I-8026W User Manual

Multifunction I/O Module

Version 1.0.0, July 2013

Written by Hans Chen

Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, beginning from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for any damage resulting from 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

Trademarks

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

Contact Us

If you have any problems, please feel free to contact us.

You can count on us for quick response.

Email: service@icpdas.com

Table of Contents

Preface 5

1. Introduction 6

1.1. Specifications 7

1.2. Pin Assignments 11

1.3. Wire Connections 1-12

1.4. Internal I/O Structure 1-13

1.5. Dimensions 1-14

1.6. Location of the Demo Programs 1-15

2. Quick Start 2-17

2.1. MiniOS7-based Controllers 2-18

2.2. Windows-based Controllers 2-21

3. APIs 3-23

3.1. i8026W_Init 3-25

3.2. i8026W_GetFirmwareVer 3-27

3.3. i8026W_GetLibVersion 3-28

3.4. i8026W_GetLibDate 3-29

3.5. i8026W_ReadAOGainOffset 3-30

3.6. i8026W_WriteAO 3-32

3.7. i8026W_WriteAOHex 3-34

3.8. i8026W_ReadbackAO 3-36

3.9. i8026W_ReadbackAOHex 3-37

3.10. i8026W_WriteDO 3-38

3.11. i8026W_WriteDOBit 3-40

3.12. i8026W_ReadDIO 3-41

3.13. i8026W_ReadAI 3-43

3.14. i8026W_ReadAIHex 3-45

3.15. i8026W_ReadAIGainOffset 3-47

4. Calibration 4-49

4.1. Introduction 4-49

4.2. Calibrating the I-8026W on i-8000 and iPAC-8000 units 4-50

4.3. Verifying the Calibration 4-57

4.4. Restoring the Default Calibration Settings 4-58

4.5. Calibrating the I-8026W AI on WinCE and WES Units 4-59

4.6. Verifying the Calibration Parameters 4-64

5. Troubleshooting 5-65

5.1. Verifying the AI functions on a WinCE or WES device 5-66

5.2. Verifying the Gain and Offset Values 5-68

5.3. Service Request Requirements 5-69

5.4. What to do when the data read from the I-8026W seems unstable 5-70

Appendix A. Error Codes 5-71

Appendix B. Performance for Read AI Functions 5-72

Appendix C. Revision Information 5-72

Preface

The I-8026W is a multifunction I/O module that provides 6 Analog Input channels, 2 Analog Output channels, 2 Digital Input channels, 2 Digital Output channels.

The information contained in this manual is divided into the following topics:

- Chapter 1, “Introduction” – This chapter provides information related to the hardware, such as the specifications, the jumper settings details and wiring guidelines.
- Chapter 2, “Quick Start” – This chapter provides information on how to get started, including an overview of the location of the demo programs, a “Getting Started Guide”, and an outline of the calibration process.
- Chapter 3, “API Functions” – This chapter describes the functions provided in the I-8026W library, together with an explanation of the differences in the naming rules used for the MiniOS7 and the Windows platforms.
- Chapter 4, “Calibration” – This chapter provides details of how to use the calibration program to calibrate the I-8026W module.
- Chapter 5, “Troubleshooting” – This chapter provides some troubleshooting techniques should you encounter any problems while operating the I-8026W module.

1. Introduction

The I-8026W is a multifunction module that provides 6 Analog Input channels, 2 Analog Output channels, 2 Digital Input channels, and 2 Digital Output channels. It also allows a programmable input range on all Analog Input channels (±10 V, ±5 V, ±2.5 V, ±1.25 V, and ±20 mA), while Analog Output channels are 12 bit at either ±10 V, ±5 V, 0 to 10 V, 0 to 5 V, or 0 to 20 mA. Each Analog Input channel can be configured for an individual range, and a high overvoltage protection of 240 Vrms is also provided. Voltage and current inputs/outputs are jumper selectable.

Applications:

- Industrial Automation
- Industrial Machinery
- Building Automation
- Food and Beverage Systems
- Semiconductor Fabrication
- Control Systems

1.1. Specifications

Analog Input	
Input Channels	6
Input Type	±10 V, ±5 V, ±2.5 V, ±1.25 V, ±20 mA (Jumper Selectable)
Resolution	12-bit
Accuracy	0.2% of FSR for +/- 1 LSB
Overvoltage Protection	240 Vrms
Input Impedance	>2 MΩ
Sampling Rate	Max to 9k Samples/sec (detailed refer to below table " Performance for Read AI Functions")
Individual Channel Configuration	Yes
Analog Output	
Output Channels	2
Output Type	±10 V, ±5 V, 0 to 10 V, 0 to 5 V, 0 to 20 mA (Jumper Selectable)
Resolution	12-bit
Accuracy	± 0.2% of FSR
Voltage Output Capability	10 V @ 20 mA
Individual Channel Configuration	Yes
Digital Input	
Output Channels:	2
Type:	Wet Contact (Sink/Source)
On Voltage Level	+10 V to +30 V
Off Voltage Level	+5 V Max.
Digital Output	
Input Channels	2 (Sink/Source)
Type	Isolated Open Collector (Sink)
Max. Load Current	100 mA/channel
Load Voltage	+5 V_{DC} to +30 V_{DC}
LED Indicators/Display	
System LED Indicator	1 LED as Power/Communication Indicator
I/O LED Indicator	4 LEDs as Digital Input & Digital Output

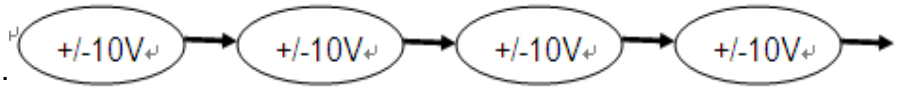
Isolation	
Intra-module Isolation, Field-to-Logic	2500 V_{DC}
EMS Protection	
ESD (IEC 61000-4-2)	±4 kV Contact for Each Terminal
	±8 kV Air for Random Point
Power	
Power Consumption	1.8 W Max.
Environment	
Operating Temperature	-25 to +75°C
Storage Temperature	-30 to +80°C
Humidity	5 to 95% RH, Non-condensing
Mechanical	
Dimensions (W x L x H)	30 mm x 102 mm x 115 mm

Performance for Read AI Functions

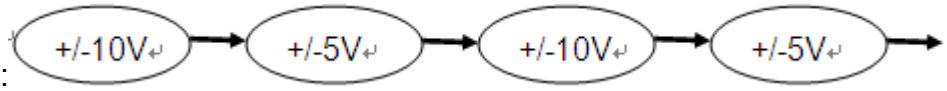
Gain usage will influence the performance for read AI in the same platform.

,

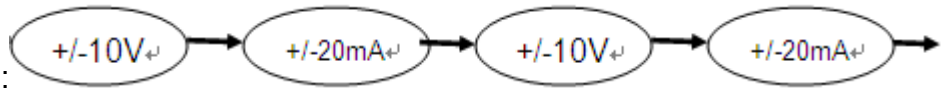
1. All using the same Gain: performance fast

   Gain arrange 1: 

2. Using different Gain, but without switch Ref. Voltage: performance normal

   Gain arrange : 

3. Using different Gain, but and switch Ref. Voltage: performance slow

   Gain arrange 3: 

The detailed value is as below:

Unit: Samples/sec

Platform	Sample function	*Gain arrange 1 (Fast)*	*Gain arrange 2, (Normal)*	*Gain arrange 3, (Slow)*
WES	ReadAIHex ReadAI	8.532 K	5.099 K	2.02 K
CE6	ReadAIHex ReadAI	8.032 K	4.672 K	1.95 K
CE5	ReadAIHex ReadAI	9.012 K	5.43 K	1.853 K
iP-8000	ReadAIHex	6.329 K	4.141 K	2.176 K
	ReadAI	4.605 K	3.386 K	1.932 K
i-8000	ReadAIHex	3.401 K	2.382 K	1.318 K
	ReadAI	1.999 K	1.572 K	1.03 K

Note 1: Using the same gain for all used channels.

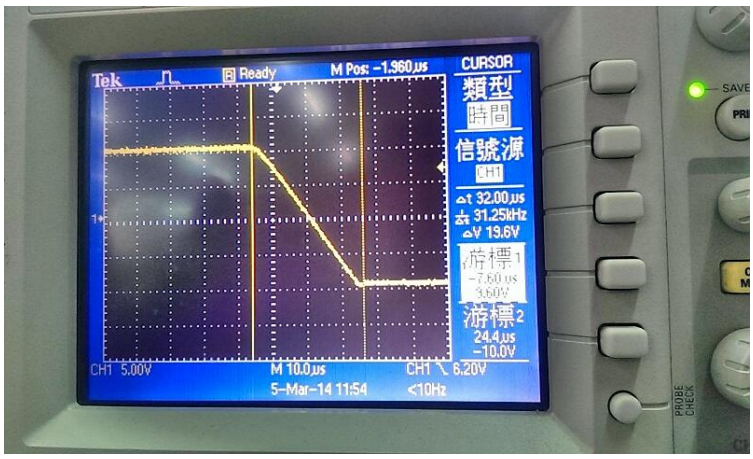
Note 2: Using different gain but the same Ref. voltage, for example1: ch0 uses gain 0, ch1 uses gain 1 and ch2 use gain 0. Example2: ch0 uses gain 2, ch1 uses gain 3, ch2 use gain 4.

Note 3: Using different gain and different Ref. voltage: for example1: ch0 uses gain 0, ch1 uses gain 2. Example2: ch0 uses gain 4, ch1 uses gain 1.

Note 4: In I-8026W module, gain 0~1 use one Ref voltage, gain 2~4 use another Ref voltage. When user uses gain for different Ref. voltage, it will waste some time to switch it, and the performance will be decrease.

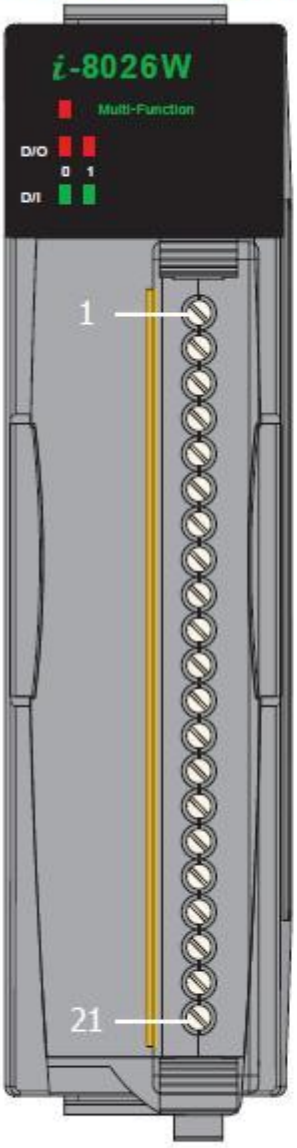
Performance for Write AO Functions

We can know the AO from 0 V to -10 V will need 32 us as below picture, it is about 31.25 k.



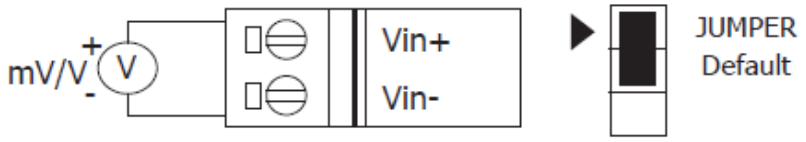
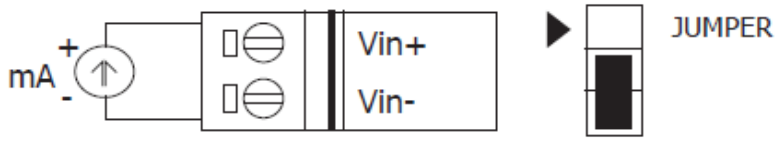
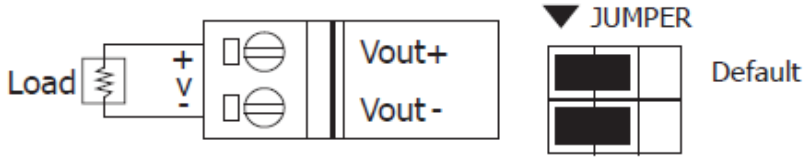
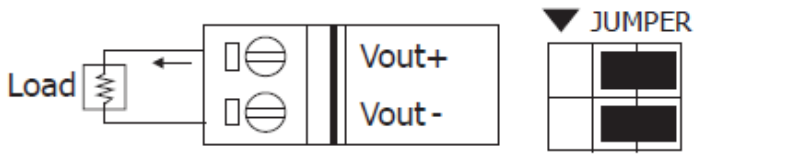
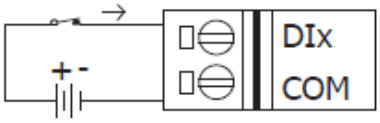
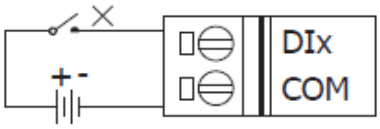
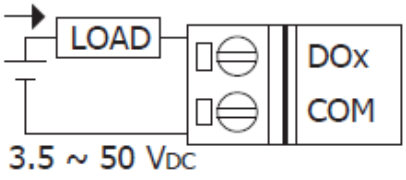
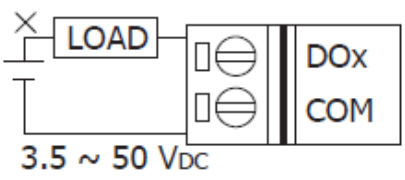
1.2. Pin Assignments

Pin Assignments

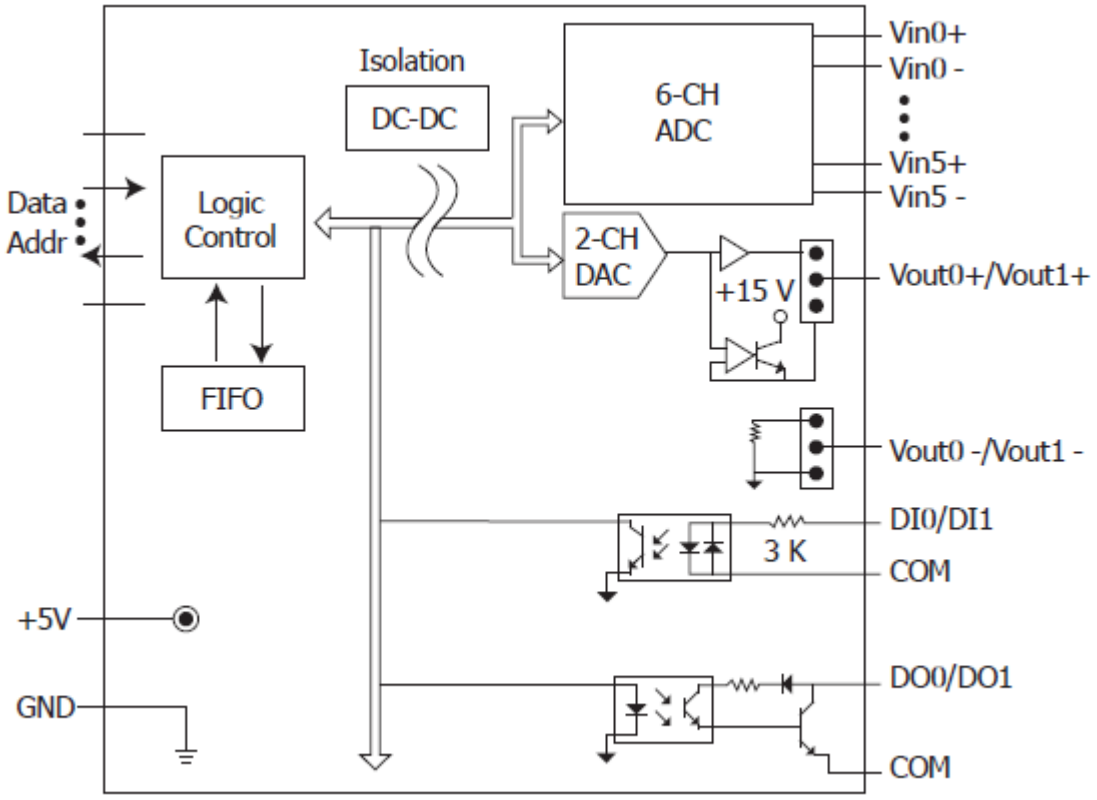


Terminal No.	Pin Assignment
01	Vin0+
02	Vin0 -
03	Vin1+
04	Vin1 -
05	Vin2+
06	Vin2 -
07	Vin3+
08	Vin3 -
09	Vin4+
10	Vin4 -
11	Vin5+
12	Vin5-
13	Vout0+
14	Vout0 -
15	Vout1+
16	Vout1 -
17	DO0
18	DO1
19	DI0
20	DI1
21	COM

1.3. Wire Connections

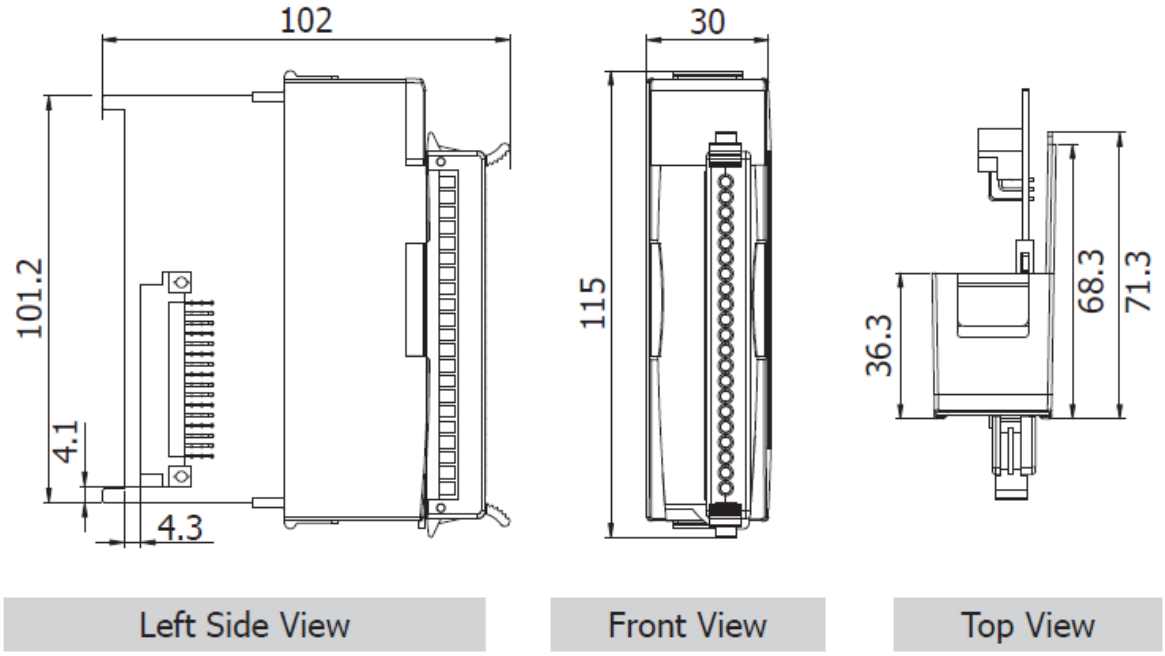
<table>
<tr><th colspan="3">Voltage Input</th></tr>
<tr><td colspan="3"> 
</td></tr>
<tr><th colspan="3">Current Input</th></tr>
<tr><td colspan="3"> 
</td></tr>
<tr><th colspan="3">Voltage Output</th></tr>
<tr><td colspan="3"> 
</td></tr>
<tr><th colspan="3">Current Output</th></tr>
<tr><td colspan="3"> 
</td></tr>
<tr><th>Digital Input/
Counter</th><th>ON State
Readback as 1</th><th>OFF State
Readback as 0</th></tr>
<tr><td>Wet Contact
(Sink)</td><td> 
</td><td> 
</td></tr>
<tr><th>Digital Output</th><th>ON State
Readback as 1</th><th>OFF State
Readback as 0</th></tr>
<tr><td>Open Collector
(Sink)</td><td> 
</td><td> 
</td></tr>
</table>

1.4. Internal I/O Structure



1.5. Dimensions

Units: mm



1.6. Location of the Demo Programs

ICP DAS provides a range of demo programs for different platforms that can be used to verify the functions of the I-8026W. The source code contained in these programs can also be reused in your own custom programs if needed. The following is a list of the locations where both the demo programs and associated libraries can be found on either the ICP DAS web site or the enclosed CD.

Platform	Location
For the I-8000 on the Web	
Library	ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/8000/841x881x/demo/lib/
Demo	ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/8000/841x881x/demo/io_in_slot/
For the I-8000 on the CD	
Library	CD:\Napdos\8000\841x881x\demo\Lib
Demo	CD:\Napdos\8000\841x881x\demo\IO_in_Slot
For the iPAC-8000 on the Web	
Library	ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/ipac8000/demo/basic/ip-84x1_ip-88x1/lib/
Demo	ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/ipac8000/demo/basic/ip-84x1_ip-88x1/io_in_slot/
For the iPAC-8000 on the CD	
Library	CD:\Napdos\iPAC8000\Demo\Basic\iP-84x1_iP-88x1\Lib
Demo	CD:\Napdos\iPAC8000\Demo\Basic\iP-84x1_iP-88x1\IO_in_Slot
For the Windows CE5 Platform on the Web	
Library	ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/io_modules/
Demo	ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/pac_io/local/ (eVC demo) ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c%23.net/pac_io/local/ (C# demo)

Platform	Location
For the Windows CE5 Platform on the CD	
Library	CD:\napdos\wp-8x4x_ce50\sdk\IO_Modules
Demo (eVC & C#)	CD:\napdos\wp-8x4x_ce50\Demo\WinPAC\eVC\PAC_IO\Local CD:\napdos\wp-8x4x_ce50\Demo\WinPAC\DOTNET\C#.NET\PAC_IO\Local
For the Windows CE6 Platform on the Web	
XP-8000-CE6	ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/sdk/special_io/ ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/demo/xpac/vc2008/io/local/ ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/demo/xpac/c%23/io/local/
XP-8000-Atom-CE6	ftp://ftp.icpdas.com/pub/cd/xpac-atom-ce6/sdk/special_io/ ftp://ftp.icpdas.com/pub/cd/xpac-atom-ce6/demo/xpac/vc2008/io/local/ ftp://ftp.icpdas.com/pub/cd/xpac-atom-ce6/demo/xpac/c%23/io/local/
For the Windows CE6 Platform on the CD	
XP-8000-CE6	CD:\SDK\Special_IO CD:\Demo\XPAC\VC2008\IO\Local CD:\Demo\XPAC\C#\IO\Local
XP-8000-Atom-CE6	CD:\SDK\Special_IO CD:\Demo\XPAC\VC2008\IO\Local CD:\Demo\XPAC\C#\IO\Local
For the Windows Embedded Standard (WES) Platform on the Web	
XP-8000	ftp://ftp.icpdas.com/pub/cd/xp-8000/sdk/io/ ftp://ftp.icpdas.com/pub/cd/xp-8000/demo/specialized_io/
XP-8000-Atom	ftp://ftp.icpdas.com/pub/cd/xpac-atom/sdk/io/ ftp://ftp.icpdas.com/pub/cd/xpac-atom/demo/specialized_io/
For the Windows Embedded Standard (WES) Platform on the CD	
XP-8000	CD:\SDK\IO CD:\Demo\Specialized_IO
XP-8000-Atom	CD:\SDK\IO CD:\Demo\Specialized_IO

2. Quick Start

This section provides a “Getting Started Guide” and details of the calibration process when using the I-8026W module on either the MiniOS7 or Windows platforms.

This section contains a “Getting Started Guide” and details of the calibration process when using the I-8026W module:

- MiniOS7-based Controllers (i-8000 and iPAC-8000 PAC)
- WinCE- and WES-based Controllers (WinPAC and XPAC PAC)
- Demo develop produce

2.1. MiniOS7-based Controllers

Getting Started Guide

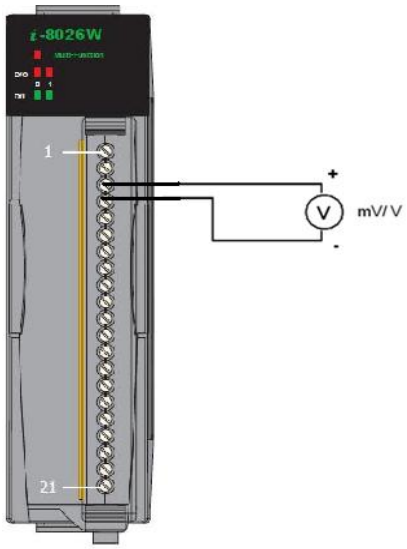
The 26demo.exe executable file, which is located in the 8026demo folder of the I-8026W demo programs, can be used to retrieve the basic configuration information related to the I-8026W module, and to verify the Analog Input read functions. The basic configuration information includes:

- The version number and the published date of the library.
- The FPGA version information
- The gain and offset values for each input range and each channel
- The data read from each channel

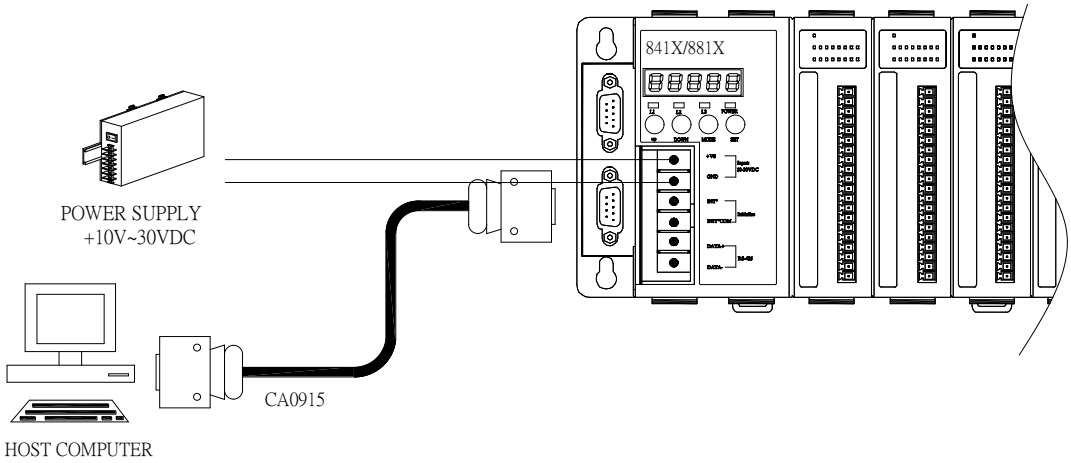
(See Location of the Demo Programs information in Section 1.6 for details of where to find the 26demo.exe file in the I-8026W demo programs folder)

Step 1. Refer to Section Wire Connections and ensure that the voltage/current jumper is in the correct position.

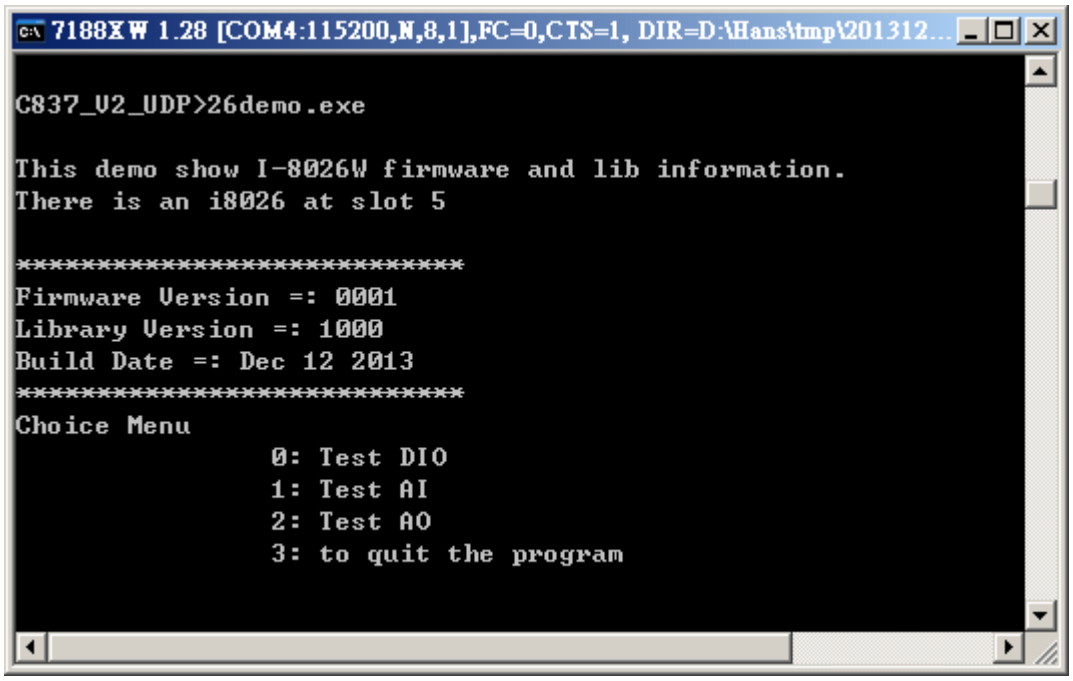
Step 2. Connect a stable signal source (e.g., a battery output) to either the Analog Input or the Analog Output of the I-8026W module, as illustrated below.



Step 3. Connect the power supply to the module, and then connect the control unit to the Host PC using an RS-232 cable.



Step 4. Launch the 26demo.exe executable file on the Host PC, and then verify that the basic configuration information and the I/O data from each channel is correct.

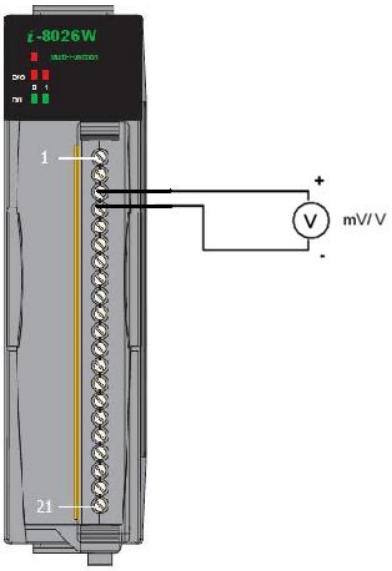


2.2. Windows-based Controllers

Getting Started Guide

The **pac_i8026Wdemo.exe** executable file, which is located in the pac_i8026WDemo folder of the I-8026W demo programs, can be used to retrieve the basic configuration information related to the I-8026W, and to verify the Analog Input read functions. The basic configuration information includes:

- The version number and the published date of the library.
- The FPGA version information
- The gain and offset values for each input range



- The data read on each channel

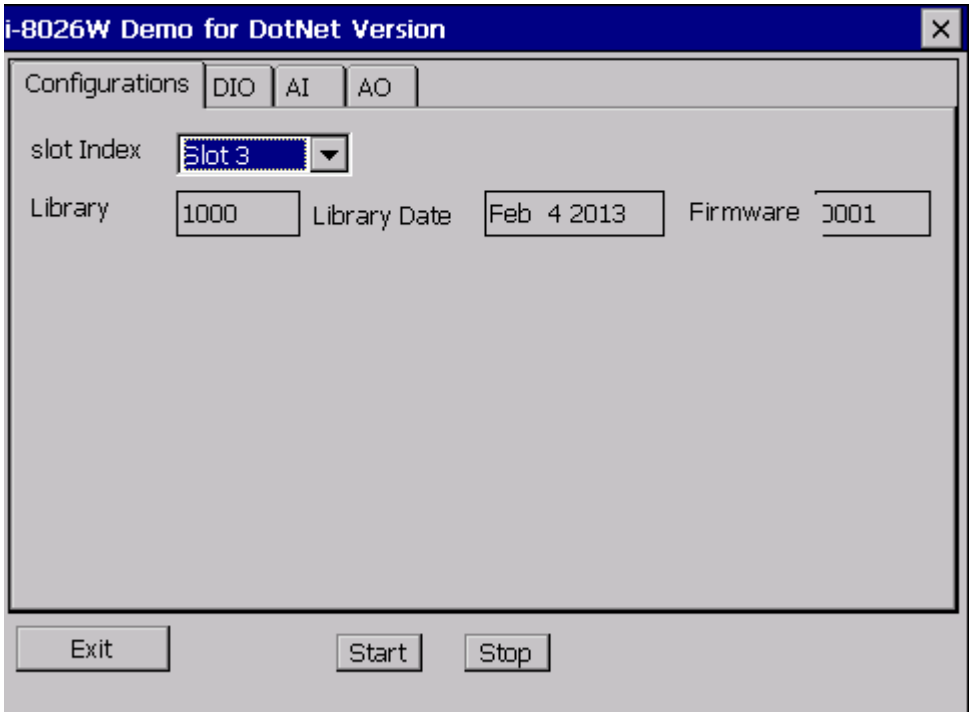
(See the Location of the Demo Programs information in Section 1.6. for details of where to find the **pac_i8026Wdemo.exe** file in the I-8026W demo programs folder)

Step 1. Refer to Section Wire Connections and ensure that the voltage/current jumper is in the correct position

Step 2. Connect a stable signal source (e.g., a battery output) to the I-8026W module.

Step 3. Insert the I-8026W into a vacant slot in the control unit and power on the controller.

Step 4. Launch the **pac_i8026Wdemo.exe** executable file on the control unit, and verify that the basic configuration information and I/O data read from each channel is correct.



Tips & Warnings

Unused channels should be connected to GND to avoid floating.

3. APIs

ICP DAS provides APIs, libraries and demo programs, including the source code, that allow integration of the I-8026W into the platforms indicated in the table below. The APIs and programming procedures are similar on both the MiniOS7 and Windows platforms, with the only difference being the **prefix characters** added to the name of the function in the library (APIs). For functions applicable to the MiniOS7 and Linux platforms, the prefix “i8026W_” is added to the function name, and the prefix “pac_i8026W_” is added to functions applicable to the Windows platform.

In this document, the function relevant to the MiniOS7 platform is used in the examples and as the title of the section for each function.

The following table gives an overview of the relationship between the different platforms and the product series, together with the respective prefix used for the function name.

Platform	Product	API Prefix
Windows CE5 Windows CE6	WP-8000 series WP-2000 series XP-8000-CE6 series	“pac_i8026W_ “+ function name
Windows Embedded Standard (WES)	XP-8000 series	“pac_i8026W_ “+ function name
MiniOS7	I-8000 series iPAC-8000 series VP-2000 series	“i8026W_” + function name
Linux	LinPAC-8000 series	“i8026W_” + function name

Function List

The following is a list of the functions provided in the 8026W.lib for the MiniOS7 platform.

Function	Description
i8026W_Init	Initializes the driver and confirms the hardware ID.
i8026W_GetFirmwareVer	Retrieves the version number for the FPGA firmware for troubleshooting purposes.
i8026W_GetLibVersion	Retrieves the version number of the 8026W.lib file.
i8026W_GetLibDate	Retrieves the release date of the 8026W.lib file.
i8026W_ReadAOGainOffset	Retrieves the Analog Output Gain and Offset values for each output type.
i8026W_WriteAO	Writes an output value to a single specified Analog Output channel in float format.
i8026W_WriteAOHex	Writes a value to a single specified Analog Output channel in hexadecimal format.
i8026W_ReadbackAO	Reads a calibrated Analog Output value from a single specified channel in float format.
i8026W_ReadbackAOHex	Reads a calibrated Analog Output value from a single specified channel in hexadecimal format.
i8026W_WriteDO	Writes the Digital Output value to the I-8026W module.
i8026W_WriteDOBit	Sets a specific Digital Output channel of the I-8026W module to ON or OFF.
i8026W_ReadDIO	Reads the Digital Input and Digital Output value from the I-8026W module.
i8026W_ReadAI	Reads a calibrated Analog Input value from a single specified channel in float format.
i8026W_ReadAIHex	Reads a calibrated Analog Input value from a single specified channel in hexadecimal format.
i8026W_ReadAIGainOffset	Retrieves the Analog Input reference Gain and Offset values for each input type and for each channel.

3.1. i8026W_Init

This function is used to initialize the driver and confirm the hardware ID information.

Prototype

For MiniOS7

short i8026W_Init(int slot);

For Windows (CE and WES using C++)

short pac_i8026W_Init(int slot);

For Windows (CE and WES using C#)

Int16 pac8026W.Init(int slot);

Parameters

slot: specifies the slot number (0 to 7)

Return Values

0 = the module inserted in the slot is an I-8026W.

-1 = there are no I-8026W modules inserted in this slot.

For other return values, see the Error Codes in Appendix A.

Note

Before executing any functions on the I-8026W, the *i8026W_Init* function needs be called once for each I-8026W module inserted in the controller unit. For example, if there are two or more I-8026W modules inserted in the controller, the *i8026W_Init* function must be individually called for each I-8026W module by including the number of the slot where the I-8026W module is inserted.

Example

[C/C++]

```
int slotIndex,err;
err=i8026W_Init(slotIndex);
if(err==0)
{
  Print("There is an I-8026W module in slot %d\n",slotIndex);
}
else
{
  Print("There is no I-8026W module in slot %d\n",slotIndex);
}
```

3.2. i8026W_GetFirmwareVer

This function is used to retrieve the version information for the FPGA firmware. The function is only used for troubleshooting or recording purposes.

Prototype

For MiniOS7

short i8026W_GetFirmwareVer(int slot);

For Windows (CE and WES)

short pac_i8026W_GetFirmwareVer(int slot);

For Windows (CE and WES using C#)

Int16 pac8026W. FirmwareVersion (int slot);

Parameters

slot: specifies the slot number (0 - 7)

Return Values

The version information of the FPGA firmware for the I-8026W module

Example

[C++]

```
short ver=0, slot=0;

ver= i8026W_GetFirmwareVer (slot);

Print( "\nFirmware Version =: %04X",ver );
```

3.3. i8026W_GetLibVersion

This function is used to retrieve the version information for the 8026W library file. The function is only used for troubleshooting or recording purposes.

Prototype

For MiniOS7

short i8026W_GetLibVersion(void);

For Windows (CE and WES C++)

short pac_i8026W_GetLibVersion(void);

For Windows (CE and WES using C#)

Int16 pac8026W.LibVersion (int slot);

Parameters

None

Return Values

The version information for the 8026W.lib file

Example

[C++]

```
short version;
version = i8026W_GetLibVersion();
Print("\nLibrary Version =: %04X",i8026W_GetLibVersion());
```

3.4. i8026W_GetLibDate

This function is used to retrieve the release (build) date of the 8026W.lib file.

Prototype

For MiniOS7

void i8026W_GetLibDate(char *LibDate);

For Windows (CE and WES C++)

void pac_i8026W_GetLibDate(char libDate[]);

For Windows (CE and WES using C#)

string pac8026W. LibDate ();

Parameters

*LibDate: **[Output]** the release (build) date of the 8026W.lib file

Return Values

None

Example

[C++]

```
char libDate [32];

i8026W_GetLibDate(libDate);
Print("\nBuild Date =: %s",libDate);
```

3.5. i8026W_ReadAOGainOffset

This function is used to read the gain and offset values for each output type set for a specified Analog Output channel.

Prototype

For MiniOS7

```
void i8026W_ReadAOGainOffset
(
int slot, int ch, int gain, unsigned short* gainValue, short* offsetValue
);
```

For Windows (CE and WES C++)

```
void pac_i8026W_ReadAOGainOffset
(
int slot, int ch, short gain, unsigned short* gainValue, short* offsetValue
);
```

For Windows (CE and WES C#)

```
void pac8026W.ReadAOGainOffset
(
int slot, int ch, Int16 gain, ref UInt16 gainValue, ref Int16 offsetValue
);
```

Parameters

slot: specifies the slot number (0 - 7)

ch: specifies the Analog Output channel number (0 - 1)

gain: specifies the input type (0 - 4), where:

0: +/-10 V, **1:** +/-5 V, **2:** +/-2.5 V, **3:** +/-1.25 V, **4:** +/-20 mA

*gainValue: **[Output]** the gain value for the Analog Output range

*offsetValue: **[Output]** the offset value for the Analog Output range

Return Values

None

Example

[C++]

```
unsigned short gVal=0;
short oVal=0;
for(ch=0;ch<2;ch++)
{
  i8026W_ReadGainOffset(slot,ch,gain,&gVal,&oVal);
  Print("\nThe Gain and Offset values for the Calibration are:
  Gain=%u; Offset=%d",gVal,oVal);
}
```

3.6. i8026W_WriteAO

This function is used to write the output value to a single specified Analog Output channel in floating point format.

Prototype

For MiniOS7

short i8026W_WriteAO(int slot, int ch, short gain, float fData);

For Windows (CE and WES C++)

short pac_i8026W_WriteAO(int slot, int ch, short gain, float fData);

For Windows (CE and WES using C#)

Int16 pac8026W.WriteAO(int slot, int ch, Int16 gain, float fData);

Parameters

slot: specifies the slot number (0 - 7)

ch: specifies the Analog Output channel number (0 - 2)

gain: specifies the input type (0 - 4), where:

0: +/-10 V, **1:** +/-5 V, **2:** +/-2.5 V, **3:** +/-1.25 V, **4:** +/-20 mA

fData: the Analog Output data in floating point format

Return Values

0 = No Error

For other return values, see the Error Codes in Appendix A.

Example

[C++]

```
int slot,ch,gain;
float fVal=0.0;

slot = 0;
gain = 0; // "+/-10 V"
for(ch=0;ch<2;ch++)
{
  fVal = 5.0;
  i8026W_WriteAO ( slot, ch, gain, fVal);
  Print("\n[%02d]= [ %05.4f ]",ch,,fVal);
}
```

3.7. i8026W_WriteAOHex

This function is used to write the output value to a single specified Analog Output channel in hexadecimal format.

Prototype

For MiniOS7

short i8026W_WriteAOHex(int slot, int ch, short gain, short hData);

For Windows (CE and WES C++)

short pac_i8026W_WriteAOHex(int slot, int ch, short gain, short hData);

For Windows (CE and WES using C#)

Int16 pac8026W.WriteAOHex(int slot, int ch, Int16 gain, short hData);

Parameters

slot: specifies the slot number (0 - 7)

ch: specifies the Analog Output channel number (0 - 2)

gain: specifies the input type (0 - 4), where:

0: +/-10 V, **1:** +/-5 V, **2:** +/-2.5 V, **3:** +/-1.25 V, **4:** +/-20 mA

hData: the Analog Output data in hexadecimal format

Return Values

0 = No Error

For other return values, see the Error Codes in Appendix A.

Example

[C++]

```
int slot,ch,gain;
short hVal=0;

slot = 0;
gain = 0; // "+/-10 V"
for(ch=0;ch<2;ch++)
{
  hVal =0x3fff;
  i8026W_WriteAOHex( slot, ch, gain, hVal);
}
```

3.8. i8026W_ReadbackAO

This function is used to read the calibrated output value from a single specified Analog Output channel floating point.

Prototype

For MiniOS7

short float i8026W_ReadbackAO(int slot, int ch, float* fVal);

For Windows (CE and WES C++)

short pac_i8026W_ReadbackAO(int slot, int ch, float* fVal);

For Windows (CE and WES using C#)

Int16 pac8026W.ReadbackAO(int slot, int ch, ref float fVal);

Parameters

slot: specifies the slot number (0 - 7)

ch: specifies the Analog Output channel number (0 - 2)

*fVal: **[Output]** the value read from memory that is written to the module

Return Values

0 = No Error

For other return values, see the Error Codes in Appendix A.

Example

[C++]

```
int slot,ch;
float fVal=0.0;

slot = 0;
for(ch=0;ch<6;ch++)
{
  i8026W_ReadbackAO( slot, ch, &fVal);
  Print("\n[%02d]= [ %05.4f ]",ch,,fVal);
}
```

3.9. i8026W_ReadbackAOHex

This function is used to read the calibrated output value from a single specified Analog Output channel in hexadecimal format.

Prototype

For MiniOS7

short i8026W_ReadbackAOHex(int slot, int ch, short* hVal);

For Windows (CE and WES C++)

short pac_i8026W_ReadbackAOHex(int slot, int ch, short* hVal);

For Windows (CE and WES using C#)

Int16 pac8026W.ReadbackAOHex(int slot, int ch, ref Int16 hVal);

Parameters

slot: specifies the slot number (0 - 7)

ch: specifies the Analog Output channel number (0 - 2)

*hVal: **[Output]** the value read from memory that written to the module

Return Values

0 = No Error

For other return values, see the Error Codes in Appendix A.

Example

[C++]

```
int slot,ch;
short hVal=0;

slot = 0;
for(ch=0;ch<2;ch++)
{
  fVal = i8026W_ReadbackAOHex (slot, ch);
  Print("\n[%02d]= [ %04X ]",ch,,hVal);
}
```

3.10. i8026W_WriteDO

This function is used to write the Digital Output value to the i-8026W module.

Prototype

For MiniOS7

short i8026W_WriteDO(int slot, short hData);

For Windows (CE and WES C++)

short pac_i8026W_WriteDO(int slot, Int16 hData);

For Windows (CE and WES using C#)

Int16 pac8026W.WriteDO(int slot, Int16 hData);

Parameters

slot: specifies the slot number (0 - 7)

hData: the Digital Output value (0 - 3), as per the table below

Output Value	CH0	CH1
0	OFF	OFF
1	ON	OFF
2	OFF	ON
3	ON	ON

Return Values

0 = No Error

For other return values, see the Error Codes in Appendix A.

Example

[C++]

```
int slot,ch,gain;
short hVal=3;

slot = 0;
  i8026W_WriteDO (slot, hVal);
```

3.11. i8026W_WriteDOBit

This function is used to set a specific Digital Output channel on the i-8026W module ON or OFF.

Prototype

For MiniOS7

short i8026W_WriteDOBit(int slot, int ch, int bitStatus);

For Windows (CE and WES C++)

short pac_i8026W_WriteDOBit(int slot, int ch, int bitStatus);

For Windows (CE and WES using C#)

Int16 pac8026W.WriteDOBit(int slot, int ch, int bitStatus);

Parameters

slot: specifies the slot number (0 - 7)

ch: specifies the Digital Output channel number (0 - 1)

bitVal**:** specifies the status of the digital output, where:

0: OFF

1: ON

Return Values

0 = No Error

For other return values, see the Error Codes in Appendix A.

Example

[C++]

```
int slot,ch, bitVal;

slot = 0;
ch = 0;
bitVal =1;
  i8026W_WriteDOBit (slot,ch, bitVal);
```

3.12. i8026W_ReadDIO

This function is used to read the Digital Input and Digital Output values from the i-8026W module.

Prototype

For MiniOS7

short i8026W_ReadDIO

(

int slot, short* diVal, short* doVal,unsigned char diBitArr[], unsigned char doBitArr[]

);

For Windows (CE and WES C++)

short pac_i8026W_ReadDIO

(

int slot, short* diVal, short* doVal,unsigned char diBitArr[], unsigned char doBitArr[]

);

For Windows (CE and WES using C#)

Int16 pac8026W.ReadDIO

(

int slot, ref Int16 diVal, ref Int16 doVal,byte[] diBitArr, byte[] doBitArr

);

Parameters

slot: specifies the slot number (0 - 7)

*diVal: **[Output]** the Digital Input data

*doVal: **[Output]** the Digital Output data

diBitArr: **[Output]** the bit status of the Digital Input data

doBitArr: **[Output]** the bit status of the Digital Output data

Return Values

0 = No Error

For other return values, see the Error Codes in Appendix A.

Example

[C++]

```
int slot;
short diVal=0, doVal=0;
unsigned char diBitArr[2], doBitArr[2];
slot = 0;

  i8026W_ReadDIO( slot, &diVal,&doVal, diBitArr,doBitArr);
  Print("\n DI=[%02X]; DO=[ %02X ]", diVal ,doVal);
```

3.13. i8026W_ReadAI

This function is used to read the calibrated input value from a single specified Analog Input channel in floating point format.

Prototype

For MiniOS7

short float i8026W_ReadAI(int slot, int ch, short gain, float* fVal);

For Windows (CE and WES C++)

short pac_i8026W_ReadAI(int slot, int ch, short gain, float* fVal);

For Windows (CE and WES using C#)

Int16 pac8026W.ReadAI(int slot, int ch, Int16 gain, ref float fVal);

Parameters

slot: specifies the slot number (0 - 7)

ch: specifies the Analog Input channel number (0 ~ 5)

gain: specifies the input type (0 - 4), where:

0: +/-10 V, **1:** +/-5 V, **2:** +/-2.5 V, **3:** +/-1.25 V, **4:** +/-20 mA

fVal: **[Output]** the input data in float format

Return Values

0 = No Error

For other return values, see the Error Codes in Appendix A.

Example

[C++]

```
int slot,ch,gain;
float fVal=0.0;

slot = 0;
gain = 0; // "+/-10 V"
for(ch=0;ch<6;ch++)
{
  i8026W_ReadAI( slot, ch, gain, &fVal);
  Print("\n[%02d]= [ %05.4f ]",ch,,fVal);
}
```

3.14. i8026W_ReadAIHex

This function is used to read the calibrated input value from a single specified Analog Input channel in hexadecimal format.

Prototype

For MiniOS7

short i8026W_ReadAIHex(int slot, int ch, short gain, short* hVal);

For Windows (CE and WES C++)

short pac_i8026W_ReadAIHex(int slot, int ch, short gain, short* hVal);

For Windows (CE and WES using C#)

Int16 pac8026W.ReadAIHex(int slot, int ch, Int16 gain, ref Int16 hVal);

Parameters

slot: specifies the slot number (0 - 7)

ch: specifies the Analog Input channel number (0 - 5)

gain: specifies the input type (0 - 4), where:

0: +/-10 V, **1:** +/-5 V, **2:** +/-2.5 V, **3:** +/-1.25 V, **4:** +/-20 mA

hVal: **[Output]** the input data in hexadecimal format

Return Values

0 = No Error

For other return values, see the Error Codes in Appendix A.

Example

[C++]

```
int slot,ch,gain;
short hVal=0.0;

slot = 0;
gain = 0; // "+/-10 V"
for(ch=0;ch<6;ch++)
{
  i8026W_ReadAIHex( slot, ch, gain,&hVal);
  Print("\n[%02d]= [ %04X ] ",ch,,hVal);
}
```

3.15. i8026W_ReadAIGainOffset

This function is used to read the reference gain and offset values for each Analog Input channel and each input type.

Prototype

For MiniOS7

```
short i8026W_ReadAIGainOffset
(
int slot, int ch, int gain, unsigned short* refGain, short * refOffset
);
```

For Windows (CE and WES C++)

```
short pac_i8026W_ReadAIGainOffset
(
int slot, int ch, short gain, unsigned short* refGain, short * refOffset
);
```

For Windows (CE and WES using C#)

```
Int16 pac8026W_ReadAIGainOffset
(
int slot, int ch, Int16 Gain, ref UInt16 refGain, ref Int16 refOffset
);
```

Parameters

slot: specifies the slot number (0 - 7)

ch: specifies the Analog Input channel number (0 – 5)

gain: specifies the input type (0 - 4), where:

0: +/-10 V, **1:** +/-5 V, **2:** +/-2.5 V, **3:** +/-1.25 V, **4:** +/-20 mA

*refGain: **[Output]** the reference gain value for the Analog Input type

*refOffset: **[Output]** the reference offset value for the Analog Input type

Return Values

0 = No Error

For other return values, see the Error Codes in Appendix A.

Example

[C++]

```
unsigned short gVal=0;
short oVal=0;
i8026W_ ReadAIGainOffset (slot,gain,&gVal, &oVal);
Print("\nThe Gain = %04X ,    Offset = %04X ",gVal, oVal);
```

4. Calibration

4.1. Introduction

Each I-8026W module is factory calibrated and well verified before shipment, so it is usually unnecessary to calibrate the module again, unless the input impedance is changed on a calibrated module, or the accuracy is lost.

In addition to inserting the I-8026W module into a controller slot, the following items are required before attempting to calibrate the I-8026W:

- A single stable calibration source, such as a 3 1/2 digit power supply (or better), or a battery output.
- A single 4 1/2 digit voltage meter (15-bit resolution or better)
- A Calibration Program: See Section 1.6 “Location of the Demo Programs” for details of where to find the demo program included in the I-8026W demo programs folder.

Tips & Warnings

An unstable calibration source will cause calibration errors and may affect the accuracy of the data acquisition.

This section contains:

➤ Calibrating the I-8026W

➤ Verifying the calibration

➤ Restoring the default calibration settings

4.2. Calibrating the I-8026W on i-8000 and iPAC-8000 units

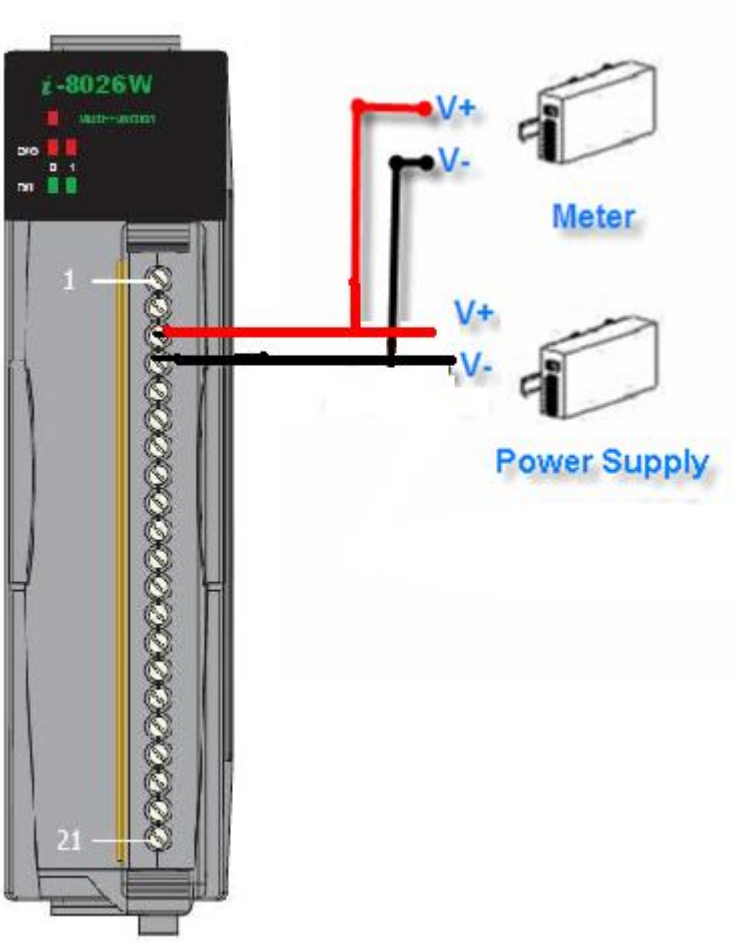
Step 1. Repeat steps 1 to 3 as described in the Quick Start guide in Chapter 2.

a. Attach the power supply to the control unit and then connect the control unit to the Host PC.

b. Connect the calibration source to channel 0 of the I-8026W module using the differential wiring method.

c. Connect the meter, as illustrated in the following figure.

d. Switch on the power to the control unit.



Step 2. Launch the MiniOS7 Utility on the Host PC, then upload the calibration program to the control unit and execute it.

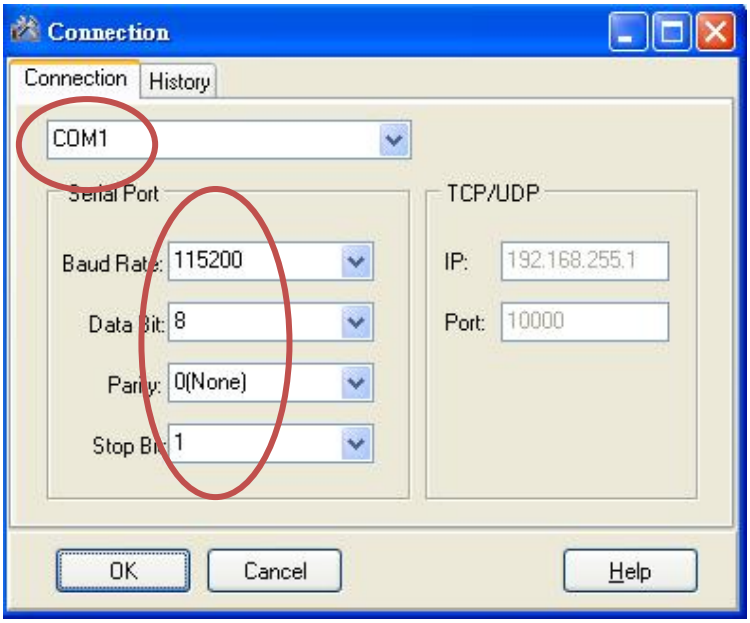
The MiniOS7 Utility can be downloaded from the following web site:
http://www.icpdas.com/download/minios7.htm

Select the appropriate calibration program for your controller.

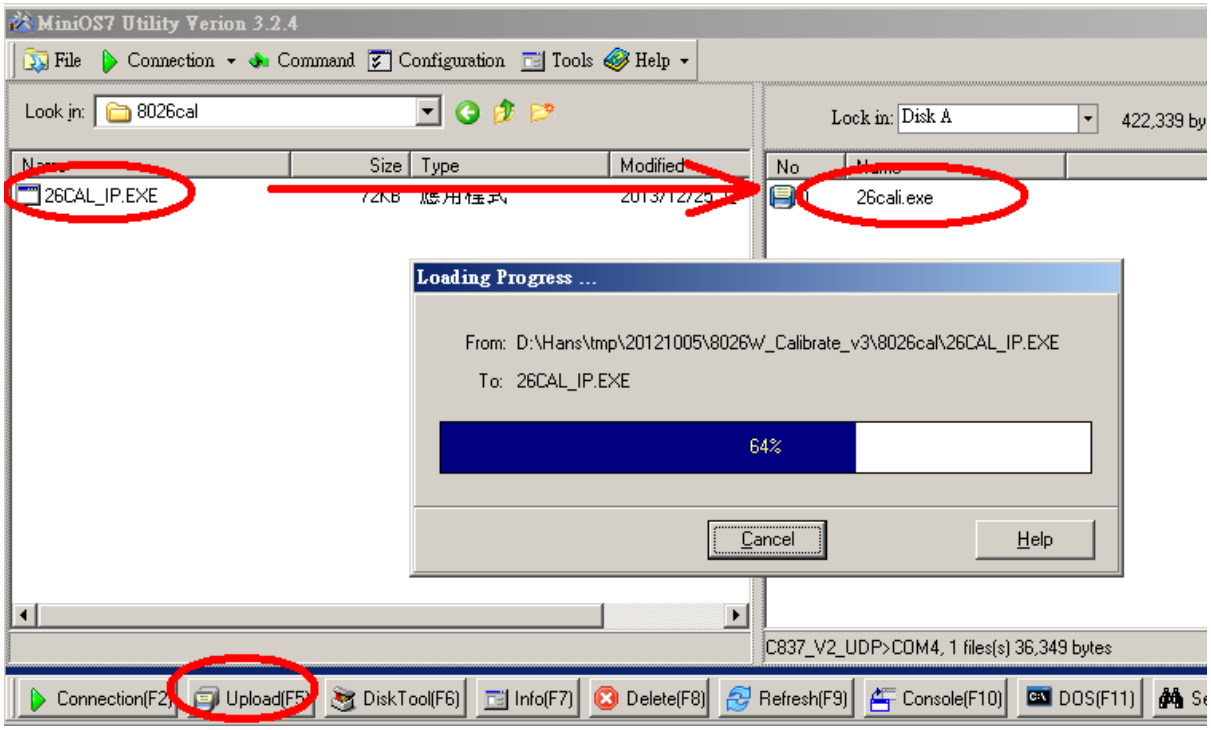
- 8026cal.exe: This is the calibration program for I-8000 units, and is located in the same folder as the demo programs for the I-8026W module. (See Section 1.6. Location of the Demo Programs)
- iP_8026cal.exe: This is the calibration program for iP-8000 units, and is located in the same folder as the demo programs for the I-8026W module. (See Section 1.6. Location of the Demo Programs)

a. Launch the MiniOS7 Utility on the Host PC, and then choose **New Connection** from the **Connection** menu, or press **F2**.

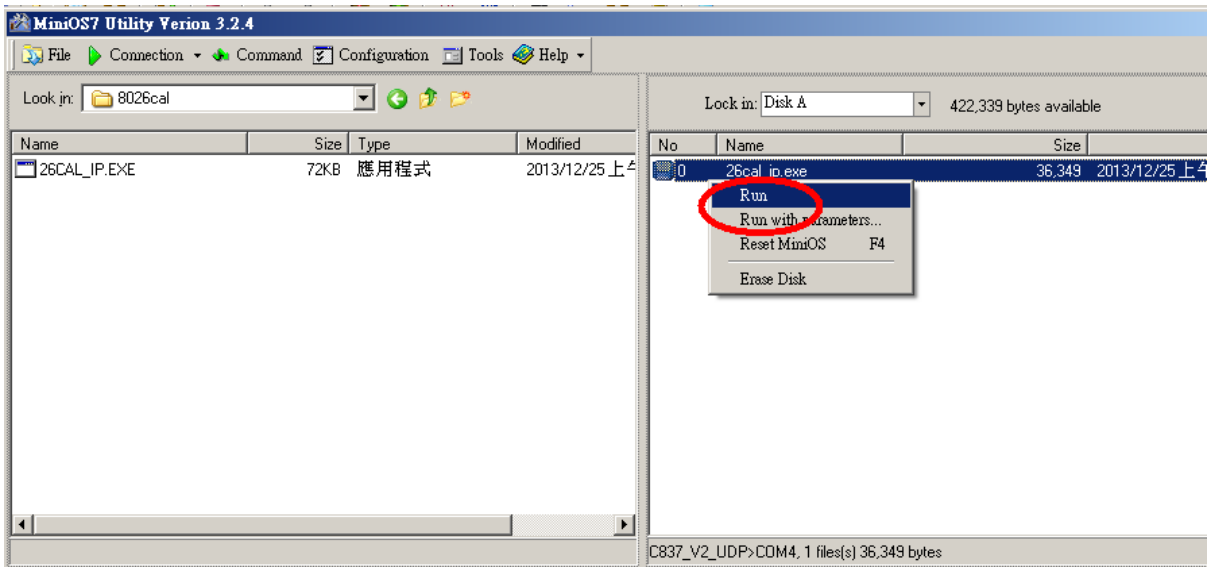
b. From the drop-down list, select the COM port for the Host PC that is connected to the control unit, configure the communication parameters to match those indicate below, and then click the **OK** button.



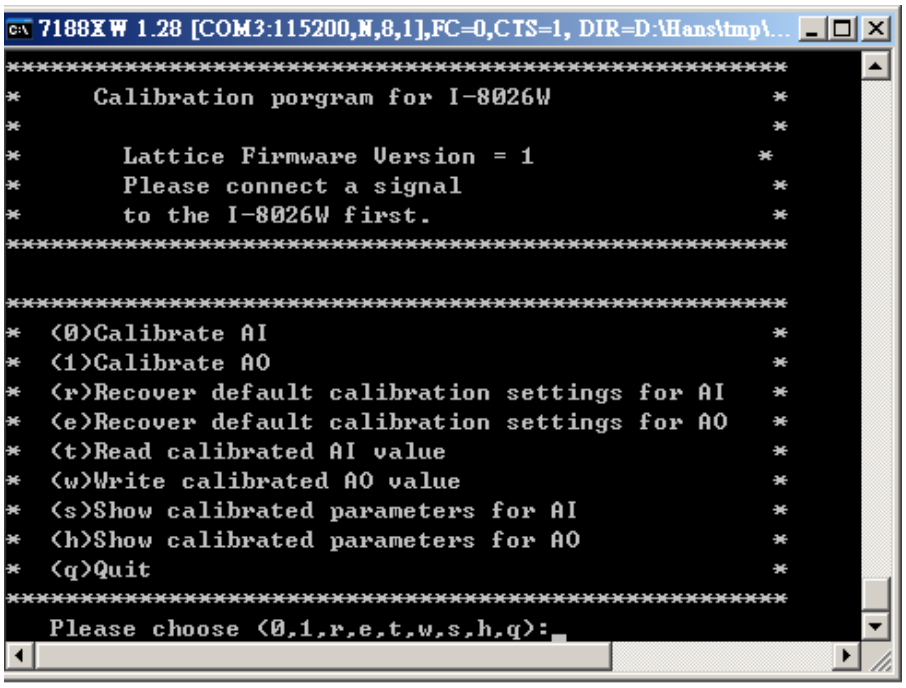
c. Select the name of the calibration program and then click the **Upload** button (or press **F5**) to upload the program for the I-8026W module.



d. Once the file has been uploaded, right-click the name of the updated calibration file and choose **Run**



The calibration program will be executed on the control unit and 7188xw.exe will be executed on the Host PC to provide a command line interface.



Select 0 or 1 to calibrate either the AI or the AO

Step 3. Calibrate the Analog Input for the I-8026W module using the following procedure.

a. Select 0 to calibrate the Analog Input, and type a Gain option (0 - 4) that is to be used for the calibration, and then press **ENTER**.

```
*********************************************************
*  (0)Calibrate AI                                      *
*  (1)Calibrate AO                                      *
*  (r)Recover default calibration settings for AI       *
*  (e)Recover default calibration settings for AO       *
*  (t)Read calibrated AI value                          *
*  (w)Write calibrated AO value                         *
*  (s)Show calibrated parameters for AI                 *
*  (h)Show calibrated parameters for AO                 *
*  (q)Quit                                              *
*********************************************************
   Please choose (0,1,r,e,t,w,s,h,q):0
Please input which gain of AI necessary calibration:
*****************************************
*  (0) Gain_0  -10.00V to +10.00V       *
*  (1) Gain_1  - 5.00V to + 5.00V       *
*  (2) Gain_2  - 2.50V to + 2.50V       *
*  (3) Gain_3  - 1.25V to + 1.25V       *
*  (4) Gain_4  -20.0mAV to +20.0mA      *
*****************************************
   Please choose (0~4):1
Please input which channel of AI necessary calibration:0
Please input a voltage (-10.0~10.0 V)to calibate ch 0:2.9
New ref Gain =0356 ,Save to EEPROM ? (y/n):y
+/-5V    ch 0 is calibrated.
```

b. Determine two values (points) within the range of the selected input type selected for the calibration process. For example, after selecting option 0 (an input range of -10 V to +10 V), +8 V and -8 V can be used as the two calibration points.

c. Set the calibration source output to one of the two points (e.g., +8.0 V in this example)

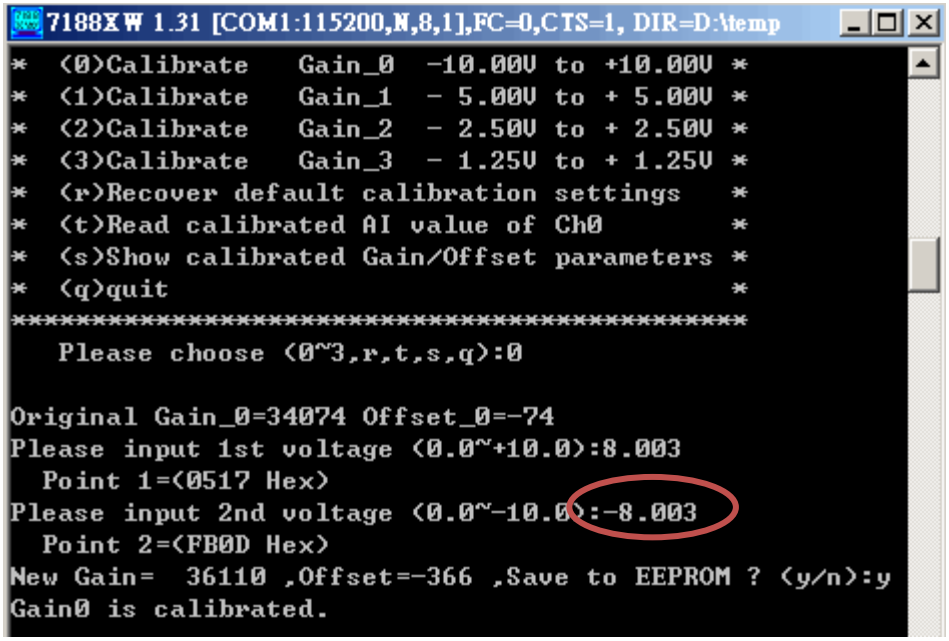
d. At the “Input 1st voltage” prompt on the console, type the value displayed on the meter and then press **ENTER**.


```
7188XW 1.31 [COM1:11520   ,1],FC=0,CTS=1, DIR=D:\temp
*  (0)Calibrate    Gai      0.00V to +10.00V *
*  (1)Calibrate    Gain_1   - 5.00V to + 5.00V *
*  (2)Calibrate    Gain_2   - 2.50V to + 2.50V *
*  (3)Calibrate    Gain_3   - 1.25V to + 1.25V *
*  (r)Recover default calibration settings    *
*  (t)Read calibrated AI value of Ch0         *
*  (s)Show calibrated Gain/Offset parameters  *
*  (q)quit                                    *
*********************************************
   Please choose (0~3,r,t,s,q):0

Original Gain_0=34074 Offset_0=-74
Please input 1st voltage (0.0~+10.0):8.003
  Point 1=(0517 Hex)
Please input 2nd voltage (0.0~-10.0):-8.003
  Point 2=(FB0D Hex)
New Gain=  36110 ,Offset=-366 ,Save to EEPROM ? (y/n):y
Gain0 is calibrated.
```

e. Set the calibration source output to the second point (e.g., -8.0 V in this example).

f. At the “Input 2nd voltage” prompt on the console, type the value displayed on the meter and then press **ENTER.**



The new gain and offset values for this calibration will then be displayed on the console as:

New Gain= 3xxxx, Offset= nnn, Save to EEPROM? (y/n):

g. Type **y** and press **ENTER** to accept the values and save the settings to EEPROM

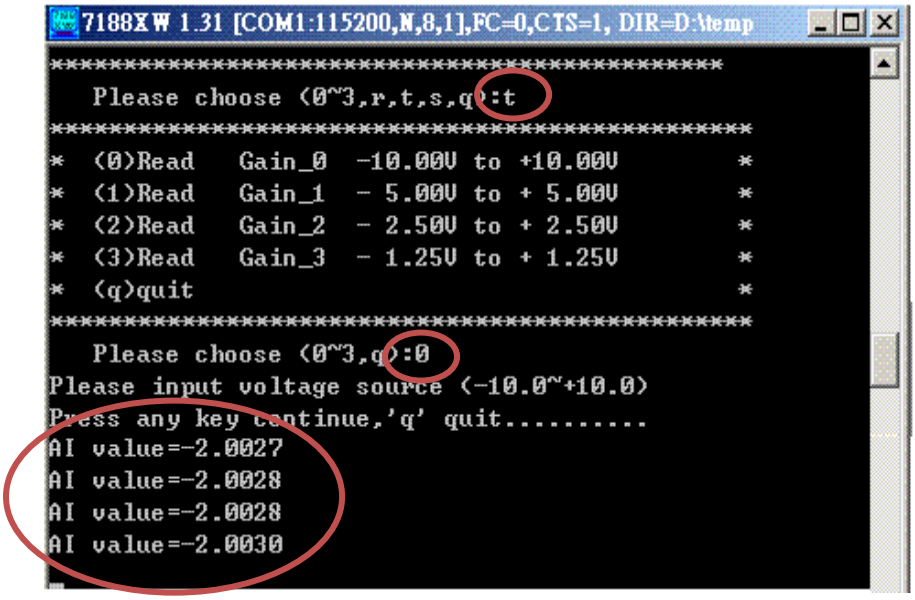
The calibration for the -10 V to +10 V input range is now complete.

4.3. Verifying the Calibration

Step 1. Set the calibration source to output a voltage to channel 0 on the I-8026W module. For example, -2.0 V.

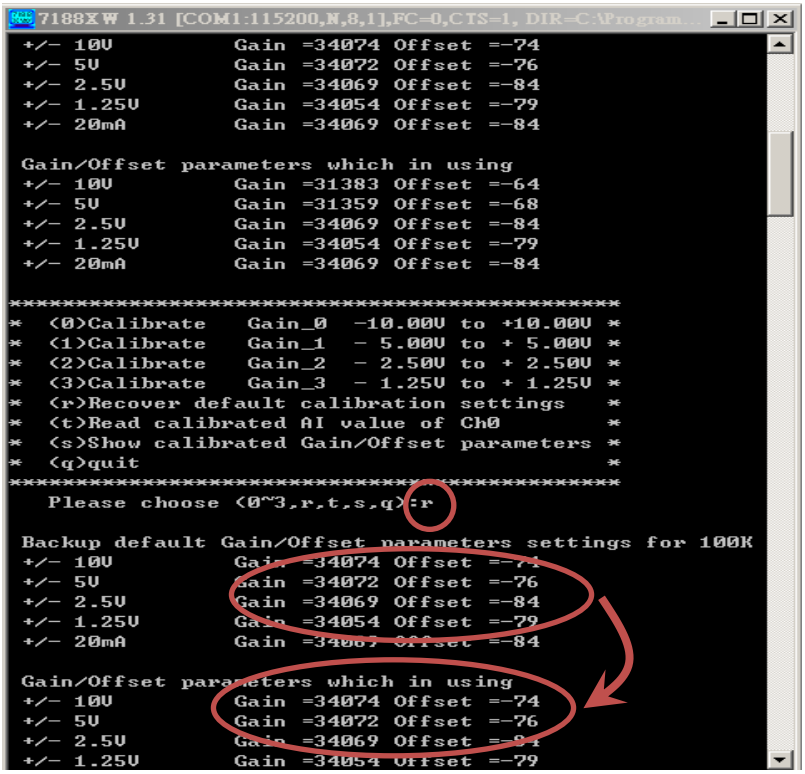
Step 2. In the same console window for the calibration program, type **t** (i.e., read the calibrated Analog Input value for Channel 0), and then select the input type that was previously calibrated (e.g., 0, -10 V to +10 V).

Step 3. Confirm that the values displayed for channel 0 are correct.



4.4. Restoring the Default Calibration Settings

For the I-8026W module, the calibration program provides a **Recover Default Calibration Settings** function **(r)** that can be used to restore the gain and offset values to the factory default settings.



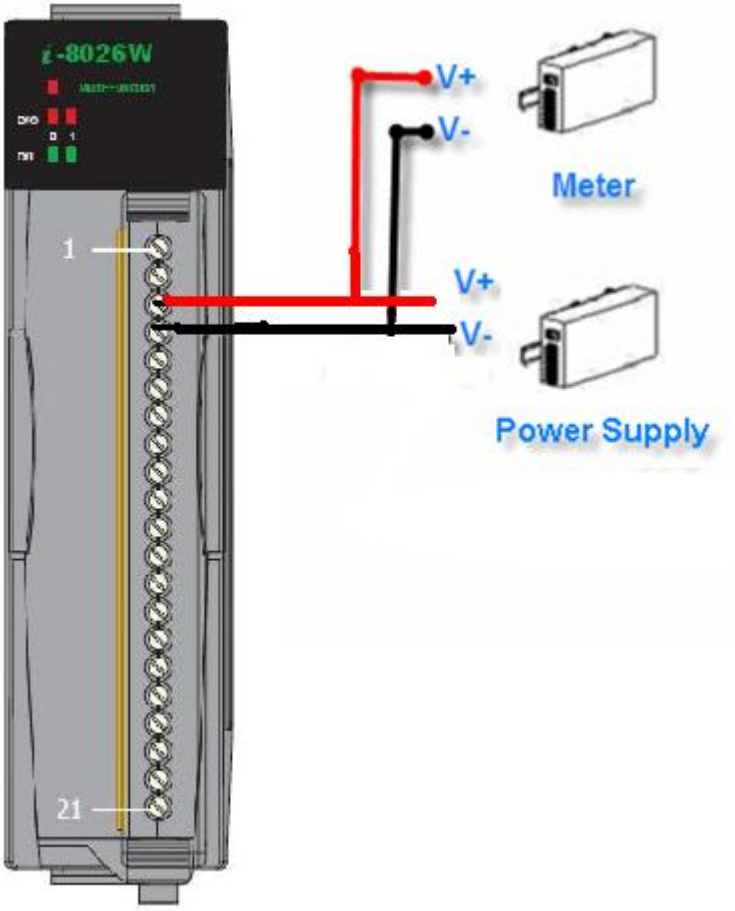
4.5. Calibrating the I-8026W AI on WinCE and WES Units

Step 1. Refer to s, and ensure that the Voltage/Current selection jumper is in the correct position.

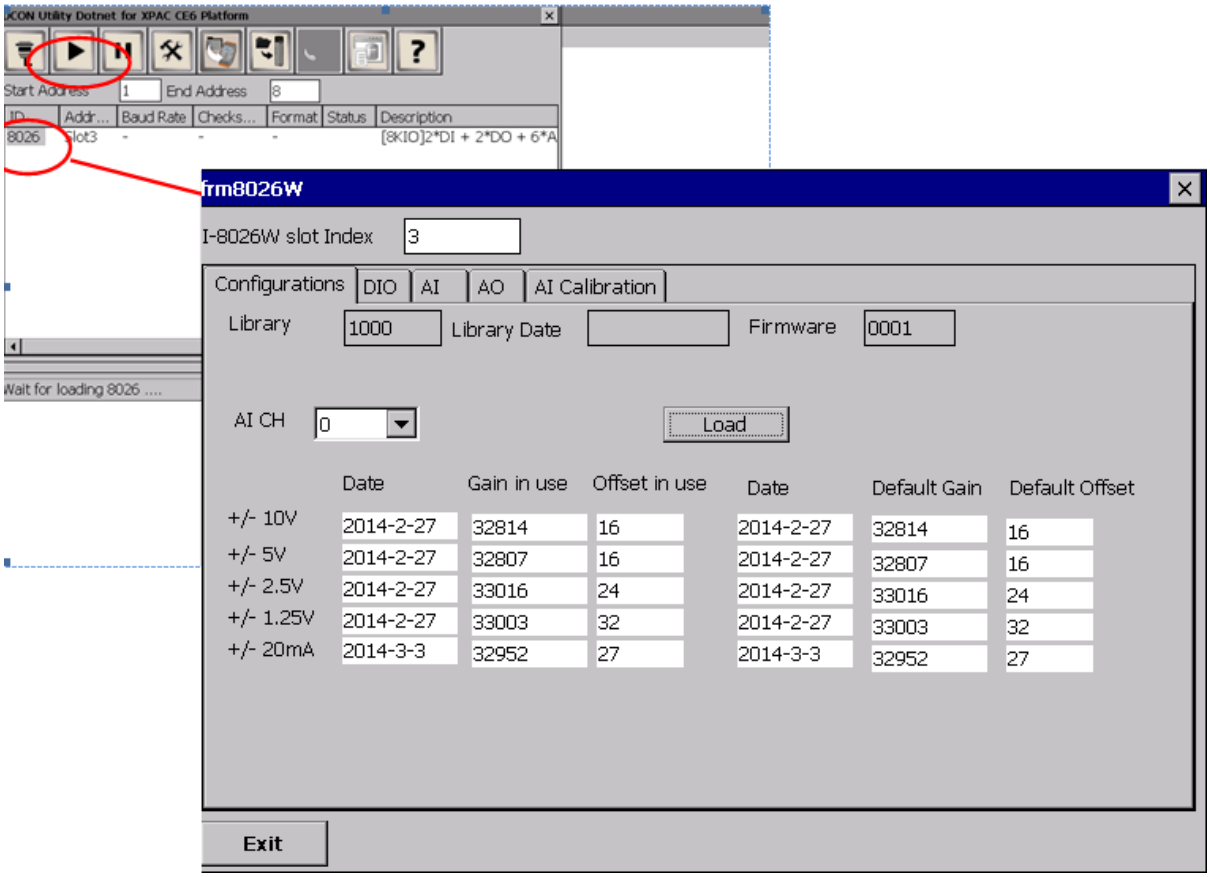
Step 2. Connect a stable calibration source to channel 0 on the I-8026W module that is to be calibrated, as illustrated below.

Step 3. Insert the I-8026W into a vacant slot on the controller and power on the controller.

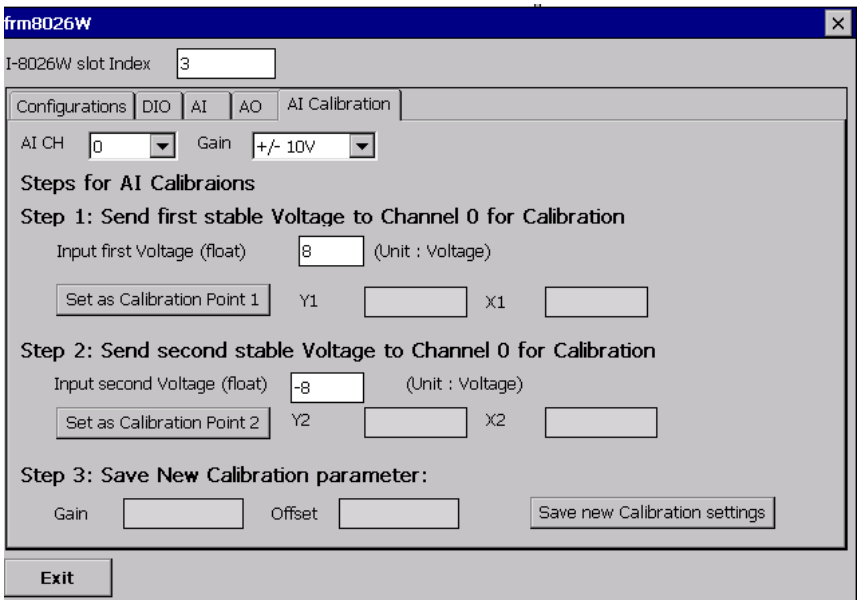
Step 4. Launch the .NET version of the DCON Utility on the WinCE or WES controller to display the Calibration dialog box.



Step 5. In the DCON Utility, search for the I-8026W and then click the name of the module to display the configuration dialog window, as illustrated below.

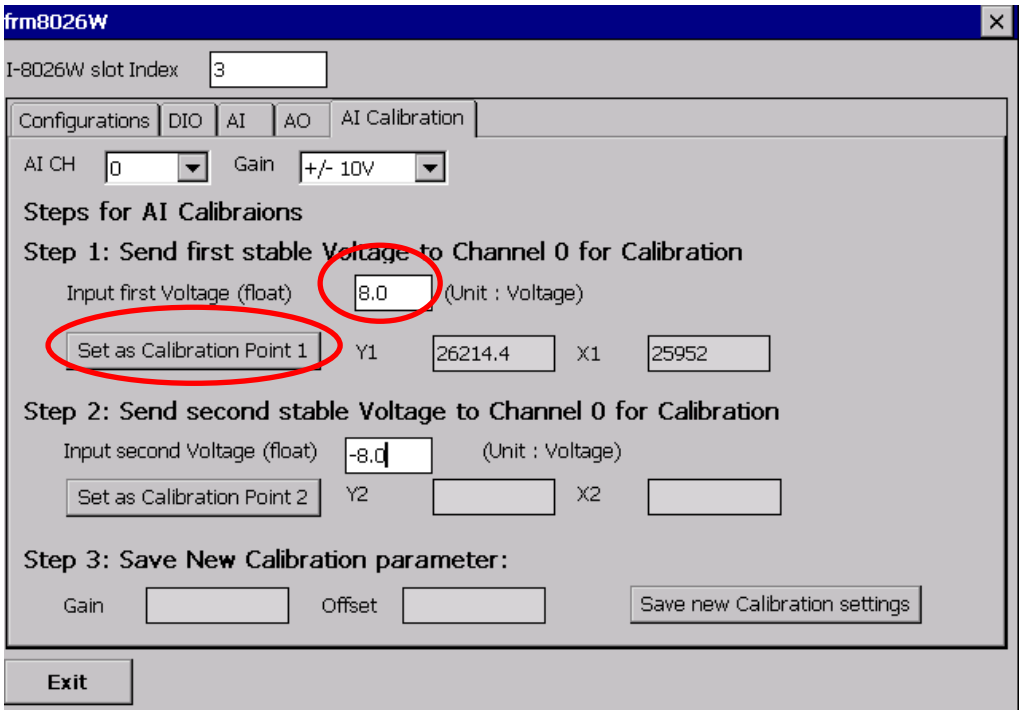


Step 6. Click the **AI Calibration** tab, and then select the **AI CH** and **Gain** values to be used for the calibration from the respective drop-down menus.



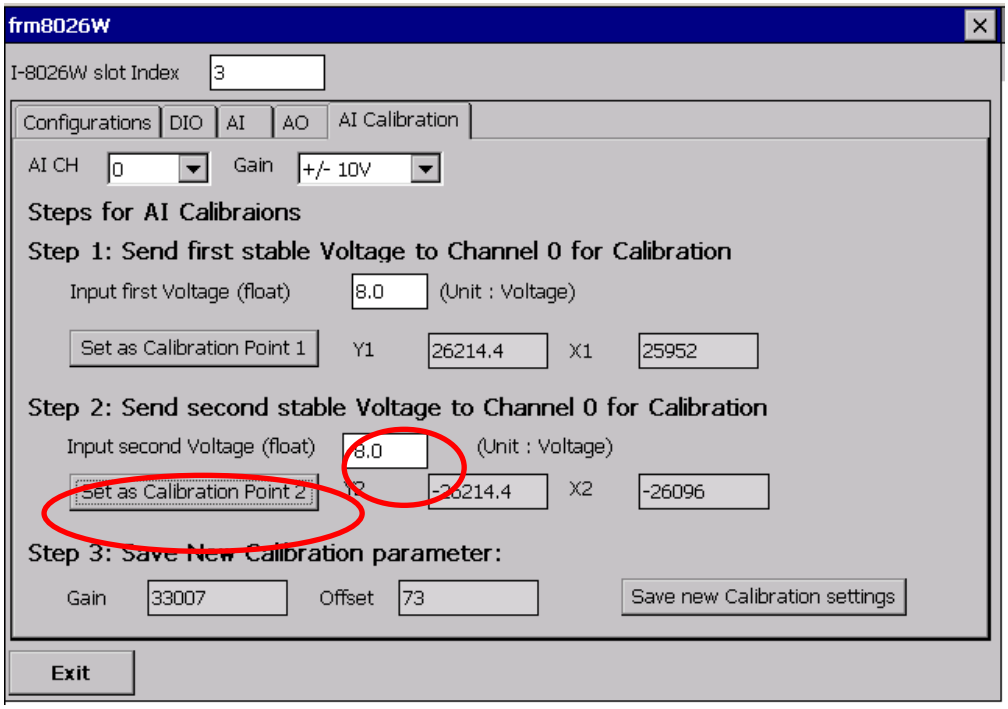
Step 7. Determine two values (points) within the range of the input type selected for the calibration process. For example, after selecting a Gain of -10 V to +10 V as the input range, +8 V and -8 V can be used as the two calibration points. Set the output of the calibration source to one of the two points (for example, +8.0 V)

Step 8. In the **Input First Voltage (float)** text field, type the value displayed on the meter (for example, 8.0), and then click the **Set as Calibration Point 1** button.

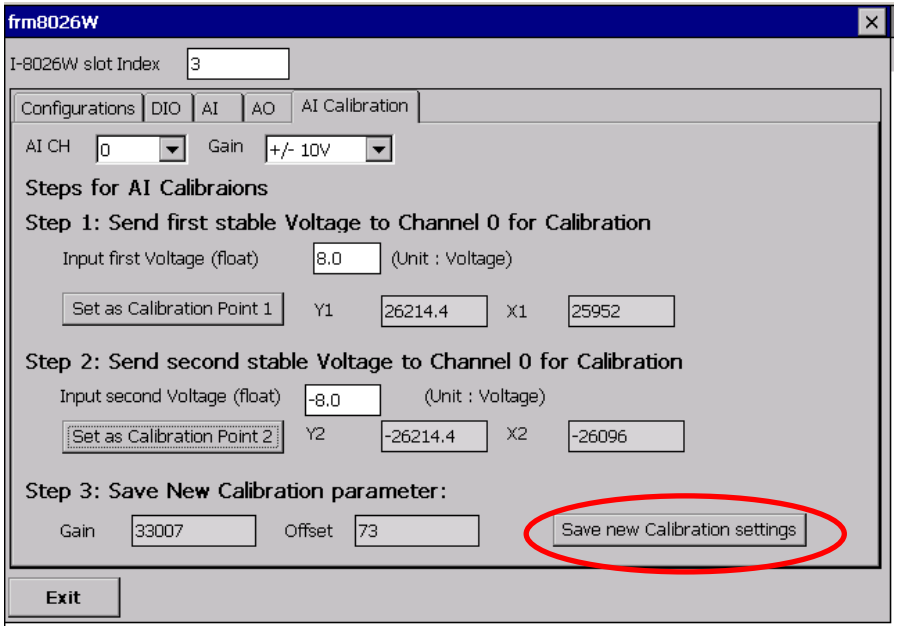


Step 9. Set the output of the calibration source to the second value (for example, -8.0 V)

Step 10. In the **Input Second Voltage (float)** text field, type the value displayed on the meter (for example, -8.0 V), and then click the **Set as Calibration Point 2** button.



Step 11. Click the **Save new Calibration Settings** button to save the new calibration parameters.

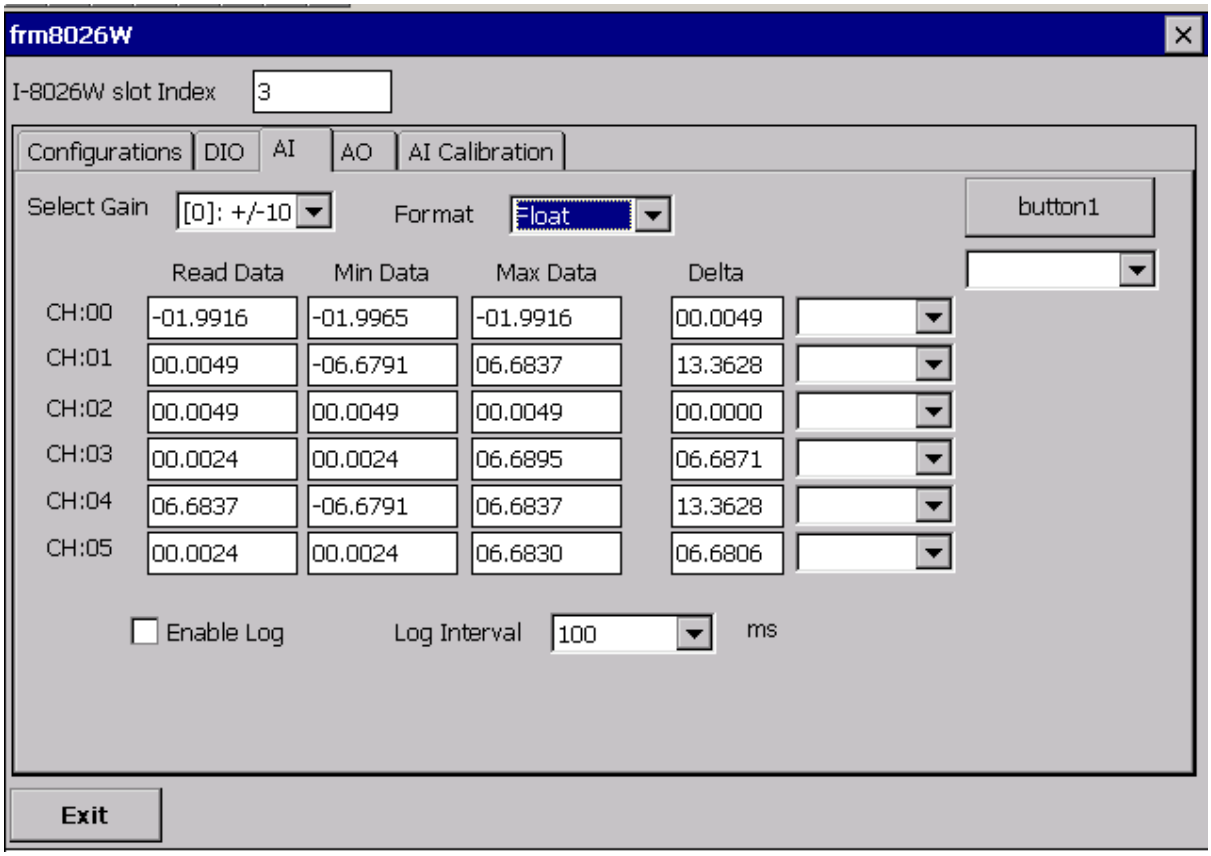


The calibration for the -10 V to +10 V input range is now complete.

4.6. Verifying the Calibration Parameters

Step 1. Set the calibration source to output a voltage to channel 0 on the I-8026W module. For example, -2 V.

Step 2. In the calibration dialog box, click the **AI** tab, and confirm that the AI values are as illustrated in the image below:



The value for Channel 0 is -1.9916 V, which means that the calibration parameters are within range.

5. Troubleshooting

This chapter discusses how to solve some common problems you may encounter while operating the I-8026W module.

This chapter contains:

- How to verify the AI functions on a WinCE or WES unit
- Service request requirements
- What to do when the data read from the I-8026W module seems unstable

5.1. Verifying the AI functions on a WinCE or WES device

If the data read from the I-8026W module is inconsistent with the input signal, and you would like to confirm the input functions, the pac_i8026W_Utility.exe program may be helpful. The utility can **only** be used with modules designed controllers using the **WinCE and WES** platforms, and is located in the I-8026W C# demo program folder for the controller. (See Section 1.6. Location of the Demo Programs for more details)

Step 1. Connect a stable signal to the I-8026W module.

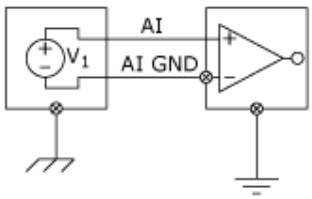
a. Connect the input signal according to whether differential or single-ended Jumper settings are used. (See Section 1.3. **Wire Connections** for more details)

b. Set the input range to +/-10 V

c. Insert the I-8026W module into a vacant slot in the WinCE and WES controller and then power on the power to the controller.

Tips & Warnings

1. A battery output should provide a sufficiently stable signal.
2. A 125 Ω resistor is required when measuring current input.
3. If the result is not as stable as the input signal when measuring voltage using the differential input type, it is recommended that an additional wire is connected between the Vn- and the AGND (analog ground) pins to enhance the accuracy. However, this method has no benefit in enhancing accuracy when measuring current input.



Step 2. Launch the pac_i8026W_demo.exe program

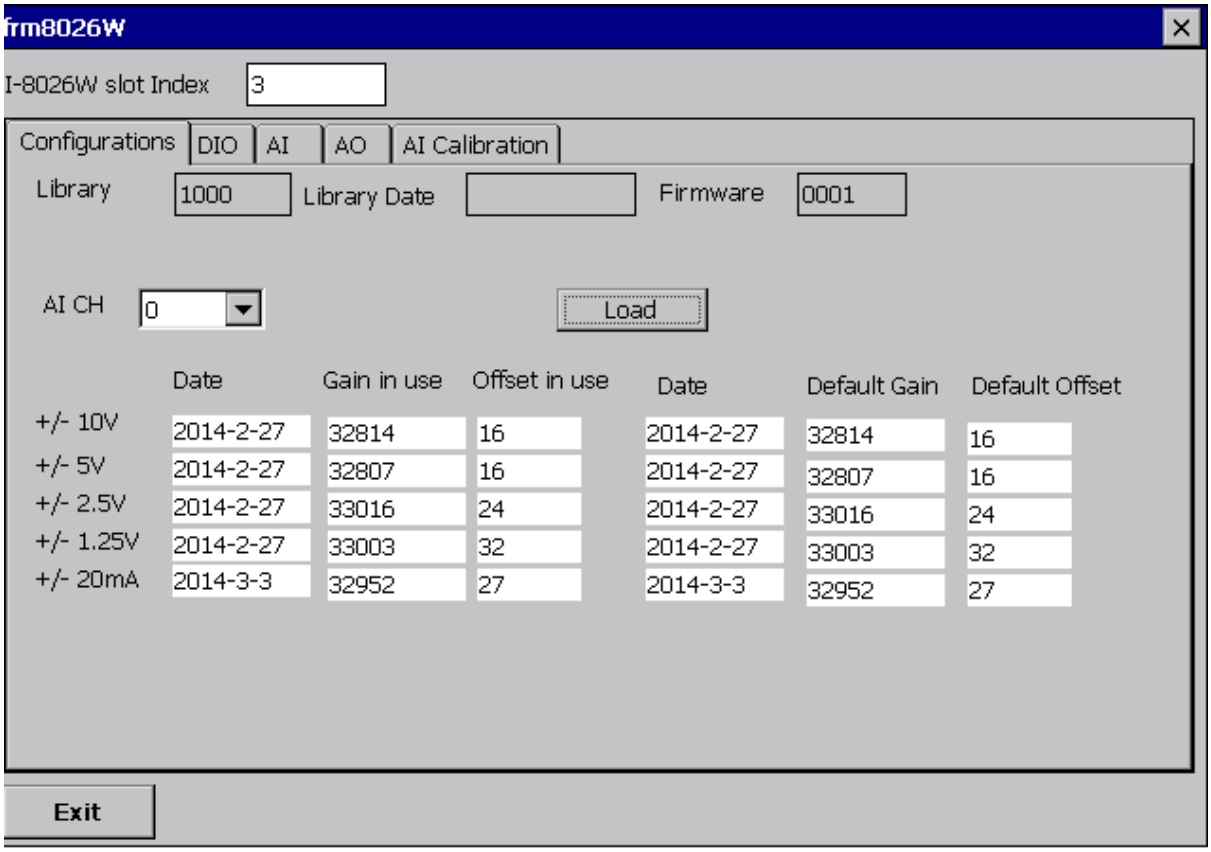
Step 3: Read the information from the I-8026W module.

a. From the I-8026W slot index drop-down list, select the slot that the I-8026W module is connected to.

b. Click the **Basic Information** tab.

The Basic Information page includes:

- The version information for the 8026W.lib FPGA firmware
- The gain and offset values for each input type



Click the **Save** button to save all the information to the ***Slot1_8026W_Info.txt*** file. This information is useful for troubleshooting when service is requested.

5.2. Verifying the Gain and Offset Values

In a normal situation, the gain value should be around 33000. If the value is greatly different from 33000, it means that the value is incorrect. To correct this situation, try the following:

a. Press **Refresh** to retrieve the gain values again and confirm whether or not they are correct

b. Relocate the I-8026W module to a different slot, and then repeat Steps 2 to 3 to confirm whether or not the gain values are correct

5.3. Service Request Requirements

If you are using a stable signal source, such as a battery, to output a signal to the I-8026W module and are receiving incorrect or unstable data, prepare the following three items and e-mail them to service@icpdas.com.

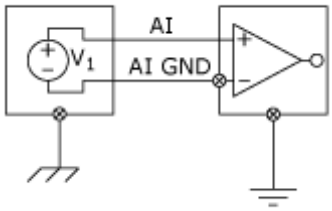
- An image of the physical wiring
- The file saved from the Basic Information tab

5.4. What to do when the data read from the I-8026W seems unstable

If the voltage can be measured correctly when testing using a battery, but not when using the real signal source, the error may be caused by any or all of the following factors:

- A noise-corrupted signal source
- Instability in the signal source
- A floating signal source that is not referenced to a system ground point (earth or building ground)

Because of the nature of the high speed data acquisition function on the I-8026W module, any noise coupled to a signal, or any change in voltage on an unstable source, is also captured. In this situation, signal filtering or isolation should be considered in order to enhance the quality of the signal.



It is recommended to connect the V- to AGND (system ground) when measuring differential signals as the figure shows as below:

Appendix A. Error Codes

Error Code	Definition	Description
0	OK	This indicates that there have been no errors.
-1	ID_ERROR	There was a problem with the module ID.
-2	SLOT_ERROR	There was a slot index error. Slot numbers should be in the range of 0 to 7.
-3	CHANNEL_ERROR	There was a channel index error. Channel numbers should be in the range of 0 to 15.
-4	GAIN_ERROR	There was a gain index error. gain numbers should be in the range of 0 to 4.
-6	NOT_SUPPORT_ERROR	Reading invalid value.

Appendix B. Performance for Read AI Functions

Appendix C. Revision Information

V1.0.0

First Release for the I-8026W module only