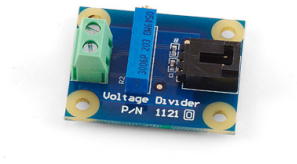


# 1121 User Guide

---



Go to this device's product page <sup>[1]</sup>

## Getting Started

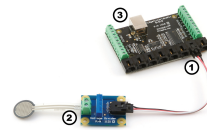
### Checking the Contents

**You should have received:**    **In order to test your new Phidget you will also need:**

- A voltage divider
- A sensor cable
- A PhidgetInterfaceKit 8/8/8 or PhidgetTextLCD
- A USB cable
- A variable resistance sensor


### Connecting 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 / 7

Make sure you have the current version of the Phidget library installed on your PC. If you don't, follow these steps:

1. Go to the Quick Downloads section on the Windows page
2. Download and run the Phidget21 Installer (32-bit, or 64-bit, depending on your system)
3. You should see the  icon on the right hand corner of the Task Bar.

### Running Phidgets Sample Program

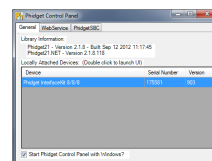
Double clicking on the  icon loads the Phidget Control Panel; we will use this program to ensure that your new Phidget works properly.

The source code for the **InterfaceKit-full** sample program can be found in the quick downloads section on the C# Language Page. If you'd like to see examples in other languages, you can visit our Languages page.

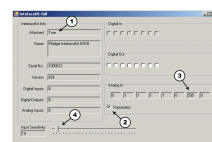
## Updating Device Firmware

If an entry in this list is red, it means the firmware for that device is out of date. Double click on the entry to be given the option of updating the firmware. If you choose not to update the firmware, you can still run the example for that device after refusing.

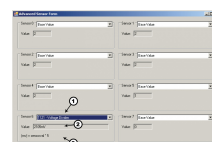
Double Click on the  icon to activate the Phidget Control Panel and make sure that the **Phidget InterfaceKit 8/8/8** is properly attached to your PC.



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

1. Go to the Quick Downloads section on the Mac OS X page
2. Download and run the Phidget OS X Installer
3. Click on System Preferences >> Phidgets (under Other) to activate the Preference Pane
4. Make sure that it is properly attached.
5. Double Click on in the Phidget Preference Pane to bring up the Sample program. This program will function in a similar way as the Windows version.

## Using Linux

For a step-by-step guide on getting Phidgets running on Linux, check the Linux page.

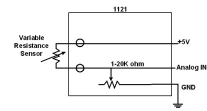
## Using Windows Mobile / CE 5.0 / CE 6.0

For a step-by-step guide on getting Phidgets running on Windows CE, check the Windows CE page.

## Technical Details

### 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:

<http://www.digikey.com>

Light Sensors - CDS Photocells (Advanced Photonics) Search for "photocell"

Examples - "PDV-P9003-1"

### Force Sensors

<http://www.digikey.com>

Force Sensors (CUI Inc.)

Search for "force sensor"

Examples - "IESP-12"

<http://www.trossenrobotics.com>

Force Sensing FSRs

### Thermistors

<http://www.digikey.com>

Search Digikey for 'Thermistor Radial'

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

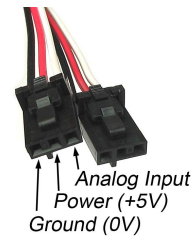
### Bend sensor

<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.

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



## API

Phidget analog sensors do not have their own API- they simply output a voltage that is converted to a digital value and accessed through the "Sensor" properties and events on the PhidgetInterfaceKit API. It is not possible to programmatically identify which sensor is attached to the Analog Input. To an InterfaceKit, every sensor looks the same. Your application will need to apply formulas from this manual to the **SensorValue** (an integer that ranges from 0 to 1000) to convert it into the units of the quantity being measured. For example, this is how you would use a temperature sensor in a C# program:

```
// set up the interfacekit object
InterfaceKit IFK = new InterfaceKit();

// link the new interfacekit object to the connected board
IFK.open("localhost", 5001);

// Get sensorvalue from analog input zero
int sensorvalue = IFK.sensors[0].Value;

// Convert sensorvalue into temperature in degrees Celsius
double roomtemp = Math.Round(((sensorvalue * 0.22222) - 61.11), 1);
```

See the PhidgetInterfaceKit User Guide for more information on the API and a description of our architecture.

For more code samples, find your preferred language on the Languages page.

## Product History

Date	Board Revision	Device Version	Comment
August 2007	0	N/A	Product Release

## References

[1] [http://www.phidgets.com/products.php?product\\_id=1121](http://www.phidgets.com/products.php?product_id=1121)

# Article Sources and Contributors

**1121 User Guide** *Source:* [http://www.phidgets.com/wiki/index.php?title=1121\\_User\\_Guide](http://www.phidgets.com/wiki/index.php?title=1121_User_Guide) *Contributors:* Burley, Mparadis

## Image Sources, Licenses and Contributors

**Image:1121.jpg** *Source:* <http://www.phidgets.com/wiki/index.php?title=File:1121.jpg> *License:* unknown *Contributors:* Mparadis

**File:1121\_0\_Connecting\_The\_Hardware.jpg** *Source:* [http://www.phidgets.com/wiki/index.php?title=File:1121\\_0\\_Connecting\\_The\\_Hardware.jpg](http://www.phidgets.com/wiki/index.php?title=File:1121_0_Connecting_The_Hardware.jpg) *License:* unknown *Contributors:* Mparadis

**File:Ph.jpg** *Source:* <http://www.phidgets.com/wiki/index.php?title=File:Ph.jpg> *License:* unknown *Contributors:* Mparadis

**File:1018\_2\_Control\_Panel\_Screen.jpg** *Source:* [http://www.phidgets.com/wiki/index.php?title=File:1018\\_2\\_Control\\_Panel\\_Screen.jpg](http://www.phidgets.com/wiki/index.php?title=File:1018_2_Control_Panel_Screen.jpg) *License:* unknown *Contributors:* Mparadis

**File:1121\_0\_InterfaceKit\_Screen.jpg** *Source:* [http://www.phidgets.com/wiki/index.php?title=File:1121\\_0\\_InterfaceKit\\_Screen.jpg](http://www.phidgets.com/wiki/index.php?title=File:1121_0_InterfaceKit_Screen.jpg) *License:* unknown *Contributors:* Mparadis

**File:1121\_0\_Advanced\_Sensor\_Form\_Screen.jpg** *Source:* [http://www.phidgets.com/wiki/index.php?title=File:1121\\_0\\_Advanced\\_Sensor\\_Form\\_Screen.jpg](http://www.phidgets.com/wiki/index.php?title=File:1121_0_Advanced_Sensor_Form_Screen.jpg) *License:* unknown *Contributors:* Mparadis

**File:1121\_0\_Schematic.jpg** *Source:* [http://www.phidgets.com/wiki/index.php?title=File:1121\\_0\\_Schematic.jpg](http://www.phidgets.com/wiki/index.php?title=File:1121_0_Schematic.jpg) *License:* unknown *Contributors:* Mparadis

**File:Analoginput.jpg** *Source:* <http://www.phidgets.com/wiki/index.php?title=File:Analoginput.jpg> *License:* unknown *Contributors:* Mparadis