

1011 User Guide



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

Getting Started

Checking the Contents

You should have received:

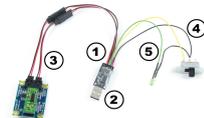
- A PhidgetInterfaceKit 2/2/2 board
- A 12-Wire Cable

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

- An LED to test the digital outputs.
- An Analog Sensor to test the analog inputs.
- A sensor cable.
- A piece of wire or a switch to test the digital inputs.

Connecting the Pieces

1. Connect the 12-Wire cable to the PhidgetInterfaceKit 2/2/2.
2. Plug the PhidgetInterfaceKit 2/2/2 into a USB port on your computer.
3. Connect the Analog Sensor to an analog input female connector using a Phidgets sensor cable. We are using both connectors since we are plugging in a Temperature/Humidity Sensor.
4. If you are using a wire, connect one end of your wire to the yellow wire (Digital Input 0) and the other end to the black wire (Ground). If you are using a switch, connect the black wire to the switch ground terminal, and connect the yellow wire to one of the switched terminals.
5. Connect the LED by soldering the long LED leg (Anode) to the Green Wire (Digital Output 0) and the shorter leg (cathode) to the black wire (Ground).



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

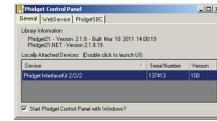
Double clicking on the **Ph** icon loads the Phidget Control Panel; we will use this program to ensure that your new Phidget works properly.

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.

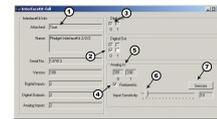
Updating Device Firmware

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 2/2/2 is properly attached to your PC.



1. Double Click on PhidgetInterfaceKit 2/2/2 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 white 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 wire. The tick mark in the box will go away. If you are using a switch, turn the switch on and off, and the tick mark in the box will appear and go away.
4. Click on the Ratiometric Box if your sensor is ratiometric. Check the sensor product manual if you are not sure.
5. The data from your sensor is displayed in the Analog In boxes.
6. You can adjust the input sensitivity by moving the slider pointer. This is the amount that an input must change to trigger an event that will modify the value in the Analog In box.
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 0 of the 1018. In our case we select the "1125 - Temperature/Humidity Sensor".
2. The ambient temperature and humidity is shown here.
3. Formula used to convert SensorValue into Temperature or Humidity



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 2/2/2 is properly attached.
5. Double Click on Phidget InterfaceKit 2/2/2 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

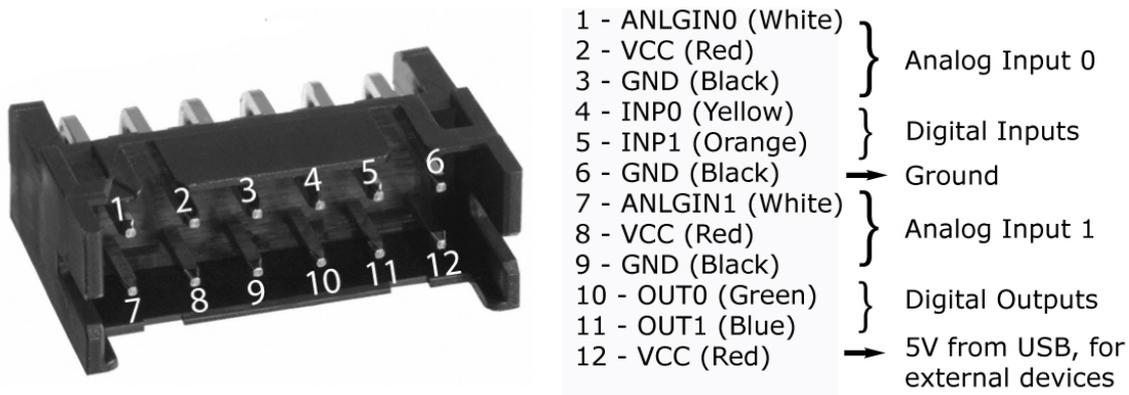
For a step-by-step guide on getting Phidgets running on Linux, check the Linux page.

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

I/O Wiring Diagram



Replacing the I/O Interface Cable

If your I/O interface cable gets damaged, you can remove it and replace it with 3013 - PhidgetInterfaceKit 2/2/2 Replacement Cable. Pull hard on the connector until it comes out.

You can also remove individual wires from the cable if they are not needed. Detach the wire assembly from the 1011, delicately pry up the plastic tab with a small flat-head screwdriver, and pull out the wire. This should only be done for wires that you never plan on needing as it is very easy to break the plastic tab.

Warning: The cable has been designed to be permanently attached to the 1011. We strongly recommend that you limit the number of times you remove the wire assembly from the 1011 to as few times as possible since repeated removals and insertions will eventually damage both connectors.

These connectors are commonly available- their part numbers are listed below. For part numbers on the analog sensor cables, see the mechanical section of the Analog Input Primer.

Cable Connectors

Manufacturer	Part Number	Description
Hirose Electric	DF11-12DP-2DS(24)	2mm Double-Row Connector (Right Angle Pin Header)
Hirose Electric	DF11-12DS-2C	2mm Double-Row Connector (Crimping Socket)

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

Further Reading

If you want to know more about the input/output capabilities of the 1011 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 = 2

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 = 2

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 = 2

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.

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.

Product History

Date	Board Revision	Device Version	Comment
April 2011	0	100	Product Release
May 2011	0	101	getLabelString fix for labels > 7 characters
February 2013	0	102	USB stack changes; was unstable with lots of set reports.
August 2013	0	103	USB bug fix. Some OUT messages not getting through

References

[1] http://www.phidgets.com/products.php?product_id=1011

[2] <http://www.digikey.com>

Article Sources and Contributors

1011 User Guide *Source:* http://www.phidgets.com/wiki/index.php?title=1011_User_Guide *Contributors:* Burley, Mparadis, Patrick

Image Sources, Licenses and Contributors

Image:1011.jpg *Source:* <http://www.phidgets.com/wiki/index.php?title=File:1011.jpg> *License:* unknown *Contributors:* Mparadis

File:1011_0_Connecting_The_Hardware.jpg *Source:* http://www.phidgets.com/wiki/index.php?title=File:1011_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:1011_0_Control_Panel_Screen.jpg *Source:* http://www.phidgets.com/wiki/index.php?title=File:1011_0_Control_Panel_Screen.jpg *License:* unknown *Contributors:* Mparadis

File:1011_0_InterfaceKit_Screen.jpg *Source:* http://www.phidgets.com/wiki/index.php?title=File:1011_0_InterfaceKit_Screen.jpg *License:* unknown *Contributors:* Mparadis

File:1011_0_Advanced_Sensor_Screen.jpg *Source:* http://www.phidgets.com/wiki/index.php?title=File:1011_0_Advanced_Sensor_Screen.jpg *License:* unknown *Contributors:* Mparadis

File:1011_0_Connector_Diagram.jpg *Source:* http://www.phidgets.com/wiki/index.php?title=File:1011_0_Connector_Diagram.jpg *License:* unknown *Contributors:* Mparadis