Long, ByRef DiData As Long) As Long
```

Delphi

```
Function W_4632_Read_Di (cardNo:Integer;portNo:Integer;
         var DiData:Integer):Integer;
```

Argument:

cardNo: Card number (1,2,3,4)

portNo: Digital Input port number (0 ~ 3)

Didata: Return digital input data

Return Code:

Error code (Please refer to PCIDAQ.H)

5.10 W_4632_Write_Do

Description:

This function is used to write data to output port. You can send 8-bit output data to EX-94632 by calling this function.

Syntax:

C/C++ (DOS)

```
WORD D_4632_Write_Do (WORDcardNo,WORDportNo,WORD Data);
```

C/C++ (Windows)

```
WORD W_4632_Write_Do (WORDcardNo,WORDportNo,WORD Data);
```

Visual BASIC (Windows)

```
Function W_4632_Write_Do (ByValcardNo As Long, ByValportNo As
         Long, ByVal Data As Long) As Long
```

Delphi

```
Function W_4632_Write_Do (cardNo:Integer;portNo:Integer;
         Data:Integer):Integer;
```

Argument:

cardNo: Card number (1,2,3,4)

portNo: Do port number (0 ~ 3)

Data: Data be written to output port

Return Code:

Error code (Please refer to PCIDAQ.H)

5.11 W_4632_Read_Do

Description:

This function is used to read current data of output port. You can read back 8-bit output data of EX-94632 by calling this function.

Syntax:

C/C++ (DOS)

```
WORD D_4632_Read_Do (WORDcardNo,WORDportNo,WORD *DoData);
```

C/C++ (Windows)

```
WORD W_4632_Read_Do (WORDcardNo,WORDportNo,WORD *DoData);
```

Visual BASIC (Windows)

```
Function W_4632_Read_Do (ByValcardNo As Long,
         ByValportNo As Long, ByRef DoData As Long) As Long
```

Delphi

```
Function W_4632_Read_Do(cardNo:Integer;portNo:Integer;
         ar DoData:Integer):Integer;
```

Argument:

cardNo: Card number (1,2,3,4)

portNo: Do port number (0 ~ 3)

Data: Return current output data

Return Code:

Error code (Please refer to PCIDAQ.H)

5.12 W_4632_Set_Do_Bit

Description:

Set a Do channel High

Syntax:

C/C++ (DOS)

```
WORD D_4632_Set_Do_Bit (WORDcardNo,WORDportNo, WORDbitNo);
```

C/C++ (Windows)

```
WORD W_4632_Set_Do_Bit (WORDcardNo,WORDportNo, WORDbitNo);
```

Visual BASIC (Windows)

```
Function W_4632_Set_Do_Bit (ByValcardNo As Long,
         ByValportNo As Long, ByValbitNo As Long) As Long
```

Delphi

```
Function W_4632_Set_Do_Bit (cardNo:Integer;portNo:Integer;
         bitNo:Integer):Integer;
```

Argument:

cardNo: Card number (1,2,3,4)

portNo: Doport number (0 ~ 3)

bitNo: Channel Number(0 to 7)

Return Code:

Error code (Please refer to PCIDAQ.H)

5.13 W_4632_Reset_Do_Bit

Description:

Set a Do channel Low

Syntax:

C/C++ (DOS)

```
WORD D_4632_Reset_Do_Bit (WORDcardNo,WORDportNo, WORDbitNo);
```

C/C++ (Windows)

```
WORD W_4632_Reset_Do_Bit (WORDcardNo,WORDportNo, WORDbitNo);
```

Visual BASIC (Windows)

```
Function W_4632_Reset_Do_Bit (ByValcardNo As Long,
         ByValportNo As Long, ByValbitNo As Long) As Long
```

Delphi

```
Function W_4632_Reset_Do_Bit (cardNo:Integer; portNo:Integer;
         bitNo:Integer):Integer;
```

Argument:

cardNo: Card number (1,2,3,4)

portNo: Do port number (0 ~ 3)

bitNo: Channel Number(0 to 7)

Return Code:

Error code (Please refer to PCIDAQ.H)

5.14 W_4632_Clear_IntStatus

Description:

Clear interrupt by writing random data to Base Port+0xD0

Syntax:

C/C++ (DOS)

```
WORD D_4632_Clear_IntStatus (WORDcardNo);
```

C/C++ (Windows)

```
WORD W_4632_Clear_IntStatus (WORDcardNo);
```

Visual BASIC (Windows)

```
Function W_4632_Clear_IntStatus (ByValcardNo As Long) As Long
```

Delphi

```
Function W_4632_Clear_IntStatus (cardNo:Integer):Integer;
```

Argument:

cardNo: Card number (1,2,3,4)

Return Code:

Error code (Please refer to PCIDAQ.H)

5.15 W_4632_IntEnable

Description:

Enable Interrupt of channel0 of input port0 or chanel0 of input port1

Syntax:

C/C++ (DOS)

```
WORD D_4632_IntEnable (WORDcardNo,BYTE
     IntMode,User_Interrupt_HANDLER userIntServiceRoutine);
```

C/C++ (Windows)

```
WORD W_4632_IntEnable (WORDcardNo,BYTE
     IntMode,User_Interrupt_HANDLER userIntServiceRoutine);
```

Visual BASIC (Windows)

```
Function W_4632_IntEnable (ByValcardNo As Long, ByVal IntMode
         As Byte, ByVal userIntServiceRoutine As Long) As Long
```

Delphi

```
Function W_4632_IntEnable (cardNo:Integer;IntMode:Integer;
         userIntServiceRoutine:Pointer):Integer;
```

Argument:

cardNo: Card number (1,2,3,4)

IntMode: Interrupt mode of channel0 of input port0 and port1

bit0=1/0:Pos/Neg edge Interrupt from DIO-0

bit1=1/0:Pos/Neg edge Interrupt from DIO-8

bit2=1/0:Disable/Enable int from DIO-0 and DIO-8 pins

userIntServiceRoutine: user Interrupt serice routine called when interrupt occurs.

Return Code:

Error code (Please refer to PCIDAQ.H)

5.16 W_4632_IntDisable

Description:

Disable interrupt of channel0 of input port0 and chanel0 of input port1

Syntax:

C/C++ (DOS)

```
void D_4632_IntDisable (WORDcardNo);
```

C/C++ (Windows)

```
void W_4632_IntDisable (WORDcardNo);
```

Visual BASIC (Windows)

```
Function W_4632_IntDisable (ByValcardNo As Long)
```

Delphi

```
Function W_4632_IntDisable (cardNo:Integer);
```

Argument:

cardNo: Card number (1,2,3,4)

Return Code:

Error code (Please refer to PCIDAQ.H)

Chapter 6
EX-9837 Terminal board

EX-9837 Screw-terminal termination board features one 37-pin D-type connector for easy maintenance, wiring, and installation. It provides 37 channels that are accessed through a 37-pin D-type connector.

6.1 Main features

- Low-cost screw-terminal board for the all Expert series with 37-pin D-type connector
- Reserved space for signal-conditioning circuits such as low-pass filter, voltage attenuator and current shunt
- Industrial type termination blocks permit heavy-duty and reliable signal connections
- Table-top mounting using nylon standoffs. Screws and washers provided for panel or wall mounting
- Dimensions: 80mm (W) x 181mm (H)