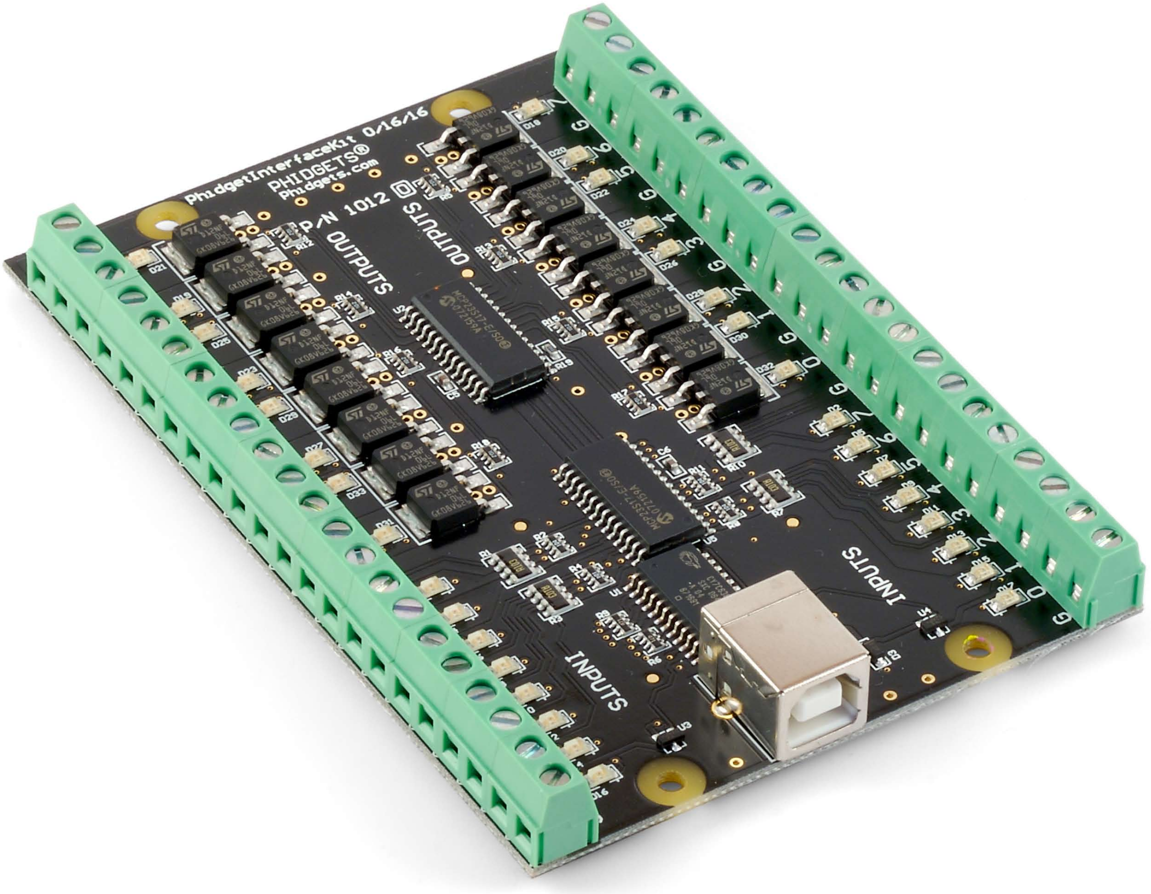
1012 - PhidgetInterfaceKit 0/16/16

For board revision 0



Programming Environment

Operating Systems: Windows 2000/XP/Vista, Windows CE, Linux, and Mac OS X

Programming Languages (APIs): VB6, VB.NET, C#.NET, C++, Flash 9, Flex, Java, LabVIEW, Python, Max/MSP, and Cocoa.

Examples: Many example applications for all the operating systems and development environments above are available for download at www.phidgets.com.

What Can the PhidgetInterfaceKit 0/16/16 Do?

The PhidgetInterfaceKit 0/16/16 board is designed to be the core of a wide variety of projects, providing digital-control interfacing with many types of external devices. It is a USB based controller with:

- 16 Digital Inputs sensing up to 30VDC
- 16 Digital Outputs controlling up to 30VDC
- LED Indicators on all I/O channels

This InterfaceKit can be used in projects that need to switch loads like incandescent lights, relays, solenoids, and motors. Digital inputs can be used to convey the state of push buttons, limit switches, or relays. They are activated by a 4 to 30 VDC signal. Digital outputs can be used to drive LEDs, solid state relays, or transistors; they can sink up to 2A at 30V. The outputs do not provide power, they just act as a switch to ground.

Phidgets are modular devices; if there is something that the PhidgetInterfaceKit 0/16/16 cannot do – for example reading analog sensors or LED dimming – add another Phidget designed for that purpose.

Getting Started

Installing the hardware

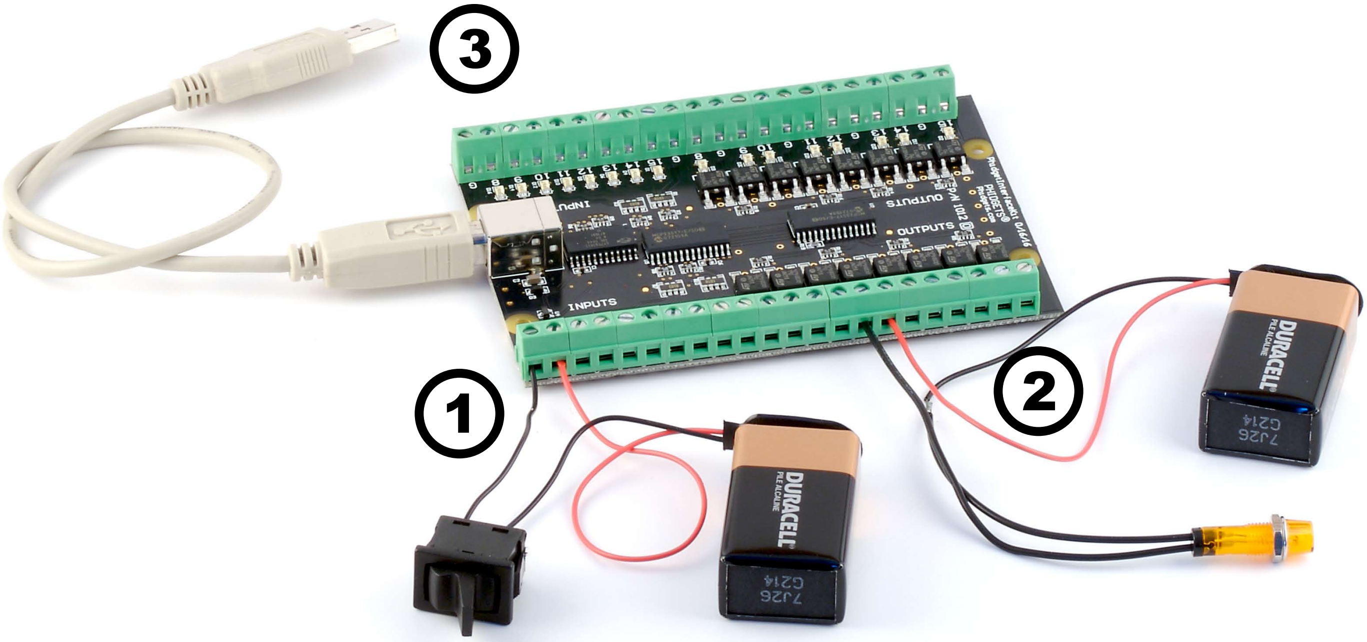
The kit contains:

- A Phidget Interface Kit
- A USB cable

You will also need:

- Two 9V batteries
- Two 9V battery connectors
- A switch
- An incandescent bulb

Connecting all the pieces



1. Connect the black wire (-) from the battery to one of the switch wires. Connect the other switch wire to the ground terminal. Connect the red wire (+) to the input terminal block number 0.
2. Connect the black wire (-) from the battery to one of the bulb wires. Connect the red wire (+) to the output terminal block number 3. Connect the other bulb wire to the ground terminal.
3. Connect the board to the PC using the USB cable.

Downloading and Installing the software

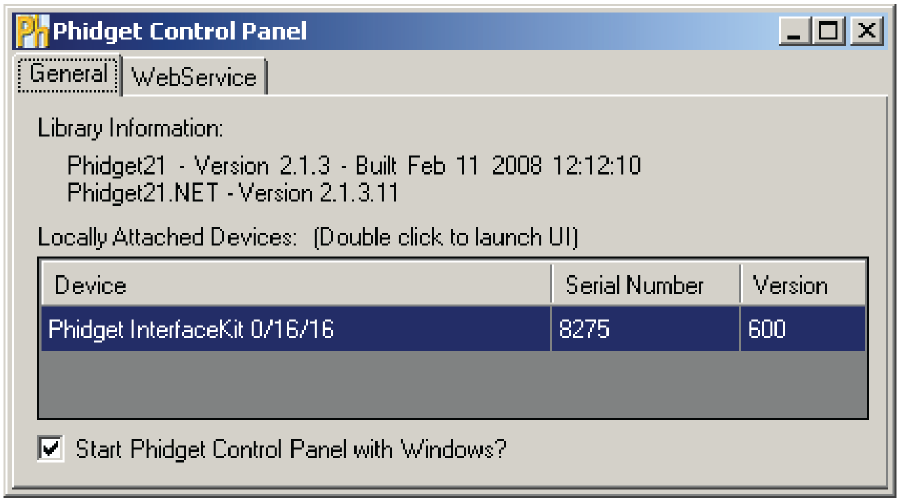
If you are using Windows 2000/XP/Vista

Go to www.phidgets.com >> Downloads >> Windows

Download and run Phidget21.MSI

You should see the Ph icon on the right hand corner of the Task Bar.

Testing the PhidgetInterfaceKit 0/16/16 Functionality



Phidget Control Panel

General | WebService

Library Information:

Phidget21 - Version 2.1.3 - Built Feb 11 2008 12:12:10
Phidget21.NET - Version 2.1.3.11

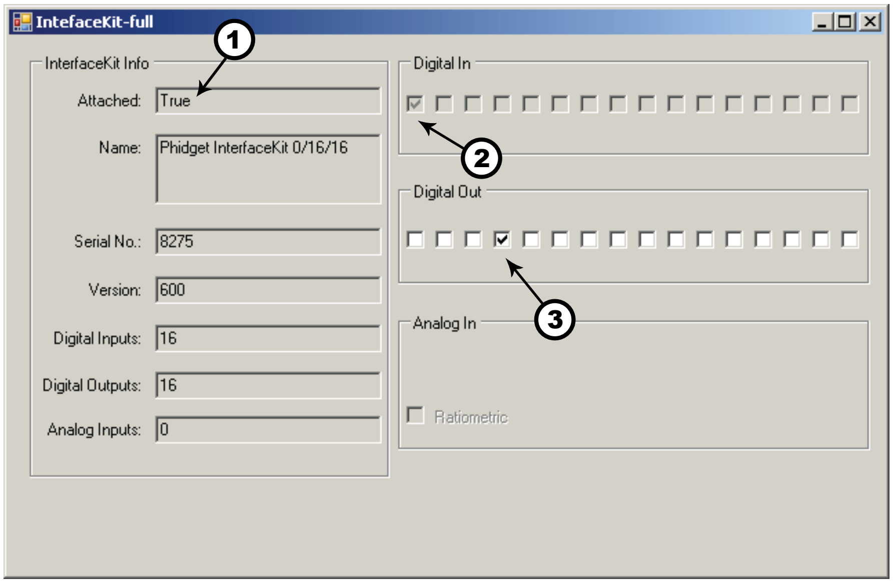
Locally Attached Devices: (Double click to launch UI)

Device	Serial Number	Version
Phidget InterfaceKit 0/16/16	8275	600

☑ Start Phidget Control Panel with Windows?

Double Click on the Ph icon to activate the Phidget Control Panel and make sure that ***PhidgetInterfaceKit 0/16/16*** is properly attached to your PC.

1. Double Click on ***PhidgetInterfaceKit 0/16/16*** in the Phidget Control Panel to bring up interfaceKit-full and check that the box labelled Attached contains the word True.
2. To test the digital input, toggle the switch on and off. When on, a tick mark will appear in the Digital In box and The on-board LED will also turn on; the tick mark will disappear when the switch is off and the on-board LED light will go off.



IntefaceKit-full

InterfaceKit Info

Attached: True

Name: Phidget InterfaceKit 0/16/16

Serial No.: 8275

Version: 600

Digital Inputs: 16

Digital Outputs: 16

Analog Inputs: 0

Digital In

☑ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

Digital Out

□ □ □ ☑ □ □ □ □ □ □ □ □ □ □ □ □

Analog In

□ Ratiometric

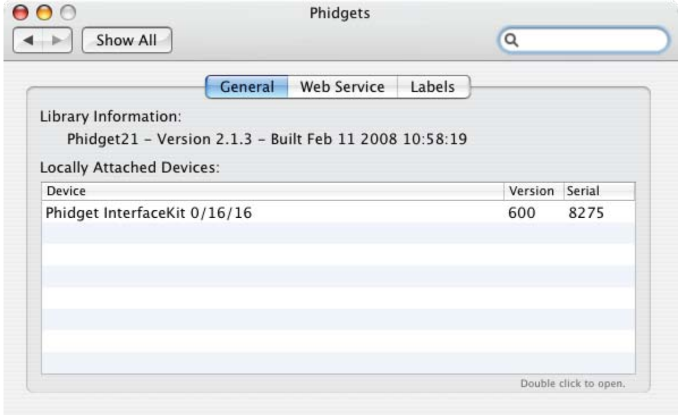
3. To test the digital output, put a tick mark in the digital out box and both the on-board LED and the incandescent bulb will turn on. If you click on the box again the tick mark will go away and both LEDs will turn off.

If you are using Mac OS X

Go to www.phidgets.com >> downloads >> Mac

Download Mac OS X Framework

Testing the PhidgetInterfaceKit 0/16/16 functionality



Library Information:

Phidget21 - Version 2.1.3 - Built Feb 11 2008 10:58:19

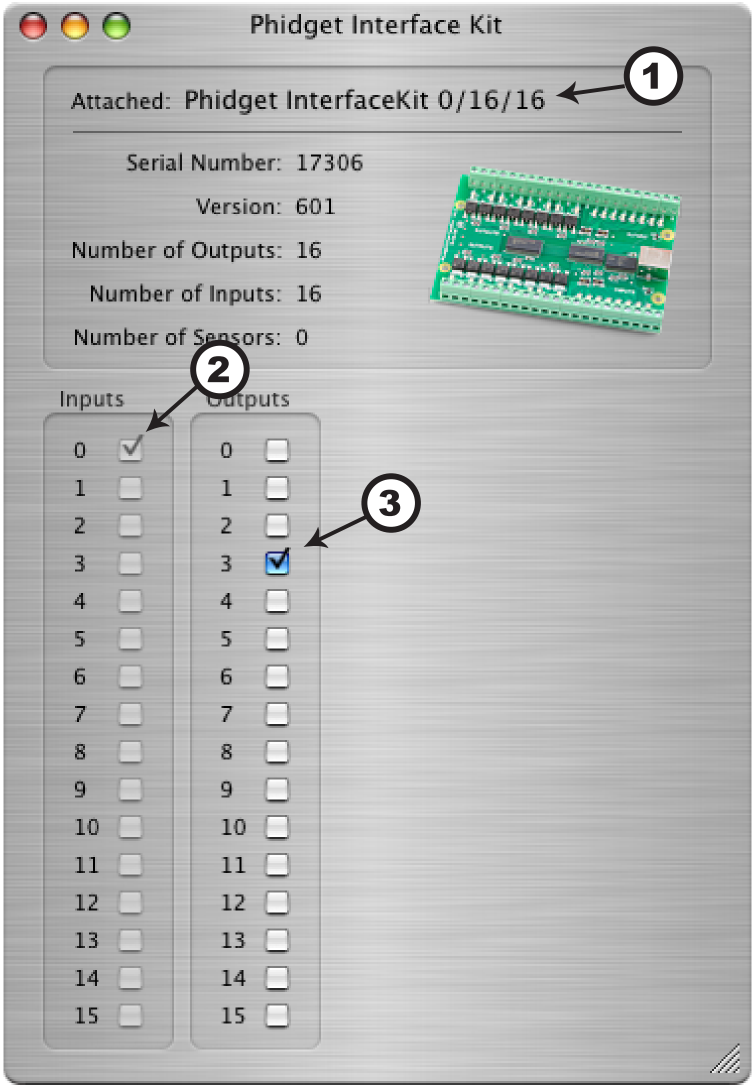
Locally Attached Devices:

Device	Version	Serial
Phidget InterfaceKit 0/16/16	600	8275

Double click to open.

Click on System Preferences >> Phidgets (under Other) to activate the Phidgets Preference Pane. Make sure that the ***PhidgetInterfaceKit 0/16/16*** is properly attached.

1. Double Click on ***PhidgetInterfaceKit 0/16/16*** in the Phidget Preference Pane to bring up the Phidget Interface Kit Example and check that the PhidgetInterfaceKit 0/16/16 is attached.
2. To test the digital input, toggle the switch on and off. When on, a tick mark will appear in the Inputs box and The on-board LED will also turn on; the tick mark will disappear when the switch is off and the on-board LED light will go off.
3. To test the digital output, put a tick mark in the Outputs box and both the on-board LED and the incandescent bulb will turn on. If you click on the box again the tick mark will go away and both LEDs will turn off.



Attached: Phidget InterfaceKit 0/16/16

Serial Number: 17306

Version: 601

Number of Outputs: 16

Number of Inputs: 16

Number of Sensors: 0

Inputs		Outputs	
0	☑	0	□
1	□	1	□
2	□	2	□
3	□	3	☑
4	□	4	□
5	□	5	□
6	□	6	□
7	□	7	□
8	□	8	□
9	□	9	□
10	□	10	□
11	□	11	□
12	□	12	□
13	□	13	□
14	□	14	□
15	□	15	□

Programming a Phidget

Where to get information

- Go to www.phidgets.com >> downloads
- Select the Operating System and the language you want to use.
- Download the appropriate API manual and read the section under the PhidgetInterfaceKit heading.
- Have a look at the source code of the InterfaceKit-full program.
- Have a look at the C# example below.
- Modify an existing program or write your own program from scratch.

Simple example written in C#

```
/* - InterfaceKit simple -
 ****************************************************************************************
 * This simple example creates an Interfacekit object, hooks the event handlers, and
 * opens it for connections to IntefaceKit Phidgets.  It will then wait for user input to
 * terminate, and in the meantime, display event generated data from the InterfaceKit.
 * For a more detailed example, please see the InterfaceKit-full example.
 *
 * Please note that this example was designed to work with only one Phidget InterfaceKit
 * connected. For an example using multiple Phidget InterfaceKits, please see a
 * “multiple” example in the InterfaceKit Examples folder.

 * Copyright 2007 Phidgets Inc.
 * This work is licensed under the Creative Commons Attribution 2.5 Canada License.
 * To view a copy of this license, visit http://creativecommons.org/licenses/by/2.5/ca/
 */

using System;
using System.Collections.Generic;
using System.Text;
//Needed for the InterfaceKit class, phidget base classes, and the PhidgetException class
using Phidgets;
//Needed for the event handling classes
using Phidgets.Events;
namespace InterfaceKit_simple
{
    class Program
    {
        //Declare an InterfaceKit object
        static InterfaceKit ifKit;

        static void Main(string[] args)
        {
            try
            {
                //Initialize the InterfaceKit object
                ifKit = new InterfaceKit();

                //Hook the basica event handlers
                ifKit.Attach += new AttachEventHandler(ifKit_Attach);
                ifKit.Detach += new DetachEventHandler(ifKit_Detach);
                ifKit.Error += new ErrorEventHandler(ifKit_Error);

                //Hook the phidget spcific event handlers
                ifKit.InputChange += new InputChangeEventHandler(ifKit_InputChange);
                ifKit.OutputChange += new OutputChangeEventHandler(ifKit_OutputChange);
                ifKit.SensorChange += new SensorChangeEventHandler(ifKit_SensorChange);

                //Open the object for device connections
                ifKit.open();

                //Wait for an InterfaceKit phidget to be attached
                Console.WriteLine(“Waiting for InterfaceKit to be attached...”);
                ifKit.waitForAttachment();

                //Wait for user input so that we can wait and watch for some event data
                //from the phidget
                Console.WriteLine(“Press any key to end...”);
                Console.Read();

                //User input was rad so we’ll terminate the program, so close the object
                ifKit.close();

                //set the object to null to get it out of memory
                ifKit = null;

                //If no expcetions where thrown at this point it is safe to terminate
                //the program
                Console.WriteLine(“ok”);
            }
            catch (PhidgetException ex)
            {
                Console.WriteLine(ex.Description);
            }
        }

        //Attach event handler...Display the serial number of the attached InterfaceKit
        //to the console
        static void ifKit_Attach(object sender, AttachEventArgs e)
        {
            Console.WriteLine(“InterfaceKit {0} attached!”,
                                e.Device.SerialNumber.ToString());
        }

        //Detach event handler...Display the serial number of the detached InterfaceKit
        //to the console
        static void ifKit_Detach(object sender, DetachEventArgs e)
        {
            Console.WriteLine(“InterfaceKit {0} detached!”,
                                e.Device.SerialNumber.ToString());
        }

        //Error event handler...Display the error description to the console
        static void ifKit_Error(object sender, ErrorEventArgs e)
        {
            Console.WriteLine(e.Description);
        }

        //Input Change event handler...Display the input index and the new value to the
        //console
        static void ifKit_InputChange(object sender, InputChangeEventArgs e)
        {
            Console.WriteLine(“Input index {0} value (1)”, e.Index, e.Value.ToString());
        }

        //Output change event handler...Display the output index and the new valu to
        //the console
        static void ifKit_OutputChange(object sender, OutputChangeEventArgs e)
        {
            Console.WriteLine(“Output index {0} value {0}”, e.Index, e.Value.ToString());
        }

        //Sensor Change event handler...Display the sensor index and it’s new value to
        //the console
        static void ifKit_SensorChange(object sender, SensorChangeEventArgs e)
        {
            Console.WriteLine(“Sensor index {0} value {1}”, e.Index, e.Value);
        }
    }
}
```

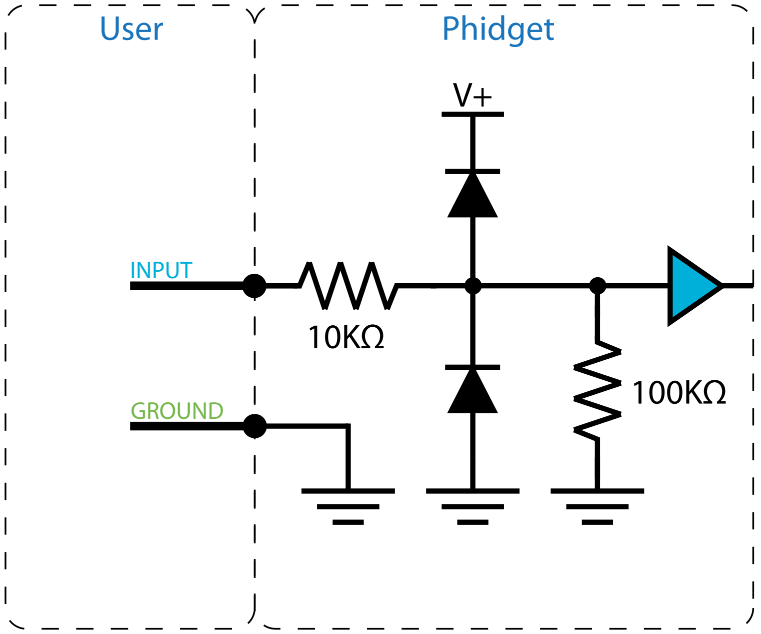
Learning more ...

- Check out the forums
- Check out the Phidgets projects

Technical Section

Using the Digital Inputs

The Digital Inputs are capable of sensing up to 30VDC. A voltage of 4VDC to 30VDC will be read as True or logical 1; below 1VDC will be read as a False or logical 0. The input is high impedance, which means current flowing into the Phidget device will be limited. Ground terminals are provided in multiple locations along the input terminal strip; it is recommended that the ground terminal located nearest the input terminal be used.



Digital Input Diagram

Using the Digital Outputs

The Digital Outputs require an external voltage source to supply power, and act as a switch to ground. The outputs can sink up to 2A at 30V, and are designed to operate with DC voltage only. This type of switching output is often referred to as a low-side switch. It is common to use the Digital Outputs with electrical devices such as motors, lamps, relays, and solenoids. When using ground terminals, it is recommended that the ground terminal located nearest the output terminal be used. If highly inductive loads are used with the Digital Outputs, the use of a clamping diode is recommended.

Ground Protection

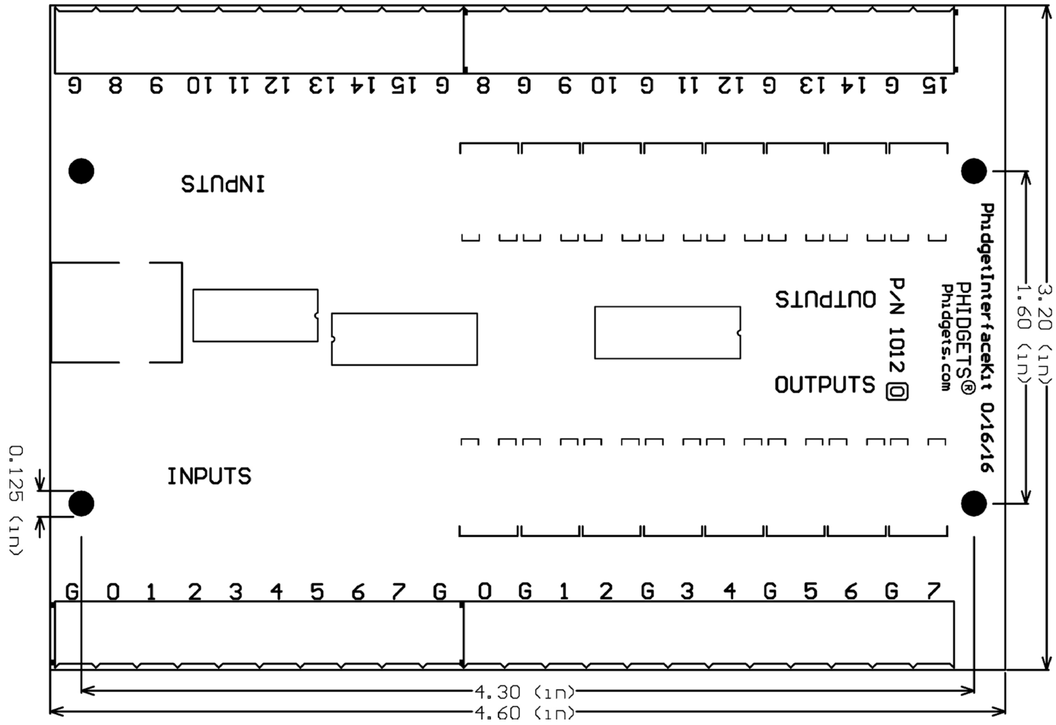
Ground terminals on this InterfaceKit share a common ground with USB ground. Because they are not internally isolated, these terminals will expose the USB ground potential of the PC to which they are connected. Be sure you are completely familiar with any circuit you intend to connect to the InterfaceKit before it is connected. If a reverse voltage or dangerously high potential is applied to the input or output terminals, damage to the Phidget or the PC may result. Limit input and output voltages to 30VDC, and always observe correct polarity.

Device Specifications

Digital Input Impedance (-0.5V to +5.5V)	110kΩ
Digital Input Impedance (>5.5V)	10kΩ
Digital Output Impedance (on)	0.2Ω
Digital Input Update Rate	125 Updates/second
Digital Output Update Rate	125 Updates/second
Digital Output Current Sinking (30V)	2A max
USB Power Current Specification	500mA max
Device Quiescent Current Consumption	18mA
Device Active Current Consumption	120mA max

Mechanical Drawing

1:1 scale



G 0 1 2 3 4 5 6 7 G 0 G 1 2 G 3 4 G 5 6 G 7
G 8 9 10 11 12 13 14 15 G 8 G 9 10 G 11 12 G 13 14 G 15
INPUTS
INPUTS
OUTPUTS
OUTPUTS
P/N 1012 0
PhidgetInterfaceKit 0/16/16
PHIDGETS®
Phidgets.com
0.125 (in)
3.20 (in)
1.60 (in)
4.30 (in)
4.60 (in)

Product History

Date	Product Revision	Comment
January 2003	DeviceVersion 600	Product Release
January 2004	DeviceVersion 601	Added State Echoing