

Getting Started

You should have received:

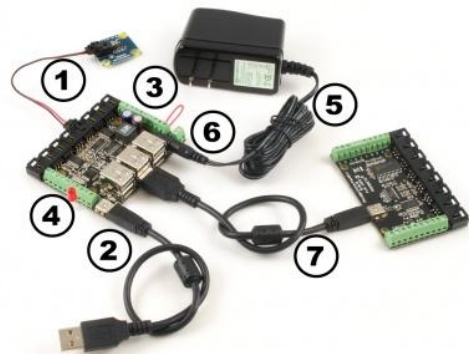
- The PhidgetInterfaceKit 8/8/8 w/6 Port Hub
- A Mini-USB cable
- A Power Supply

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

- A piece of wire to test the digital inputs
- An LED to test the digital outputs
- An analog Sensor to test the analog inputs
- A Phidget to test the USB Port Hub
- A USB Cable


Connecting the Pieces

1. Connect the Analog Sensor to the analog input port 3 using a Phidgets sensor cable.
2. Connect the 1019 - PhidgetInterfaceKit to the PC using the USB cable.
3. Connect one end of the wire to digital input port 0 and the other end to the ground connector.
4. Connect the LED by inserting the longer LED wire into the digital output port 7 and the shorter wire to Ground.
5. Connect the Power Supply to the InterfaceKit board
6. You can also connect a power supply to the Terminal Block. Be sure to observe correct polarity.
7. Connect the other Phidget to the InterfaceKit board using a 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  on the right hand corner of the Task Bar.


Running Phidgets Sample Program

Double clicking on the  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](#).

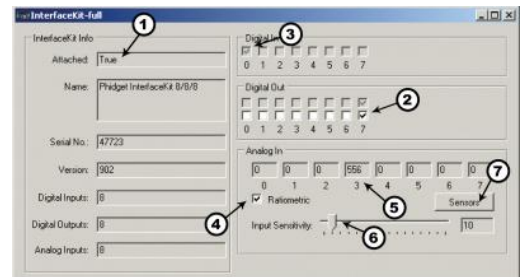
Updating Device Firmware

If an entry in this list is red, it means the firmware for that device is out of date.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.

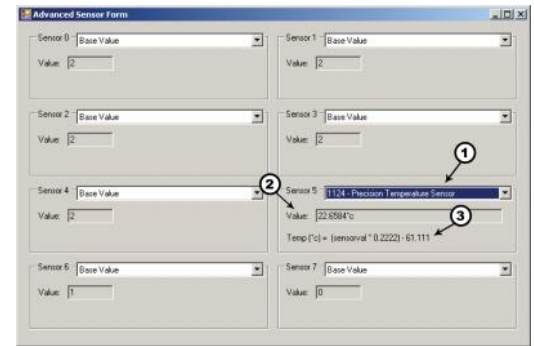
Double Click on the  located on the right hand corner of the Task Bar to activate the Phidget Control Panel. Make sure that both the **1019 - Phidget InterfaceKit 8/8/8** and the other Phidget plugged into it are properly connected to your PC.



1. Double Click on the first **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. Test the digital output by clicking on the box to turn on the LED. Clicking again will turn the LED off. The bottom row shows the status of the request, while the top row displays the status of the digital output as reported by the device.
3. Test the digital input by disconnecting the wire end connected to the digital input. connector. The tick mark in the box will go away.
4. Click on the Ratiometric Box if your sensor is ratiometric. Check the sensor product manual if you are not sure.
5. Test the analog input sensor by observing the sensor value as you activate the Phidget sensor.
6. You can adjust the input sensitivity by moving the slider pointer.



7. Click on Sensors to launch the Advanced Sensor Form.
1. In the drop down menu, select the Sensor you have attached to the analog input port 3 of the 1019. In our case we select the 1124 - Precision Temperature Sensor.
2. The ambient temperature sensed by the 1124.
3. Formula used to convert the analog input sensorval into temperature



Note: If you have connected a sensor that uses a formula; the calculated value and the formula are displayed under the drop down menu.

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 Phidget InterfaceKit 8/8/8 is properly attached.
5. Double Click on Phidget InterfaceKit 8/8/8 in the Phidget Preference Pane to bring up the InterfaceKit-full 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

Powering the PhidgetInterfaceKit

The PhidgetInterfaceKit with 6-port Hub is not powered from the PC-USB bus. An external 6 - 15V supply must be used to power the PhidgetInterfaceKit and any attached USB devices. However, the USB hub will not actually be powered until a USB connection is made. The reason for this is that the voltage regulator chip on the 1019 is what routes power to the hub, and it gets its power from the USB line. The 1019 will consume a maximum of 10mA from the USB host cable - allowing it to be directly connected to small hosts that do not provide full USB power.

Connecting additional USB devices to the PhidgetInterfaceKit is as easy as plugging them into the on-board 6-port hub. Each USB port on the hub has a maximum current supply of 500mA. Ensure the power supply selected has a high enough current output to supply the required current to all external USB

devices as well as the PhidgetInterfaceKit and any sensors or devices connected to it. The worst case requirement is 3 Watts input power per USB device. A 24 Watt 12VDC / 2 Amp power supply is provided with the 1019 - more than sufficient.

The USB Hub actually has 7 ports, but only 6 of them are used for connecting additional devices since one port is dedicated to the internal 8/8/8 InterfaceKit.

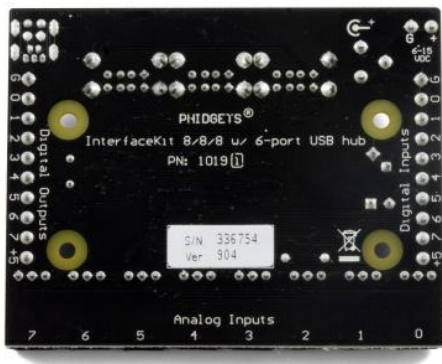
The USB Hub is a full-speed hub with a transfer rate of 12Mbits/second. We chose to go with a full speed implementation since it is fast enough to handle traffic from Phidgets; an added benefit is lower power consumption.

To summarize the power distribution of this board:

- Each USB port has 500 mA available
- All analog inputs share a total of 500 mA
- The digital outputs, +5V terminals, USB controller, and pull-ups all share a total of 500 mA
- The power inbound from the USB line does not contribute to any of the above

Connections

The ports and terminal blocks on this board are labelled on the underside to save space:



Chaining the USB Hubs

The 1019 follows USB specifications and can be daisy chained to the maximum hub depth of 5. A sixth PhidgetInterfaceKit with a hub plugged into the fifth hub will not be usable at all because the InterfaceKit portion is connected after the hub. However, other Phidgets plugged into the fifth hub will operate normally.

Further Reading

If you want to know more about the input/output capabilities of the 1019 InterfaceKit, check the [Digital Input Primer](#), [Digital Output Primer](#), and the [Analog Input Primer](#).

API

We document API Calls specific to this product in this section. Functions common to all Phidgets and functions not applicable to this device are not covered here. This section is deliberately generic. For

calling conventions under a specific language, refer to the associated API manual in the [Quick Downloads](#) section for that language. For exact values, refer to the device specifications.

Functions

int InputCount() [get] : Constant = 8

Returns the number of digital inputs supported by this PhidgetInterfaceKit.

bool InputState(int InputIndex) [get]

Returns the state of a particular digital input. Digital inputs read True where they are activated and false when they are in their default state.

int OutputCount() [get] : Constant = 8

Returns the number of digital outputs supported by this PhidgetInterfaceKit.

bool OutputState (int OutputIndex) [get,set]

Sets/returns the state of a digital output. Setting this to true will activate the output, False is the default state. Reading the OutputState immediately after setting it will not return the value set - it will return the last state reported by the Phidget.

int SensorCount() [get] : Constant = 8

Returns the number of sensors (Analog Inputs) supported by this PhidgetInterfaceKit. Note that there is no way of determining is a sensor is attached, and what sensor is attached.

int SensorValue(int SensorIndex) [get]

Returns the sensed value of a particular Analog Input. SensorValue varies between 0-1000, corresponding to the 0-5V input range of the Analog Input. If you are using an Analog Sensor from Phidgets Inc., it's manual will specify the formula used to convert SensorValue into the measured property.

int SensorRawValue (int SensorIndex) [get]

Returns the full resolution of the Analog Input. This is a more accurate version of SensorValue. The valid range is 0-4095. Note however that the analog outputs on the Interface Kit 8/8/8 are only 10-bit values and this value represents an oversampling to 12-bit.

double SensorChangeTrigger (int SensorIndex) [get,set]

Returns the change trigger for an analog input. This is the amount that an inputs must change between successive SensorChangeEvents. This is based on the 0-1000 range provided by getSensorValue. This value is by default set to 10 for most Interface Kits with analog inputs. SensorChangeTrigger is sometimes referred to as sensitivity.

int DataRate (int SensorIndex) [get,set]

Gets/sets the data rate for an analog input. This corresponds to the fastest rate at which SensorChange events will be fired. The data rate is superseded by SensorChangeTrigger, which can be set to 0 if a constant data rate is required. Data Rate is in milliseconds and corresponds to the amount of time between events. Data Rate is bounded by DataRateMax and DataRateMin. The analog inputs cannot all be set to the fastest data rate at the same time - if this is attempted, an exception will be thrown when the data bandwidth has been exceeded. For data rates less than the maximum, data is still sampled at the maximum speed, and averaged between events for the user. Supported data rates are: 1, 2, 4, 8, and every multiple of 8 until DataRateMin. Setting an unsupported data rate (ie. 3, 9, 17) will result in a thrown exception. Note that data rate is limited to 16ms when opening over the Phidget Webservice.

int DataRateMax (int SensorIndex) [get]

The maximum data rate that can be set for an analog input, in milliseconds.

int DataRateMin (int SensorIndex) [get]

The minimum data rate that can be set for an analog input, in milliseconds. This is usually 1000.

bool Ratiometric() [get,set]

Sets/returns the state of Ratiometric. Ratiometric = true configures the Analog Inputs to measure relative to VCC (nominal 5V). Ratiometric = false configures the Analog Inputs to measure relative to an internal precision 5V reference. Ratiometric is not updated from the Phidget. It is recommended to explicitly set Ratiometric when the Interfacekit is opened. After changing the ratiometric state, wait until the ratiometric property matches what was set before reading analog data.

Events

OnInputChange(int InputIndex, bool State) [event]

An event that is issued when the state of a digital input changes.

OnOutputChange(int OutputIndex, bool State), [event]

An event that is issued when the state of a digital output changes.

OnSensorChange(int SensorIndex, int SensorValue), [event]

An event that is issued when the returned value from a sensor (Analog Input) varies by more than the SensorChangeTrigger property.