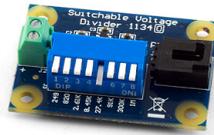


1134 User Guide



Go to this device's product page ^[1]

Getting Started

Checking the Contents

You should have received:

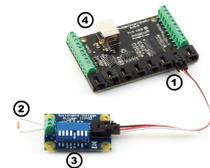
- A Switchable Voltage Divider board
- A Sensor cable

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

- A PhidgetInterfaceKit 8/8/8
- A USB cable
- A variable resistance sensor or any resistor (we are using a 27K resistor)

Connecting the Pieces

1. Connect the Switchable Voltage Divider to an Analog Input on the PhidgetInterfaceKit 8/8/8 board using the sensor cable.
2. Connect the variable resistance sensor to the terminal block on the Voltage Divider.
3. Select an appropriate measurement range for your application by toggling the desired DIP switch. We are toggling the 27.4K switch. When the switch is down, the associated measuring resistor is selected. It is possible to have more than one switch toggled and have the resistors in parallel. For our purpose only toggle one switch.
4. Connect the InterfaceKit board 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 **Ph** icon on the right hand corner of the Task Bar.

Running Phidgets Sample Program

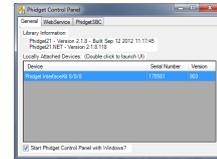
Double clicking on the **Ph** icon loads the Phidget Control Panel; we will use this program to ensure that your new Phidget works properly.

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.

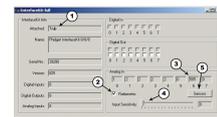
Updating Device Firmware

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.

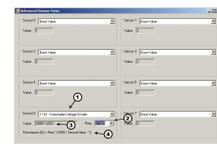
Double Click on the  icon to activate the Phidget Control Panel and make sure that the **Phidget InterfaceKit 8/8/8** is properly attached to your PC.



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. The Analog In box displays a number corresponding to the resistor value.
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 1134 - Switchable Voltage Divider from the drop down menu.
2. Select the appropriate resistor from the drop-down menu that has been selected with the DIP switch on the board.
3. The resistance measured by the sensor is shown here.
4. Formula used to convert the analog input SensorValue into resistance.



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 the 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

The 1134 Switchable Voltage Divider is a voltage divider with switches. The switches can be used to control which resistors are used in the voltage divider. From left to right, the resistance values (in Ohms) are 249, 820, 2.61K, 8.45K, 27.4K, 91K, 300K and 1M. It is possible to have multiple switches flipped, putting the corresponding resistors in parallel.

Formulas

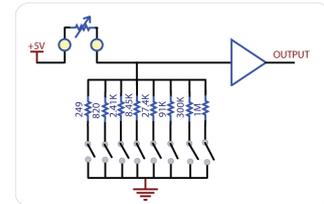
The Formula to translate SensorValue into resistance is:

$$\text{Resistance (Ohms)} = R_{eq} \times \frac{1000}{\text{SensorValue} - 1}$$

Where Req is the equivalent resistance of the switches that are flipped on in Ohms. The Switchable Voltage Divider has an internal buffer on the output. This allows the output to be read without distortion caused by the device reading the measurement, such as the PhidgetInterfaceKit.

Configuring the Voltage Divider

To optimize the voltage divider to meet your needs, you have to adjust the switchable resistance to match the characteristics of your chosen variable resistor. First, adjust the resistance for no stimuli on your sensor, then apply maximum stimuli and make sure that the Sensor Value stays within a range that gives you an acceptable resolution. The formula is most accurate at a SensorValue of 500, with an error of 0.4%. At 100 and 900, the error increases to approximately 1.1% and at 50 and 950, it increases to 2.1%. As a general rule, if the SensorValue is over 900, a lower resistance should be switched in. If the SensorValue is below 100, a higher switch should be toggled. Continue adjusting the switches until you find an acceptable range.



Warning: It is possible for the 1134 to apply up to 5V to the variable resistor. If the resistor cannot handle this voltage, it can be damaged. The 1134 is a very simple way of measure resistance. When interfacing to sensors with small resistances (like 100 ohm RTDs, for instance), large currents can flow - up to 30 mA. This level of current may cause substantial heating in the variable resistor, distorting the measurement or even possibly permanently damaging it.

Variable resistance sensors

Here are some interesting variable resistance sensors that could be used with the Voltage Divider.

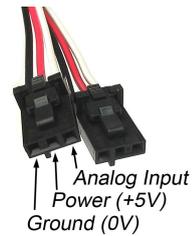
Sensor Type	Manufacturer	Example	Digikey Search
Light Sensors	Advanced Photonics	PDV-P9003-1	photcell
Force Sensors	CUI Inc.	IESP-12	force sensor
Thermistors	US Sensor	615-1037-ND	termistor radial
Bend Sensors	imagesco.com	FLX-01	n/a

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 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
March 2010	0	N/A	Product Release

References

[1] http://www.phidgets.com/products.php?product_id=1134

Article Sources and Contributors

1134 User Guide *Source:* http://www.phidgets.com/wiki/index.php?title=1134_User_Guide *Contributors:* Burley, Mparadis

Image Sources, Licenses and Contributors

Image:1134.jpg *Source:* <http://www.phidgets.com/wiki/index.php?title=File:1134.jpg> *License:* unknown *Contributors:* Mparadis

File:1134_0_Connecting_The_Hardware.jpg *Source:* http://www.phidgets.com/wiki/index.php?title=File:1134_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 *License:* unknown *Contributors:* Mparadis

File:1134_0_InterfaceKit_Screen.jpg *Source:* http://www.phidgets.com/wiki/index.php?title=File:1134_0_InterfaceKit_Screen.jpg *License:* unknown *Contributors:* Mparadis

File:1134_0_Advanced_Sensor_Form_Screen.jpg *Source:* http://www.phidgets.com/wiki/index.php?title=File:1134_0_Advanced_Sensor_Form_Screen.jpg *License:* unknown *Contributors:* Mparadis

File:1134_0_InternalDiagram.jpg *Source:* http://www.phidgets.com/wiki/index.php?title=File:1134_0_InternalDiagram.jpg *License:* unknown *Contributors:* Mparadis

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