

# EX-94232

## Isolated 16 channel D/I and 16 channel D/O

### Copy Right Notice

The information in this manual is subject to change without prior notice in order to improve reliability, design and function and DOES not represent a commitment on the part of the manufacturer. No part of this manual may be reproduced, copied, or transmitted in any form without the prior written permission of manufacturer.

### Acknowledgment

Products mentioned in this manual are mentioned for identification purpose only. Products names appearing in this manual may or may not be registered trademarks or copyright of their respective companies

Printed Aug. 2002 Rev 1.0

## Table of Contents

<b>Chapter 1 Introduction.....</b>	<b>4</b>
1.1 Introduction.....	4
1.2 Features .....	4
1.3 Applications .....	4
1.4 Specifications.....	5
1.5 Software Supporting .....	6
1.6 Programming Library .....	6
<b>Chapter 2 Installation.....</b>	<b>7</b>
2.1 What You Have .....	7
2.2 Unpacking.....	7
2.3 Hardware Installation Outline .....	7
2.4 PCB Layout.....	8
2.5 Installation Procedures .....	9
2.6 Device Installation for Windows Systems .....	9
2.7 Connector Pin Assignment of EX-94232.....	10
2.8 Card number setting .....	11
<b>Chapter 3 Registers Format .....</b>	<b>12</b>
3.1 PCI PnP Registers.....	12
3.2 Reset control registers.....	12
3.3 PCI controller register address map .....	13
<b>Chapter 4 Operation Theorem.....</b>	<b>15</b>
4.1 Isolated Digital Input Channels .....	15
4.2 Isolated Digital Output Channels .....	16
4.3 Edge Change Detection.....	17
<b>Chapter 5 Libraries.....</b>	<b>18</b>
5.1 Libraries Installation .....	18
5.2 How to use the Functions in PCIDAQ.DLL .....	18
5.3 Summary of function calls.....	19
5.4 W_4232_Open.....	20
5.5 W_4232_GetCardsID: .....	21
5.6 W_4232_Version .....	22
5.7 W_4232_GetBusSlot .....	23
5.8 W_4232_Close .....	24
5.9 W_4232_Read_Di .....	25

5.10	W_4232_Write_Do.....	26
5.11	W_4232_Read_Do.....	27
5.12	W_4232_Set_Do_Bit.....	28
5.13	W_4232_Reset_Do_Bit.....	29
5.14	D_4232_Read_IntStatus.....	30
5.15	W_4232_Clear_IntStatus.....	31
5.16	W_4232_IntEnable.....	32
5.17	W_4232_IntDisable.....	33
<b>Chapter 6</b>	<b>EX-9837 Terminal board.....</b>	<b>34</b>
6.1	Main features.....	34

# Chapter 1

## Introduction

### 1.1 Introduction

The EX-94232 is 16-CH high-density isolated digital input and 16-CH output product. This I/O card is isolated up-to 5000 Vdc (excluding cables) for channel-to-computer isolation. It protects your computer against damage caused by accidental contact with high external voltage and eliminates troublesome ground loops.

The EX-94232 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-94232 Isolated digital I/O card provide the following advanced features:

- ◆ 16 Isolated digital Input channels (non-polarity input for EX-94232)
- ◆ 16 Isolated digital output channels
- ◆ High output driving capability, 500mA sink current on isolated output
- ◆ 5000 Vrms high voltage isolation
- ◆ External interrupt signal on DI channels
- ◆ 37-pin D-type connector (Pin compatible to EX-9837)(see page 34)

### 1.3 Applications

- ◆ Laboratory and Industrial automation
- ◆ Watchdog timer
- ◆ Event counter
- ◆ Frequency counter and generator
- ◆ Low level pulse generator
- ◆ Time delay

## 1.4 Specifications

### ◆ Optical Isolated Input Channel

Numbers of Channel: 16 digital inputs

Input polarity: polarity sensitive for EX-94232, and non-polarity for EX-94232

Input Voltage: 0 - 24V dc

Logic H: 3~24V

Logic L: 0~2.4V

Input resistance: 4.7K $\Omega$  @ 0.5W

Isolated voltage: 5000 Vrms

Throughput: 10K Hz (0.1 ms)

### ◆ Optical Isolated Output Channel

Numbers of Channel: 16 digital outputs

Output type: Darlington transistors with common ground

Output voltage: 5V<sub>DC</sub> min, 90V<sub>DC</sub> maximum

Output Device: ULN2803(common ground)

Sink Current: Max. 500mA/ch for only one of the ULN2803 transistor is ON

Power Dissipation: 1.47W per ULN2803 device (8 channels)

Isolated voltage: 5000 V<sub>DC</sub>

### ◆ Interrupt Sources

Channel 0 and channel 8 of digital input channels

### ◆ 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 @ 330 mA typical

Dimension: 144mm(W) x110mm (H)

## 1.5 Software Supporting

**Topscoc** 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 functions descriptions are included in this user's guide.
- ◆ **Windows 98/2000/NT/Me/XP DLL:** For VB, VC++, BC5, the functions descriptions 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-94232 card. Please follow the follow steps to install the EX-94232 card.

### 2.1 What You Have

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

- ◆ EX-94232 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-94232 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 handling 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

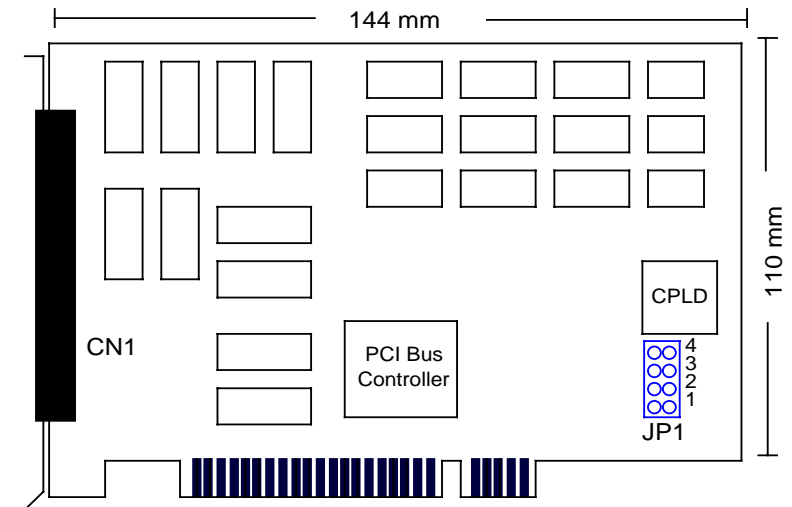
#### ◆ PCI configuration

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: 37-pin D-type connector

JP1: Card number setting 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-94232

The pin assignment of the 37-pin D-type connector is an isolated signal connector, EX-94232's pin assignment is as shown in Figure 2.7

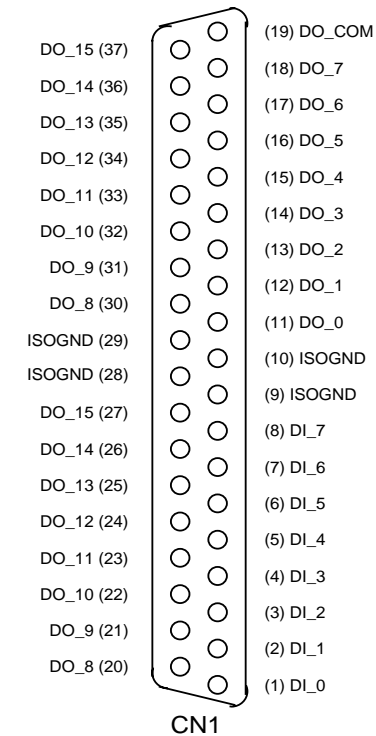


Figure 2.7 Pin Assignment of EX-94232 connector CN1

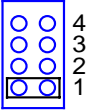
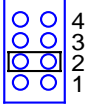
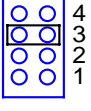
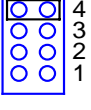
### Legend:

- ◆ **DI\_n**: Isolated digital input channel #n
- ◆ **DO\_n**: Isolated digital output channel #n
- ◆ **DO\_COM**: Power input signal for fly-wheel diode of DO channels
- ◆ **ISOGND**: Ground return path of isolated input and output channels

## 2.8 Card number setting

Maximum four EX-94232 cards can be installed in system simultaneously with each has a unique card number.

A jumper called "JP1" (see page 8) on the card is used to set the card number starts from 1 to 4

JP1	Card number
	1 (default setting)
	2
	3
	4

# 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](http://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-94232 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 Reset control registers

The EX-94232 is in inactive state when the system power on, and should be activated by set bit 0 of this register to "1" state

**Address:** Base + 0x00

**Attribute:** Write only

Bit	7	6	5	4	3	2	1	0	State
Base+0x00	0	0	0	0	0	0	0	0	Inactive (reset) state (Default)
Base+0x00	0	0	0	0	0	0	0	1	Active state

**Note:** Bit 0 of this register should be set to "1" before using EX-94232

### 3.3 PCI controller register address map

#### ◆ Reset control register

The EX-94232 is in inactive state when the system power on, and should be activated by set bit 0 of this register to "1" state

**Address:** Base + 0x00h

**Attribute:** Write only

**Value:** 01

#### ◆ PCI Internal special control register

EX-94232 internal control register, should be written with value 0FH before controlling EX-94232 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\_9

bit #3=1,bit #2=0 : Enable falling edge interrupt form DI\_8

bit #3=1,bit #2=1 : 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 output port #0 (DO_0~DO_7)
	Read	Read back current port #0 data (DO_0~DO_7)
Base+0C4h	Write	Write data to output port #1 (DO_8~DO_15)
	Read	Read back current port #1 data (DO_8~DO_15)
Base+0CC h	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

## Operation Theorem

### 4.1 Isolated Digital Input Channels

The isolated digital input is open collector transistor structure. The input voltage range form 0V to 24V and input resistor is 4.7K ohms. The connection between outside signal and EX-94232 is shown in Figure 4-1.

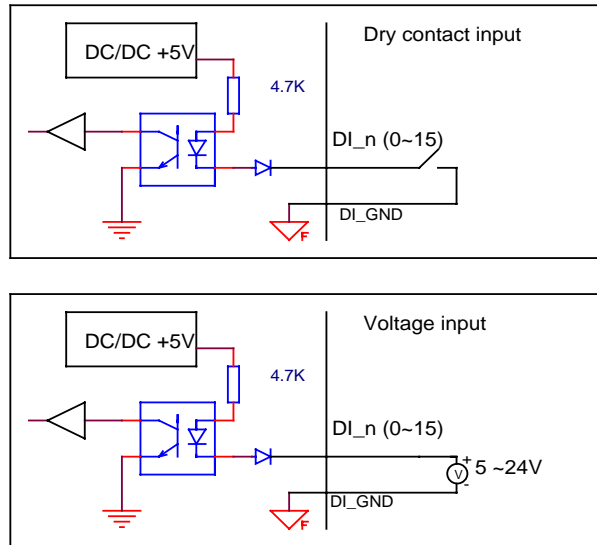


Figure 4-1 Isolated digital inputs of EX-94232

### 4.2 Isolated Digital Output Channels

On EX-94232, the DO\_COM pin is used as "fly-wheel" diode, which can protect the driver if the loading is inductance loading such as relay, motor or solenoid. If the loading is resistance loading such as resistor or LED, the connection to fly-wheel diode is not necessary.

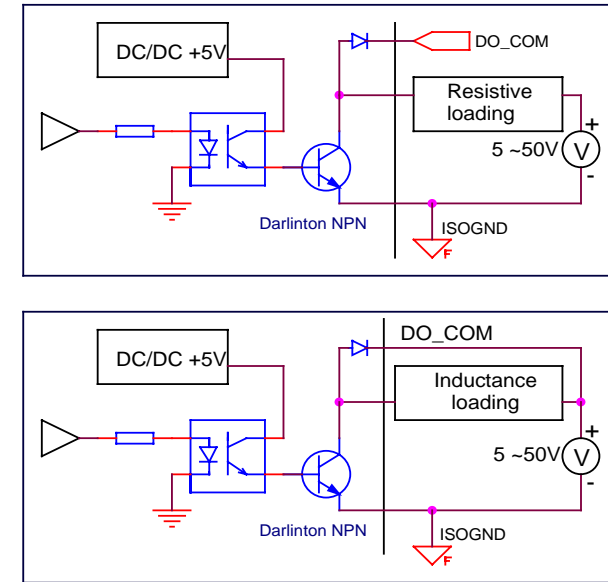


Figure 4-2 isolated digital outputs

Please note that when the loading is as "inductance type loading" such as relay, coil or motor, the DO\_COM pin must be connected to the external power source. The extra connection is to utilize the 'fly-wheel diode' to form a current-release closed loop, so that the transistor won't be destroyed by the reverse high voltage which is generated by the inductance load when the output switches from "ON" to "OFF".



### 4.3 Edge Change Detection

The ECD (Edge Change Detection) detection circuit is used to detect the edge of level change. In the EX-94232, the detection circuit is applied to two input channels (DI\_0 and DI\_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

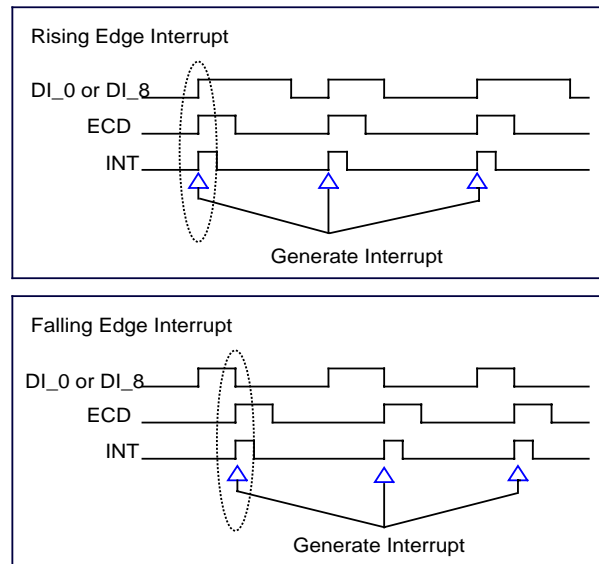


Figure 4-3 Input debounce block diagram

## Chapter 5 Libraries

This chapter describes the software library for operating this card. Only the functions in DOS library and Windows 98/2000/NT/XP DLL are described. Please refer to the PCIDAQ function reference manual, which included in Topsgcc 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 Topsgcc CD also includes the detail examples and readme files

### 5.2 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:

4. Add file './Include/Declare.bas' in your project.

#### ◆ Delphi:

5. Add file './Include/Declare.pas' in your project
6. Add this sentence "uses Declare;" in the head of your unit.pas

#### ◆ C++Builder:

7. Add file './Include/PCIDAQ.H' and './Lib/PCIDAQ\_CB.lib' to your project
8. 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_4232_Open	Initial EX-94232 card before using other functions	20
W_4232_GetCardsID	Get EX-94232 card number	21
W_4232_Version	Get version number of PCIDAQ.DLL	22
W_4232_GetBusSlot	Get PCI bus and slot number occupied by EX-94232	23
W_4232_Close	Close EX-94232 card before terminating program	24
W_4232_Read_Di	Read digital input port data (8-bit)	25
W_4232_Write_Do	Write data (8-bit) to digital output port	26
W_4232_Read_Do	Read back current relay port value	27
W_4232_Set_Do_Bit	Set a bit of port to high	28
W_4232_Reset_Do_Bit	Reset a bit of port to low	29
D_4232_Read_IntStatus	Read interrupt status register (DOS only)	30
W_4232_Clear_IntStatus	Clear interrupt status register	31
W_4232_IntEnable	Enable digital input change interrupt	32
W_4232_IntDisable	Disable digital input interrupt	33

### 5.4 W\_4232\_Open

#### Description:

Because the EX-94232 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-94232 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 D_4232_Open (WORD *ExistCards);
```

##### Visual BASIC (Windows)

```
Function W_4232_Open (ByRef ExitedCards As Long) As Long
```

##### Delphi

```
Function W_4232_Open (var ExistedCards:Integer): Integer;
```

#### Argument:

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

existCards: The number of installed EX-94232 cards. (for Windows only). This return value shows how many EX-94232 cards are installed in your system.

#### Return Code:

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

## 5.5 W\_4232\_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_4232_GetCardsID (WORD *CardsIDArray);
```

#### Visual BASIC (Windows)

```
Function W_4232_GetCardsID (ByRef CardsIDArray As Long)
    As Integer
```

#### Delphi

```
Function W_4232_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, then pass the array's pointer to this function.

### Return Code:

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

## 5.6 W\_4232\_Version

### Description:

PCIDAQ.DLL driver drives the EX-94232 cards. This function returns the version of PCIDAQ.DLL driver

### Syntax:

#### C/C++ (DOS)

```
void D_4232_Version(char *version)
```

#### C/C++ (Windows)

```
WORD D_4232_Version (void)
```

#### Visual BASIC (Windows)

```
Function W_4232_Version () As Long
```

#### Delphi

```
Function W_4232_Version ():Integer;
```

### Argument:

Version: Return version string (DOS only)

### Return Code:

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

## 5.7 W\_4232\_GetBusSlot

### Description:

Get the PCI bus and slot occupied by EX-94232

### Syntax:

#### C/C++ (DOS)

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

#### C/C++ (Windows)

```
WORD W_4232_GetBusSlot (WORD cardNo, WORD *bus,WORD *slot);
```

#### Visual BASIC (Windows)

```
Function W_4232_GetBusSlot (ByVal cardNo As Long,  
    ByRef bus As Long, ByRef slot As Long) As Long
```

#### Delphi

```
Function W_4232_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)

## 5.8 W\_4232\_Close

### Description:

The IRQ and base\_address of EX-94232 (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)

```
W_4232_Close (void)
```

#### Visual BASIC (Windows)

```
Function W_4232_Close ()
```

#### Delphi

```
Function W_4232_Close ();
```

### Argument:

None

### Return Code:

None

## 5.9 W\_4232\_Read\_Di

### Description:

This function is used to read data from digital input port. There are two 8-bit digital inputs on the EX-94232. You can get 8-bit input data from EX-94232 by calling this function.

### Syntax:

#### C/C++ (DOS)

```
WORD D_4232_Read_Di (WORD cardNo,WORD portNo,WORD *DiData)
```

#### C/C++ (Windows)

```
WORD W_4232_Read_Di (WORD cardNo,WORD portNo,WORD *DiData)
```

#### Visual BASIC (Windows)

```
Function W_4232_Read_Di (ByVal cardNo As Long,  
                        ByVal portNo As Long, ByVal DiData As Long) As Long
```

#### Delphi

```
Function W_4232_Read_Di (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)

## 5.10 W\_4232\_Write\_Do

### Description:

This function is used to write data to output port. There are two 8-bit digital outputs port on the EX-94232. You can send 8-bit output data to EX-94232 by calling this function.

### Syntax:

#### C/C++ (DOS)

```
WORD D_4232_Write_Do (WORD cardNo,WORD portNo,WORD Data);
```

#### C/C++ (Windows)

```
WORD W_4232_Write_Do (WORD cardNo,WORD portNo,WORD Data);
```

#### Visual BASIC (Windows)

```
Function W_4232_Write_Do (ByVal cardNo As Long,  
                        ByVal portNo As Long, ByVal Data As Long) As Long
```

#### Delphi

```
Function W_4232_Write_Do (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)

## 5.11 W\_4232\_Read\_Do

### Description:

This function is used to read current data of output port. There are two 8-bit digital outputs port on the EX-94232. You can read back 8-bit output data of EX-94232 by calling this function.

### Syntax:

#### C/C++ (DOS)

```
WORD D_4232_Read_Do (WORD cardNo,WORD portNo,
                    WORD *RelayData);
```

#### C/C++ (Windows)

```
WORD W_4232_Read_Do (WORD cardNo,WORD portNo,WORD
                    *RelayData);
```

#### Visual BASIC (Windows)

```
Function W_4232_Read_Do (ByVal cardNo As Long,
                        yVal portNo As Long, ByRef RelayData As Long) As Long
```

#### Delphi

```
Function W_4232_Read_Do (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)

## 5.12 W\_4232\_Set\_Do\_Bit

### Description:

Set a relay channel ON (energized)

### Syntax:

#### C/C++ (DOS)

```
WORD D_4232_Set_Do_Bit (WORD cardNo,WORD portNo,
                       WORD bitNo);
```

#### C/C++ (Windows)

```
WORD W_4232_Set_Do_Bit (WORD cardNo,WORD portNo,
                       WORDbitNo);
```

#### Visual BASIC (Windows)

```
Function W_4232_Set_Do_Bit (ByVal cardNo As Long,
                           ByVal portNo As Long, ByVal bitNo As Long) As Long
```

#### Delphi

```
Function W_4232_Set_Do_Bit (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)

### 5.13 W\_4232\_Reset\_Do\_Bit

**Description:**

Set a relay channel Off (dis-energized)

**Syntax:**
**C/C++ (DOS)**

```
WORD D_4232_Reset_Do_Bit (WORD cardNo,WORD portNo,
WORD bitNo);
```

**C/C++ (Windows)**

```
WORD W_4232_Reset_Do_Bit (WORD cardNo,WORD portNo,
WORD bitNo);
```

**Visual BASIC (Windows)**

```
Function W_4232_Reset_Do_Bit (ByVal cardNo As Long,
ByVal portNo As Long, ByVal bitNo As Long) As Long
```

**Delphi**

```
Function W_4232_Reset_Do_Bit (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)

### 5.14 D\_4232\_Read\_IntStatus

**Description:**

Get the interrupt status (for DOS only)

**Syntax:**
**C/C++ (DOS)**

```
WORD D_4232_Read_IntStatus (WORD cardNo,WORD *IntStatus)
```

**Argument:**

cardNo: card number set by jumper on the card

IntStatus: return PCI interrupt status.

if bit 0 = 1,interrupted by channel 0 (DI\_0)

if bit 1 = 1,interrupted by channel 9 (DI\_8)

**Return Code:**

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

## 5.15 W\_4232\_Clear\_IntStatus

### Description:

Clear interrupt by writing data to Base Port+0xC8

### Syntax:

#### C/C++ (DOS)

```
WORD D_4232_Clear_IntStatus (WORD cardNo);
```

#### C/C++ (Windows)

```
WORD W_4232_Clear_IntStatus (WORD cardNo);
```

#### Visual BASIC (Windows)

```
Function W_4232_Clear_IntStatus (ByVal cardNo As Long)
    As Long
```

#### Delphi

```
Function W_4232_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)

## 5.16 W\_4232\_IntEnable

### Description:

Enable Interrupt of channel #0 (DI\_0) and/or channel #8 (DI\_8)

### Syntax:

#### C/C++ (DOS)

```
WORD D_4232_IntEnable (WORD cardNo,WORD Int1Mode,
    WORD Int2Mode,User_Interrupt_HANDLER
    userIntServiceRoutine);
```

#### C/C++ (Windows)

```
WORD W_4232_IntEnable (WORD cardNo,WORD Int1Mode,
    WORD Int2Mode, User_Interrupt_HANDLER
    userIntServiceRoutine);
```

#### Visual BASIC (Windows)

```
Function W_4232_IntEnable (ByVal cardNo As Long,
    ByVal Int1Mode As Long, ByVal Int2Mode As Long,
    ByVal userIntServiceRoutine As Long) As Long
```

#### Delphi

```
Function W_4232_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)



## 5.17 W\_4232\_IntDisable

### Description:

Disable interrupt of digital input channel #0 (DI\_0) and channel #8 (DI\_8)

### Syntax:

#### C/C++ (DOS)

```
WORD W_4232_IntDisable (WORD cardNo);
```

#### C/C++ (Windows)

```
WORD W_4232_IntDisable (WORD cardNo);
```

#### Visual BASIC (Windows)

```
Function W_4232_IntDisable (ByVal cardNo As Long)
```

#### Delphi

```
Function W_4232_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 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)

