



PISO-813 Series Classic Driver DLL Software Manual

Version 2.0, Feb. 2014

SUPPORTS

Board includes PISO-813 and PISO-813U.

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

Copyright © 2014 by ICP DAS. All rights are reserved.

TRADEMARK

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

CONTACT US

If you have any question, please feel to contact us at:

service@icpdas.com; service.icpdas@gmail.com

We will give you quick response within 2 workdays.



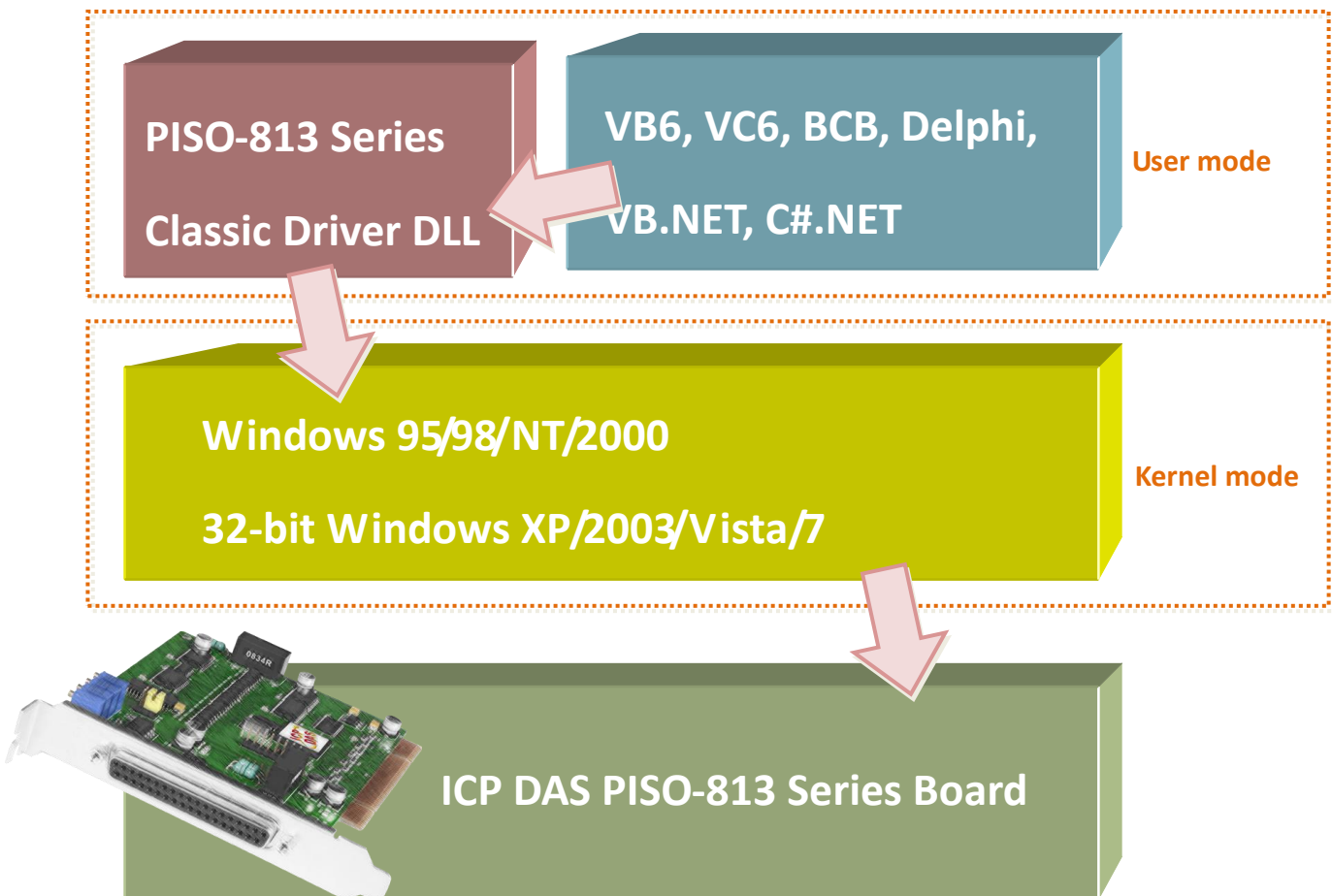
TABLE OF CONTENTS

1.	INTRODUCTION	2
1.1	OBTAINING THE DRIVER INSTALLER PACKAGE	3
1.2	DRIVER INSTALLING PROCEDURE	4
1.3	PNP DRIVER INSTALLATION	6
2.	DLL FUNCTION DESCRIPTIONS	8
2.1	ERROR CODE TABLE	10
2.2	A/D GAIN CODE TABLE	11
2.3	TEST FUNCTIONS	12
	<i>PISO813_GetDllVersion</i>	12
	<i>PISO813_ShortSub</i>	12
	<i>PISO813_FloatSub</i>	13
2.4	DRIVER FUNCTIONS	14
	<i>PISO813_GetDriverVersion</i>	14
	<i>PISO813_DriverInit</i>	14
	<i>PISO813_DriverClose</i>	15
	<i>PISO813_GetConfigAddressSpace</i>	15
2.5	DIGITAL INPUT/OUTPUT FUNCTIONS	17
	<i>PISO813_OutputByte</i>	17
	<i>PISO813_InputByte</i>	17
	<i>PISO813_OutputWord</i>	18
	<i>PISO813_InputWord</i>	18
2.6	ANALOG INPUT FUNCTIONS	19
	<i>PISO813_SetChGain</i>	19
	<i>PISO813_AD2F</i>	20
	<i>PISO813_AD_Hex</i>	21
	<i>PISO813_AD_Float</i>	22
	<i>PISO813_Ads_Hex</i>	23
	<i>PISO813_Ads_Float</i>	24
3.	DEMO PROGRAMS	25
3.1	FOR MICROSOFT WINDOWS	25
3.2	FOR DOS	28
4.	PROGRAMS ARCHITECTURE	29
5.	PROBLEMS REPORT	30

1. Introduction

The software is a collection of digital I/O, analog input subroutines for PISO-813 series card add-on cards for **Windows 95/98/NT/2000** and **32-bit Windows XP/2003/Vista/7** applications. The application structure is presented in the following diagram.



The subroutines in **PISO813.DLL** are easy understanding as its name standing for. It provides powerful, easy-to-use subroutine for developing your data acquisition application. Your program can call these DLL functions by **VB, VC, Delphi, BCB, VB.NET 2005 and C#.NET 2005** easily. To speed-up your developing process, some demonstration source program are provided.



1.1 Obtaining the Driver Installer Package

PIO-821 series card can be used on Linux and Windows 95/98/NT/2000 and 32-bit XP/2003/Vista/7 based systems, and the drivers are fully Plug and Play (PnP) compliant for easy installation.

The driver installer package for the PISO-813 series can be found on the supplied CD-ROM, or can be obtained from the ICP DAS FTP web site. The location and addresses are indicated in the table below:

	CD:\\ NAPDOS\\PCI\\PISO-813\\DLL_OCX\\
	<u>ftp://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/piso-813/dll_ocx/</u>
	<u>http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/piso-813/dll_ocx/</u>

Install the appropriate driver for your operating system, as follows:

Name	OS
Win2K_XP_7	For Windows 2000 and 32- bit Windows XP/Vista/7
Win98	For Windows 95/98/ME
WinNT	For Windows NT 4.0

1.2 Driver Installing Procedure

Before the driver installation, you must complete the hardware installation. For detailed information about the hardware installation, please refer to hardware user manual of PISO-813 series card.

The hardware user manual is contained in:



CD:\NAPDOS\PCI\PISO-813 \Manual\



<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/piso-813/manual/>

To install the PISO-813 series classic drivers, follow the procedure described below:

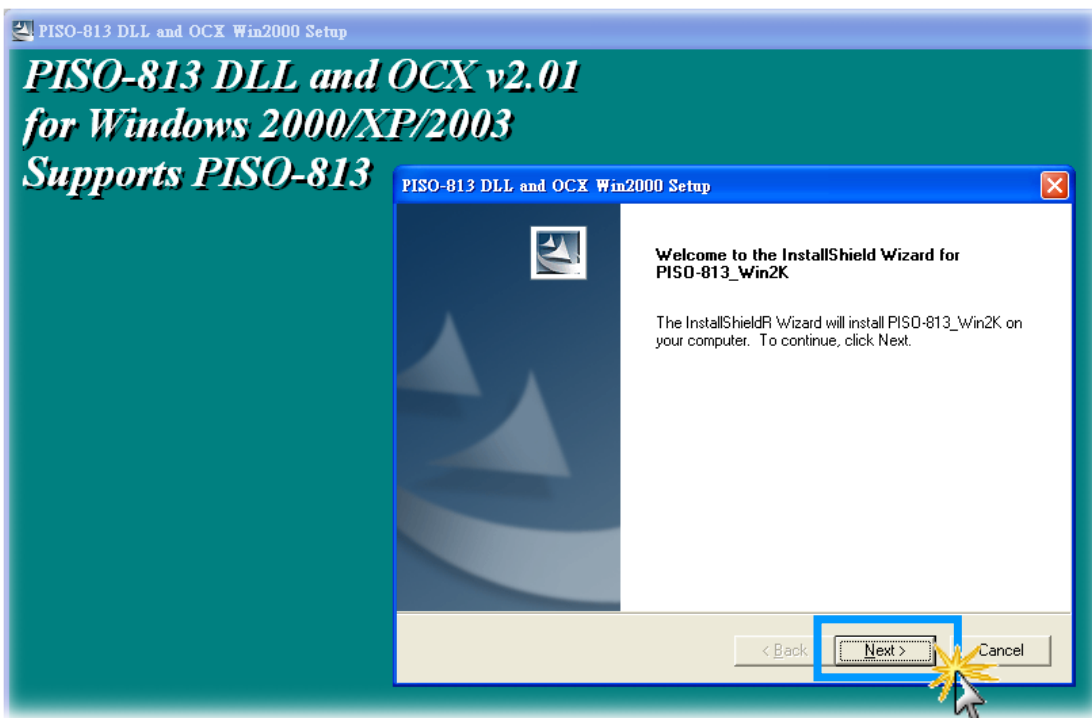


PISO_813_Win2K_v202
ICPDAS

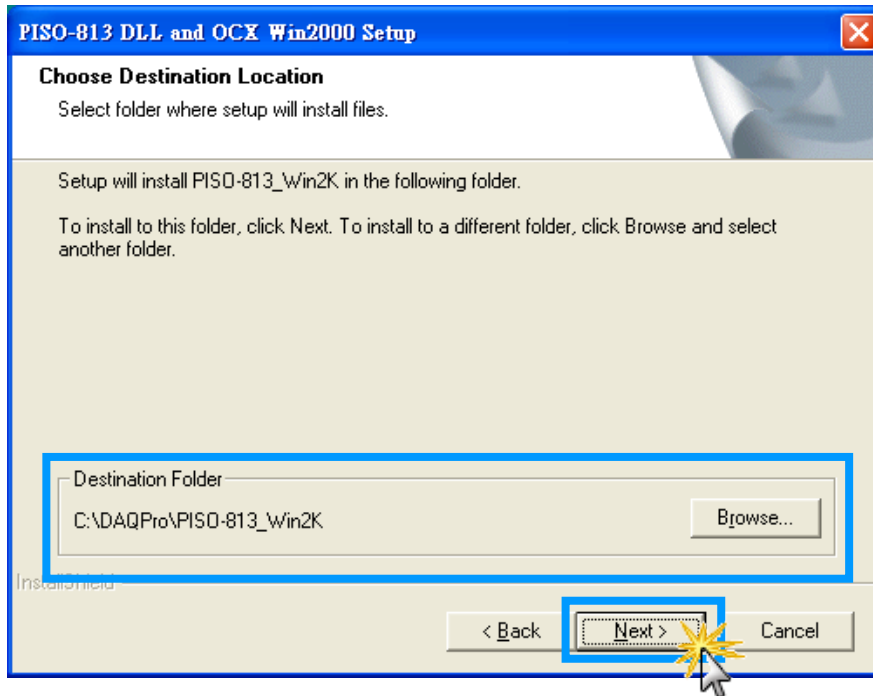


Step 1: Double-Click **“PISO-813_Win2K_xxxx.exe”** to install driver.

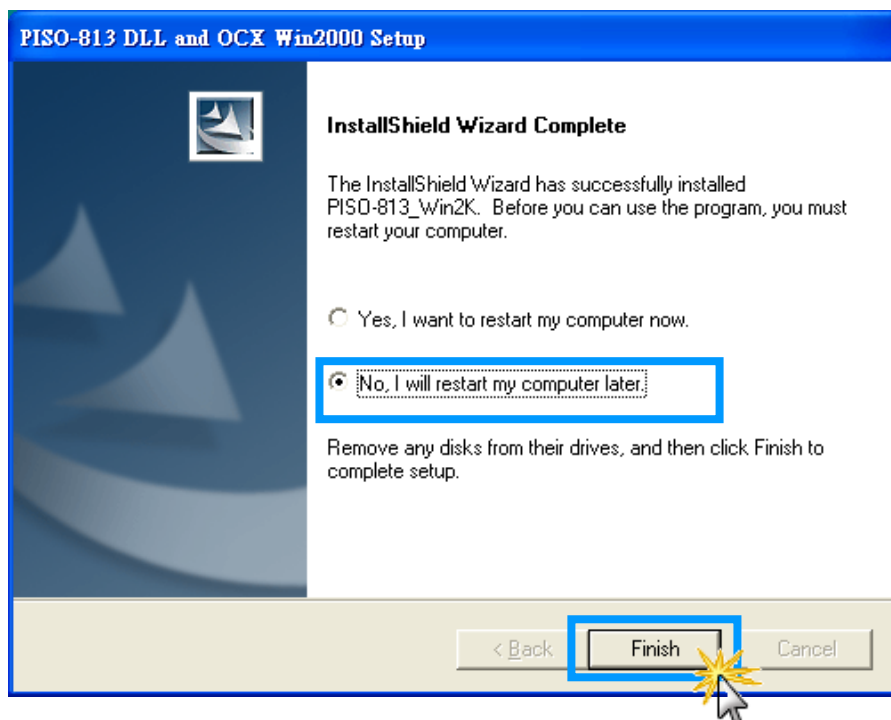
Step 2: Click the **“Next>”** button to start the installation on the **“PISO-813 DLL and OCX Win2000 Setup”** window.



Step 3: Click the “**Next>**” button to install the driver into the **default** folder.



Step 4: Selection “**No, I will restart my computer later**” and then click the “**Finish**” button.



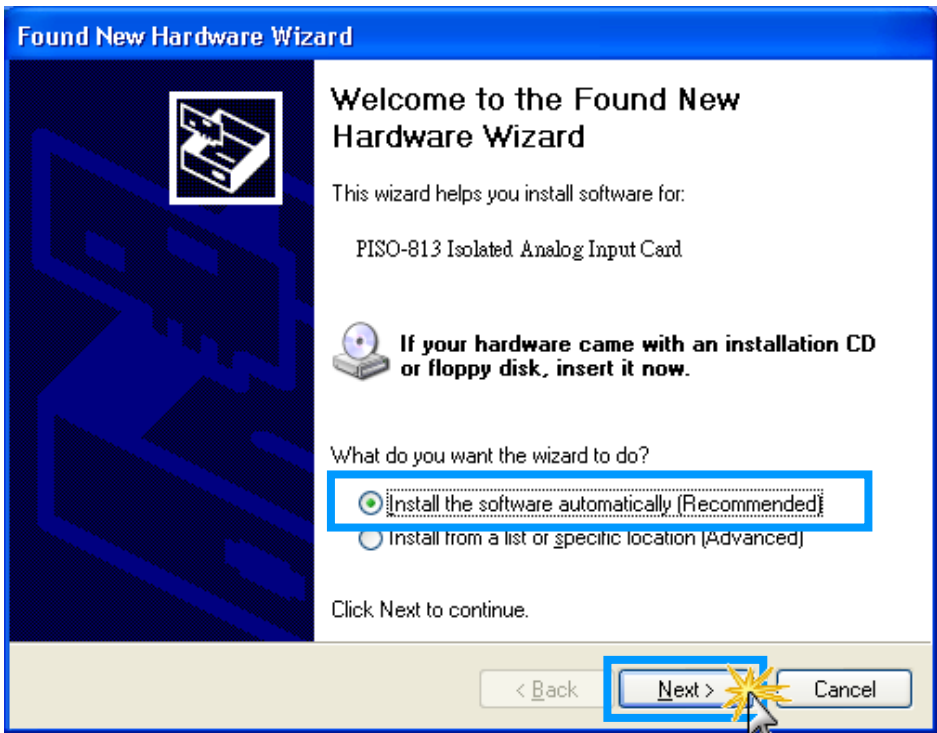
1.3 PnP Driver Installation

Step 1: The system should find the new card and then continue to finish the Plug&Play steps.

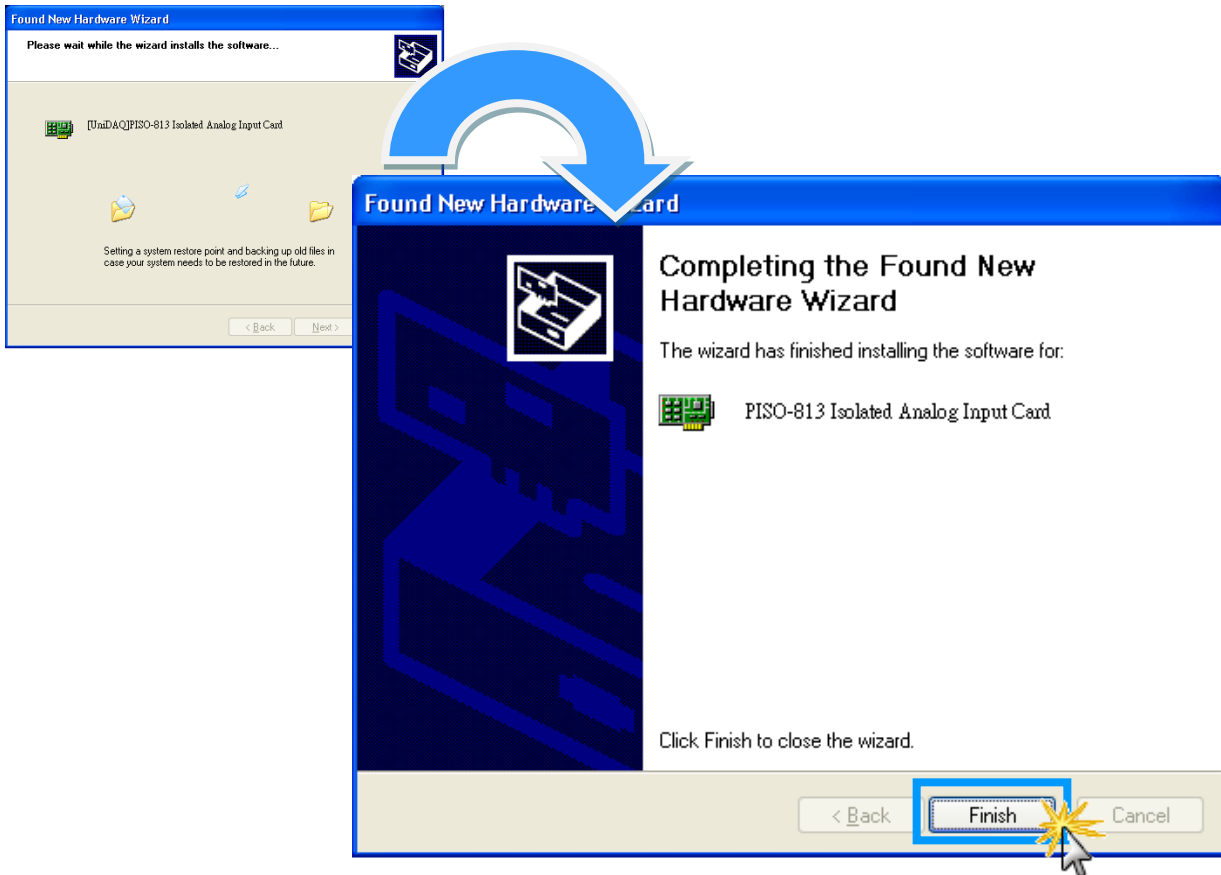
Note: Some operating system (such as Windows Vista/7) will find the new card and make it work automatically, so the Step2 to Step4 will be skipped.



Step 2: Select “Install the software automatically [Recommended]” and click the “Next>” button.



Step 3: Click the **“Finish”** button.



Step 4: Windows pops up **“Found New Hardware”** dialog box again.



2. DLL Function Descriptions

All of the functions provided for PISO-813 series card are listed below in Tables 2-1 to 2-4. This list of functions is expanded on in the text that follows. However, in order to make a clear and simplified description of the functions, the attributes of the input and output parameters for every function is indicated as [input] and [output] respectively, as shown in following table. Furthermore, the error code of all functions supported by PISO-813 is also listed in Section 2-1.

Keyword	Parameter must be set by the user before calling the function	Data/value from this parameter is retrieved after calling the function
[Input]	Yes	No
[Output]	No	Yes
[Input, Output]	Yes	Yes

Table2-1: Test Functions Table of PISO813.DLL

Section	Function Definition
2.3	Test Functions
	float PISO813_FloatSub(float fA, float fB);
	short PISO813_ShortSub(short nA, short nB);
	WORD PISO813_GetDllVersion(void);

Table2-2: Driver Functions Table of PISO813.DLL

Section	Function Definition
2.4	Driver Functions
	WORD PISO813_DriverInit(void);
	void PISO813_DriverClose(void);
	WORD PISO813_SearchCard(WORD *wBoards, DWORD dwPIOCardID);
	WORD PISO813_GetDriverVersion(WORD *wDriverVersion);
	WORD PISO813_GetConfigAddressSpace (WORD wBoardNo, DWORD *wAddrBase, WORD *wIrqNo, WORD *wSubVendor, WORD *wSubDevice, WORD *wSubAux, WORD *wSlotBus, WORD *wSlotDevice);

Table2-3: DIO Functions Table of PISO813.DLL

Section	Function Definition
2.5	Digital Input/Output Functions
	void PISO813_OutputWord(DWORD wPortAddress, DWORD wOoutData);
	void PISO813_OutputByte(DWORD wPortAddr, WORD bOutputValue);
	DWORD PISO813_InputWord(DWORD wPortAddress);
	WORD PISO813_InputByte(DWORD wPortAddr);

Table2-4: A/D Functions Table of PISO813.DLL

Section	Function Definition
2.6	Analog Input Functions
	WORD PISO813_SetChGain(DWORD wAddrBase, WORD wChannel, WORD wGainCode);
	WORD PISO813_AD_Hex(DWORD wAddrBase,);
	WORD PISO813_Ads_Hex(DWORD wAddrBase, WORD *wBuffer, DWORD dwDataNo);
	float PISO813_AD_Float(DWORD wAddrBase, WORD wJump20v, WORD wBipolar);
	float PISO813_Ads_Float (DWORD wAddrBase, WORD wJump20v, WORD wBipolar, float *fBuffer, DWORD dwDataNo);
	float PISO813_AD2F(WORD whex, WORD wGainCode, WORD wJump20v, WORD wBipolar);

2.1 Error Code Table

For the most errors, it is recommended to check:

1. Does the device driver installs successful?
2. Does the card have plugged?
3. Does the card conflicts with other device?
4. Close other applications to free the system resources.
5. Try to use another slot to plug the card.
6. Restart your system to try again.

Error Code	Error ID	Error String
0	PISO813_NoError	OK (No Error)
1	PISO813_DriverOpenError	Device driver can't be opened
2	PISO813_DriverNoOpen	The PISO813_DriverInit() function must be called first
3	PISO813_GetDriverVersionError	Get driver version error
4	PISO813_CallDriverError	Call driver is error
5	PISO813_FindBoardError	Cannot find board
6	PISO813_ExceedBoardNumber	The board number exceeds the maximum board number (7).
0xffff	PISO813_TimeOutError	Delay time out
-100.0	PISO813_ADError2	A/D converter error

2.2 A/D Gain Code Table

➤ JP2: Bipolar Mode GAIN Control Code Table

GAIN	Input Range		GAIN2	GAIN1	GAIN0	Gain Code
	JP1: 10 V	JP1: 20 V				
1	±5 V	±10 V	0	0	0	0x0
2	±2.5 V	±5 V	0	0	1	0x1
4	±1.25 V	±2.5 V	0	1	0	0x2
8	±0.625 V	±1.25 V	0	1	1	0x3
16	Not Use	±0.625 V	1	0	0	0x4

➤ JP2: Unipolar Mode GAIN Control Code Table

GAIN	Input Range		GAIN2	GAIN1	GAIN0	Gain Code
	JP1: 10 V	JP1: 20 V				
1	0 ~ 10 V	Not Use	0	0	0	0
2	0 ~ 5 V	Not Use	0	0	1	1
4	0 ~ 2.5 V	Not Use	0	1	0	2
8	0 ~ 1.25 V	Not Use	0	1	1	3
16	0 ~ 0.625 V	Not Use	1	0	0	4

2.3 Test Functions

PISO813_GetDllVersion

To get the version number of PISO813.DLL.

- **Syntax:**
WORD **PISO813_GetDllVersion**(void);
- **Parameters:**
None
- **Returns:**
DLL version information.
For example: If 200(hex) value is return, it means driver version is 2.00.

PISO813_ShortSub

To perform the subtraction as **nA - nB** in short data type. This function is provided for testing DLL linkage purpose.

- **Syntax:**
short **PISO813_ShortSub**(short **nA**, short **nB**);
- **Parameters:**

nA
[Input] 2 bytes short data type value

nB
[Input] 2 bytes short data type value
- **Returns:**
The value of **nA - nB**

PISO813_FloatSub

To perform the subtraction as **fA - fB** in float data type. This function is provided for testing DLL linkage purpose.

➤ **Syntax:**

```
float PISO813_FloatSub(float fA, float fB);
```

➤ **Parameters:**

fA

[Input] 4 bytes floating point value

fB

[Input] 4 bytes floating point value

➤ **Returns:**

The value of fA - fB

2.4 Driver Functions

PISO813_GetDriverVersion

This subroutine will read the version number of PISO-813 driver.

➤ **Syntax:**

```
WORD PISO813_GetDriverVersion(WORD *wDriverVersion);
```

➤ **Parameters:**

wDriverVersion

[Output] address of wDriverVersion

➤ **Returns:**

PISO813_NoError	OK
PISO813_DriverNoOpen	The PISO-813 driver no open
PIDO813_GetDriverVersionError	Read driver version error

PISO813_DriverInit

This subroutine will open the PISO-813 driver and allocate the resource for the device. This function must be called once before calling other PISO-813 functions.

➤ **Syntax:**

```
WORD PISO813_DriverInit();
```

➤ **Parameters:**

None

➤ **Returns:**

PISO813_NoError	OK
PISO813_DriverNoOpen	Open PISO-813 driver error

PISO813_DriverClose

W This subroutine will close the PISO-813 Driver and release the resource from the device. This function must be called once before exit the user's application.

➤ **Syntax:**

```
void PISO813_DriverClose();
```

➤ **Parameters:**

None

➤ **Returns:**

None

PISO813_GetConfigAddressSpace

Get the I/O address of PISO-813 board n.

➤ **Syntax:**

```
WORD PISO813_GetConfigAddressSpace (WORD wBoardNo,  
                                     DWORD *wAddrBase,  
                                     WORD *wIrqNo,  
                                     WORD *wSubVendor,  
                                     WORD *wSubDevice,  
                                     WORD *wSubAux,  
                                     WORD *wSlotBus,  
                                     WORD *wSlotDevice  
                                     );
```

➤ **Parameters:**

wBoardNo

[Input] PISO-813 board number.

wAddrBase

[Output] The base address of PISO-813 board. Only the low WORD is valid.

wIrqNo

[Output] The IRQ number that the PISO-813 board using.

wSubVendor

[Output] Sub Vendor ID.

wSubDevice

[Output] Sub Device ID.

wSubAux

[Output] Sub Aux ID.

wSlotBus

[Output] Slot Bus number.

wSlotDevice

[Output] Sub Device ID.

➤ **Returns:**

PISO813_NoError	OK
PISO813_FindBoardError	Handshake check error
PIDO813_ExceedBoardError	wBoardNo is invalidated

2.5 Digital Input/Output Functions

PISO813_OutputByte

This subroutine will send the 8 bits data to the desired I/O port.

➤ **Syntax:**

```
void PISO813_OutputByte(DWORD wPortAddr, WORD bOutputVal);
```

➤ **Parameters:**

wPortAddr

[Input] I/O port addresses, please refer to function PISO813_GetConfigAddressSpace().
Only the low WORD is valid.

bOutputVal

[Input] 8 bit data send to I/O port. Only the low BYTE is valid.

➤ **Returns:**

None

PISO813_InputByte

This subroutine will input the 8 bit data from the desired I/O port.

➤ **Syntax:**

```
WORD PISO813_InputByte(DWORD wPortAddr);
```

➤ **Parameters:**

wPortAddr

[Input] I/O port addresses, please refer to function PISO813_GetConfigAddressSpace().
Only the low WORD is valid.

➤ **Returns:**

16 bits data with the leading 8 bits are all 0. (Only the low BYTE is valid.)

PISO813_OutputWord

This subroutine will send the 16 bits data to the desired I/O port.

➤ **Syntax:**

```
void PISO813_OutputWord(DWORD wPortAddr, WORD wOutputVal);
```

➤ **Parameters:**

wPortAddr

[Input] I/O port addresses, please refer to function PISO813_GetConfigAddressSpace().
Only the low WORD is valid.

wOutputVal

[Input] 16 bit data send to I/O port. Only the low WORD is valid.

➤ **Returns:**

None

PISO813_InputWord

This subroutine will input the 16 bit data from the desired I/O port.

➤ **Syntax:**

```
WORD PISO813_InputWord(DWORD wPortAddr);
```

➤ **Parameters:**

wPortAddr

[Input] I/O port addresses, please refer to function PISO813_GetConfigAddressSpace().
Only the low WORD is valid.

➤ **Returns:**

16 bits data. Only the low WORD is valid.

2.6 Analog Input Functions

PISO813_SetChGain

This subroutine will setting the channel number and Gain Code (Refer to [Section 2.2](#)) for the AD converter.

➤ **Syntax:**

```
WORD PISO813_SetChGain(DWORD wBase, WORD wChannel, WORD wGainCode);
```

➤ **Parameters:**

wBase

[Input] I/O port base addresses, please refer the PISO813_GetConfigAddressSpace().

wChannel

[Input] A/D channel number 0 to 31.

wGainCode

[Input] The value is 0 to 4, refer to Section 2.2.

➤ **Returns:**

PISO813_NoError OK

PISO813_AD2F

This subroutine will convert the Hex value to floating value depending on Gain Code , Bipolar/Unipolar and 10 V/20 V.

➤ **Syntax:**

```
float PISO813_AD2F(WORD wHexValue, WORD wGainCode, WORD wJump20v,  
                  WORD wBipolar);
```

➤ **Parameters:**

wHexValue

[Input] Hex Value 0 to 0xffff.

wGainCode

[Input] The value is 0 to 4. Refer to Section 2.2 for detail information.

wJump20v

[Input] 1: 20 V (HW default);
0: 10 V

wBipolar

[Input] 1: Bipolar (HW default)
0: Unipolar

➤ **Returns:**

PISO813_ADError2	A/D converter error (return -100.0)
Other value	The floating-point value of A/D conversion (-10 to 10)

PISO813_AD_Hex

This subroutine will perform an A/D conversion by polling. The A/D converter is 12 bits for PISO-813. Refer to PISO813_SetChGain().

➤ **Syntax:**

WORD PISO813_AD_Hex(DWORD wBase);

➤ **Parameters:**

wBase

[Input] I/O port base addresses, please refer to PISO813_GetConfigAddressSpace().

➤ **Returns:**

PISO813_TimeOutError

A/D converter error (return 0xffff)

Other value

The **Hex value** of A/D conversion (0 to 0x0ffff)

PISO813_AD_Float

This subroutine will perform an A/D conversion by polling. The A/D converter is 12 bits for PISO-813. This subroutine will compute the result according to the configuration code ([Section 2.2](#)). Refer to PISO813_SetChGain().

➤ **Syntax:**

```
float PISO813_AD_Float(DWORD wBase , WORD wJump20v, WORD wBipolar);
```

➤ **Parameters:**

wBase

[Input] I/O port base addresses, please refer to PISO813_GetConfigAddressSpace().

wJump20v

[Input] 1: 20 V (HW default);
0: 10 V

wBipolar

[Input] 1: Bipolar (HW default)
0: Unipolar

➤ **Returns:**

PISO813_ADError2	A/D converter error (return -100.0)
Other value	The floating-point value of A/D conversion (-10 to 10)

PISO813_Ads_Hex

This subroutine will perform a number of A/D conversions by polling. This subroutine is very similar to PISO813_AD_Hex except that this subroutine will perform `wCount` of conversions instead of just one conversion. After A/D conversing, the A/D data are stored in a buffer in Hex format. The `wBuf` is the starting address of this data buffer. Refer to PISO813_SetChGain().

➤ **Syntax:**

WORD PISO813_Ads_Hex (DWORD `wBase`, WORD `wBuf[]`, DWORD `wCount`);

➤ **Parameters:**

wBase

[Input] I/O port base addresses, please refer to PISO813_GetConfigAddressSpace().

wBuf[]

[Output] Starting address of the data buffer The user must allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. The user cans analyze these data from the buffer after calling this function.

wCount

[Input] Number of A/D conversions will be performed.

➤ **Returns:**

PISO813_NoError	Operation is OK
PISO813_TimeOutError	A/D converter error (0xffff)

PISO813_Ads_Float

This subroutine will perform a number of A/D conversions by polling. This subroutine is very similar to PISO813_AD_Float except that this subroutine will perform *wCount* of conversions instead of just one conversion. Then the A/D data are stored in a data buffer in Float format. The **fBuf** is the starting address of this data buffer. Refer to PISO813_SetChGain().

➤ **Syntax:**

WORD PISO813_Ads_Float (DWORD **wBase**, WORD **wJump20v**, WORD **wBipolar**,
float **fBuf[]**, DWORD **wCount**);

➤ **Parameters:**

wBase

[Input] I/O port base addresses, please refer to PISO813_GetConfigAddressSpace().

wJump20v

[Input] 1: 20 V (HW default);
0: 10 V

wBipolar

[Input] 1: Bipolar (HW default)
0: Unipolar

wBuf[]

[Output] Starting address of the data buffer (**in float format**). The user must allocate spaces for this buffer and send the address into the function. This function will fill the data into this buffer. The user can analyze these data from the buffer after calling this function.

wCount

[Input] Number of A/D conversions will be performed.

➤ **Returns:**












PISO813_NoError	Operation is OK
PISO813_TimeOutError	A/D converter error (0xffff)

3. Demo Programs

3.1 For Microsoft Windows

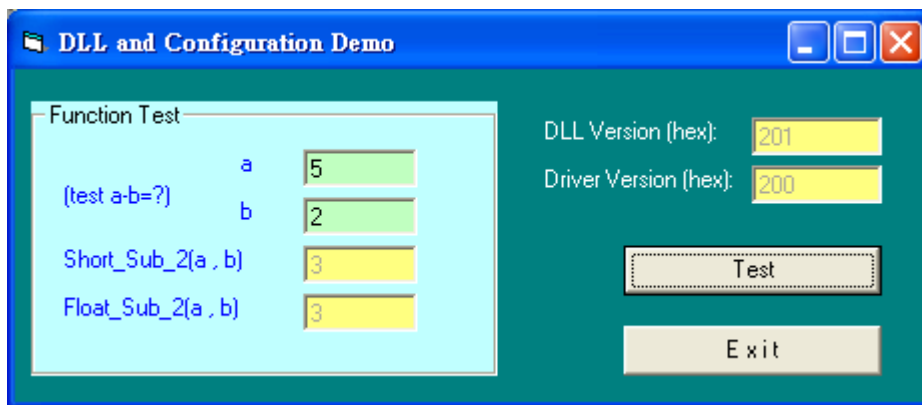
ICP DAS PISO-813 Series Classic Driver DLL contains a set of functions. It can be used in various application programs for PISO-813 series card. The API functions supports many development environments and programming languages, including Microsoft Visual C++ , Visual Basic , Borland Delphi , Borland C Builder++ , Microsoft Visual C#.NET , Microsoft Visual VB.NET.

The demo programs of Windows OS for the PISO-813 series can be found on the supplied CD-ROM, or can be obtained from the ICP DAS FTP web site. The location and addresses are indicated in the table below:

	CD:\NAPDOS\PCI\PISO-813\DLL_OCX\Demo\
	http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/piso-813/dll_ocx/demo/
 BCB4 → for Borland C++ Builder 4 PISO813.H → Header files PISO813.LIB → Linkage library for BCB only	 Delphi4 → for Delphi 4 PISO813.PAS → Declaration files
 VC6 → for Visual C++ 6 PISO813.H → Header files PISO813.LIB → Linkage library for VC only	 VB6 → for Visual Basic 6 PISO813.BAS → Declaration files
 VB.NET2005 → for VB.NET2005 PISO813.vb → Visual Basic Source files	 CSharp2005 → for C#.NET2005 PISO813.cs → Visual C# Source files
The list of demo programs:	
 Config Demo: Get cards information	
 AD Float Demo: Get the AD floating value	
 AD Hex Demo: Get the AD Hex value	

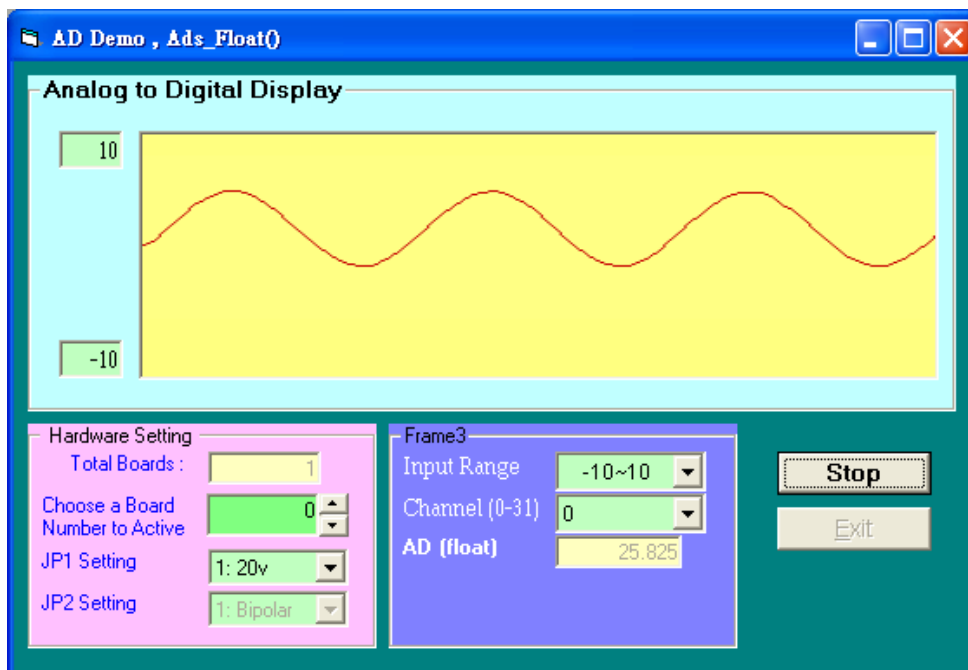
Config Demo: Get cards information

Following figure is the result for the demo program. It can be applied to obtain the hardware information and function test of the PISO-813 board.



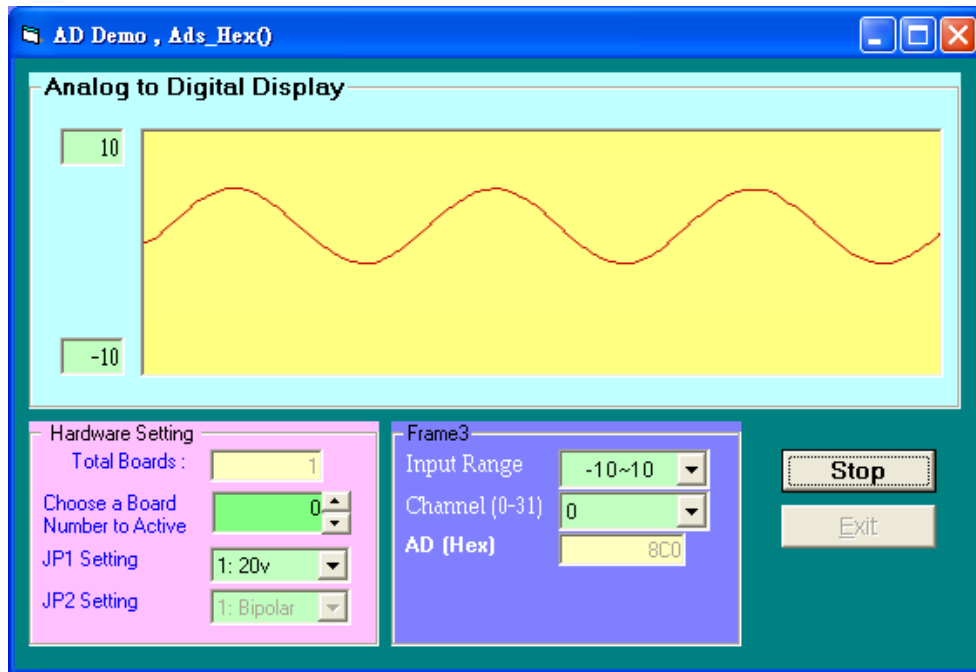
AD Float Demo: Get the AD floating value

This demo program provides the get the AD floating value.



AD Hex Demo: Get the AD Hex value

This demo program provides the get the AD Hex value.



3.2 For DOS

The demo program is contained in:



CD:\NAPDOS\PCI\PISO-813\DOS\PISO813\



<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/piso-813/dos/piso-813/>

- ⊕ \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 File
- ⊕ \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

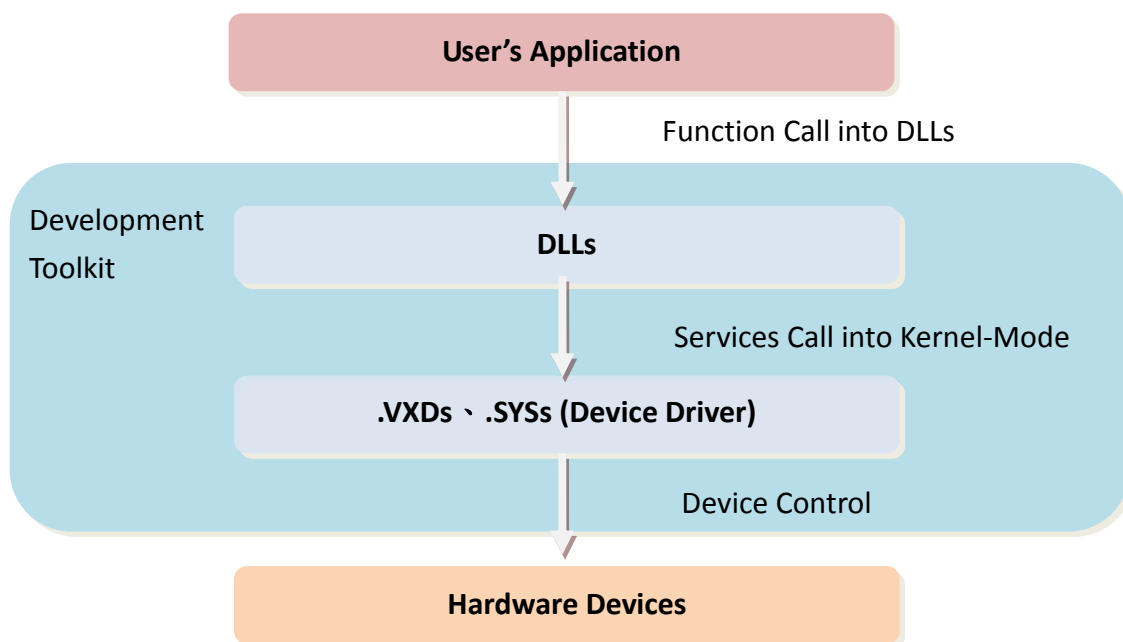
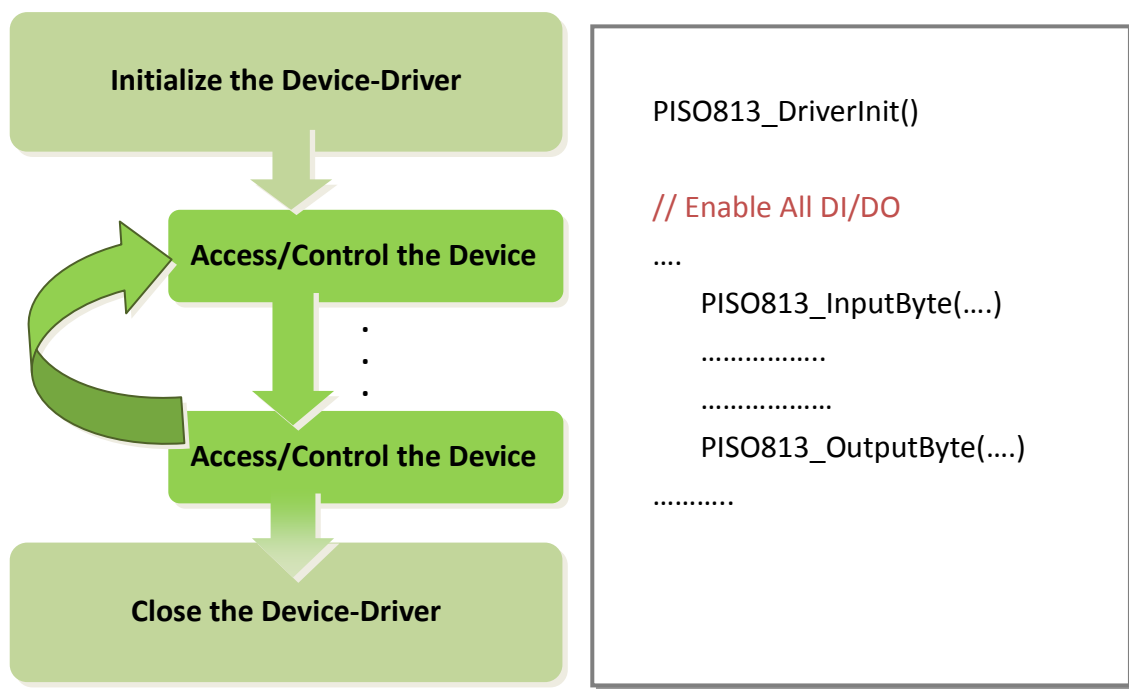
- ⊕ \BC\LIB\Large\PIO.H → BC Declaration File
- ⊕ \BC\LIB\Large\BCPIO_L.LIB → BC Large Model Library File
- ⊕ \BC\LIB\Huge\PIO.H → BC Declaration File
- ⊕ \BC\LIB\Huge\BCPIO_H.LIB → BC Huge Model Library File

The list of demo programs:

- ⊕ Dome1: Measure 32-channel A/I. Bipolar range: -10 V to +10V

Note that all of the hardware control functions need to be provided and processed by user themselves.

4. Programs Architecture



5. Problems Report

Technical support is available at no charge as described below. The best way to report problems is to send electronic mail to Service@icpdas.com or Service.icpdas@gmail.com on the Internet.

When reporting problems, please include the following information:

1. Is the problem reproducible? If so, how?
2. What kind and version of **platform** that you using? For example, Windows 98, Windows 2000 or 32-bit Windows XP/2003/Vista/2008/7.
3. What kinds of our **products** that you using? Please see the product's manual.
4. If a dialog box with an **error message** was displayed, please include the full text of the dialog box, including the text in the title bar.
5. If the problem involves **other programs** or **hardware devices**, what devices or version of the failing programs that you using?
6. **Other comments** relative to this problem or **any suggestions** will be welcomed.

After we had received your comments, we will take about two business days to test the problems that you said. And then reply as soon as possible to you. Please check that if we had received you comments? And please keeps contact with us.