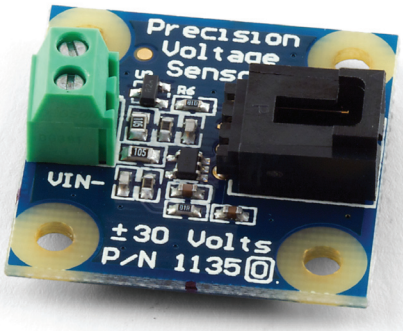
Product Manual

1135 - Precision Voltage Sensor



Phidgets 1135 - Product Manual

For Board Revision 0

Contents

Product Features

- Measures the differential voltage on the input terminals from a range of -30V to +30V
- Outputs an analog signal proportional to the voltage difference
- Interfaces easily with a Phidget InterfaceKit

Connections

Designed to connect to a:

- 1018 - PhidgetInterfaceKit 8/8/8
- 1019 - PhidgetInterfaceKit 8/8/8 w/6 Port Hub
- 1070 - PhidgetSBC
- 1203 - PhidgetTextLCD

Type of Measurement

The sensor uses non-ratiometric measurement.

Getting Started

Checking the Contents

You should have received:

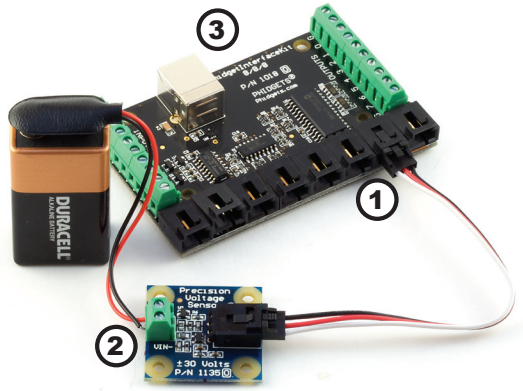
- A Precision Voltage Sensor
- A Sensor Cable

In order to test your new Phidget you will also need:

- A PhidgetInterfaceKit 8/8/8
- A USB Cable
- A 9 volt battery

Connecting all the pieces

1. Connect the Precision Voltage Sensor to an Analog Input on the PhidgetInterfaceKit 8/8/8 board using the sensor cable.
2. Connect the voltage source to the terminal block. It is possible to connect the leads of the voltage source to either terminal - the reported voltage may just be of the opposite polarity.
3. Connect the InterfaceKit board to your PC using the USB cable.



Testing Using Windows 2000/XP/Vista

Downloading the Phidgets drivers

Make sure that you have the current version of the Phidget library installed on your PC. If you don't, do the following:

Go to www.phidgets.com >> Drivers

Download and run Phidget21 Installer (32-bit, or 64-bit, depending on your PC)

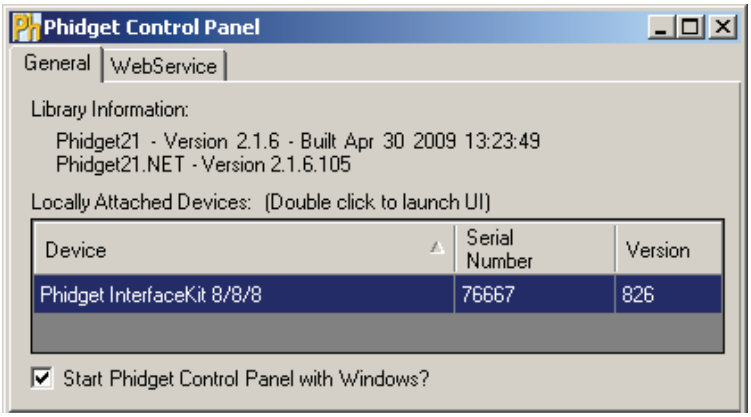
You should see the icon on the right hand corner of the Task Bar.

Running Phidgets Sample Program

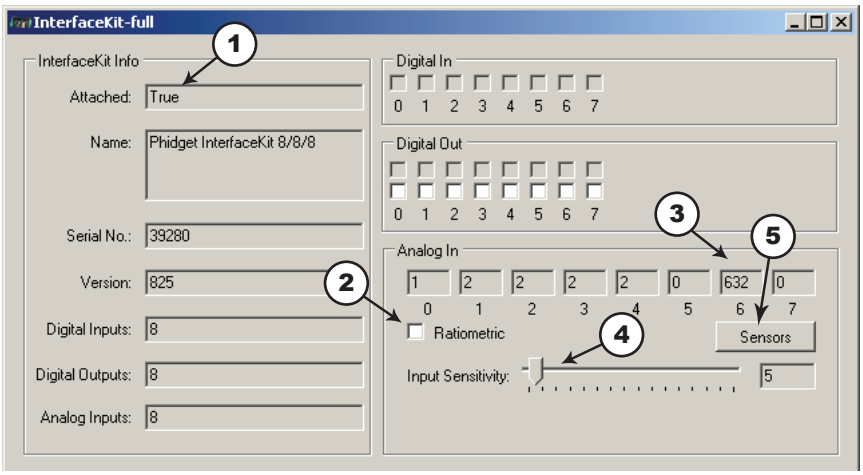
Double clicking on the icon loads the Phidget Control Panel; we will use this program to make sure that your new Phidget works properly. Since the sensor is connected to a 1018, the computer will see only the 1018. The sensor is providing data through the Analog input it is connected to.

The source code for the InterfaceKit-full sample program can be found under C# by clicking on www.phidgets.com >> Programming.

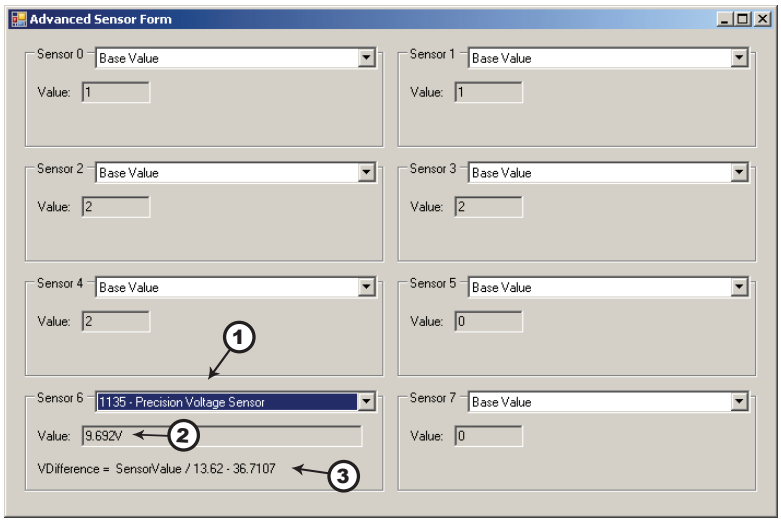
Double Click on the icon to activate the Phidget Control Panel and make sure that the **Phidget InterfaceKit 8/8/8** is properly attached to your PC.



1. Double Click on Phidget InterfaceKit 8/8/8 in the Phidget Control Panel to bring up InterfaceKit-full and check that the box labelled Attached contains the word True.
2. Make sure that the Ratiometric box is **not** Ticked.
3. With no connections on the input terminal blocks, the Analog In value should be approximately 500. The value displayed shows the 9V battery actual voltage when the battery's + wire is connected to the VIN+ terminal block.
4. You can adjust the input sensitivity by moving the slider pointer.
5. Click on the Sensors button to bring up the Advanced Sensor Form.



1. In the Advanced Sensor Form, select the 1135 - Voltage Sensor from the drop down menu.
2. The differential voltage measured by the sensor is shown here.
3. Formula used to convert the analog input SensorValue into voltage.



Testing Using Mac OS X

- Click on System Preferences >> Phidgets (under Other) to activate the Preference Pane
- Make sure that the Phidget InterfaceKit 8/8/8 is properly attached.
- Double Click on Phidget InterfaceKit 8/8/8 in the Phidget Preference Pane to bring up the InterfaceKit-Full example. This example will function in a similar way as the Windows version, but note that it does not include an Advanced Sensor Display.

Programming a Phidget

Phidgets' philosophy is that you do not have to be an electrical engineer in order to do projects that use devices like sensors, motors, motor controllers, and interface boards. All you need to know is how to program. We have developed a complete set of Application Programming Interfaces (API) that are supported for Windows, Mac OS X, and Linux. When it comes to languages, we support VB6, VB.NET, C#.NET, C, C++, Flash 9, Flex, Java, LabVIEW, Python, Max/MSP, and Cocoa.

Code Samples

We have written sample programs to illustrate how the APIs are used.

Due to the large number of languages and devices we support, we cannot provide examples in every language for every Phidget. Some of the examples are very minimal, and other examples will have a full-featured GUI allowing all the functionality of the device to be explored. Most developers start by modifying existing examples until they have an understanding of the architecture.

Go to www.phidgets.com >> Programming to see if there are code samples written for your device. Find the language you want to use and click on the magnifying glass besides "Code Sample". You will get a list of all the devices for which we wrote code samples in that language.

If this is your first time writing a program to control a Phidget, you should read the Getting Started Guide for the language you are planning to use.

Coding for your Sensor

Phidget analog sensors do not have their own API, but instead their output is a voltage that is converted to a digital value and accessed through the SensorValue properties and events on a PhidgetInterfaceKit. It is not possible to programmatically identify which sensor is attached to the Analog Input. Your application will need to apply any formulas from this manual to the SensorValue to translate it into usable data.

See the PhidgetInterfaceKit product manual for an overview of its API and a description of our architecture.

Technical Information

General

The Voltage Sensor measures the differential voltage between the input terminals and outputs the difference proportionally. The maximum differential voltage that can be measured accurately is +/-30V. When the positive and negative inputs are equal, the analog output value is 2.5V. When the positive input is 30V greater than the negative input, the analog output is 4.5V and when the positive input is 30V less than the negative input, the analog output is 0.5V.

Since the 1135 Voltage Sensor can measure a differential voltage, the common mode rejection (CMR) is an important specification. CMR refers to the amount of voltage that both input terminals of an differential amplifier can be offset without affecting the output gain. For example, if the positive terminal sees a voltage of 7V and the negative terminal sees a voltage of 5V, then the CMR would be 5V and would output a value of 2V at unity gain. For the 1135 Voltage Sensor, it is able to measure the differential voltage of +/-10V with a CMR of 40V while keeping the accuracy within 2%.

Please note that the error specifications do not include the error introduced by the Analog to Digital Conversion on the Analog Input. (if you are using the 1135 with a PhidgetInterfaceKit) The majority of error introduced by the Analog to Digital Conversion is from the error in the voltage reference (0.5% max), and the limitation of resolution in the SensorValue property. The best accuracy can be achieved by using a 2 or more point calibration of your system - effectively calibrating the 1135 and the PhidgetInterfaceKit in a single step. If you are calibrating, be sure to use a good quality multimeter to determine the voltage being applied.

Using RawSensorValue in the formula will increase the resolution, which is limited by SensorValue to about 67mV.

Formulas

The Formula to translate SensorValue into differential voltage is:

$$Vdifference = ((SensorValue / 200) - 2.5) / 0.0681$$

where Vdifference is defined as Vpositive - Vnegative.

Other Interfacing Alternatives

If you want maximum accuracy, you can use the RawSensorValue property from the PhidgetInterfaceKit. To adjust a formula, substitute (SensorValue) with (RawSensorValue / 4.095)

If the sensor is being interfaced to your own Analog to Digital Converter and not a Phidget device, our formulas can be modified by replacing (SensorValue) with (Vin * 200). It is important to consider the voltage reference and input voltage range of your ADC for full accuracy and range.

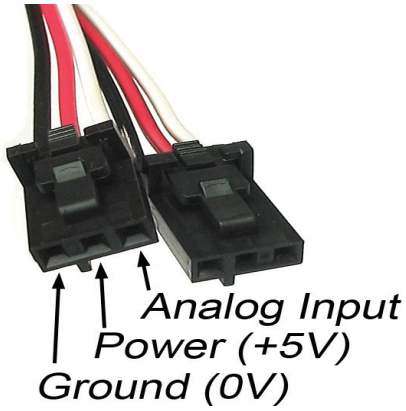
Coding for your Sensor

Phidget analog sensors do not have their own API, but instead their output is a voltage that is typically converted to a digital value and accessed through the SensorValue properties and events on a Phidget InterfaceKit. It is not possible to programmatically identify which sensor is attached to the Analog Input. Your application will need to apply the formula from this manual to the SensorValue to translate it into usable data.

Please see the Phidget InterfaceKit product manual for code samples, an overview of its API, and a description of our architecture.

Analog Input Cable Connectors

Each Analog Input uses a 3-pin, 0.100 inch pitch locking connector. Pictured here is a plug with the connections labeled. The connectors are commonly available - refer to the Table below for manufacturer part numbers.



Cable Connectors		
Manufacturer	**Part Number**	**Description**
Molex	50-57-9403	3 Position Cable Connector
Molex	16-02-0102	Wire Crimp Insert for Cable Connector
Molex	70543-0002	3 Position Vertical PCB Connector
Molex	70553-0002	3 Position Right-Angle PCB Connector (Gold)
Molex	70553-0037	3 Position Right-Angle PCB Connector (Tin)
Molex	15-91-2035	3 Position Right-Angle PCB Connector - Surface Mount

Note: Most of the above components can be bought at www.digikey.com

Device Specifications

Characteristic	Value
Current Consumption	3.6mA
Input Impedance	1MegaOhm
Recommended Max Difference between Vin+ and Vin-	30V
Absolute Maximum Difference between Vin+ and Vin-	35V
Input CMR	40V
Ouput Range	0 to 5V
Min/Max Error	±2.0%
Typical Error	±0.7%
Maximum Offset at 0V input	±100mV

Product History

Date	Board Revision	Comment
March 2010	0	Product Release

Support

Call the support desk at 1.403.282.7335 9:00 AM to 5:00 PM Mountain Time (US & Canada) - GMT-07:00

or

E-mail us at: support@phidgets.com