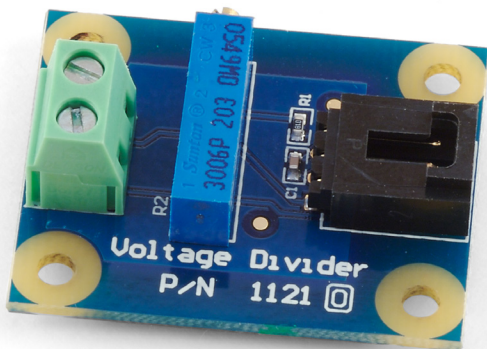


Product Manual

1121 - Voltage Divider



Phidgets 1121 - Product Manual

For Board Revision 0

© Phidgets Inc. 2009

Contents

4 Product Features

- 4 Connections
- 4 Type of Measurement

5 Getting Started

- 5 Checking the Contents
- 5 Connecting all the pieces
- 5 Testing Using Windows 2000/XP/Vista
- 7 Testing Using Mac OS X

7 Programming a Phidget

- 7 Code Samples
- 7 Coding for your Sensor

8 Technical Information

- 8 Configuring the Voltage Divider
- 8 Non-Phidgets Sensors
- 9 Other Interfacing Alternatives
- 9 Analog Input Cable Connectors
- 9 Mechanical Drawing
- 10 Device Specifications

10 Product History

10 Support

Product Features

- Provides an interface to non-Phidgets resistance sensors such as: Force-Sensing Resistors (FSR), light sensors, thermistors (heat sensors), and bend sensors.
- Eliminates the need to connect a resistor to an analog input by putting a high precision potentiometer on the board to allow the adjustment of the 'fixed' resistor.
- The interface to the variable resistance sensor is a convenient terminal block.

Connections

Designed to connect to a:

- 1018 - PhidgetInterfaceKit 8/8/8
- 1019 - PhidgetInterfaceKit 8/8/8 w/6 Port Hub
- 1070 - PhidgetSBC
- 1202 - PhidgetTextLCD

Type of Measurement

The sensor uses ratiometric measurement.

Getting Started

Checking the Contents

You should have received:

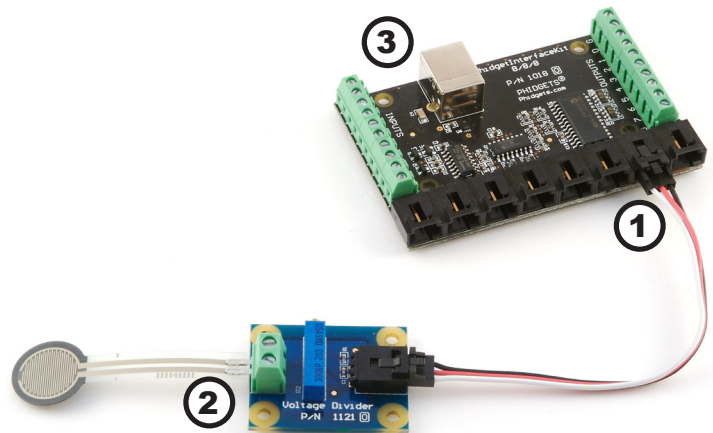
- A Voltage Divider
- A Sensor Cable

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

- A PhidgetInterfaceKit 8/8/8 or a PhidgetTextLCD
- A USB Cable
- A variable resistance sensor

Connecting all the pieces

1. Connect the Voltage Divider to the Analog Input 6 on the PhidgetInterfaceKit 8/8/8 using the sensor cable.
2. Connect the variable resistance sensor to the terminal block on the Voltage Divider.
3. Connect the PhidgetInterfaceKit 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)

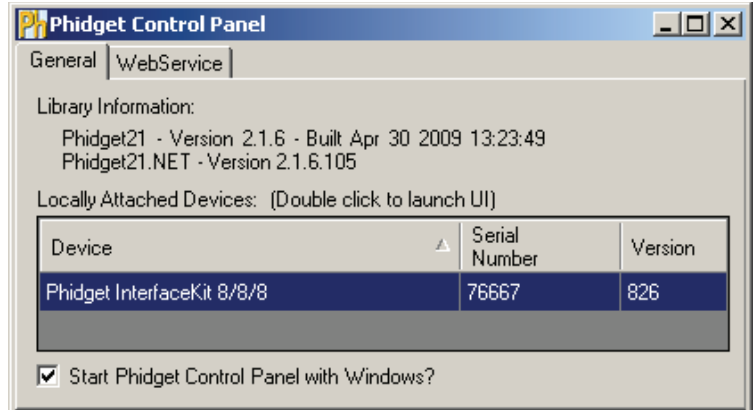
You should see the  icon on the right hand corner of the Task Bar.

Running Phidgets Sample Program

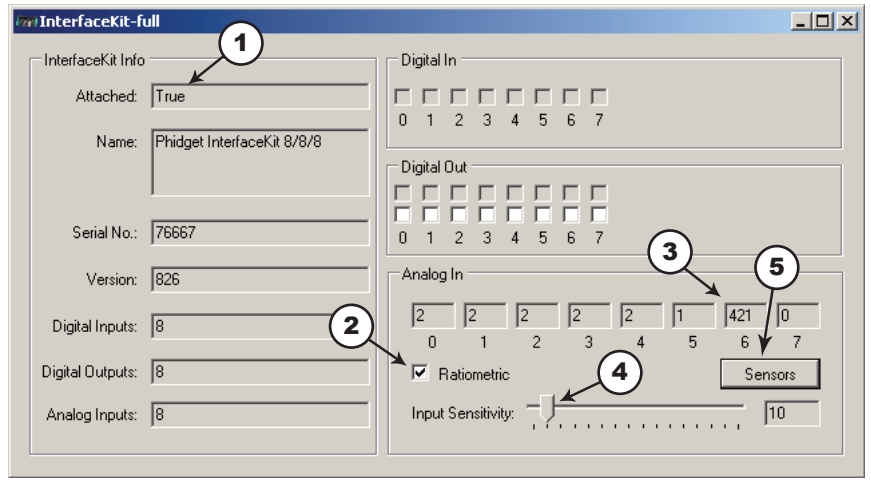
Double clicking on the  icon loads the Phidget Control Panel; we will use this program to make sure that your new Phidget works properly. Since the sensor is connected to a 1018, the computer will see only the 1018. The sensor is providing data through the Analog input it is connected to.

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

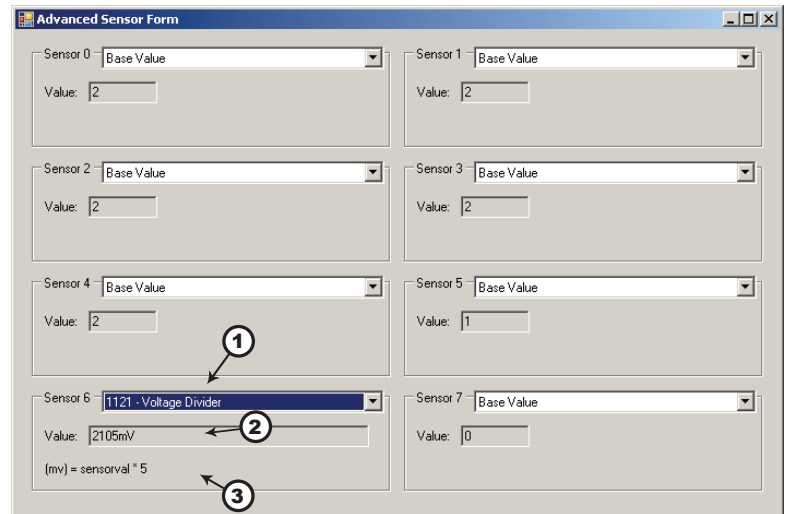
Double Click on the  icon to activate the Phidget Control Panel and make sure that the **Phidget InterfaceKit 8/8/8** is properly attached to your PC.



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 Ticked.
3. As you apply a stimuli (in our example, pushing on the variable force sensor) you should see the value change in the Analog In box.
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 Sensor 6 box, select the 1121 - Voltage Divider from the drop down menu.
2. The voltage measured across 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.

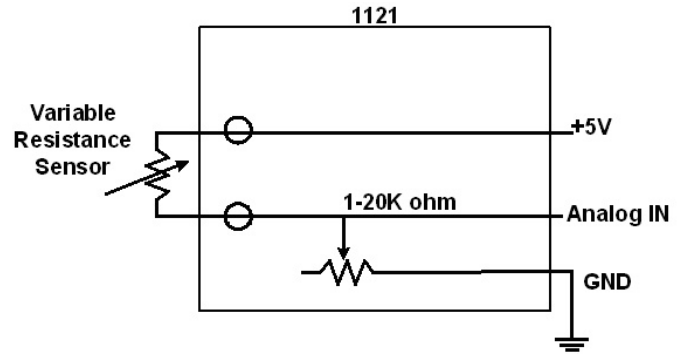
See the PhidgetInterfaceKit product manual for an overview of its API and a description of our architecture.

Technical Information

Configuring the Voltage Divider

Why is adjusting the resistance important?

The Analog Input can measure a voltage between 0V and 5V. The analog measurement is represented in the software as a value between 0 and 1000, so a sensor value of 1 unit represents a voltage of approximately 5 millivolts. When using variable resistance devices a voltage divider is commonly used to measure the changing resistance. The two resistors in the divider form a voltage, based on the ratio between them. This voltage is then measured by the Analog Input.



To optimize the voltage divider to meet your need

you have to adjust the fixed resistance to match the characteristics of your chosen sensor. To adjust the resistance on the Voltage Divider, simply turn the screw on the potentiometer clockwise to decrease the resistance and counterclockwise to increase it. You get between 900 ohm to 1000 ohm change in resistance per revolution.

First, adjust the resistance for no stimuli on your sensor and then apply maximum stimuli and make sure that Sensor Value stays within a range that gives you expectable resolution. At the extremes (<50, >950), a very large change in stimuli (and therefore change in the resistance) produces only a small change in Sensor Value. Keep adjusting until you get a range you are happy with.

Non-Phidgets Sensors

Here is a list of interesting variable resistance sensors that could be used with the Voltage Divider.

Light Sensors:

www.digikey.com

Light Sensors - CDS Photocells (Advanced Photonics)

Search for "photocell"

Examples - "PDV-P9003-1"

Force Sensors

www.digikey.com

Force Sensors (CUI Inc.)

Search for "force sensor"

Examples - "IESP-12"

www.trossenrobotics.com

Force Sensing FSRs

Thermistors

www.digikey.com

Search Digikey for 'Thermistor Radial'

Choose a thermistor with a resistance in the range 10K ohms.

Bend sensor

<http://www.imagesco.com>

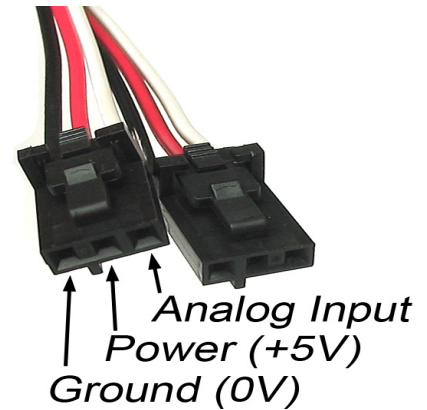
Other Interfacing Alternatives

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.

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.

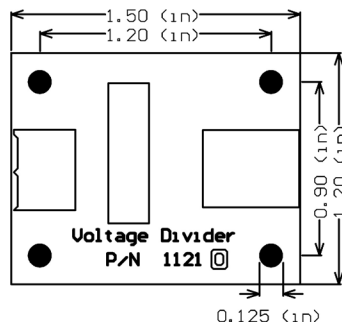


Cable Connectors		
Manufacturer	Part Number	Description
Molex	50-57-9403	3 Position Cable Connector
Molex	16-02-0102	Wire Crimp Insert for Cable Connector
Molex	70543-0002	3 Position Vertical PCB Connector
Molex	70553-0002	3 Position Right-Angle PCB Connector (Gold)
Molex	70553-0037	3 Position Right-Angle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

Mechanical Drawing

1:1 scale



Note: When printing the mechanical drawing, "Page Scaling" in the Print panel must be set to "None" to avoid re-sizing the image.

Device Specifications

Characteristic	Value
Max Resistance	21K ohms
Min Resistance	1K ohm
Optimal Resistance Measuring Range @ 21K	2K - 200K
Optimal Resistance Measuring Range @ 1K	100 - 10K
Current Consumption	5mA Max
Output Impedance	1K - 20K ohms
Supply Voltage	3.0VDC to 5.25VDC

Product History

Date	Board Revision	Comment
August 2007	n/a	Product Release

Support

Call the support desk at 1.403.282.7335 8:00 AM to 5:00 PM Mountain Time (US & Canada) - GMT-07:00

or

E-mail us at: support@phidgets.com