PISO-P8R8/P8R8U
PISO-P8SSR8AC/P8SSR8UAC
PISO-P8SSR8DC/P8SSR8UDC

User Manual
Version 1.9
Oct. 2011

Warranty

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

Warning

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

Copyright

Trademark

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

Tables of Contents

1   Introduction

The PISO-P8R8U/P8SSR8UAC/P8SSR8UDC card is the new generation product that ICP DAS provides to meet RoHS compliance requirement. The new PISO-P8R8U/P8SSR8UAC/P8SSR8UDC card is designed as a drop-in replacement for the PISO-P8R8/P8SSR8AC/P8SSR8DC, and users can replace the PISO-P8R8/P8SSR8AC/P8SSR8DC by the PISO-P8R8U/P8SSR8UAC/ P8SSR8UDC directly without software/driver modification.

The PISO-P8R8/P8SSR8AC/P8SSR8DC is a PCI card supports both the +5 V PCI bus, while the PISO-P8R8U/P8SSR8UAC/P8SSR8UDC is a Universal PCI card supports both the +3.3 V/+5 V PCI bus. The PISO-P8R8 series is an 8 channels isolated input and output interface board for the PCI bus computers. The PISO-P8R8(U) card provides 8 electromechanical relay outputs and 8 optically isolated input, while PISO-P8SSR8(U)AC and PISO-P8SSR8(U)DC provide 8 solid state relay output and 8 optically isolated inputs. The PISO-P8R8, PISO-P8R8U, PISO-P8SSR8AC and PISO-P8SSR8DC can be used in various applications including contact closure, external voltage sensing, and loading sensing and designed for control and sensing applications. The PISO-P8R8(U), PISO-P8SSR8(U)AC and PISO-P8SSR8(U)DC has one 37-pin D-Type connector.

The PISO-P8R8U/P8SSR8UAC/P8SSR8UDC also adds a Card ID switch on-board. Users can set Card ID and then recognizes the board by the ID via software when using two or more PISO-P8R8U/P8SSR8UAC/P8SSR8UDC cards in one computer.

1.1 Features

- Support the +5 V PCI bus for PISO-P8R8/P8SSR8AC/P8SSR8DC

- Support the +3.3 V/+5 V PCI bus for PISO-P8R8U/P8SSR8UAC/ P8SSR8UDC

- 8 electromechanical relay output channels for PISO-P8R8(U)

- 8 AC-type solid state relay output channels for PSIO-P8SSR8(U)AC

- 8 DC-type solid state relay output channels for PISO-P8SSR8(U)DC

- 8 optically isolated digital input channels

- Card ID function (dip-switch) for PISO-P8R8U/ P8SSR8UAC/P8SSR8UDC

- Selectable DC signal input filter

- AC Signal input with filter

- Onboard relay output status LED indicators

- Supports Plug & Play to obtain I/O resources

- No more manually settings of I/O address and IRQ

1.2 Applications

This PISO-P8R8 series board may be applied to the following areas:

- Factory automation
- Product test
- Laboratory automation

1.3 Specifications

1.3.1 PISO-P8R8/P8R8U

Model Name	PISO-P8R8U	PISO-P8R8
Digital Input		
Isolation Voltage	3750 Vrms (Photo-couple)	
Channels	8	
Compatibility	5 V/TTL	
Input Voltage	Logic 1: AC/DC 5 ~ 24 V(AC 50 ~ 1 kHz) Logic 0: AC/DC 0 ~ 1 V	
Response Speed	Without Filter: 50 kHz (Typical) With Filter: 0.455 kHz (Typical)	
Input Impedance	1.2 KΩ, 1 W	1.2 KΩ, 0.5 W
Relay Output		
Channels	8	
Relay Type	SPST N.O.(Form A)	
Contact Rating	AC: 250 V @1.6 A DC: 30 V @5 A	
Operating Time	6 ms (typical)	
Release Time	3 ms (typical)	
Insulation Resistance	1000 MΩ @ 500 V_{DC}	
Life	Mechanical: 20000000 ops. Electrical: 100000 ops.	
General		
Bus Type	3.3 V/5 V Universal PCI, 32-bit, 33 MHz	5 V PCI, 32-bit, 33 MHz
Data Bus	8-bit	
Card ID	Yes (4-bit)	No
I/O Connector	Female DB37 x 1	
Dimensions (L x W x D)	149 mm x 105 mm x 22 mm	
Power Consumption	300 mA @ +5 V	
Operating Temperature	0 ~ 60 °C	
Storage Temperature	-20 ~ 70 °C	
Humidity	5 ~ 85% RH, non-condensing	

1.3.2 PISO-P8SSR8AC/P8SSR8UAC

Model Name	PISO-P8SSR8UAC	PISO-P8SSR8AC
Digital Input		
Isolation Voltage	3750 Vrms (Photo-couple)	
Channels	8	
Compatibility	5 V/TTL	
Input Voltage	Logic 1:AC/DC 5 ~ 24 V(AC 50 ~ 1 kHz) Logic 0: AC/DC 0 ~ 1 V	
Response Speed	Without Filter: 50 kHz (Typical) With Filter: 0.455 kHz (Typical)	
Relay Output		
Channels	8	
Relay Type	SPST N.O.(Form A)	
Contact Rating	AC:24 ~ 265 V	
Operating Time	1 ms (typical)	
Release Time	0.5 cycle + 1 ms (typical)	
Insulation Resistance	1000 MΩ @ 500 V_{DC}	
Life	Mechanical: 20000000 ops. Electrical: 100000 ops.	
General		
Bus Type	3.3 V/5 V Universal PCI, 32-bit, 33 MHz	5 V PCI, 32-bit, 33 MHz
Data Bus	8-bit	
Card ID	Yes (4-bit)	No
I/O Connector	Female DB37 x 1	
Dimensions (L x W x D)	149 mm x 105 mm x 22 mm	
Power Consumption	300 mA @ +5 V	
Operating Temperature	0 ~ 60 °C	
Storage Temperature	-20 ~ 70 °C	
Humidity	5 ~ 85% RH, non-condensing	

1.3.3 PISO-P8SSR8DC/P8SSR8UDC

Model Name	PISO-P8SSR8UDC	PISO-P8SSR8DC
Digital Input		
Isolation Voltage	3750 Vrms (Photo-couple)	
Channels	8	
Compatibility	5 V/TTL	
Input Voltage	Logic 1:AC/DC 5 ~ 24 V(AC 50 ~ 1 kHz) Logic 0: AC/DC 0 ~ 1 V	
Response Speed	Without Filter: 50 kHz (Typical) With Filter: 0.455 kHz (Typical)	
Relay Output		
Channels	8	
Relay Type	SPST N.O.(Form A)	
Contact Rating	DC: 3 ~ 30 V	
Operating Time	1 ms (typical)	
Release Time	1 ms (typical)	
Insulation Resistance	1000 MΩ @ 500 V_{DC}	
Life	Mechanical: 20000000 ops. Electrical: 100000 ops.	
General		
Bus Type	3.3 V/5 V Universal PCI, 32-bit, 33 MHz	5 V PCI, 32-bit, 33 MHz
Data Bus	8-bit	
Card ID	Yes (4-bit)	No
I/O Connector	Female DB37 x 1	
Dimensions (L x W x D)	149 mm x 105 mm x 22 mm	
Power Consumption	300 mA @ +5 V	
Operating Temperature	0 ~ 60 °C	
Storage Temperature	-20 ~ 70 °C	
Humidity	5 ~ 85% RH, non-condensing	

1.4 Order Description

- **PISO-P8R8U**
  Universal PCI, 8 channels isolated digital input, 8 channels relay output board

- **PISO-P8SSR8UAC**
  Universal PCI, 8 channels isolated digital input, 8 channels AC-Type solid state relay output board

- **PISO-P8SSR8UDC**
  Universal PCI, 8 channels isolated digital input, 8 channels DC-Type solid state relay output board

- **PISO-P8R8**
  PCI bus, 8 channels isolated digital input, 8 channels relay output board

- **PISO-P8SSR8AC**
  PCI bus, 8 channels isolated digital input, 8 channels AC-Type solid state relay output board

- **PISO-P8SSR8DC**
  PCI bus, 8 channels isolated digital input, 8 channels DC-Type solid state relay output board

1.4.1 Options

- DN-37: I/O connector block with DIN-Rail mounting and 37-pin D-type Connector

- DB-37: 37-pin D-type connector pin to pin screw terminal for any 37 pin D-type connector of I/O board

1.5 Product Check List

The shipping package includes the following items:

- One PISO-P8R8(U)/P8SSR8(U)AC/P8SSR8(U)DC series card
- One companion floppy diskette or CD
- One Quick Start Guide

It is recommended that you read the Quick Start Guide first. All the necessary and essential information is given in the Quick Start Guide, including:

- Where to get the software driver, demo programs and other resources.
- How to install the software.
- How to test the card.

Attention!

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

1.6 PCI Data Acquisition Family

We provide a family of PCI-BUS data acquisition cards. These cards can be divided into three groups as follows:

1. **PCI-series: first generation, isolated or non-isolated cards**
   PCI-1002/1202/1800/1802/1602: multi-function family, non-isolated
   PCI-P16R16/P16C16/P16POR16/P8R8: D/I/O family, isolated
   PCI-TMC12: timer/counter card, non-isolated

2. **PIO-series: cost-effective generation, non-isolated cards**
   PIO-823/821: multi-function family
   PIO-D168/D144/D96/D64/D56/D48/D24: D/I/O family
   PIO-DA16/DA8/DA4: D/A family

3. **PISO-series: cost-effective generation, isolated cards**
   PISO-813: A/D card
   PISO-P32C32/P64/C64: D/I/O family
   PISO-P8R8/P8SSR8AC/P8SSR8DC: D/I/O family
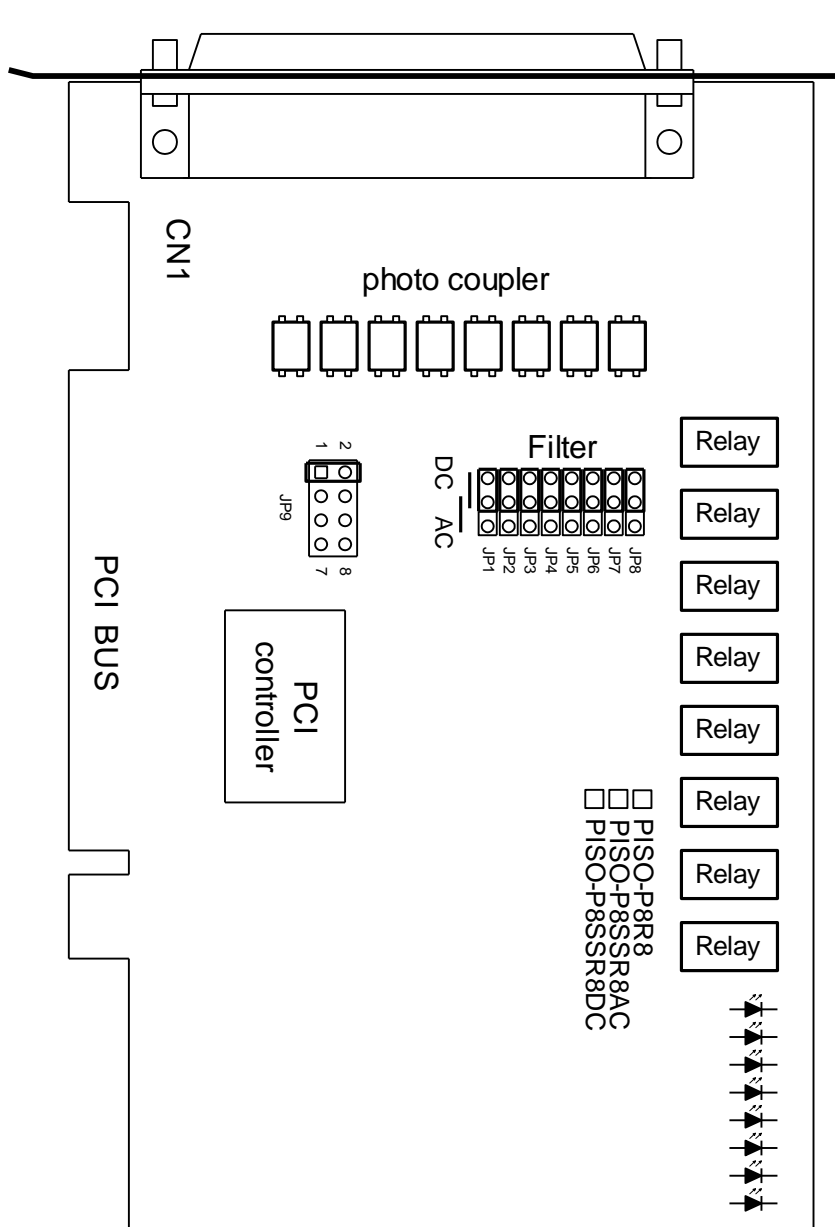   PISO-730: D/I/O card
   PISO-DA2: D/A card

2 Hardware Configuration

2.1 Board Layout
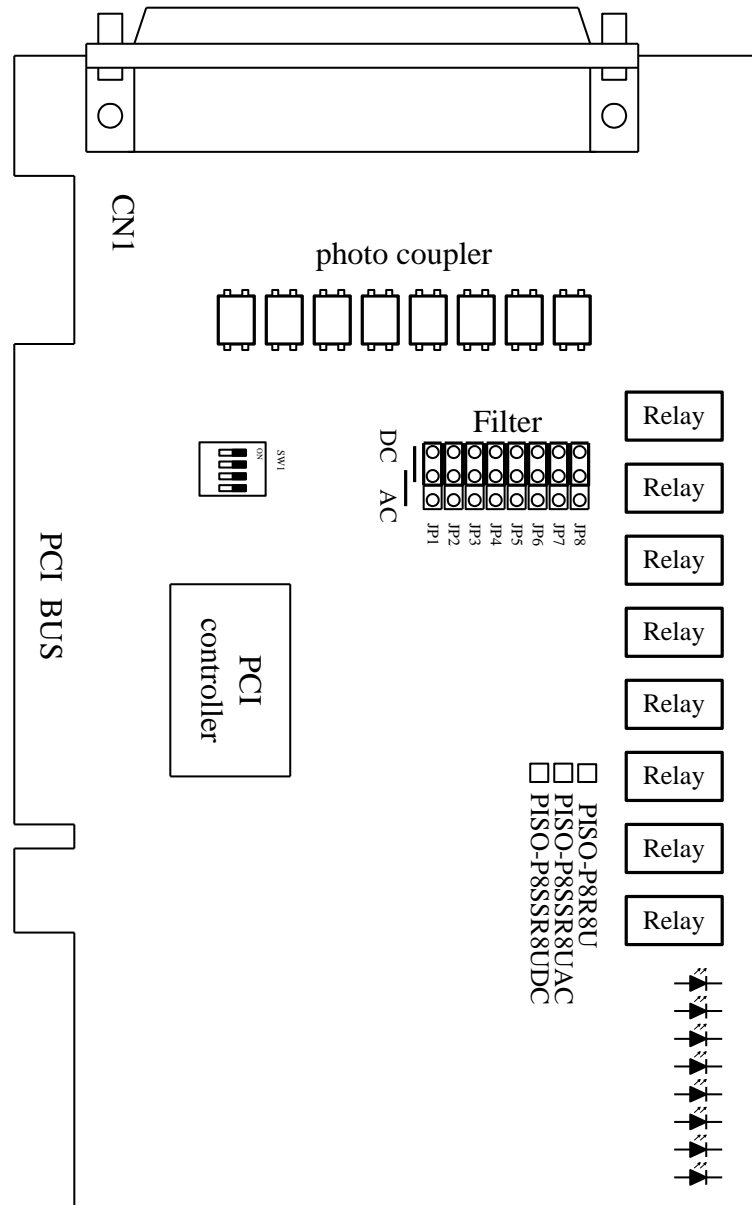
■ PISO-P8R8/P8SSR8AC/P8SSR8DC:

CN1: 8 channels isolated D/I and 8 channels isolated D/O

JP1 ~ JP8: Filter

JP9: Reserved

■ PISO-P8R8U/P8SSR8UAC/P8SSR8UDC:

CN1: 8 channels isolated D/I and 8 channels isolated D/O

JP1 ~ JP8: Filter

SW1: Card ID function (dip-switch)

2.2 I/O Operation

2.2.1 Isolated Input Architecture

The PISO-P8R8(U) (PISO-P8SSR8(U)AC and PISO-P8SSR8(U)DC) provides 8 channels isolated digital input. Each of the isolated digital input accepts voltages from 3.5~30 V_{DC}.

Each input channel provides a selectable RC filter by jumper setting. The single-pole, RC filter with 1.2 ms time constant. User has to short the AC filter pin2-pin3 of the corresponding jumper when using AC signal.

■ The block diagram of isolated input is given as follows:
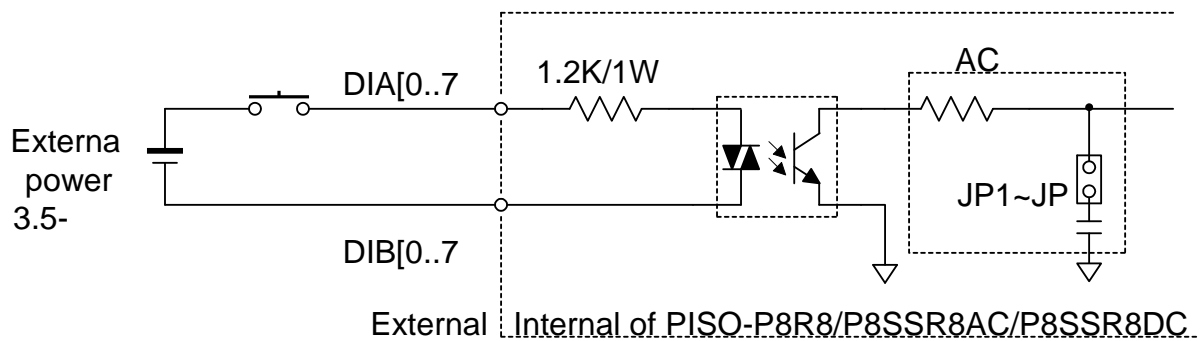
■ JP1~JP8 Filter Selector as follows:

Jumper	Channels
JP1	DIA0-DIB0
JP2	DIA1-DIB1
JP3	DIA2-DIB2
JP4	DIA3-DIB3
JP5	DIA4-DIB4
JP6	DIA5-DIB5
JP7	DIA6-DIB6
JP8	DIA7-DIB7
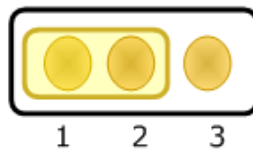
Without filter for DC signal **(Default Setting)**
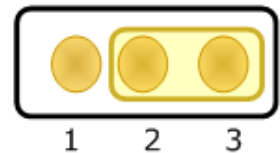
With filter for AC signal (or for DC signal)

> **!** Note: For rejecting noise purpose, the AC filter is optional when using DC input signal.

2.2.2 Isolated Output Architecture

When the PC is power-up, all states of output relay are "open". The enable/disable of output operation is controlled by the RESET\ signal. Refer to Sec. 3.3.1 for more information about RESET\ signal.

- The RESET\ is in Low-state → all output operation are disable
- The RESET\ is in High-state → all output operation are enable

■   The block diagram of isolated output is given as follows:
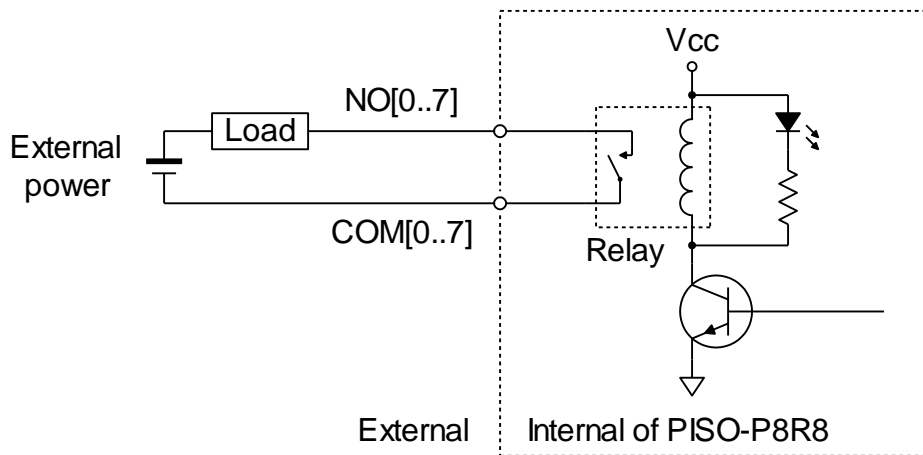
The architecture of PISO-P8SSR8(U)AC (PISO-P8SSR8(U)DC) is similar to the PISO-P8R8(U). The difference between PISO-P8R8(U) and PISO-P8SSR8(U)AC (PISO-P8SSR8(U)DC) is that replace the relay by the SSR (solid state relay). The SSR has several special properties as listing: (For more detail specifications please refer to Sec.1.3)

- Silence
- Quick response
- High reliability, long life & maintenance free
- A longer life time due to contactless system
- No malfunction caused by vibration and shock
- No degradation in performance cause by dust, gas, etc.

2.2.3 Output waveform (at Resistive Load)

Relay

Source voltage of load — 0V

Input signal — ON

Load current — 0A

Figure(a)

DC-Type SSR

Source voltage of load — 0V

Input signal — ON

Load current — 0A

Figure(b)

AC-Type SSR

Source voltage of load — 0V

Input signal — ON

Load current — 0A (Note)

Figure(c)
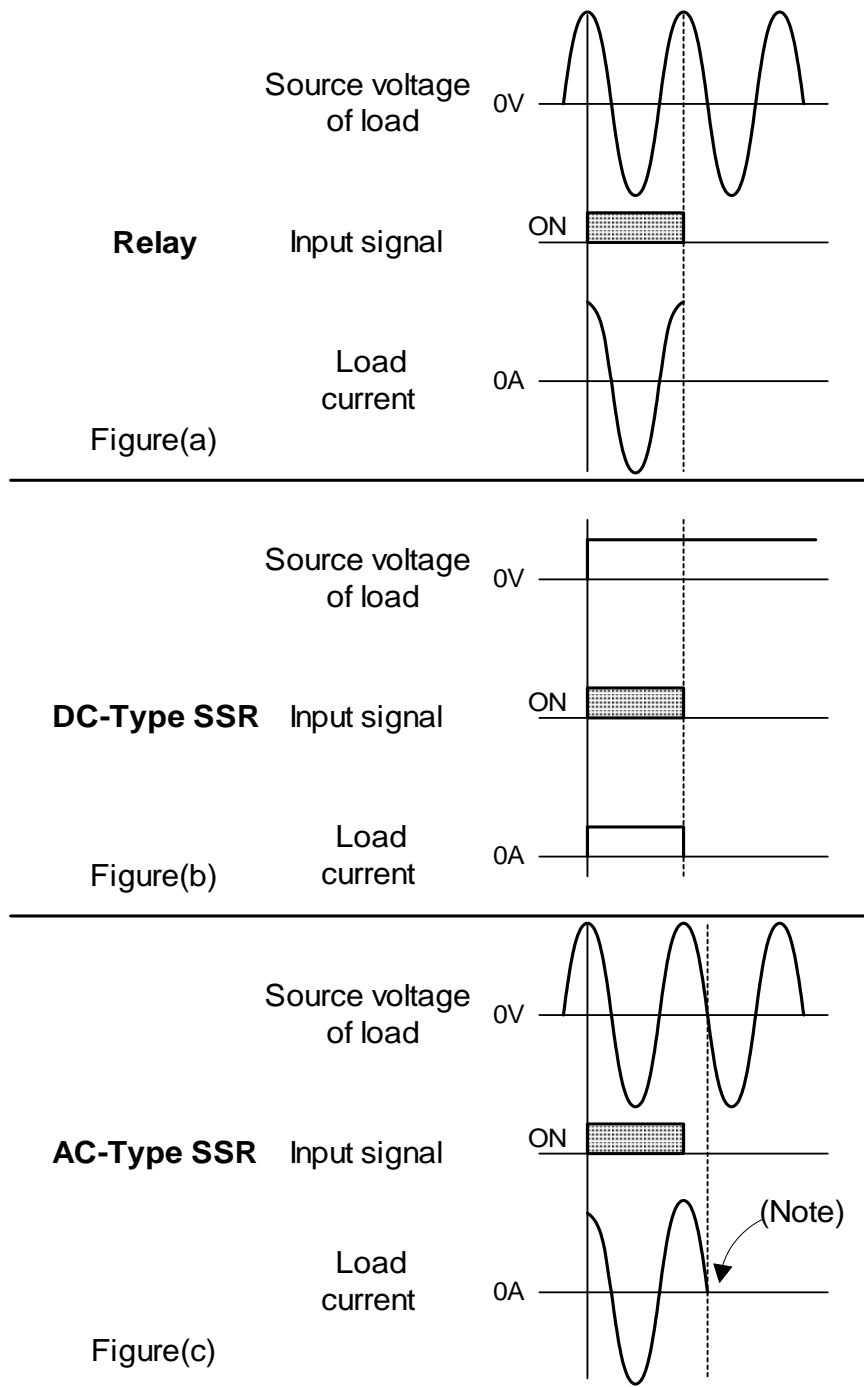
> **!** ***Note***: The AC-Type SSR is a non zero-crossing SSR. It uses a phototriac coupler to isolate the input from the output. When the input signal is activated, the output immediately turns on, since there is no zero-crossing detector circuit. **The load current is maintained by the triac's latching effect after the input signal is deactivated, until the AC load voltage crosses zero.(Refer to Figure(c).)**
> **(For more detail information about SSR, please refer to web site www.fujitsufta.com)**

2.3   Daughter Boards

2.3.1   DB-37

Direct connection board

■   37-pin D-type connector pin to pin screw terminal for any 37-pin D-type connector of I/O board
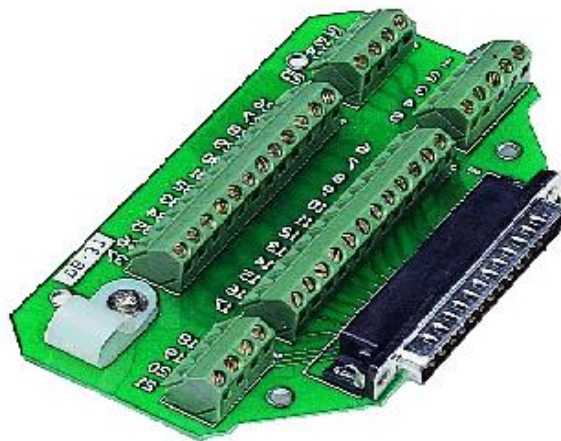
2.3.2   DN-37

I/O connector block with DN-Rail mounting

■   Two 37-pin D-type connector (one for extension)
■   Pin to pin screw terminal for I/O connector
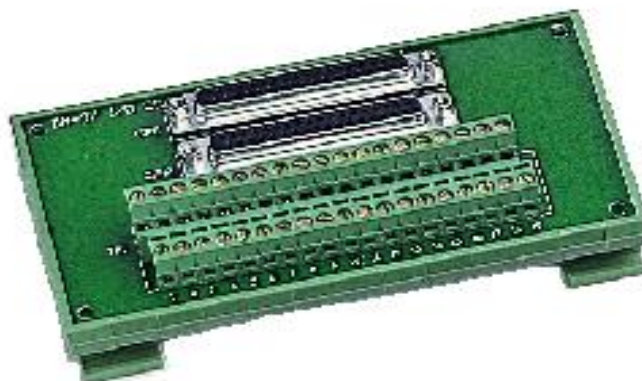
2.4   Pin Assignment

■   **CON1: Isolated I/O connector (37-pin of D-type female connector)**

CON1

Pin	Signal		Pin	Signal
1	NO0		20	NO3
2	COM0		21	COM3
3	x		22	x
4	NO1		23	NO4
5	COM1		24	COM4
6	x		25	NO5
7	NO2		26	COM5
8	COM2		27	NO6
9	x		28	COM6
10	NO7		29	x
11	COM7		30	DIB0
12	DIA0		31	DIB1
13	DIA1		32	DIB2
14	DIA2		33	DIB3
15	DIA3		34	DIB4
16	DIA4		35	DIB5
17	DIA5		36	DIB6
18	DIA6		37	DIB7
19	DIA7			
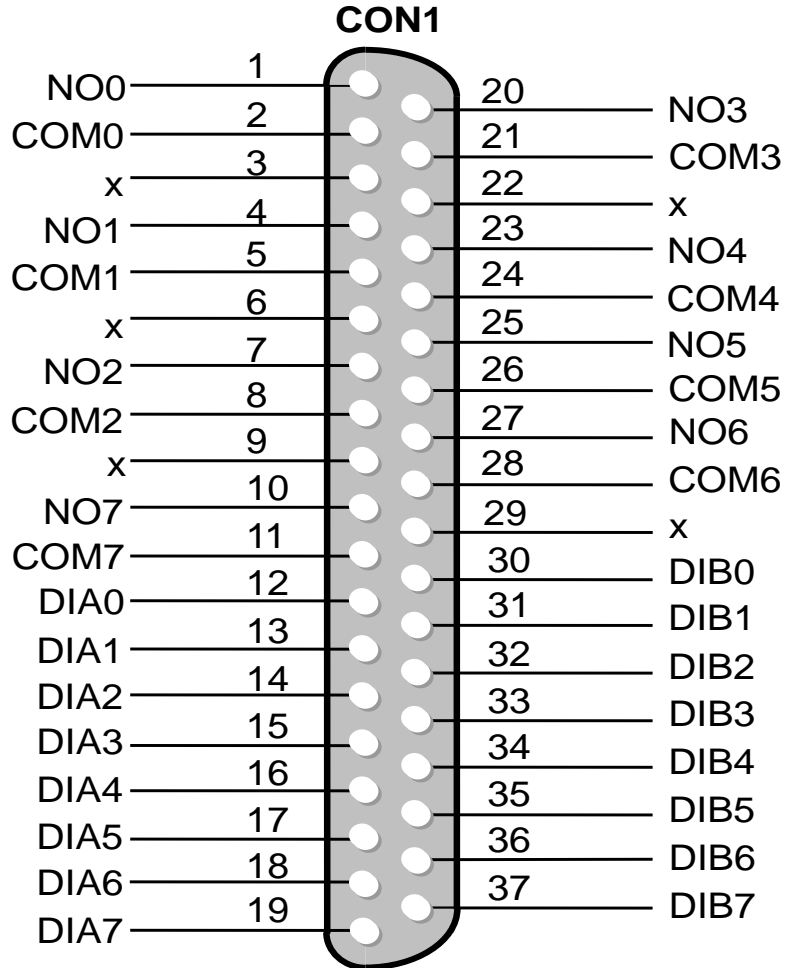
2.5 Jumper and Dip-Switch Settings

2.5.1 JP1-JP8 Filter Selector

Jumper	Channel
JP1	DIA0−DIB0
JP2	DIA1−DIB1
JP3	DIA2−DIB2
JP4	DIA3−DIB3
JP5	DIA4−DIB4
JP6	DIA5−DIB5
JP7	DIA6−DIB6
JP8	DIA7−DIB7
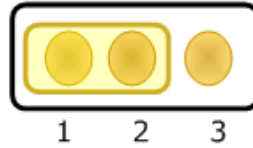
Without filter for DC signal **(Default Setting)**
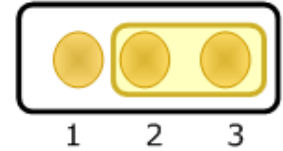
With filter for AC signal (or for DC signal)

! *Note*: **For rejecting noise purpose, the AC filter is optional when using DC input signal.**

2.5.2 JP9 Reserved

JP9
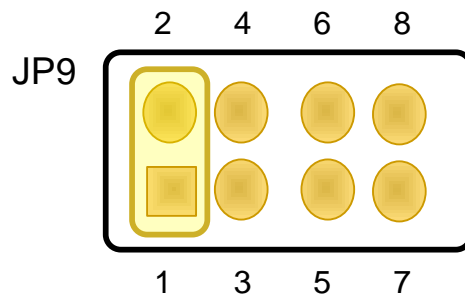
Note: Reserved

2.5.3   Card ID Dip-Switch (SW1)

The PISO-P8R8U/P8SSR8UAC/P8SSR8UDC also adds a Card ID switch on-board. Users can set Card ID and then recognizes the board by the ID via software when using two or more PISO-P8R8U cards in one computer.

SW1

ID0   ID1   ID2   ID3   (Default Setting)
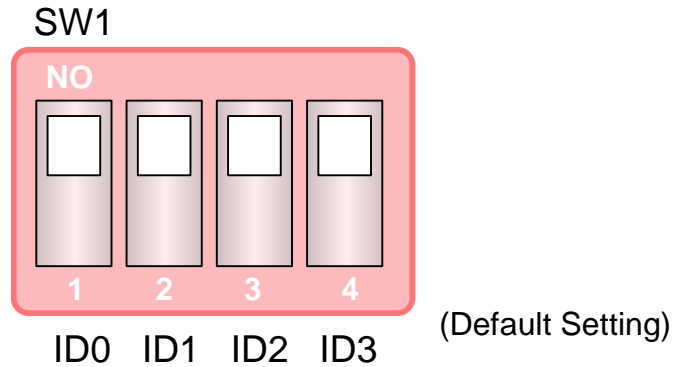
Card ID	1 ID0	2 ID1	3 ID2	4 ID3
(*) 0	ON	ON	ON	ON
1	OFF	ON	ON	ON
2	ON	OFF	ON	ON
3	OFF	OFF	ON	ON
4	ON	ON	OFF	ON
5	OFF	ON	OFF	ON
6	ON	OFF	OFF	ON
7	OFF	OFF	OFF	ON
8	OFF	OFF	OFF	ON
9	OFF	ON	ON	OFF
10	ON	OFF	ON	OFF
11	OFF	OFF	ON	OFF
12	ON	ON	OFF	OFF
13	OFF	ON	OFF	OFF
14	ON	OFF	OFF	OFF
15	OFF	OFF	OFF	OFF

(*) Default setting

3   I/O Control Register

3.1   How to Find the I/O Address

The plug & play BIOS will assign a appropriate I/O address to each PIO/PISO series card during the power-on stage. The fixed IDs of PIO/PISO series cards are given as following:

For PISO-P8R8(U)/P8SSR8(U)AC/P8SSR8(U)DC			
Rev 1.0		**Rev 2.0**	
Vendor ID	0xE159	Vendor ID	0xE159
Device ID	0x02	Device ID	0x01
Sub-vendor ID	0x80	Sub-vendor ID	0x4A80 0x4200
Sub-device ID	0x08	Sub-device ID	0x00
Sub-aux ID	0x30	Sub-aux ID	0x30

We provide all necessary functions as follows:

1. **PIO_DriverInit(&wBoard, wSubVendor, wSubDevice, wSubAux)**
   For detailed PIO_DriverInit function information, refer to Section 3.1.1.

2. **PIO_GetConfigAddressSpace(wBoardNo,*wBase,*wIrq, *wSubVendor, *wSubDevice, *wSubAux, *wSlotBus, *wSlotDevice)**
   For detailed PIO_GetConfigAddressSpace function information, refer to Section 3.1.2.

3. **Show_PIO_PISO(wSubVendor, wSubDevice, wSubAux)**
   For detailed Show_PIO_PISO function information, refer to Section 3.1.3.

All functions are defined in PIO.H. Refer to Chapter 5 for more information. The important driver information is given as follows:

1. Resource-allocated information:
- wBase : BASE address mapping in this PC
- wIrq: IRQ channel number allocated in this PC

2. PIO/PISO identification information:
- wSubVendor: subVendor ID of this board
- wSubDevice: subDevice ID of this board
- wSubAux: subAux ID of this board

3. PC's physical slot information:
- wSlotBus: hardware slot ID1 in this PC's slot position
- wSlotDevice: hardware slot ID2 in this PC's slot position

The **PIO_PISO.EXE utility program** can be used to detect and display the details of PIO/PISO cards installed in this PC. Refer to Section 5.3 for more information.

3.1.1 PIO_DriverInit

PIO_DriverInit(&wBoards, wSubVendor,wSubDevice,wSubAux)

- wBoards=0 to N → number of boards found in this PC
- wSubVendor → subVendor ID of board to find
- wSubDevice → subDevice ID of board to find
- wSubAux → subAux ID of board to find

This function can detect all PIO/PISO series card in the system. It is implemented based on the PCI plug & play mechanism-1. It will find all PIO/PISO series cards installed in this system and save all their resource in the library.

■ Sample program 1: Detect all PISO-P8R8(U) series card in the PC

```
wSubVendor=0x80; wSubDevice=8; wSubAux=0x30; /*for PISO-P8R8 */
wRetVal=PIO_DriverInit(&wBoards, wSubVendor,wSubDevice,wSubAux);
printf("Threr are %d PISO-P8R8(SSR8AD/SSR8DC) Cards in this PC\n",wBoards);
/* step2: save resource of all PISO-P8R8(SSR8AC/SSR8DC) cards installed in
this PC */
for (i=0; i<wBoards; i++)
  {
   PIO_GetConfigAddressSpace(i,&wBase,&wIrq,&wID1,&wID2,&wID3,
                                 &wID4,&wID5);
   printf("\nCard_%d: wBase=%x, wIrq=%x", i,wBase,wIrq);
   wConfigSpace[i][0]=wBaseAddress; /*save all resource of this card */
   wConfigSpace[i][1]=wIrq;             /* save all resource of this card  */
  }
```

■   Sample program 2: Detect all PIO/PISO in the PC (refer to Section 5.3 for more information)

```
wRetVal=PIO_DriverInit(&wBoards,0xff,0xff,0xff);  /*find all PIO_PISO*/
printf("\nThrer are %d PIO_PISO Cards in this PC",wBoards);
if (wBoards==0 ) exit(0);

printf("\n---------------------------------------------------");
for(i=0; i<wBoards; i++)
  {
  PIO_GetConfigAddressSpace(i,&wBase,&wIrq,&wSubVendor,
          &wSubDevice,&wSubAux,&wSlotBus,&wSlotDevice);

  printf("\nCard_%d:wBase=%x,wIrq=%x,subID=[%x,%x,%x],
          SlotID=[%x,%x]",i,wBase,wIrq,wSubVendor,wSubDevice,
          wSubAux,wSlotBus,wSlotDevice);
  printf(" --> ");
  ShowPioPiso(wSubVendor,wSubDevice,wSubAux);
  }
```

3.1.2   PIO_GetConfigAddressSpace

PIO_GetConfigAddressSpace(wBoardNo,*wBase,*wIrq, *wSubVendor,
                        ***wSubDevice, *wSubAux, *wSlotBus, *wSlotDevice)**

- wBoardNo=0 to N → totally N+1 boards found by PIO_DriveInit(….)
- wBase   → base address of the board control word
- wIrq   → allocated IRQ channel number of this board
- wSubVendor   → subVendor ID of this board
- wSubDevice   → subDevice ID of this board
- wSubAux   → subAux ID of this board
- wSlotBus   → hardware slot ID1 of this board
- wSlotDevice   → hardware slot ID2 of this board

The user can use this function to save resource of all PIO/PISO cards installed in this system. Then the application program can control all functions of PIO/PISO series card directly.

■   The sample program source is given as follows:

```
/* step1: detect all PISO-P8R8(SSR8AC/SSR8DC) cards first */
 wSubVendor=0x80; wSubDevice=8; wSubAux=0x30;  /*for PISO-P8R8 */
wRetVal=PIO_DriverInit(&wBoards, wSubVendor,wSubDevice,wSubAux);
 printf("Threr are %d PISO-P8R8(SSR8AC/SSR8DC) Cards in this PC\n",wBoards);
/* step2: save resource of all PISO-P8R8(SSR8AC/SSR8DC) cards installed in
this PC */
 for (i=0; i<wBoards; i++)
   {
   PIO_GetConfigAddressSpace(i,&wBase,&wIrq,&t1,&t2,&t3,&t4,&t5);
   printf("\nCard_%d: wBase=%x, wIrq=%x", i,wBase,wIrq);
  wConfigSpace[i][0]=wBaseAddress;   /*save all resource of this card */
  wConfigSpace[i][1]=wIrq;               /*save all resource of this card */
  }
/* step3: control the PISO-P8R8(SSR8AC/SSR8DC) directly */
wBase=wConfigSpace[0][0];               /*get base address the card_0 */
outport(wBase,1);                      /*enable all D/I/O operation of card_0*/

 wBase=wConfigSpace[1][0];     /* get base address the card_1*/
 outport(wBase,1);                /* enable all D/I/O operation of card_1 */
```

3.1.3 Show_PIO_PISO

Show_PIO_PISO(wSubVendor,wSubDevice,wSubAux)

- wSubVendor → subVendor ID of board to find
- wSubDevice → subDevice ID of board to find
- wSubAux   → subAux ID of board to find
-

This function will show a text string for this special subIDs. This text string is the same as that defined in PIO.H

■ The demo program is given as follows:

```
wRetVal=PIO_DriverInit(&wBoards,0xff,0xff,0xff);  /*find all PIO_PISO*/
printf("\nThrer are %d PIO_PISO Cards in this PC",wBoards);
if (wBoards==0 ) exit(0);

printf("\n----------------------------------------------------");
for(i=0; i<wBoards; i++)
  {
  PIO_GetConfigAddressSpace(i,&wBase,&wIrq,&wSubVendor,
         &wSubDevice,&wSubAux,&wSlotBus,&wSlotDevice);

  printf("\nCard_%d:wBase=%x,wIrq=%x,subID=[%x,%x,%x],
         SlotID=[%x,%x]",i,wBase,wIrq,wSubVendor,wSubDevice,
         wSubAux,wSlotBus,wSlotDevice);
  printf(" --> ");
  ShowPioPiso(wSubVendor,wSubDevice,wSubAux);
  }
```

3.2 The Assignment of I/O Address

The plug & play BIOS will assign the appropriate I/O address to the PIO/PISO series card. If there is only one PIO/PISO board, the board will be identified as card_0. If there are two PIO/PISO boards in the system, it is very difficult to identify which board is card_0. The software driver can support a maximum of 16 boards. Therefore, the user can install 16 PIO/PSIO series cards in one PC system. Details of how to locate and identify card_0 and card_1 are provided below:

The simplest way to identify which card is card_0 is to use wSlotBus and wSlotDevice functions as follows:

Step 1: Remove all PISO-P8R8(U)/P8SSR8(U)AC/P8SSR8(U)DC series cards from the PC.

Step 2: Install a single PISO-P8R8(U)/P8SSR8(U)AC/P8SSR8(U)DC series cards into the PCI_slot1 in the PC, then run PIO_PISO.EXE and record the wSlotBus1 and wSlotDevice1 information.

Step 3: Remove all PISO-P8R8(U)/P8SSR8(U)AC/P8SSR8(U)DC series cards from the PC.

Step 4: Install a single PISO-P8R8(U)/P8SSR8(U)AC/P8SSR8(U)DC series cards into the PCI_slot2 in the PC, then run PIO_PISO.EXE and record the wSlotBus2 and wSlotDevice2 information.

Step 5: Repeat Steps 3 and 4 for all PCI_slots and record all wSlotBus and wSlotDevice information.

The records may be similar to the following table:

Table 3-1

PC's PCI slot	WslotBus	wSlotDevice
Slot_1	0	0x07
Slot_2	0	0x08
Slot_3	0	0x09
Slot_4	0	0x0A
PCI-BRIDGE		
Slot_5	1	0x0A
Slot_6	1	0x08
Slot_7	1	0x09
Slot_8	1	0x07

The above procedure is used to record all the wSlotBus and wSlotDevice information for the PC. These values will be mapped to this PC's physical slots and this mapping will not be changed for any PIO/PISO cards. Therefore, this information can be used to identify the specific PIO/PISO card using the following steps:

Step1: Using the wSlotBus and wSlotDevice information from Table 3-1 enter the board number into the PIO_GetConfigAddressSpace(…) function to get the information for a specific card information, especially wSlotBus and wSlotDevice details.

Step2: Identify the specific PIO/PISO card by comparing the data of the wSlotBus and wSlotDevice from Step 1.

! **Note that, normally, the card that is installed in slot 0 is card0 and the card installed in slot1 is card1 for PIO/PISO series cards.**

3.3   The I/O Address Map

The I/O address of PIO / PISO series card is automatically assigned by the main board ROM BIOS. The I/O address can also be re-assigned by user, but **It is strongly recommended that the I/O address is not changed by user. The plug&play BIOS will assign an appropriate I/O address to each PIO/PISO series card.** The I/O address of PISO-P8R8(U)/P8SSR8(U)AC/P8SSR8(U)DC series cards are as follows, and are based on the base address of each card.

Address	Read	Write
wBase+0	RESET\ control register	Same
wBase+0xc0	DI0~DI7	DO0~DO7
WBase+0xf0	Read the Card ID	-

Note. Refer to <u>Sec. 3.1</u> **for more information about wBase.**

3.3.1   RESET\ Control Register

(Read/Write): wBase+0

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

Note. Refer to <u>Sec. 3.1</u> **for more information about wBase.**

When the PC is first powered-on, the RESET\ signal is in the Low-state. **This will disable all D/I/O operations.** The user has to set the RESET\ signal to the High-state before performing any D/I/O commands.

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

3.3.2   I/O Data Register

(Read): wBase+0xC0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0

(Write): wBase+0xC0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0

Note. Refer to <u>Sec. 3.1</u> for more information about wBase.

```
outportb(wBase+0xc0,0xff);            /* write 0xff to DO0~DO7  */
DiValue=inportb(wBase+0xc0);          /* read states from DI0~DI7 */
```

3.3.2   Card ID Register

(Read): wBase+0xF0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ID7	ID6	ID5	ID4	0	0	0	0

Note. Refer to <u>Sec. 3.1</u> for more information about wBase.

```
wCardID = (inportb (wBase+0xf0) & 0xf0) >> 4   /* read card ID */
```

> **!**   **Note: The Card ID function is only supported by the PISO-P8R8U/
> P8SSR8UAC/P8SSR8UDC (Ver. 2.2 or above)**

4 Software Installation

The PISO-P8R8 series can be used in DOS and Windows 98/ME/NT/2K and 32-bit/64-bit Windows XP/2003/Vista/7. The recommended installation procedure for windows is given in Sec. 4.1 ~ 4.3. Or refer to Quick Start Guide (CD:\NAPDOS\PCI\PISO-DIO\Manual\QuickStart\).
http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/piso-dio/manual/quickstart/

4.1 Software Installing Procedure

■ UniDAQ SDK driver (32-bit/64-bit Windows XP/2003/Vista/7):

Step 1: Insert the companion CD into the CD-ROM drive and after a few seconds the installation program should start automatically. If it doesn't start automatically for some reason, double-click the **AUTO32.EXE** file in the **NAPDOS** folder on this CD.

Step 2: Click the item: "**PCI Bus DAQ Card**".

Step 3: Click the item: "**UniDAQ**".

Step 4: Click the item: "**DLL for Windows 2000 and XP/2003/Vista 32-bit**".

Step 5: Double-Click "**UniDAQ_Win_Setup_x.x.x.x_xxxx.exe**" file in the **Driver** folder.

■ Windows driver (Windows 98/NT/2K and 32-bit Windows XP/2003/Vista/7):

Step 1: Insert the companion CD into the CD-ROM drive and after a few seconds the installation program should start automatically. If it doesn't start automatically for some reason, double-click the **AUTO32.EXE** file in the **NAPDOS** folder on this CD.

Step 2: Click the item: "**PCI Bus DAQ Card**".

Step 3: Click the item: "**PISO-DIO**".

Step 4: Click the item "**DLL and OCX for Windows 98/NT/2K/XP/2003**".

Step 5: Double-Click "**PISO_DIO_Win_xxx.exe**" file in the **Driver** folder.

The setup program will then start the driver installation and copy the relevant files to the specified directory and register the driver on your computer. The directory where the drive is stoned is different for different windows versions, as shown below.

■ **Windows 64-bit Windows XP/2003/Vista/7:**

The UniDAQ.DLL file will be copied into the C:\WINNT\SYSTEM32 folder
The NAPWNT.SYS and UniDAQ.SYS files will be copied into the
C:\WINNT\SYSTEM32\DRIVERS folder

> ! **For more detailed UniDAQ.DLL function information, please refer to UniDAQ SDK user manual** (CD:\NAPDOS\PCI\UniDAQ\Manual\)**.**
> http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/unidaq/maunal/

■ **Windows NT/2K and 32-bit Windows XP/2003/Vista/7:**

The PISODIO.DLL file will be copied into the C:\WINNT\SYSTEM32 folder
The NAPWNT.SYS and PISO.SYS files will be copied into the
C:\WINNT\SYSTEM32\DRIVERS folder

■ **Windows 95/98/ME:**

The PISODIO.DLL and PISODIO.Vxd files will be copied into the
C:\Windows\SYSTEM folder

> ! **For more detailed PISODIO.DLL function information, please refer to PISO-DIO_Win32_SDK_Manual.pdf**(CD:\NAPDOS\PCI\PISO-DIO\Manual\)**".**
> http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/piso-dio/manual/

4.2 PnP Driver Installation

Power off the computer and install the PISO-P8R8(U)/P8SSR8(U)AC/P8SSR8(U)DC series cards. Turn on the computer and Windows 98/Me/2K and 32-bit/64-bit Windows XP/2003/Vista/7 should automatically defect the new PCI device(s) and then ask for the location of the driver files for the hardware. If a problem is encountered during installation, refer to the PnPinstall.pdf file for more information.
 (CD:\Napdos/pci/manual/pci_isa_pnp_driver_installation_in_win9x_2k_xp.pdf)

4.3   Confirm the Successful Installation

Make sure the PISO-P8R8(U)/P8SSR8(U)AC/ P8SSR8(U)DC series cards installed are correct on the computer as follows:

Step 1: Select **"Start"** → **"Control Panel"** and then double click the **"System"** icon on Windows.

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

Step 3: Check the PISO-P8R8(U)/P8SSR8(U)AC/ P8SSR8(U)DC series card which listed correctly or not, as illustrated below.
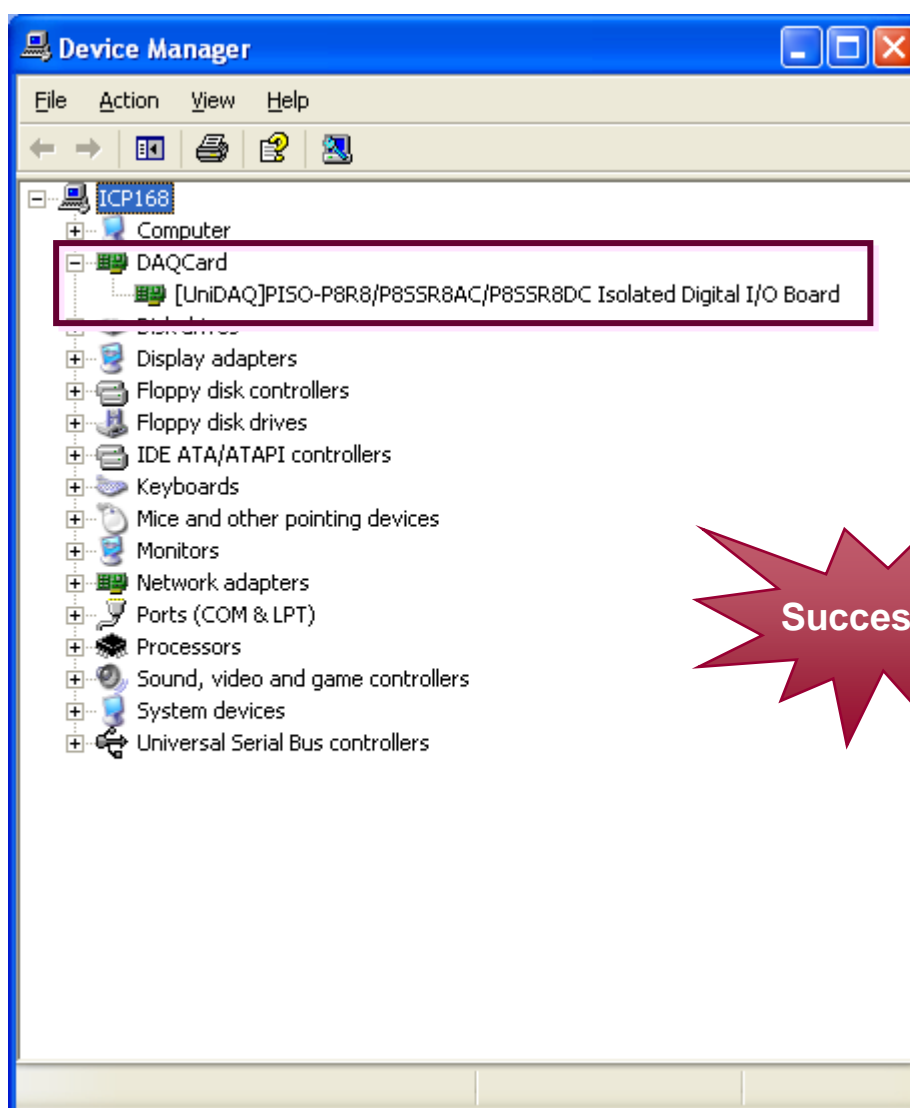
5   Demo Program

5.1   Demo Programs for Windows

Please note that none of the demo programs will work normally if the DLL driver has not been installed correctly. During the DLL driver installation process, the install shield will register the correct kernel driver to the operating system and copy the DLL driver and demo programs to the correct location depending on the driver software package you have selected (Win98/Me/NT/2000 and 32-bit Win XP/2003/Visa/7). After installing the driver, the related demo programs, development library and declaration header files for the different development environments will be available in the following folders.

The demo program is contained in:

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

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

■   BCB 4 → For Borland C^{++} Builder 4
PISODIO.H → Header files
PISODIO.LIB → Linkage library for BCB

■   Delphi4 → For Delphi 4
PISODIO.PAS → Declaration files

■   VB6 → For Visual Basic 6
PISODIO.BAS → Declaration files

■   VC6 → For Visual C^{++} 6
PISODIO.H → Header files
PISODIO.LIB → Linkage library for VC6

■   VB.NET2005 → For VB.NET2005
PISODIO.vb → Declaration files

■   CSharp2005 → For C#.NET2005
PISODIO.cs → Declaration files

A list of available demo programs is as follows:

■ DIO: D/I/O demo

5.2   Demo Programs for DOS

The related DOS software and demos are located on the CD as below:

CD:\NAPDOS\PCI\PISO-DIO\dos\P8R8\

http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/piso-dio/dos/p8r8/

After installing the software, the following drivers will be installed onto your hard disk:

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

- \TC\LIB*.*   → for TC library
- \TC\DEMO*.*   → for TC demo program
- \TC\DIAG*.*   → for TC diagnostic program

- \TC\LIB\Large*.*   → TC large model library
- \TC\LIB\Huge*.*   → TC huge model library
- \TC\LIB\Large\PIO.H   → TC declaration file
- \TC\\LIB\Large\TCPIO_L.LIB   → TC large model library file
- \TC\LIB\Huge\PIO.H   → TC declaration file
- \TC\\LIB\Huge\TCPIO_H.LIB   → TC huge model library file

- \MSC\LIB\Large\PIO.H   → MSC declaration file
- \MSC\LIB\Large\MSCPIO_L.LIB   → MSC large model library file
- \MSC\LIB\Huge\PIO.H   → MSC declaration file
- \MSC\\LIB\Huge\MSCPIO_H.LIB   → MSC huge model library file

- \BC\LIB\Large\PIO.H   → BC declaration file
- \BC\LIB\Large\BCPIO_L.LIB   → BC large model library file
- \BC\LIB\Huge\PIO.H   → BC declaration file
- \BC\\LIB\Huge\BCPIO_H.LIB   → BC huge model library file

NOTE: The library is available for all PIO/PISO series cards.

5.3 PIO_PISO

```c
/* ------------------------------------------------------------ */
/* Find all PIO_PISO series cards in this PC system          */
/* step 1 : plug all PIO_PISO cards into PC                   */
/* step 2 : run PIO_PISO.EXE                                  */
/* ------------------------------------------------------------ */

#include "PIO.H"

WORD wBase,wIrq;
WORD wBase2,wIrq2;

int main()
{
int  i,j,j1,j2,j3,j4,k,jj,dd,j11,j22,j33,j44;
WORD wBoards,wRetVal;
WORD wSubVendor,wSubDevice,wSubAux,wSlotBus,wSlotDevice;
char c;
float ok,err;

clrscr();
wRetVal=PIO_DriverInit(&wBoards,0xff,0xff,0xff);  /*for PIO-PISO*/
printf("\nThrer are %d PIO_PISO Cards in this PC",wBoards);
if (wBoards==0 ) exit(0);

printf("\n----------------------------------------------------");
for(i=0; i<wBoards; i++)
   {
   PIO_GetConfigAddressSpace(i,&wBase,&wIrq,&wSubVendor,
           &wSubDevice,&wSubAux,&wSlotBus,&wSlotDevice);

   printf("\nCard_%d:wBase=%x,wIrq=%x,subID=[%x,%x,%x],
           SlotID=[%x,%x]",i,wBase,wIrq,wSubVendor,wSubDevice,
           wSubAux,wSlotBus,wSlotDevice);
   printf(" --> ");
   ShowPioPiso(wSubVendor,wSubDevice,wSubAux);
   }

PIO_DriverClose();
}
```
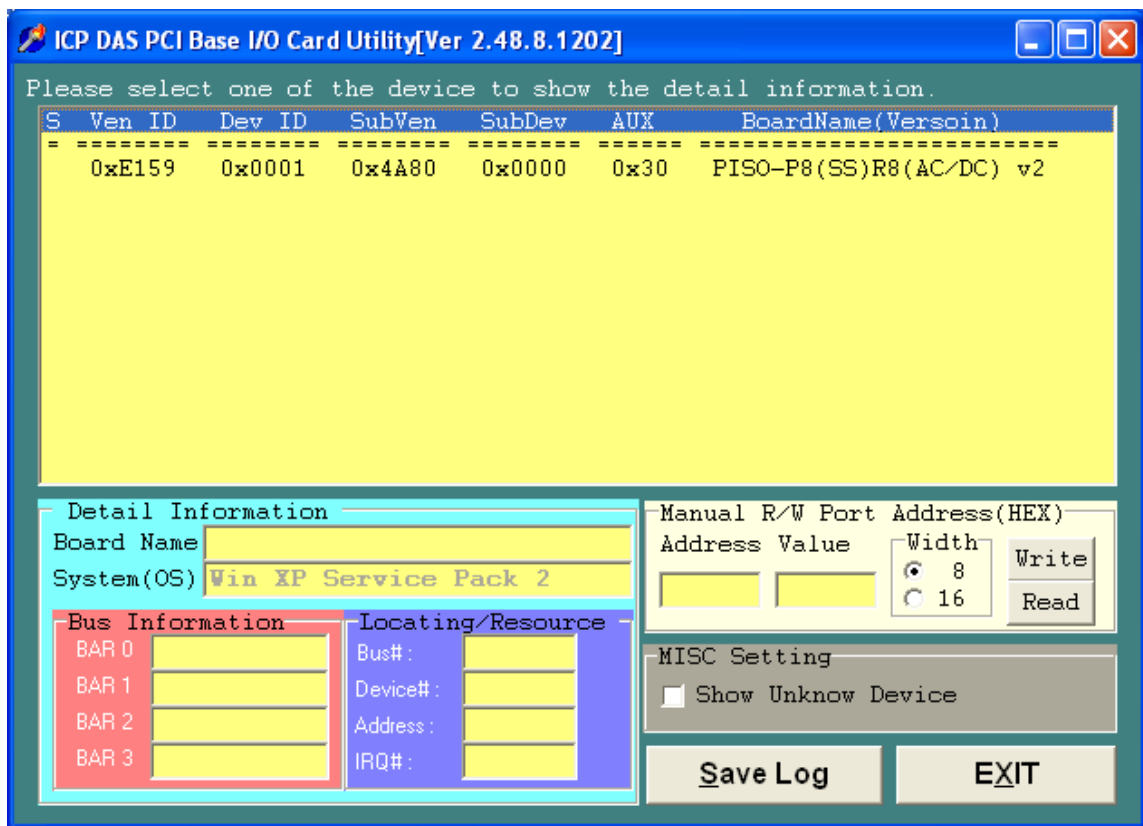
5.3.1   PIO_PISO.EXE for Windows

The PIO_PISO.exe utility is located on the CD as below and is useful for all PIO/PISO series cards.

CD:\NAPDOS\PCI\Utility\Win32\PIO_PISO\

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

After executing the utility, detailed information for all PIO/PISO cards that are installed in the PC will be shown, as illustrated below:

Note: The PIO_PISO.EXE application is valid for all PIO/PISO cards. The user can execute the PIO_PISO.EXE file to retrieve the following information:

- List all PIO/PISO cards installed in the PC
- List the resources allocated to each PIO/PISO card
- List the wSlotBus and wSlotDevice details for identification of specific PIO/PISO cards. (Refer to Section 3.2 for more information)

5.4 DEMO1 for DOS

```c
/* -------------------------------------------------------------- */
/* DEMO1  : PISO-P8R8 (PISO-P8SSR8AC/PISO-P8SSR8DC) DO demo        */
/* step 1 : Run DEMO1.EXE                                         */
/* Note   : Relay states will be show on LED                      */
/* -------------------------------------------------------------- */

#include "PIO.H"
WORD wBase,wIrq;

int main()
{
int  i;
WORD wBoards,wRetVal,t1,t2,t3,t4,t5,t6;
WORD wSubVendor,wSubDevice,wSubAux,wSlotBus,wSlotDevice;
char c;

clrscr();

/* step1 : find address-mapping of PIO/PISO cards                 */
wRetVal=PIO_DriverInit(&wBoards,0x80,0x08,0x30); /* for PISO-P8R8 */
printf("\n(1) Threr are %d PISO-P8R8 Cards in this PC",wBoards);
if ( wBoards==0 ) exit(0);

printf("\n\n------------- The Configuration Space -------------");
for(i=0;i<wBoards;i++)
    {
    PIO_GetConfigAddressSpace(i,&wBase,&wIrq,&wSubVendor,&wSubDevice,
                              &wSubAux,&wSlotBus,&wSlotDevice);

    printf("\nCard_%d: wBase=%x,wIrq=%x,subID=[%x,%x,%x],SlotID=
           [%x,%x]",i,wBase,wIrq,wSubVendor,wSubDevice,wSubAux,
           wSlotBus,wSlotDevice);

    printf(" --> ");
    ShowPioPiso(wSubVendor,wSubDevice,wSubAux);
    }

                                             /* select card_0 */
PIO_GetConfigAddressSpace(0,&wBase,&wIrq,&t1,&t2,&t3,&t4,&t5);

/* step2 : enable all D/I/O port                                  */
outportb(wBase,1);                                /* /RESET -> 1  */

i=1;
for (;;)
    {
    outportb(wBase+0xc0,i);
    i=i<<1;
    if (i>0xff) i=1;
    gotoxy(1,7);
    printf("Output=%2x",i);
    delay(20000);
    if (kbhit()!=0) break;
    }
PIO_DriverClose();
}
```

5.5 DEMO2 for DOS

```c
/* ------------------------------------------------------------- */
/* DEMO 2 : PISO-P8R8 (PISO-P8SSR8AC/PISO-P8SSR8DC) DI/O demo     */
/* step 1 : Run DEMO2.EXE                                        */
/* Note   : Relay states will be show on LED                     */
/* ------------------------------------------------------------- */

#include "PIO.H"
WORD wBase,wIrq;

int main()
{
int  i,j;
WORD wBoards,wRetVal,t1,t2,t3,t4,t5,t6;
WORD wSubVendor,wSubDevice,wSubAux,wSlotBus,wSlotDevice;
char c;

clrscr();

/* step1 : find address-mapping of PIO/PISO cards               */
wRetVal=PIO_DriverInit(&wBoards,0x80,0x08,0x30); /* for PISO-P8R8 */
printf("\n(1) Threr are %d PISO-P8R8 Cards in this PC",wBoards);
if ( wBoards==0 ) exit(0);

printf("\n\n-------------- The Configuration Space --------------");
for(i=0;i<wBoards;i++)
    {
    PIO_GetConfigAddressSpace(i,&wBase,&wIrq,&wSubVendor,
                       &wSubDevice,&wSubAux,&wSlotBus,&wSlotDevice);

    printf("\nCard_%d: wBase=%x,wIrq=%x,subID=[%x,%x,%x],SlotID=
            [%x,%x]",i,wBase,wIrq,wSubVendor,wSubDevice,
           wSubAux,wSlotBus,wSlotDevice);

    printf(" --> ");
    ShowPioPiso(wSubVendor,wSubDevice,wSubAux);
    }
                                              /* select card_0 */
PIO_GetConfigAddressSpace(0,&wBase,&wIrq,&t1,&t2,&t3,&t4,&t5);

/* step2 : enable all D/I/O port                                */
outportb(wBase,1);                            /* /RESET -> 1  */

i=1;
for (;;)
    {
    outportb(wBase+0xc0,i);
    delay(20000);
    j=(inportb(wBase+0xc0)&0xff);

    gotoxy(1,7);
    printf("Output=[%2x] => Input=[%2x]",i,j);
     i=(i<<1)&0xff;
    if (i==0)  i=1;
    if (kbhit()!=0) break;
    }
PIO_DriverClose();
}
```

6   Diagnostic Procedures

Please follow the below steps to finish the test.

Digital Input and Digital Output Test

1. Power-off the PC.
2. Install the PISO-P8R8(U)/P8SSR8(U)AC/ P8SSR8(U)DC card into an available PCI slot (5 V bus).
3. Power-on the PC with a bootable floppy disk or CD with MS-DOS.
4. Run the PISO-P8R8 DOS "Demo1.exe" program.
5. It shows how many PISO- P8R8 series board(s) found in the screen.
6. Is the number correct?
7. Does the on/off status of output indicative LEDs correspond with the DO value showed in the screen?