PISO-DA2

User's Manual

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, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

The names used for identification only may be registered trademarks of their respective companies.

Contents

1.General information

1.1 Introduction

The PISO-DA2 is a PCI bus analog output board with two output channels and isolated 12-bit revolution for IBM Personal computer and compatible. The output range can be configured as voltage output by one of the different ranges: ±10V, ±5V, 0~10V, 0~5V or current output using loop sink by the range: 0~20mA or 4~20mA. Besides, output channel is isolated each other and its isolation range is increased the voltage more than 3000Vdc. It can be thought as the most cost-effective isolated D/A board for the PCI bus interface.

The output terminals of the PISO-DA2 are two sets of 9-pin D-type female connectors. PISO-DA2 can be installed in a 5V PCI slot and truly supports “Plug & Play” function of PCI-bus. The key features are described as below:

1.2 Features

- PCI BUS
- Two independent 12 bits analog output channels
- 3750VDC isolation protection
- Analog output range
  - Voltage output

    Bipolar: ±10V, ±5V

    Unipolar: 0~10V, 0~5V
  - Current output

    Current loop sink:0~20mA, 4~20mA
- 3000Vdc isolation DC/DC converter build-in
- Software calibration
- Unipolar or bipolar output available from each converter
- Two pacer timer interrupt source
- The voltage or current output can be set to arbitrary values or reset to zero when the power is on.
- Double buffered D/A latches.
- The calibration data is fully stored in EEPROM.

1.3 Specifications

Analog Outputs

- D/A converter with resolution: 12 bits
- Conversion rate: 10KS/s max.
- Output Range:(selection by jumper)
  - ■ Voltage output

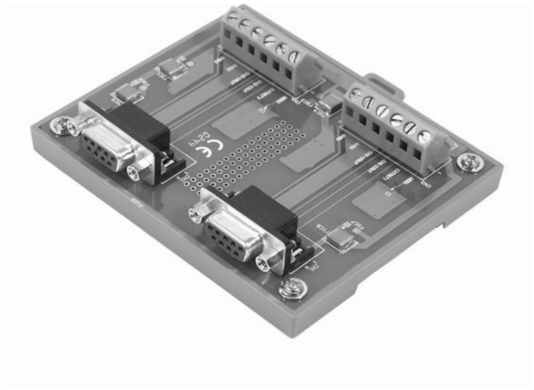
    Bipolar: ±10V, ±5V

    Unipolar: 0~10V, 0~5V
  - ■ Current output

    Current loop sink:0~20mA, 4~20mA
- Reference Voltage:
  - ■ Internal:-5V and -10V
  - ■ External: -5V and -10V
- Current Loop Excitation Voltage: +8V~+36V
- Isolation Voltage: 3000Vrms
- Converter: AD7541 or equivalent
- Trigger Mode: Software trigger
- Data Transfer: software control
- Accuracy: 0.015% of reading FSR (Full Scale Range)
- Linearity: ±1/2 LSB

Optional daughter board

- The DB-8425 Screw Terminal Board with 1.5 meter D-Sub 9-pin cables is provided for easy wire connection with the controlled device or equipment. The Daughter Board is not the standard component included in PISO-DA2 package.



1.4 General Specifications

- Bus Type: PCI Bus
- Connector: Two 9-pin D-type female connectors
- Operating temp: 0~50ºC
- Storage temp: -20ºC ~ 70ºC
- Humidity: 0~90% non-condensing
- Dimensions: 170mm×122mm

1.5 Applications

This PISO-DA2 board with two isolated analog output channel may be applied to the following areas:

- Arbitrary waveform generation
- Control of Valves, switches, relays
- Programmable voltage sources
- Servo Control
- Programmable current sink

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

This section will describe the hardware setting of the PISO-DA2. At first, the contents in the package and unpacking information that you should have are described. The jumper settings for the PISO-DA2 according to reference voltage sources, output voltage range and voltage or current output are also presented in the second stage.

2.1 What You Have

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

- One piece of PISO-DA2 2-channel Isolated Analog Output Card
- One piece of ICPDAS floppy diskette or CD
- One piece of release note

It is recommended to read the release note firstly. All importance information will be given in the release note as follows:

- Where you can find the software driver, utility and demo programs.
- How to install software & utility.
- Where is the diagnostic program?
- FAQ and answer

Attention!

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 PISO-DA2's Layout

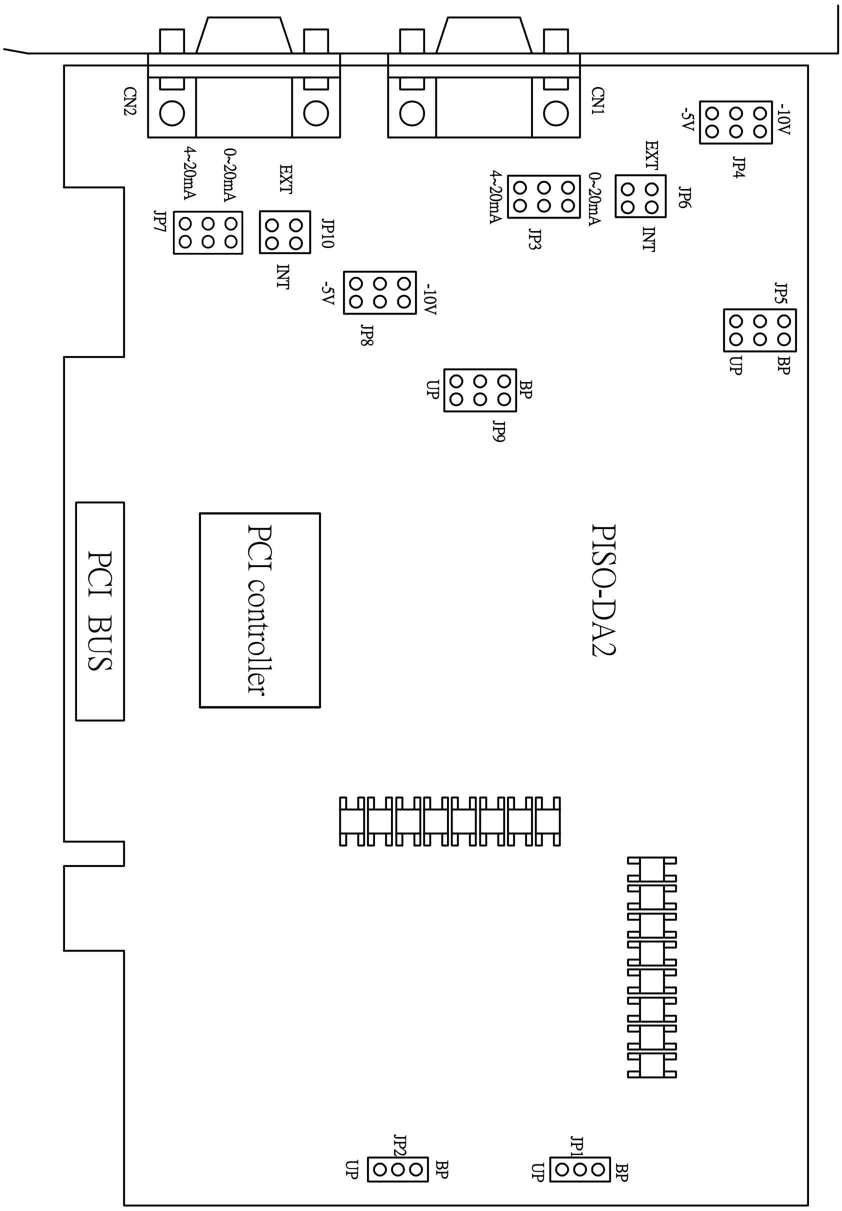


Figure2.1 PISO-DA2 LAYOUT

Note:

CN1: The terminal of D/A converter channel-1 for voltage or current output

CN2: The terminal of D/A converter channel-2 for voltage or current output

JP1, JP2, JP5 and JP9: Bipolar or Unipolar setting

JP4 and JP8: -5V or –10V internal reference voltage setting

JP3 and JP7: 0~20mA or 4~20mA current loop setting

JP6 and JP10: External or internal reference voltage setting

2.3 Configuration of D/A output signal

The each D/A channel of PISO-DA2 can be independently configured as voltage or current output using different range to fit your application, as shown in Table 4-1 and Table 4-2. Generally, the range of voltage output, which includes bipoloar and unipoloar, is configured by jumper JP1, JP2, JP5, and JP9. In addition, the reference voltage and source are arranged by Jumper JP6 and JP10. JP4 and JP8 define the internal reference voltage source as -5V or -10V for channel 1 and 2 respectively.

For the current output, the user needs to define the reference voltage source as internal -5V by jumper JP4 and JP8 and use jumper JP5 and JP9 to define signal as unipolar polarity. And then the range of the current output can be configured by jumper JP3 and JP7 for channel 1 and 2 respectively. **Note that each output channel can be set up as voltage or current output independently by using the corresponding setting jumper.** The more detail configuration method for the analog output will be demonstrated in the following section.

Table 2.1: Jumper setting

Output voltage	JP1, JP2, JP5, JP9	JP6, JP10, JP4, JP8
	Polarity selection	Ext/Int selection
-10V ~ +10V	Bipolar	Internal (-5 or -10V)
-5V ~ +5V	Bipolar	Internal (-5 or -10V)
0V ~ 10V	Unipolar	Internal (-5 or -10V)
0V ~ 5V	Unipolar	Internal (-5 or -10V)
(Ext_ref voltage)~ -(Ext_ref. voltage)	Bipolar	External
0~ - (Ext_ref. voltage)	Unipolar	External

Table 2.2: Jumper setting for current sink

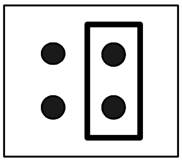
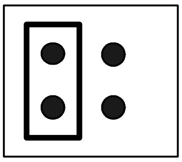
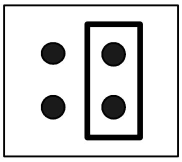
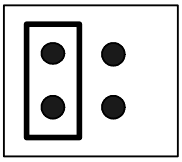
Current sink	JP3, JP7	JP5, JP10	JP6, JP10 JP4, JP8
		Polarity selection	Ext/Int selection
0~20 mA	Output range setting	Unipolar	Internal (-5 or -10V)
4~20mA	Output range setting	Unipolar	Internal (-5 or -10V)

2.3.1 The configuration of Voltage output

Reference Source Setting

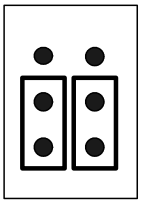
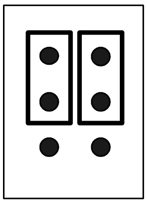
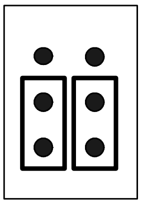
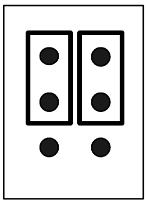
The reference voltage source of PISO-DA2's D/A converter can be provided by the internal generator or external reference voltage, which is coming from the connector CN1 and CN2. The setting of the reference sources for Channel 1 and Channel 2 are controlled by the jumper JP6 and JP10 respectively, as depicted in the following table.

Table 2.3: Setting reference source for channel 1 and 2.

Channel	Internal Reference Voltage (Default)	External Reference Voltage
Channel 1	JP6  EXT INT	JP6  EXT INT
Channel 2	JP10  EXT INT	JP10  EXT INT

If the internal reference voltage source is selected to implement the analog output of the D/A board, then the reference voltage level must be set as the internal voltage source -5V or -10V, which is precision voltage source provided by PISO-DA2, by jumper JP4 (Channel 1) and JP8 (Channel 2). The detail setting method is illustrated as Table 2.3. The default setting is -5V.

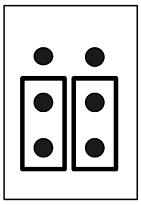
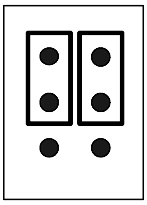
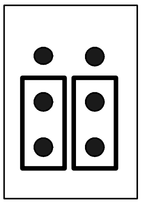
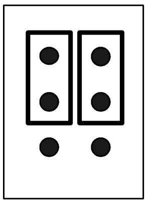
Table 2.4: Internal Reference Voltage Setting

Channel	-5V (Default)	-10V
Channel 1	JP4  -10V -5V	JP4  -10V -5V
Channel 2	JP8  -10V -5V	JP8  -10V -5V

Output Range Setting

After the configuration of reference voltage source, the practical voltage output range of PISO-DA2 can be regulated as either Bipolar or Unipolar by jumper JP5 and JP9. The detail setting is demonstrated in Table 2.4. For example, if the jumper JP4 and JP5 are set as -5V and Unipolar respectively, then the range of voltage output is 0~5V for channel 1.

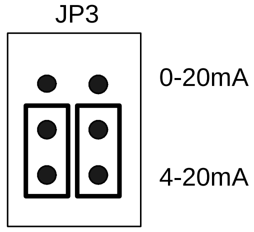
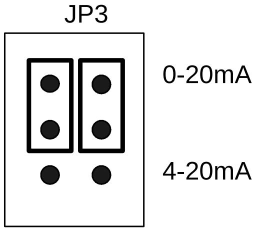
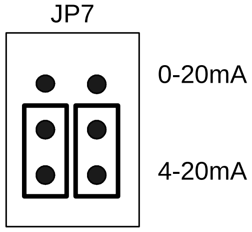
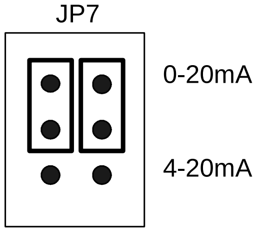
Table 2.5: Output range setting

Channel	Unipolar (Default)	Bipolar
Channel 1	JP5  BP UP	JP5  BP UP
Channel 2	JP9  BP UP	JP9  BP UP

2.3.2 The configuration of current sink

If users want to apply PISO-DA2 to function as current output board, it provides the probability setting of current output mode using 0-20mA or 4-20mA loop current sink. In order to activate the current output, the PISO-DA2 board must be configured as the output voltage by the setting of **unipolar** and **internal reference voltage -5V** for each output channel, in advance. Then Jumper JP3 and JP7 of PSIO-DA2 board can be utilized to select the current output range for channel 1 and 2 respectively. The detail setting is depicted as below table.

Table 2.6: Setting the output range of current sink.

Channel	4-20mA (Default)	0-20mA
Channel 1		
Channel 2		

2.4 Connector Pin Assignment

The PISO-DA2 is equipped with two sets of 9-pin D-type female connectors for wire connection of the output signal. CN1 and CN2 stand as are signal output connections of Channel 1 and CN2 respectively. The connector’s pin assignment is specified as follows:

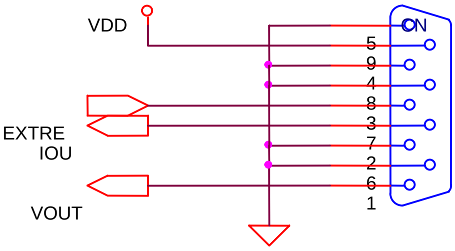


Figure 2.1: The legend of CN1 pin assignments

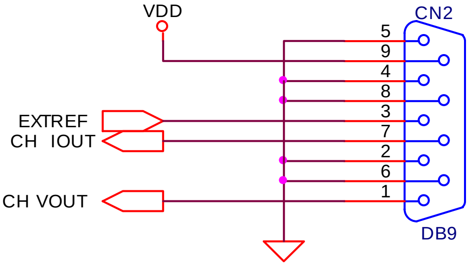


Figure 2.2: The legend of CN2 pin assignments

Table 2.7: Pin assignment of CH1 and CH2

CN1 pin assignment		CN2 pin assignment	
1	Voltage output	1	Voltage output
2	Signal GND	2	Signal GND
3	EXTREF: Ref. Voltage Input	3	EXTREF: Ref. Voltage Input
4	Signal GND	4	Signal GND
5	Signal GND	5	Signal GND
6	Signal GND	6	Signal GND
7	I_{OUT}: Current output	7	I_{OUT}: Current output
8	Signal GND	8	Signal GND
9	VDD:+15V output	9	VDD:+15V output

2.5 Signal Connection

This section will demonstrate the correct signal connection skill because it plays an important role for sending data accurately. According to the different applications, the wire connection may be different as shown in the following subsection. **Generally, when PISO-DA2 board is functioned as the voltage output application, the minimum load resistor must be bigger than 1kΩ. However, when current output function of PISO-DA2 board is selected, then the maximum load resistor of the current loop sink has to be smaller than 400Ω.-**

2.5.1 Voltage Output Connection

Figure 2.3 presents the wire connection for voltage output from PISO-DA2. It works fine when the external load resistor is bigger than **1kΩ.**

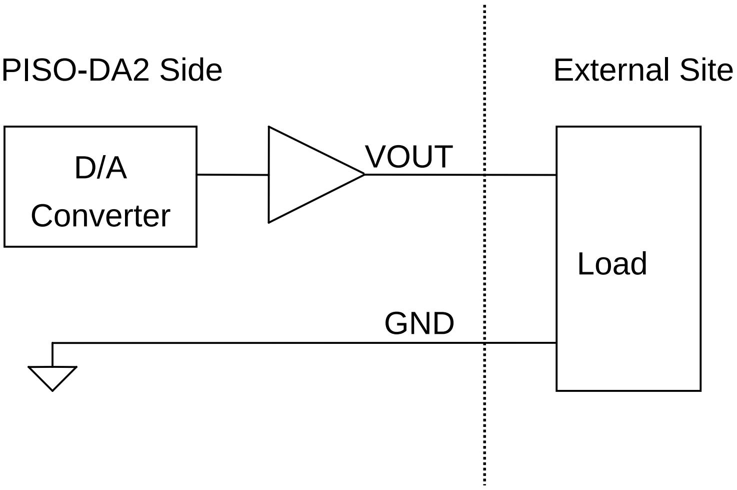


Figure 2.3: Voltage Output Connection

2.5.2 Current Sink Connection

Figure 2.4 depicts the wire connection for current output from PISO-DA2 when external power supply is used. It works fine when the external load resistor is smaller than **400Ω. Note that the external power supply used in this wire connection must be a DC source between +8V and +36VDC.**

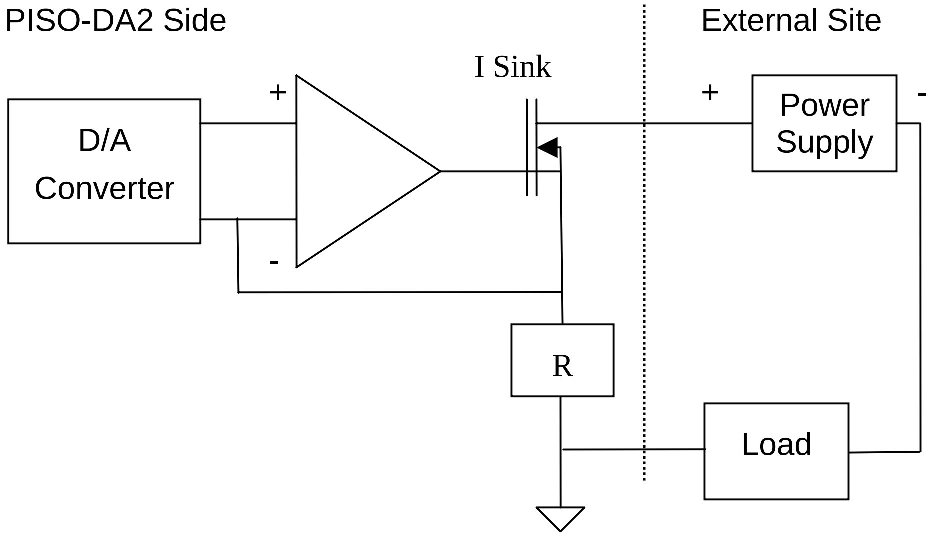


Figure 2.4: Current output Connection when external power is used

2.5.3 Current sink with internal power supply

If your system does not offer external power supply, a built-in +15V DC power source provided by PISO-DA2 can be applied to build the signal current output in loop current sink type. The wire connection is illustrated as below.

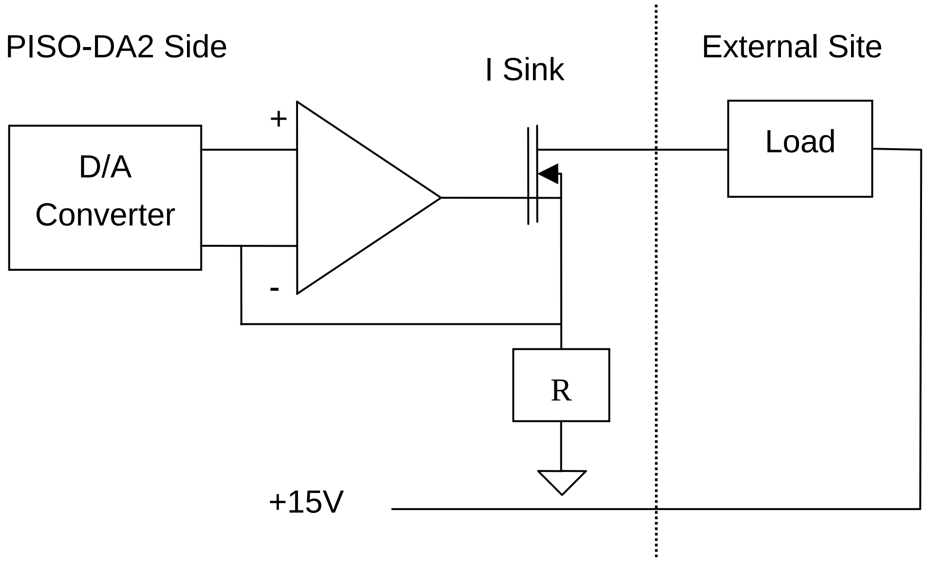


Figure 2.5: current output connection when internal power supply is used

2.6 Output Range and Resolution

The voltage and current output range of PISO-DA2 is represented as follow. The D/A converter is 12 bits. Based on the different output range, the resolution of every channel is figured out as Table 2.8.

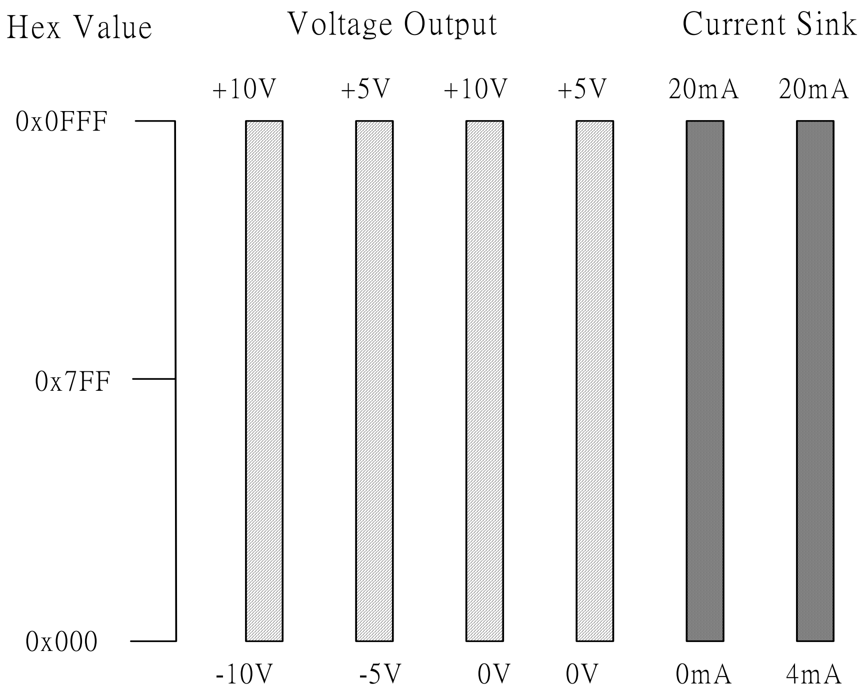


Figure 2.6: Output range and the corresponding revolution

Table 2.8: The resolution of each range

Configuration	Equivalent Bit	Resolution
-10V ~ +10V	12 bits	4.884mV
-5V ~ +5V	12 bits	2.442mV
0 ~ 10V	12 bits	2.442mV
0 ~ 5V	12 bits	1.221mV
0mA ~ 20mA	12 bits	4.884uA
4mA~20mA	12 bits	3.907uA

2.7 Calibration

The PISO-DA2 is shipped fully calibrated from the factory with calibration coefficients stored in the EEPROM on board. For more precise application of voltages or currents at the "system end", the following procedure provides a method that allows you to calibrate the board within your system, for correct voltages or currents at your field connection. This calibration allows the user to remove the effects of voltage drops caused by IR loss in the cable and connector.

At first, the user has to prepare two equipments for calibration: (1) precise multi-meter and power supply (External reference) for current calibration. Note that the calibrated values for analog output channel are stored to the address in the first 16 words of the EEPROM, as shown in Table 2.9.

Table 2.9 Calibration values stored to the EEPROM address

Output range	The address of the EEPROM for CH1	The address of the EEPROM for CH2
0~10V	0	8
-10~10V	1	9
0~5V	2	10
-10~10V	3	11
0mA~20mA	4 (minimum)	12(minimum)
	5(Maximum)	13(Maximum)
4mA~20mA	6(minimum)	14(minimum)
	7(Maximum)	15(Maximum)

After understanding the calibration mapping address and having precise multi-meter, the calibration procedure for each analog output channel is as the example for calibrating analog channel in 0~10V setting and the calibration interface is shown in Figure 2.7.

Step 1: Please follow the jumper setting according to your analog output configuration.

Step 2: Run calibration tool, which is located in “/program files/ DAQpro/PISO-DA2/calibation.exe”, to open configuration interface, as shown in Figure 2.7.

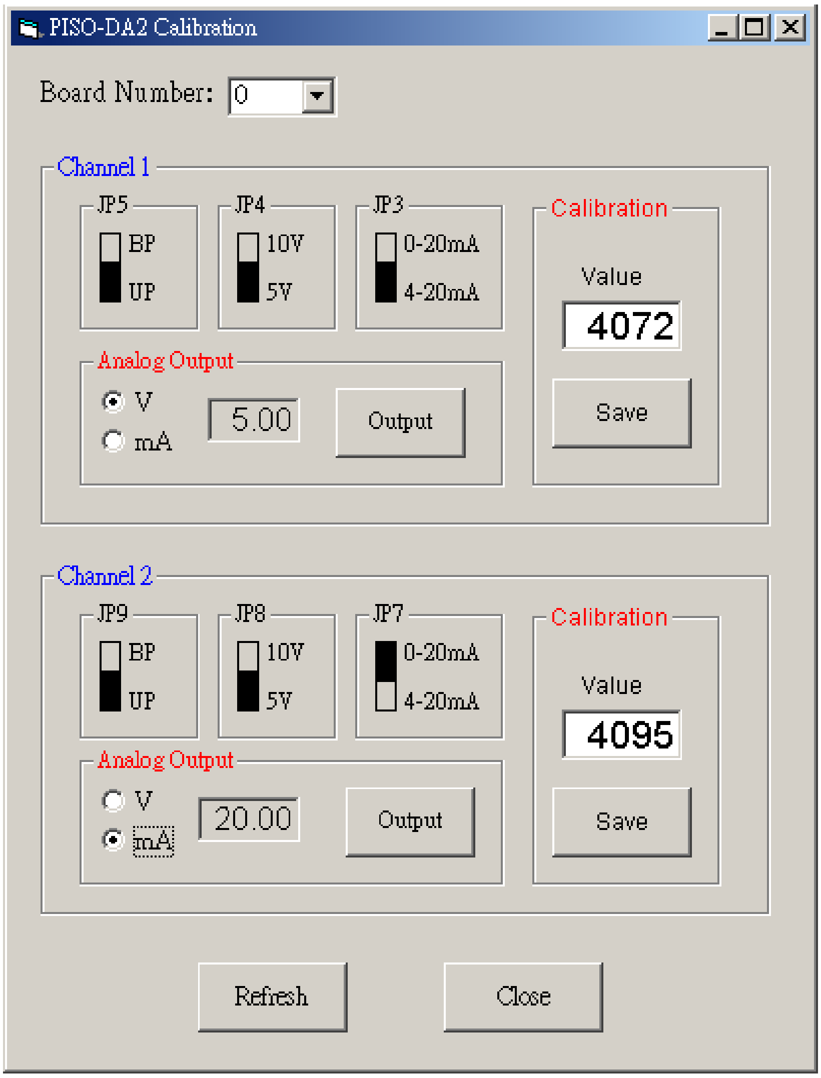
Step 3: If we want to calibrate channel 1, for example, then let the calibration value set as 4095 and click “save” to save this value to the corresponding EEPROM.

Step 4: Click output button and then use precise multi-meter to measure the analog output.

Step 5: If the analog output is smaller or bigger than the allowance maximum value of analog output channel 1, then go to step 3~4 to change the saved value in EEPRON and output voltage value until the value is equal to the allowance maximum voltage output value.

Step 6: If the analog output is equal to the maximum allowance analog output, it means that the calibration process is finished for the setting range of analog output for channel 1. If user want to calibrate the other analog output range, please go to step 1~5 to proceed the individual calibration process.

Note that the maximum and minimum calibration process of the current output must be proceeded if the user wants to calibrate the current output. And the procedure is the same the above.



PISO-DA2 Calibration

Board Number: 0

Channel 1
- JP5: BP / UP
- JP4: 10V / 5V
- JP3: 0-20mA / 4-20mA
- Analog Output: ● V ○ mA — 5.00
- Calibration Value: 4072

Channel 2
- JP9: BP / UP
- JP8: 10V / 5V
- JP7: 0-20mA / 4-20mA
- Analog Output: ○ V ● mA — 20.00
- Calibration Value: 4095

Figure 2.7: Calibration tool interface.

3.Software Installation

The PISO-DA2 can be used in DOS and Windows 98/Me/NT/2000/XP. For these Windows O.S, the recommended installation steps are given in Sec 3.1 ~ 3.2

3.1 Software Installing Procedure

Step 1: Insert the companion CD into the CD-ROM driver and wait a few seconds until the installation program starts automatically. If it cannot be started automatically for some reasons, please double-click the file **\NAPDOS\AUTO32.EXE** in this CD.

Step 2: Click the first item; Toolkits (Software) / Manuals.

Step 3: Click the item PCI Bus DAQ Card.

Step 4: Click PISO-DA2.

Step 5: Click “install Toolkit for Windows 98 (Or Me, NT, 2000, XP)”.

Then, the InstallShield will start the driver installation process to copy the related material to the indicated directory and register the driver in your computer.

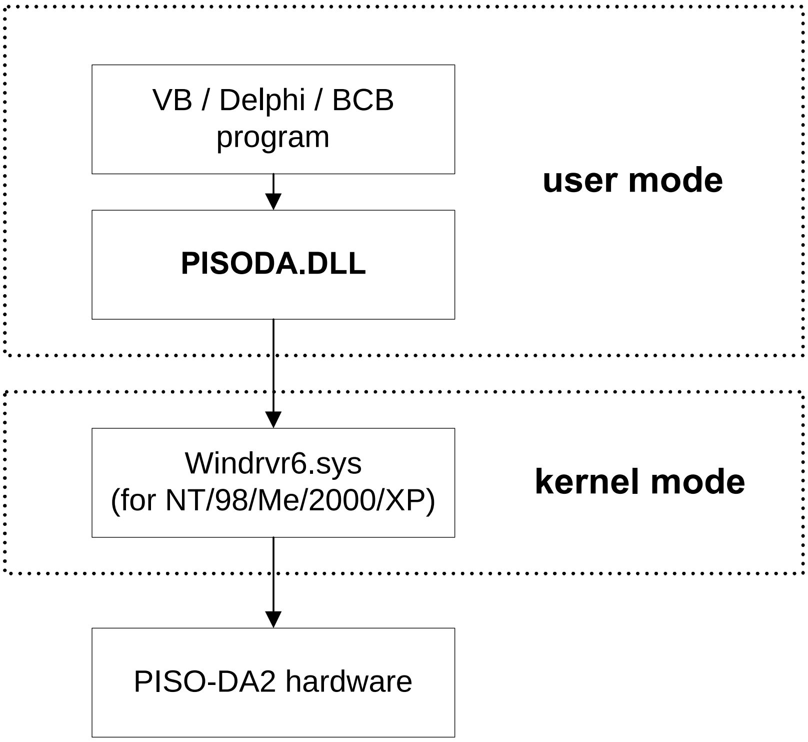
Windows NT/2000 – **WINNT\SYSTEM32\DRIVERS**
Windows 98/Me/XP – **WINDOWS\SYSTEM32\DRIVERS**

3.2 PnP Driver Installation

After installing the hardware (PISO-DA2) and power on your PC, Windows 98/Me/2000/XP can find a PCI card device and ask user to provide a PISODA2.inf to install hardware driver on the computer. If user has trouble to proceed this process, please refer to PnPinstall.pdf for more information.

4.DLL Driver

The DLL driver is the collection of function calls of the PISO-DA2 cards for Windows 98/Me/NT/2000/XP system. The application structure is presented as following figure. The user application program developed by designate tools like VB, Delphi and Borland C^{++} Builder can call PISODA.DLL driver in user mode. And then DLL driver will bypass the function call to Windrvr6.sys to access the hardware system.


VB / Delphi / BCB program
PISODA.DLL
user mode
Windrvr6.sys (for NT/98/Me/2000/XP)
kernel mode
PISO-DA2 hardware

4.1 Table of ErrorCode and ErrorString

Error Code	Error ID	Comment
0	PISODA_NoError	OK
1	PISODA_ActiveBoardError	This board can not be activated.
2	PISODA_ExceedFindBoards	The board number exceeds the current total board number (N).
3	PISODA_DriverNoOpen	Kernel driver can't be found.
4	PISODA_BoardNoActive	The board is not activated
5	PISODA_WriteEEPROMError	Fail to write data to EEPROM.
6	PISODA_ParameterError	Parameter is null or out of range

4.2 Function definition and description

All of the functions provided for PISO-DA2 are listed as below and the detail information for every function will be presented in the following section. However, in order to make the description simplify and clearly, the attribute of the input and output parameter of the function is indicated as [input] and [output] respectively, as shown in following table.

Keyword	Set parameter by user before calling this function?	Get the data from this parameter after calling this function?
[input]	Yes	No
[output]	No	Yes

Function definition
WORD PISODA_GetDllVersion();
WORD PISODA_ActiveBoard(BYTE BoardNo)
WORD PISODA_CloseBoard(BYTE BoardNo);
WORD PISODA_TotalBoard();
WORD PISODA_GetCardInf(BYTE BoardNo, DWORD *dwVID, DWORD *dwDID DWORD *dwSVID, DWORD *dwSDID, DWORD *dwSAuxID, DWORD *dwIrq);
BYTE PISODA_IsBoardActive(BYTE BoardNo);
WORD PISODA_DA_Hex(BYTE BoardNo,BYTE bChannel,WORD wValue);
WORD PISODA_DA(BYTE BoardNo,BYTE bChannel,BYTE bOpt,float fValue)
WORD PISODA_ReadJumper(BYTE BoardNo, BYTE *Jumper);
WORD PISODA_ReadEEP(BYTE BoardNo,WORD *wValue);
WORD PISODA_WriteEEP(BYTE BoardNo,WORD *wValue);
BYTE PISODA_InputByte(BYTE BoardNo,DWORD dwOffset);
void PISODA_OutputWord(BYTE BoardNo,DWORD dwOffset,WORD wValue);
WORD PISODA_InputWord(BYTE BoardNo,DWORD dwOffset);

4.2.1 PISODA_GetDllVersion

- **Description:**

  Obtain the version information of PISODA.DLL driver.

- **Syntax:**

  WORD PISODA_GetDllVersion(viod)

- **Parameter:**

  None

- **Return:**

  DLL version information. For example: If 101(hex) is return, it means driver version is 1.01.

4.2.2 PISODA_ActiveBoard

- **Description:**

  Activate the device. It must be called once before using the other functions of PISO-DA2 board.

- **Syntax:**

  WORD PISODA_ActiveBoard(BYTE BoardNo)

- **Parameter:**

  BoardNo: [input] PISO-DA2 board number (0~15).

- **Return:**

PISODA_NoError:	OK
PISODA_DriverNoOpen:	Kernel driver can not be found.
PISODA_ExceedFindBoards:	BoardNo exceeds the current total board number (N).
PISODA_ActiveBoardError :	This board can not be activated.

4.2.3 PISODA_CloseBoard(BYTE BoardNo)

- **Description:**

  Stop and close the PISO-DA2 kernel driver and release the resources the device resource from computer device resource. This method must be called once before exiting the user's application program.

- **Syntax:**

  WORD PISODA_CloseBoard(BYTE BoardNo)

- **Parameter:**

  BoardNo: [input] PISO-DA2 board number (0~15).

- **Return:**

PISODA_NoError:	OK
PISODA_BoardNoActive:	The board is not activated
PISODA_ExceedFindBoards:	BoardNo exceeds the current total board number (N).

4.2.4 PISODA_TotalBoard

- **Description:**

  Obtain the total board number of PISO-DA2 boards installed in the PCI bus.

- **Syntax:**

  WORD CALLBACK PISODA_TotalBoard(void)

- **Parameter:**

  None

- **Return:**

  Return the total board number.

4.2.5 PISODA_GetCardInf

- **Description:**

  Obtain the information of PISO-DA2 boards, which include vender ID, device ID and interrupt number.

- **Syntax:**

  WORD CALLBACK PISODA_GetCardInf(BYTE BoardNo, DWORD *dwVID, DWORD *dwDID, DWORD *dwSVID,DWORD *dwSDID, DWORD *dwSAuxID, DWORD *dwIrq)

- **Parameter:**

BoardNo:	[input] PISO-DA2 board number (0~15)
*dwVID:	[output] vendor ID of this board
*dwDID:	[output] device ID of this board
*dwSVID:	[output] sub-vendor ID of this board
*dwSDID:	[output] sub-device ID of this board
*dwSAuxID:	[output] sub-auxiliary ID of this board
*dwIrq:	[output] logical interrupt number of this board

- **Return:**

PISODA_NoError:	OK
PISODA_DriverNoOpen:	Kernel driver can not be opened.
PISODA_ExceedFindBoards:	BoardNo exceeds the current total board number (N).

4.2.6 PISODA_IsBoardActive

- **Description:**

  Obtain the information about the specific board is active or not.

- **Syntax:**

  BYTE PISODA_IsBoardActive(BYTE BoardNo)

- **Parameter:**

  BoardNo: [input] PISO-DA2 board number (0~15)

- **Return:**

  “0” means the board is inactive.
  “1” means the board is active.

4.2.7 PISODA_DA_Hex

- **Description:**

  Output the analog output data in Hex format through channel 1 or 2 of the PSIO-DA2. Note that the analog output can be configured voltage or current sink type, which is dependent on the hardware jumper setting.

- **Syntax:**

  WORD PISODA_DA_Hex(BYTE BoardNo,BYTE bChannel,WORD wValue)

- **Parameter:**

BoardNo:	[input] PISO-DA2 board number (0~15).
bChannel:	[input] D/A Channel number 1 or 2.
wValue:	[input] analog output value (0~0xfff).

- **Return:**

PISODA_NoError:	OK
PISODA_DriverNoOpen:	Kernel driver can not be opened.
PISODA_ExceedFindBoards:	BoardNo exceeds the current total board number (N).
PISODA_BoardNoActive:	The board is not activated
PISODA_ParameterError:	wValue is out of range.

4.2.8 PISODA_DA

- **Description:**

  Output the analog output data in float format through channel 1 or 2 of the PSIO-DA2. Note that the analog output can be configured voltage or current sink type, which is dependent on the hardware jumper setting.

- **Syntax:**

  WORD PISODA_DA(BYTE BoardNo, BYTE bChannel, BYTE bOpt, float fValue)

- **Parameter:**

BoardNo:	[input] PISO-DA2 board number (0~15)
bChannel:	[input] D/A Channel number 1 or 2.
bOpt:	[input] 0 for voltage output; 1 for current sink.
wValue:	[input] analog output value .

- **Return:**

PISODA_NoError:	OK
PISODA_DriverNoOpen:	Kernel driver can not be opened.
PISODA_ExceedFindBoards:	BoardNo exceeds the current total board number (N).
PISODA_BoardNoActive:	The board is not activated
PISODA_ParameterError:	wValue is out of range or jumper is improper.

Note: Refer to DEMO2 for more information.

4.2.9 PISODA_ReadJumper

- **Description:**

  Obtain the configuration status of the jumper JP3, JP4, JP5, JP7, JP8, and JP9. Please call PISODA_ActiveBoard first before using this function.

- **Syntax:**

  WORD PISODA_ReadJumper(BYTE BoardNo, BYTE *Jumper)

- **Parameter:**

  BoardNo: [input] PISO-DA2 board number (0~15)
  *Jumper: [output] a value of jumper status. Please refer to following table:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	JP9	JP8	JP7	JP5	JP4	JP3

Table 3.1: Jumper Status

Jumper	Jumper value=0	Jumper value =1
JP3 (ch1)	Current output is 0-20mA	Current output is 4-20mA
JP4 (ch1)	Reference voltage is –10V	Reference voltage is –5V
JP5 (ch1)	Bipolar setting	Unipolar setting
JP7 (ch2)	Current output is 0-20mA	Current output is 4-20mA
JP8 (ch2)	Reference voltage is –10V	Reference voltage is –5V
JP9 (ch2)	Bipolar setting	Unipolar setting

Example:
If *Jumper is 0x27 (0010 0111), then it means that JP3=1, JP4=1, JP5=1, JP7=0, JP8=0,J P9=1.

- **Return:**

PISODA_NoError:	OK
PISODA_DriverNoOpen:	Kernel driver can not be opened.
PISODA_ExceedFindBoards:	BoardNo exceeds the current total board number (N).
PISODA_BoardNoActive:	The board is not activated

4.2.10 PISODA_ReadEEP

- **Description:**

  Obtain the 64 words(128 bytes) data from the EEPROM of the PISO-DA2 board. Please call PISODA_ActiveBoard first before using this function.

- **Syntax:**

  WORD PISODA_ReadEEP(BYTE BoardNo,WORD *wValue)

- **Parameter:**

BoardNo:	[input] PISO-DA2 board number (0~15)
*wValue:	[output] the first word(16-bit) of data from EEPROM

- **Return:**

PISODA_NoError:	OK
PISODA_DriverNoOpen:	Kernel driver can not be opened.
PISODA_ExceedFindBoards:	BoardNo exceeds the current total board number (N).
PISODA_BoardNoActive:	The board is not activated

Note:

Refer to DEMO3 for more information.

4.2.11 PISODA_WriteEEP

- **Description:**

  Write 64 words (128 bytes) data into the EEPROM of the PISO-DA2 board. Please call PISODA_ActiveBoard first before using this function.

- **Syntax:**

  WORD PISODA_WriteEEP(BYTE BoardNo,WORD *wValue)

- **Parameter:**

BoardNo:	[input] PISO-DA2 board number (0~15)
*wValue:	[input] the first word(16-bit) of data.

- **Return:**

PISODA_NoError:	OK
PISODA_DriverNoOpen:	Kernel driver can not be opened.
PISODA_ExceedFindBoards:	BoardNo exceeds the current total board number (N).
PISODA_BoardNoActive:	The board is not activated
PISODA_WriteEEPROMError:	Fail to write data to EEPROM.

Note: Refer to DEMO3 for more information.

4.2.12 PISODA_InputByte

- **Description:**

  Obtain a byte data from the specific address mapping of the PISO-DA2 board. Please call PISODA_ActiveBoard first before using this function. This function is designed for advance user to access the hardware data based on the register of PISO-DA2.

- **Syntax:**

  BYTE PISODA_InputByte(BYTE BoardNo,DWORD dwOffset)

- **Parameter:**

BoardNo:	[input] PISO-DA2 board number (0~15)
dwOffset:	[input] The offset value of the base address of the PISO-DA2 board for the mapping address, from 0 to 0xff.

- **Return:**

  One byte value or data.

4.2.13 PISODA_OutputByte

- **Description:**

  Write a byte data to the defined address of the PISO-DA2 board. This function is designed for advance user to write into the hardware based on the register of PISO-DA2.

- **Syntax:**

  void PISODA_OutputByte(BYTE BoardNo,DWORD dwOffset,BYTE bValue)

- **Parameter:**

BoardNo:	[input] PISO-DA2 board number (0~15)
dwOffset:	[input] The offset of base address of the PISO-DA2 board for the mapping address, from 0 to 0xff.
bValue:	a byte value for output.

- **Return:**

  None

4.2.14 PISODA_InputWord

- **Description:**

  Obtain a word (two bytes) data from the specific mapping address of the PISO-DA2 board. Please call PISODA_ActiveBoard first before using this function. This function is designed for advance user to access the hardware data based on the register of PISO-DA2.

- **Syntax:**

  BYTE PISODA_InputWord(BYTE BoardNo,DWORD dwOffset)

- **Parameter:**

BoardNo:	[input] PISO-DA2 board number (0~15)
dwOffset:	[input] The offset of base address of the PISO-DA2 board for the mapping address, from 0 to 0xff.

- **Return:**

  One word value or data.

4.2.15 PISODA_OutputWord

- **Description:**

  Write a word(two bytes) data to the defined address of the PISO-DA2 board. This function is designed for advance user to write into the hardware based on the register of PISO-DA2.

- **Syntax:**

  void PISODA_OutputWord(BYTE BoardNo,DWORD dwOffset,WORD wValue)

- **Parameter:**

  BoardNo: [input] PISO-DA2 board number (0~15)

  dwOffset: [input] The offset of base address of the PISO-DA2 board for the mapping address, from 0 to 0xff

  wValue: a word value.

- **Return:**

  None

5.Demo Programs

5.1 Demo Programs for Windows

All of demo programs will not work normally if DLL driver would not be installed correctly. During the installation process of DLL driver, the install-shields will register the correct kernel driver to the operation system and copy the DLL driver and demo programs to the correct position based on the driver software package you have selected (Win98,Me,NT,win2000,XP). After driver installation, the related demo programs and development library and declaration header files for different development environments are presented as follows.

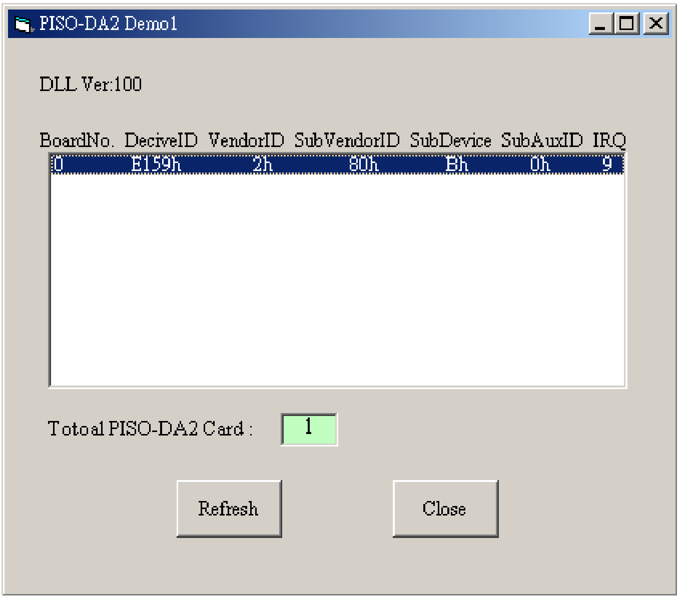
```
|--\Demo                              →demo program
   |--\BCB3                           → for Borland C++ Builder 3
   |    |--\PISODA.H                  → Header file
   |       \PISODA.LIB                → Linkage library for BCB only
   |
   |--\Delphi5                        → for Delphi 5
   |    |--\PISODA.PAS                → Declaration file
   |
   |--\VB6                            → for Visual Basic 6
       |--\PISODA.BAS                 →Declaration file
```

The list of demo programs:

DEMO1: Get cards information
DEMO2: D/A output
DEMO3: Read/Write from/to EEPROM and software calibration.
DEMO4: Two cards D/A output

DEMO1: Get cards information

Following figure is the result for the demo_1 program. It can be applied to obtain the hardware information of the PISO-DA2 board.



PISO-DA2 Demo1

DLL Ver:100

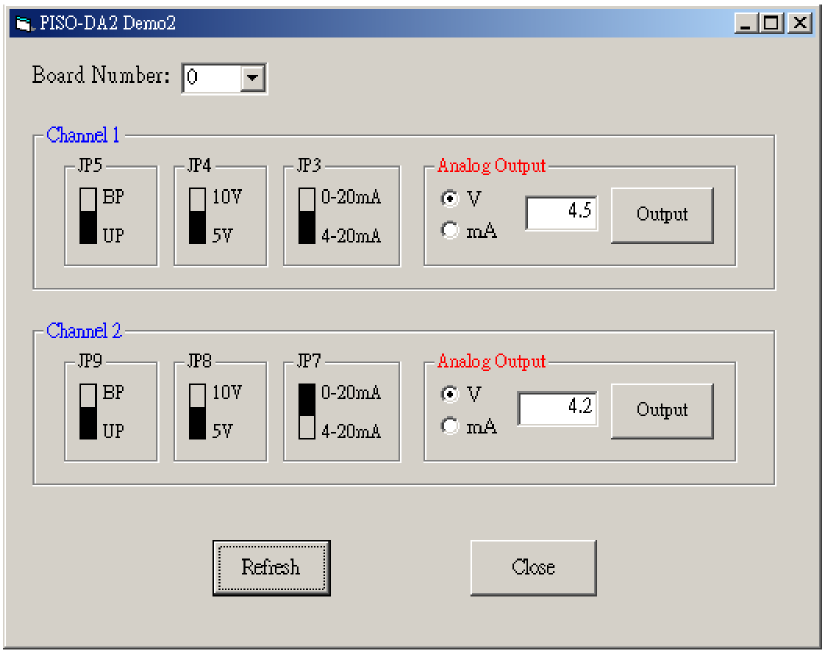
```
BoardNo. DeciveID  VendorID  SubVendorID  SubDevice  SubAuxID  IRQ
0        E159h     2h        80h          Bh         0h        9
```

Totoal PISO-DA2 Card : 1

Figure 6.1: The demo_1 program

DEMO2: D/A output

This demo program can be applied to drive the voltage or current output for channel 1 and 2 independently. And the jumper setting statuses of the hardware are also displayed on the graphic interface for setting checking. For more information, please refer to the hardware register, which is presented in section 7.



PISO-DA2 Demo2

Board Number: 0

Channel 1

JP5: BP / UP

JP4: 10V / 5V

JP3: 0-20mA / 4-20mA

Analog Output: ● V ○ mA 4.5

Channel 2

JP9: BP / UP

JP8: 10V / 5V

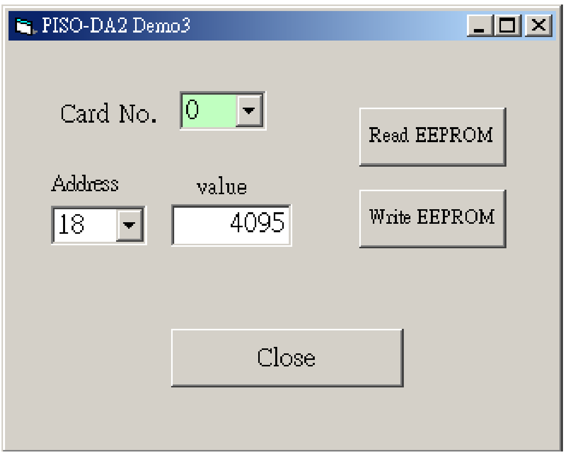
JP7: 0-20mA / 4-20mA

Analog Output: ● V ○ mA 4.2

Figure 6.2: The demo_2 program

DEMO2: D/A output

This program demonstrates the method for how to write the data to EEPROM and then read them out. Note that this method is based on the PSIODA_ReadEEP and PISODA_WriteEEP, which access the hardware by a word at one time.



PISO-DA2 Demo3

Card No. 0

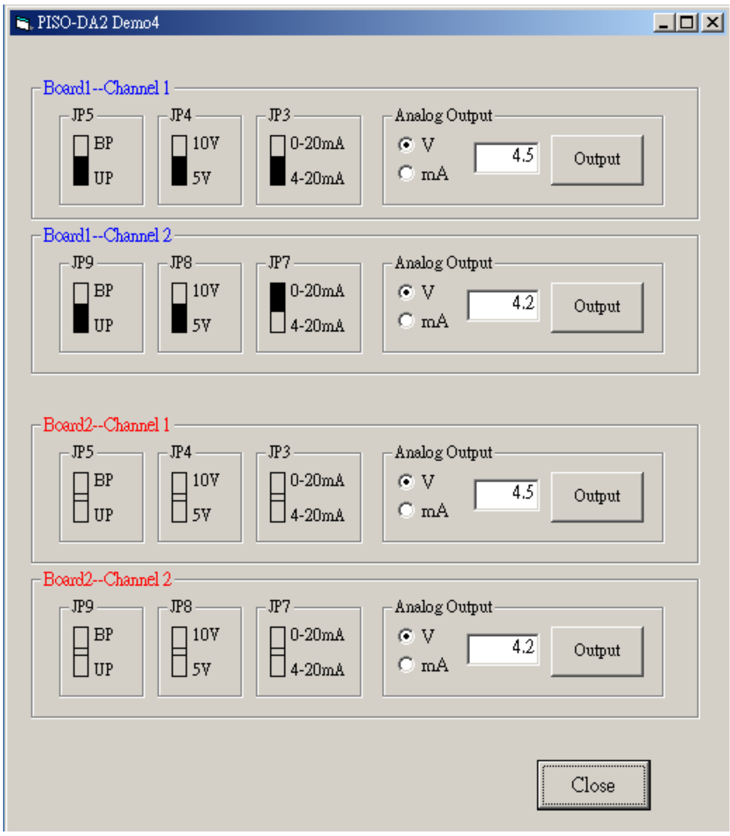
Address 18

value 4095

Figure 6.3: The demo_3 program

DEMO4: Multiboard D/A output

This demo program presents the same function as DEMO2 to output analog signal, but using 2 boards instead of one board.



PISO-DA2 Demo4

Board1--Channel 1: JP5 BP / UP; JP4 10V / 5V; JP3 0-20mA / 4-20mA; Analog Output ● V ○ mA 4.5

Board1--Channel 2: JP9 BP / UP; JP8 10V / 5V; JP7 0-20mA / 4-20mA; Analog Output ● V ○ mA 4.2

Board2--Channel 1: JP5 BP / UP; JP4 10V / 5V; JP3 0-20mA / 4-20mA; Analog Output ● V ○ mA 4.5

Board2--Channel 2: JP9 BP / UP; JP8 10V / 5V; JP7 0-20mA / 4-20mA; Analog Output ● V ○ mA 4.2

Figure 6.4: The demo_4 program

5.2 Demo Programs for DOS

There are several of demo programs given in the company floppy disk or CD-ROM. The demo program and the driver are installed into disk as follows:

- \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

For every development environments, it fully includes the following demo programs.

DEMO1: INT_CHAN_0 & INT_CHAN_1 timer interrupt demo
DEMO2: D/A Output
DEMO3: Write the data to EEPROM
DEMO4: Read data from EEPROM.
DEMO5: Software calibration

6. The Hardware Register

The detailed descriptions of the registers format for PISO-DA2 will be presented here for advance user. This information is quite useful for the programmers who hope to handle the card by themselves. However we suggest that user need to understand the hardware system more clearly before starting to design the program for controlling it by them. The following section will help users to understand the registers system of the PISO-DA2.

6.1 The hardware information of PISO-DA2

The Hardware IDs of PISO-DA2 are given as follows:

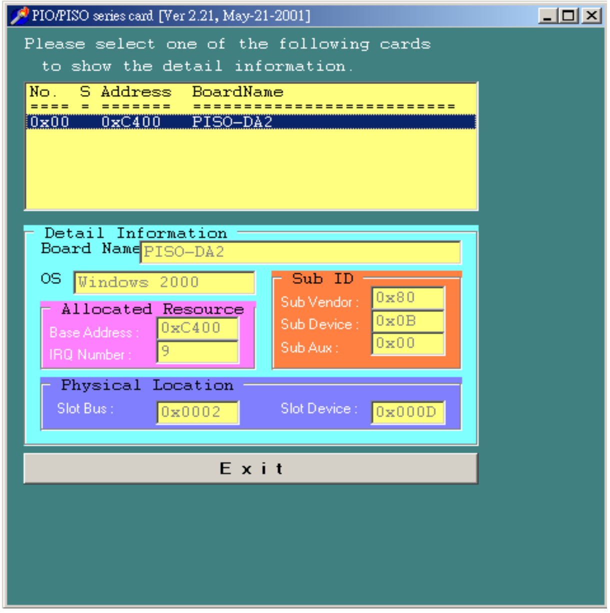
<Rev1.0~Rev4.0>

Vendor ID=	0xE159
Device ID=	0x02
Sub-Vendor ID=	0x80
Sub-Device ID=	0x0B
Sub-Aux ID=	0x00

<Rev5.0>

Vendor ID=	0x0E159
Device ID=	0x01
Sub-Vendor ID=	0x4280
Sub-Device ID=	0x03
Sub-Aux ID=	0x00

The utility program under Windows operation system, **PIO_PISO.EXE**, can detect and display all the hardware information of PIO/PISO cards installed in the PC. User can find this utility in the company CD or the website (http://www.icpdas.com). It is very useful for understanding hardware information of all PIO/PISO series card. After executing the utility, the detail information for all PIO/PISO cards installed in the PC can be demonstrated as follows. The detail definition of the hardware system for PIO/PISO PCI interface board is represented in Table 6.1.



PIO/PISO series card [Ver 2.21, May-21-2001]

```
Please select one of the following cards
  to show the detail information.

No.  S Address  BoardName
==== = =======  ===========================
0x00   0xC400   PISO-DA2
```

Detail Information

Board Name: PISO-DA2

OS: Windows 2000

Allocated Resource

Base Address : 0xC400

IRQ Number : 9

Sub ID

Sub Vendor : 0x80

Sub Device : 0x0B

Sub Aux : 0x00

Physical Location

Slot Bus : 0x0002

Slot Device : 0x000D

6.2 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. It is strongly recommended that users themselves do not change the I/O address. The plug & play BIOS of the PCI-board will automatically assign the proper I/O address to each PIO/PISO series card very well. The I/O addresses of the PISO-DA2 are given as follows, which is based on the base address **wBase**.

Table 6.2 : I/O address of the PISO-DA2 where wBase.

Address	Read	Write
wBase+0x0	RESET control register	RESET control register
wBase+0x2	AUX control register	AUX control register
wBase+0x3	AUX data register	AUX data register
wBase+0x5	INT mask control register	INT mask control register
wBase+0x7	AUX pin status register	AUX pin status register
wBase+0x2a	INT polarity control register	INT polarity control register
wBase+0xc0	Not used	Write to the high byte of D/A
wBase+0xc4	Not used	Write to the low byte of D/A
wBase+0xc8	Not used	Write to the high byte of D/A
wBase+0xcc	Not used	Write to the low byte of D/A

wBase+0xd0	Read from 8254-counter0	Write to 8254-counter0
wBase+0xd4	Read from 8254-counter1	Write to 8254-counter1
wBase+0xd8	Read from 8254-counter2	Write to 8254-counter2
wBase+0xdc	Read from 8254 control word	Write to 8254 control word
wBase+0xe0	Read in Jumper status	Not used

6.2.1 RESET\ the Control Register

When the PC is first power-up, the RESET\ signal is in Low-state. **This will disable all D/A operations.** The user has to set the RESET\ signal to High-state before using any D/A command. Note that **wBase** is the base address of PISO-DA2 board mapping from your PC.

Table 6.3: Read/Write 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\

6.2.2 AUX Control Register

This register is designed for feature extension and for enable or disable of the reading/writing data from or to the EEPROM. And it is reversed for internal utilization and do not apply this control register under any consideration.

Table 6.4: Aux Control Register

(Read/Write): wBase+2

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Aux7	Aux6	Aux5	Aux4	Aux3	Aux2	Aux1	Aux0

6.2.3 AUX data Register

This register controls the read/write function of the EEPROM on board. There are all reversed by ICPDAS internal use. If the user wants to access this EEPROM, please refer to function read/write of the EEPROM provided by the driver toolkit.

Table 6.5: Aux data Register

(Read/Write): wBase +3

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Aux7	Aux6	Aux5	Aux4	Aux3	Aux2	Aux1	Aux0

6.2.4 INT Mask Control Register

The INT mask control register is presented as following table. The detail function for these control register is described as below.

Table 6.6: INT mask control Register

(Read/Write): wBase+5

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	EN1	EN0

EN0=0→ disable INT0 to be an interrupt signal (default)
EN0=1→ enable INT0 to be an interrupt signal

EN1=0→ disable INT1 to be an interrupt signa (default)
EN1=1→ enable INT1 to be an interrupt signal

The following is the partial programs for DOS C development environment enable or disable interrupt function. For more information, please refer to the DOS demo program demo1.c .

```
outportb(wBase+5,0);          // disable all interrupts
outportb(wBase+5,1);          // enable interrupt of INT0
outportb(wBase+5,2);          // enable interrupt of INT1
outportb(wBase+5,3);          // enable all two channels of interrupt
```

6.2.5 Aux Status Register

Based on the auxiliary status register, Aux0 (bit 0) and Aux 1(bit 1) stand as INT0 and INT1 signal respectively. Aux2~3 (bit 2~3) represents the control register of the EEPROM and Aux4~7 (bit 4~7) depicts the Aux-ID. Generally, the Aux 0~1 are used as interrupt sources. Interrupt service has to check this register to start service routing.

Table 6.7: AUX Status Register

(Read/Write): wBase+7

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Aux7	Aux6	Aux5	Aux4	Aux3	Aux2	Aux1	Aux0

6.2.6 Interrupt Polarity Control Register

The interrupt polarity control register is presented as following table. It is used to invert the interrupt signal or not. The detail function for these control register is described as below.

Table 6.8: Interrupt polarity control Register

(Read/Write): wBase+0x2A

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	×	×	INV1	INV0

INV0=0→ invert signal from INT0;
INV0=1→ do not invert signal from INT0;
INV1=0→ invert signal from INT1;
INV1=1→ do not invert signal from INT0;

The following is the partial programs for DOS C development environment enable or disable inverting function for interrupt signal.

```
outportb(wBase+0x2a,0);   /* select the inverted input from all 2 channels       */
outportb(wBase+0x2a,3);   /* select the non-inverted input from all 2 channels */

outportb(wBase+0x2a,2);   /* select the inverted input of INT0                   */
                          /* select the non-inverted input from the others       */
```

6.2.7 D/A Data Output

Table 6.9 and 6.10 is the output data buffer for D/A channel-1 and Table 6.11 and 6.12 is the output data buffer for D/A channel-2. These registers are write only and user can use these to output the data through analog output CH1 and CH2.

Table 6.9: high byte of D/A channel-1

(Write): wBase+0xc0

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

Table 6.10: Low byte of D/A channel-1

(Write): wBase+0xc4

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

Table 6.11: high byte of D/A channel-2

(Write): wBase+0xc8

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

Table 6.12: Low byte of D/A channel-2

(Write): wBase+0xcc

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

6.2.8 Jumper Status Register

This register shows the status of Hardware jumper setting. About the detail information, please refer to the following description.

Table 6.13: Jumper status register

(Read): wBase+0xe0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X	x	JP9	JP8	JP7	JP5	JP4	JP3

JP3=0→ JP3 is set at 0-20mA for the current output of channel 1.
JP3=1→ JP3 is set at 4-20mA for the current output of channel 1

JP4 =0→ JP4 is set at –10V for internal refernce voltage source of channel 1.
JP4 =1→ JP4 is set at –5V for internal refernce voltage source of channel 1.

JP5=0→ JP5 is set at Bipolar for channel 1
JP5=1→ JP5 is set at Unipolar for channel 1

JP7=0→ JP7 is set at 0-20mA for the current output of channel 2
JP7=1→ JP7 is set at 4-20mA for the current output of channel 2

JP8 =0→ JP8 is set at –10V for internal refernce voltage source of channel 2
JP8 =1→ JP8 is set at –5V for internal refernce voltage source of channel 2

JP9=0→ JP9 is set at Bipolar for channel 2
JP9=1→ JP9 is set at Unipolar for channel 2