EX-94288
16-channel isolated Digital Input and 16-channel relay output

Printed Sep. 2002 Rev 1.0

Table of Contents

Chapter 1 Introduction

1.1 Introduction

The EX-94288 is a 16-ch relay actuator and 16-ch isolated digital input card for the PCI bus. Its sixteen on-board SPDT relays are ideal for applications such as device ON/OFF control or small power switched. For easy monitoring, each relay is equipped with one red LED to show its ON/OFF status.

The EX-94288's sixteen optically-isolated digital input channels are ideal for digital input in noisy environments or with floating potentials.

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-94288 isolated digital I/O card provide the following advanced features:

- 16 relay output channels and 16 isolated digital input channels
- LED indicators to show activated relays
- Output status read-back
- Keep relay output values when hot system software reset
- High-voltage isolation on input channels (2,500 V_{DC})
- On-board digital filter circuit
- High over-voltage protection (70 V_{DC})
- Wide input range (0 ~ 30 V_{DC})
- Interrupt handling capability
- 37-pin D-type connector
- Board card number

1.3 Applications

- Laboratory and Industrial automation
- Industrial ON/OFF control
- Switch status sensing
- PC-based Industrial Machinery
- Testing & Measurement
- Laboratory & Education
- External relay driving

1.4 Specifications

- **Relay Outputs**

  Contact Rating: 0.3A/120V AC,DC/1A 30V DC

  Contact arrangement: Output channel 0 -3 are form C

  Output channel 4-7 are Form A

  Operating time: 5 mSec

  Release time: 10 mSec

  Expected life > 100,000 times (at 30V, 1A)

- **Isolated Inputs**

  Type: opto-isolated (PC-814)

  Input voltage: DC 5-24V (0- 1KHz)

  Input impedance: 4.7K ohm (or optional 47K ohms for +24Vdc input range)

  Response time: 20u Sec

  Isolated: 500v channel-channel / channel - Ground

  Interrupt: Pos. or Neg. edge interrupt programmable (Ch #0,Ch #8)

- **Software:**

  Software Toolkit (DLL, ACtiveX) for Windows 98/2000/NT/XP

  Software Toolkit for Labview, Intouch

- **Environmental:**

  Power requirements: +5V 600mA (typical)

  Operation Temp: 0 to 50C

  Storage Temp: -20 to 70C

  Humidity: 0 to 90% none-condensing

  Dimensions: 180 X 105 mm

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:** Borland C/C++ and Microsoft C++, the functions descriptions 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-94288 card. Please follow the follow steps to install the EX-94288 card.

2.1 What You Have

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

- EX-94288 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-94288 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

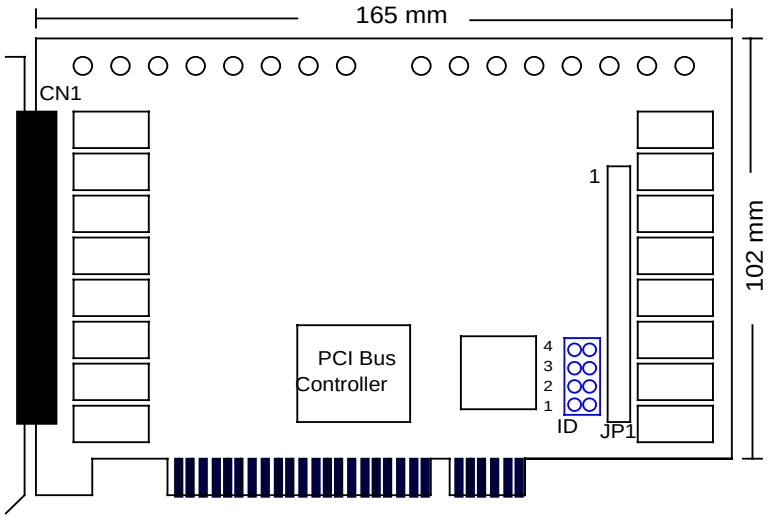
- **PCI configuration**

  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: Optical digital input and relay channel #0~channel #7 connector

JP1: Relay channel #8~channel #15 connector

ID: Card number selection jumper

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-94288

There are two connectors labeled "CN1" and "CN2". The CN1 connector is a 37-pin D-type connector and CN2 connector is a 40-pin FRC connector.

The CN2 can be converted to 37-pin D-type connector by using converting cable attached in the package.

The pin assignment of the CN2 (37-pins D-type connector) includes the isolated digital input channels (DI_0~DI_15) and relay output channels (Relay_0~Relay_7)

- **The CN2 pin assignment is as shown in Figure 2-1**

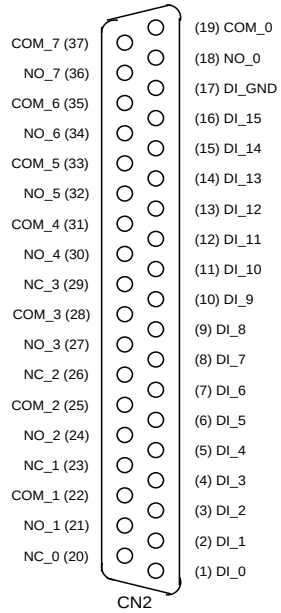


Figure 2-1 Pin Assignment of EX-94288 connector CN2

Legend:

DI_n: Digital input /output channel #n

DI_GND: Ground return path of input channels

NO_n: Relay output channel #n (normal open)

NC_n: Relay output channels #n (normal close)

COM_n: Relay common #n

- **The CN1 pin assignment is as shown in Figure 2-2**

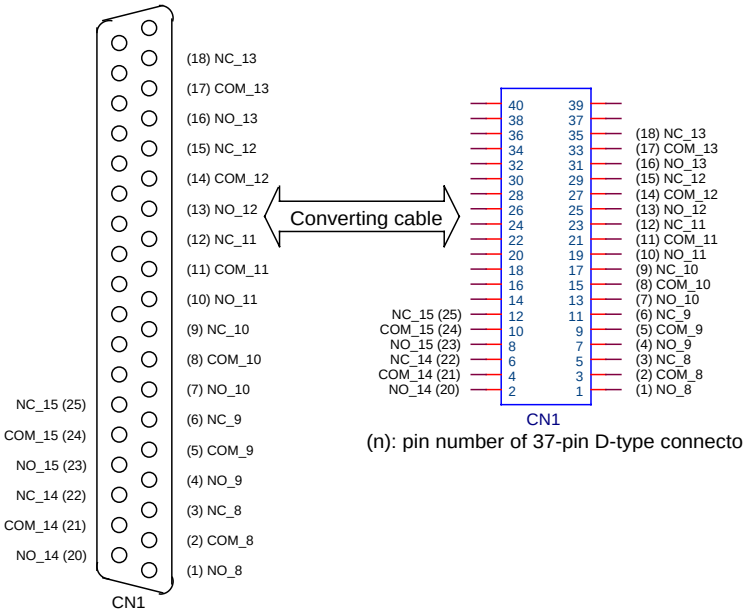


Figure 2-2 Pin Assignment of EX-94288 connector CN1

Legend:

NO_n: Relay output channel #n (normal open)

NC_n: Relay output channels #n (normal close)

COM_n: Relay common #n

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 16 isolated digital inputs and 16 relay output channels on EX-94288, each bit of based address is corresponding to a signal on the digital input or output channel.

3.3 PCI controller register address map

- **Reset control register**

  The EX-94288 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 + 0x00h

  Attribute: Write only

  Value: 01

- **PCI Internal special control register**

  EX-94288 internal control register, should be written with value 0FH before controlling EX-94288 card

  Address: Base + 002h

  Attribute: Write only

  Value: always are 0Fh

- **Interrupt mask control register**

  Enable or disable PCI interrupt INT #A

  Address: Base + 0x05h

  Attribute: Write only

  Value: 10H =enable PCI INT A#

  00H=disable PCI INT #A

- **Interrupt mode control register**

  Control the interrupt mode of DI_0 and DI_8 channels

  Address: Base + 0x03h

  Attribute: Write only

  Value:

  bit #1=0 : Disable interrupt form DI_0

  bit #1=1,bit #0=0 : Enable falling edge interrupt form DI_0

  bit #1=1,bit #0=1 : Enable rising edge interrupt form DI_0

  bit #3=0 : Disable interrupt form DI_8

  bit #3=1,bit #2=0 : Enable falling edge interrupt form DI_8

  bit #3=1,bit #2=0 : Enable rising edge interrupt form DI_8

- **Interrupt status register**

  Read the interrupt status of DI_0 and/or DI_8 channels or clear the interrupt status register

  Address: Base+0C8h

  Attribute: Read/write

  Value:

  Write: any data to clear interrupt status

  Read: bit #0= DI_0 interrupt, bit #1=DI_8 interrupt

- **I/O control registers**

  Address: Base + 0C0h~Base + 0E0h

  Attribute: Read/Write

  Value:

Port	Mode	Function
Base+0C0h	Write	Write data to relay output port #0 (NO_0~NO_7)
	Read	Read back current relay port #0 data (NO_0~NO_7)
Base+0C4h	Write	Write data to relay output port #1 (NO_8~NO_15)
	Read	Read back current relay port #1 data (NO_8~NO_15)
Base+0CCh	Write	No used
	Read	Read digital input port #0 (DI_0~DI_7)
Base+0E0h	Write	No used
	Read	Read digital input port #1 (DI_8~DI_15)

Table 3-1

Chapter 4
Jumper setting

4.1 Card number setting

Maximum four EX-94288 cards can be installed in system simultaneously with each has a unique card number.

A jumper called "ID" (see page 8) on the card is used to set the card number starts from 1 to 4

ID	Card number
	1 (default setting)
	2
	3
	4

Chapter 5
Operation Theorem

5.1 Digital Input Channels

The isolated digital inputs of EX-94288 can accept both dry contact input and voltage input (wet contact). The input voltage range form 0V to 24V and input resistor is 4.7K ohms (47K option). The connection between outside signal and EX-94288 is shown in Figure 5-3

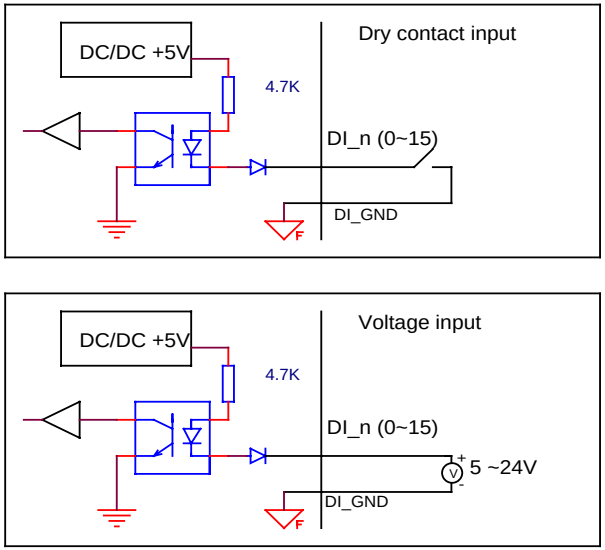


Figure 5-3 digital inputs of EX-94288

5.2 Digital Output Channels

On EX-94288, the COM_n pin is used as relay channel #n COMMON, NO_n pin is used as relay channel #n normal open output and NC_n pin is used as relay channel #n normal close output .The block as shown in Figure 5-4

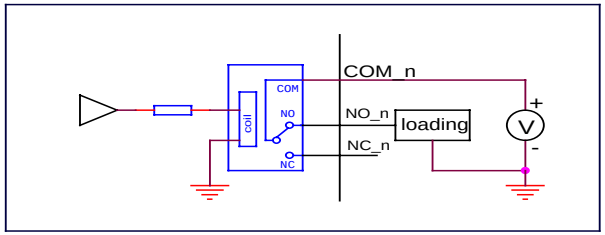


Figure 5-4 relay output of EX-94288

5.3 Edge Change Detection

The ECD (Edge Change Detection) detection circuit is used to detect the edge of level change. In the EX-94288, the detection circuit is applied to two input channels (DIO_0 and DIO_8). If channel is programmed to be rising edge or falling 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

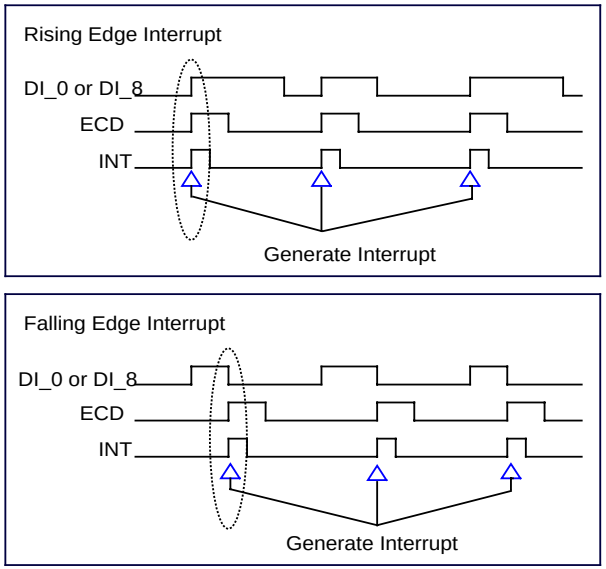


Figure 5-5 Debounce block diagram of EX-94288

Chapter 6 Libraries

This chapter describes the software library for operating this card. Only the functions in DOS library and Windows 98/2000 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.

6.1 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

6.2 How to use the Functions in PCIDAQ.DLL

- **VC++6.0:**
  1. Add file '../Include/PCIDAQ.H' in your project
  2. In link page of menu project| setting, add '../LIB/PCIDAQ.LIB' in the blank of Objects/Library Modules
  3. 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
  2. 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
  2. Add this sentence "#include '../Include/PCIDAQ.H' " to head of your main file.

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

6.3 Summary of function calls

6.4 W_4288_Open

Description:

Because the EX-94288 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-94288 cards have to be initialized by this function before calling other functions.

Syntax:

C/C++ (Dos)

```
WORD D_4288_Open (WORD cardNo);
```

C/C++ (Windows)

```
WORD D_4288_Open (WORD *ExistCards);
```

Visual BASIC (Windows)

Function W_4288_Open (ByRef ExitedCards As Long) As Long

Delphi

```
Function W_4288_Open (var ExistedCards:Integer): Integer;
```

Argument:

CardNo: card number be opened (for DOS only)

existCards: The number of installed EX-94288 cards. (for Windows only)

This return value shows how many EX-94288 cards are installed in your system.

Return Code:

Error code (Please refer to PCIDAQ.H or DOSDAQ.H)

6.5 W_4288_GetCardsID:

Description:

Get the cards number that is set by jumper on cards.

Syntax:

C/C++(DOS)

```
void D_4232_GetCardsID(WORD *CardsIDArray);
```

C/C++(Windows)

```
WORD W_4288_GetCardsID (WORD *CardsIDArray);
```

Visual BASIC (Windows)

```
Function W_4288_GetCardsID (ByRef CardsIDArray As Long)
         As Integer
```

Delphi

```
Function W_4288_GetCardsID (var CardsIDArray:Word):Word;
```

Argument:

CardsIDArray : This array return card number(1,2,3,4) which is set by jumper on card. You should define a 4 elements array, and then pass the array's pointer to this function.

Return Code:

Error code (Please refer to PCIDAQ.H or DOSDAQ.H)

6.6 W_4288_Version

Description:

PCIDAQ.DLL driver drives the EX-94288 cards. This function returns the version of PCIDAQ.DLL driver

Syntax:

C/C++ (Dos)

```
void D_4288_Version(char *version)
```

C/C++ (Windows)

```
WORD D_4288_Version (void)
```

Visual BASIC (Windows)

```
Function W_4288_Version () As Long
```

Delphi

```
Function W_4288_Version ():Integer;
```

Return Code:

Version: Version string (DOS only)

The version of PCIDAQ.DLL

6.7 W_4288_GetBusSlot

Description:

Get the PCI bus and slot occupied by EX-94288

Syntax:

C/C++ (Dos)

```
WORD D_4288_GetBusSlot (WORD cardNo, WORD *bus,WORD *slot);
```

C/C++ (Windows)

```
W_4288_GetBusSlot (WORD cardNo, WORD *bus,WORD *slot);
```

Visual BASIC (Windows)

```
Function W_4288_GetBusSlot (ByVal cardNo As Long,
         ByRef bus As Long, ByRef slot As Long) As Long
```

Delphi

```
Function W_4288_GetBusSlot (cardNo:Integer;
         var portNo:Integer;var bitNo:Integer):Integer;
```

Argument:

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

Bus: return PCI bus Number

Slot: return PCI slot Number of the bus

Return Code:

Error code (Please refer to PCIDAQ.H or DOSDAQ.H)

6.8 W_4288_Close

Description:

The IRQ and base_address of EX-94288 (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_4288_Close (WORD cardNo)
```

C/C++ (Windows)

```
W_4288_Close (void)
```

Visual BASIC (Windows)

```
Function W_4288_Close ()
```

Delphi

```
Function W_4288_Close ();
```

Argument:

None

Return Code:

None

6.9 W_4288_ReadDI

Description:

This function is used to read data from digital input port. There are two 8-bit digital inputs on the EX-94288. You can get 8-bit input data from EX-94288 by calling this function.

Syntax:

C/C++ (Dos)

```
WORD D_4288_ReadDI (WORD cardNo,WORD portNo,WORD *DiData)
```

C/C++ (Windows)

```
W_4288_ReadDI (WORD cardNo,WORD portNo,WORD *DiData)
```

Visual BASIC (Windows)

```
Function W_4288_ReadDI (ByVal cardNo As Long,
         ByVal portNo As Long, ByRef DiData As Long) As Long
```

Delphi

```
Function W_4288_ReadDI (cardNo:Integer;portNo:Integer;var
         DoData:Integer): Integer;
```

Argument:

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

portNo: Digital Input port number (0 or 1)

DiData: return digital input data

Return Code:

Error code (Please refer to PCIDAQ.H or DOSDAQ.H)

6.10 W_4288_WriteRelay

Description:

This function is used to write data to output port. There are two 8-bit digital outputs port on the EX-94288. You can send 8-bit output data to EX-94288 by calling this function.

Syntax:

C/C++ (Dos)

```
WORD D_4288_WriteRelay (WORD cardNo,WORD portNo,WORD Data);
```

C/C++ (Windows)

```
W_4288_WriteRelay (WORD cardNo,WORD portNo,WORD Data);
```

Visual BASIC (Windows)

```
Function W_4288_WriteRelay (ByVal cardNo As Long, ByVal
         portNo As Long, ByVal Data As Long) As Long
```

Delphi

```
Function W_4288_WriteRelay (cardNo:Integer;portNo:Integer;
         Data:Integer): Integer;
```

Argument:

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

portNo: Relay output port number (0 or 1)

Data: Data be written to output port

Return Code:

Error code (Please refer to PCIDAQ.H or DOSDAQ.H)

6.11 W_4288_ReadRelay

Description:

This function is used to read current data of realy output port. There are two 8-bit digital outputs port on the EX-94288. You can read back 8-bit output data of EX-94288 by calling this function.

Syntax:

C/C++ (Dos)

```
WORD D_4288_ReadRelay (WORD cardNo,WORD portNo,WORD
     *RelayData);
```

C/C++ (Windows)

```
WORD W_4288_ReadRelay (WORD cardNo,WORD portNo,WORD
     *RelayData);
```

Visual BASIC (Windows)

```
Function W_4288_ReadRelay (ByVal cardNo As Long,
         ByVal portNo As Long, ByRef RelayData As Long) As
         Long
```

Delphi

```
Function W_4288_ReadRelay (cardNo:Integer;portNo:Integer;
         var DoData: Integer): Integer;
```

Argument:

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

portNo: Relay output port number (0 or 1)

Data: Return current output data

Return Code:

Error code (Please refer to PCIDAQ.H or DOSDAQ.H)

6.12 W_4288_Set_RelayBit

Description:

Set a relay channel ON (energized)

Syntax:

C/C++ (Dos)

```
WORD D_4288_Set_RelayBit (WORD cardNo,WORD portNo,WORD
     bitNo);
```

C/C++ (Windows)

```
W_4288_Set_RelayBit (WORD cardNo,WORD portNo, WORDbitNo);
```

Visual BASIC (Windows)

```
Function W_4288_Set_RelayBit (ByVal cardNo As Long,
         ByVal portNo As Long, ByVal bitNo As Long) As Long
```

Delphi

```
Function W_4288_Set_RelayBit (cardNo:Integer;portNo:
         Integer; bitNo:Integer):Integer;
```

Argument:

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

portNo: Relay output port number (0 or 1)

bitNo: Channel Number(0 to 7)

Return Code:

Error code (Please refer to PCIDAQ.H or DOSDAQ.H)

6.13 W_4288_Reset_RelayBit

Description:

Set a relay channel Off (dis-energized)

Syntax:

C/C++ (Dos)

```
WORD D_4288_Reset_RelayBit (WORD cardNo,WORD portNo,WORD
     bitNo);
```

C/C++ (Windows)

```
WORD W_4288_Reset_RelayBit (WORD cardNo,WORD portNo, WORD
     bitNo);
```

Visual BASIC (Windows)

```
Function W_4288_Reset_RelayBit (ByVal cardNo As Long,
         ByVal portNo As Long, ByVal bitNo As Long) As Long
```

Delphi

```
Function W_4288_Reset_RelayBit (cardNo:Integer;portNo:
         Integer; bitNo:Integer): Integer;
```

Argument:

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

portNo: Relay output port number (0 or 1)

bitNo: Channel Number(0 to 7)

Return Code:

Error code (Please refer to PCIDAQ.H or DOSDAQ.H)

6.14 D_4288_Read_IntStatus

Description:

Get the interrupt status (for DOS only)

Syntax:

C/C++ (DOS)

```
WORD D_4288_Read_IntStatus (WORD cardNo,WORD *IntStatus)
```

Argument:

cardNo: card number set by jumper on the card

IntStatus: return PCI interrupt status.

if bit0 = 1,interrupted by channel 0 (DI_0)

if bit1 = 1,interrupted by channel 9 (DI_8)

Return Code:

Error code (Please refer to PCIDAQ.H or DOSDAQ.H)

6.15 W_4288_Clear_IntStatus

Description:

Clear interrupt by writing data to Base Port+0xC8

Syntax:

C/C++ (Dos)

```
WORD D_4288_Clear_IntStatus (WORD cardNo);
```

C/C++ (Windows)

```
W_4288_Clear_IntStatus (WORD cardNo);
```

Visual BASIC (Windows)

```
Function W_4288_Clear_IntStatus (ByVal cardNo As Long) As
         Long
```

Delphi

```
Function W_4288_Clear_IntStatus(cardNo:Integer):Integer;
```

Argument:

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

Return Code:

Error code (Please refer to PCIDAQ.H or DOSDAQ.H)

6.16 W_4288_IntEnable

Description:

Enable Interrupt of channel #0 (DI_0) and/or channel #8 (DI_8)

Syntax:

C/C++ (Dos)

```
WORD D_4288_IntEnable (WORD cardNo,WORD Int1Mode,
     WORD Int2Mode,User_Interrupt_HANDLER
     userIntServiceRoutine);
```

C/C++ (Windows)

```
WORD W_4288_IntEnable (WORD cardNo,WORD Int1Mode,WORD
     Int2Mode, User_Interrupt_HANDLER
     userIntServiceRoutine);
```

Visual BASIC (Windows)

```
Function W_4288_IntEnable (ByVal cardNo As Long, ByVal
         Int1Mode As Long, ByVal Int2Mode As Long,
         ByVal userIntServiceRoutine As Long) As Long
```

Delphi

```
Function W_4288_IntEnable (cardNo:Integer;Int1Mode:Integer;
         Int2Mode:Integer; userIntServiceRoutine:Pointer):
         Integer;
```

Argument:

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

Int1Mode: Interrupt mode of channel #0 (DI_0)

Bit #0=1 or 0: rising or falling edge Interrupt

Bit #1=1 or 0: enable/Disable interrupt

Int2Mode: Interrupt mode of channel #8 (DI_8)

Bit #0=1 or 0: rising or falling edge Interrupt

Bit #1=1 or 0: enable/disable interrupt

userIntServiceRoutine: user Interrupt service routine called when interrupt occurs.

Return Code:

Error code (Please refer to PCIDAQ.H or DOSDAQ.H)

6.17 W_4288_IntDisable

Description:

Disable interrupt of digital input channel #0 (DI_0) and channel #8 (DI_8)

Syntax:

C/C++ (Dos)

```
WORD W_4288_IntDisable (WORD cardNo);
```

C/C++ (Windows)

```
WORD W_4288_IntDisable (WORD cardNo);
```

Visual BASIC (Windows)

```
Function W_4288_IntDisable (ByVal cardNo As Long)
```

Delphi

```
Function W_4288_IntDisable (cardNo:Integer);
```

Argument:

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

Return Code:

Error code (Please refer to PCIDAQ.H or DOSDAQ.H)

Chapter 7
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.

7.1 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)