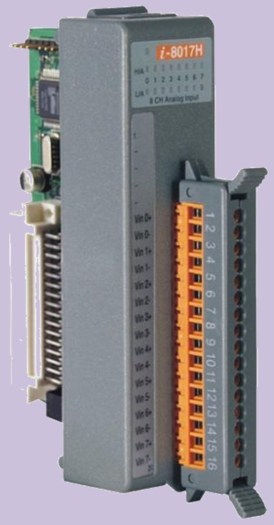
# I-8017/I-9017 Series
# I/O ModuleUser Manual

**V 3.0.0 January 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

# 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 can be connected to one module. It features 14-bit resolution, 100Ks/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-8017HW, however, an external 125 Ω resistor needs to be added in order 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

For more details related to resolution or to high-performance modules, please refer to selection guide on the web page below and select the appropriate module.
http://www.icpdas.com/products/Remote_IO/i-8ke/selection_analog_i8k.htm

Each of the three 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-8017HW, however, an external 125 Ω resistor needs to be added in order to measure the current source.
For more details, see Section 1.2 "Jumper Settings" and Section 1.5 "Wire Connections".

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

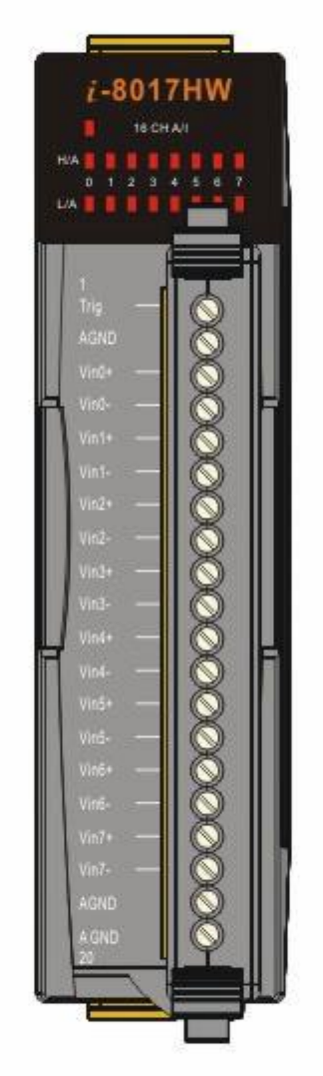
| Model | I-8017HW | I-8017HCW | I-8017DW |
|---|---|---|---|
| **Analog Output** | | | |
| Channels | 8-ch Differential/16-Single-ended | | |
| Voltage Input Range | ±1.25, ±2.5, ±5 V, ±10 V- | | |
| Current Input Range | -20 mA ~ +20 mA(Requires OptionalExternal 125 Ω Resistor) | | -20 mA ~ +20 mA(Jumper Select) |
| Resolution | 14-bit | | |
| Sample Rate | Single Channel Polling Mode :100K S/s<br>Single Channel Interrupt Mode: 50K S/s<br>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 | | |
| 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 | | |
| **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 ~ +75°C | | |
| Humidity | 5% ~ 95% RH, non-condensing | | |

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

| Model | I-9017 | I-9017-15 | I-9017C-15 |
|---|---|---|---|
| | | | |
| Channels | 8-ch Differential/16-Single-ended | 15-ch Differential/30-Single-ended | 15-ch Differential |
| Voltage Input Range | ±1.25, ±2.5, ±5 V, ±10 V | | - |
| Current Input Range | -20 mA ~ +20 mA(Requires OptionalExternal 125 Ω Resistor) | | -20 mA ~ +20 mA |
| Resolution | 14-bit | | |
| Sample Rate | Single Channel Polling Mode :100K S/s<br>Single Channel Interrupt Mode: 50K S/s<br>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) | | 125 Ω |
| **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<br>±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 31 mm x 134 mm | | |
| **Environment** | | | |
| Operating Temperature | -25 °C ~ +75°C | | |
| Storage Temperature | -40°C ~ +85°C | | |
| Humidity | 10 % ~ 90% RH, non-condensing | | |

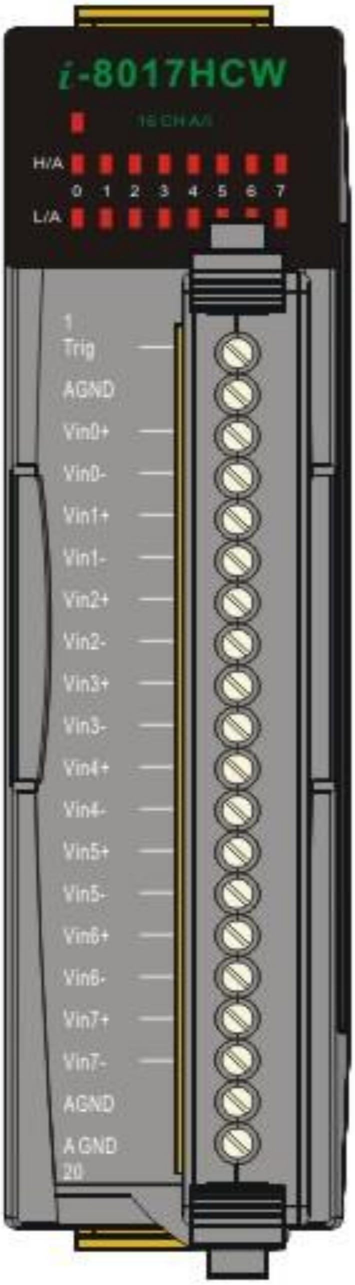
## 1.2. Pin Assignments

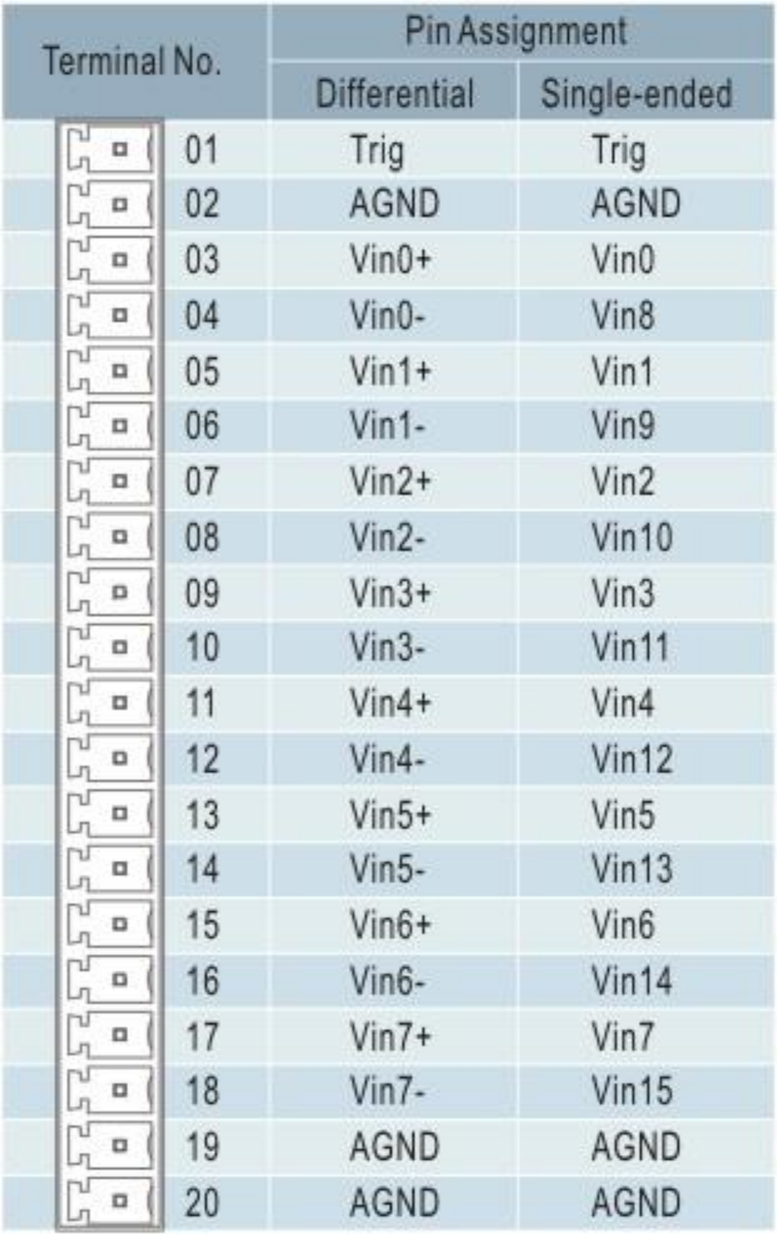
### I-8017HW:



| 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 |

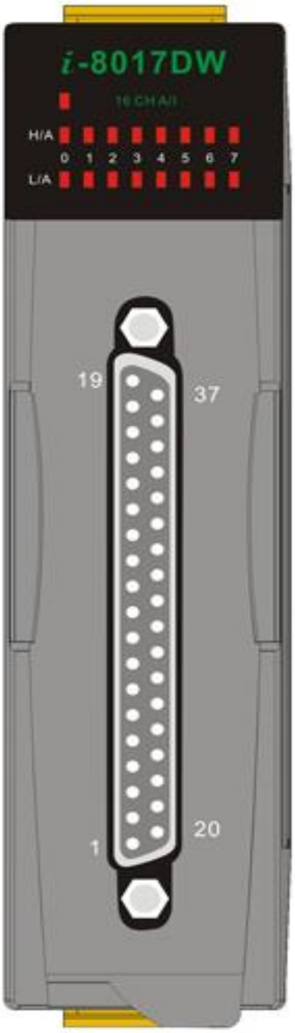
## I-8017HCW:

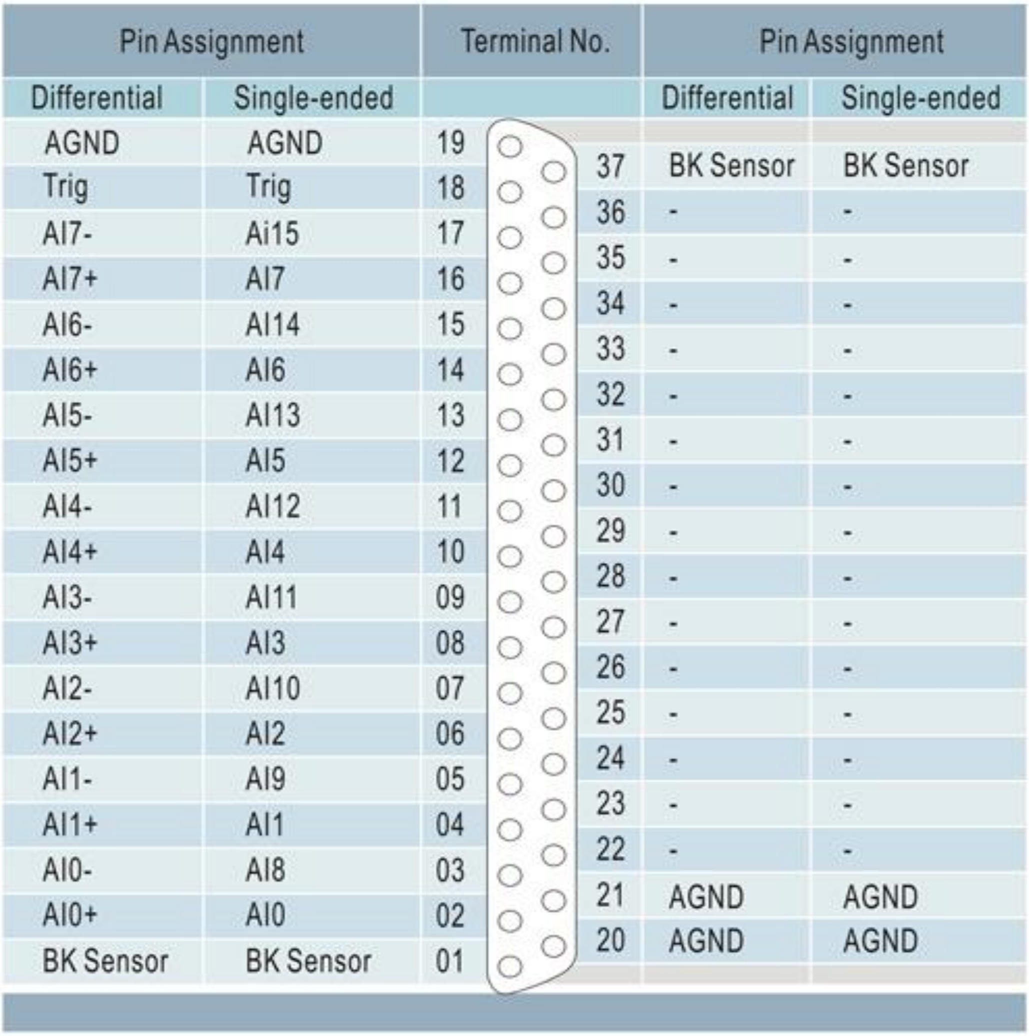




| Terminal No. | Pin Assignment | |
|---|---|---|
| | Differential | Single-ended |
| 01 | Trig | Trig |
| 02 | AGND | AGND |
| 03 | Vin0+ | Vin0 |
| 04 | Vin0- | Vin8 |
| 05 | Vin1+ | Vin1 |
| 06 | Vin1- | Vin9 |
| 07 | Vin2+ | Vin2 |
| 08 | Vin2- | Vin10 |
| 09 | Vin3+ | Vin3 |
| 10 | Vin3- | Vin11 |
| 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 |

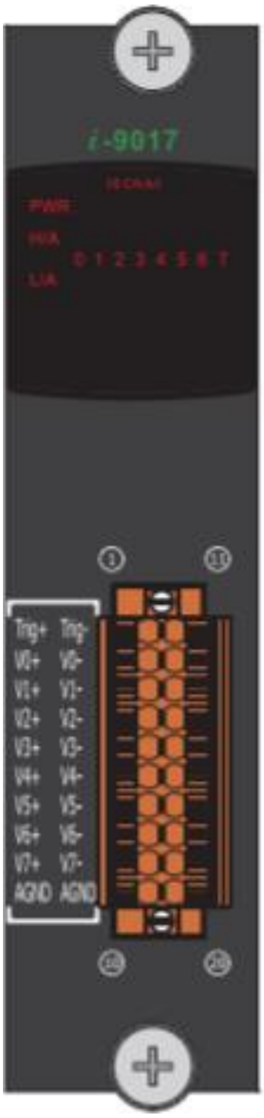
## I-8017DW:

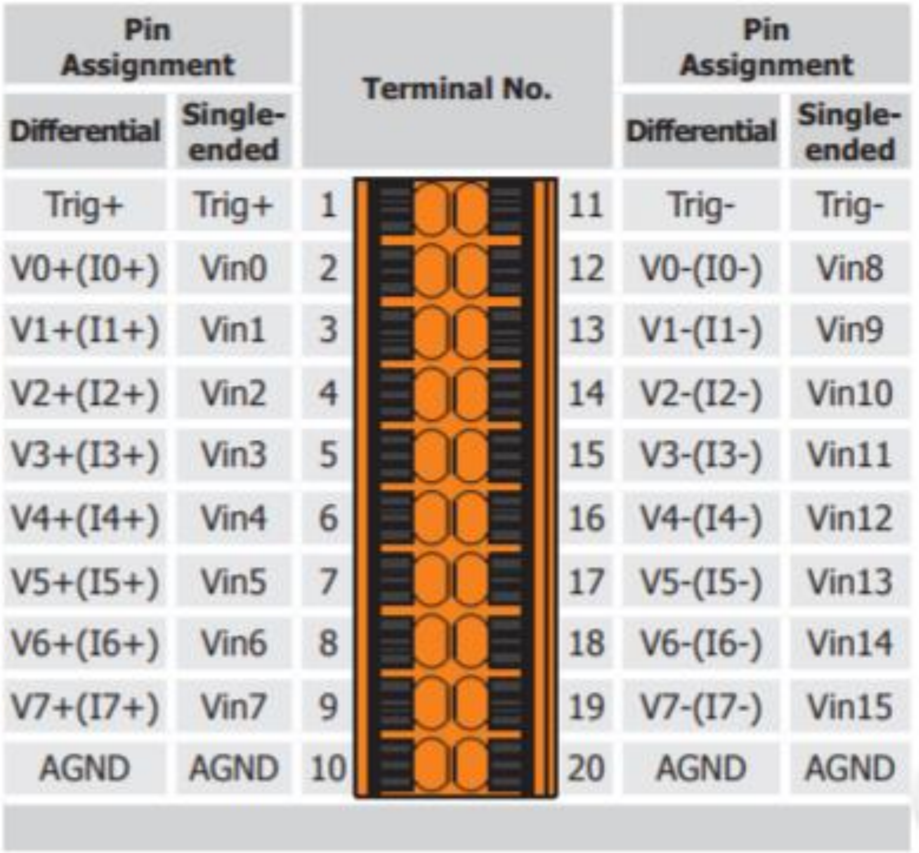




| Pin Assignment |  | Terminal No. |  | Pin Assignment |  |
|---|---|---|---|---|---|
| Differential | Single-ended |  |  | Differential | Single-ended |
| AGND | AGND | 19 | 37 | BK Sensor | BK Sensor |
| Trig | Trig | 18 | 36 | - | - |
| AI7- | Ai15 | 17 | 35 | - | - |
| AI7+ | AI7 | 16 | 34 | - | - |
| 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 |  |  |  |

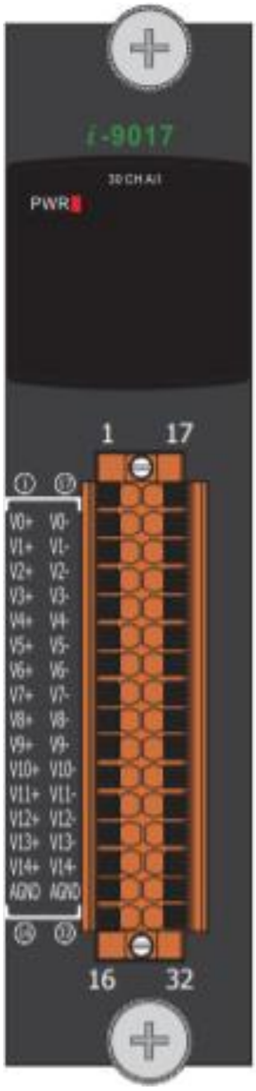
## I-9017:

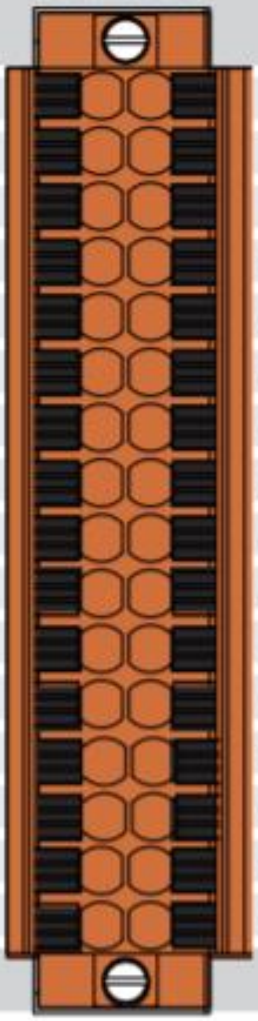




| Pin Assignment | | Terminal No. | | Pin Assignment | |
|---|---|---|---|---|---|
| Differential | Single-ended | | | Differential | Single-ended |
| Trig+ | Trig+ | 1 | 11 | Trig- | Trig- |
| V0+(I0+) | Vin0 | 2 | 12 | V0-(I0-) | Vin8 |
| V1+(I1+) | Vin1 | 3 | 13 | V1-(I1-) | Vin9 |
| V2+(I2+) | Vin2 | 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:

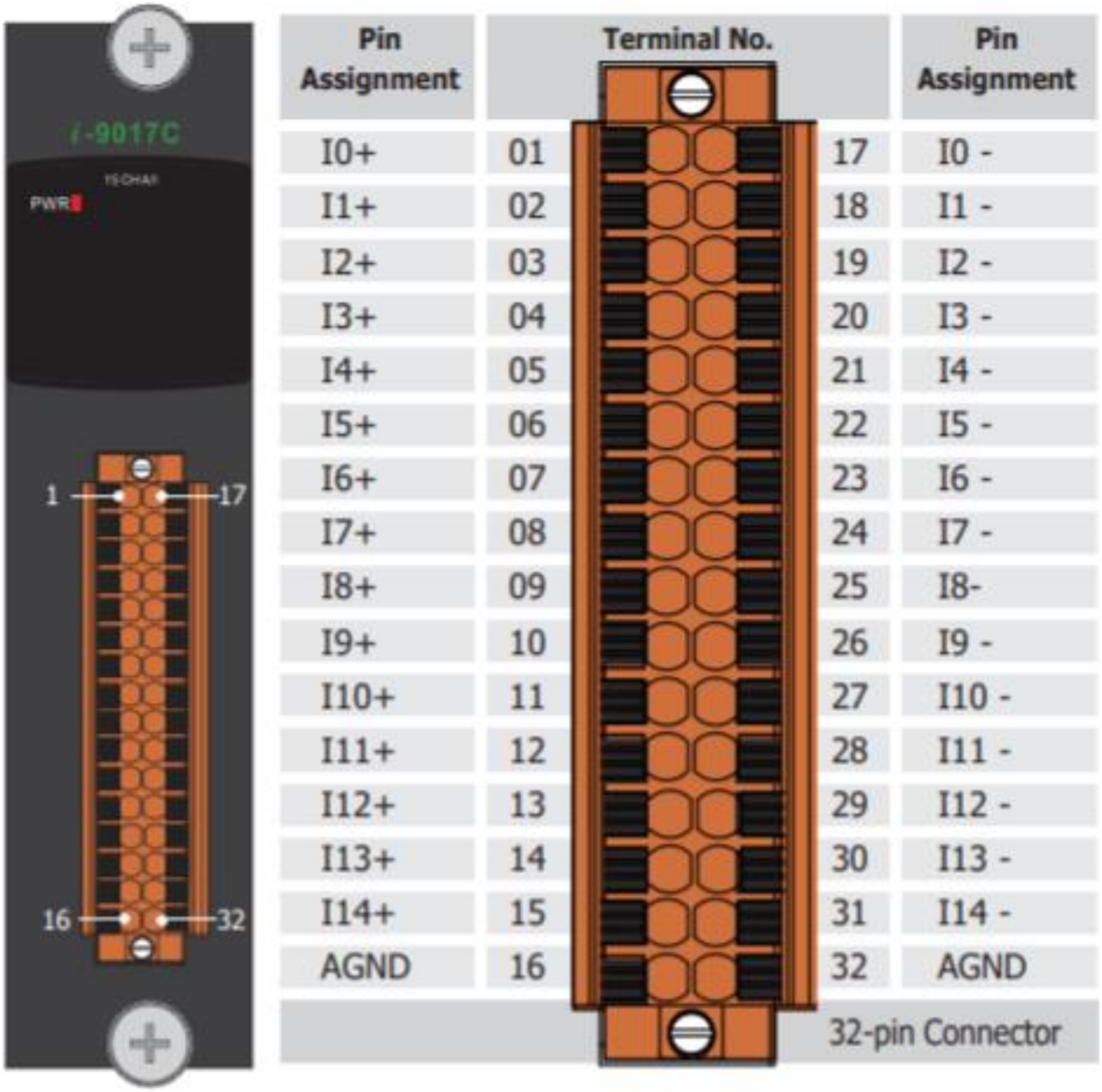




| Pin Assignment | | Terminal No. | | Pin Assignment | |
|---|---|---|---|---|---|
| V0+ | Vin0 | 01 | 17 | V0 - | Vin15 |
| V1+ | Vin1 | 02 | 18 | V1 - | Vin16 |
| V2+ | Vin2 | 03 | 19 | V2 - | Vin17 |
| V3+ | Vin3 | 04 | 20 | V3 - | Vin18 |
| V4+ | Vin4 | 05 | 21 | V4 - | Vin19 |
| V5+ | Vin5 | 06 | 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:



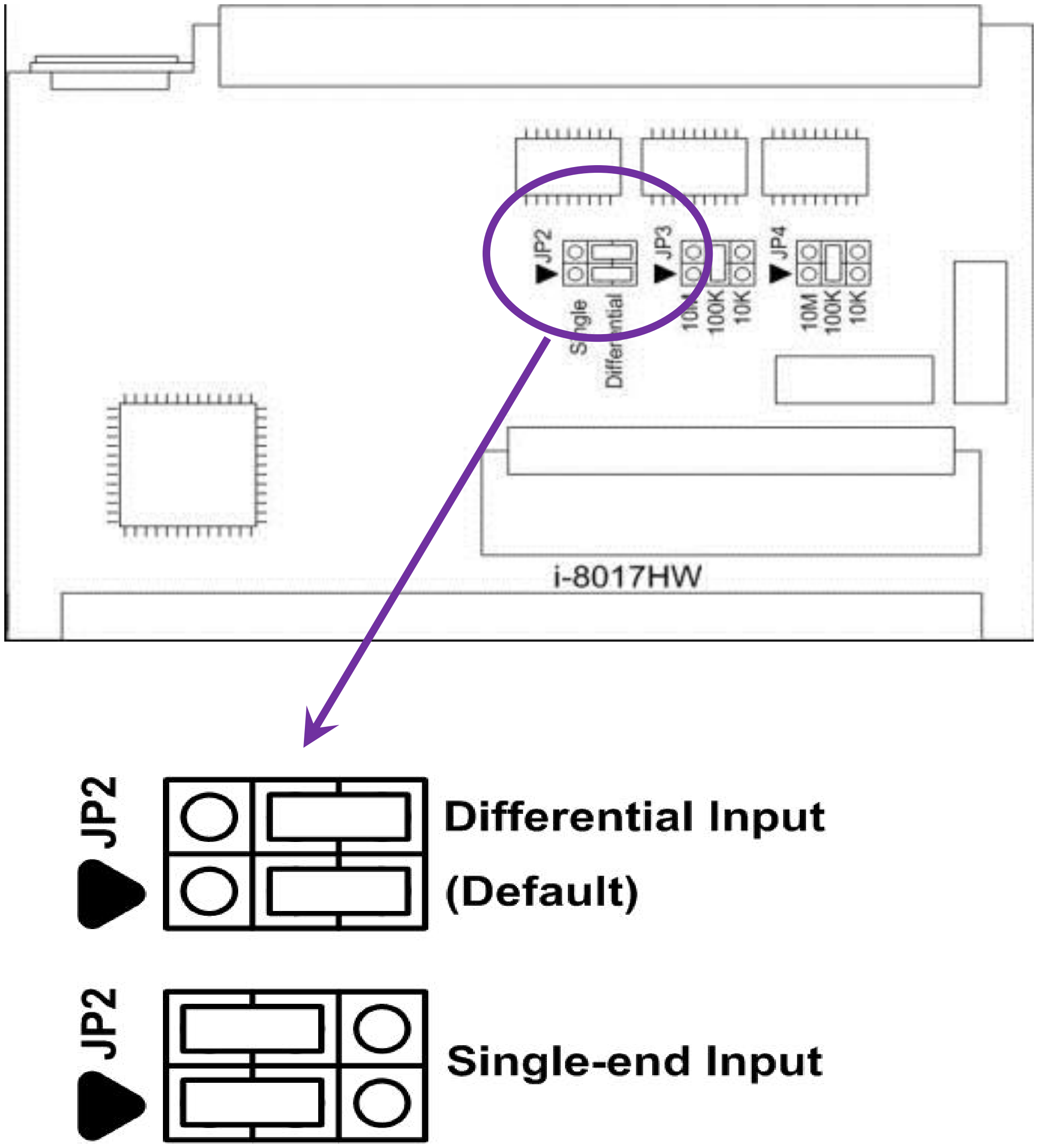
| Pin Assignment | Terminal No. | | Pin Assignment |
|---|---|---|---|
| I0+ | 01 | 17 | I0 - |
| I1+ | 02 | 18 | I1 - |
| I2+ | 03 | 19 | I2 - |
| I3+ | 04 | 20 | I3 - |
| I4+ | 05 | 21 | I4 - |
| I5+ | 06 | 22 | I5 - |
| I6+ | 07 | 23 | I6 - |
| I7+ | 08 | 24 | I7 - |
| I8+ | 09 | 25 | I8- |
| I9+ | 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

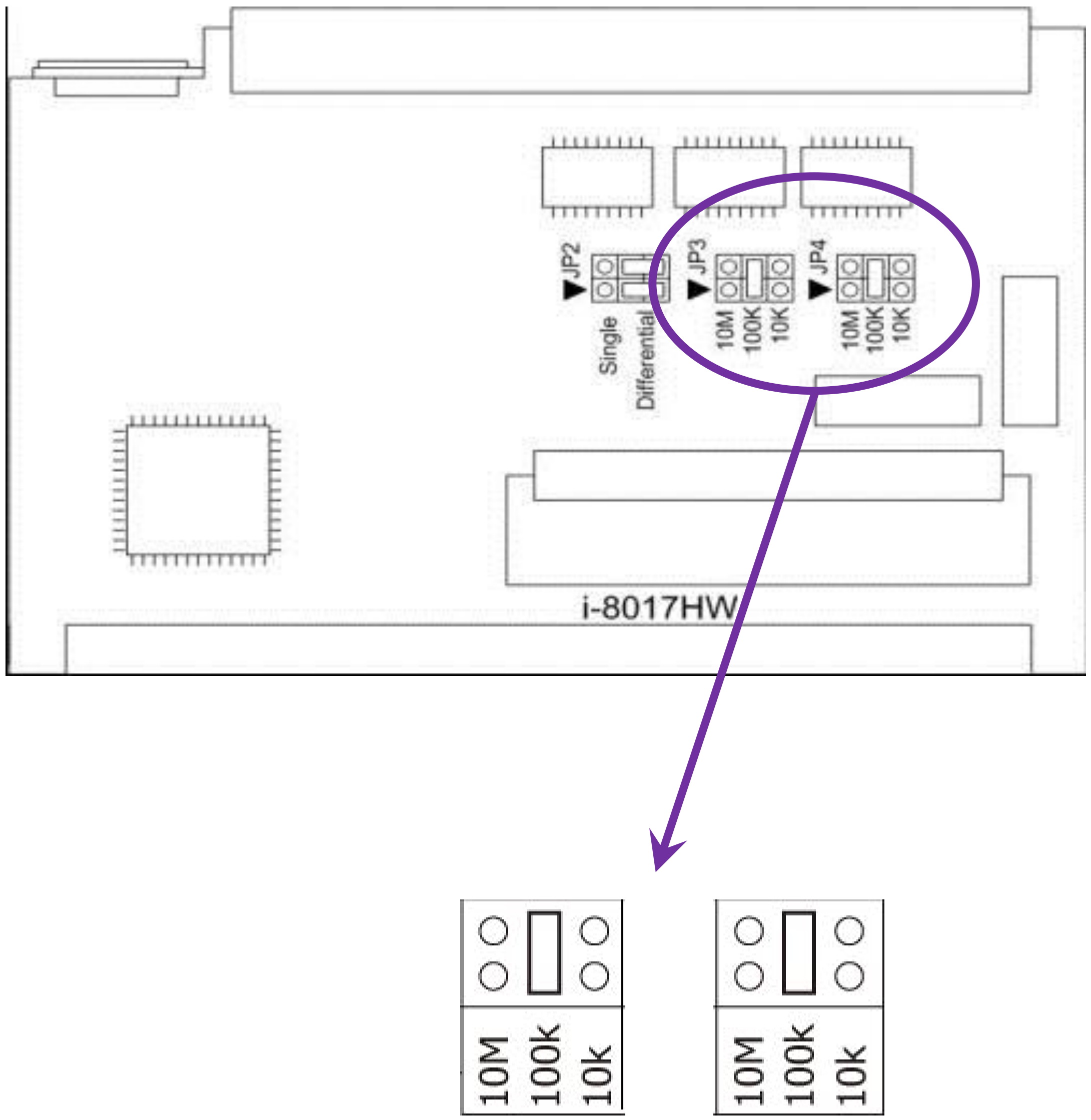
### I-8017HW

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)

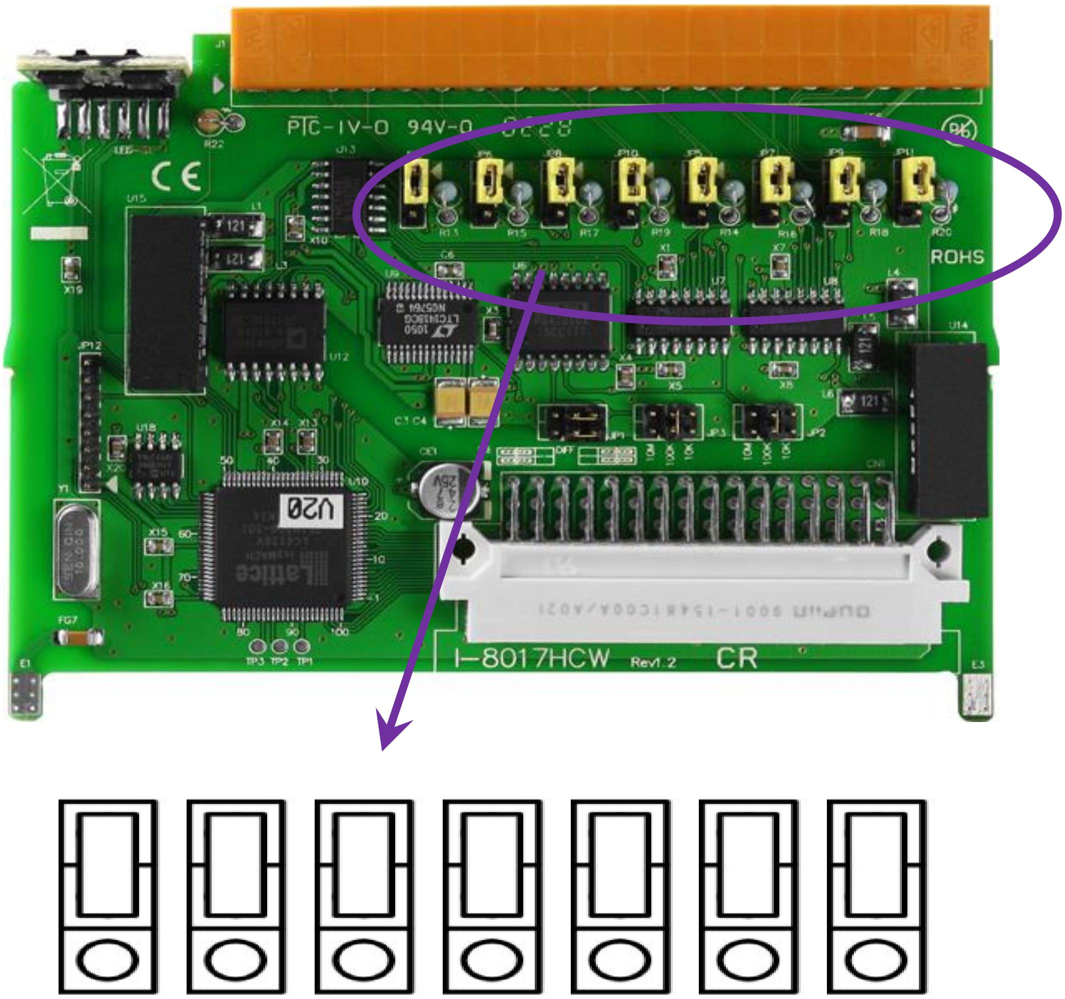
Note: 1. The Jumpers should set on the same value
2. Input Impedance = 2 x setting value

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

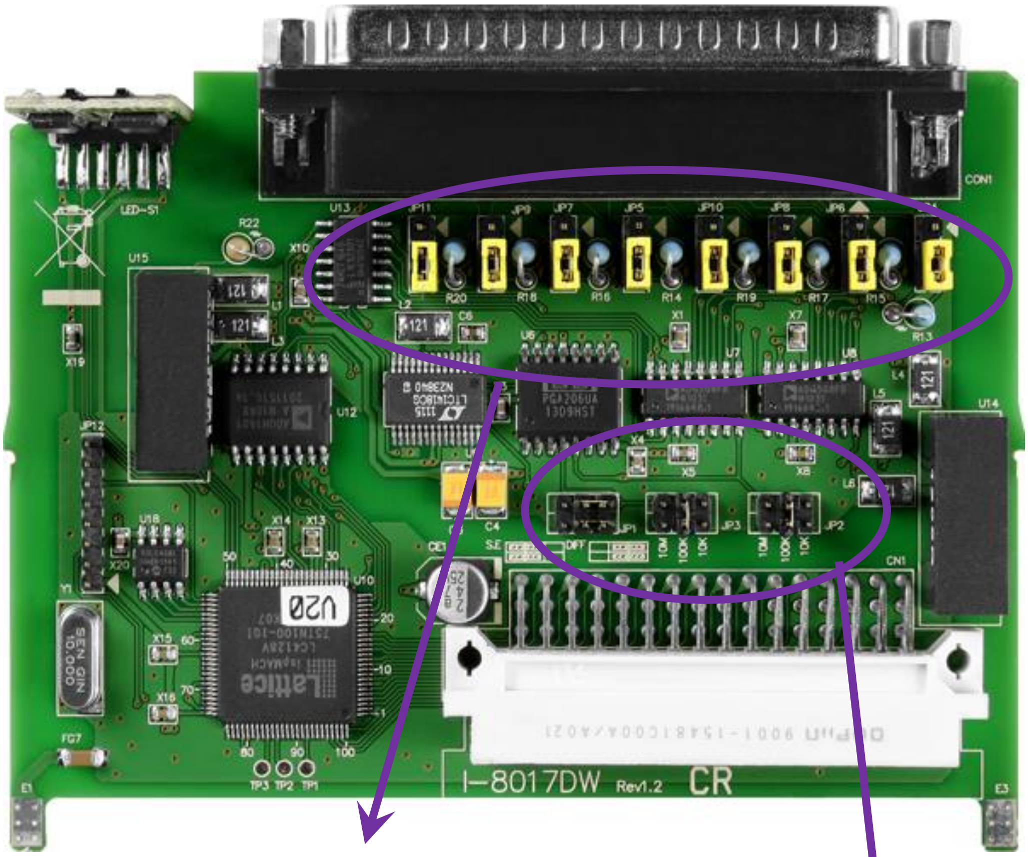
In most cases, 200kΩ is sufficient. Note that each time the input impedance is adjusted on a calibrated module, the module must be recalibrated. For more details, refer to the relevant Calibration information, which can be found in Section 5.1 if you are using either an I-8000 or iPAC-8000 (MiniOS7 platform)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

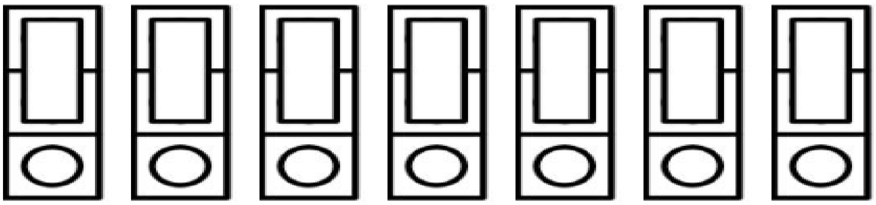
125Ω Resistor Jumper

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



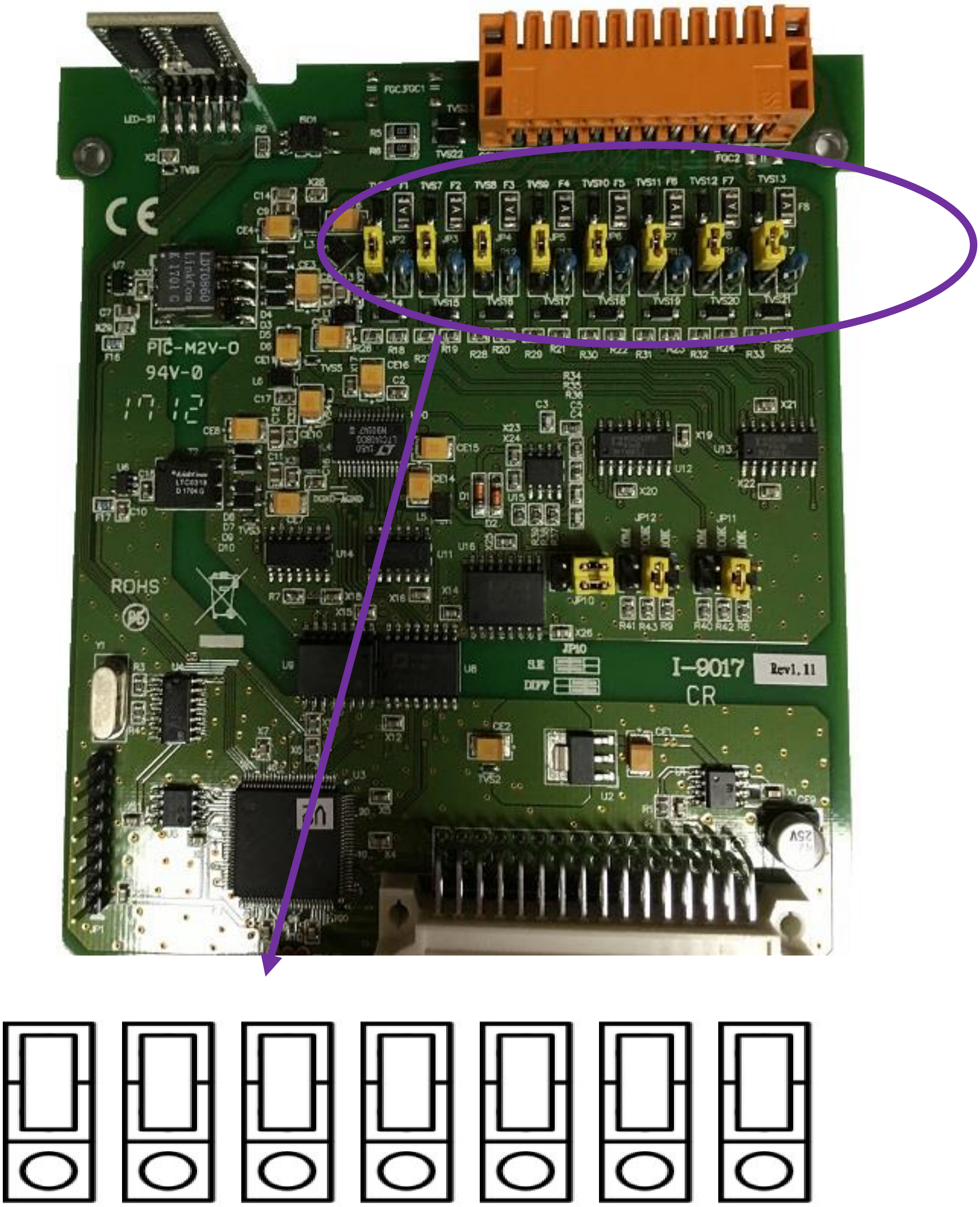
The default jumper position for current measurement on the I-8017HCW





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

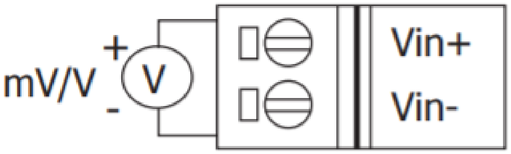
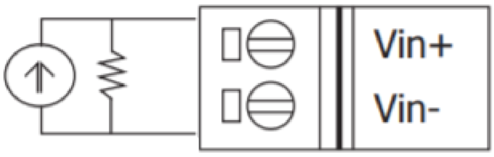
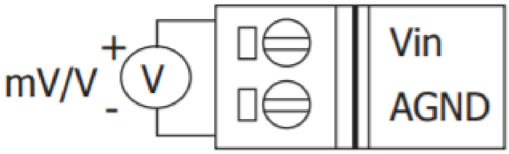
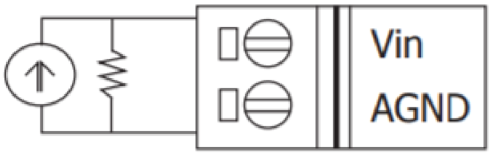
The jumper settings used to adjust both the input impedance and the single-ended and differential input on the I-8017HW are the same as those for the I-8017DW,I-8017HCW and the I-9017.



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

## 1.4. Wire Connections

### I-9017

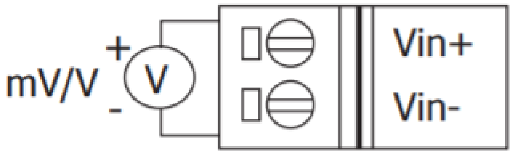
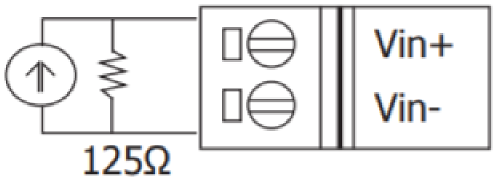
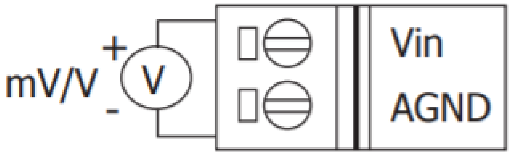
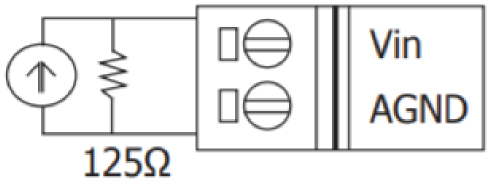
| | Voltage Input Wiring | Current Input Wiring |
|---|---|---|
| **Differential** |  |  |
| **Single-ended** |  |  |

**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.

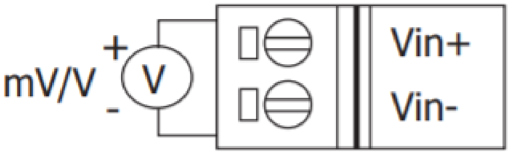
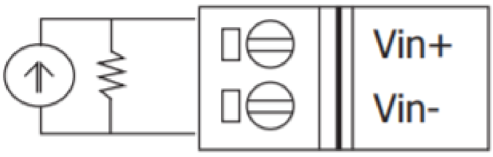
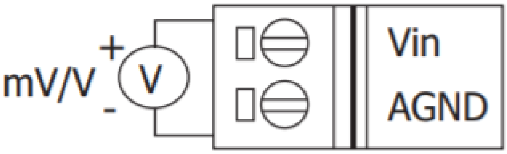
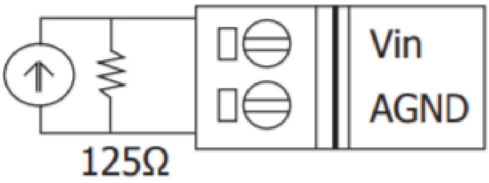
### I-8017HW/I-9017-15:

| | 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/I-9017C-15

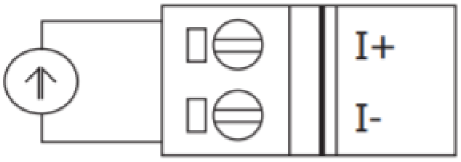
| | Voltage Input Wiring | Current Input Wiring |
|---|---|---|
| **Differential** |  |  |
| **Single-ended** |  |  |

**Note:**

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

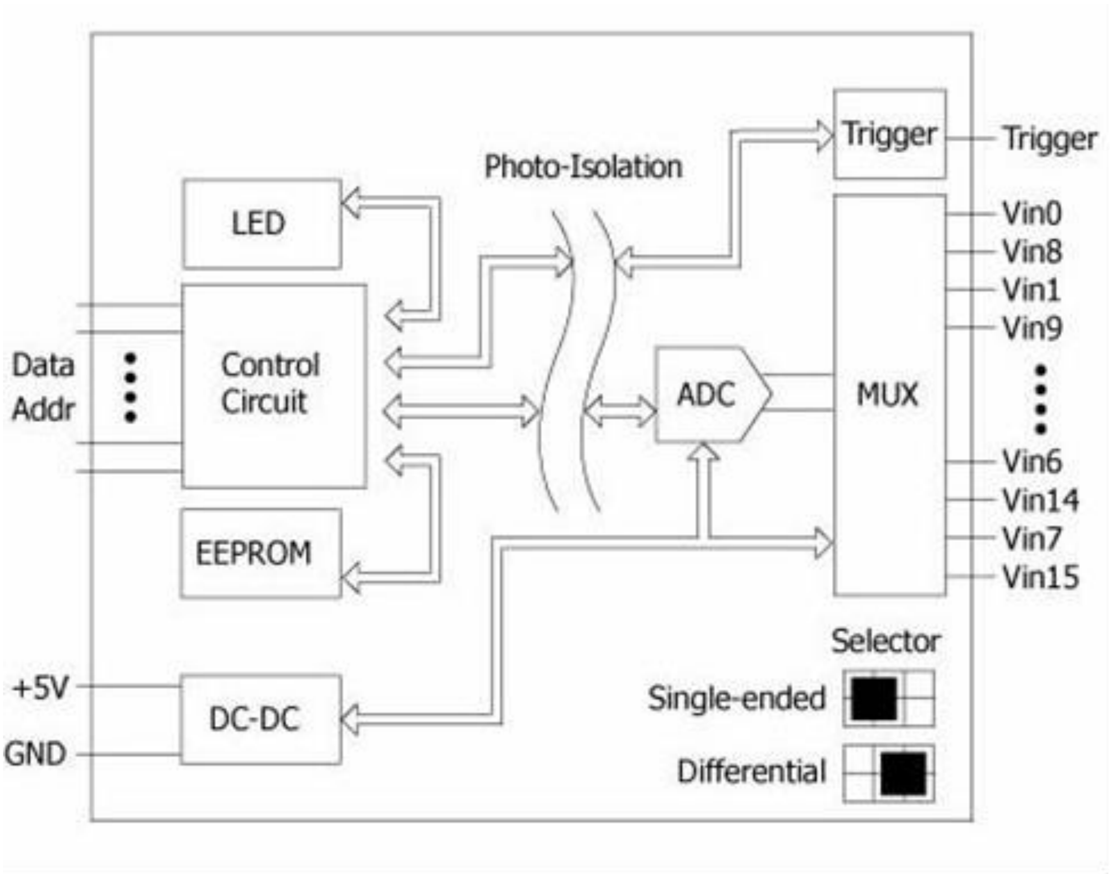
Single-ended Input Type: Current Input Wiring need to jumper at voltage input, an options external 125 Ω resistor is required.

### I-9017C-15C

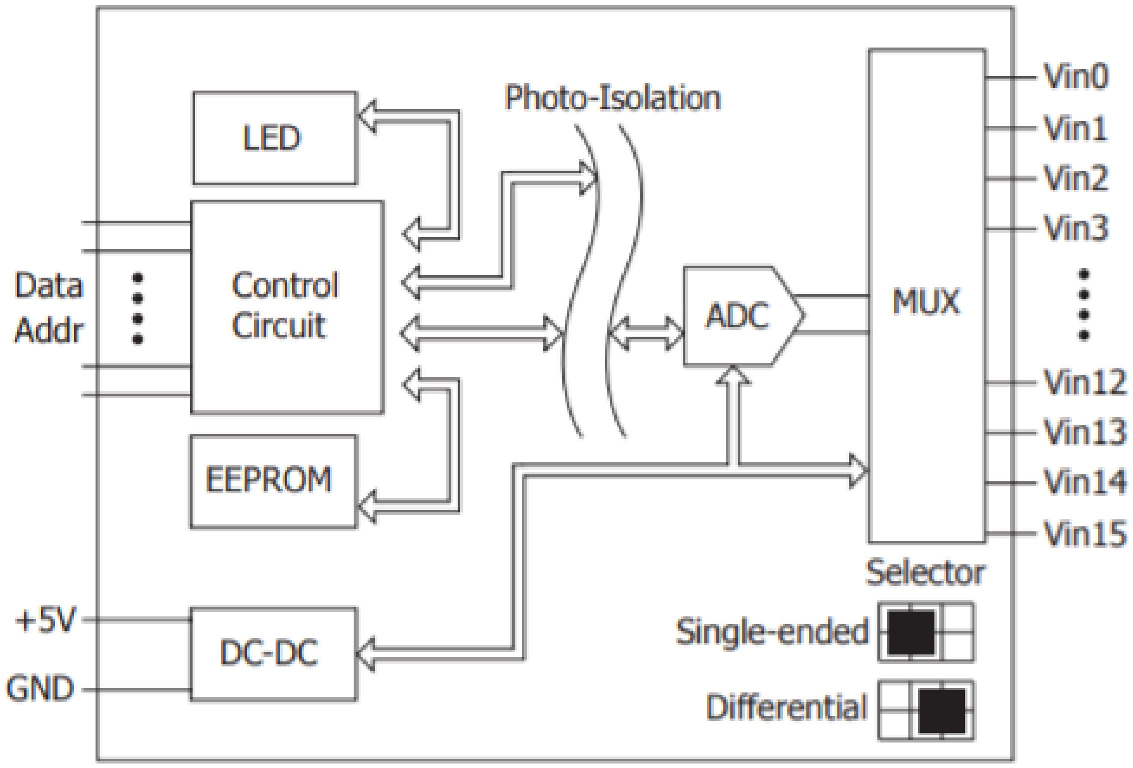
| | Current Input Wiring |
|---|---|
| **Differential** |  |

## 1.5. Block Diagram

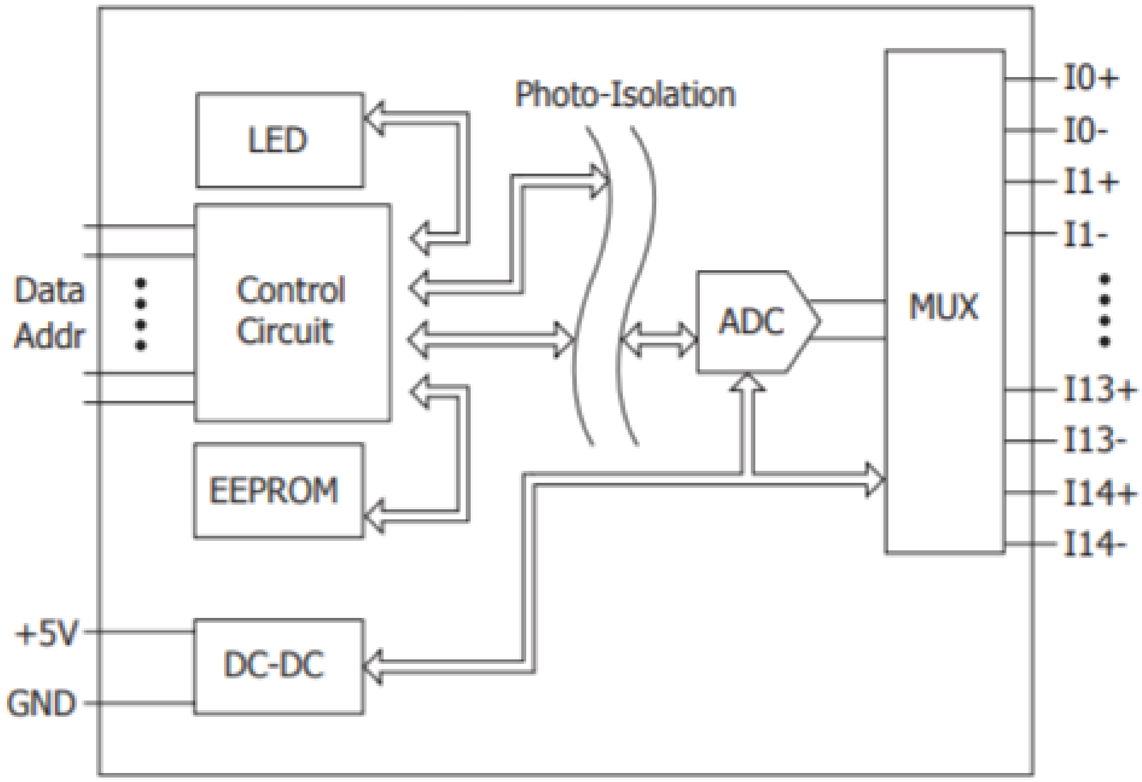
### I-8017HW/I-8017DW/I-8017HCW:



### I-9017/I-9017-15



### I-9017C-15



## 1.6. 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<br>ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/8000/841x881x/demo/lib/ |
| Demo | CD:\Napdos\8000\841x881x\demo\IO_in_Slotor<br>ftp://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<br>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<br>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<br>ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/io_modules/ |
| Demo | eVC Demo:<br>CD:\napdos\wp-8x4x_ce50\Demo\WinPAC\eVC\IO\Local or<br>ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/io/local/<br>C# Demo:<br>CD:\napdos\wp-8x4x_ce50\Demo\WinPAC\C#\IO\Local or<br>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<br>ftp://ftp.icpdas.com/pub/cd/winpac_am335x/wp-9000/sdk/io_modules/ |
| Demo | VC2008 Demo:<br>CD:\WinPAC_AM335x\wp-9000\demo\PAC\Vc2008\IO\Local<br>ftp://ftp.icpdas.com/pub/cd/winpac_am335x/wp-9000/demo/pac/vc2008/io/local/<br>C# Demo:<br>CD:\WinPAC_AM335x\wp-9000\demo\PAC\C#\IO\Local<br>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<br>ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/sdk/special_io/ |
| Demo | VC2005 Demo:<br>CD:\Demo\XPAC\VC2005\IO\Local<br>ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/demo/xpac/vc2005/io/local/<br>C# Demo:<br>CD:\Demo\XPAC\C#\IO\Local<br>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<br>ftp://ftp.icpdas.com/pub/cd/xpac-atom-ce6/sdk/special_io/ |
| Demo | VC2005 Demo:<br>CD:\Demo\XPAC\VC2005\IO\Local<br>ftp://ftp.icpdas.com/pub/cd/xpac-atom-ce6/demo/xpac/vc2005/io/local/<br>C# Demo:<br>CD:\Demo\XPAC\C#\IO\Local<br>ftp://ftp.icpdas.com/pub/cd/xpac-atom-ce6/demo/xpac/c%23/io/local/ |
| **For XP-8000-CE6** | |
| Library | CD:\SDK\IO<br>ftp://ftp.icpdas.com/pub/cd/xp-8000/sdk/io/ |
| Demo | VC2005 Demo:<br>CD:\Demo \XPAC \VC2005\IO\Local<br>ftp://ftp.icpdas.com/pub/cd/xp-8000/demo/pacsdk/vc/io/local/<br>CD:\Demo \XPAC\C#\IO\Local<br>ftp://ftp.icpdas.com/pub/cd/xp-8000/demo/pacsdk/csharp.net/io/local/windows_forms/ |

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

# 2. Quick Start

This section provides a Getting Started guide and details of the calibration process when using the I-8014W 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-8014W:

- 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

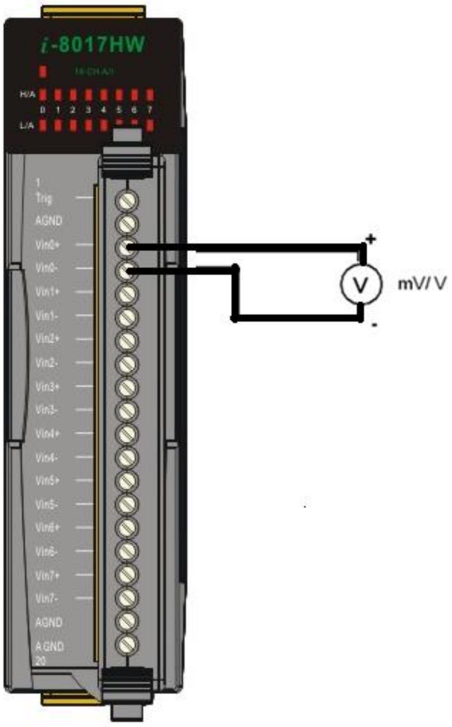
The 8017ai.exe executable file, which is located in the 8017h_ReadAI folder of the demo programs, can be used to retrieve the basic configuration information related to the module and to verify the AI read functions. The basic configuration information includes:

- 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

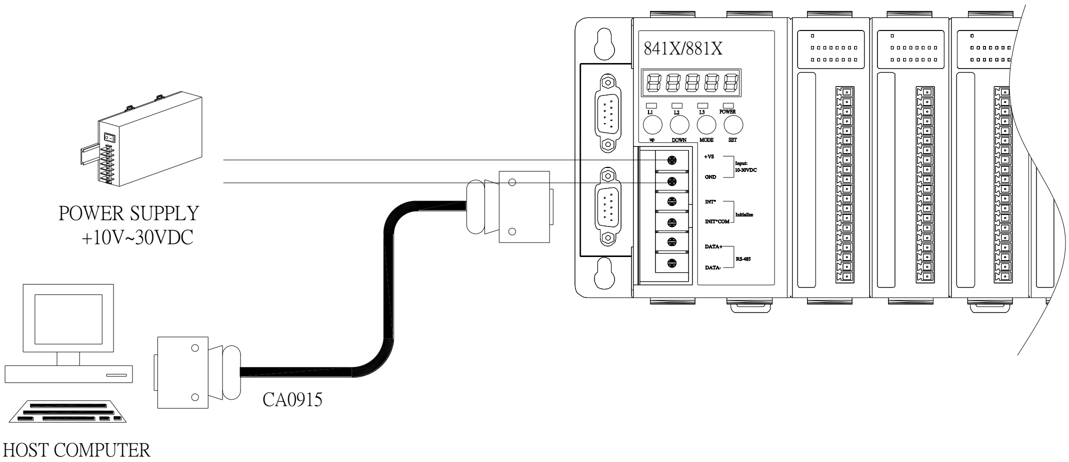
(See the Location of the Demo Programs section for details of where to find the 8017ai.exe file in the demo programs folder)

**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.



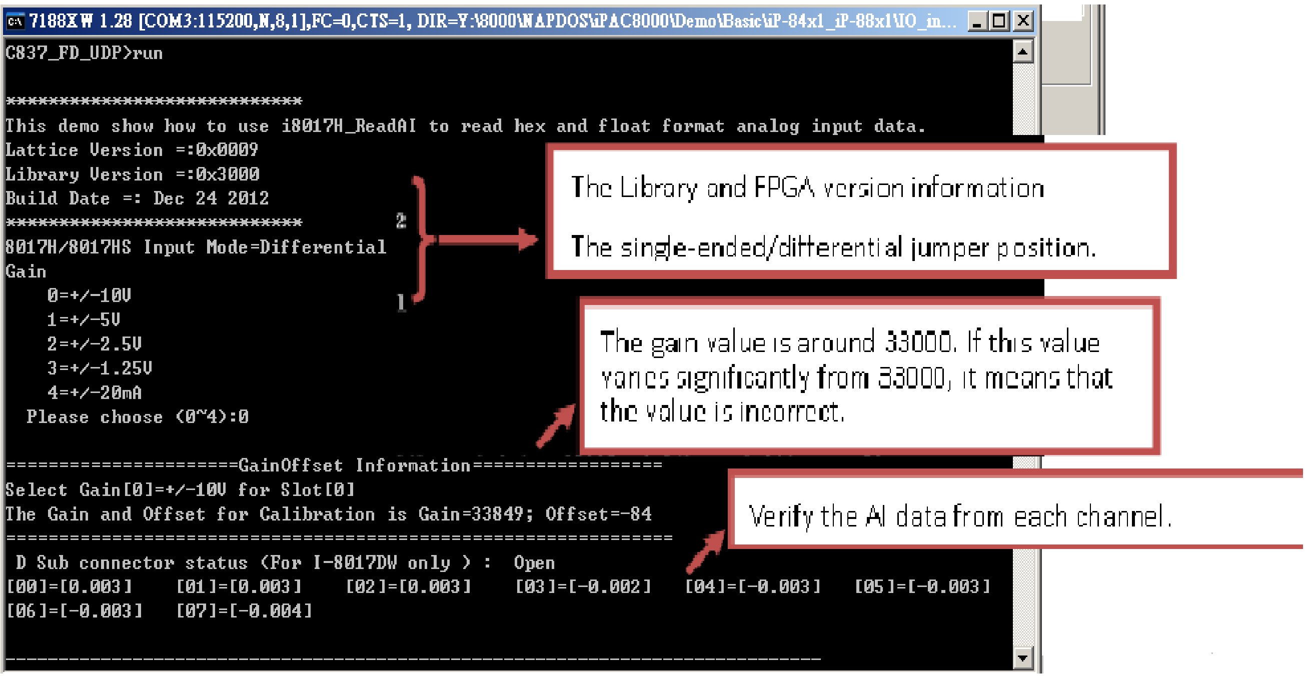
**Step 3.** Connect the power supply to the module, and connect the control unit to the Host PC using an RS-232 cable.



**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.


7188XW 1.28 [COM3:115200,N,8,1],FC=0,CTS=1, DIR=Y:\8000\NAPDOS\iPAC8000\Demo\Basic\iP-84x1_iP-88x1\IO_in...
2
1

```
C837_FD_UDP>run

******************************
This demo show how to use i8017H_ReadAI to read hex and float format analog input data.
Lattice Version =:0x0009
Library Version =:0x3000
Build Date =: Dec 24 2012
******************************
8017H/8017HS Input Mode=Differential
Gain
    0=+/-10V
    1=+/-5V
    2=+/-2.5V
    3=+/-1.25V
    4=+/-20mA
  Please choose (0~4):0

======================GainOffset Information==================
Select Gain[0]=+/-10V for Slot[0]
The Gain and Offset for Calibration is Gain=33849; Offset=-84
===============================================================
 D Sub connector status (For I-8017DW only ) :  Open
[00]=[0.003]    [01]=[0.003]    [02]=[0.003]    [03]=[-0.002]   [04]=[-0.003]   [05]=[-0.003]
[06]=[-0.003]   [07]=[-0.004]

------------------------------------------------------------------------------
```

The Library and FPGA version information

The single-ended/differential jumper position.

The gain value is around 33000. If this value varies significantly from 33000, it means that the value is incorrect.

Verify the AI data from each channel.

## 2.2. Windows-based Controllers

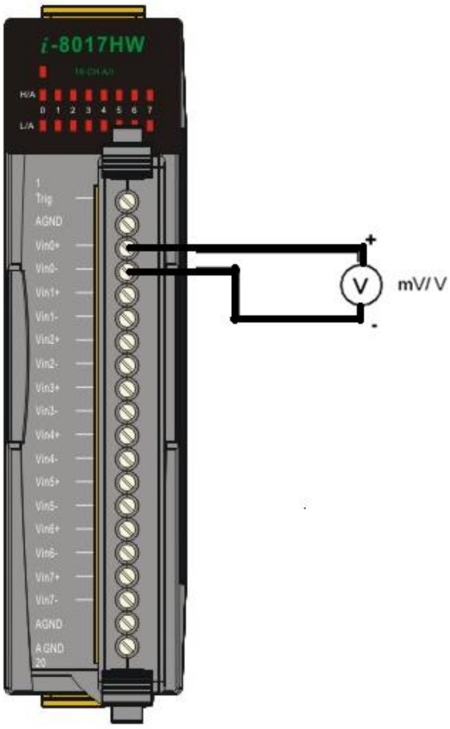
The pac_i8017HW_Utility.exe executable file, which is located in the pac_i8017HW_Utility folder of the demo programs, can be used to retrieve the basic configuration information related to the module, and to verify the AI read functions. The basic configuration information includes:

- 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.



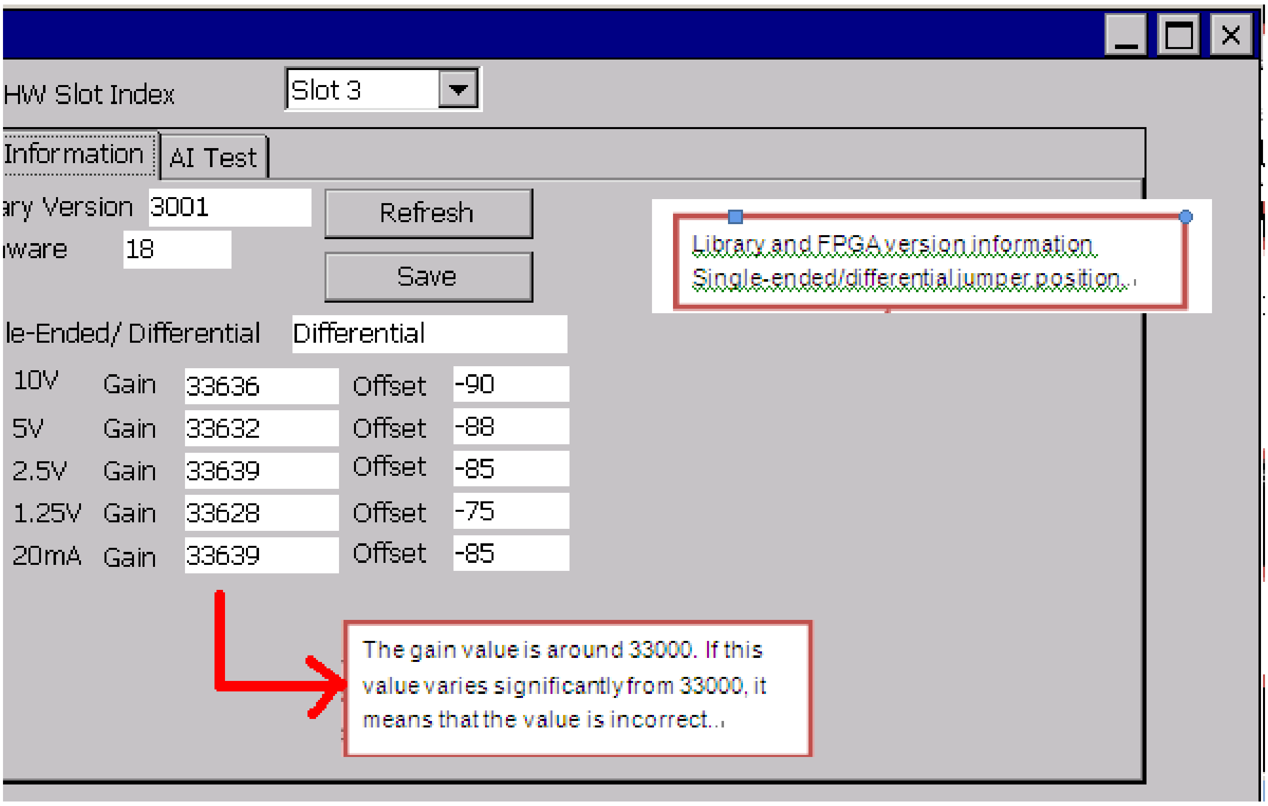
**Step 3.** Insert the module into a vacant slot in the control unit and power on the PAC controller

**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



HW Slot Index: Slot 3

ary Version: 3001

ware: 18

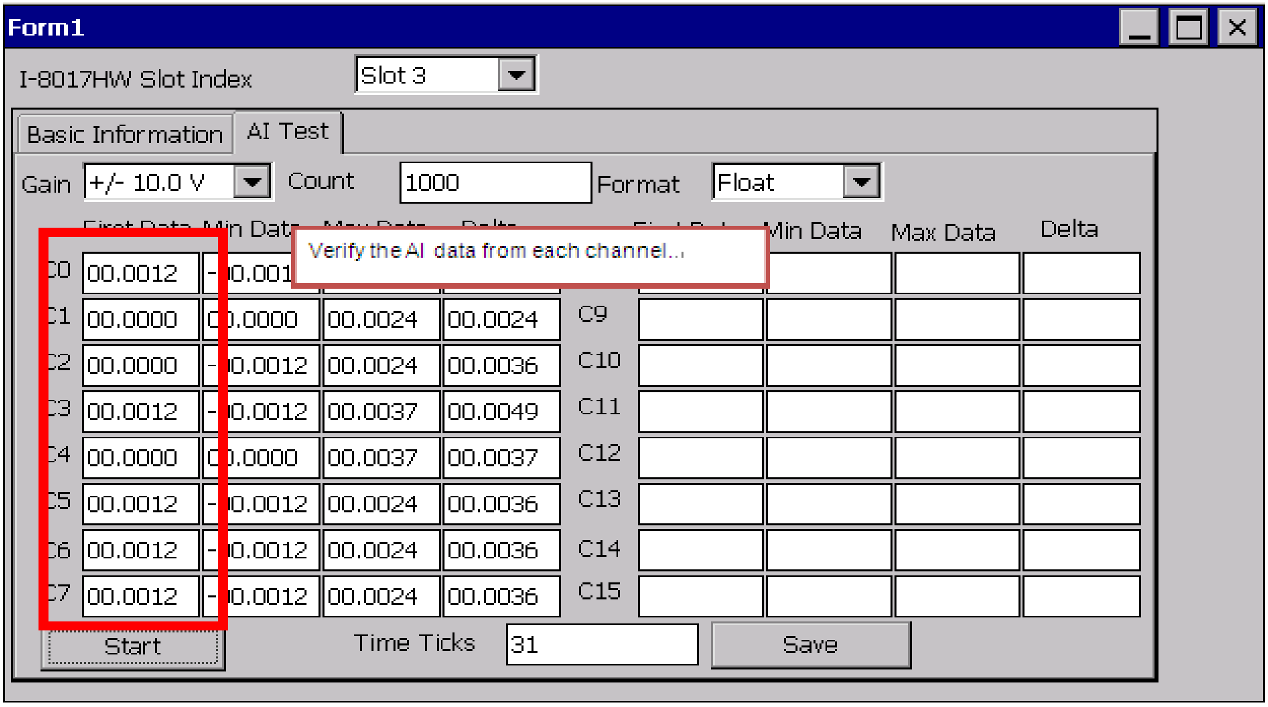
le-Ended/ Differential: Differential

| | | | | |
|---|---|---|---|---|
| 10V | Gain | 33636 | Offset | -90 |
| 5V | Gain | 33632 | Offset | -88 |
| 2.5V | Gain | 33639 | Offset | -85 |
| 1.25V | Gain | 33628 | Offset | -75 |
| 20mA | Gain | 33639 | Offset | -85 |

Library and FPGA version information
Single-ended/differential jumper position.

The gain value is around 33000. If this value varies significantly from 33000, it means that the value is incorrect..

## Read the AI information from each channel



Form1

I-8017HW Slot Index: Slot 3

Gain: +/- 10.0 V  Count: 1000  Format: Float

Verify the AI data from each channel...

| | First Data | Min Data | Max Data | Delta | | First Data | Min Data | Max Data | Delta |
|---|---|---|---|---|---|---|---|---|---|
| C0 | 00.0012 | -00.001 | | | | | | | |
| C1 | 00.0000 | 00.0000 | 00.0024 | 00.0024 | C9 | | | | |
| C2 | 00.0000 | -00.0012 | 00.0024 | 00.0036 | C10 | | | | |
| C3 | 00.0012 | -00.0012 | 00.0037 | 00.0049 | C11 | | | | |
| C4 | 00.0000 | 00.0000 | 00.0037 | 00.0037 | C12 | | | | |
| C5 | 00.0012 | -00.0012 | 00.0024 | 00.0036 | C13 | | | | |
| C6 | 00.0012 | -00.0012 | 00.0024 | 00.0036 | C14 | | | | |
| C7 | 00.0012 | -00.0012 | 00.0024 | 00.0036 | C15 | | | | |

Time Ticks: 31

# 3. API for Windows-Based Controllers

The following table lists the functions provided in pac_8017HW.lib for Windows platform.

| Function | Description |
|---|---|
| pac_i8017HW_GetLibVersion | Used to read the version and build information for the currently installed Library |
| pac_i8017HW_GetLibDate | Used to read the build date information for the currently installed Library |
| pac_i8017HW_GetFirmwareVersion | Used to read the firmware (FPGA) version information |
| pac_i8017HW_Init | Used to initialize the module |
| pac_i8017HW_ReadGainOffset_Info | Used to read the calibrated voltage Gain and Offset values |
| pac_i8017HW_Read_mA_GainOffset | Used to read the calibrated currents Gain and Offset values |
| pac_i8017HW_SetLED | Used to set the LEDs |
| pac_i8017HW_GetSingleEndJumper | Used to read the status of the input jumper (Differential or Single-ended mode) |
| pac_i8017HW_ReadAI | Used to read the Analog Input value from a specific channel in float format |
| pac_i8017HW_ReadAIHex | Used to read the Analog Input value from a specific channel in 16-bit hexadecimal format |

## 3.1. pac_i8017HW_GetLibVersion

This function is used to read the version and build information for the Library currently installed on the I-8017 module inserted in a specific slot.

### Syntax

**For Windows (CE and WES)**

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

### Parameters

This function does not require any parameters

### 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++]**

```
short version = pac_i8017HW_GetLibVersion();
```

**[C#]]**

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

## 3.2. pac_i8017HW_GetLibDate

This function is used to read the build date information for the Library currently installed on the I-8017 module inserted in a specific slot

### Syntax

**For Windows (CE and WES)**

```
void pac_i8017HW_GetLibDate(char libDate[]);
```

### Parameters

None

### Return Values

*libDate*

A string indicating the build date of the Library used by the module with anull terminal..

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

### Examples

**[C++]**

```
char* builtDate;
pac_i8017HW_GetLibDate(builtDate);
```

**[C#]**

```
stringbuiltDate;;
builtDate = pac_i8017HW_GetLibDate();
```

## 3.3. pac_i8017HW_GetFirmwareVersion

This function is used to read the firmware (FPGA) version information for the I-8017 module inserted in a specific slot.

### Syntax

#### For Windows (CE and WES)

```
short pac_i8017HW_GetFirmwareVersion (
    int slot,
    short* version
);
```

### Parameters

*slot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

*version*

The firmware version information for the I-8017 module.

### Return Values

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

### Examples

#### [C++]

```
int slot;
short ret, firmware;
ret = pac_i8017HW_GetFirmwareVersion(slot, &firmware);
```

#### [C#]

```
int slot;
Int16 firmware = 0;
Int16 ret = pac_i8017HW_GetFirmwareVersion(slot, ref firmware);
```

## 3.4. pac_i8017HW_Init

This function is used to initialize the I-8017 module inserted in a specified slot, and must be called at least once before using any other function.

### Syntax

#### For Windows (CE and WES)

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

### Parameters

*slot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

### Return Values

Refer to Appendix A: "Error Code Definitions" for more details.

### Examples

#### [C++]

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

#### [C#]

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

## 3.5. 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 Windows (CE and WES)**

```
shortpac_i8017HW_ReadGainOffset_Info (
    int slot,
    int Gain,
    unsigned short* GainValue,
    short* OffsetValue
);
```

### Parameters

*iSlot:*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

*iGain:*

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

*iGainValue:*

Specifies the calibrated Gain value

*iOffsetValue:*

Specifies the calibrated Offset value

### Return Values

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

## Examples

### [C++]

```
intslot,Gain=0;
unsigned short GainValue;
shortOffsetValue;
pac_ i8017HW_ReadGainOffset_Info (slot,Gain,&GainValue,&OffsetValue);
```

### [C#]

```
Intslot,Gain=0;
UInt16 GainValue;
Int16 OffsetValue;
pac8017HW.GainOffset_Info(slot,Gain,refGainValue,refOffsetValue);
```

## 3.6. pac_i8017HW_Read_mA_GainOffset

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

### Syntax

**For Windows (CE and WES)**

```
shortpac_i8017HW_Read_mA_GainOffset (
   int slot,
   shortch ,
   unsigned short* GainValue,
   short* offsetValue
);
```

### Parameters

*iSlot:*

Specifies the slot where the I-8017C/I-9017C-15 module is inserted. Valid range = 0 to 7

*iCh:*

Specifies the channel

Valid range :

I-8017CW = 0 to 7

I-9017-15 = 0 to 14

*iGainValue:*

Specifies the calibrated Gain value

*iOffsetValue:*

Specifies the calibrated Offset value

### Return Values

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

### Examples

[C++]

```
int slot;
shortch;
unsigned short GainValue;
shortOffsetValue;
pac_ i8017HW_Read_mA_GainOffset (slot,ch,&GainValue,&OffsetValue);
```

[C#]

```
Int slot;
Int16 ch;
UInt16 GainValue;
Int16 OffsetValue;
Int16ret = pac8017HW.mA_GainOffset_Info(slot,ch,refGainValue,refOffsetValue);
```

## 3.7. pac_i8017HW_SetLED

This function is used to set the LEDs for the I-8017 module inserted in a specified slot to ON or OFF. The LEDs can also be used to act as an alarm.

### Syntax

#### For Windows (CE and WES)

```
short pac_i8017HW_SetLED(
    int slot,
    unsigned short led
);
```

### Parameters

*slot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

*led*

Specifies the LED to be set. Valid range = 0 to 0xFFFF

### Return Values

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

### Examples

**[C++]**

```
int slot;
unsigned short led;
short ret = pac_i8017HW_Init(int slot);
if (ret == 0)
pac_i8017HW_SetLED(slot, led);
```

**[C#]**

```
int slot;
UInt16 led;
Int16ret = pac8017HW.Init(slot);
If (ret == 0)
pac8017HW.SetLED(slot, led);
```

## 3.8. pac_i8017HW_GetSingleEndJumper

This function is used to read whether the input jumper for the I-8017 module inserted in a specified slot is set to either Differential or Single-ended mode.

### Syntax

**For Windows (CE and WES)**

```
short pac_i8017HW_GetSingleEndJumper(int slot);
```

### Parameters

*slot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

### Return Values

The status of the input jumper

0: Differential Mode

1: Single-ended Mode

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

### Examples

**[C++]**

```
int slot=1;
short Jumper = pac_i8017HW_GetSingleEndJumper(slot);
```

**[C#]**

```
int slot;
Int16 Jumper = pac8017HW. SingleEndJumper(slot);
```

## 3.9. pac_i8017HW_ReadAI

This function is used to read the Analog Input value in float format from a specific channel of the I-8017 module inserted in a specified slot according to a given Gain value

### Syntax

#### For Windows (CE and WES)

```
float pac_i8017HW_ReadAI(
    intiSlot,
    intiChannel,
    intiGain
);
```

### Parameters

*iSlot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

*iChannel*

Specifies the channel number

0 to 7 if using Differential Mode

0 to 15 if using Single-ended Mode

*iGain*

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

### Return Values

Return (float): he analog input value in float format.

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

### Examples

#### [C++]

```
int slot=1, ch=0, gain=1;
float data;
pac_i8017HW_Init(slot);
data = pac_i8017HW_ReadAI(slot,ch,gain);
```

#### [C#]

```
int slot=1, ch=0, gain=1;
float data;
pac8017HW.Init (slot);
data= pac8017HW. ReadAI(slot, ch, gain);
```

## 3.10. pac_i8017HW_ReadAIHex

This function is used to read the Analog Input value in 16-bit hexadecimal format from a specific channel of the I-8017 module inserted in a specified slot according to a given Gain value.

### Syntax

#### For Windows (CE and WES)

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

### Parameters

*iSlot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

*iChannel*

Specifies the channel number

0 to 7 if using Differential Mode

0 to 15 if Single-ended Mode

*iGain:*

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

### Return Values

Return (short): he analog input value in hexadecimal format.

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

## Examples

**[C++]**

```
int slot=1, ch=0, gain=1;
short data;
pac_i8017HW_Init(slot);
data = pac_i8017HW_ReadAIHex(slot,ch,gain);
```

**[C#]**

```
int slot=1, ch=0, gain=1;
float data;
pac8017HW.Init (slot);
data= pac8017HW. ReadAIHex(slot, ch, gain);
```

## Note

the I-8017HW, I-8017DW and I-8017HCW modules use a 14-bit AD chip, so it is more convenient for the pac_i8017HW_ReadHex function to return 16-bit data when the user needs to scale the hexadecimal data. Consequently, there will be least two bits of invalid data.

The following code can be used to convert 16-bit data to 14-bit.

```
short Convert16To14(short bit16Data)
{
    short bit14Data= bit16Data >> 2;
    bit14Data &= 0x3fff;
    return bit14Data;
}
```

# 4. API for MiniOS7-based Controller

The following table lists the functions provided in 8017HW.lib for MiniOS7 platform.

| Function | Description |
|---|---|
| i8017H_GetLibVersion | Used to read the version and build information for the currently installed Library |
| i8017H_GetLibDate | Used to read the built date information for the currently installed Library |
| i8017H_GetFirmwareVersion | Used to read the firmware (FPGA) version information |
| i8017H_Init | Used to initialize the module |
| i8017H_ReadGainOffset_Info | Used to read the calibrated Gain and Offset values |
| i8017H_GetSingleEndJumper | Used to read the status of the input jumper (Differential or Single-ended Mode) |
| i8017H_SetLED | Used to set the LEDs |
| i8017H_ReadAI | Used to read the Analog Input value from a specific channel in float format |
| i8017H_ReadAIHex | Used to read the Analog Input value from a specific channel in 16-bit hexadecimal format |
| i8017H_Set_ChannelGainMode | Used to set the MUX configuration |
| i8017H_Get_AD_FValue | Used to read the A/D value in float format |
| i8017H_Get_AD_HValue | Used to read the A/D value in 14-bit hexadecimal format |
| i8017H_AD_Polling | Used to set the polling mode to sample A/D values from a single channel |
| i8017H_AD_TimerINT | Used to set the timer interrupt to sample A/D values from a single channel |
| i8017H_AD_TimerINT_Scan | Used to set the timer interrupt to sample A/D values from all channels |

## 4.1. i8017H_GetLibVersion

This function is used to read the version and build information for the Library currently installed on the I-8017 module inserted in a specific slot

### Syntax

**For MiniOS7 (i-8000/iP-8000)**

```
short i8017H_GetLibVersion(void);
```

### Parameters

This function does not require any parameters.

### Return Values

The version number and build information for the Library used by the module. Example: 0x106; = Rev:1.0.6

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

### Examples

**[C++]**

```
short version = i8017H_GetLibVersion();
```

## 4.2. i8017H_GetLibDate

This function is used to read the build date information for the Library currently installed on the I-8017 module inserted in a specific slot

### Syntax

#### For MiniOS7 (i-8000/iP-8000)

```
void i8017H_GetLibDate(char libDate[]);
```

### Parameters

*libDate*

A string indicating the build date of the Library used by the module.

### Return Values

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

### Examples

#### [C++]

```
char* builtDate;
i8017H_GetLibDate(&builtDate);
```

## 4.3. i8017H_GetFirmwareVersion

This function is used to read the firmware (FPGA) version information for the I-8017 module inserted in a specific slot.

Syntax

**For MiniOS7 (i-8000/iP-8000)**

```
short i8017H_GetFirmwareVersion(
    intiSlot,
    short* firmware
);
```

### Parameters

*iSlot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

*firmware*

The firmware version information for the I-8017HW module.

### Return Values

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

### Examples

**[C++]**

```
int slot;
short ret, firmware;
ret = i8017H_GetFirmwareVersion(slot, &firmware);
```

## 4.4. i8017H_Init

This function is used to initialize the I-8017 module inserted in a specific slot, and must be called at least once before using any other function.

### Syntax

**For MiniOS7 (i-8000/iP-8000)**

```
void i8017H_Init(intiSlot);
```

### Parameters

*iSlot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

### Return Values

None

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

### Examples

**[C++]**

```
int slot;
short ret = i8017H_Init(slot);
```

## 4.5. i8017H_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 (i-8000/iP-8000)**

```
short i8017H_ReadGainOffset_Info
(
    intiSlot,
    intiGain,
    unsigned short* iGainValue,
    short* iOffsetValue
);
```

### Parameters

*iSlot:*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

*iGain:*

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

*iGainValue:*

Specifies the calibrated Gain value

*iOffsetValue:*

Specifies the calibrated Offset value

### Return Values

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

## Examples

### [C++]

```
int slot, ch, gain;
unsigned short cgain;
shortcoffset;
i8017H_Init(slot);
short ret = ReadGainOffset_Info (slot,ch,gain,&cgain,&coffset);
```

## 4.6. i8017H_GetSingleEndJumper

This function is used to read whether the input jumper for the I-8017 module inserted in a specified slot is set to either Differential or Single-ended mode.

Syntax

**For MiniOS7 (i-8000/iP-8000)**

```
short i8017H_GetSingleEndJumper(
    intiSlot,
    short* selectJumper
);
```

### Parameter

*iSlot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

*selectJumper*

The status of the jumper

0: Differential Mode

1: Single-ended Mode

### Return Values

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

### Examples

**[C++]**

```
int slot;
short Jumper = i8017H_GetSingleEndJumper(slot);
```

## 4.7. i8017H_SetLED

This function is used to set the LEDs for the I-8017 module inserted in a specified slot to ON or OFF. The LEDs can also be used to act as an alarm.

### Syntax

**For MiniOS7 (i-8000/iP-8000)**

```
void i8017H_SetLED(intiSlot, unsigned short iLedValue);
```

### Parameters

*iSlot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

*iLedValue*

Specifies the LED to be set. Valid range = 0 to 0xFFFF

### Return Values

None

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

### Examples

**[C++]**

```
int slot;
unsigned short led;
i8017H_Init(int slot);
i8017H_SetLED(slot, led);
```

## 4.8. i8017H_ReadAI

This function is used to read the Analog Input value in float format from a specific channel of the I-8017 module inserted in a specified slot according to a given Gain value.

### Syntax

**For MiniOS7 (i-8000/iP-8000)**

```
short i8017H_ReadAI
(
    intiSlot,
    intiChannel,
    intiGain,
    float* fValue
);
```

### Parameters

*iSlot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

*iChannel*

Specifies the channel number

0 to 7 if using Differential Mode

0 to 15 if using Single-ended Mode

*iGain*

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

*fValue*

The analog input data in float format.

## Return Values

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

## Examples

### [C++]

```
int slot, ch, gain;
float data;
i8017H_Init(slot);
short ret = i8017H_ReadAI(slot,ch,gain,&data);
```

## 4.9. i8017H_ReadAIHex

This function is used to read the Analog Input value in 16-bit hexadecimal format from a specific channel of the I-8017 module inserted in a specified slot according to a given Gain value.

### Syntax

**For MiniOS7 (i-8000/iP-8000)**

```
short i8017H_ReadAIHex
(
    intiSlot,
    intiChannel,
    intiGain,
    short* iValue
);
```

### Parameters

*iSlot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

*iChannel*

Specifies the channel number

0 to 7 if using Differential Mode

0 to 15 if using Single-ended Mode

*iGain*

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

*iValue*

The Analog Input value in 16-bit hexadecimal format

## Return Values

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

## Examples

### [C++]

```
int slot, ch, gain;
short* data;
i8017H_Init(slot);
short ret = i8017H_ReadAIHex (slot,ch,gain,&data);
```

## 4.10. i8017H_Set_ChannelGainMode

This function is used to configure the Analog Input channel range for the I-8017 module inserted in a specified slot prior to using the ADC (analog to digital converter).

**Tips & Warnings**

⚠

1. If the channel, range or mode to be sampled is different to a previous sampling, this function must be called first to change the MUX configuration.
2. Note that calling this function only needs to be considered when using the following functions:

   1. i8017H_Get_AD_FValue

   2. i8017H_Get_AD_FValue

   This is because other functions that are used to sample multiple data signals already include a function to configure the MUX.

### Syntax

**For MiniOS7 (i-8000/iP-8000)**

```
void i8017H_Set_ChannelGainMode
(
   intiSlot,
   intiChannel,
   intiGain,
   intiMode
);
```

## Parameters

*iSlot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

*iChannel*

Specifies the channel number

0 to 7 if using Differential Mode

0 to 15 if using Single-ended Mode

*iGain*

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

*iMode*

Specifies the Mode to be used

0 = Polling Mode

1 = Timer Interrupt Mode

(Use 0 for the i8017_Get_AD_Value and i8017_Get_AD_Value_Hex functions)

## Return Values

None

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

## Examples

### [C++]

```
int slot, ch, gain,mode;
i8017H_Init(slot);
short ret =i8017H_Set_ChannelGainMode (slot,ch,gain,mode);
```

# 4.11. i8017H_Get_AD_FValue

This function is used to read the A/D value for both voltage and current from the I-8017 module inserted in a specified slot in float format.

### Syntax

**For MiniOS7 (i-8000/iP-8000)**

```
float i8017H_Get_AD_FValue(intiGain);
```

### Parameters

*iGain*

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

### Return Values

Refer to Appendix A: "Error Code Definitions" for more details.

### Examples

**[C++]**

```
int slot, gain;
i8017H_Init(slot);
float data = i8017H_Get_AD_FValue (gain);
```

## 4.12. i8017H_Get_AD_HValue

This function is used to read the A/D value for both voltage and current from the I-8017 module inserted in a specified slot in hexadecimal format.

The resolution for the I-8017HW, I-8017DW and I-8017HCW modules is 14-bit and uses 14-bit 2's Complement format to store values.

When the A/D value is between 0000 and 1FFF → 0 to + max. value

When the A/D value is between 2000 and 3FFF → - max. value to 0 (a little less than 0)

The algorithm used to convert a 14-bit integer value (iValue) to a float value (fValue) is as follows:

fValue = iValue/8192*span,

fValue>= 0, iValue between 0000 and 1FFF

fValue = (8192-iValue)/8192*span,

fValue< 0, iValue between 2000 and 3FFF

Syntax

**For MiniOS7 (i-8000/iP-8000)**

```
int i8017H_Get_AD_HValue(void);
```

### Parameters

None

### Return Values

None

Refer to Appendix A: "Error Code Definitions" for more details.

### Examples

**[C++]**

```
int slot;
i8017H_Init(slot);
int data = i8017H_Get_AD_HValue();
```

## 4.13. i8017H_AD_Polling

This function is used to set the polling mode to sample the A/D values from a single channel of the I-8017 module inserted in a specified slot.

**Tips & Warnings**

The data is an uncalibrated 16-bit value.

ARRAY_HEX_TO_FLOAT or HEX_TO_FLOAT can be used to calibrate the data and convert it to a float value.

### Syntax

**For MiniOS7 (i-8000/iP-8000)**

```
int i8017H_AD_Polling
(
    intiSlot,
    intiChannel,
    intiGain,
    intiDataCount,
    int *iDataPointer
);
```

## Parameters

*iSlot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

*iChannel*

Specifies the channel number

0 to 7 if using Differential Mode

0 to 15 if using Single-ended Mode

*iGain*

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

*iDataCount*

Specifies the data length to read in, Valid range = 1 to 8192

*iDataPointer*

Specifies the pointer to the data buffer

## Return Values

None

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

## 4.14. i8017H_AD_TimerINT

This function is used to set the timer interrupt to sample the A/D values from a single channel of the I-8017 module inserted in a specified slot.

**Tips & Warnings**

⚠ The data is an uncalibrated 16-bit value.

ARRAY_HEX_TO_FLOAT or HEX_TO_FLOAT can be used to calibrate the data and convert it to a float value.

### Syntax

**For MiniOS7 (i-8000/iP-8000)**

```
int i8017H_AD_TimerINT
(
    intiSlot,
    intiChannel,
    intiGain,
    intiDataCount,
    unsignedintiSampleCLK,
    int *iDataPointer
);
```

## Parameters

*iSlot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

*iChannel*

Specifies the channel number

0 to 7 if using Differential Mode

0 to 15 if using Single-ended Mode

*iGain*

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

*iDataCount*

Specifies the data length to read in, Valid range = 1 to 8192

*iSampleCLK*

Specifies the sampling rate ranged = 200 to 50K (units: Hz)

*iDataPointer*

Specifies the pointer to the data buffer

## Return Values

None

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

# 4.15. i8017H_AD_TimerINT_Scan

This function is used to set the timer interrupt to sample the A/D values from all 8 channels of the I-8017 module inserted in a specified slot.

### Tips & Warnings

⚠ The data is an uncalibrated 16-bit value.

ARRAY_HEX_TO_FLOAT or HEX_TO_FLOAT can be used to calibrate the data and convert it to a float value.

## Syntax

### For MiniOS7 (i-8000/iP-8000)

```
int i8017H_AD_TimerINT_Scan
(
    intiSlot,
    intiGain,
    intiDataCount,
    unsignedintiSampleCLK,
    int *iDataPointer
);
```

## Parameters

*iSlot*

Specifies the slot where the I-8017 module is inserted. Valid range = 0 to 7

*iChannel*

Specifies the channel number

0 to 7 if using Differential Mode

0 to 15 if using Single-ended Mode

*iGain*

Specifies the input range

0: +/- 10.0V

1: +/- 5.0V

2: +/- 2.5V

3: +/- 1.25V

4: +/- 20mA

*iDataCount*

Specifies the data length to read in, valid length is ranged from 1 to 8192

*iSampleCLK*

Specifies the sampling rate ranged = 200 to 50K (units: Hz)

*iDataPointer*

Specifies the pointer to the data buffer

The data stored in the buffer is arranged as follows:

Differential Input Mode:

ch0 ch1 ch2...ch7 ch0 ch1....ch7 ch0 ch1 ...

Single-ended Input Mode:

ch0 ch1 ch2...ch7 ch0 ch1....ch15 ch0 ch1 ...

## Return Values

None

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

# 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<br>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<br>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)<br>CD:\napdos\wp-8x4x_ce50\Demo\WinPAC\DOTNET\C#.NET\PAC_IO\Local\pac_i8017HW_Dotnet\pac_i8017HW_calibration<br>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<br>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<br>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<br>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<br>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

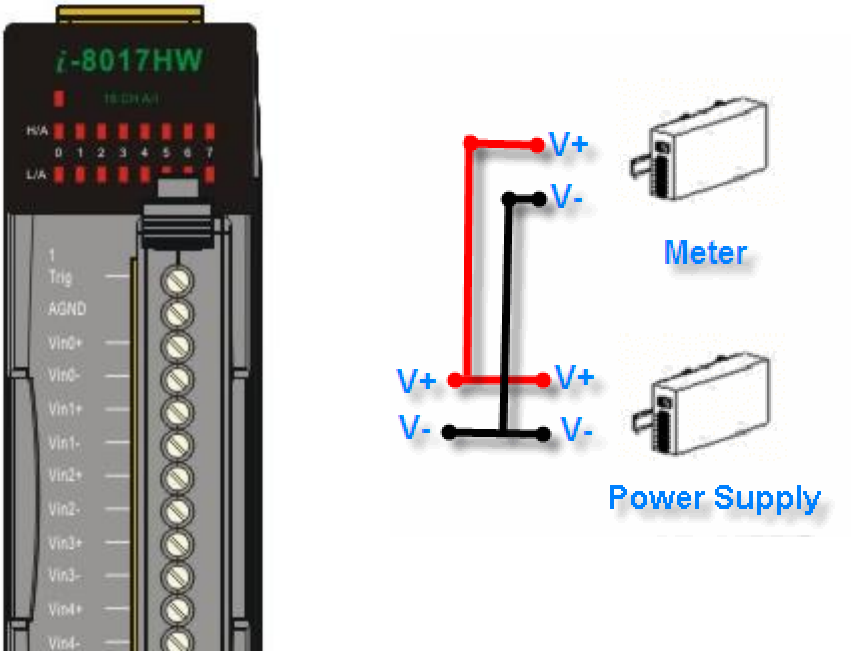
**Step 1.** Repeat Steps 1 to 3 as described in the Quick Start guide.

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.

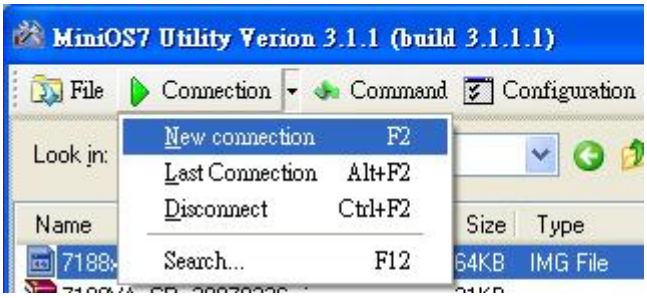


**Step 2.** Launch the MiniOS7 Utility on the Host PC. Upload the calibration program to the control unit and execute it.

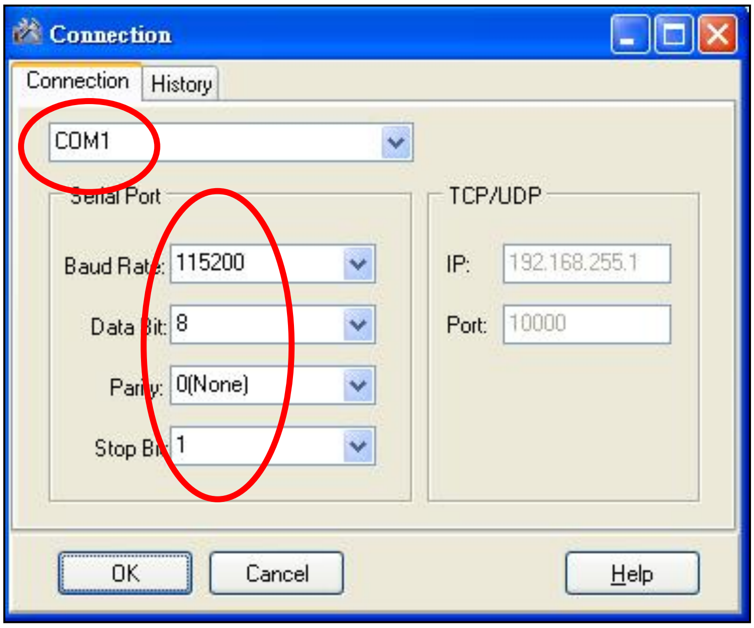
The MiniOS7 Utility can be downloaded from the web site shown below. Select the appropriate calibration program for your controller.

- 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.



COM1

Serial Port

Baud Rate: 115200

Data Bit: 8

Parity: 0(None)

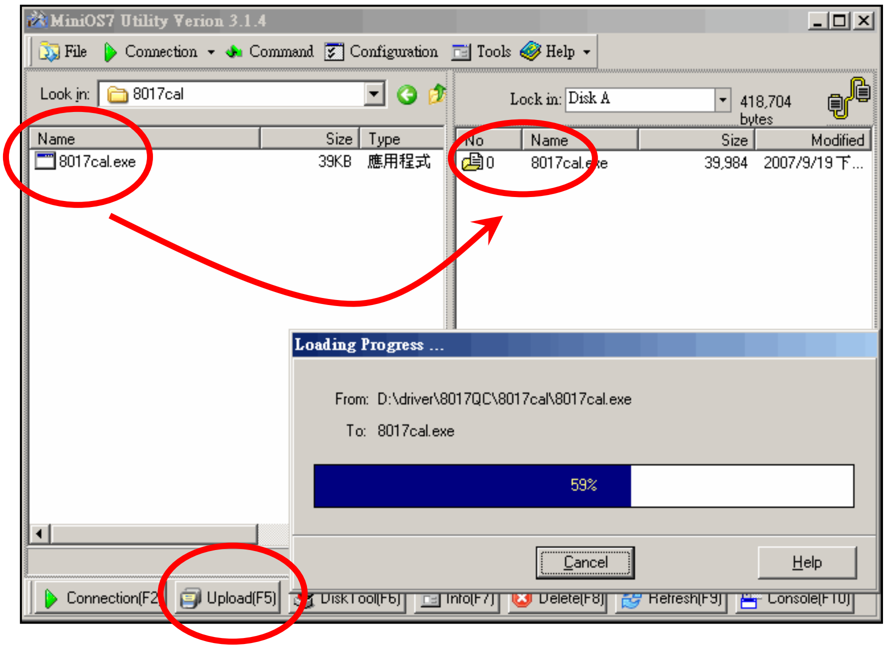
Stop Bit: 1

TCP/UDP

IP: 192.168.255.1

Port: 10000

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.



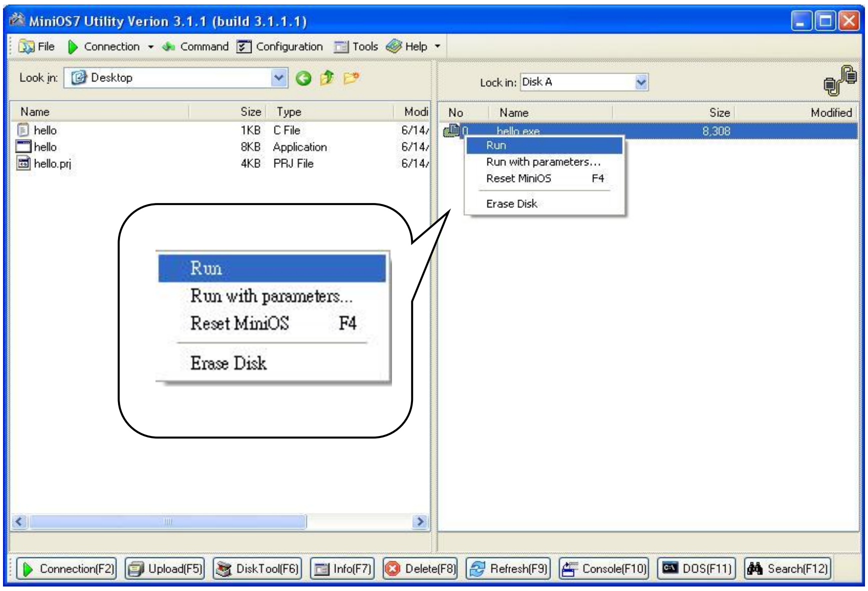
Loading Progress ...

From: D:\driver\8017QC\8017cal\8017cal.exe

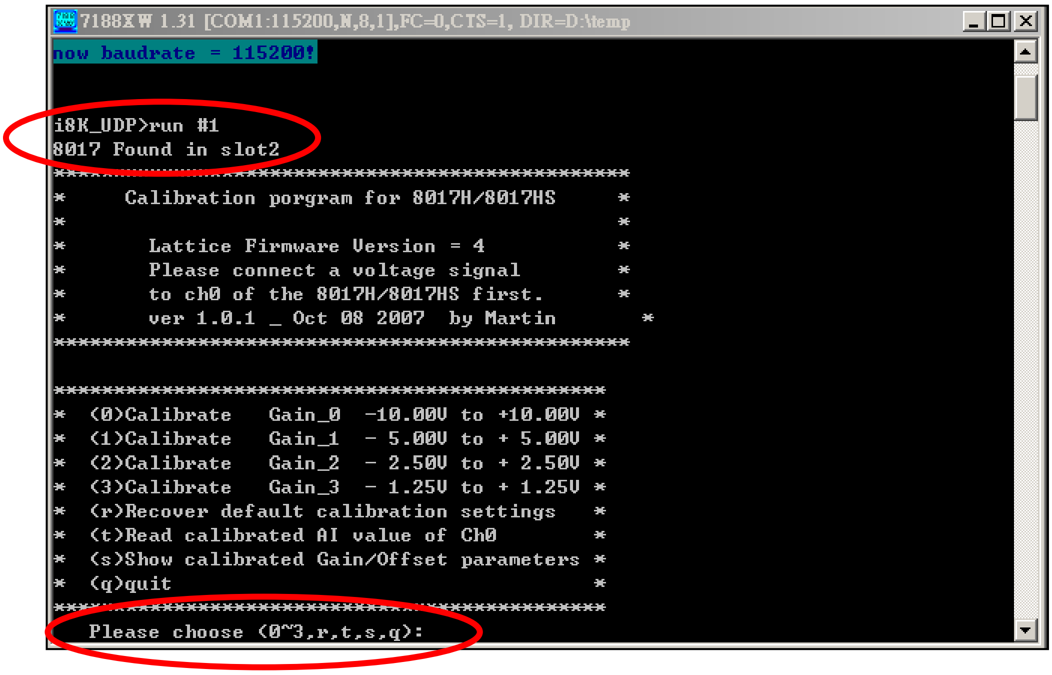
To: 8017cal.exe

59%

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.



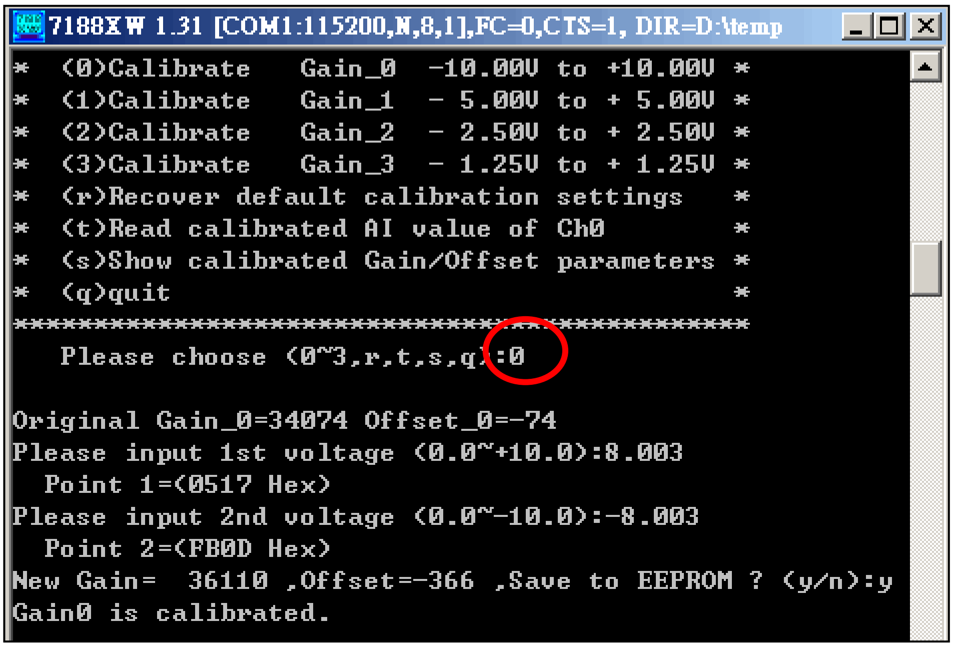
```
now baudrate = 115200!


i8K_UDP>run #1
8017 Found in slot2
************************************************
*     Calibration porgram for 8017H/8017HS     *
*                                              *
*       Lattice Firmware Version = 4           *
*       Please connect a voltage signal        *
*       to ch0 of the 8017H/8017HS first.      *
*       ver 1.0.1 _ Oct 08 2007  by Martin       *
************************************************

**********************************************
*  (0)Calibrate   Gain_0  -10.00V 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):
```

**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.



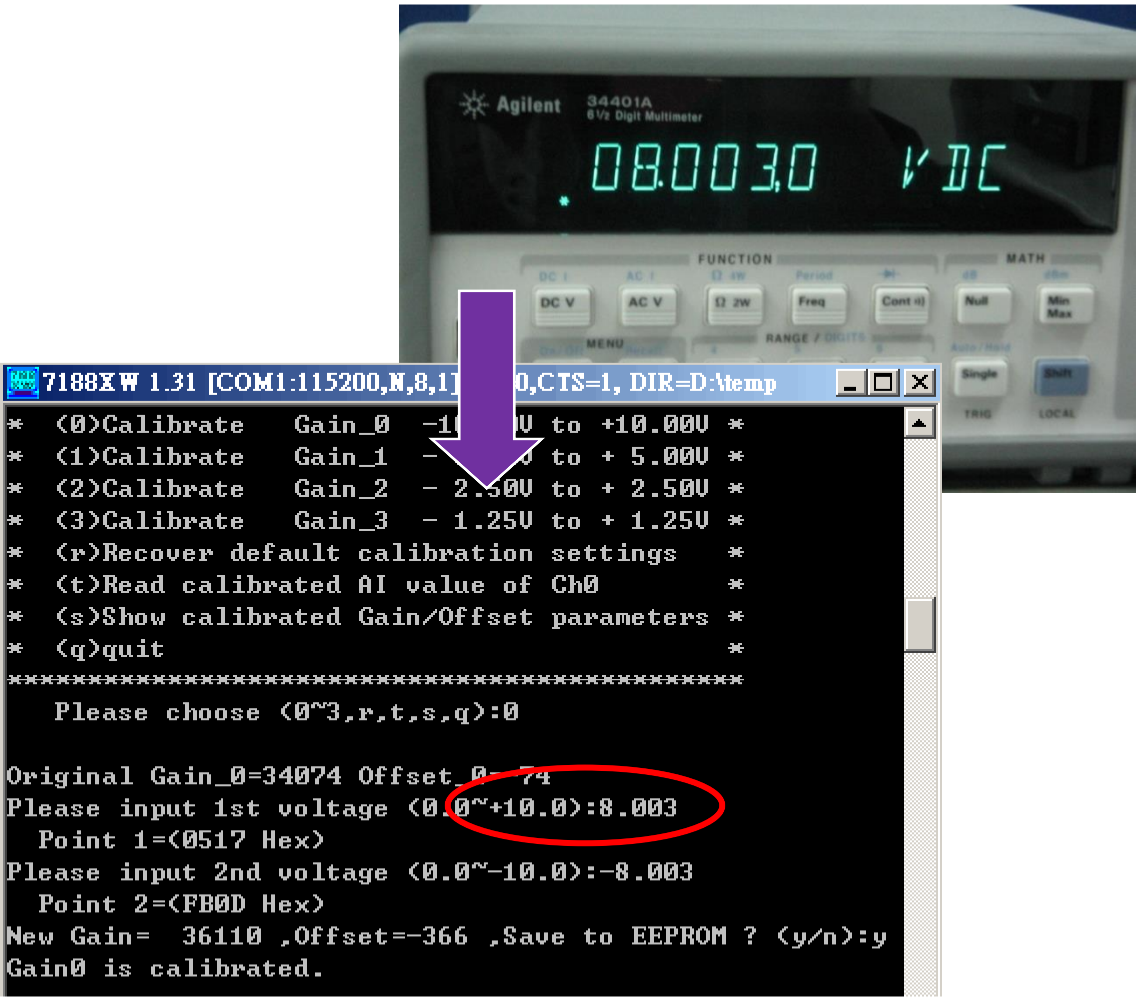
```
*  (0)Calibrate   Gain_0  -10.00V to +10.00V *
*  (1)Calibrate   Gain_1  - 5.00V to + 5.00V *
*  (2)Calibrate   Gain_2  - 2.50V to + 2.50V *
*  (3)Calibrate   Gain_3  - 1.25V to + 1.25V *
*  (r)Recover default calibration settings   *
*  (t)Read calibrated AI value of Ch0        *
*  (s)Show calibrated 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.
```

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.

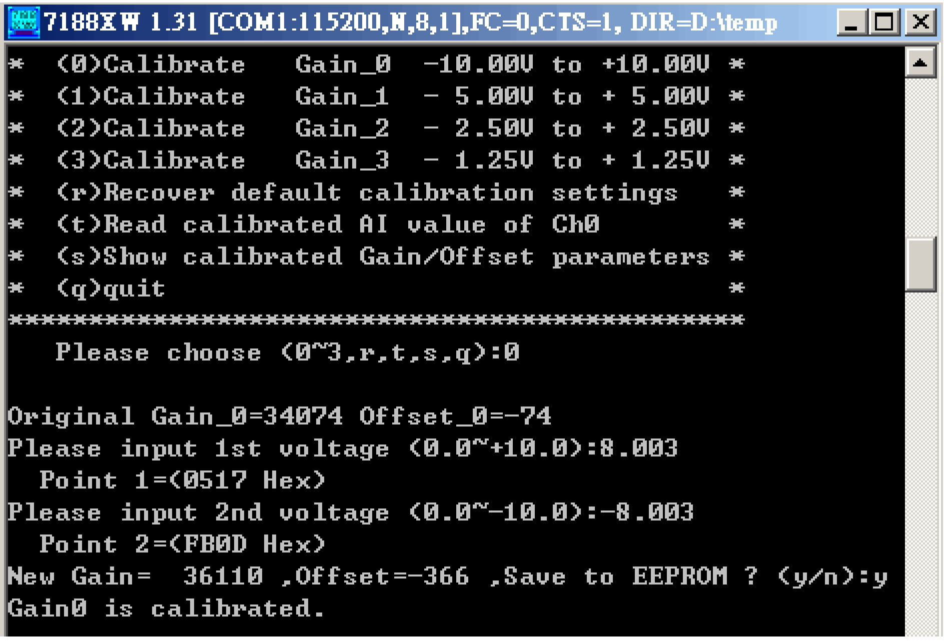


```
*  (0)Calibrate   Gain_0  -10.00V to +10.00V *
*  (1)Calibrate   Gain_1  - 5.00V to + 5.00V *
*  (2)Calibrate   Gain_2  - 2.50V to + 2.50V *
*  (3)Calibrate   Gain_3  - 1.25V to + 1.25V *
*  (r)Recover default calibration settings   *
*  (t)Read calibrated AI value of Ch0        *
*  (s)Show calibrated 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 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



```
*  (0)Calibrate   Gain_0  -10.00V to +10.00V *
*  (1)Calibrate   Gain_1  - 5.000V to + 5.000V *
*  (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.
```

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.

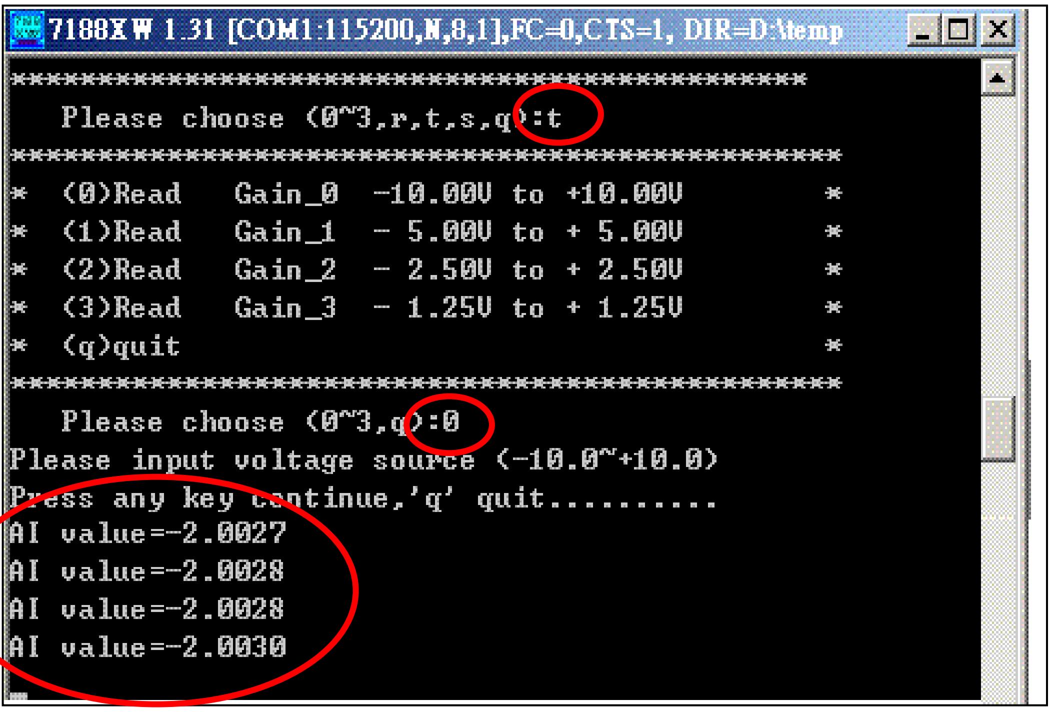
The calibration for the -10 V to +10 V input range is now complete.

## 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).

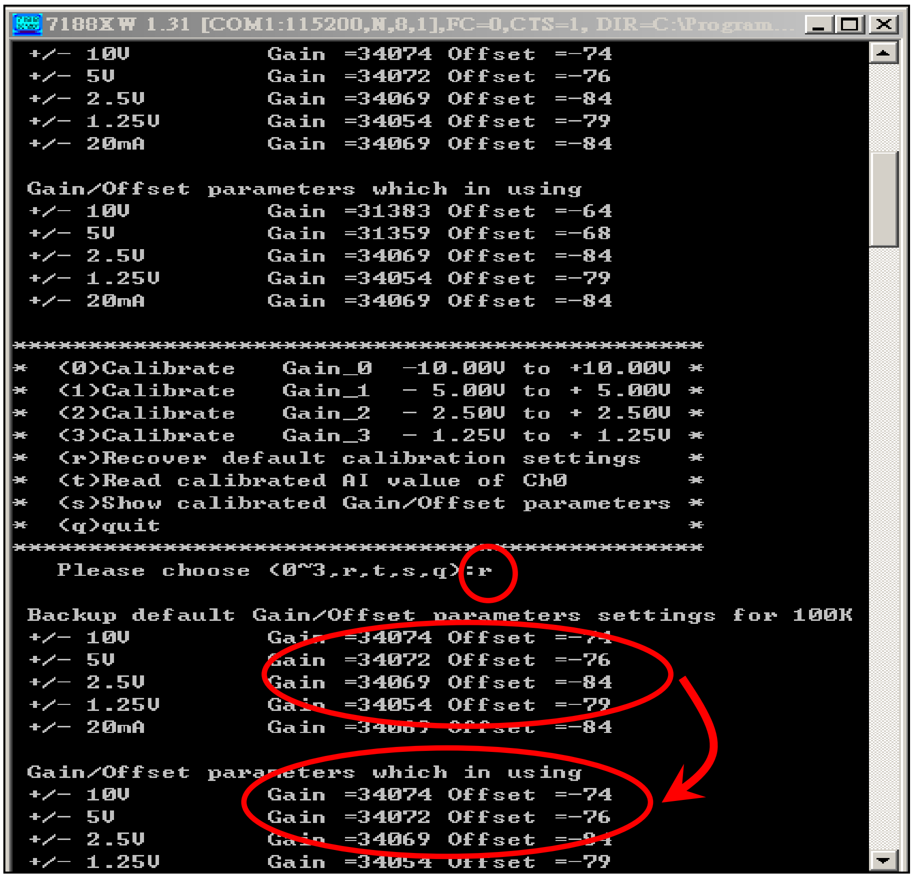
**Step 3.** Confirm that the values displayed for channel 0 are correct.



```
***********************************************
   Please choose (0~3,r,t,s,q):t
*************************************************
*  (0)Read   Gain_0  -10.00V to +10.00V        *
*  (1)Read   Gain_1  - 5.00V to + 5.00V        *
*  (2)Read   Gain_2  - 2.50V to + 2.50V        *
*  (3)Read   Gain_3  - 1.25V to + 1.25V        *
*  (q)quit                                     *
*************************************************
   Please choose (0~3,q):0
Please input voltage source (-10.0~+10.0)
Press any key continue,'q' quit..........
AI value=-2.0027
AI value=-2.0028
AI value=-2.0028
AI value=-2.0030
```

## 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.



```
7188XW 1.31 [COM1:115200,N,8,1],FC=0,CTS=1, DIR=C:\Program...
 +/- 10V             Gain =34074 Offset =-74
 +/- 5V              Gain =34072 Offset =-76
 +/- 2.5V            Gain =34069 Offset =-84
 +/- 1.25V           Gain =34054 Offset =-79
 +/- 20mA            Gain =34069 Offset =-84

 Gain/Offset parameters which in using
 +/- 10V             Gain =31383 Offset =-64
 +/- 5V              Gain =31359 Offset =-68
 +/- 2.5V            Gain =34069 Offset =-84
 +/- 1.25V           Gain =34054 Offset =-79
 +/- 20mA            Gain =34069 Offset =-84

**********************************************
*  (0)Calibrate    Gain_0  -10.00V 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):r

 Backup default Gain/Offset parameters settings for 100K
 +/- 10V             Gain =34074 Offset =-74
 +/- 5V              Gain =34072 Offset =-76
 +/- 2.5V            Gain =34069 Offset =-84
 +/- 1.25V           Gain =34054 Offset =-79
 +/- 20mA            Gain =34069 Offset =-84

 Gain/Offset parameters which in using
 +/- 10V             Gain =34074 Offset =-74
 +/- 5V              Gain =34072 Offset =-76
 +/- 2.5V            Gain =34069 Offset =-84
 +/- 1.25V           Gain =34054 Offset =-79
```

## 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.

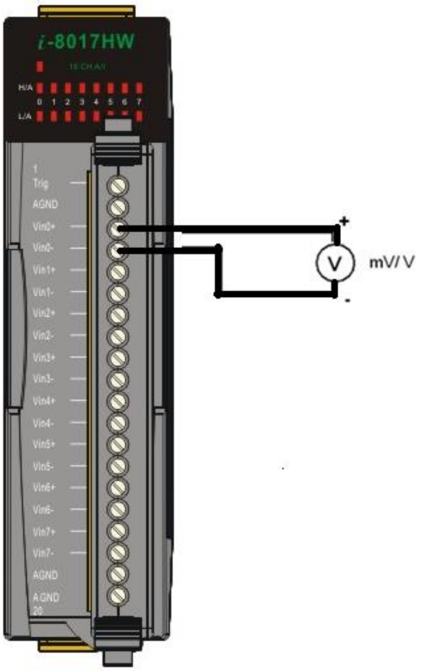
2. If you wish to perform calibration using ±20 mA, select ±2.5 V instead as both types use the same gain and offset values.

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.

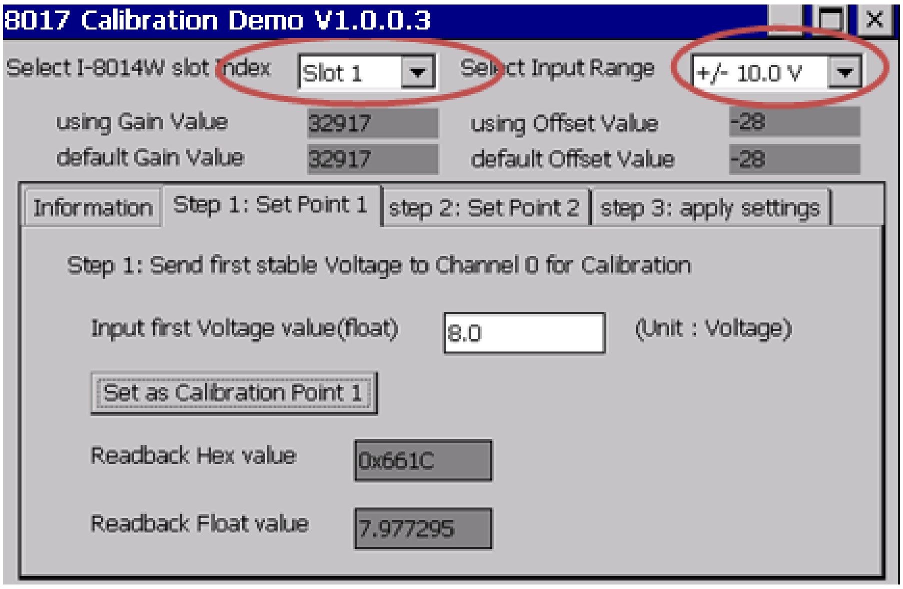
**Step 4.** Launch the pac_i8017W_Calibration.exe executable file on the controller to display the Calibration dialog box.

(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.



8017 Calibration Demo V1.0.0.3

Select I-8014W slot index: Slot 1

Select Input Range: +/- 10.0 V

using Gain Value: 32917

using Offset Value: -28

default Gain Value: 32917

default Offset Value: -28

Step 1: Send first stable Voltage to Channel 0 for Calibration

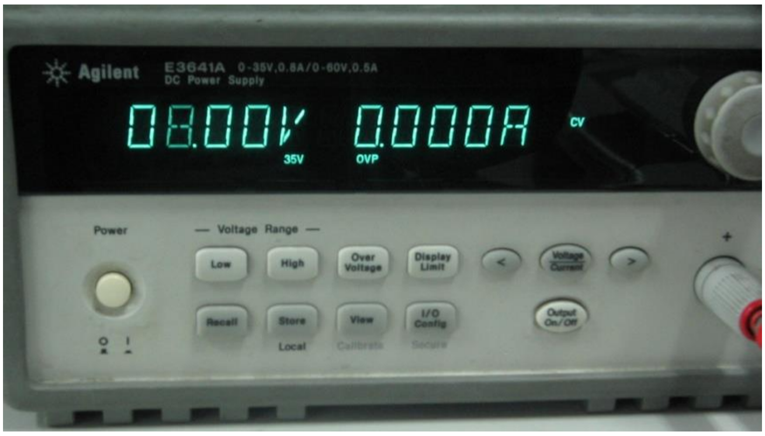
Input first Voltage value(float): 8.0 (Unit : Voltage)

Readback Hex value: 0x661C

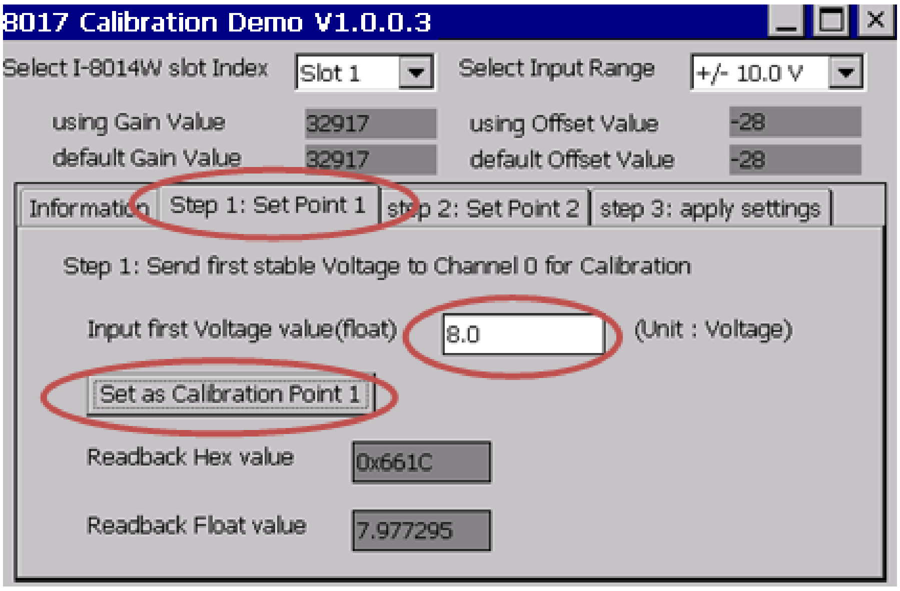
Readback Float value: 7.977295

**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.



8017 Calibration Demo V1.0.0.3

Select I-8014W slot Index: Slot 1

Select Input Range: +/- 10.0 V

using Gain Value: 32917

using Offset Value: -28

default Gain Value: 32917

default Offset Value: -28

Step 1: Send first stable Voltage to Channel 0 for Calibration

Input first Voltage value(float): 8.0 (Unit : Voltage)

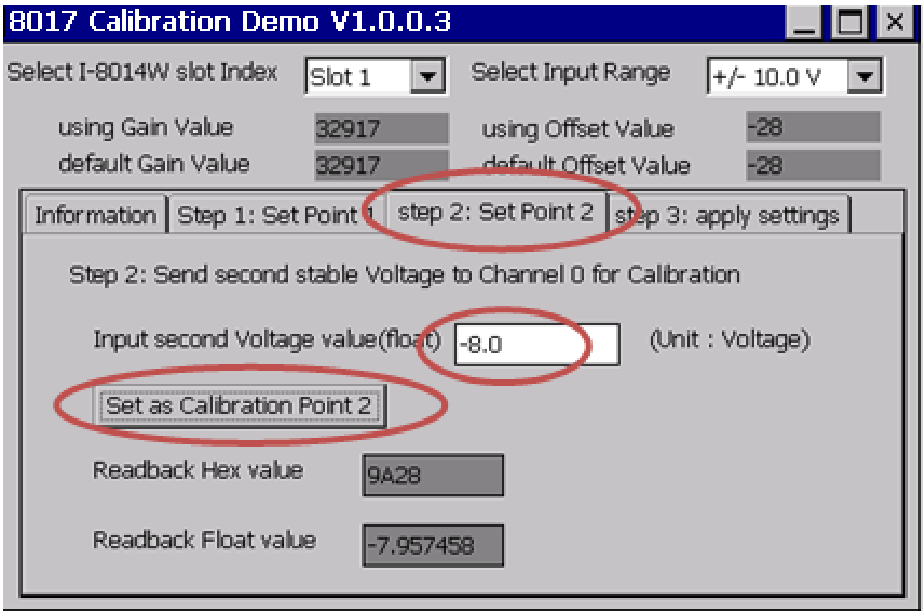
Readback Hex value: 0x661C

Readback Float value: 7.977295

**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.



8017 Calibration Demo V1.0.0.3

Select I-8014W slot Index: Slot 1

Select Input Range: +/- 10.0 V

using Gain Value: 32917

using Offset Value: -28

default Gain Value: 32917

default Offset Value: -28

Step 2: Send second stable Voltage to Channel 0 for Calibration

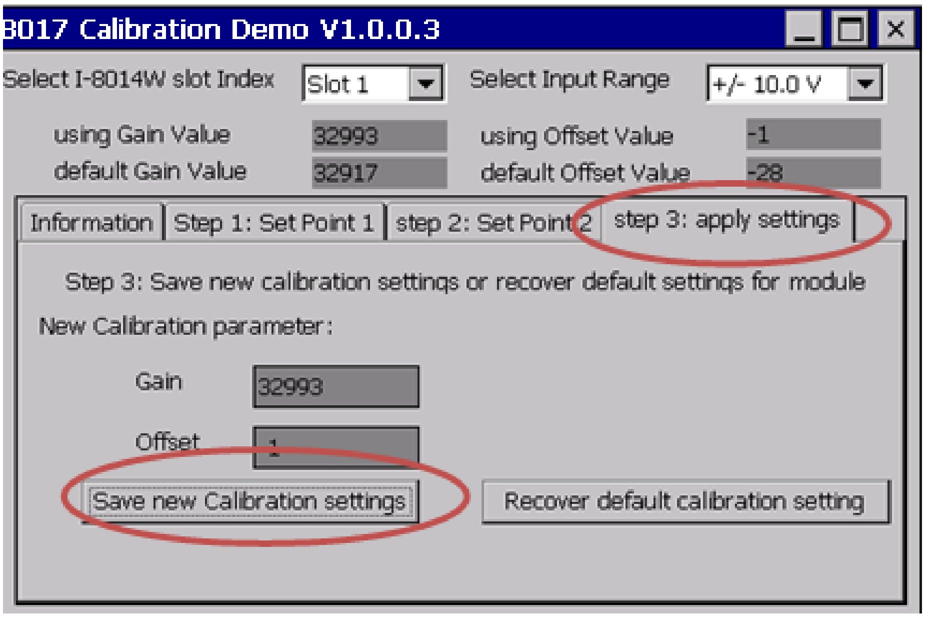
Input second Voltage value(float): -8.0 (Unit : Voltage)

Readback Hex value: 9A28

Readback Float value: -7.957458

**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.



8017 Calibration Demo V1.0.0.3

Select I-8014W slot Index: Slot 1

Select Input Range: +/- 10.0 V

using Gain Value: 32993

default Gain Value: 32917

using Offset Value: -1

default Offset Value: -28

Step 3: Save new calibration settings or recover default settings for module

New Calibration parameter:

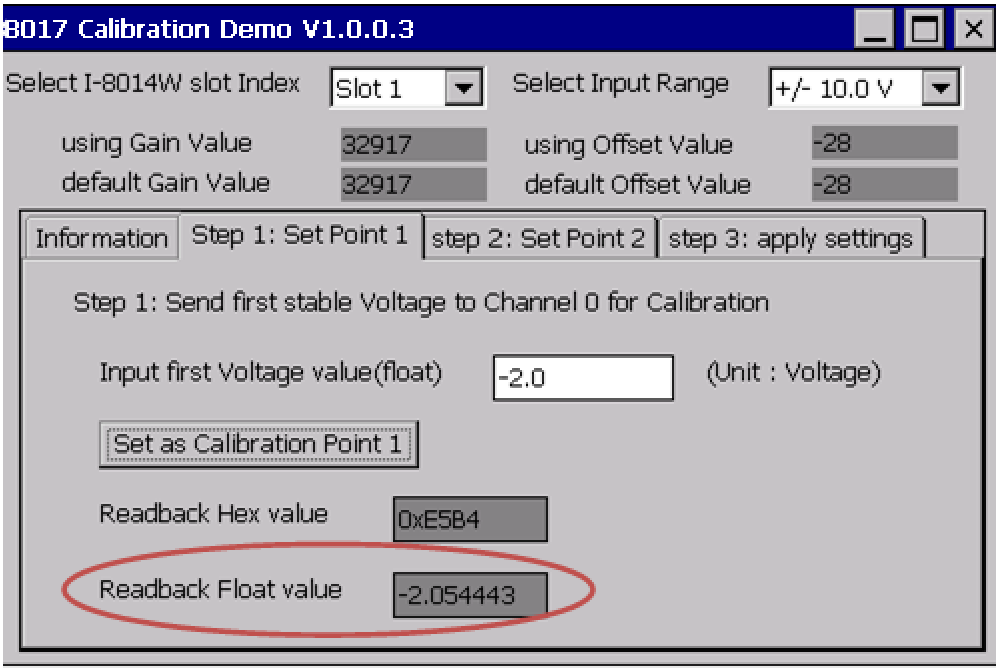
Gain: 32993

Offset: -1

### 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:



8017 Calibration Demo V1.0.0.3

Select I-8014W slot Index: Slot 1

Select Input Range: +/- 10.0 V

using Gain Value: 32917

default Gain Value: 32917

using Offset Value: -28

default Offset Value: -28

Step 1: Send first stable Voltage to Channel 0 for Calibration

Input first Voltage value(float): -2.0 (Unit : Voltage)

Readback Hex value: 0xE5B4

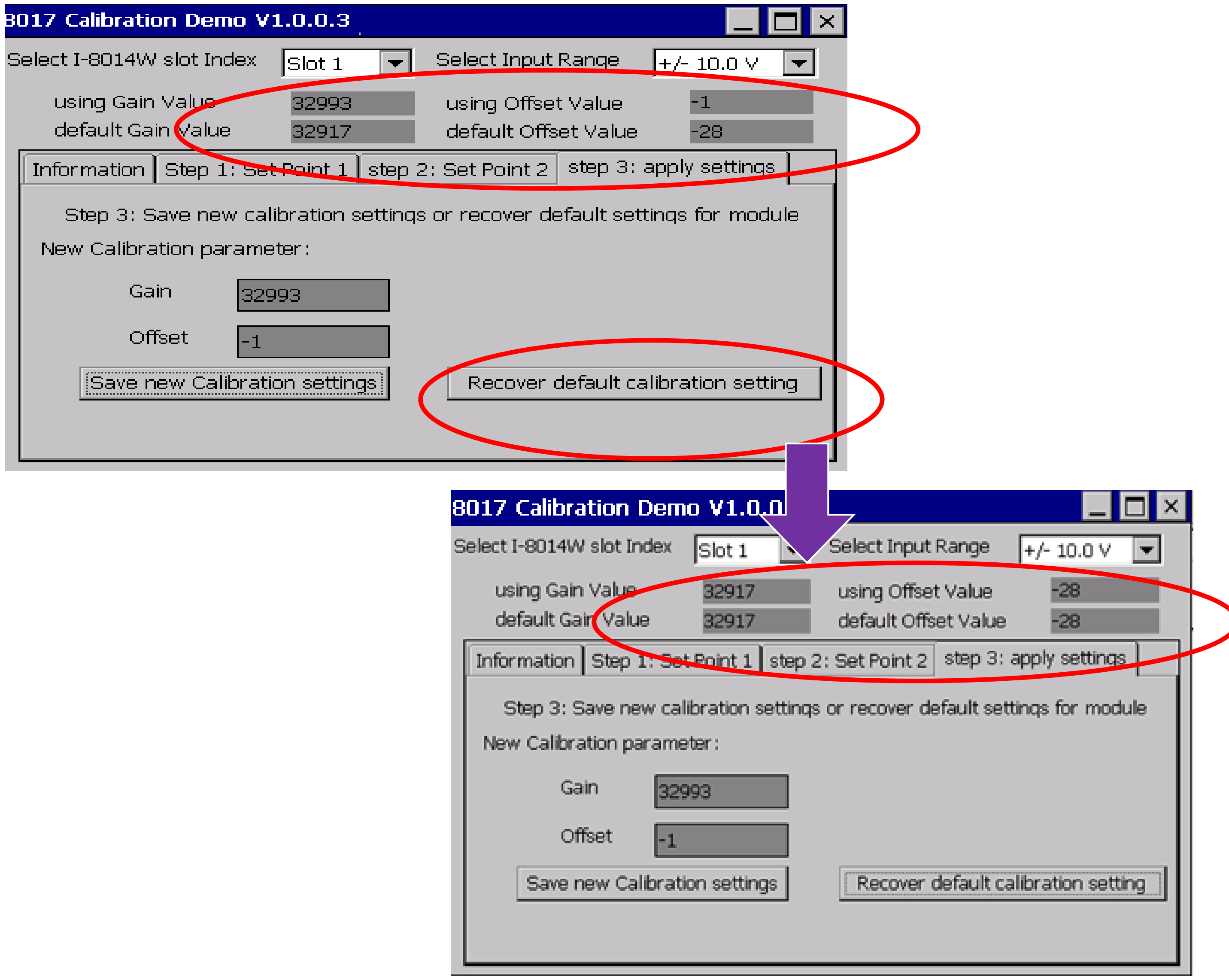
Readback Float value: -2.054443

## 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:



8017 Calibration Demo V1.0.0.3

Select I-8014W slot Index: Slot 1
Select Input Range: +/- 10.0 V

using Gain Value: 32993
default Gain Value: 32917
using Offset Value: -1
default Offset Value: -28

Step 3: Save new calibration settings or recover default settings for module

New Calibration parameter:

Gain: 32993
Offset: -1

8017 Calibration Demo V1.0.0

Select I-8014W slot Index: Slot 1
Select Input Range: +/- 10.0 V

using Gain Value: 32917
default Gain Value: 32917
using Offset Value: -28
default Offset Value: -28

Step 3: Save new calibration settings or recover default settings for module

New Calibration parameter:

Gain: 32993
Offset: -1

# 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.

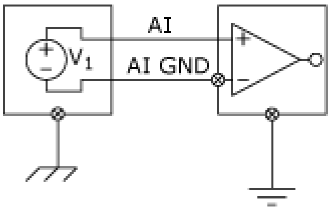
**Tips & Warnings**

⚠

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

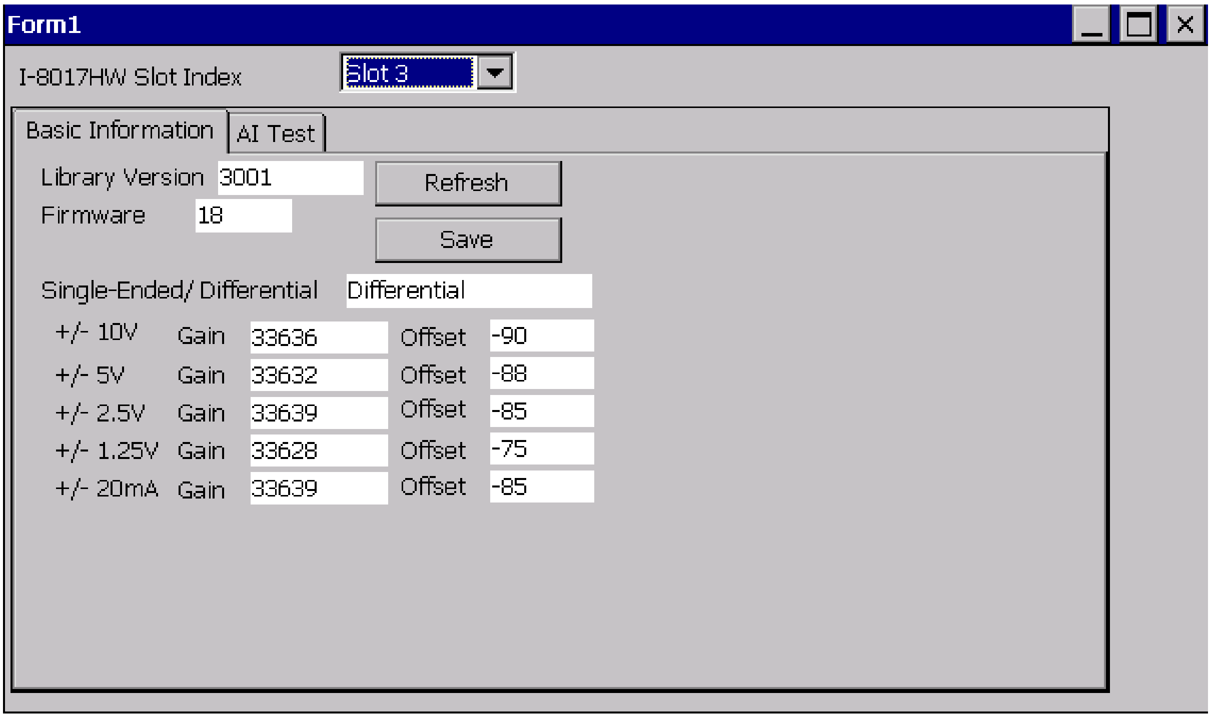
**Step 3.** Read the information from the module

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



I-8017HW Slot Index: Slot 3

Library Version: 3001

Firmware: 18

Single-Ended/ Differential: Differential

| | | | | |
|---|---|---|---|---|
| +/- 10V | Gain | 33636 | Offset | -90 |
| +/- 5V | Gain | 33632 | Offset | -88 |
| +/- 2.5V | Gain | 33639 | Offset | -85 |
| +/- 1.25V | Gain | 33628 | Offset | -75 |
| +/- 20mA | Gain | 33639 | Offset | -85 |

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.

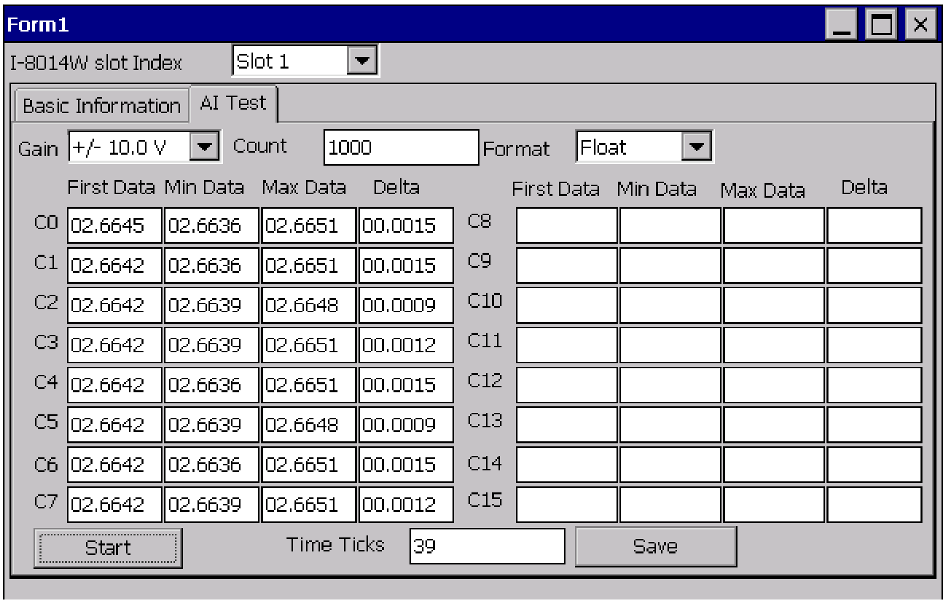
b. Relocate the module to a different slot, and then repeat Steps 2 and 3 to confirm whether or not the Gain values are correct.

### 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.



I-8014W slot Index: Slot 1

Gain: +/- 10.0 V   Count: 1000   Format: Float

| | First Data | Min Data | Max Data | Delta | | First Data | Min Data | Max Data | Delta |
|---|---|---|---|---|---|---|---|---|---|
| C0 | 02.6645 | 02.6636 | 02.6651 | 00.0015 | C8 | | | | |
| C1 | 02.6642 | 02.6636 | 02.6651 | 00.0015 | C9 | | | | |
| C2 | 02.6642 | 02.6639 | 02.6648 | 00.0009 | C10 | | | | |
| C3 | 02.6642 | 02.6639 | 02.6651 | 00.0012 | C11 | | | | |
| C4 | 02.6642 | 02.6636 | 02.6651 | 00.0015 | C12 | | | | |
| C5 | 02.6642 | 02.6639 | 02.6648 | 00.0009 | C13 | | | | |
| C6 | 02.6642 | 02.6636 | 02.6651 | 00.0015 | C14 | | | | |
| C7 | 02.6642 | 02.6639 | 02.6651 | 00.0012 | C15 | | | | |

Time Ticks: 39

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)

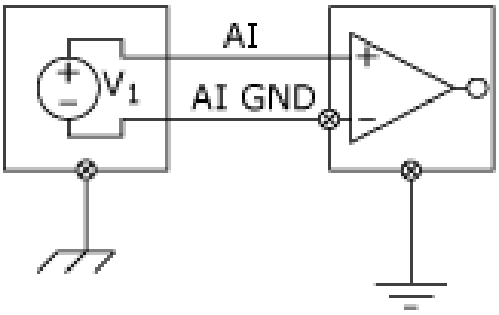
## 6.3. What to do when the data read from the module seems unstable

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 was a problem with the module ID |
| -2 | SLOT_ERROR | There was a Slot index error (0 - 7) |
| -3 | CHANNEL_ERROR | There was a Channel index error (0 - 15) |
| -4 | GAIN_ERROR | There was a Gain error (0 - 4) |
| -5 | INT_MODE_ERROR | |
| -6 | NOT_SUPPORT_ERROR | |
| -7 | INVALID_Calibration | |
| -8 | Bad_Calibration | |

# 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<br>• Added calibration instructions for modules based on the Windows platform<br>• Added performance information for all platforms |
| 3.0.0 | March 2018 | • Modify library , demo path<br>• Added WP-9000 , ippc-wes7 library , demo path<br>• Added 3.5 pac_i8017HW_ReadGainOffset_Info<br>• Added 3.6pac_i8017HW_Read_mA_GainOffset |