User's Guide

OME-PISO-813

PCI Data Acquisition Board

Hardware Manual

omega.com®
OMEGA®

OMEGAnet® Online Service www.omega.com	Internet e-mail info@omega.com

Servicing North America:

USA: ISO 9001 Certified
One Omega Drive, P.O. Box 4047
Stamford CT 06907-0047
TEL: (203) 359-1660 FAX: (203) 359-7700
e-mail: info@omega.com

Canada:
976 Bergar
Laval (Quebec) H7L 5A1, Canada
TEL: (514) 856-6928 FAX: (514) 856-6886
e-mail: info@omega.ca

For immediate technical or application assistance:

USA and Canada: Sales Service: 1-800-826-6342 / 1-800-TC-OMEGA®
Customer Service: 1-800-622-2378 / 1-800-622-BEST®
Engineering Service: 1-800-872-9436 / 1-800-USA-WHEN®
TELEX: 996404 EASYLINK: 62968934 CABLE: OMEGA

Mexico: En Español: (001) 203-359-7803 e-mail: espanol@omega.com
FAX: (001) 203-359-7807 info@omega.com.mx

Servicing Europe:

Benelux:
Postbus 8034, 1180 LA Amstelveen, The Netherlands
TEL: +31 (0)20 3472121 FAX: +31 (0)20 6434643
Toll Free in Benelux: 0800 0993344
e-mail: sales@omegaeng.nl

Czech Republic:
Frystatska 184, 733 01 Karviná, Czech Republic
TEL: +420 (0)59 6311899 FAX: +420 (0)59 6311114
Toll Free: 0800-1-66342 e-mail: info@omegashop.cz

France:
11, rue Jacques Cartier, 78280 Guyancourt, France
TEL: +33 (0)1 61 37 29 00 FAX: +33 (0)1 30 57 54 27
Toll Free in France: 0800 466 342
e-mail: sales@omega.fr

Germany/Austria:
Daimlerstrasse 26, D-75392 Deckenpfronn, Germany
TEL: +49 (0)7056 9398-0 FAX: +49 (0)7056 9398-29
Toll Free in Germany: 0800 639 7678
e-mail: info@omega.de

United Kingdom: ISO 9002 Certified
One Omega Drive, River Bend Technology Centre
Northbank, Irlam, Manchester
M44 5BD United Kingdom
TEL: +44 (0)161 777 6611 FAX: +44 (0)161 777 6622
Toll Free in United Kingdom: 0800-488-488
e-mail: sales@omega.co.uk

Tables of Contents

1. Introduction

The OME-PISO-813 is a bus-type isolated 12-bit A/D board for the PCI bus for IBM or compatible PC. It features a 10 KHz data acquisition rate under DOS and Windows. The OME-PISO-813 provides 32 single-ended analog input channels. It also provides 3000 volts of electrical isolation between the computer and card. The OME-PISO-813 has one 37-pin D-type connector. It can be installed in a 5V PCI slot and is fully "Plug & Play" compatible.

1.1 Features

- 32 single-ended analog input channels
- 12 bit A/D converter
- 3000Vdc optical isolation protection
- Analog input range

  Bipolar: ±10V, ±5V, ±2.5V, ±1.25V, ±0.625V

  Unipolar: 0 to 10V; 0 to 5V; 0 to 2.5V; 0 to 1.25V; 0 to 0.625V
- Programmable gain control: 1, 2, 4, 8, 16
- A/D trigger mode: software trigger
- A/D data transfer mode: polling
- PCI Bus
- One 37-pin D-type connector for analog inputs
- SMD, short card
- Automatic detection by Windows

1.2 Specifications

Analog Input Specifications

- No. of Channel: 32 single-ended
- Resolution: 12 bits
- Conversion rate: 10KS/s max.
- Input impedance: 10MΩ
- Overvoltage protection: ±35V
- Accuracy: 0.01% of reading ±1 bit
- Linearity: ±1bit
- On chip sample & hold
- Zero drift: ±25 PPM/ºC of FS max.

Power Requirements: 860mA/+5V max.

General Environmental

- Operating temp.: 0 to 50ºC
- Storage temp.: -20ºC to 70ºC
- Humidity: 0 to 90% non-condensing
- Dimensions: 180mm×105mm

1.3 Order Description

- OME-PISO-813: 32 channel isolated analog input board with Windows driver software

1.3.1 Options

- OME-DB-8325: Daughter board
- OME-DN-37: I/O connector block with DIN-Rail mounting and 37-pin D-type connector
- OME-DB-37: 37-pin D-type connector pin to pin screw terminal for any 37 pin D-type connector of I/O board

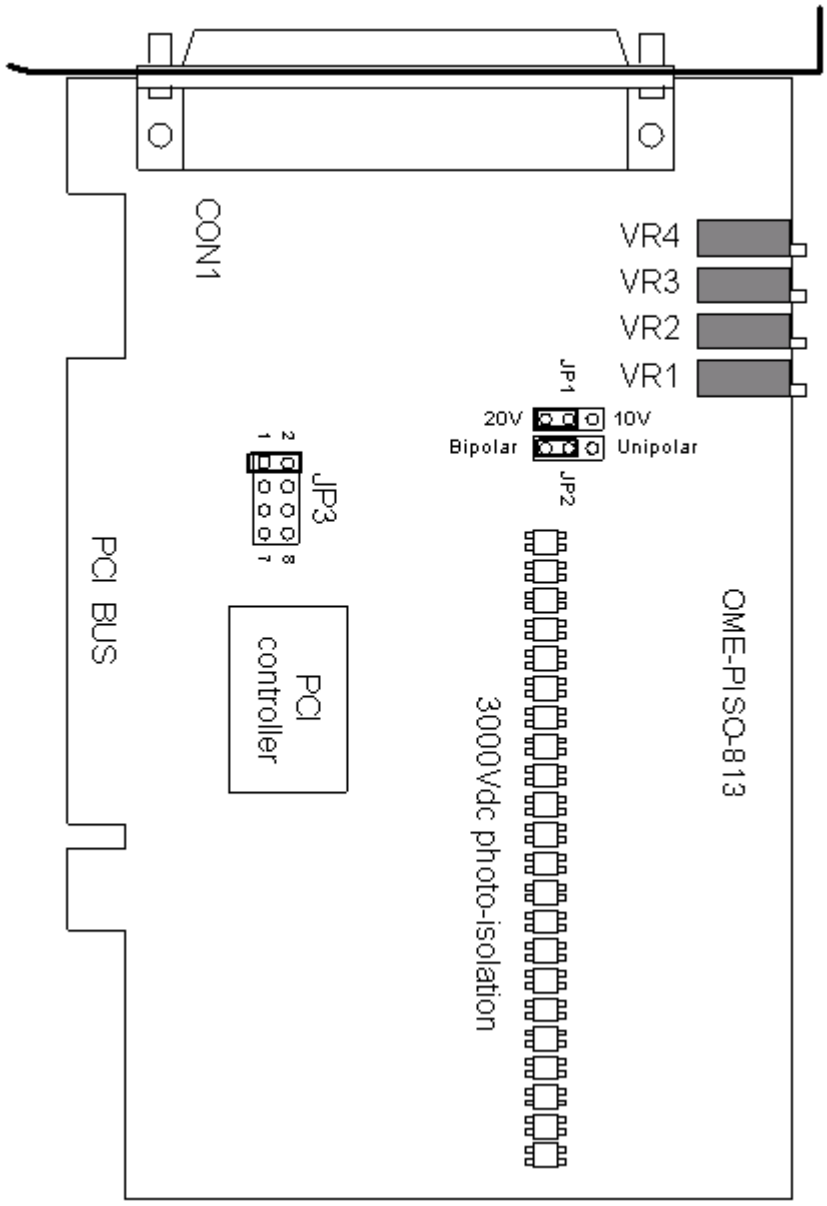
1.3.2 OME PCI Data Acquisition Family

The OME family of PCI-BUS data acquisition cards includes the following models:

- OME- PCI-1002/1202/1800/1802/1602: multi-function family, non-isolated
- OME-PCI-TMC12: timer/counter card, non-isolated
- OME-PIO-D144/D96/D64/D56/D48/D24: D/I/O family

- OME-PIO-DA16/DA8/DA4: D/A family
- OME-PISO-813: A/D card

1.4 Board Layout



CON1: 32 channels analog input

JP1: Input range setting

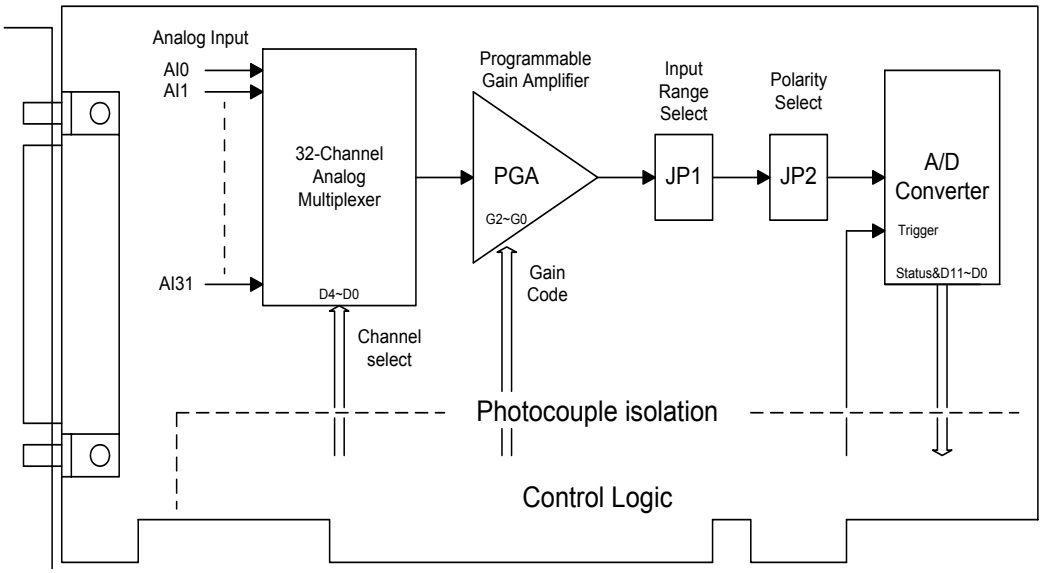
JP2: Unipolar/Bipolar setting

JP3: Reserved

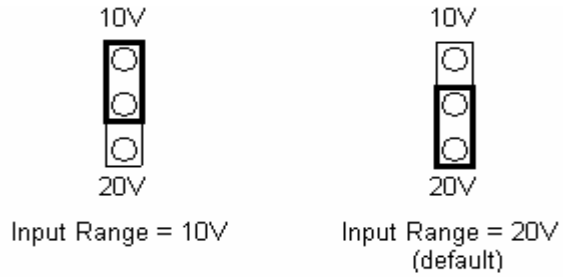
VR1 to VR4: For factory calibration

1.5 A/D Converter Operation

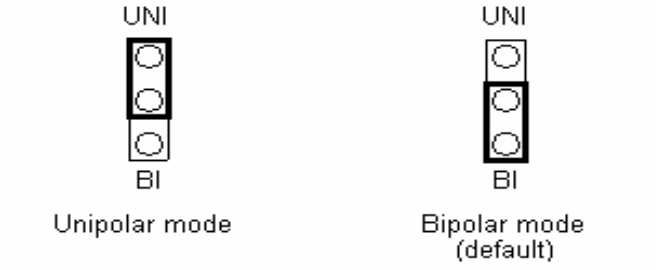
1.5.1 A/D Conversion Block Diagram



1.5.2 JP1: Analog Input Range Selection



1.5.3 JP2: Analog Input Polarity Selection



1.5.4 Jumper Setting Reference

Analog Input	JP2 Polarity	JP1 Range	Gain
-10V to +10V	Bipolar	20V	1
-5V to + 5V	Bipolar	20V	2
		10V	1
-2.5V to +2.5V	Bipolar	20V	4
		10V	2
-1.25V to +1.25V	Bipolar	20V	8
		10V	4
-0.625 to +0.625V	Bipolar	20V	16
		10V	8
0 to 10V	Unipolar	10V	1
0 to 5V	Unipolar	10V	2
0 to 2.5V	Unipolar	10V	4
0 to 1.25V	Unipolar	10V	8
0 to 0.625V	Unipolar	10V	16

NOTE: Refer to Sec.3.3.4 for further information about gain setting

1.5.5 Steps for Making an A/D Conversion

Step 1. Find address of OME-PISO-813. (Refer to Sec.3.1)

Step 2. Enable the OME-PISO-813. (Refer to Sec.3.3.1)

Step 3. Select the proper range and polarity of the analog input signal.
As show in Sec.2.2.4.

Step 4. Select input channel. (Refer to Sec.3.3.3)

Step 5. delay 10μS. (for optocoupler propagation delay and analog multiplexer settling time)

Step 6. Trigger A/D converter. (Refer to Sec.3.3.5)

Step 7. Delay 70μS. (for optocoupler propagation delay and A/D conversion time)

Step 8. Read high byte of A/D conversion data. Check the status of A/D converter until conversion ready. (Refer to Sec.3.3.2)

Step 9. Read low byte of A/D conversion data. (Refer to Sec.3.3.2)

Step10. A/D conversion complete

Refer to DEMO1.C

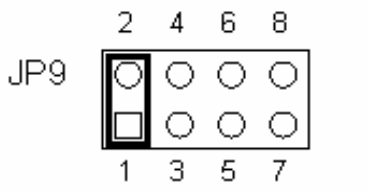
1.6 Pin Assignment

1.6.1 Analog Input Connector

CON1: 37-pin D-type female connector

Pin Number	Description	Pin Number	Description
1	AI0	20	AI1
2	AI2	21	AI3
3	AI4	22	AI5
4	AI6	23	AI7
5	AI8	24	AI9
6	AI10	25	AI11
7	AI12	26	AI13
8	AI14	27	AI15
9	AGND	28	AGND
10	AGND	29	AGND
11	AI16	30	AI17
12	AI18	31	AI19
13	AI20	32	AI21
14	AI22	33	AI23
15	AI24	34	AI25
16	AI26	35	AI27
17	AI28	36	AI29
18	AI30	37	AI31
19	AGND	×	

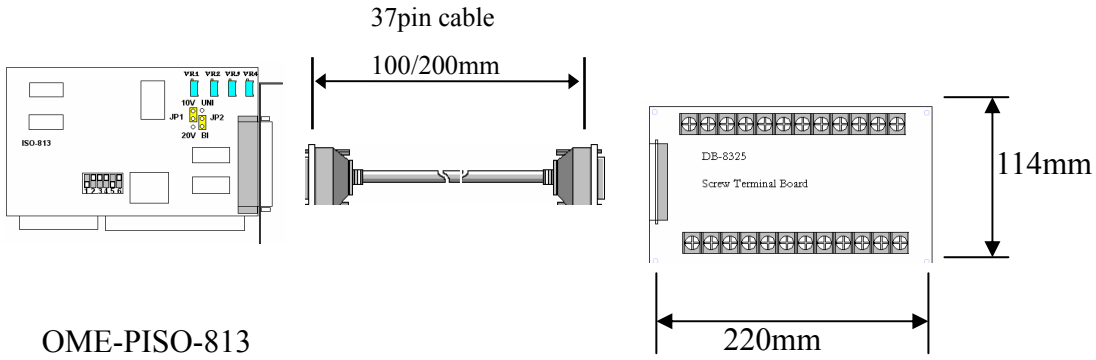
1.6.2 JP9 Reserved



Note: Reserved

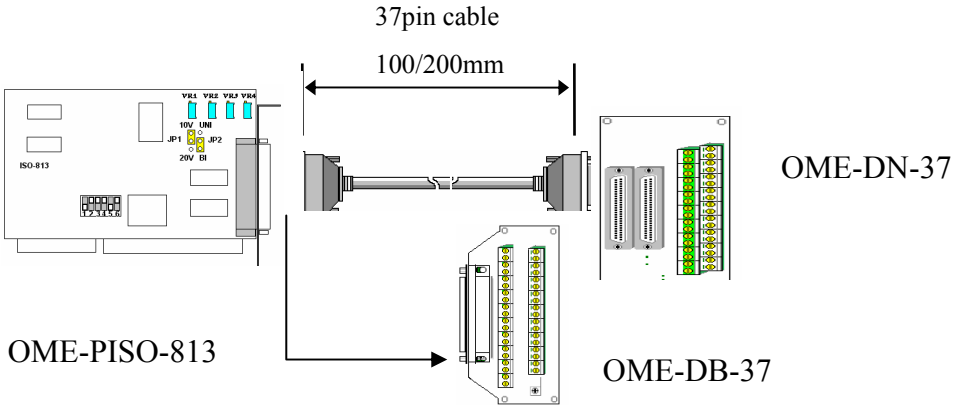
1.7 Daughter Boards

1.7.1 OME-DB-8325 Screw Terminal Board



1.7.2 OME-DB-37/ OME-DN-37 General Purpose Screw Terminal Boards

The OME-DB-37/OME-DN-37 is a general purpose terminal board. The OME-DB-37 directly connects to a 37-pin D-sub connector. The OME-DN-37 connects via 37-pin D-sub cable connection.



2. I/O Control Register

2.1 How to Find the I/O Address

During the power-up stage, the plug & play BIOS will assign a valid I/O address to every OME-PISO series card. The fixed IDs of OME-PISO series card are given as follows:

- **Vendor ID = 0xE159**
- **Device ID = 0x0002**

The sub IDs of **OME-PISO-813** are given as follows:

- **Sub-vendor ID = 0x80**
- **Sub-device ID = 0x0A**
- **Sub-aux ID = 0x00**

The following software functions are provided:

1. **PIO_DriverInit(&wBoard, wSubVendor, wSubDevice, wSubAux)**
2. **PIO_GetConfigAddressSpace(wBoardNo,*wBase,*wIrq, *wSubVendor, *wSubDevice, *wSubAux, *wSlotBus, *wSlotDevice)**
3. **Show_PIO_PISO(wSubVendor, wSubDevice, wSubAux)**

All functions are defined in PIO.H. Refer to Chapter 4 for further details. The function parameters are described below

1. Resource information:

- wBase : BASE I/O address of the card
- wIrq: IRQ channel number allocated

2. OME-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
- wSlotDevice: hardware slot ID2

The utility program, OME-PIO_PISO**.EXE**, will detect & show all OME-PISO cards installed in this PC. Refer to Sec. 4.1 for more information.

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

Sample program 1: find all OME-PISO-813 in this PC

```
wSubVendor=0x80; wSubDevice=0xa; wSubAux=0x00;/* for OME-PISO-813 */
wRetVal=PIO_DriverInit(&wBoards, wSubVendor,wSubDevice,wSubAux);
printf("There are %d OME-PISO-813 Cards in this PC\n",wBoards);
/* step2: save resource of all OME-PISO-813 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: find all OME-PIO/PISO in this PC(refer to Sec. 4.1 for more information)

```
wRetVal=PIO_DriverInit(&wBoards,0xff,0xff,0xff);  /*find all OME-PIO_PISO*/
printf("\nThrer are %d OME-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);
}
```

The sub-IDs of OME-PIO/PISO series card are given as follows:

OME-PIO/PISO series card	**Description**	**Sub_vendor**	**Sub_device**	**Sub_AUX**
OME-PIO-D144	144 * D/I/O	80	01	00
OME-PIO-D96	96 * D/I/O	80	01	10
OME-PIO-D64	64 * D/I/O	80	01	20
OME-PIO-D56	24* D/I/O + 16*D/I + 16*D/O	80	01	40
OME-PIO-D48	48*D/I/O	80	01	30
OME-PIO-D24	24*D/I/O	80	01	40
OME-PIO-DA16	16*D/A	80	04	00
OME-PIO-DA8	8*D/A	80	04	00
OME-PIO-DA4	4*D/A	80	04	00
OME-PISO-813	**32 * isolated A/D**	**80**	**0A**	**00**

Note: Additional sub-IDs may be added without notice. Refer to PIO.H for the latest information.

2.1.2 PIO_GetConfigAddressSpace

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

- wBoardNo=0 to N → total 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

This function can be used to save the resources of all OME-PIO/PISO cards installed in this system.

Sample program code is provided below:

```
/* step1: detect all OME-PISO-813 cards */
wSubVendor=0x80; wSubDevice=0xa; wSubAux=0x0; /* for OME-PISO-813 */
wRetVal=PIO_DriverInit(&wBoards, wSubVendor,wSubDevice,wSubAux);
printf("There are %d OME-PISO-813 Cards in this PC\n",wBoards);

/* step2: save the resources of all OME-PISO-813 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 OME-PISO-813 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*/
```

2.1.3 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 the subIDs. This text string is the same as that defined in PIO.H

Sample code is shown below:

```
wRetVal=PIO_DriverInit(&wBoards,0xff,0xff,0xff);  /*find all OME-PIO_PISO*/
printf("\nThrer are %d OME-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);
   }
```

2.2 The Assignment of I/O Address

The plug & play BIOS will assign a valid I/O address to OME-PIO/PISO series card. If there is only one OME-PIO/PISO board, the user can identify the board as card_0. If there are two OME-PIO/PISO boards in the system, it is more difficult to identify which board is card_0? The software driver can support up to16 boards in a computer.

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

1. Remove all OME-PISO-813 cards from the PC
2. Install one OME-PISO-813 into the PC's PCI_slot1,
   run OME-PIO_PISO.EXE & record the wSlotBus1 & wSlotDevice1
3. Remove all OME-PISO-813 from the PC
4. Install one OME-PISO-813 into the PC's PCI_slot2,
   run OME-PIO_PISO.EXE & record the wSlotBus2 & wSlotDevice2
5. repeat (3) & (4) for all PCI_slots, record all wSlotBus & wSlotDevice

The recorded information may appear as follows:

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**

These values will be mapped to this PC's physical slot. This mapping will not change for any OME-PIO/PISO cards. This can be used to identify the specific OME-PIO/PISO card. The procedure is as follows:

Step1: Record all wSlotBus? and wSlotDevice

Step2: Use PIO_GetConfigAddressSpace(…) to get the specific card's wSlotBus & wSlotDevice

Step3: The user can identify the specific OME-PIO/PISO card if he compares the wSlotBus and wSlotDevice of step2 to step1.

2.3 The I/O Address Map

The I/O address of OME-PIO/PISO series cards is automatically assigned by the motherboard ROM BIOS. The I/O address can also be re-assigned by user. **It is strongly recommended that the user not change the BIOS assigned I/O address**. This could result in system problems if an invalid address is used.

The I/O address map of OME-PISO-813 is shown below:

Address	Read	Write
WBase+0	RESET\ control register	Same
WBase+0xd0	Low byte of A/D Data	×
WBase+0xd4	High byte of A/D Data	×
WBase+0xe0	×	Multiplexer channel select register
WBase+0xe4	×	PGA gain code register
WBase+0xf0	×	A/D trigger control register

2.3.1 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\

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

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

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

2.3.2 A/D Data Register

(Read): wBase+0xD0 → Low Byte of A/D Conversion Data

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D7	D6	D5	D4	D3	D2	D1	D0

(Read): wBase+0xD4 → High Byte of A/D Conversion Data

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
×	×	×	Status	D11	D10	D9	D8

×: don't care
D11 to D0: A/D Conversion Data
Status = 0 : A/D conversion is completed
1 : A/D conversion is not completed
The status bit is used as an indicator for A/D conversion. It is used in software polling.

```
do
{
HighByte=inportb(wBase+0xd4);
}while(HighByte&0x10);        /* check status until conversion complete */
LowByte=inportb(wBase+0xd0);
Data=(HighByte<<8)+LowByte;
```

2.3.3 Multiplexer Channel Select Register

(Write): wBase+0xe0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
×	×	×	D4	D3	D2	D1	D0

```
outportb(wBase+0xe0,0);          /* Select analog input channel  0 */
outportb(wBase+0xe0,1);          /* Select analog input channel  1 */
outportb(wBase+0xe0,31);         /* Select analog input channel 31 */
```

2.3.4 PGA Gain Code Register

(Write): wBase+0xe4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
×	×	×	×	×	G2	G1	G0

```
outportb(wBase+0xe4,0);          /* Select PGA Gain = × 1   */
outportb(wBase+0xe4,1);          /* Select PGA Gain = × 2   */
outportb(wBase+0xe4,2);          /* Select PGA Gain = × 4   */
outportb(wBase+0xe4,3);          /* Select PGA Gain = × 8   */
outportb(wBase+0xe4,4);          /* Select PGA Gain = × 16 */
```

2.3.5 A/D Trigger Control Register

(Write): wBase+0xf0

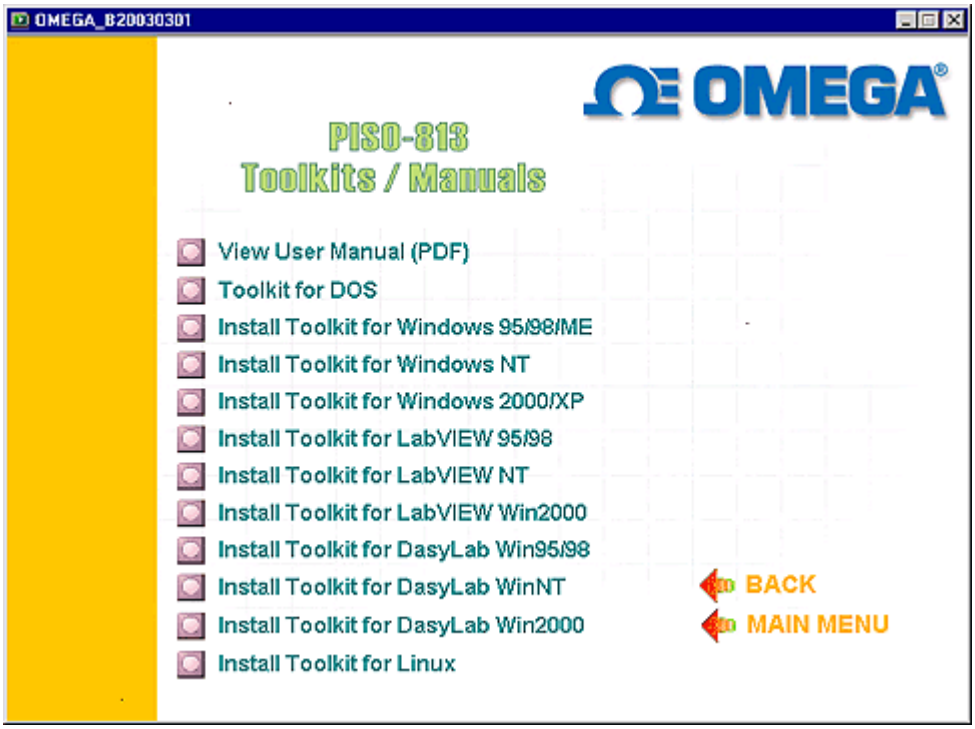
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
×	×	×	×	×	×	×	×

The OME-PISO-813 A/D data is transferred by software polling. Before reading the data, the A/D converter must be triggered by a dummy write to the A/D Trigger Control Register.(Refer to Sec.2.2.5 for more information about A/D converter operation)

```
outportb(wBase+0xf0,0);          /* Trigger A/D converter */
```

3. Demo Programs

The OME-PISO-813 includes a comprehensive set of drivers and demonstration programs for programming under various operating systems and programming platforms. During the software installation process, the following screen allows the user to install the appropriate files for a specific installation.



Please read the release notes for the most up to date information regarding the OME-PISO-813 hardware and software.

The first step for most OME-PISO-813 programs is to identify the cards in the computer system and determine the resources allocated to those cards. The PIO_PISO.EXE program performs that function. The source code for the PIO_PISO.EXE program is shown below.

3.1 PIO_PISO.EXE Program

```
/* ------------------------------------------------------------ */
/* Find all OME-PIO_PISO series cards in this PC system         */
/* step 1: plug all OME-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("\nThere are %d OME-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();
}
```

NOTE**: PIO_PISO.EXE is valid for all OME-PIO/PISO cards**. It can be found in the DIAG directory. Running PIO_PISO.EXE will provide the following information:

- List all OME-PIO/PISO cards installed in this PC
- List all resources allocated to every OME-PIO/PISO cards
- List the wSlotBus & wSlotDevice for specified OME-PIO/PISO card identification. (Refer to Sec. 3.2 for more information)

Although most users will use the functions in the software driver to program the PISO-813, DEMO1.C demonstrates how to measure 32 analog inputs by programming directly to the registers on the board.

3.2 DEMO1

```
/* ------------------------------------------------------------ */
/* DEMO1.C : OME-PISO-813                                       */
/* Note    : Measure 32-channel A/I.Bipolar range: -10V - +10V  */
/* ------------------------------------------------------------ */
#include "PIO.H"
WORD Read_AD_Data(void);
WORD wBase,wIrq;

int main()
{
int  i,l,h,x,y;
WORD wBoards,wRetVal,AdResult,t1,t2,t3,t4,t5;
WORD wSubVendor,wSubDevice,wSubAux,wSlotBus,wSlotDevice;
char c;
float ok,err,v,k;

clrscr();
/* step 1: find address-mapping of OME-PIO/PISO cards
*/
wRetVal=PIO_DriverInit(&wBoards,0x80,0x0a,0x00);  /* for OME-PISO-813
*/
printf("\nThrer are %d OME-PISO-813 Cards in this PC",wBoards);
if (wBoards==0) exit(0);

printf("\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);
   }
PIO_GetConfigAddressSpace(0,&wBase,&wIrq,&t1,&t2,&t3,&t4,&t5);
/* step 2: enable all D/I/O port                                */
outportb(wBase+0,1);                          /* enable D/I/O  */

i=0;x=1;y=1;

clrscr();

/* Step 3: gain setting                                         */
/* Delay more than 5.6us for PGA gain change and optocouple     */
/* propagation delay 6.0us. (5.6+6.0)us                         */
outportb(wBase+0xe4,0x00);                  /* Gain control,G=1  */
delay(1000);

for(;;)
   {
    gotoxy(x,y);
    printf("Channel %2d ",i);
```

```
/* step 4: channel select                                          */
    outportb(wBase+0xe0,i);                  /* channel select = i*/
/* step 5: delay 10us                                              */
    delay(200);


/* step 6: software trigger                                        */
    outportb(wBase+0xf0,0x00);               /* software trigger  */
/* step 7: delay 70us                                              */
    delay(200);

    AdResult=Read_AD_Data();

    k=((float)AdResult-2047.0)*10.0/2048.0;

    printf(",value = %2.4f",k);

    i++;
    if (i==0x20) i=0;

    y=i+1;
    x=1;

    if (i>=16)
    {
      x=40;
      y=i-15;
    }

    if (kbhit()!=0)
       {
        c=getch(); if ((c=='q') || (c=='Q')) break;
       }
   }
gotoxy(1,20);
PIO_DriverClose();
}

/* ---------------------------------------------------------------- */
WORD Read_AD_Data(void)
{
int  LowByte;
WORD HighByte,Data;
char c;
/* step 8: read high byte of A/D result                            */
do
   {
   HighByte=inportb(wBase+0xd4);    /* Read high byte of A/D data */
   if (kbhit()!=0) break;
   }
   while(HighByte&0x10);/* Chech status until convertion complete */
/* step 9: read low byte of A/D result                             */
LowByte=inportb(wBase+0xd0);
Data=((HighByte<<8)+LowByte)&0xfff;
return(Data);
/* step 10: A/D conversion complete                                */
}
```

WARRANTY/DISCLAIMER

OMEGA ENGINEERING, INC. warrants this unit to be free of defects in materials and workmanship for a period of 13 months from date of purchase. OMEGA's WARRANTY adds an additional one (1) month grace period to the normal one (1) year product warranty to cover handling and shipping time. This ensures that OMEGA's customers receive maximum coverage on each product.

If the unit malfunctions, it must be returned to the factory for evaluation. OMEGA's Customer Service Department will issue an Authorized Return (AR) number immediately upon phone or written request. Upon examination by OMEGA, if the unit is found to be defective, it will be repaired or replaced at no charge. OMEGA's WARRANTY does not apply to defects resulting from any action of the purchaser, including but not limited to mishandling, improper interfacing, operation outside of design limits, improper repair, or unauthorized modification. This WARRANTY is VOID if the unit shows evidence of having been tampered with or shows evidence of having been damaged as a result of excessive corrosion; or current, heat, moisture or vibration; improper specification; misapplication; misuse or other operating conditions outside of OMEGA's control. Components which wear are not warranted, including but not limited to contact points, fuses, and triacs.

OMEGA is pleased to offer suggestions on the use of its various products. However, OMEGA neither assumes responsibility for any omissions or errors nor assumes liability for any damages that result from the use of its products in accordance with information provided by OMEGA, either verbal or written. OMEGA warrants only that the parts manufactured by it will be as specified and free of defects. OMEGA MAKES NO OTHER WARRANTIES OR REPRESENTATIONS OF ANY KIND WHATSOEVER, EXPRESS OR IMPLIED, EXCEPT THAT OF TITLE, AND ALL IMPLIED WARRANTIES INCLUDING ANY WARRANTY OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE HEREBY DISCLAIMED. LIMITATION OF LIABILITY: The remedies of purchaser set forth herein are exclusive, and the total liability of OMEGA with respect to this order, whether based on contract, warranty, negligence, indemnification, strict liability or otherwise, shall not exceed the purchase price of the component upon which liability is based. In no event shall OMEGA be liable for consequential, incidental or special damages.

CONDITIONS: Equipment sold by OMEGA is not intended to be used, nor shall it be used: (1) as a "Basic Component" under 10 CFR 21 (NRC), used in or with any nuclear installation or activity; or (2) in medical applications or used on humans. Should any Product(s) be used in or with any nuclear installation or activity, medical application, used on humans, or misused in any way, OMEGA assumes no responsibility as set forth in our basic WARRANTY/DISCLAIMER language, and, additionally, purchaser will indemnify OMEGA and hold OMEGA harmless from any liability or damage whatsoever arising out of the use of the Product(s) in such a manner.

RETURN REQUESTS/INQUIRIES

Direct all warranty and repair requests/inquiries to the OMEGA Customer Service Department. BEFORE RETURNING ANY PRODUCT(S) TO OMEGA, PURCHASER MUST OBTAIN AN AUTHORIZED RETURN (AR) NUMBER FROM OMEGA'S CUSTOMER SERVICE DEPARTMENT (IN ORDER TO AVOID PROCESSING DELAYS). The assigned AR number should then be marked on the outside of the return package and on any correspondence.

The purchaser is responsible for shipping charges, freight, insurance and proper packaging to prevent breakage in transit.

FOR WARRANTY RETURNS, please have the following information available BEFORE contacting OMEGA:

1. Purchase Order number under which the product was PURCHASED,
2. Model and serial number of the product under warranty, and
3. Repair instructions and/or specific problems relative to the product.

FOR NON-WARRANTY REPAIRS, consult OMEGA for current repair charges. Have the following information available BEFORE contacting OMEGA:

1. Purchase Order number to cover the COST of the repair,
2. Model and serial number of the product, and
3. Repair instructions and/or specific problems relative to the product.

OMEGA's policy is to make running changes, not model changes, whenever an improvement is possible. This affords our customers the latest in technology and engineering.

OMEGA is a registered trademark of OMEGA ENGINEERING, INC.

Where Do I Find Everything I Need for Process Measurement and Control? OMEGA...Of Course!

Shop online at www.omega.com

TEMPERATURE

- Thermocouple, RTD & Thermistor Probes, Connectors, Panels & Assemblies
- Wire: Thermocouple, RTD & Thermistor
- Calibrators & Ice Point References
- Recorders, Controllers & Process Monitors
- Infrared Pyrometers

PRESSURE, STRAIN AND FORCE

- Transducers & Strain Gages
- Load Cells & Pressure Gages
- Displacement Transducers
- Instrumentation & Accessories

FLOW/LEVEL

- Rotameters, Gas Mass Flowmeters & Flow Computers
- Air Velocity Indicators
- Turbine/Paddlewheel Systems
- Totalizers & Batch Controllers

pH/CONDUCTIVITY

- pH Electrodes, Testers & Accessories
- Benchtop/Laboratory Meters
- Controllers, Calibrators, Simulators & Pumps
- Industrial pH & Conductivity Equipment

DATA ACQUISITION

- Data Acquisition & Engineering Software
- Communications-Based Acquisition Systems
- Plug-in Cards for Apple, IBM & Compatibles
- Datalogging Systems
- Recorders, Printers & Plotters

HEATERS

- Heating Cable
- Cartridge & Strip Heaters
- Immersion & Band Heaters
- Flexible Heaters
- Laboratory Heaters

ENVIRONMENTAL MONITORING AND CONTROL

- Metering & Control Instrumentation
- Refractometers
- Pumps & Tubing
- Air, Soil & Water Monitors
- Industrial Water & Wastewater Treatment
- pH, Conductivity & Dissolved Oxygen Instruments

M3933/0203