EX-94164/AC
Isolated
64 channel D/I

Acknowledgment

Products mentioned in this manual are mentioned for identification purpose only. Products manes 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

Chapter 1 Introduction

1.1 Introduction

The EX-94164/AC is 64-CH high-density isolated digital input product. This input 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-94164/AC 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-94164/AC Isolated digital I/O card provide the following advanced features:

- 64 Isolated digital Input channels (non-polarity input for EX-94164AC)
- 5000 Vrms high voltage isolation
- External interrupt signal on digital input channels
- Built-in digital debounce

1.3 68-pin SCSI-II connector (Pin compatible to EX-98068) (see page 34)

1.4 Specifications

- **Optical Isolated Input Channel**

  Numbers of Channel: 64 digital input channels

  Input polarity: polarity sensitive for EX-94164, and non-polarity for EX-94164AC

  Input Voltage: 0 - 24V dc

  Logic H: 3~24V

  Logic L: 0~2.4V

  Input resistance: 4.7KΩ @ 0.5W

  Isolated voltage: 5000 Vrms

  Throughput: 10K Hz (0.1 ms)

- **Interrupt Sources**

  Channel #0 to channel #15 of digital input channels

- **Digital debounce**

  Software programmable from 20 msec to 300 msec for all digital input channels

- **General Specifications**

  Connector: **68**-pin SCSI-II 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 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-94164/AC card. Please follow the follow steps to install the EX-94164/AC card.

2.1 What You Have

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

- EX-94164/AC 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-94164/AC 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

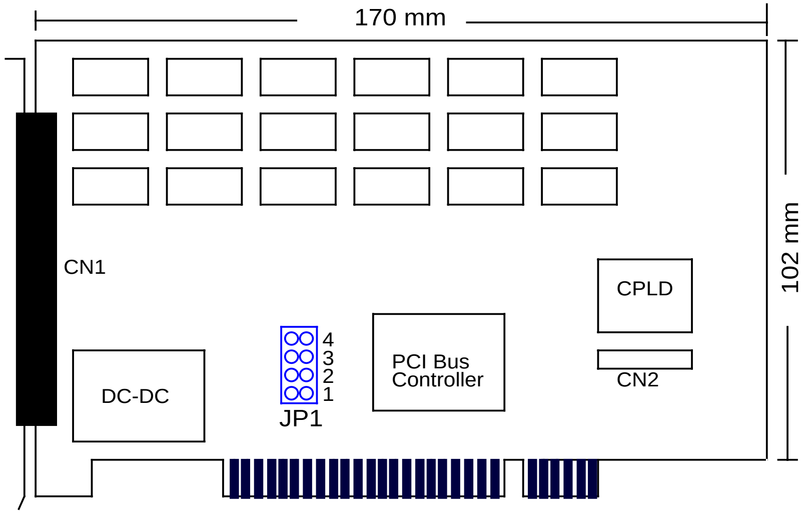
- **PCI configuration**

  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 connector (DI_0~D_-63), ISOGND, and ISOVCC (+15VDC)

CN2: No used

JP1: Card number setting

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-94164/AC

The pin assignment of the 68 pins SCSI-II connector is an isolated signal connector, EX-94164/AC's pin assignment is as shown in Figure 2.7

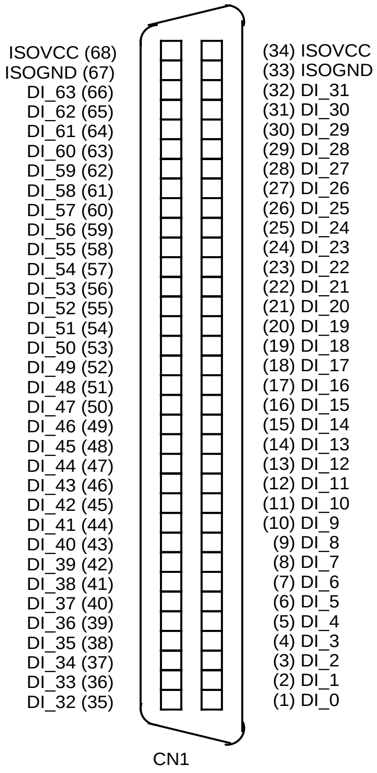


Figure 2.7 Pin Assignment of EX-94164/AC connector CN2

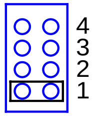
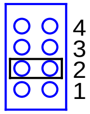
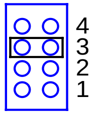
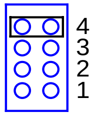
Legend:

- **DI_n:** Isolated digital input channel #n
- **ISOVC**: Isolated +15VDC output (100 mA max.)
- **ISOGND**: Ground return path of isolated input and output channels

2.8 Card number setting

Maximum four EX-94164/AC 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..

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 EX-94164/AC 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 PCI controller register address map

- **Reset control register**

The EX-94164 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 + 00H

Attribute: Write only

Value: 01

- **PCI Internal special control register**

EX-98354 internal control register, should be written with value 00H before controlling EX-98354 card

Address: Base + 02H

Attribute: Write only

Value: always are 00H

- **Interrupt mask control register**

Enable or disable PCI interrupt INT #A

Address: Base + 05H

Attribute: Write only

Value: 10H =enable PCI INT A#

00H=disable PCI INT #A

3.3 Digital Input Register Address Map

There are 64 isolated digital input channels on EX-94164/AC, each bit of based address is corresponding to a signal on the digital input channel.

Address: BASE+0C0H~ BASE+0DCH (port 0 ~ port 7)

Attribute: read only

Value:

Bit	Port	7	6	5	4	3	2	1	0
Base+0C0H	0	DI_7	DI_6	DI_5	DI_5	DI_3	DI_2	DI_1	DI_0
Base+0C4H	1	DI_15	DI_14	DI_13	DI_12	DI_11	DI_10	DI_9	DI_8
Base+0C8H	2	DI_23	DI_22	DI_21	DI_20	DI_19	DI_18	DI_17	DI_16
Base+0CCH	3	DI_31	DI_30	DI_29	DI_28	DI_27	DI_26	DI_25	DI_24
Base+0D0H	4	DI_39	DI_38	DI_37	DI_36	DI_35	DI_34	DI_33	DI_32
Base+0D4H	5	DI_47	DI_46	DI_45	DI_44	DI_43	DI_42	DI_41	DI_40
Base+0D8H	6	DI_55	DI_54	DI_53	DI_52	DI_51	DI_50	DI_49	DI_48
Base+0DCH	7	DI_63	DI_62	DI_61	DI_60	DI_59	DI_58	DI_57	DI_56

3.4 Debounce control registers

There are total thirty two bits on EX-94164 to enable or disable debounce function of digital input ports (port #0~port #3)

Address: Base + 0C8H

Attribute: Write only

Value:

Bit no.	Value	Port no.	Debounce mode
Bit 0	0	Port #0	Disable
	1		Enable
Bit 1	0	Port #1	Disable
	1		Enable
Bit 2	0	Port #2	Disable
	1		Enable
Bit 3	0	Port #3	Disable
	1		Enable
Bit 4~Bit 7	Debounce factor		
	0000~1111		

Debounce period =10 msec x Debounce factor

3.5 Interrupt control register

There are total 8 digital input channels (DI_0~DI_7) on the EX-94164/AC can generate interrupt signal. This register control the interrupt mode of input channels DI_0~DI-7 (port #0)

3.5.1 Interrupt mask control register

This register can individually enables or disable interrupt of each digital input channels DI_0~DI_7 (port #0)

Address: BASE + 0C0H

Attribute: write only

Value:

Bit no.	Value	Channel no.	Interrupt
Bit 0	0	Channel #0 (DI_0)	Disable
	1		Enable
Bit 1	0	Channel #1 (DI_1)	Disable
	1		Enable
Bit 2	0	Channel #2 (DI_2)	Disable
	1		Enable
Bit 3	0	Channel #3 (DI_3)	Disable
	1		Enable
Bit 4	0	Channel #4 (DI_4)	Disable
	1		Enable
Bit 5	0	Channel #5 (DI_5)	Disable
	1		Enable
Bit 6	0	Channel #6 (DI_6)	Disable
	1		Enable
Bit 7	0	Channel #7 (DI_7)	Disable
	1		Enable

3.5.2 Interrupt trigger control register

Interrupt signal can be triggered at rising edge or falling edge of each input on the EX-94164 card. This register is used to control interrupt trigger mode

Address: BASE + 0C4H

Attribute: write only

Value:

Bit no.	Value	Channel no.	Trigger mode
Bit 0	0	Channel #0 (DI_0)	Rising edge
	1		Falling edge
Bit 1	0	Channel #1 (DI_1)	Rising edge
	1		Falling edge
Bit 2	0	Channel #2 (DI_2)	Rising edge
	1		Falling edge
Bit 3	0	Channel #3 (DI_3)	Rising edge
	1		Falling edge
Bit 4	0	Channel #4 (DI_4)	Rising edge
	1		Falling edge
Bit 5	0	Channel #5 (DI_5)	Rising edge
	1		Falling edge
Bit 6	0	Channel #6 (DI_6)	Rising edge
	1		Falling edge
Bit 7	0	Channel #7 (DI_7)	Rising edge
	1		Falling edge

3.6 Interrupt status register

There is one register store the interrupt status of input channels (DI_0~DI_7)

Address: Base+0E0H

Attribute: Read only

Value:

Bit Port no	**Interrupt status**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Base+0E0H	DO_7	DO_6	DO_5	DO_5	DO_3	DO_2	DO_1	DO_0

Note: Bit #n=1 means that the corresponding channel is interrupted

3.7 Clear Interrupt register

This register is used to clear interrupt status registers to accept next interrupt cycle

Write any value to this register after interrupt handle completed

Address: Base+0CCH

Attribute: Write only

Value: any value

Chapter 4
Operation Theorem

4.1 Isolated Digital Input Channels

The isolated digital input is can accept dry contact and/or voltage input signal. The input voltage range form 0V to 24V and input resister is 4.7K ohms. The connection between outside signal and EX-94164/AC is shown in Figure 4-1 and Figure 4-2

- **Dry contact digital input**

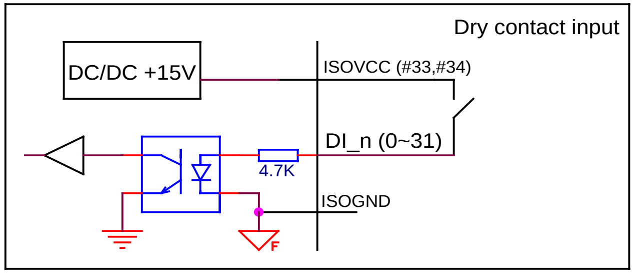


Figure 4-1 Dry contact inputs of EX-94164/AC

- **Voltage digital input**

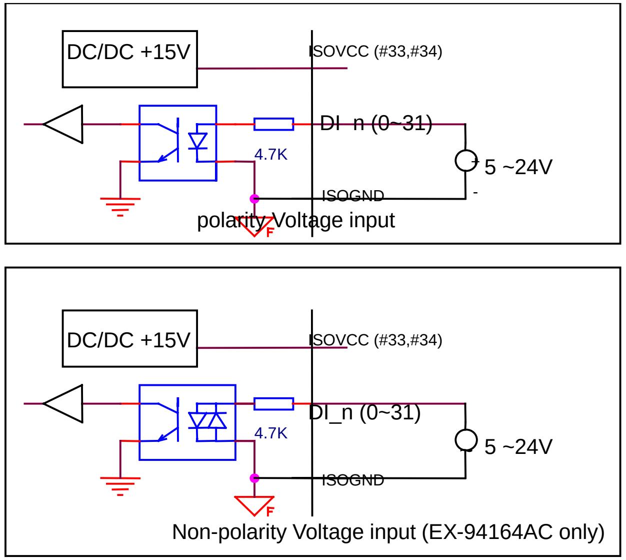


Figure 4-2 Voltage digital inputs of EX-94164/AC

Note: The digital input connections of EX-94164AC are not polarity sensitive whether used on AC or DC voltage.

4.2 Edge Change Detection

The ECD (Edge Change Detection) detection circuit is used to detect the edge of level change. In the EX-94164/AC, the detection circuit is applied to 16 input channels (DI_0 ~ DI_15). 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

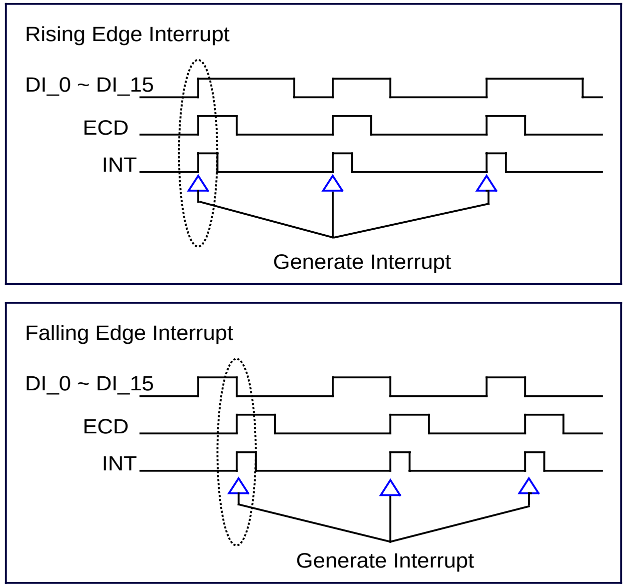


Figure 4-3 inputs interrupt block diagram

4.3 Digital debounce

Each digital input channel has a programmable digital debounce for eliminating unexpected signals and noise from the card circuitry. The user can set different digital debouncing parameters for each input channel in different applications. The following is a functional description of the digital debounce.

1. When a digital debounce is enabled, the EX-94164/AC will sample the signals at the enabled input channel at a 20 ms sampling rate.
2. When a high or low signal is present at a digital input channel whose digital debounce function is enabled, the signal will be filtered out as noise unless it lasts for an effective period.
3. The effective period is determined by multiplying the sampling rate (20 ms) by the sampling number (1 ~15) chosen by the user, i.e.

   Effective period = debounce factor x 20 ms.
4. See Sec 5.9 (page 30) to more detail using of debounce function

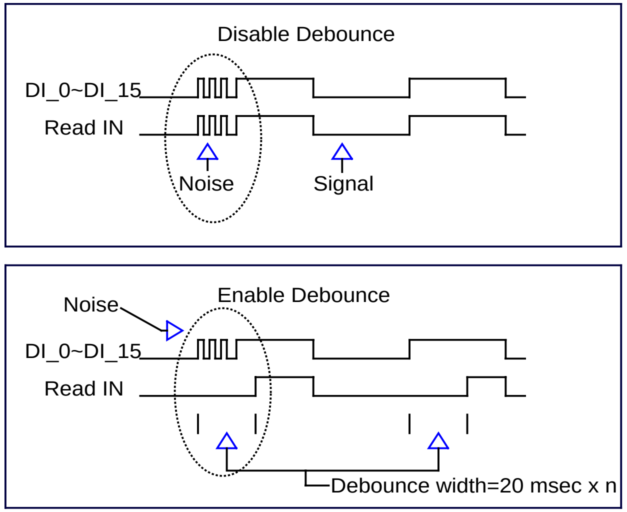


Figure 4-4 inputs debouce block diagram

Chapter 5
Libraries

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

5.1 Libraries Installation

EX-94164/AC card comes with software drivers for DOS and Windows. The following sections show you how to install the software libraries DOSDAQH.LIB for DOS, or PCIDAQ for Windows 98/NT/2000.

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

5.1.1 How to use the DOSDAQH.LIB in DOS

- **For BC compiler**

1. Large mode: Add ..\LIB\BC\DOSDAQL.LIB in your project

@ Huge mode: Add ..\LIB\BC\DOSDAQH.LIB in your project

@ Include DOSDAQ.H in your source file

- **For MSC compiler**

1. Large mode: Add ..\LIB\MSC\DOSDAQL.LIB in your project

@ Huge mode: Add ..\LIB\MSC\DOSDAQH.LIB in your project

@ Include DOSDAQ.H in your source file

- **For TC compiler**

1. Large mode: Add ..\LIB\TC\DOSDAQL.LIB in your project

@ Huge mode: Add ..\LIB\TC\DOSDAQH.LIB in your project

@ Include DOSDAQ.H in your source file

5.1.2 How to use the PCIDAQ.DLL s in Windows

- **VC++6.0:**

1. 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/'

5.2 Summary of function calls

The EX-94164 provides you eleven function calls to access the digital input signals.

Function	Description	page
Open card	Initial EX-94164/AC card before using	24
Get Card's ID	Get PCI ID code of EX-94164/AC	25
Get Driver Version	Get version number of PCIDAQ.DLL	26
Close card	Close EX-94164/AC card before terminating program	27
Get PCI Bus and Slot number	Get PCI bus and slot number occupied by EX-94164/AC	28
Read digital input data	Read digital input port data (8-bit)	29
Set debounce time	Set debounce timer of digital input signals	30
Enable interrupt	Enable interrupt by input(DI_0 ~ DI_7)	31
Disable interrupt	Disable digital input interrupt	32
Read interrupt status	Read channels which generate interrupt	33
Clear interrupt status register	Clear interrupt status register	33

5.3 Open card

Description:

Because the EX-94164/AC is PCI bus architecture and meets the plug and play design, the IRQ and base address are assigned by system BIOS directly. EX-94164/AC cards have to be initialized by this function before calling other functions.

Syntax:

C/C++(DOS)

```
WORD D_4164_Open (WORD cardNo);
```

C/C++ (Windows)

```
WORD W_4164_Open (WORD *ExistedCards);
```

Visual BASIC (Windows)

```
Function W_4164_Open (ByRef ExistedCards As Long) As Long
```

Delphi

```
Function W_4164_Open (var ExistedCards:Integer):Integer;
```

Argument:

cardNo: Card number set by jumper on card (DOS only)

existCards: Pointer to return a value shows how many EX-94164/AC cards installed. (Windows only)

Return Code:

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

5.4 Get Card's ID:

Description:

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

Syntax:

C/C++(DOS)

```
void D_4164_GetCardsID(WORD *CardsIDArray);
```

C/C++(Windows)

```
WORD W _4164_GetCardsID (WORD *CardsIDArray);
```

Visual BASIC (Windows)

```
Function W_4164_GetCardsID (ByRef CardsIDArray As Long)
         As Integer
```

Delphi

```
Function W_4164_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)

5.5 Get Driver Version

Description:

The EX-94164/AC card is driven by PCIDAQ. DLL driver. This function returns the version of PCIDAQ.DLL driver

Syntax:

C/C++(DOS):

```
void D_4164 _Version(char *version);
```

C/C++ (Windows)

```
Int W_4164_Version (void);
```

Visual BASIC (Windows)

```
Function W_4164_Version () As Long
```

Delphi

```
Function W_4164_Version ():Integer;
```

Argument:

version: This string pointer return the version of DOSDAQ.DLL (DOS only)

Return Code:

Return the version number in integer data format (Windows only)

5.6 Close card

Description:

The IRQ and base address of EX-94164/AC (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_4164 _Close (WORD cardNo);
```

C/C++ (Windows)

```
Void W_4164_Close (void);
```

Visual BASIC (Windows)

```
Function W_4164_Close ()
```

Delphi

```
Function W_4164_Close ();
```

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)

5.7 Get PCI Bus and Slot number

Description:

Get the PCI bus and slot number occupied by EX-94164/AC card

Syntax:

C/C++(DOS)

```
WORD D_4164_GetBusSlot (WORD cardNo, WORD *bus,WORD *slot);
```

C/C++ (Windows)

```
WORD W_4164_GetBusSlot (WORD cardNo, WORD *bus,WORD *slot);
```

Visual BASIC (Windows)

```
Function W_4164_GetBusSlot (ByVal cardNo As Long,
         ByRef bus As Long, ByRef slot As Long) As Long
```

Delphi

```
Function W_4164_GetBusSlot (cardNo:Integer;var bus:
         Integer;var slot: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 Read digital input data

Description:

This function is used to read data from digital input port. You can get 8-bit input data from EX-94164/AC by calling this function.

Syntax:

C/C++(DOS)

```
WORD D_4164_Read_Di (WORD cardNo,WORD portNo,WORD *DiData);
```

C/C++ (Windows)

```
WORD W_4164_Read_Di (WORD cardNo,WORD portNo,WORD *DiData);
```

Visual BASIC (Windows)

```
Function W_4164_Read_Di (ByVal cardNo As Long,
         ByVal portNo As Long, ByRef DiData As Long) As Long
```

Delphi

```
Function W_4164_Read_Di (cardNo:Integer;portNo:Integer;
         var DiData:Integer): Integer;
```

Argument:

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

portNo: Digital Input port number (0 ~ 7)

Didata: return digital input data

Return Code:

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

5.9 Set debounce time of digital inputs

Description:

The all digital input channels (DI_0 ~ DI_31) are grouped into 4 ports, each port can has an individually programmable digital debounce circuit which can filter the bounce of input signals

Syntax:

C/C++(DOS)

```
WORD D_4164_Set_DebounceMode (WORD cardNo,
     WORD DebounceMode);
```

C/C++ (Windows)

```
WORD W_4164_Set_DebounceMode (WORD cardNo,
     BYTE DebounceMode);
```

Visual BASIC (Windows)

```
Function W_4164_Set_DebounceMode (ByVal cardNo As Long, ByVal
         DebounceMode As Long) As Long
```

Delphi

```
Function W_4164_Set_DebounceMode (cardNo:Integer;
         DebounceMode: Integer):Integer;
```

Argument:

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

DebounceMode:

Bit0:	=0	Disable port #0 debounce
	=1	Enable port #0 debounce
Bit1:	=0	Disable port #1 debounce
	=1	Enable port #1 debounce
Bit2:	=0	Disable port #2 debounce
	=1	Enable port #2 debounce
Bit3:	=0	Disable port #3 debounce
	=1	Enable port #3 debounce

Bit4~Bit7: 0001~1111 debounce factor

Debounce period =10 msec x Debounce factor

For example: if Bit4~Bit7=0010, then debounce period=20ms*2=40ms

Return Code:

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

5.10 Enable Interrupt

Description:

Enable Interrupt of digital inputs

Syntax:

C/C++(DOS)

```
WORD D_4164_IntEnable (WORD cardNo,WORD IntMode,WORD
     IntMask,*UserIntServiceRoutine());
```

C/C++ (Windows)

```
WORD W_4164_IntEnable (WORD cardNo,WORD IntMode,
     WORD IntMask,User_Interrupt_HANDLER
     userIntServiceRoutine);
```

Visual BASIC (Windows)

```
Function W_4164_IntEnable (ByVal cardNo As Long,
         ByVal IntMode As Long, IntMask as Long,ByVal
         userIntServiceRoutine As Long) As Long
```

Delphi

```
Function W_4164_IntEnable (cardNo:Integer;IntMode:Integer;
         IntMask:Integer; userIntServiceRoutine:
         Pointer):Integer;Argument:
```

Argument:

cardNo: card number (1,2,3,4),It's set by jumper JP1 on card

IntMode: Triggering edge of input port 0's channels

Bit #n: =0 Rising edge interrupt of channel #n (DI_n) *(n=0~7)*
=1 Falling edge interrupt of channel #n (DI_n) *(n=0~7)*

IntMask : Interrupt mask of input port 0's channels

Bit #n: =0 Disable interrupt of channel #n (DI_n) *(n=0~7)*
=1 Enable interrupt of channel #n (DI_n) *(n=0~7)*

userIntServiceRoutine: User Interrupt service routine pointer will be called when interrupt occurs.

For C++:

void userIntServiceRoutine(WORD CardNo,WORD IntStatus);

For VB:

Sub UserInterruptRutine(ByVal CardNo As Long, ByVal IntStatus As Long)

For Delphi:

Procedure userIntServiceRoutine (CardNo:Word;IntStatus:Word);StdCall;

This function will pass CardNo and IntStatus parameters to user's service routine when interrupt occurred.

Where

IntStatus: Bit #n=1 Interrupted by DI_n
Bit #n=0 No interrupted by DI_n
CardNo: Card number

Return Code:

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

5.11 Disable Interrupt

Description:

Disable interrupt of input

Syntax:

C/C++(DOS)

```
WORD D_4164_IntDisable (WORD cardNo);
```

C/C++ (Windows)

```
Void W_4164_IntDisable (WORD cardNo);
```

Visual BASIC (Windows)

```
Function W_4164_IntDisable (ByVal cardNo As Long)
```

Delphi

```
Function W_4164_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)

5.12 Read Interrupt Status Register

Description:

Read the digital channel number which generate interrupt (DOS only)

Syntax:

C/C++(DOS)

```
WORD D_4164_Read_IntStatus (WORD cardNo,WORD *IntStatus);
```

Argument:

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

IntStatus:

Bit #n=1 Interrupted by DI_n *(n=0~7)*

Bit #n=0 No Interrupted by DI_n *(n=0~7)*

Return Code:

Error code (Please refer to DOSDAQ.H)

5.13 Clear Interrupt Status Register

Description:

Clear interrupt status register

Syntax:

C/C++(DOS)

```
WORD D_4164_Clear_IntStatus (WORD cardNo);
```

C/C++ (Windows)

```
WORDW_4164_Clear_IntStatus (WORD cardNo);
```

Visual BASIC (Windows)

```
Function W_4164_Clear_IntStatus (ByVal cardNo As Long)
         As Long
```

Delphi

```
Function W_4164_Clear_IntStatus (cardNo:Integer):Integer;
```

Argument:

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

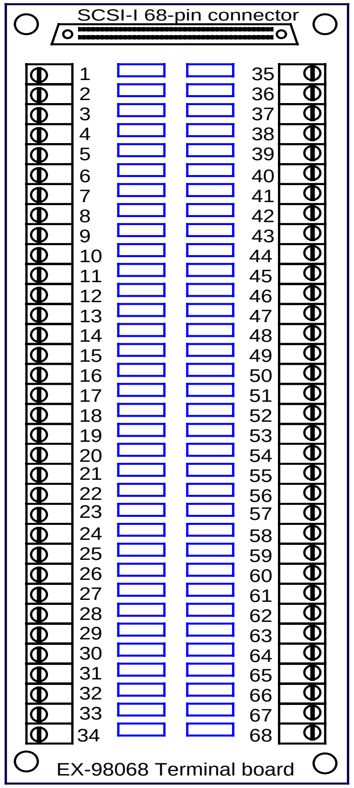
Return Code:

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

Chapter 6
EX-98068 Terminal board

EX-98068 digital input/output termination board features one DIN socket for easy maintenance, wiring, and installation. It provides 68 channels that are accessed through a SCSI-68 connector.

Each terminal pin is in serial with 0 ohms resistor to relative pin on the DIN connector. These resistors can be changed to the desired value to meet the requirement of your applications


SCSI-I 68-pin connector
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
EX-98068 Terminal board