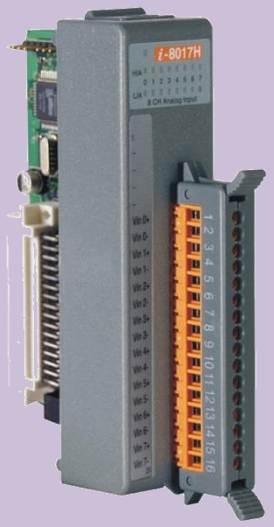
I-8017/I-9017 Series
I/O Module User Manual

V 3.0.1, Nov. 2018



Written by Edward Wu

Edited by Anna Huang

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 purposes 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 a quick response.
Email: service@icpdas.com

Table of Contents

Table of Contents 3

1. Introduction 5
   - 1.1. Specifications 7
   - 1.2. Pin Assignments 9
   - 1.3. Jumper Settings 15
   - 1.4. Wire Connections 20
   - 1.5. Block Diagram 22
2. Quick Start 24
   - 2.1. MiniOS7-based Controllers 25
   - 2.2. Windows-based Controllers 27
3. Demo Programs 30
4. API References 33
   - 4.1. pac_i8017HW_Init 35
   - 4.2. pac_i8017HW_GetFirmwareVersion 37
   - 4.3. pac_i8017HW_GetLibVersion 39
   - 4.4. pac_i8017HW_GetLibDate 40
   - 4.5. pac_i8017HW_GetSingleEndJumper 42
   - 4.6. pac_i8017HW_ReadAI 44
   - 4.7. pac_i8017HW_ReadAI_AVG 46
   - 4.8. pac_i8017HW_ReadAIHex 48
   - 4.9. pac_i8017HW_ReadAIHex_AVG 51
   - 4.10. pac_i8017HW_ReadGainOffset_Info 53
   - 4.11. pac_i8017HW_Read_mA_GainOffset 55
   - 4.12. pac_i8017HW_Select_SingleEnd 57
   - 4.13. pac_i8017HW_Get_D_Sub_Status 59
5. Calibration 61
   - 5.1. MiniOS7-based Controller 63
     - 5.1.1. Calibrating on i-8000 and iPAC-8000 64
     - 5.1.2. Verifying the Calibration 70
     - 5.1.3. Restoring the Default Calibration Settings 71
   - 5.2. Windows-based Controllers 72
     - 5.2.1. Calibrating on WinCE and WES PAC Units 73
     - 5.2.2. Verifying the Calibration 77
     - 5.2.3. Restoring the Default Calibration Settings 78
6. Troubleshooting 79

6.1. Verifying Analog Input functionality on a WinCE or WES PAC device 80
6.2. Service Request Requirements 83
6.3. What to do when the data read from the module seems unstable 84
Appendix A. Error Code 85
Appendix B. Read AI Function Performance 86
Appendix C. Revision History 87

1. Introduction

The I-8017W/I-9017 I/O modules are high performance analog input module, up to 16-channel single-ended or 8-channel differential inputs. It features 14-bit resolution, 90Ks/s sampling rates. It provides isolation protection of 2500 Vrms.

The I-8017 I/O modules can be used to measure both voltage and current source. The I-8017DW and I-8017HCW modules include a Jumper that can be used to set the discrete input circuits to add a125 Ω resistor, so it's not necessary to add external resistor for Differential input. For the I-8017HWan optional external 125Ω resistor is required to measure the current source.

Applications

- High speed data acquisition systems
- Process monitoring and control
- Vibration analysis
- Digital pattern generator from the digital I/O port

Applicable Platform table

The following table shows which platform the module applies to.

Platform	OS	Module
XPAC	XP-8000(WES)	I-8017HW/I-8017DW/I-8017HCW
	XP-8000-Atom (WES)	I-8017HW/I-8017DW/I-8017HCW
	XP-8000-WES7 (WES7)	I-8017HW/I-8017DW/I-8017HCW
	XP-8000-CE6 (WinCE 6.0)	I-8017HW/I-8017DW/I-8017HCW
	XP-8000-Atom-CE6 (WinCE 6.0)	I-8017HW/I-8017DW/I-8017HCW
	XP-9000-WES7(WES7)	I-9017/I-9017-15/I-9017C-15
WinPAC	WP-8000 (CE 5.0/7.0)	I-8017HW/I-8017DW/I-8017HCW
	WP-9000-CE7 (CE 7.0)	I-9017/I-9017-15/I-9017C-15
LinPAC	LinPAC-8000(Linux kernel 3.2/4.4)	I-8017HW/I-8017DW/I-8017HCW
	LinPAC-9000(Linux kernel 3.2/4.4)	I-9017/I-9017-15/I-9017C-15
IPAC	iPAC-8000 (MiniOS7)	I-8017HW/I-8017DW/I-8017HCW
	I-8000(MiniOS7)	I-8017HW/I-8017DW/I-8017HCW

The I-8017DW module is equipped with a D-sub connection, meaning that it can be connected using a 37-pin D-sub Connector, as shown in the image below:

For more detailed information regarding 37-pin D-sub Connectors refer to the models indicated in the table below:

Model	Description
DN-37-A	I/O Connector Block with DIN-Rail Mounting and 37-pin D-sub Connector (Pitch: 5.08 mm)
DN-37-381-A	I/O Connector Block with DIN-Rail Mounting and 37-pin D-sub Connector (Pitch: 3.81 mm)
CA-3705A	Male-Female D-sub Cable 0.5 m
CA-3710A	Male-Female D-sub Cable 1 m
CA-3715A	Male-Female D-sub Cable 1.5 m

1.1. Specifications

I-8017HW/I-8017HCW/I-8017DW

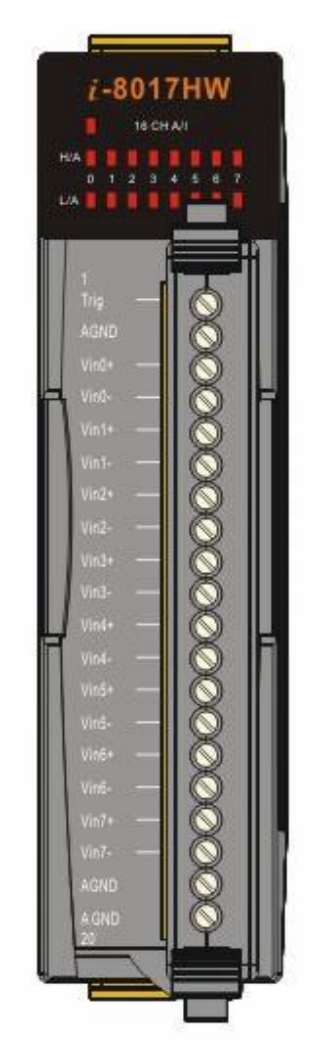
Model	I-8017HW	I-8017DW	I-8017HCW
Analog Output			
Channels	8-ch Differential/16-Single-ended		
Voltage Input Range	±1.25, ±2.5, ±5 V, ±10 V		-
Current Input Range	±20 mA (Requires OptionalExternal 125 Ω Resistor)		±20 mA (Jumper Select)
Resolution	14-bit		
Sample Rate	Single Channel Polling Mode :90K S/s Single Channel Interrupt Mode: 50K S/s 8 channel Scan Mode : 16 K S/s		
Accuracy	0.1% of FSR		
Zero Drift	± 0.1 uV/°C		
Span Drift	± 10 ppm/°C		
ESD Protection	±4 kV Contact for each Terminal		
Input Impedance	20 K, 200 K, 20 M (Jumper Select)		
Input Bandwidth	100 KHz		
Connector	20 Pin Terminal Block		
LED Indicators			
System LED Indicator	1 LED as Power Indicator		
I/O LED Indicator	16 LEDs as User defined Indicators		
Isolation			
Intra-module Isolation, Field-to-Logic	2500 Vrms		
Power			
Power Consumption	2 W Max.		
Mechanical			
Dimension (L x W x H)	102 mm x 30 mm x 115 mm		
Environment			
Operating Temperature	-25 °C ~ +75°C		
Storage Temperature	-30 °C ~ +80°C		
Humidity	10% ~ 90% RH, non-condensing		

I-9017/I-9017-15/I-9017C-15

Model	I-9017	I-9017-15	I-9017C-15
Analog Output			
Channels	8 Differential/ 16 Single-ended	15 Differential/ 30 Single-ended	15 Differential
Voltage Input Range	±1.25, ±2.5, ±5 V, ±10 V		-
Current Input Range	±20 mA (Requires OptionalExternal 125 Ω Resistor)		±20 mA
Resolution	14-bit		
Sample Rate	Single Channel Polling Mode :90K S/s Single Channel Interrupt Mode: 50K S/s 15-channel Scan Mode : 16 K S/s		
Accuracy	0.1% of FSR		
Input Bandwidth	100 KHz		
Zero Drift	± 0.1 uV/°C		
Span Drift	± 10 ppm/°C		
Input Impedance	20 K, 200 K, 20 M (Jumper Select)		-
LED Indicators			
System LED Indicator	1 LED as Power Indicator		
I/O LED Indicator	16 LED as Status Indicator		
EMS Protection			
ESD (IEC 61000-4-2)	±4 kV Contact for each Terminal ±8 kV Air for Random Point		
Isolation			
Intra-module Isolation, Field-to-Logic	2500 Vrms		
Power			
Power Consumption	2 W Max.		
Mechanical			
Dimension (L x W x H)	144 mm x 30.3 mm x 134 mm		
Environment			
Operating Temperature	-25 °C ~ +75°C		
Storage Temperature	-40°C ~ +85°C		
Humidity	10 % ~ 90% RH, non-condensing		

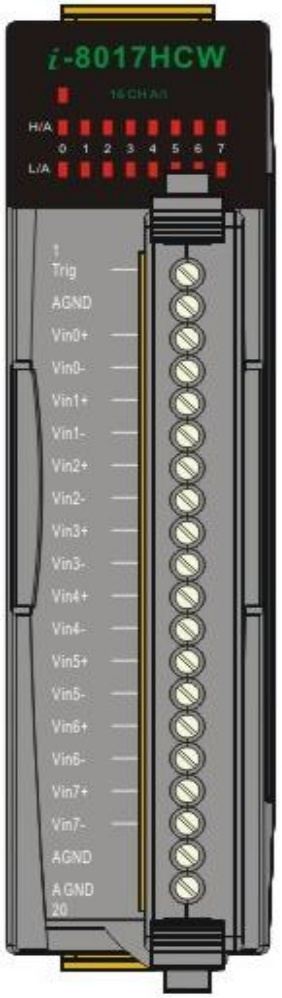
1.2. Pin Assignments

I-8017HW:



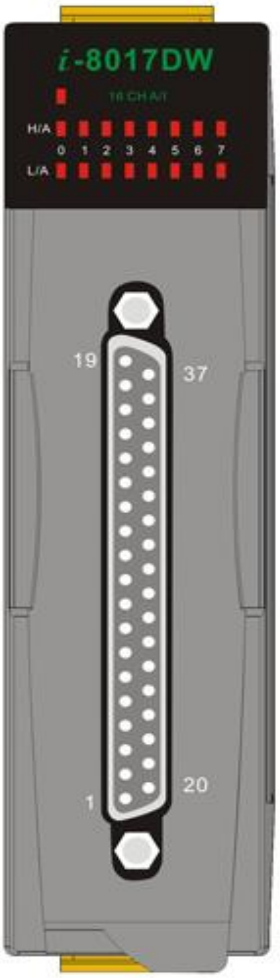
Terminal No.	Pin Assignment Name	
	Differential	Single-ended
01	Trig	Trig
02	AGND	AGND
03	Vin0 +	Vin0
04	Vin0 -	Vin8
05	Vin1 +	Vin1
06	Vin1 -	Vin9
07	Vin2 +	Vin2
08	Vin2 -	Vin10
09	Vin3 +	Vin3
10	Vin3 -	Vin11
11	Vin4 +	Vin4
12	Vin4 -	Vin12
13	Vin5 +	Vin5
14	Vin5 -	Vin13
15	Vin6 +	Vin6
16	Vin6 -	Vin14
17	Vin7 +	Vin7
18	Vin7 -	Vin15
19	AGND	AGND
20	AGND	AGND

I-8017HCW:



Terminal No.		Pin Assignment	
		Differential	Single-ended
	01	Trig	Trig
	02	AGND	AGND
	03	Vin0+	Vin0
	04	Vin0-	Vin8
	05	Vin1+	Vin1
	06	Vin1-	Vin9
	07	Vin2+	Vin2
	08	Vin2-	Vin10
	09	Vin3+	Vin3
	10	Vin3-	Vin11
	11	Vin4+	Vin4
	12	Vin4-	Vin12
	13	Vin5+	Vin5
	14	Vin5-	Vin13
	15	Vin6+	Vin6
	16	Vin6-	Vin14
	17	Vin7+	Vin7
	18	Vin7-	Vin15
	19	AGND	AGND
	20	AGND	AGND

I-8017DW:

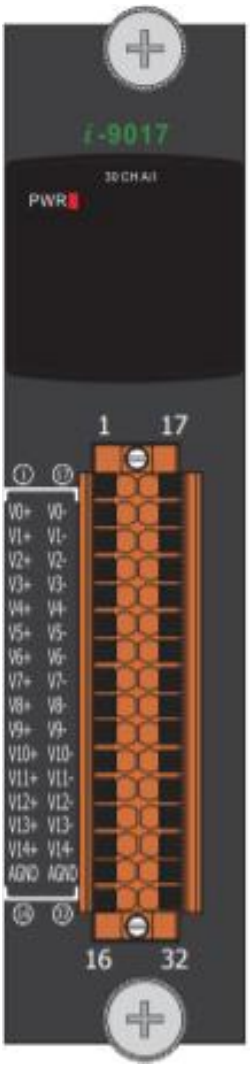


Pin Assignment		Terminal No.		Pin Assignment	
Differential	Single-ended			Differential	Single-ended
AGND	AGND	19	37	BK Sensor	BK Sensor
Trig	Trig	18	36	-	-
AI7-	Ai15	17	35	-	-
AI7+	AI7	16	34	-	-
AI6-	AI14	15	33	-	-
AI6+	AI6	14	32	-	-
AI5-	AI13	13	31	-	-
AI5+	AI5	12	30	-	-
AI4-	AI12	11	29	-	-
AI4+	AI4	10	28	-	-
AI3-	AI11	09	27	-	-
AI3+	AI3	08	26	-	-
AI2-	AI10	07	25	-	-
AI2+	AI2	06	24	-	-
AI1-	AI9	05	23	-	-
AI1+	AI1	04	22	-	-
AI0-	AI8	03	21	AGND	AGND
AI0+	AI0	02	20	AGND	AGND
BK Sensor	BK Sensor	01			

I-9017:

Pin Assignment		Terminal No.		Pin Assignment	
Differential	Single-ended			Differential	Single-ended
Trig+	Trig+	1	11	Trig-	Trig-
V0+(I0+)	Vin0	2	12	V0-(I0-)	Vin8
V1+(I1+)	Vin1	3	13	V1-(I1-)	Vin9
V2+(I2+)	Vin2	4	14	V2-(I2-)	Vin10
V3+(I3+)	Vin3	5	15	V3-(I3-)	Vin11
V4+(I4+)	Vin4	6	16	V4-(I4-)	Vin12
V5+(I5+)	Vin5	7	17	V5-(I5-)	Vin13
V6+(I6+)	Vin6	8	18	V6-(I6-)	Vin14
V7+(I7+)	Vin7	9	19	V7-(I7-)	Vin15
AGND	AGND	10	20	AGND	AGND

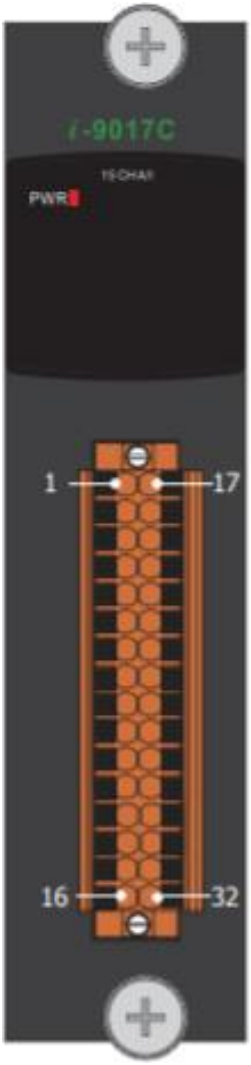
I-9017-15:



Pin Assignment		Terminal No.		Pin Assignment	
V0+	Vin0	01	17	V0 -	Vin15
V1+	Vin1	02	18	V1 -	Vin16
V2+	Vin2	03	19	V2 -	Vin17
V3+	Vin3	04	20	V3 -	Vin18
V4+	Vin4	05	21	V4 -	Vin19
V5+	Vin5	06	22	V5 -	Vin20
V6+	Vin6	07	23	V6 -	Vin21
V7+	Vin7	08	24	V7 -	Vin22
V8+	Vin8	09	25	V8-	Vin23
V9+	Vin9	10	26	V9 -	Vin24
V10+	Vin10	11	27	V10 -	Vin25
V11+	Vin11	12	28	V11 -	Vin26
V12+	Vin12	13	29	V12 -	Vin27
V13+	Vin13	14	30	V13 -	Vin28
V14+	Vin14	15	31	V14 -	Vin29
AGND	AGND	16	32	AGND	AGND

32-pin Connector

I-9017C-15:



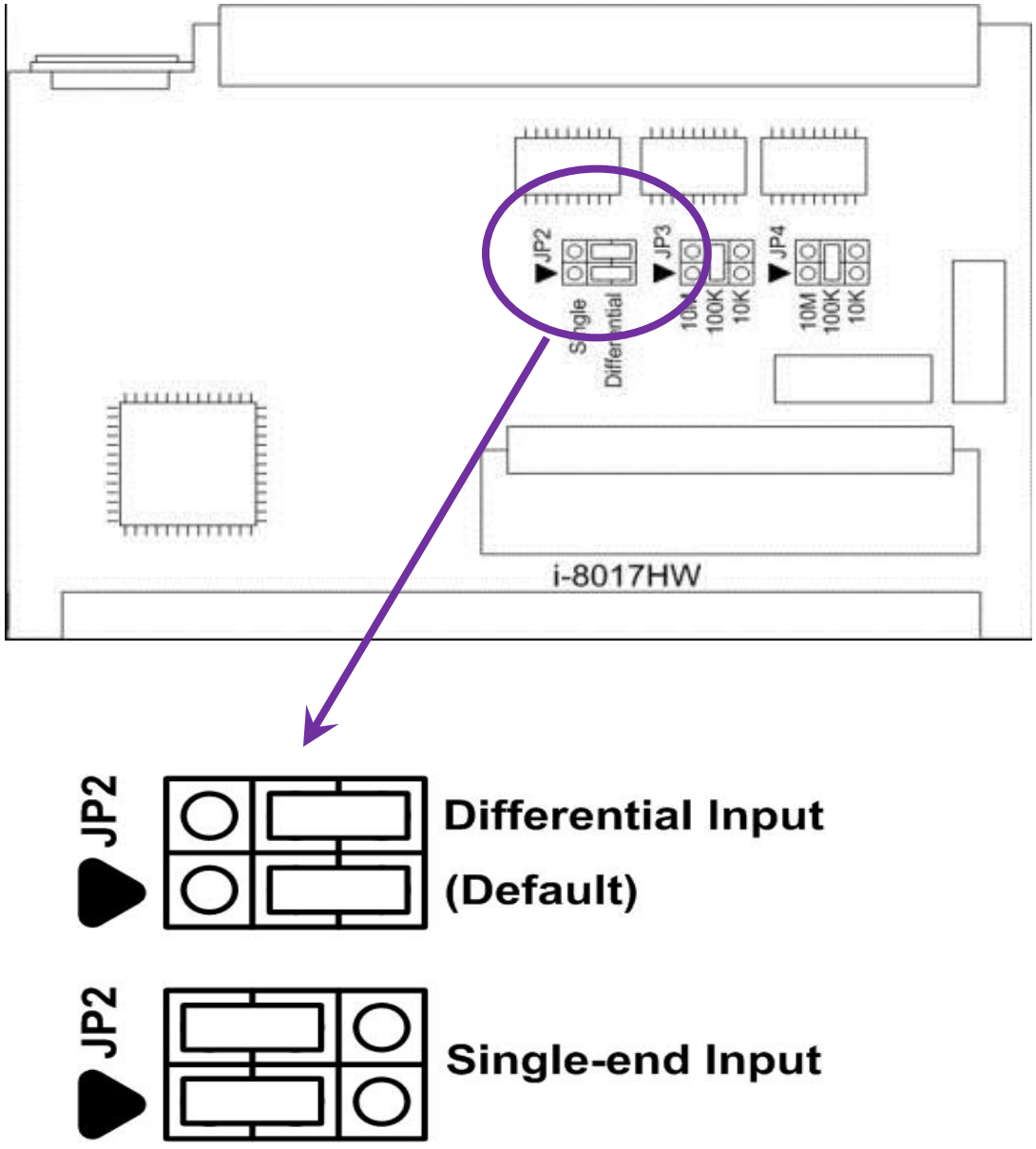
Pin Assignment	Terminal No.		Pin Assignment
I0+	01	17	I0 -
I1+	02	18	I1 -
I2+	03	19	I2 -
I3+	04	20	I3 -
I4+	05	21	I4 -
I5+	06	22	I5 -
I6+	07	23	I6 -
I7+	08	24	I7 -
I8+	09	25	I8-
I9+	10	26	I9 -
I10+	11	27	I10 -
I11+	12	28	I11 -
I12+	13	29	I12 -
I13+	14	30	I13 -
I14+	15	31	I14 -
AGND	16	32	AGND

32-pin Connector

1.3. Jumper Settings

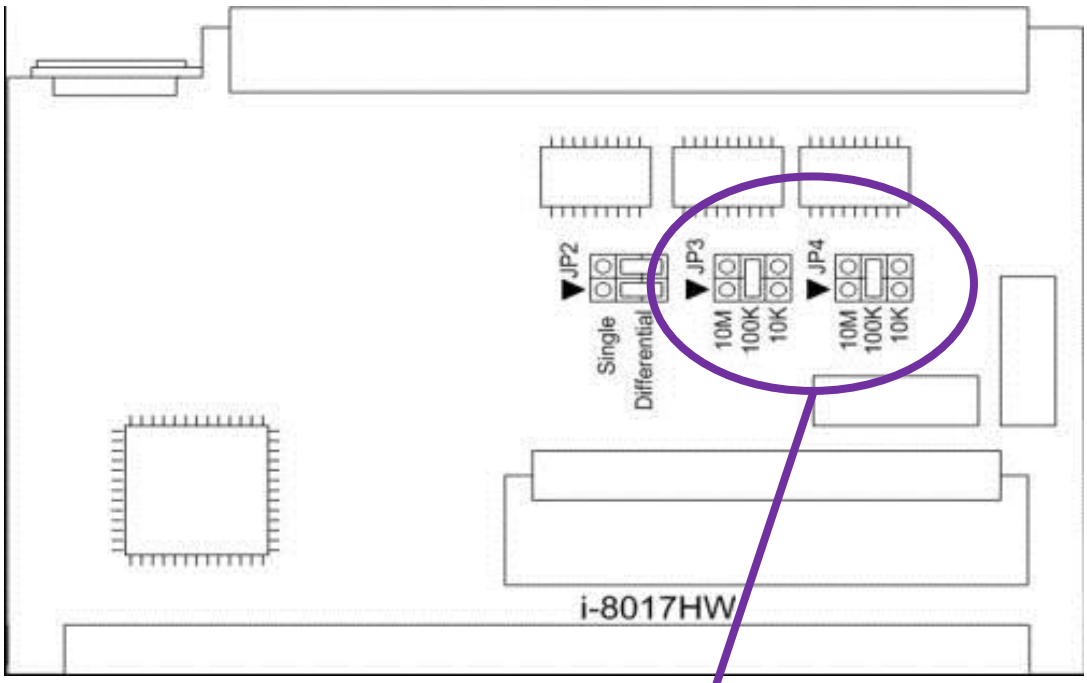
I-8017HW

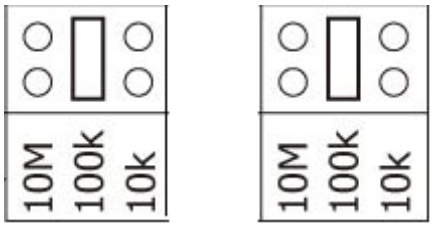
Single-ended and Differential Jumper:



This jumper is used to set the discrete input circuits as either “Single-ended” or “Differential” inputs.

Adjusting the Input impedance





Select Input Impedance: 200 kΩ (Default)

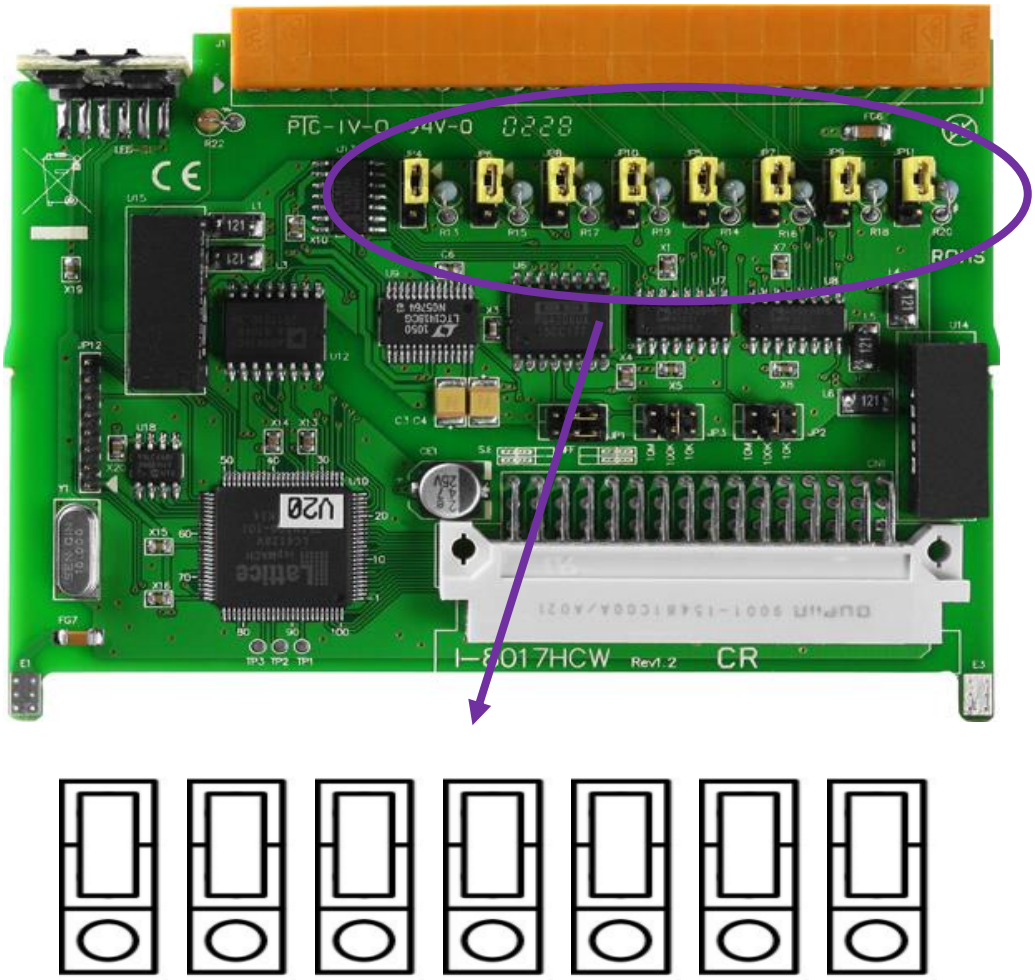
Note: 1. The Jumpers should set on the same value
2. Input Impedance = 2 x setting value

The I-8017 series modules allows three input impedance options including as 20 kΩ, 200kΩ (the default setting) and 20MΩ to meet system requirements.

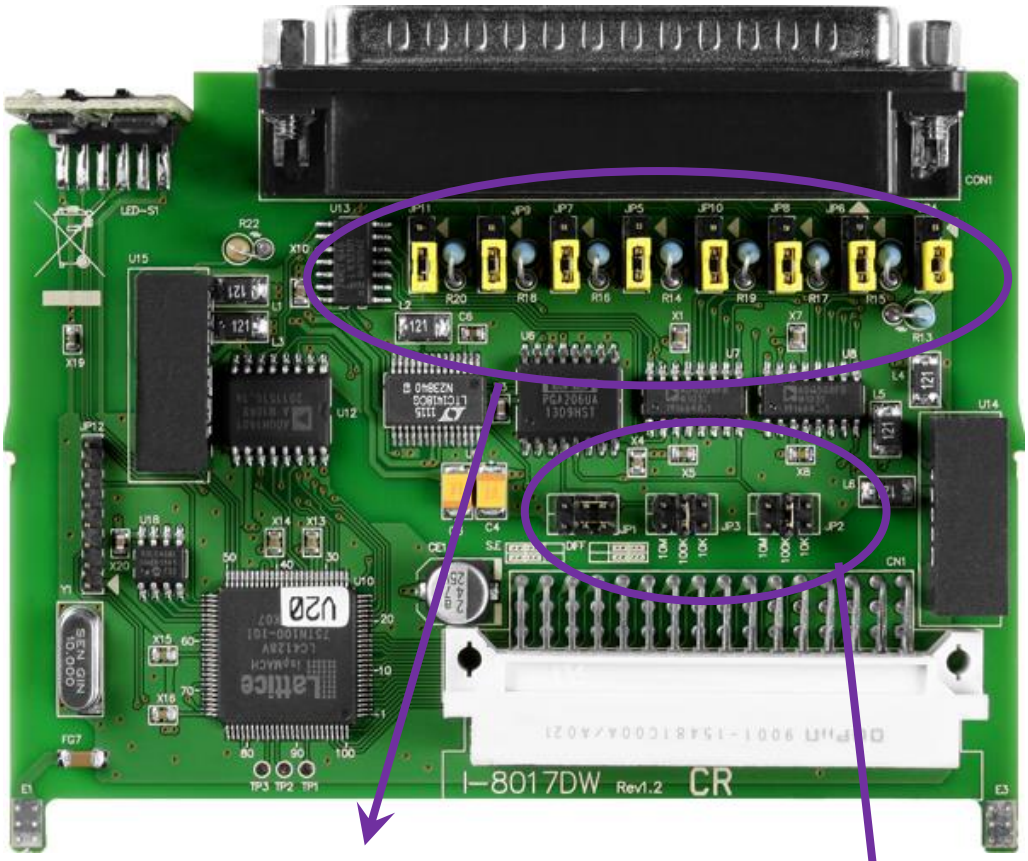
In most cases, 200kΩ is sufficient. Note that each time the input impedance is adjusted on a calibrated module, the module must be recalibrated. For more details, refer to the relevant Calibration information, which can be found in Section 5.1 if you are using either an I-8000 or iPAC-8000 (MiniOS7 platform) controller, or in Section 5.2 if you are using a WinCE or WES platform unit.

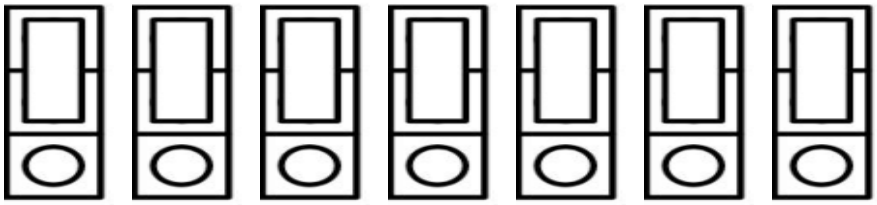
125Ω Resistor Jumper

By default, the I-8017HCW module is configured for current source measurement, and the I-8017DW is configured for voltage measurement, as illustrated below:



The default jumper position for current measurement on the I-8017HCW





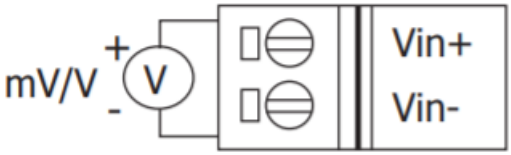
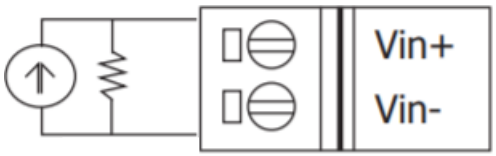
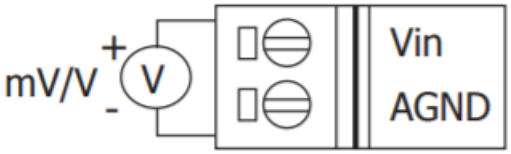
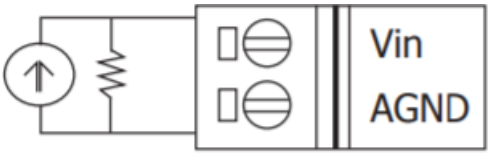
The default jumper position for voltage measurement on the I-8017DW

The jumper settings used to adjust both the input impedance and the single-ended and differential input on the I-8017HW are the same as those for the I-8017DW,I-8017HCW and the I-9017.

The default jumper position for voltage measurement on the I-9017

1.4. Wire Connections

I-9017

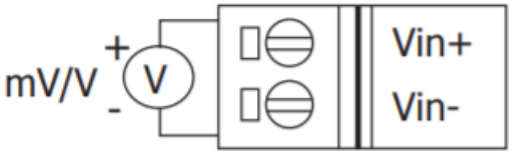
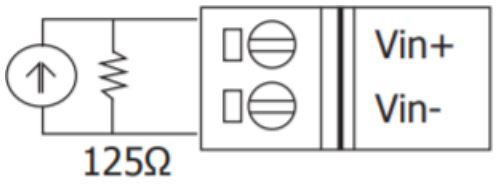
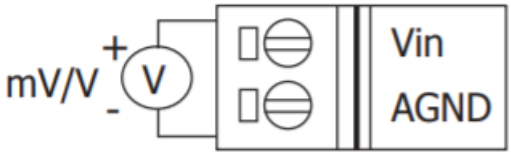
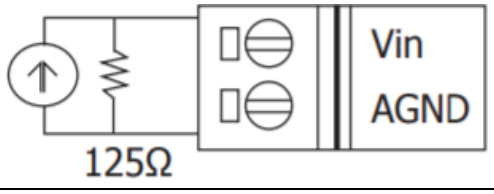
	Voltage Input Wiring	Current Input Wiring
Differential		
Single-ended		

Note:

Differential Input Type: Current Input Wiring need to jumper at current input.

Single-ended Input Type: Current Input Wiring need to jumper at voltage input, an options external 125 Ω resistor is required.

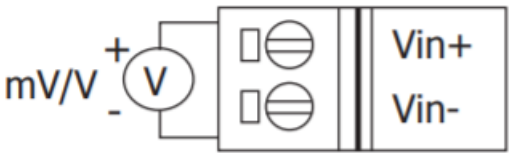
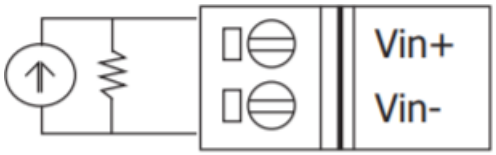
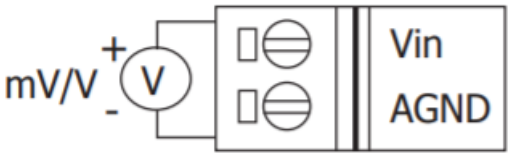
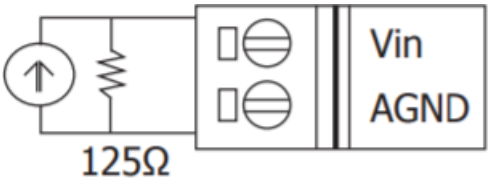
I-8017HW/I-9017-15:

	Voltage Input Wiring	Current Input Wiring
Differential		
Single-ended		

Note:

When connecting to a current source, an optional external 125 Ω resistor is required.

I-8017DW/I-8017HCW

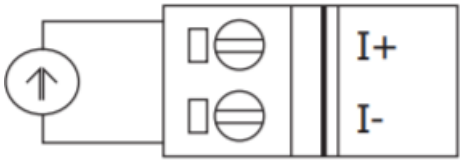
	Voltage Input Wiring	Current Input Wiring
Differential		
Single-ended		

Note:

Differential Input Type: Current Input Wiring need to jumper at current input.

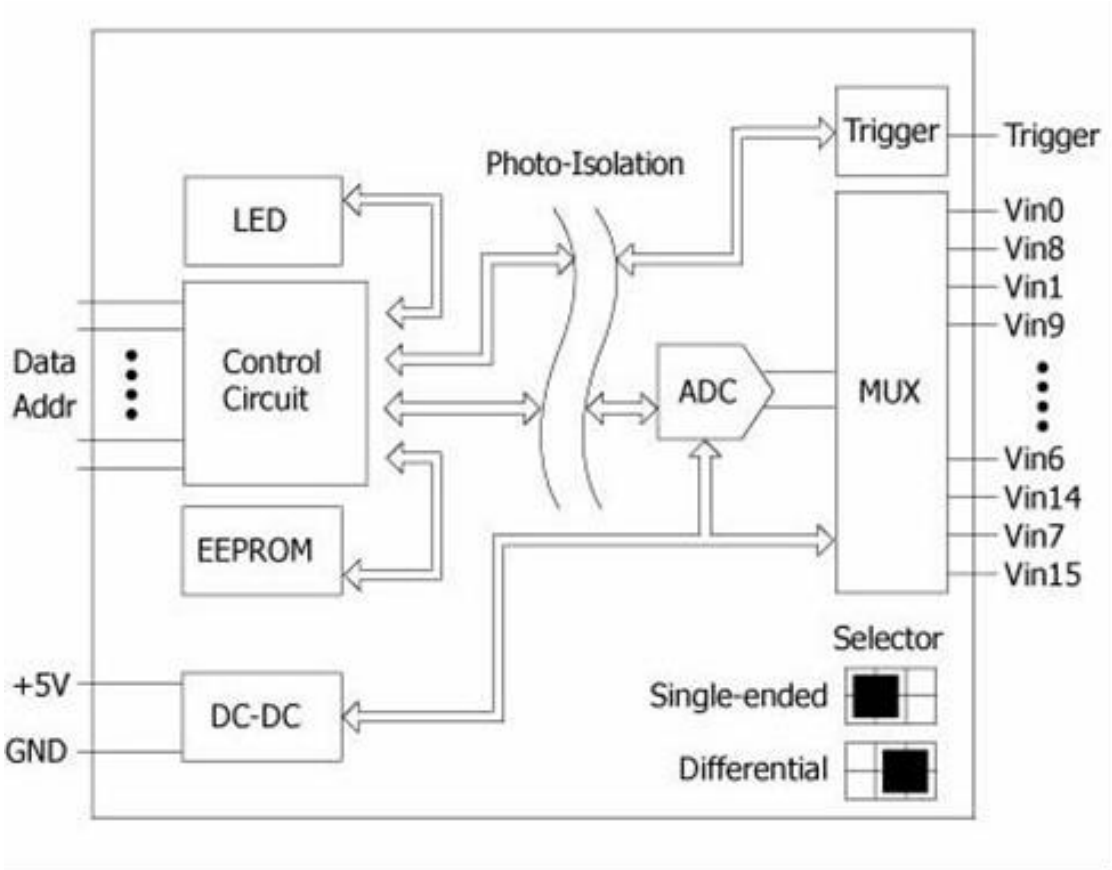
Single-ended Input Type: Current Input Wiring need to jumper at voltage input, an options external 125 Ω resistor is required.

I-9017C-15

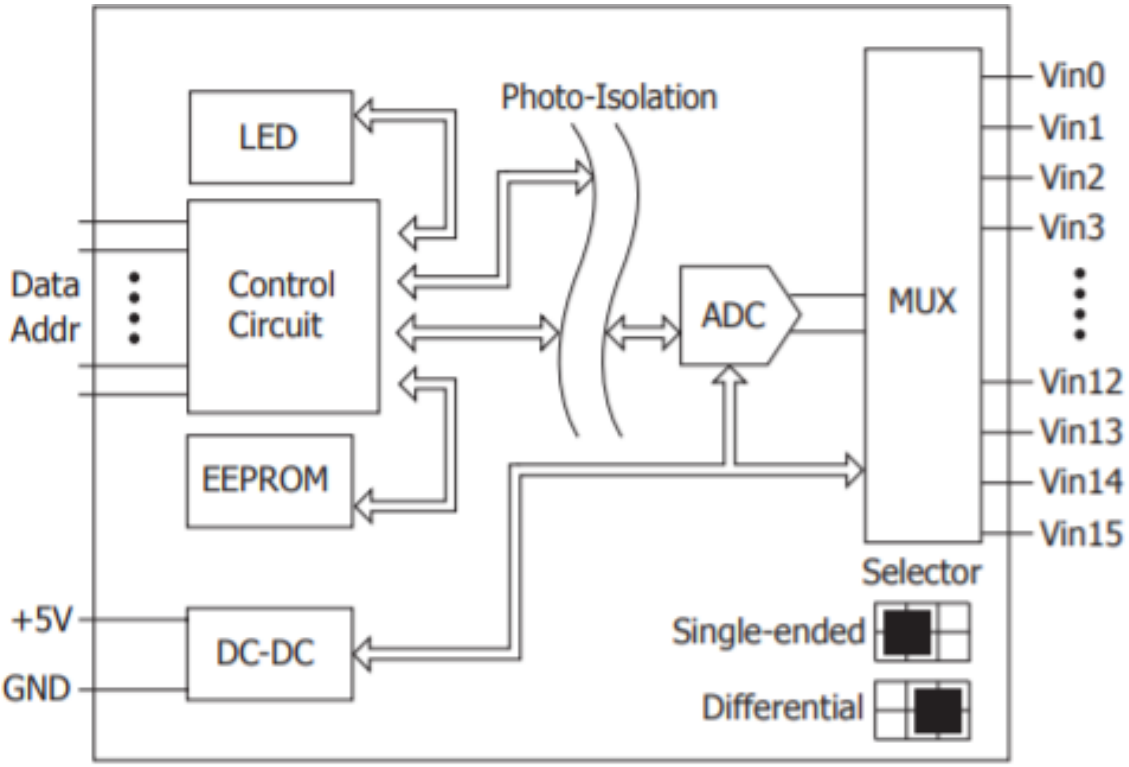
	Current Input Wiring
Differential	

1.5. Block Diagram

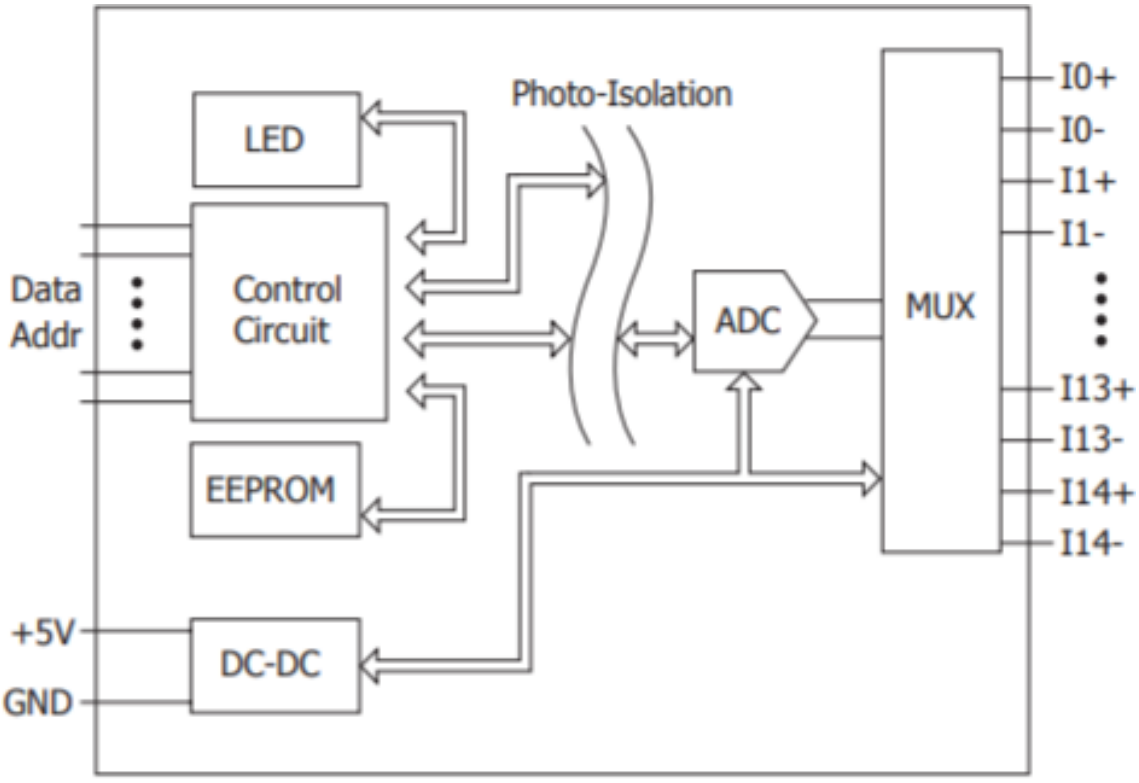
I-8017HW/I-8017DW/I-8017HCW:



I-9017/I-9017-15



I-9017C-15



2. Quick Start

This section provides a Getting Started guide when using on either the MiniOS7 or Windows platforms.

- For MiniOS7-based Controllers, see section 2.1 (i-8000 and iPAC-8000 modules)
- For Windows-based Controllers, see section 2.2 (WinCE and WES modules)

2.1. MiniOS7-based Controllers

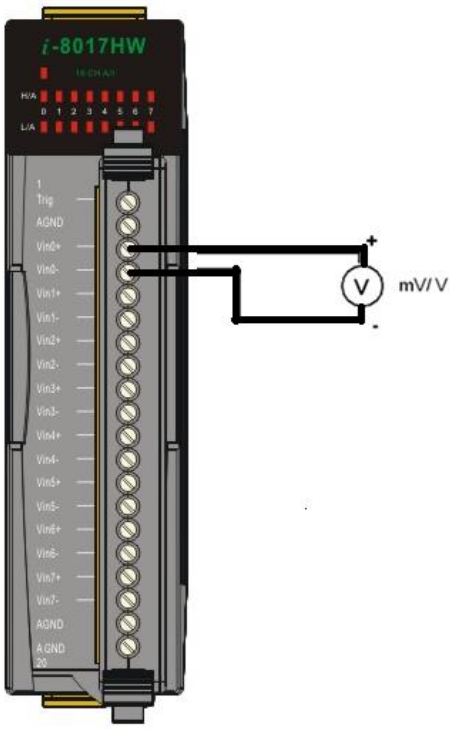
The 8017ai.exe executable file, which is located in the 8017h_ReadAI folder of the demo programs, can be used to retrieve the basic configuration information related to the module and to verify the AI read functions. The basic configuration information includes:

- The version number and the published date of the library.
- The FPGA version
- The Differential/Single-ended jumper settings
- The Gain and Offset values for each input range
- The data read from each channel

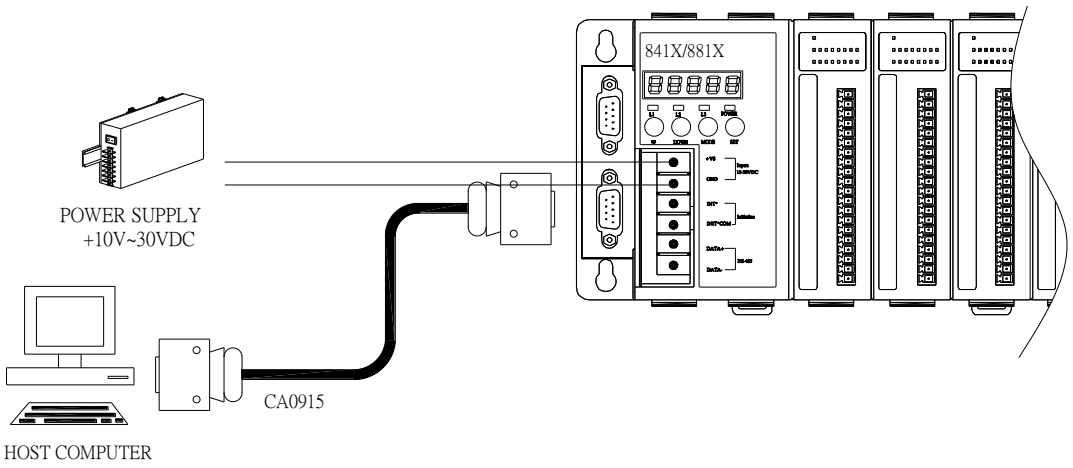
(See the Location of the Demo Programs section for details of where to find the 8017ai.exe file in the demo programs path)

Step 1. Refer to the Jumper Settings section. Ensure that the Differential/Single-ended selection jumper is in the Differential position.

Step 2. Connect a stable signal source to the module (e.g., a battery output) using the differential wiring method, as illustrated below.



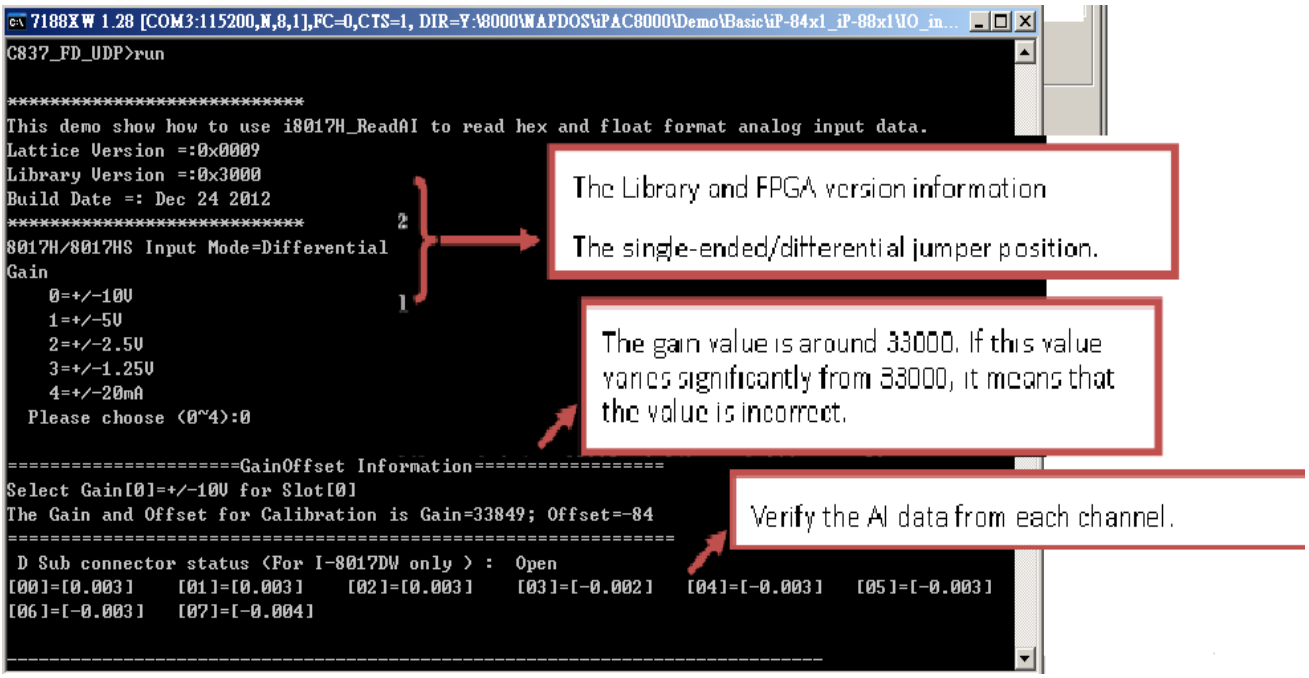
Step 3. Connect the power supply to the module, and connect the control unit to the Host PC using an RS-232 cable.



Step 4. Launch the 8017ai.exe executable file on the Host PC, and then verify that the basic information and the AI data from each channel is correct, as indicated in the diagram below:

Tips & Warnings

Unused channels should be connected to GND to avoid floating.



2.2. Windows-based Controllers

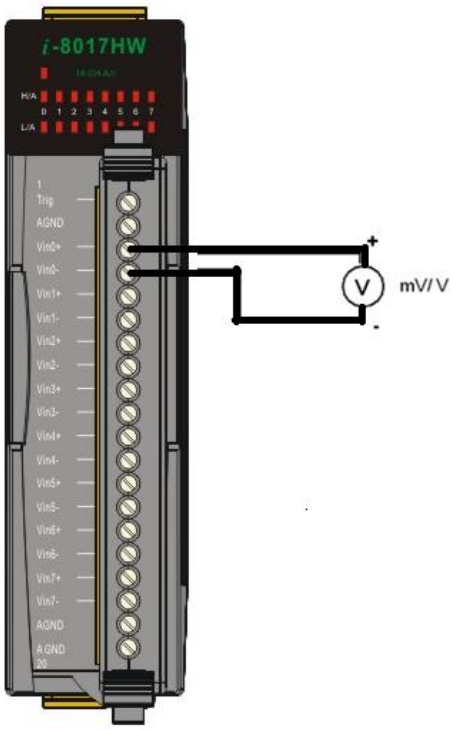
The pac_i8017HW_Utility.exe executable file, which is located in the pac_i8017HW_Utility folder of the demo programs, can be used to retrieve the basic configuration information related to the module, and to verify the AI read functions. The basic configuration information includes:

- The version number and the published date of the library.
- The FPGA version
- The Differential/Single-ended jumper settings
- The Gain and Offset values for each input range
- The data read from each channel

(See the Location of the Demo Programs section for details of where to find the 8017ai.exe file in the demo programs folder)

Step 1. Refer to the Jumper Settings section. Ensure that the Differential/Single-ended selection jumper is in the Differential position.

Step 2. Connect a stable signal source to the module (e.g., a battery output) using the differential wiring method, as illustrated below.



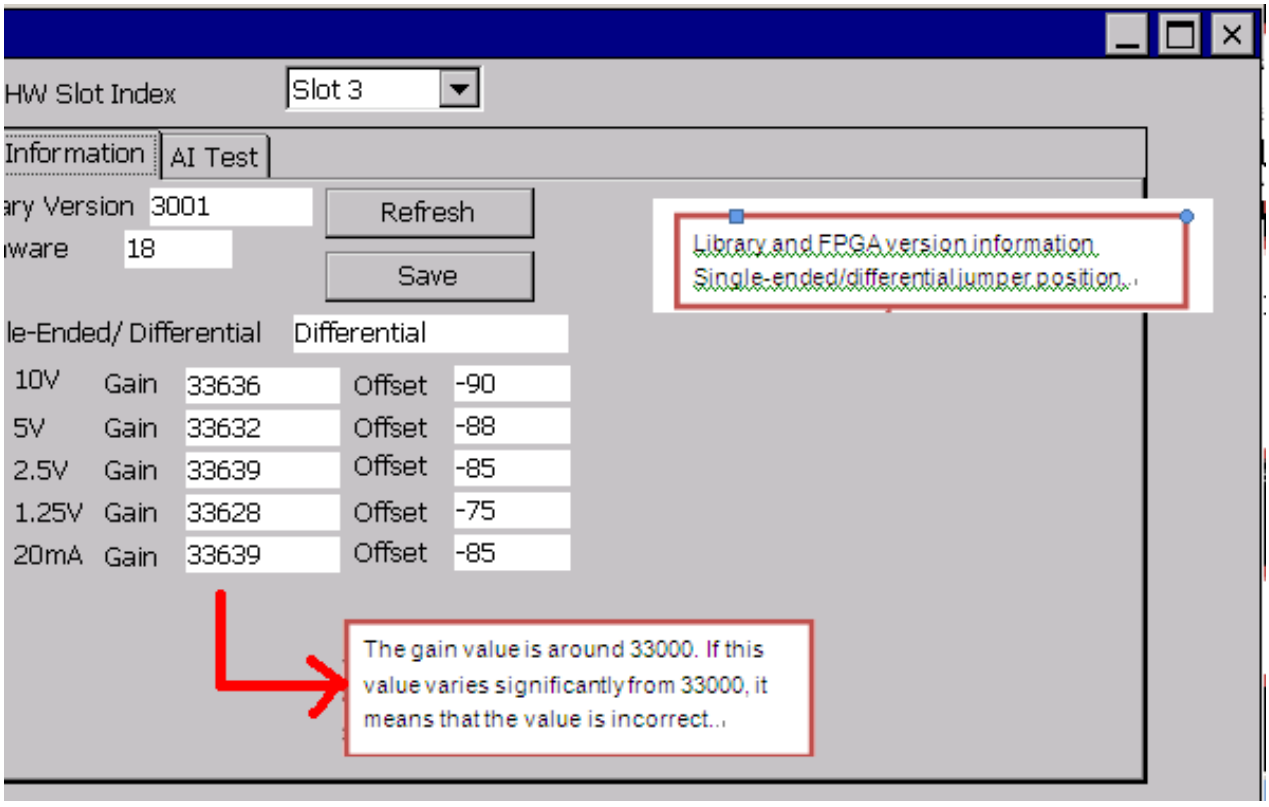
Step 3. Insert the module into a vacant slot in the control unit and power on the PAC controller

Step 4. Launch the pac_i8017HW_Utility.exe executable file on the controller, and then verify that the basic information and the AI data read from each channel is correct, as indicated in the diagram below:

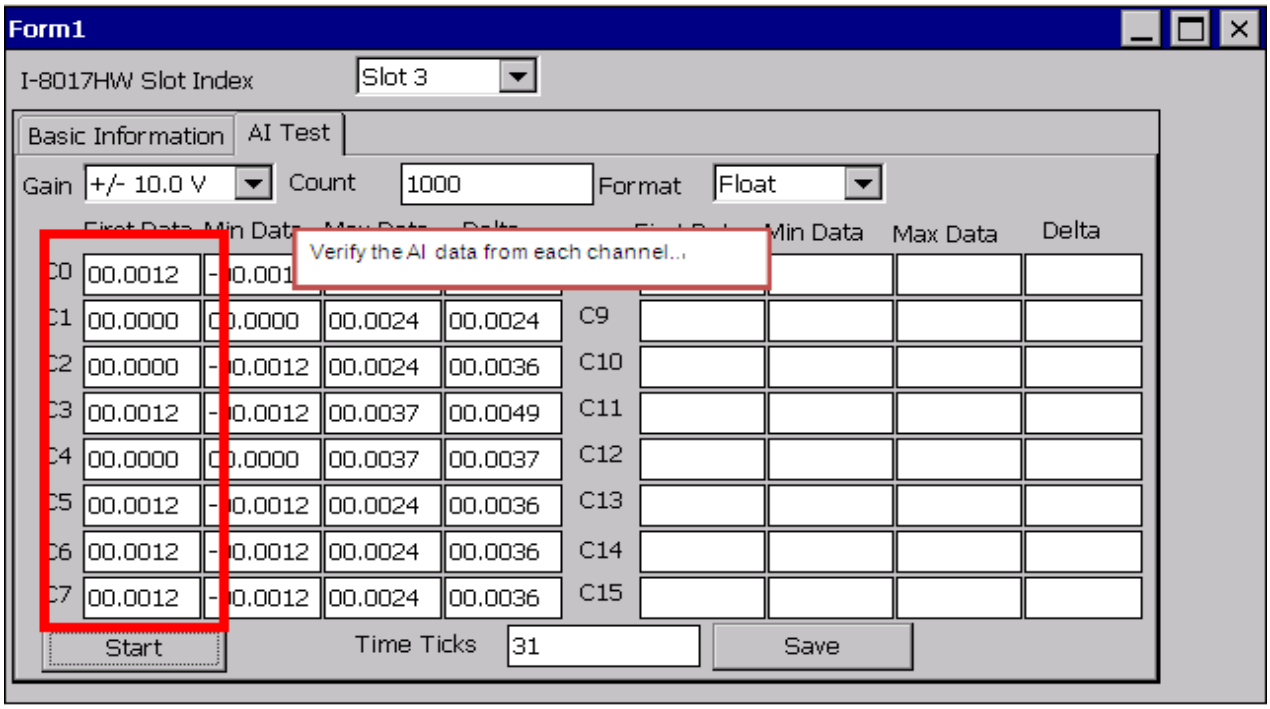
Tips & Warnings

Unused channels should be connected to GND to avoid floating, it may get some noise values..

Read the FPGA version information and the Gain and Offset values for each voltage range



Read the AI information from each channel



3. Demo Programs

ICP DAS provides a range of demo programs for different platforms that can be used to verify the functions of the module. 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 I-8000	
Library	CD:\Napdos\8000\841x881x\demo\Lib or ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/8000/841x881x/demo/lib/
Demo	CD:\Napdos\8000\841x881x\demo\IO_in_Slotorftp://ftp.icpdas.com/pub/cd/8000cd/napdos/8000/841x881x/demo/io_in_slot/
For iPAC-8000	
Library	CD:\Napdos\iPAC8000\Demo\Basic\iP-84x1_iP-88x1\Lib or ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/ipac8000/demo/basic/ip-84x1_ip-88x1/lib/
Demo	CD:\Napdos\iPAC8000\Demo\Basic\iP-84x1_iP-88x1\IO_in_Slot or ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/ipac8000/demo/basic/ip-84x1_ip-88x1/io_in_slot/
For Windows CE5	
Library	CD:\napdos\wp-8x4x_ce50\sdk\IO_Modules or ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/io_modules/
Demo	eVC Demo: CD:\napdos\wp-8x4x_ce50\Demo\WinPAC\eVC\IO\Local or ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/io/local/ C# Demo: CD:\napdos\wp-8x4x_ce50\Demo\WinPAC\C#\IO\Local or ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/c%23/io/local/

For WP-9000	
Library	CD:\WinPAC_AM335x\wp-9000\SDK\IO_Modules ftp://ftp.icpdas.com/pub/cd/winpac_am335x/wp-9000/sdk/io_modules/
Demo	VC2008 Demo: CD:\WinPAC_AM335x\wp-9000\demo\PAC\Vc2008\IO\Local ftp://ftp.icpdas.com/pub/cd/winpac_am335x/wp-9000/demo/pac/vc2008/io/local/ C# Demo: CD:\WinPAC_AM335x\wp-9000\demo\PAC\C#\IO\Local ftp://ftp.icpdas.com/pub/cd/winpac_am335x/wp-9000/demo/pac/c%23/io/local/
For XP-8000-CE6	
Library	CD:\SDK\Special_IO ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/sdk/special_io/
Demo	VC2005 Demo: CD:\Demo\XPAC\VC2005\IO\Local ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/demo/xpac/vc2005/io/local/ C# Demo: CD:\Demo\XPAC\C#\IO\Local ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/demo/xpac/c%23/io/local/
For XP-8000-Atom-CE6	
Library	CD:\SDK\Special_IO ftp://ftp.icpdas.com/pub/cd/xpac-atom-ce6/sdk/special_io/
Demo	VC2005 Demo: CD:\Demo\XPAC\VC2005\IO\Local ftp://ftp.icpdas.com/pub/cd/xpac-atom-ce6/demo/xpac/vc2005/io/local/ C# Demo: CD:\Demo\XPAC\C#\IO\Local ftp://ftp.icpdas.com/pub/cd/xpac-atom-ce6/demo/xpac/c%23/io/local/
For XP-8000-CE6	
Library	CD:\SDK\IO ftp://ftp.icpdas.com/pub/cd/xp-8000/sdk/io/
Demo	VC2005 Demo: CD:\Demo \XPAC \VC2005\IO\Local ftp://ftp.icpdas.com/pub/cd/xp-8000/demo/pacsdk/vc/io/local/ CD:\Demo \XPAC\C#\IO\Local ftp://ftp.icpdas.com/pub/cd/xp-8000/demo/pacsdk/csharp.net/io/local/windows_forms/

For XP-8000-Atom	
Library	CD:\SDK\IO ftp://ftp.icpdas.com/pub/cd/xpac-atom/sdk/io/
Demo	VC Demo: CD:\Demo \pacsdk \vc\IO\Local ftp://ftp.icpdas.com/pub/cd/xpac-atom/demo/pacsdk/vc/io/local/ C# Demo: CD:\Demo \pacsdk\csharp.net\IO\Local\windows_forms ftp://ftp.icpdas.com/pub/cd/xpac-atom/demo/pacsdk/csharp.net/io/local/windows_forms/
For ippc-WES7	
Library	CD:\ippc-wes7\sdk\IO ftp://ftp.icpdas.com/pub/cd/ippc-wes7/sdk/io/
Demo io-8k	VC Demo: CD:\ippc-wes7\demo\pacsdk\vc\io\local\io-8k ftp://ftp.icpdas.com/pub/cd/ippc-wes7/demo/pacsdk/vc/io/local/io-8k/ C# Demo: CD:\ippc-wes7\demo\pacsdk\csharp.net\io\local\io-8k ftp://ftp.icpdas.com/pub/cd/ippc-wes7/demo/pacsdk/csharp.net/io/local/io-8k/
Demo io-9k	VC Demo: CD:\ippc-wes7\demo\pacsdk\vc\io\local\io-9k ftp://ftp.icpdas.com/pub/cd/ippc-wes7/demo/pacsdk/vc/io/local/io-9k/ C# Demo: CD:\ippc-wes7\demo\pacsdk\csharp.net\io\local\io-9k ftp://ftp.icpdas.com/pub/cd/ippc-wes7/demo/pacsdk/csharp.net/io/local/io-9k/

4. API References

API naming table

The following table describes the platforms and in which the product series included and the different part of function name.

Platform	Product included	API prefix characters
Windows CE5 Windows CE6	WP-8000 series XP-8000-CE6 series XP-8000-Atom-CE6 series	“pac_i8017W_ “+ function name
Windows CE7	WP-8000 series WP-9000-CE7 series	“pac_i8017W_ “+ function name
WES	XP-8000 series XP-8000-Atom series	“pac_i8017W_ “+ function name
WES7	XP-8000-WES7 series XP-9000 series	“pac_i8017W_ “+ function name
MiniOS7	I-8000 series iPAC-8000 series	“i8017W_” + function name
Linux	LinPAC-8000series LinPAC-9000series	“i8017W_” + function name

The following is an overview of the functions provided in the 8017HW.lib and pac_i8017HW.lib. Detailed information related to individual functions can be found in the following sections.

API for I-8017HW seriesand I-9017 series

Function for MiniOS7	Function for Windows	Description
i8017HW_Init	pac_i8017HW_Init	Used to initialize the module
i8017HW_GetFirmwareVersion	pac_i8017HW_GetFirmwareVersion	Usedto read the firmware (FPGA) version information
i8017HW_GetLibVersion	pac_i8017HW_GetLibVersion	Used to read the version and build information for the currently installed Library
i8017HW_GetLibDate	pac_i8017HW_GetLibDate	Used to read the build date information for the currently installed Library

i8017HW_GetSingleEndJumper	pac_i8017HW_GetSingleEndJumper	Used to read the status of the input jumper (Differential or Single-ended mode)
i8017HW_ReadAI	pac_i8017HW_ReadAI	Used to read the Analog Input value from a specific channel in float format
i8017HW_ReadAI_AVG	pac_i8017HW_ReadAI_AVG	Used to read the average Analog input value from a specific channel in float format
i8017HW_ReadAIHex	pac_i8017HW_ReadAIHex	Used to read the Analog Input value from a specific channel in 16-bit hexadecimal format
i8017HW_ReadAIHex_AVG	pac_i8017HW_ReadAIHex_AVG	Used to read the average Analog input value from a specific channel in hexadecimal format
i8017HW_ReadGainOffset_Info	pac_i8017HW_ReadGainOffset_Info	Used to read the calibrated voltage Gain and Offset values
i8017HW_Read_mA_GainOffset	pac_i8017HW_Read_mA_GainOffset	Used to read the calibrated currents Gain and Offset values

API for I-9017-15

Function for MiniOS7	Function for Windows	Description
N/A	pac_i8017HW_Select_SingleEnd	Used to set the Single-ended/ differential mode of I-9017-15.

API for I-8017DW

Function for MiniOS7	Function for Windows	Description
i8017HW_Get_D_Sub_Status	pac_i8017HW_Get_D_Sub_Status	Used to get connector status between D sub and 8017DW.

4.1. pac_i8017HW_Init

This function is used to initialize the module and must be called at least once before using any other function.

Syntax

For MiniOS7

```
shorti8017HW_Init(
   intslot
);
```

For Windows (CE and WES)

```
short pac_i8017HW_Init(
   int slot
);
```

Parameters

slot:

specifies the slot number (0 - 7).

Return Values

Refer to Appendix A: “Error Code” for more details.

Examples

[C/C++]

```
int slot;
i8017HW_Init(slot);
```

[C#]

```
int slot;
pac8017HW.Init(slot);
```

4.2. pac_i8017HW_GetFirmwareVersion

This function is used to read the firmware (FPGA) version information for the module.

Syntax

For MiniOS7

```
shorti8017HW_GetFirmwareVersion(
    intslot,
    short* firmware
);
```

For Windows (CE and WES)

```
short pac_i8017HW_GetFirmwareVersion(
    intslot,
    short* firmware
);
```

Parameters

slot

specifies the slot number (0 - 7).

**firmware*

[Output]The firmware version information for the I-8017 module.

Return Values

Refer to Appendix A: “Error Code” for more details.

Examples

[C/C++]

```
int slot;
short firmware;
i8017HW_GetFirmwareVersion(slot, &firmware);
```

[C#]

```
int slot;
Int16 firmware = 0;
pac8017HWNet.pac8017HW.FirmwareVersion(slot, ref firmware);
```

4.3. pac_i8017HW_GetLibVersion

This function is used to read the version and build information for the Library.

Syntax

For MiniOS7

```
short i8017HW_GetLibVersion(void);
```

For Windows (CE and WES)

```
short pac_i8017HW_GetLibVersion(void);
```

Parameters

None

Return Values

The version number and build information for the Library used by the module.

Others: Refer to Appendix A: “Error Code Definitions” for more details.

Examples

[C/C++]

```
short version;
version = i8017HW_GetLibVersion();
```

[C#]]

```
Int16 version;
version = pac8017HWNet.pac8017HW.LibVersion();
```

4.4. pac_i8017HW_GetLibDate

This function is used to read the build date information for the Library.

Syntax

For MiniOS7

```
voidi8017HW_GetLibDate(
   charlibDate[]
);
```

For Windows (CE and WES)

```
void pac_i8017HW_GetLibDate(
   charlibDate[]
);
```

Parameters

libDate[]

A string indicating the build date of the Library.

Return Values

Refer to Appendix A: “Error Code” for more details.

Examples

[C/C++]

```
chardate;
i8017HW_GetLibDate(date);
```

[C#]

```
string date;
date= pac8017HWNet.pac8017HW.LibDate();
```

4.5. pac_i8017HW_GetSingleEndJumper

This function is used to read whether the jumper is set to either Differential or Single-ended mode.

Syntax

For MiniOS7

```
short pac_i8017HW_GetSingleEndJumper(
    intiSlot,
    short* selectJumper
);
```

For Windows (CE and WES)

```
short pac_i8017HW_GetSingleEndJumper(
    intiSlot,
    short* selectJumper
);
```

Parameters

iSlot

specifies the slot number (0 - 7).

**selectJumper*

[Output]The status of module.

0: Differential Mode

1: Single-ended Mode

Return Values

Refer to Appendix A: “Error Code” for more details.

Examples

[C/C++]

```
intslot,jumper;
i8017HW_GetSingleEndJumper(slot,&jumper);
```

[C#]

```
intslot,jumper;
pac8017HWNet.pac8017HW.SingleEndJumper(slot, ref jumper);
```

4.6. pac_i8017HW_ReadAI

This function is used to read the Analog Input value in float format from a specific channel of the module.

Syntax

For MiniOS7

```
shorti8017HW_ReadAI(
    intiSlot,
    intiChannel,
    intiGain,
    float* fValue
);
```

For Windows (CE and WES)

```
short pac_i8017HW_ReadAI(
    intiSlot,
    intiChannel,
    intiGain,
    float* fValue
);
```

Parameters

iSlot

specifies the slot number (0 - 7).

iChannel

Specifies the channel number

iGain

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

**fValue*

[Output]the analog input value in float format.

Return Values

Others: Refer to Appendix A: “Error Code” for more details.

Examples

[C++]

```
int slot, ch, gain;
floatfValue;
pac_i8017HW_ReadAI(slot, ch, gain,&fValue);
```

[C#]

```
int slot, ch, gain;
floatfValue;
pac8017HWNet.pac8017HW.ReadAI(slot, ch, gain, ref fValue);
```

4.7. pac_i8017HW_ReadAI_AVG

This function is used to read the average Analog Input value in float format from themodule.

Syntax

For MiniOS7

```
shorti8017HW_ReadAI_AVG(
    int slot,
    intiChannel,
    intiGain,
    unsigned short averageCnt,
    float* fValue
);
```

For Windows (CE and WES)

```
short pac_i8017HW_ReadAI_AVG(
    int slot,
    intiChannel,
    intiGain,
    unsigned short averageCnt,
    float* fValue
);
```

Parameters

Slot

specifies the slot number (0 - 7).

iChannel

Specifies the channel number

iGain

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

averageCnt

the average count for each sampling routine.

**fValue*

[Output]the analog input value in float format.

Return Values

Others: Refer to Appendix A: “Error Code” for more details.

Examples

[C++]

```
int slot, ch, gain;
unsigned shortcnt;
floatfValue;
pac_i8017HW_ReadAI_AVG(slot, ch, gain,cnt,&fValue);
```

[C#]

```
int slot, ch, gain;
Uintcnt;
floatfValue;
pac8017HWNet.pac8017HW.ReadAI_AVG(slot, ch, gain, cnt, ref fValue);
```

4.8. pac_i8017HW_ReadAIHex

This function is used to read the Analog Input value in 16-bit hexadecimal format.

Syntax

For MiniOS7

```
short i8017HW_ReadAIHex(
    intiSlot,
    intiChannel,
    intiGain,
    short* iValue
);
```

For Windows (CE and WES)

```
short pac_i8017HW_ReadAIHex(
    intiSlot,
    intiChannel,
    intiGain,
    short* iValue
);
```

Parameters

iSlot

specifies the slot number (0 - 7).

iChannel

Specifies the channel number

iGain

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

**iValue*

[Output]the analog input value in hexadecimal format.

Return Values

Others: Refer to Appendix A: “Error Code” for more details.

Examples

[C++]

```
int slot, ch, gain;
shorthval;
pac_i8017HW_ReadAIHex(slot, ch, gain,&hval);
```

[C#]

```
int slot, ch, gain;
inthval;
pac8017HWNet.pac8017HW.ReadAIHex(slot, ch, gain, ref hval);
```

Note

the I-8017HW/I-8017DW/I-8017HCW/I-9017/I-9017-15/I-9017C-15 modules use a 14-bit AD chip,when user needs to scale the hexadecimal data ,it is convenient to use the pac_i8017HW_ReadHex function to return a 16-bit data.

4.9. pac_i8017HW_ReadAIHex_AVG

This function is used to read the average Analog Input value in 16-bit hexadecimal format.

Syntax

For MiniOS7

```
short i8017HW_ReadAIHex_AVG(
    int slot,
    intiChannel,
    intiGain,
    unsigned short averageCnt,
    short* iValue
);
```

For Windows (CE and WES)

```
short pac_i8017HW_ReadAIHex_AVG(
    int slot,
    intiChannel,
    intiGain,
    unsigned short averageCnt,
    short* iValue
);
```

Parameters

slot

specifies the slot number (0 - 7).

iChannel

Specifies the channel number

iGain

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

averageCnt

the average count for each sampling routine.

**iValue*

[Output]the analog input value in hexadecimal format.

Return Values

Others: Refer to Appendix A: “Error Code” for more details.

Examples

[C++]

```
int slot, ch, gain;
unsigned shortcnt;
shorthval;
pac_i8017HW_ReadAIHex_AVG(slot, ch, gain,cnt,&hval);
```

[C#]

```
int slot, ch, gain;
uintcnt;
inthval;
pac8017HWNet.pac8017HW.ReadAIHex_AVG(slot, ch, gain, cnt, ref hval);
```

4.10. pac_i8017HW_ReadGainOffset_Info

This function is used to read the calibrated Gain and Offset values for the I-8017 module inserted in a specific slot

Syntax

For MiniOS7

```
short i8017HW_ReadGainOffset_Info(
    intiSlot,
    intiGain,
    unsigned short* iGainValue,
    short* iOffsetValue
);
```

For Windows (CE and WES)

```
short pac_i8017HW_ReadGainOffset_Info(
    intiSlot,
    intiGain,
    unsigned short* iGainValue,
    short* iOffsetValue
);
```

Parameters

iSlot

specifies the slot number (0 - 7).

iGain

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

**iGainValue*

[Output]Specifies the calibrated Gain value

**iOffsetValue*

[Output]Specifies the calibrated Offset value

Return Values

Refer to Appendix A: “Error Code” for more details.

Examples

[C++]

```
Intslot,Gain;
unsigned short GainValue;
shortOffsetValue;
short pac_i8017HW_ReadGainOffset_Info(slot,Gain,&GainValue,&OffsetValue);
```

[C#]

```
Intslot,Gain;
unsigned short GainValue;
shortOffsetValue;
pac8017HWNet.pac8017HW.GainOffset_Info(slot, Gain, ref GainValue, ref OffsetValue);
```

4.11. pac_i8017HW_Read_mA_GainOffset

This function is used to read the calibrated Gain and Offset values for the I-8017HCW/I-9017/I-9017C-15 module inserted in a specific slot.

Syntax

For MiniOS7

```
short i8017H_Read_mA_GainOffset(
    int slot,
    shortch,
    unsigned short* GainValue,
    short* offsetValue
);
```

For Windows (CE and WES)

```
shortpac_i8017H_Read_mA_GainOffset(
    int slot,
    shortch,
    unsigned short* GainValue,
    short* offsetValue
);
```

Parameters

slot

specifies the slot number (0 - 7).

ch

Specifies the channel

Valid range :

I-8017HCW/I-9017 = 0 to 7

I-9017C-15 = 0 to 14

**GainValue*

Specifies the calibrated Gain value

**offsetValue*

Specifies the calibrated Offset value

Return Values

Refer to Appendix A: “Error Code” for more details.

Examples

[C/C++]

```
int slot;
shortch;
unsigned short GainValue;
shortOffsetValue;
i8017H_Read_mA_GainOffset(slot,ch,&GainValue, &OffsetValue);
```

[C#]

```
Int slot;
Int16 ch;
UInt16 GainValue;
Int16 OffsetValue;
pac8017HWNet.pac8017HW.Ch_mAGainOffset(slot,ch,refGainValue,refOffsetValue);
```

4.12. pac_i8017HW_Select_SingleEnd

This function is used to set the Single-ended/differential mode of I-9017-15.

Syntax

For MiniOS7

```
short i8017HW_Select_SingleEnd (
    int slot,
    short selection
);
```

For Windows (CE and WES)

```
shortpac_i8017HW_Select_SingleEnd (
    int slot,
    short selection
);
```

Parameters

slot

specifies the slot number (0 - 7).

selection

The status of module.

0: Differential Mode

1: Single-ended Mode

Return Values

Refer to Appendix A: “Error Code” for more details.

Examples

[C/C++]

```
int slot;
shortstatus;
pac_i8017H_Select_SingleEnd(slot,status);
```

[C#]

```
Int slot;
Int16 selection;
pac8017HWNet.pac8017HW.Select_SingleEnd_Differential (slot,selection);
```

4.13. pac_i8017HW_Get_D_Sub_Status

This function is usedto get connector status between D sub and 8017DW.

Syntax

For MiniOS7

```
short i8017HW_Get_D_Sub_Status(
   intiSlot,
   short* D_Sub_Status
);
```

For Windows (CE and WES)

```
short pac_i8017HW_Get_D_Sub_Status(
   intiSlot,
   short* D_Sub_Status
);
```

Parameters

slot

specifies the slot number (0 - 7).

**D_Sub_Status*

The status of D_Sub.

1 : Open

0 :Close

Return Values

Refer to Appendix A: “Error Code” for more details.

Note

function for I-8017DW module only, in the others 8017 series module,

the value of D Sub Status will always be 1.

Examples

[C/C++]

```
int slot;
short status;
pac_i8017HW_Get_D_Sub_Status(slot,status);
```

[C#]

```
Int slot;
Int16 selection;
pac8017HWNet.pac8017HW.D_Sub_Status(slot,selection);
```

5. Calibration

Each I-8017 module is factory calibrated and thoroughly tested and 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.
To calibrate the module, in addition to inserting the module into a controller slot, the following items are required:

- 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 the Location of the Calibration Demo Programs section below for details.

Tips & Warnings

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

2. If you wish to perform calibration using ± 20 mA, select ± 2.5 V instead as both types use the same Gain and Offset values.

3. The calibration program only uses channel 0 for input of the calibration source.

Select the appropriate usage platform to locate the relevant calibration demo program for the I-8017HW, I-8017DW and I-8017HCW modules.

Platform	Location
For I-8000	
CD:\Napdos\8000\841x881x\demo\IO_in_Slot\8017h\Calibration ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/8000/841x881x/demo/io_in_slot/8017h/calibration/	
For iPAC-8000	
CD:\Napdos\iPAC8000\Demo\Basic\iP-84x1_iP-88x1\IO_in_Slot\8017h\Calibration ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/ipac8000/demo/basic/ip-84x1_ip-88x1/io_in_slot/8017h/calibration/	

For Windows CE5
(C# demo) CD:\napdos\wp-8x4x_ce50\Demo\WinPAC\DOTNET\C#.NET\PAC_IO\Local\pac_i8017HW_Dotnet\pac_i8017HW_calibration ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/c%23/io/local/pac_i8017hw_dotnet/pac_i8017hw_calibration/
For XP-8000-CE6
CD:\Demo\XPAC\C#\IO\Local\pac_i8017HW_Dotnet\pac_i8017HW_calibration ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/demo/xpac/c%23/io/local/pac_i8017hw_dotnet/pac_i8017hw_calibration/
For XP-8000-Atom-CE6
CD:\Demo\XPAC\C#\IO\Local\pac_i8017HW_Dotnet\pac_i8017HW_calibration ftp://ftp.icpdas.com/pub/cd/xpac-atom-ce6/demo/xpac/c%23/io/local/pac_i8017hw_dotnet/pac_i8017hw_calibration/
For XP-8000
CD:\Demo\XPAC\csharp.net\IO\Local\windows_forms\pac_i8017HW_Dotnet\pac_i8017HW_calibration ftp://ftp.icpdas.com/pub/cd/xp-8000/demo/xpac/csharp.net/io/local/windows_forms/pac_i8017hw_dotnet/pac_i8017hw_calibration/
For XP-8000-Atom
CD:\Demo\XPAC\csharp.net\IO\Local\windows_forms\pac_i8017HW_Dotnet\pac_i8017HW_calibration ftp://ftp.icpdas.com/pub/cd/xpac-atom/demo/xpac/csharp.net/io/local/windows_forms/pac_i8017hw_dotnet/pac_i8017hw_calibration/

5.1. MiniOS7-based Controller

This section contains:

- Calibrating the modules on i-8000 and iPAC-8000 Units
- Verifying the Calibration
- Restoring the Default Calibration Settings

5.1.1. Calibrating on i-8000 and iPAC-8000

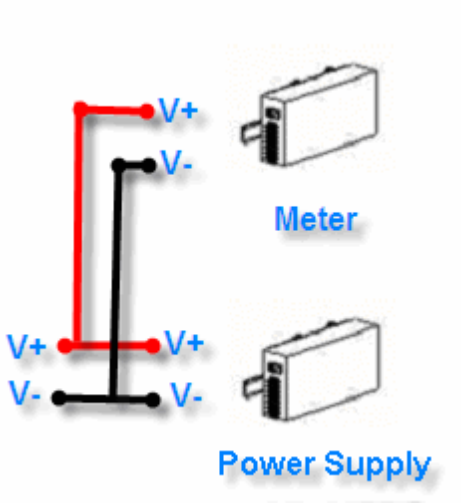
Step 1. Repeat Steps 1 to 3 as described in the Quick Start guide.

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

b. Set the Differential/Single-ended input jumper to the Differential position and connect the calibration source to channel 0 using the differential wiring method.

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

d. Turn on the control unit.



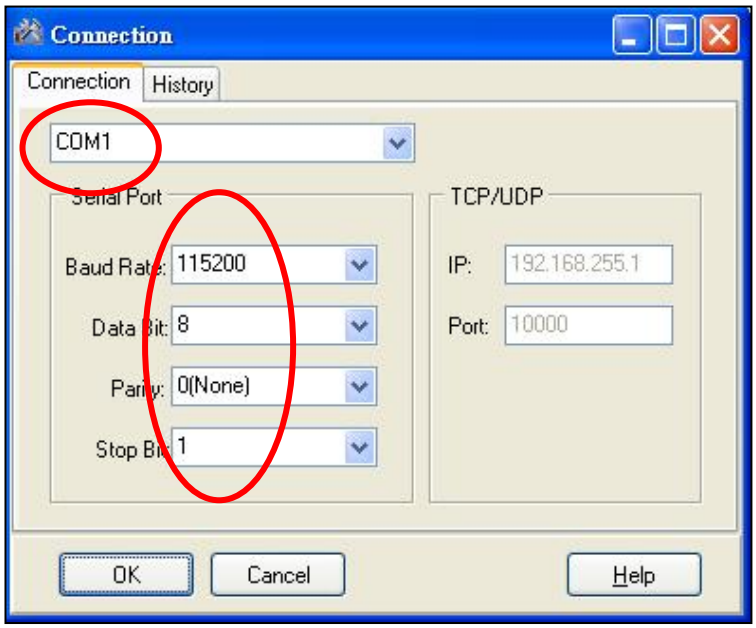
Step 2. Launch the MiniOS7 Utility on the Host PC. Upload the calibration program to the control unit and execute it.

The MiniOS7 Utility can be downloaded from the web site shown below. Select the appropriate calibration program for your controller.

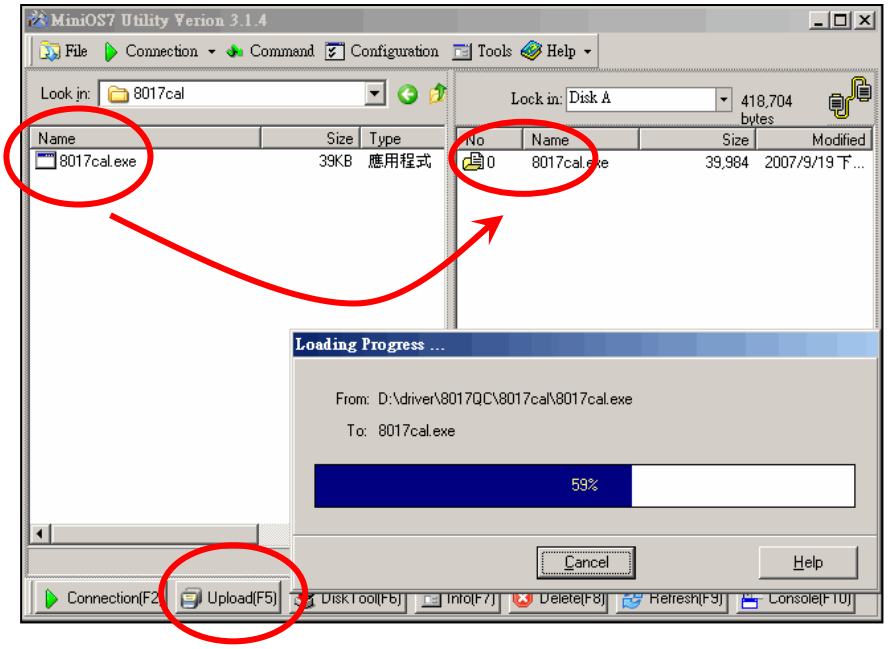
- MiniOS7 Utility: http://www.icpdas.com/download/minios7.htm
- 8017cal.exe: This is the calibration program for I-8000 units, which is located in the same folder as the demo programs. (See the Location of the Demo Programs section)
- iP_8017cal.exe: This is the calibration program for iP-8000 units, which is located in the same folder as the demo programs. (See the Location of the Demo Programs section)

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

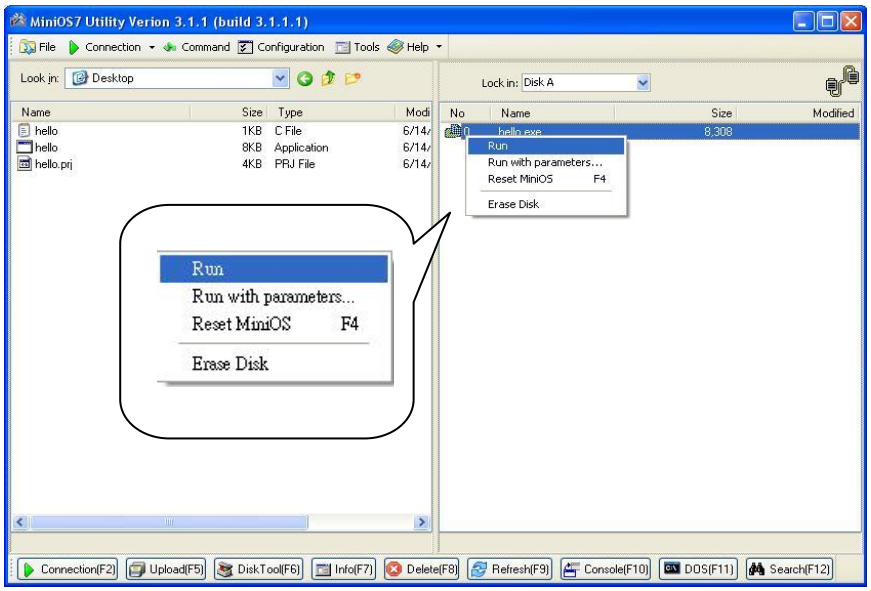
b. From the drop-down list, select the COM Port for the Host PC that is connected to the control unit, configure the communication parameters to match those indicated below, and then click the OK button.



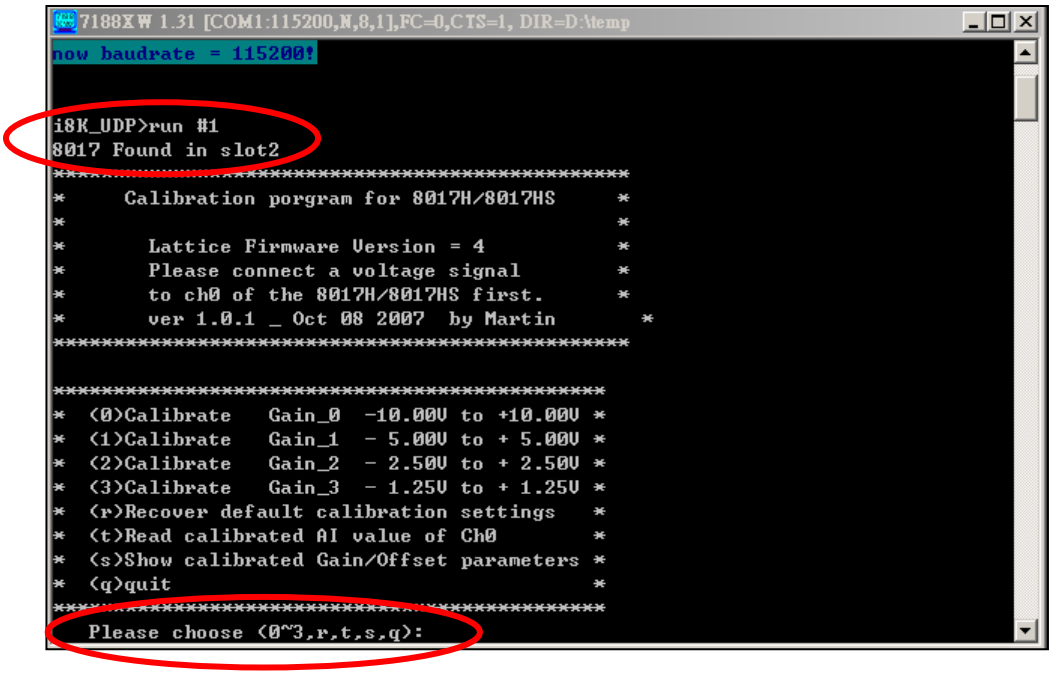
c. Select the name of the calibration program and then click the Upload button (or press F5) to upload the program to the MiniOS7 PAC unit.



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

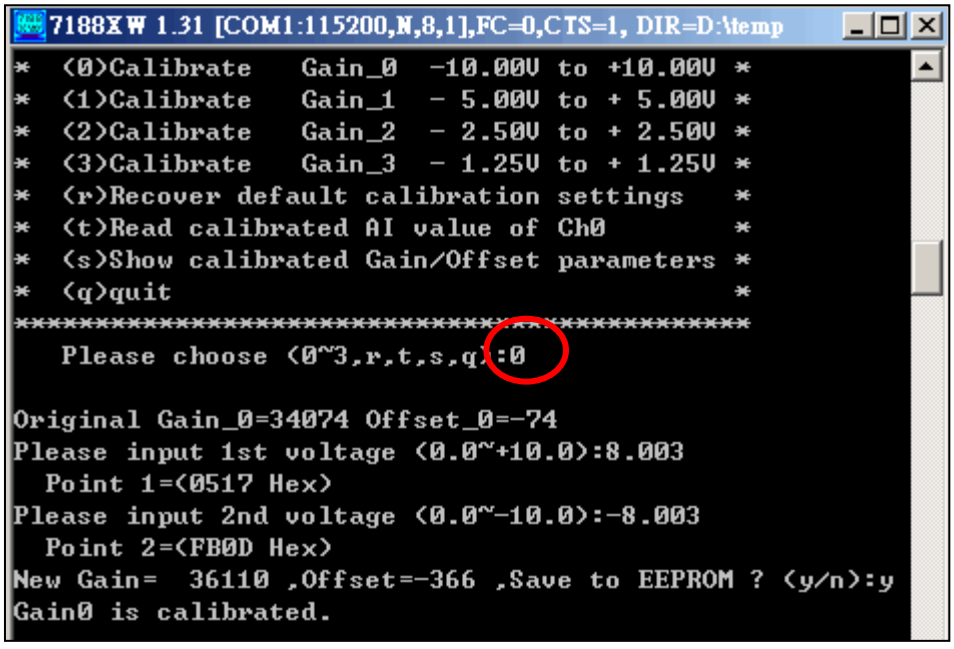


The calibration program will be executed on the control unit and 7188xw.exe will be executed on the Host PC to provide a command line interface.



Step 3. Calibrate the module using the following procedure.

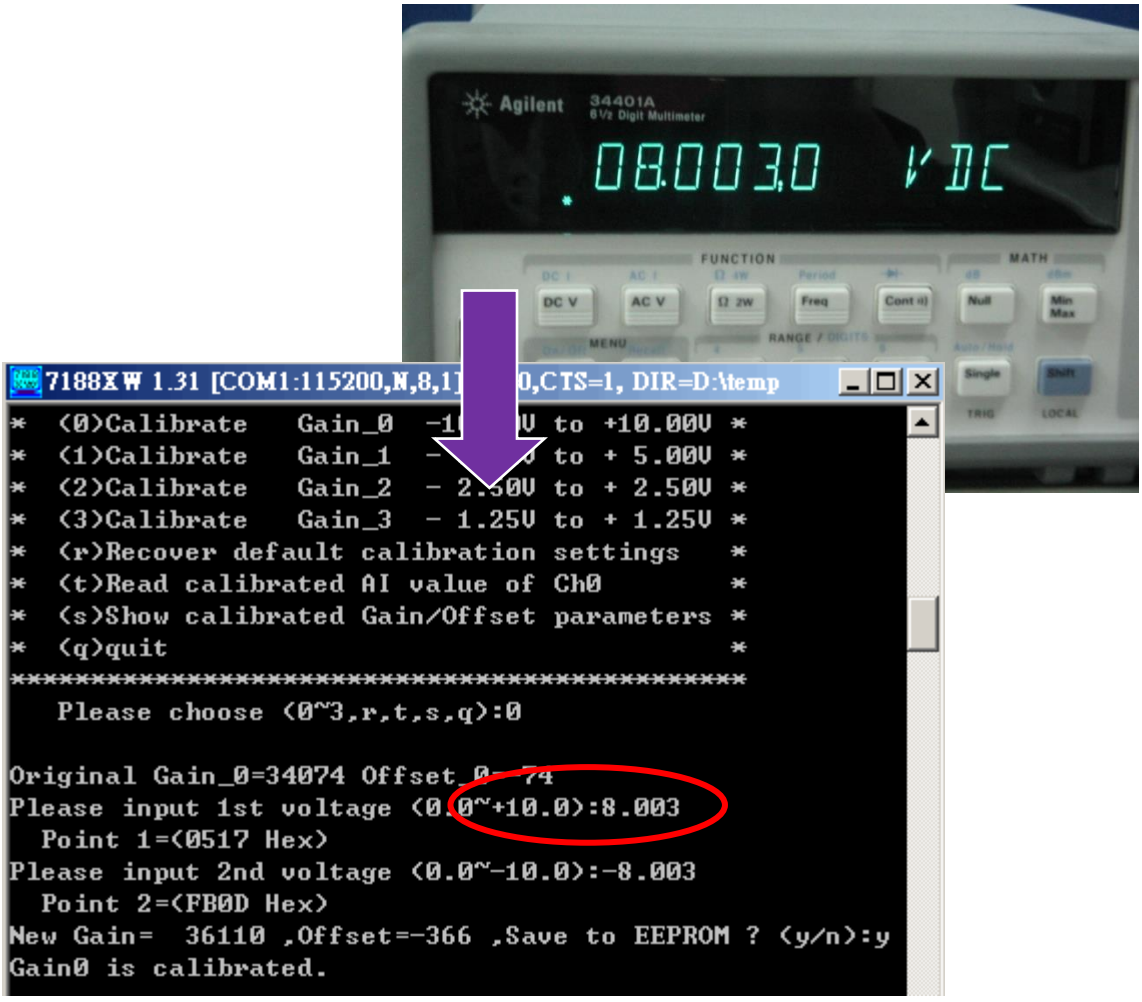
a. Select the required input type by typing an option from 0 to 3, and then press Enter.



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

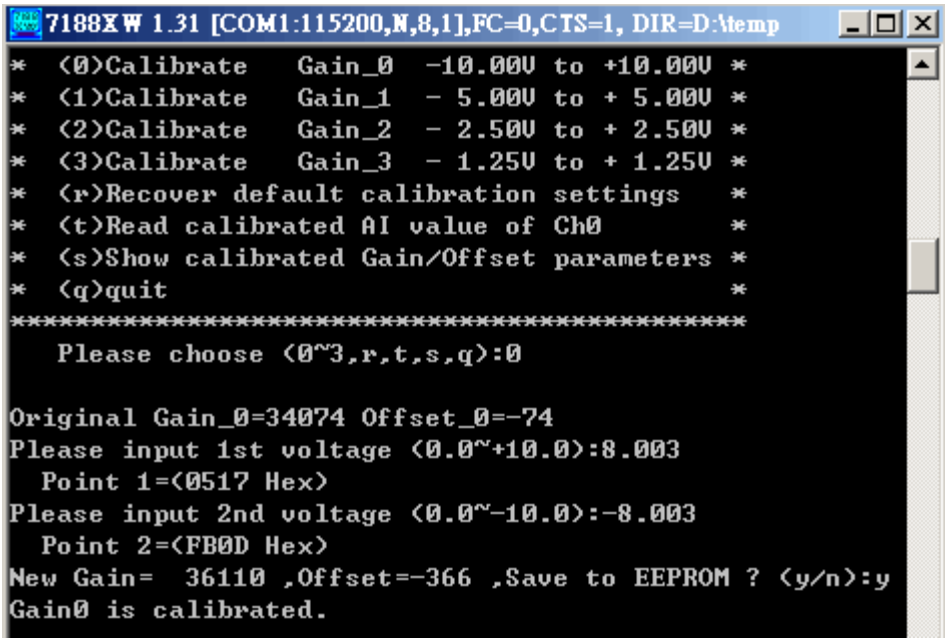
c. Set the calibration source output to one of the two points (e.g., 8 V in this example)

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



e. Set the calibration source output to the second point (e.g., - 8 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.

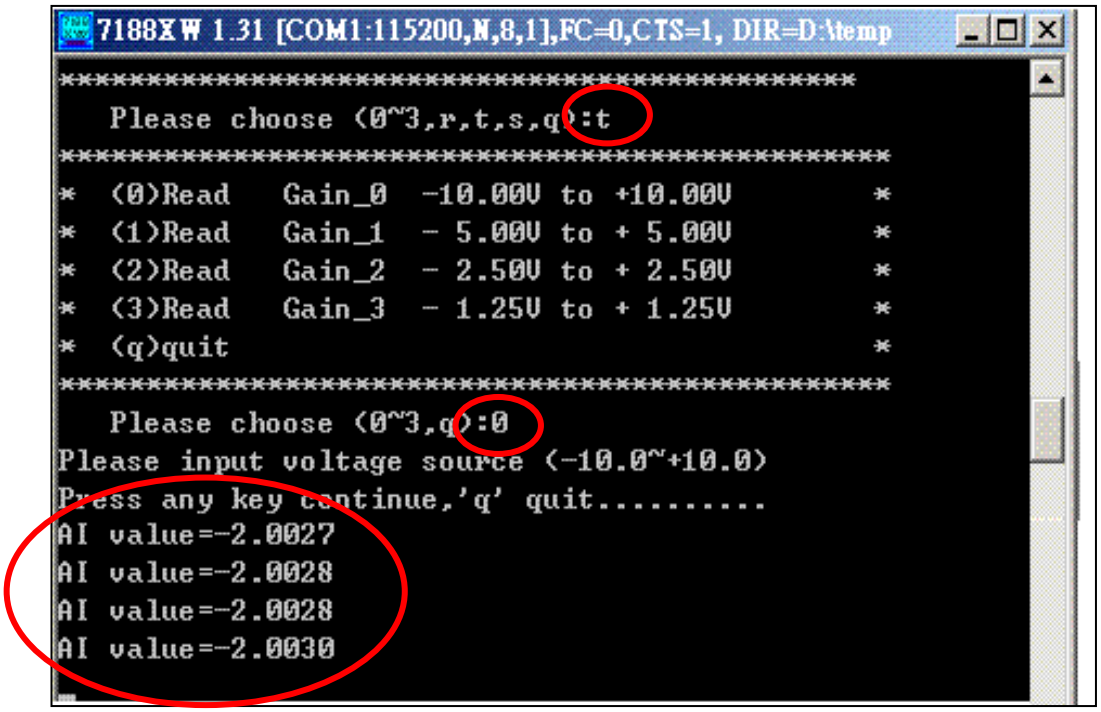
The calibration for the -10 V to +10 V input range is now complete.

5.1.2. Verifying the Calibration

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

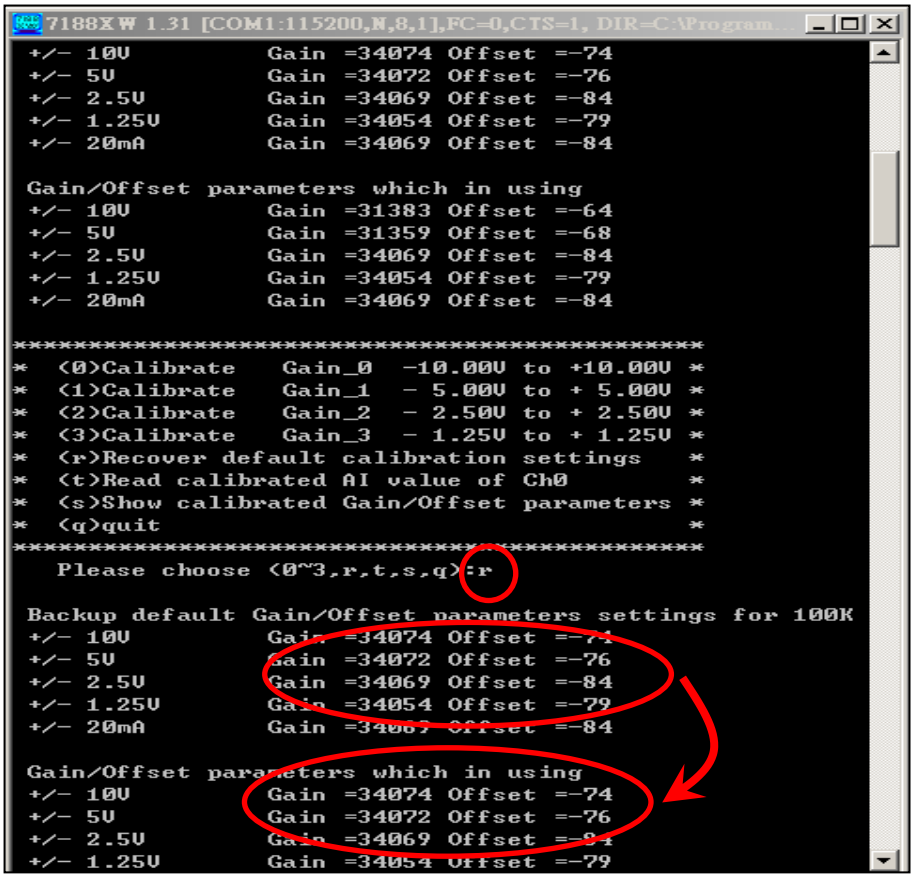
Step 2. In the same calibration program console window, type t (Read the calibrated AI value for channel 0), and then select the input type that was just calibrated (e.g., 0, -10 V to +10 V).

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



5.1.3. Restoring the Default Calibration Settings

When using the default input impedance of 200 kΩ, 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.



5.2. Windows-based Controllers

Each 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.

To calibrate it, in addition to inserting the module into a controller slot, the following items are required:

- A single stable calibration source, such as a 3 1/2 digit power supplier (or better), or a battery output.
- A single 4 1/2 digit voltage meter (15-bit resolution or better)
- A Calibration Program. See Location of the Demo Programs section for the contained in the demo programs folder.

Tips & Warnings

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

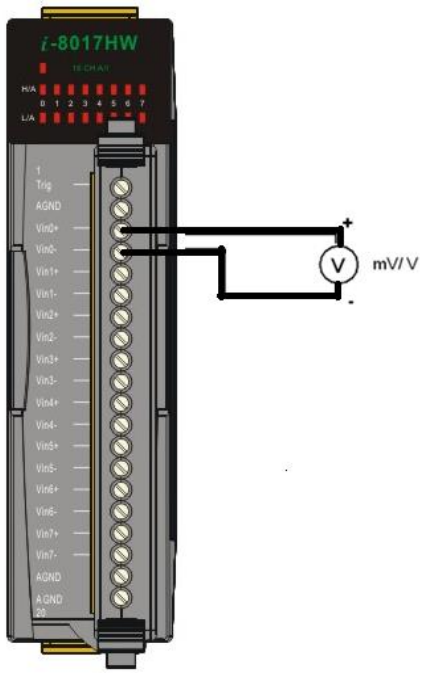
2. If you wish to perform calibration using ±20 mA, select ±2.5 V instead as both types use the same gain and offset values.

3. The calibration program uses channel 0 to accept the calibration source only.

This section contains:

- Calibrating the module on WinCE and WES PAC Units
- Verifying the Calibration
- Restoring the Default Calibration Settings

5.2.1. Calibrating on WinCE and WES PAC Units



Step 1. Refer to the Jumper Settings section. Ensure that the Differential/Single-ended input selection jumper is in the Differential position.

Step 2. Connect the calibration source to channel 0 of the module using the differential wiring method, as illustrated.

Step 3. Insert the module into a vacant slot on the controller and power on the controller.

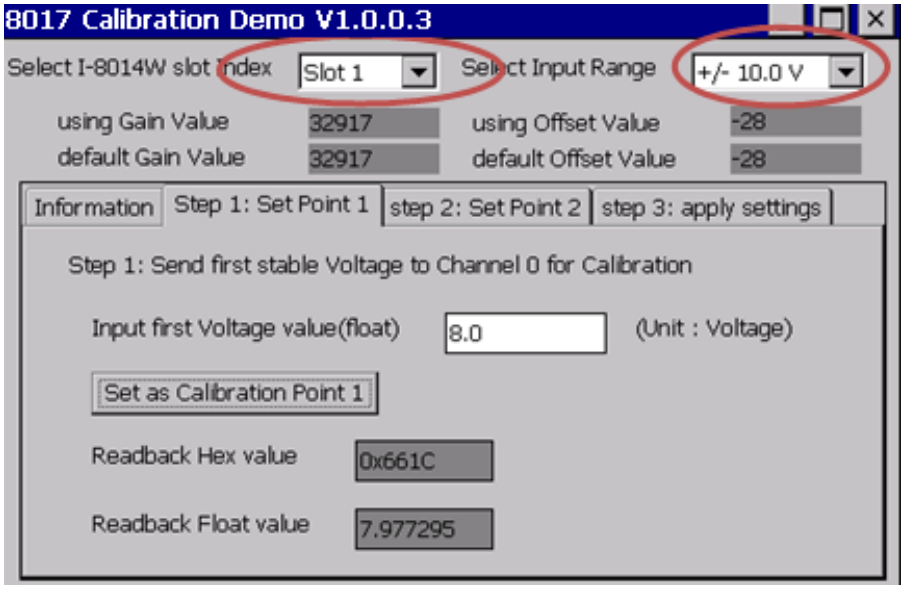
Step 4. Launch the pac_i8017W_Calibration.exe executable file on the controller to display the Calibration dialog box.

(See the Location of the Demo Programs section for details of where to find the c# demo programs for the module)

Tips & Warnings

Only channel 0 can be used to perform calibration.

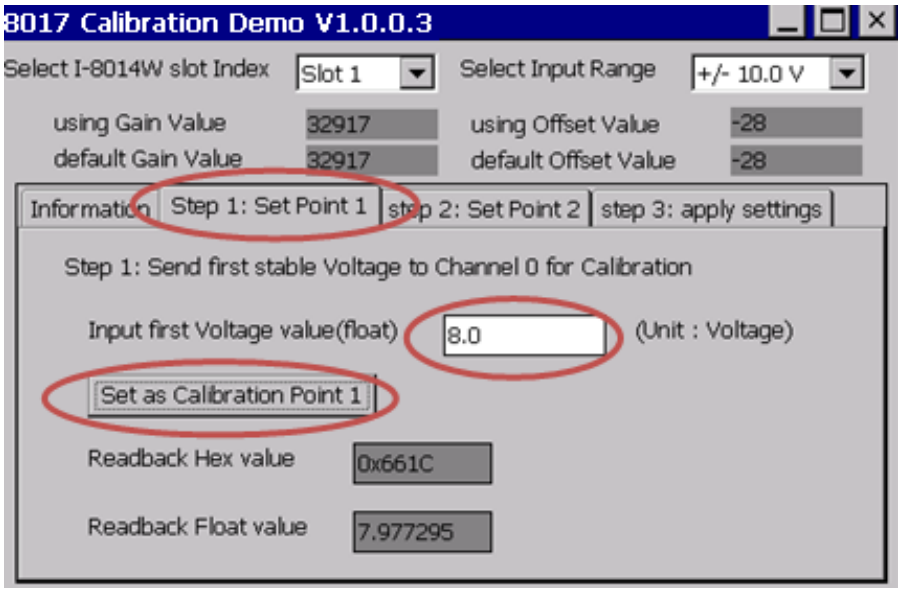
Step 5. In the upper section of the Calibration dialog box, select the I-8014W slot number and the input range from the respective drop-down lists.



Step 6. Determine two values (points) within the range of the input type selected for the calibration process. For example, after selecting -10 V to +10 V as the input range, +8 V and -8 V can be used as the two calibration points:

Step 7. Set the calibration source output to one of the two points (e.g., 8 V)

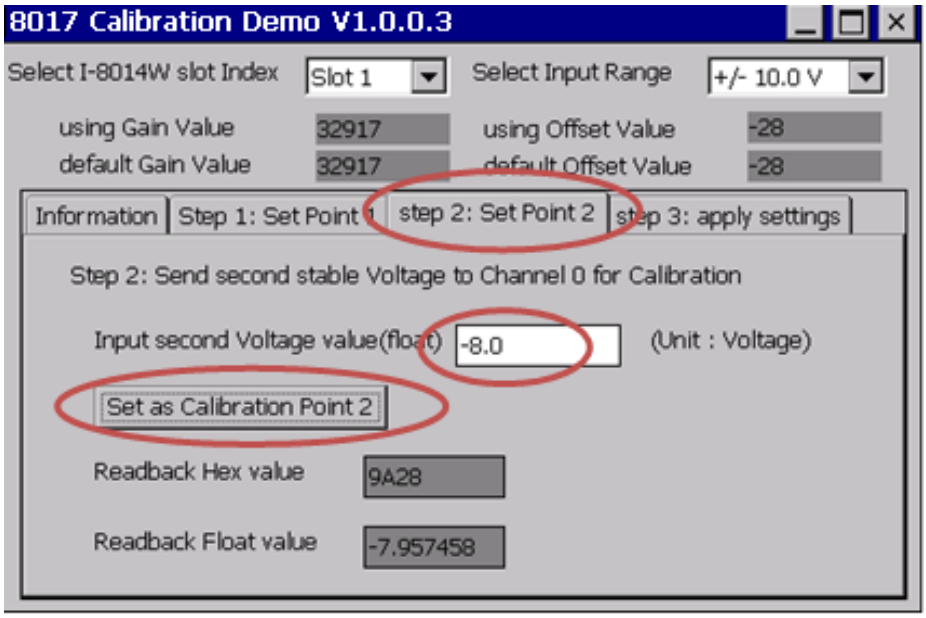
Step 8. Click the Step 1: Set Point 1 tab and type the value displayed on the meter (e.g., 8.0) in the Input First Voltage Value text box, and then click the Set as Calibration Point 1 button.



Step 9. Set the calibration source output to the second value (e.g., - 8 V in this example)

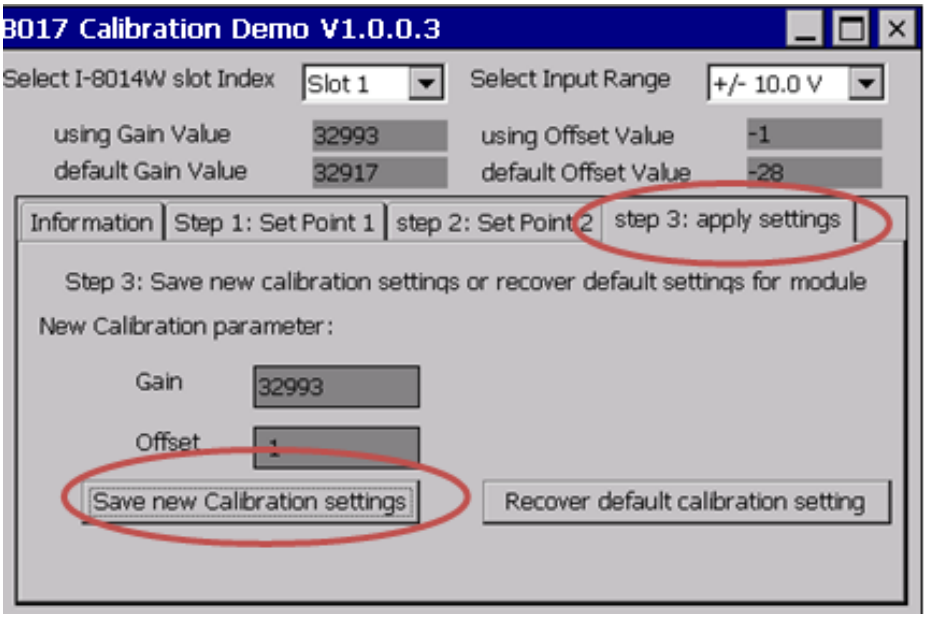
Step 10. Step 10. Click the Step 2: Set Point 2 tab and type the value displayed on the meter

(e.g., - 8.0) in the Input Second Voltage Value text box, and then click the Set as Calibration Point 2 button.



Step 11. Click the Step 3: Apply Settings tab, and check that the calibration parameters are correct. Click the Save New Calibration Settings button to save the calibration settings.

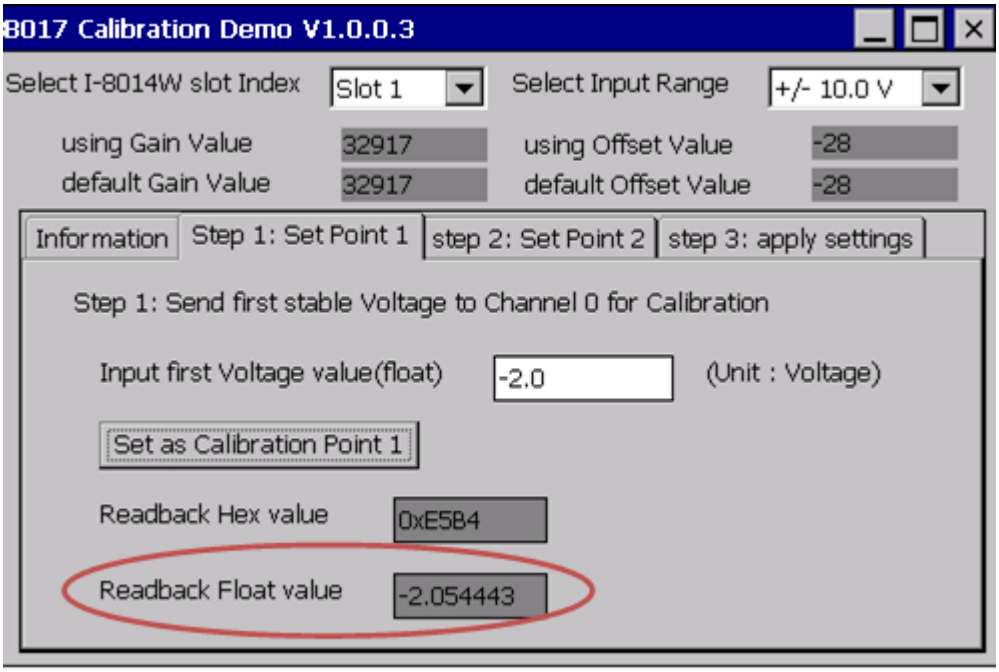
The calibration for the -10 V - +10 V input range is now complete.



5.2.2. Verifying the Calibration

Step 1. Set the calibration source to output a voltage to channel 0 on the I-8014W module. For example, -2 V.

Step 2. In the Calibration dialog box, click the Step 1: Set Point 1 tab and confirm that the AI Readback Float value is as illustrated in the image below:

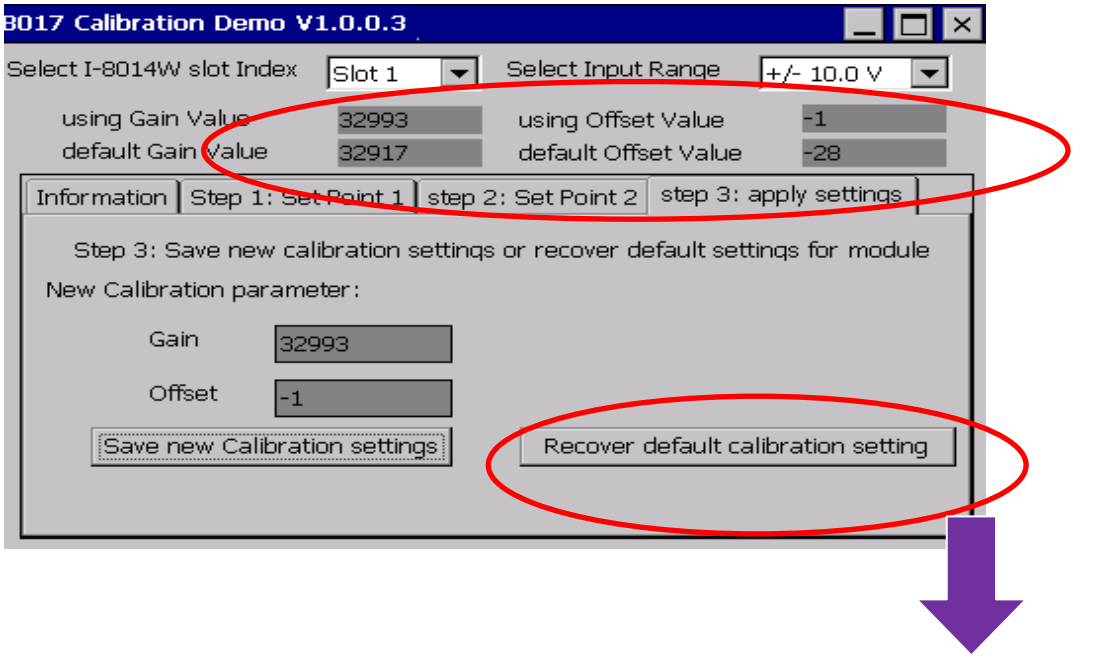


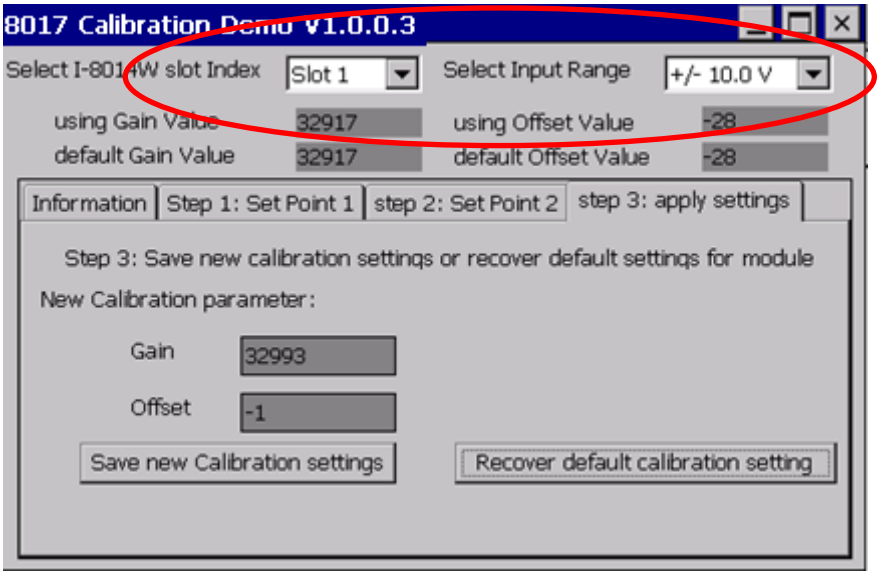
5.2.3. Restoring the Default Calibration Settings

When using the default input impedance of 200 kΩ, the calibration program includes a Recover Default Calibration Settings function that can be used to restore the Gain and Offset values to the factory default values:

Click the Step3: Apply Settings tab, and then click the Recover Default Calibration Settings button. The Gain and Offset settings will be restored to the factory default values and will be displayed in the upper section of the Calibration dialog box.

For an input impedance of 200 kΩ (the default setting), the calibration program provides Recover default calibration settings function to restore the Gain and Offset values to factory default:





6. Troubleshooting

This chapter discusses how to solve some common problems you may encounter.

This chapter contains:

- How to verify the AI function on a WinCE or WES PAC Service/Request Requirements
- What to do when the data read from the module seems unstable

6.1. Verifying Analog Input functionality on a WinCE or WES PAC device

If the data read from the module is inconsistent with the input signal, and you would like to confirm the input function, the pac_i8017W_Utility.exe tool may be helpful. The utility can only be used with modules designed for controllers using the WinCE and WES platforms and is located in the I-8017W C# demo program folder for the controller. (See the Location of the Demo Programs section for more details)

Step 1. Connect a stable signal to the module.

a. Connect your input signal according to whether differential or single-ended Jumper settings are used. (See the Jumper Settings section for more details)

b. The input range can be from -10 V to +10 V.

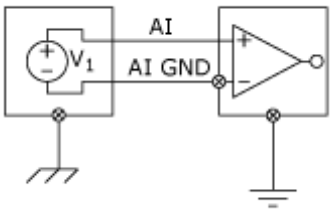
c. Insert the module into a slot in a Windows platform controller and then power on the controller.

Tips & Warnings

1. A battery output should provide a stable enough 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 the 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. Note that this method has no benefit in enhancing accuracy when measuring current input.



Step 2. Launch the pac_i8017W_Utility.exe

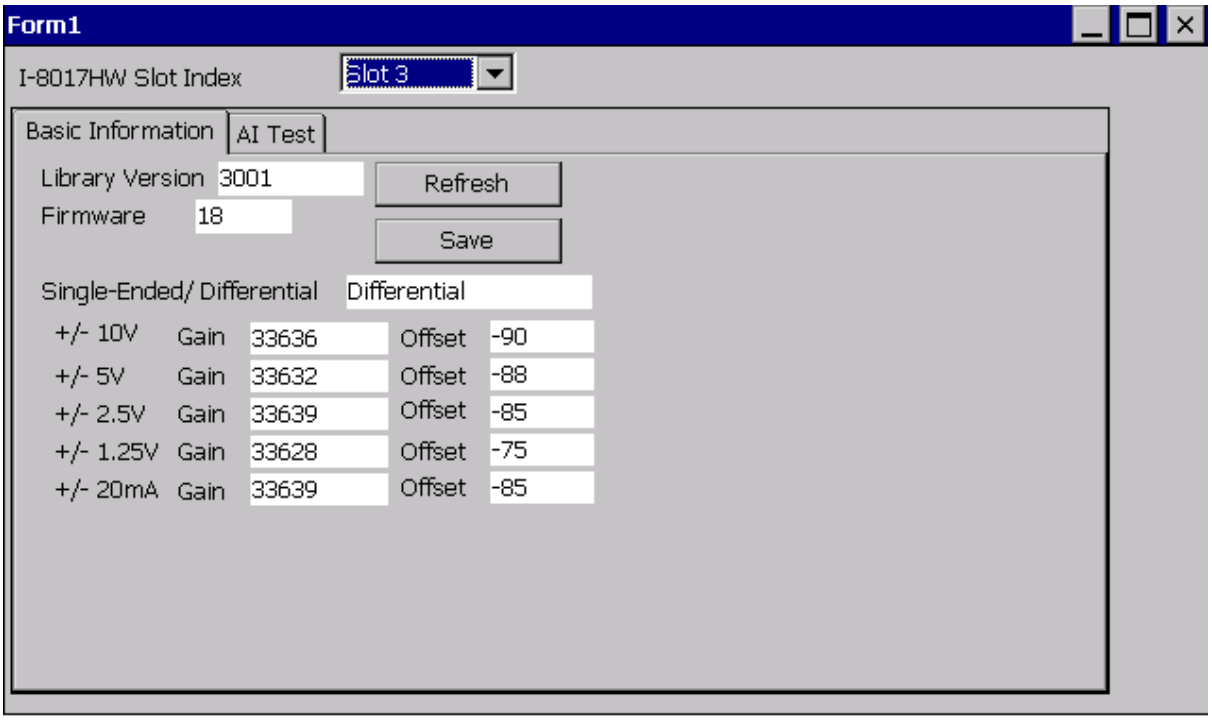
Step 3. Read the information from the module

a. Select the slot that the module is connected to from the slot index drop-down list.

b. Click the Basic Information tab.

The Basic Information page includes:

- The version information for the FPGA firmware
- The current position of the Differential/Single-ended jumper
- The Gain and Offset values for each input type



Click the Save button to save all the information to the Slot1_8017W_Info.txt file. This information is useful for troubleshooting when requesting service.

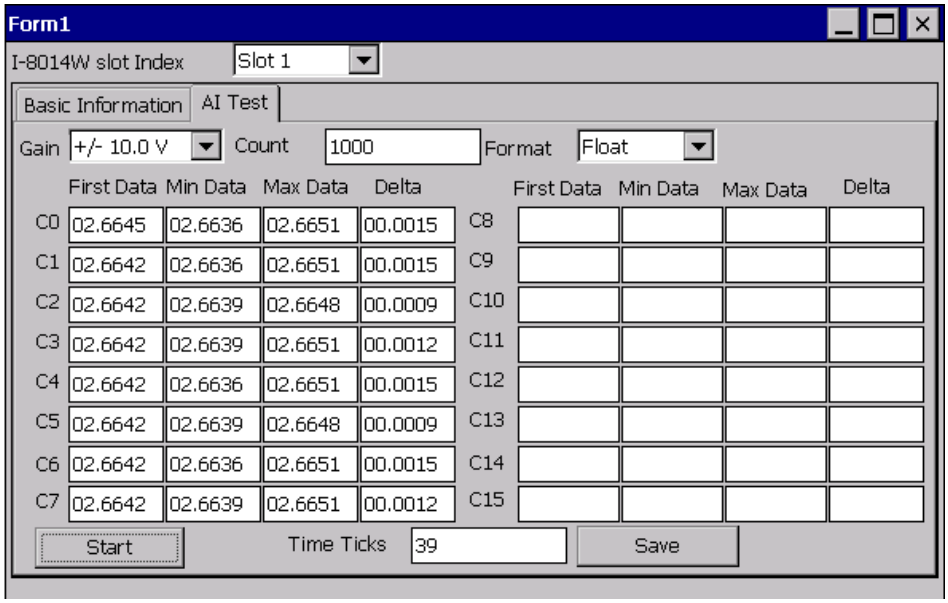
Verifying the Gain and Offset Values

In a normal situation, the Gain value should be around 33000 (33000 to 34000). 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 module to a different slot, and then repeat Steps 2 and 3 to confirm whether or not the Gain values are correct.

Test the input function.

a. Click the AI test tab, and then select the required input range from the Gain drop-down list.
b. Enter the required sample count, and choose the data format from the Format drop-down list.
c. Click the Start button.



After the sampling process is completed, the data will be displayed in the respective columns for each channel.

d. If necessary, click the Save button to save the data and the sampling time to the SampleData_Hex_mm_dd_hh_mim_sec.csv file.

6.2. Service Request Requirements

If you are using a stable signal source to output a signal to the module, such as a battery, 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 (See step 3 in Section 6.1 above)
- The file saved from the AI Test tab (See step 4 in Section 6.1 above)

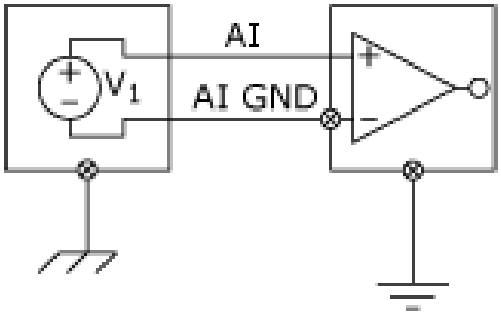
6.3. What to do when the data read from the module seems unstable

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(earth or building ground)

Because of the nature of the high-speed data acquisition function on the 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 that the V- pin is connected to the AGND (system ground) pin when measuring differential signals, as shown in the figure below.



Appendix A. Error Code

Error Code	Definition	Description
0	NoError	This indicates that there have been no errors
-1	ID_ERROR	There is a problem with the module ID
-2	SLOT_ERROR	There is a Slot index error (0 - 7)
-3	CHANNEL_ERROR	There is a Channel index error (0 - 15)
-4	GAIN_ERROR	There is a Gain error (0 - 4)
-6	NOT_SUPPORT_ERROR	The function is not support the Firmware
-7	NOT_Calibration	The module is not calibrated

Appendix B. Read AI Function Performance

Using a single channel

Platform	ReadAI (Polling)	ReadAIHex (Polling)
WES	90~95 KHz	90~95 KHz
CE6	90~95 KHz	90~95 KHz
CE5	90 KHz	90~95 KHz
IP-8000	7.6KHz	36KHz
I-8000	2KHz	12KHz

Using multiple channels

Platform	ReadAI (Polling)	ReadAIHex (Polling)
WES	35 KHz	35 KHz
CE6	35 KHz	35 KHz
CE5	35 KHz	35 KHz
IP-8000	6.6KHz	22KHz
I-8000	2KHz	9KHz

Notes

1. There is no need to switch the MUX when using a single channel as it provides the best performance. However, when using multiple channels the MUX needs to be switched and you should be aware that the performance will be affected by switching the MUX.
2. The MiniOS7 system is not designed for mathematical operations, so it is more suitable for non-continuous data sampling in high speed applications.
3. Large amounts of non-continuous data samples can be saved on the other memory devices, for example MicroSD cards or NAND flash memory.
4. A Backplane Timer Interrupt can be used for the CE5 and CE6 platforms when performing continuous data sampling.
5. The Timer on the WES platform can be affected by Ethernet communication or when using a mouse. If greater accuracy is required for the sample frequency (less than 50 ms), it is recommended that either the CE5 or the CE6 platform is used.

Appendix C. Revision History

This chapter provides revision history information to this document.

The table below shows the revision history.

Revision	Date	Description
1.0.0	January 2018	Initial issue
2.0.0	January 2018	• Added content for the I-8017DW and I-8017HCW modules • Added calibration instructions for modules based on the Windows platform • Added performance information for all platforms
3.0.0	July 2018	• Added content for the I-9017, I-9017-15,I-9017C-15 modules • Modified library , demo path • Added WP-9000 , ippc-wes7 library , demo path • Modified API functions
3.0.1	November 2018	• Modified the specification of the Single Channel Polling Mode