

Analog Input Primer

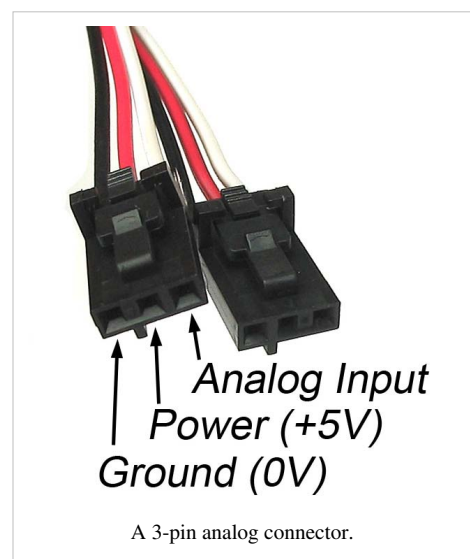
Analog Inputs are used to interface many different types of sensors, such as temperature, humidity, position, or pressure sensors. Phidget InterfaceKits such as the 1018 - PhidgetInterfaceKit 8/8/8 ^[1] have multiple analog inputs. Each Analog Input provides power (Nominal +5VDC), ground, and an analog voltage return wire driven by the sensor to some voltage. The Interface Kit continuously measures this return voltage and reports it to the application.

Phidgets offers a wide variety of sensors that can be plugged directly into the board using the cable included with the sensor.

Mechanical Specifications

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 |

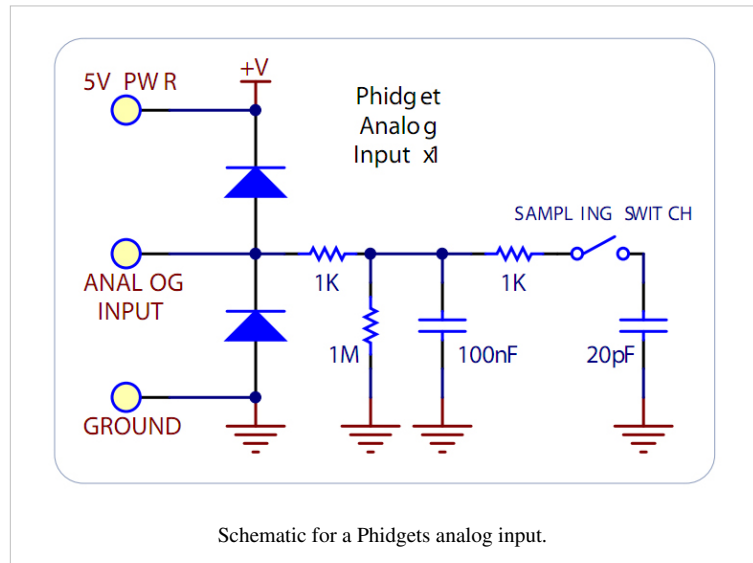
Note: Most of the above components can be bought at an electronic parts supplier such as www.digikey.com ^[2].

Electrical Specifications

The maximum total current consumed by all Analog Inputs should be limited to 400mA. The analog measurement is represented in the software through the SensorValue as a value between 0 and 1000. 5V corresponds to a high sensor value, and 0V corresponds to zero sensor activity. A sensor value of 1 unit represents a voltage of approximately 5 millivolts. The RawSensorValue property brings out a 12-bit value (0-4095) for users who require maximum accuracy. Please note that the sampling is actually done with an oversampled 10-bit ADC, but reported as a 12-bit value to allow future expansion.

One side effect of this oversampling is that very low noise signals will not get full resolution from the ADC without dithering noise injected into the signal.

The analog inputs on Phidgets interface kits are designed for a maximum of 5V. More than this will cause unpredictable behaviour and could damage the board.



Ratiometric Configuration

The group of Analog Inputs can be collectively set to Ratiometric mode from software using the Ratiometric property. If you are using a sensor whose output changes linearly with variations in the sensor's supply voltage level, it is said to be ratiometric. Most of the sensors sold by Phidgets are ratiometric (this is specified on the web product page and in the sensor's product manual).

Setting Ratiometric causes the reference to the internal Analog to Digital Converter to be set to the power supply voltage level. When Ratiometric is enabled, the maximum voltage returned on the Analog Input should be the +5V nominal power provided by the PhidgetInterfaceKit.

If Ratiometric is false, the ADC reference is set to a 5.0V 0.5% stable voltage reference. The maximum voltage returned on the Analog Input should be maximum 5.0V. Note that the Analog Input power supply voltage is not affected by the setting of the Ratiometric property.

The ratiometric setting of the interface kit applies to all inputs. Trying to read a sensor with the wrong ratiometric setting will result in incorrect readings. However, it is technically possible to use both ratiometric and non-ratiometric sensors at the same time if you poll for one sensor and then switch the kit's ratiometric setting before polling the other sensor. When switching ratiometric mode on and off, you need to include a short delay (50ms should be more than enough) before you begin reading values in. This gives the device time to react.

Factors that can affect Accuracy

- **High Output Impedance** - Sensors that have a high output impedance will be distorted by the 900K input impedance of the Analog Input.

If your output impedance is high, it is possible to correct for this distortion to some extent in your software application.

- **Power Consumption** - Sensor cables have some resistance, and the power consumption of the sensor will cause the sensor to have a slightly different ground from the Analog Input on the PhidgetInterfaceKit. The more power consumed by the sensor, and the longer the sensor cable, the more pronounced this effect will be.
- **Intrinsic Error In Sensors** - For many sensors, the error is quite predictable over the life of the sensor, and it can be measured and calibrated out in software.
- **Non-Ratiometric Configuration** - Voltage Reference error. The 5.0VDC voltage reference is accurate to 0.5%.

This can be a significant source of error in some applications, but can be easily measured and compensated for.

Some sensors are accurate enough to take advantage of using the raw sensor value. If you want maximum accuracy, you can use the RawSensorValue property. To modify the formula, substitute (SensorValue) with (RawSensorValue / 4.095)

Changing the Data Rate

You can change the data rate for each Analog Input from 1 millisecond to 1 second. By default, the analog input data set is sent to the PC every 8ms. For analog inputs, 8 ms is the maximum transmission rate. This limit is set by the USB processor we use, so there isn't much you can do to get around it.

For values less than 8 ms, the data rate sets the sampling rate, not the transmission rate. If, for example, you set the data rate to 1ms, you will receive a packet containing 8 milliseconds worth of 1 ms samples every 8ms. Setting the data rate at 1, 2, or 4ms will not allow you to react to received sensor data any faster than every 8ms; You will simply get more data samples within the 8ms. This feature is useful if you need to log sensor data at less than 8 ms resolution but don't need to react to it in real-time.

Setting the data rate of the analog inputs is an alternative to setting the sensitivity. Setting one of these values will disable the other, since you can either have events trigger on a regular interval, or on a minimum change.

When the Data rate is set at a multiple of 8 ms, the data rate sets both the sampling rate and the transmission rate. There is also a limit as to how many channels can be set at a high sampling rate, since you will, at one point run out of bandwidth. We estimate that you can set up to 4 channels to 1ms or you could set all channels to 2ms. You can't turn channels off entirely to save bandwidth, you can only set them to a slower data rate. You will get an error when you exceed the available bandwidth, warning you of lost data samples.

Note that data rate is limited to at most 16ms when opening over the Phidget Webservice. Actual data rate will depend on network latency.

The method to change data rate is different in each API (See your API for more information). For example, here's how you would change data rate in C#:

```
// Create a new InterfaceKit object
InterfaceKit IFK = new InterfaceKit();

// Open a local InterfaceKit over the webservice
IFK.open("localhost", 5001);

// Change the data rate of analog input zero to 8ms (125 samples/sec)
IFK.sensors[0].DataRate = 8;
```

Changing the Sensitivity/Change Trigger

You can change the sensitivity of an analog sensor using our API. You can think of sensitivity as a minimum amount needed to register a change (thus triggering a `SensorChangeEvent`). Whenever your sensor generates a new value, it will be compared to the sensitivity. If the difference between the previous data point and the new data point is less than the sensitivity value, no event will be generated. Sensitivity is measured in sensorvalue, which is a value between 1-1000 that represents the full range of a sensor. Setting the sensitivity of an analog input is an alternative to setting the data rate. Setting one of these values will disable the other, since you can either have events trigger on a regular interval, or on a minimum change.

The method to change sensitivity/change trigger is different in each API (See your API for more information). For example, here's how you would change sensitivity in C#:

```
// Create a new InterfaceKit object
InterfaceKit IFK = new InterfaceKit();

// Open a local InterfaceKit over the webservice
IFK.open("localhost", 5001);

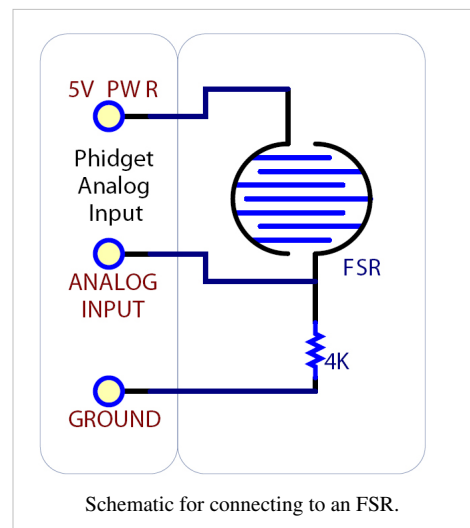
// Change the sensitivity of analog input zero to "10".
IFK.sensors[0].Sensitivity = 10;
```

Connecting non-Phidget devices to the Analog Inputs

Here are some circuit diagrams that illustrate how to connect various non Phidgets devices to the analog inputs on your Phidget.

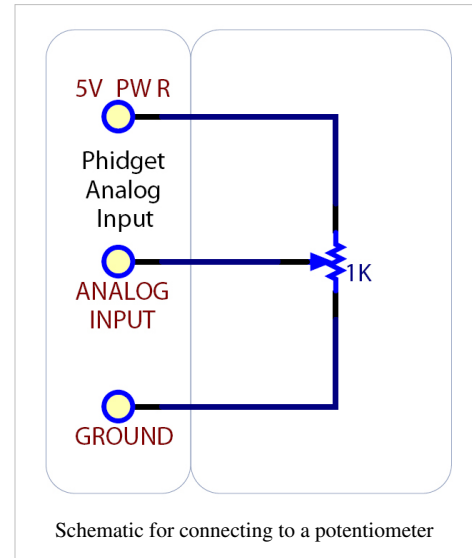
Sensing the Value of a Variable Resistance Sensor

In this diagram, an FSR (Force Sensitive Resistor) is shown.



Sensing the Position of a Potentiometer

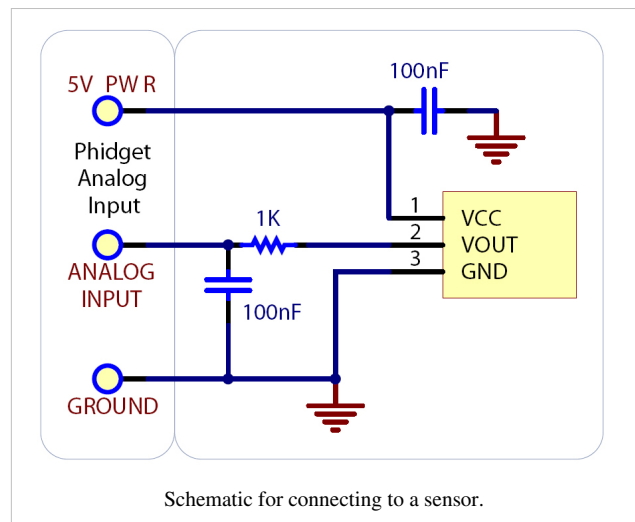
This diagram shows how to monitor the position of a potentiometer.



Interfacing to an Arbitrary Sensor

Normally, you can connect a sensor directly to the analog input as long as it has a 0-5V range (or smaller). However, if the sensor is not designed to send its signal across a long cable, you may need to add components as shown in the image.

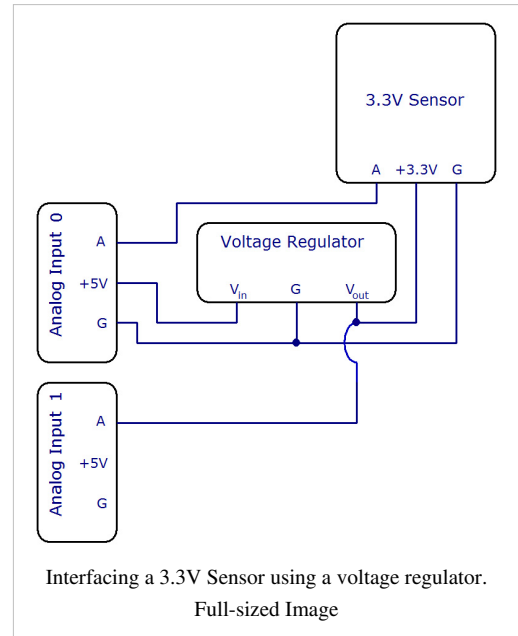
Note the use of power supply decoupling and the RC Filter on the output. The RC filter also prevents VOUT from oscillating on many sensors.



Interfacing 3.3V Sensors

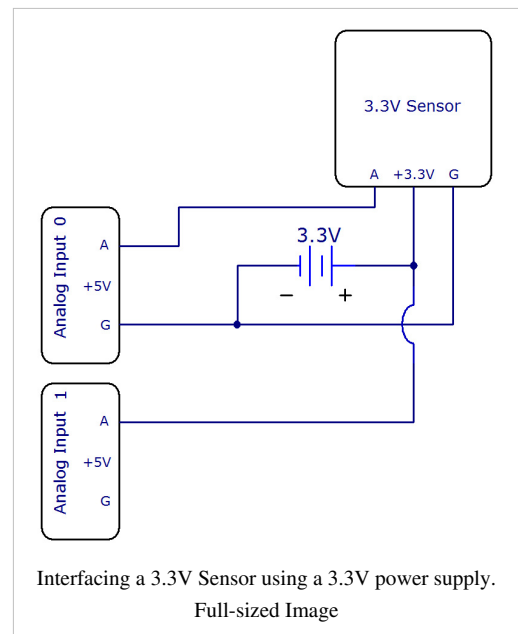
When using a 3.3V sensor with the analog input of a Phidget InterfaceKit, the main challenge is generating the 3.3V supply. You can either buy a 3.3V power supply, or you can buy a voltage regulator to convert the 5V line on the analog input to a 3.3V line, as illustrated in the diagrams. You can also use a second analog input to monitor the output of the 3.3V supply on the regulator. Note that SensorValue will still be equal to the input voltage multiplied by 200, so a 3.3V sensor will saturate at a SensorValue of 667 instead of the usual 1000.

If the sensor is ratiometric, you can determine the measured value of the sensor expressed as a percentage of the maximum value by dividing the sensor value of analog input 0 by analog input 1. For example, if analog input 1 measures a SensorValue of 667, and analog input 0 measures a SensorValue of 133, then you know that the sensor's measured value is twenty percent of the sensor's maximum value.



Interfacing 4-20mA Sensors

You can use the 1132 - 4-20mA Sensor Interface ^[3] to read a 4-20mA sensor with an analog input. For more information on 4-20mA sensors, see the 4-20mA Sensor Interface Primer.



List of Devices with Analog Input

| Product # | Name | # of Analog Inputs | Additional Features |
|-----------|---|--------------------|--|
| 1010 | PhidgetInterfaceKit 8/8/8 Mini-Format ^[4] | 8 | DIP-36 Package for compact and OEM applications, 8 Digital Inputs, 8 Digital Outputs |
| 1011 | PhidgetInterfaceKit 2/2/2 ^[5] | 2 | Compact USB Dongle Size, 2 Digital Inputs, 2 Digital Outputs |
| 1018 | PhidgetInterfaceKit 8/8/8 ^[1] | 8 | 8 Digital Inputs, 8 Digital Outputs |
| 1019 | PhidgetInterfaceKit 8/8/8 w/6 Port Hub ^[6] | 8 | 8 Digital Inputs, 8 Digital Outputs, 6-port USB Hub |
| 1065 | PhidgetMotorControl 1-Motor ^[7] | 2 | 1 DC Motor control, 2 Digital Inputs, 1 Encoder Input |
| 1073 | PhidgetSBC3 ^[8] | 8 | Single Board Computer, 8 Digital Inputs, 8 Digital Output, 6-port USB Hub |
| 1203 | PhidgetTextLCD ^[9] | 8 | LCD Character Screen, 8 Digital Inputs, 8 Digital Outputs |

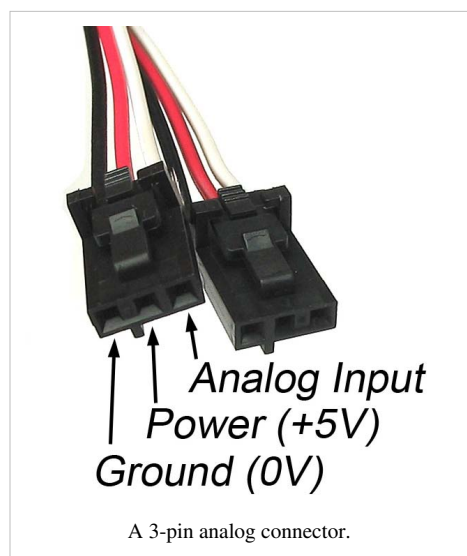
Analog Sensor Cables

| Product # | Name |
|-----------|------------------------------------|
| 3002 | Sensor Cable 60cm ^[10] |
| 3003 | Sensor Cable 10cm ^[10] |
| 3004 | Sensor Cable 350cm ^[10] |
| 3034 | Sensor Cable 15cm ^[10] |

Connector Details

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 an electronic parts supplier such as www.digikey.com ^[2].

Non Phidgets Analog Sensors

In addition to Phidgets sensors, any sensor that returns a signal between 0 and 5 volts can be easily interfaced. Here is a list of interesting sensors that can be used with the PhidgetInterfaceKit 8/8/8. Note: these sensors are not “plug & play” like the sensors manufactured by Phidgets.

| Manufacturer | Part Number | Description |
|-------------------------|-------------------|--|
| MSI Sensors | FC21/FC22 | Load cells - measure up to 100lbs of force |
| Humirel | HTM2500VB | Humidity sensors |
| Measurement Specialties | MSP-300 | Pressure sensors - ranges up to 10,000 PSI |
| Freescale Semiconductor | MPXA/MPXH | Gas Pressure Sensors |
| Allegro | ACS7 series | Current Sensors - ranges up to 200 Amps |
| Allegro | A1300 series | Linear Hall Effect Sensors - to detect magnetic fields |
| Analog | TMP35 TMP36 TMP37 | Temperature Sensor |
| Panasonic | AMN series | Motion Sensors |
| Honeywell | FS01, FS03 | Small, accurate Piezo-resistive load cells |
| AllSensors-Europe | BARO-A-4V | Barometric Pressure Sensor - 600 to 1,100 mbar |

!+Analog Sensors

Note: Most of the above components can be bought at www.digikey.com ^[2]

Analog sensors often have their precision measured in mV/Unit. This value represents how many millivolts the sensor will output given a certain measured value. For example, a temperature sensor might output 1 mV per degree Celsius. You can use this value to build a formula so your program can convert it to the measured quantity. Some sensors can have their mV/Unit output changed, which allows you to tweak the sensor's full scale of measurement. Read your sensor's data sheet for conversion formulae and calibration information.

References

- [1] http://www.phidgets.com/products.php?product_id=1018
 - [2] <http://www.digikey.com>
 - [3] http://www.phidgets.com/products.php?product_id=1132
 - [4] http://www.phidgets.com/products.php?product_id=1010
 - [5] http://www.phidgets.com/products.php?product_id=1011
 - [6] http://www.phidgets.com/products.php?product_id=1019
 - [7] http://www.phidgets.com/products.php?product_id=1065
 - [8] http://www.phidgets.com/products.php?product_id=1073
 - [9] http://www.phidgets.com/products.php?product_id=1203
 - [10] http://www.phidgets.com/products.php?product_id=3002
-

Article Sources and Contributors

Analog Input Primer *Source:* http://www.phidgets.com/wiki/index.php?title=Analog_Input_Primer *Contributors:* Burley, Mparadis

Image Sources, Licenses and Contributors

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

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

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

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

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

File:33vregulator.jpg *Source:* <http://www.phidgets.com/wiki/index.php?title=File:33vregulator.jpg> *License:* unknown *Contributors:* Mparadis

File:33vsensor.jpg *Source:* <http://www.phidgets.com/wiki/index.php?title=File:33vsensor.jpg> *License:* unknown *Contributors:* Mparadis