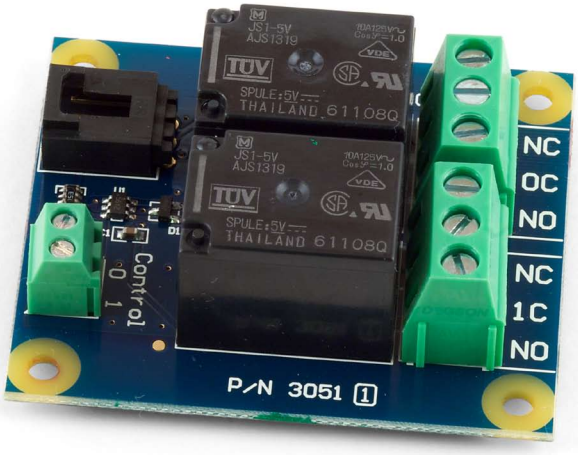
Product Manual

3051 - Dual Relay Board



Phidgets 3051 - Product Manual

For Board Revision 1

Contents

Product Features

- The Dual Relay Board allows digital outputs of the 1018 - PhidgetInterfaceKit 8/8/8, the 1202 - PhidgetTextLCD or the 1203 - PhidgetTextLCD to control larger loads and devices like AC or DC motors, electromagnets, solenoids, and incandescent light bulbs.
- The 3051 contains 2 relays for switching AC or DC power.
- The relays are Single Pole Double Throw (SPDT)
- The relays can switch up to 240 V AC at 10 Amps and 100 V DC at 5 Amps.

Connections

Designed to connect to a:

- 1018 - PhidgetInterfaceKit 8/8/8
- 1019 - PhidgetInterfaceKit 8/8/8 w/6 Port Hub
- 1070 - PhidgetSBC
- 1202 or 1203 - PhidgetTextLCD

Getting Started

Checking the Contents

You should have received:

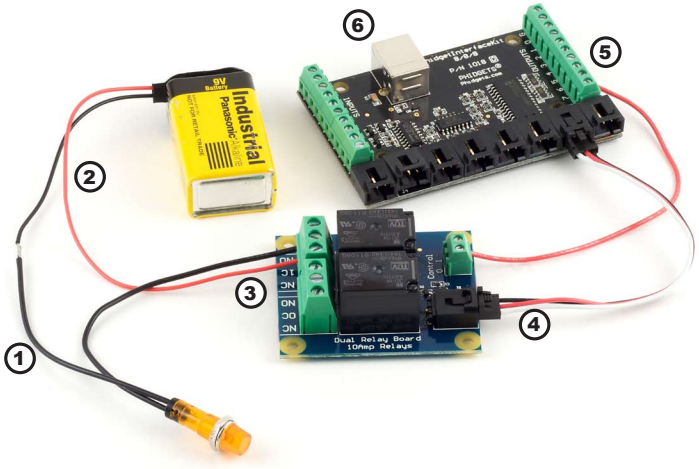
- A Dual Relay Board
- A Sensor Cable

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

- A PhidgetInterfaceKit 8/8/8 or a PhidgetTextLCD
- A USB Cable
- A 9V battery, a battery connector
- A piece of wire
- An incandescent bulb

Connecting all the pieces

1. Connect the black/negative(-) wire from battery connector to one of the bulb wire.
2. Connect the red/positive (+) wire from the battery connector to the 1C (Common) connector on the Dual Relay Board.
3. Connect the other bulb wire to the NO (Normally Open) connector on the Dual Relay Board.
4. Connect the Dual Relay Board to the InterfaceKit 8/8/8 using the sensor cable.
5. Connect Control 1 on the Dual Relay Board to Digital Output 6 on the InterfaceKit 8/8/8 using a piece of wire.
6. Connect the 1018 - PhidgetInterfacekit to your PC using the USB cable.



Testing Using Windows 2000/XP/Vista

Downloading the Phidgets drivers

Make sure that you have the current version of the Phidget library installed on your PC. If you don't, do the following:

Go to www.phidgets.com >> Drivers

Download and run Phidget21 Installer (32-bit, or 64-bit, depending on your PC)

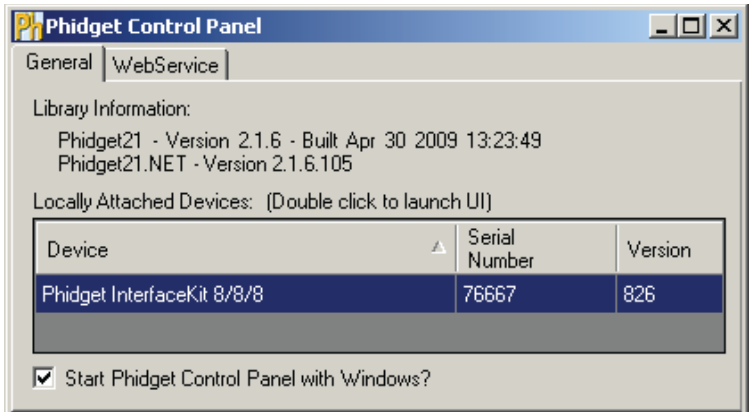
You should see the  icon on the right hand corner of the Task Bar.

Running Phidgets Sample Program

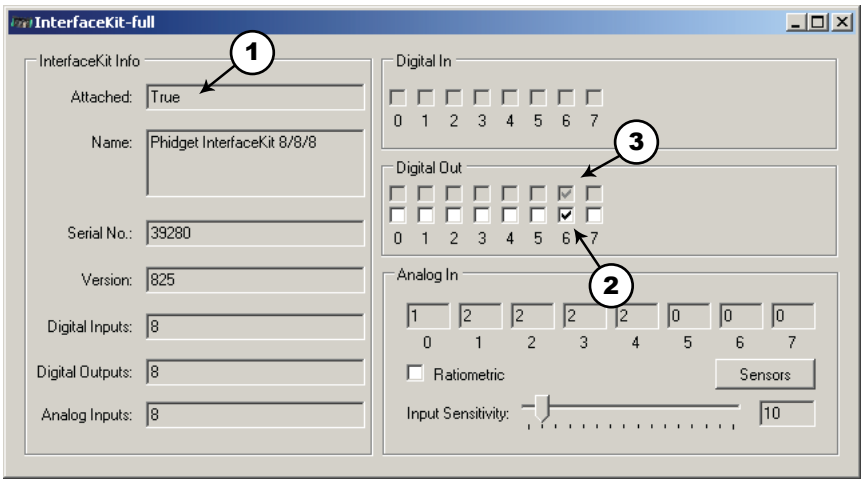
Double clicking on the  icon loads the Phidget Control Panel; we will use this program to make sure that your new Phidget works properly. Since the sensor is connected to a 1018, the computer will see only the 1018. The sensor is providing data through the Analog input it is connected to.

The source code for the InterfaceKit-full sample program can be found under C# by clicking on www.phidgets.com >> Programming.

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. Click on the Digital Out box. A tick mark appears in the box and the bulb lights up. Click on the box again. The tick mark goes away and light goes out. If you unplug the Dual Relay Board while the light is on, it will go off. Move the bulb wire from NO to NC (Normally Closed). Now the light is on when there is no tick mark and off when there is. If you unplug the Dual Relay Board when the light is on, it will stay on.



3. The bottom row of the digital out shows the status of the request, while the top row displays the status of the digital output as reported by the 3051 Relay.

Testing Using Mac OS X

- Click on System Preferences >> Phidgets (under Other) to activate the Preference Pane
- Make sure that the Phidget InterfaceKit 8/8/8 is properly attached.
- Double Click on Phidget InterfaceKit 8/8/8 in the Phidget Preference Pane to bring up the InterfaceKit-Full example. This example will function in a similar way as the Windows version, but note that it does not include an Advanced Sensor Display.

Programming a Phidget

Phidgets' philosophy is that you do not have to be an electrical engineer in order to do projects that use devices like sensors, motors, motor controllers, and interface boards. All you need to know is how to program. We have developed a complete set of Application Programming Interfaces (API) that are supported for Windows, Mac OS X, and Linux. When it comes to languages, we support VB6, VB.NET, C#.NET, C, C++, Flash 9, Flex, Java, LabVIEW, Python, Max/MSP, and Cocoa.

Code Samples

We have written sample programs to illustrate how the APIs are used.

Due to the large number of languages and devices we support, we cannot provide examples in every language for every Phidget. Some of the examples are very minimal, and other examples will have a full-featured GUI allowing all the functionality of the device to be explored. Most developers start by modifying existing examples until they have an understanding of the architecture.

Go to www.phidgets.com >> Programming to see if there are code samples written for your device. Find the language you want to use and click on the magnifying glass besides "Code Sample". You will get a list of all the devices for which we wrote code samples in that language.

If this is your first time writing a program to control a Phidget, you should read the Getting Started Guide for the language you are planning to use.

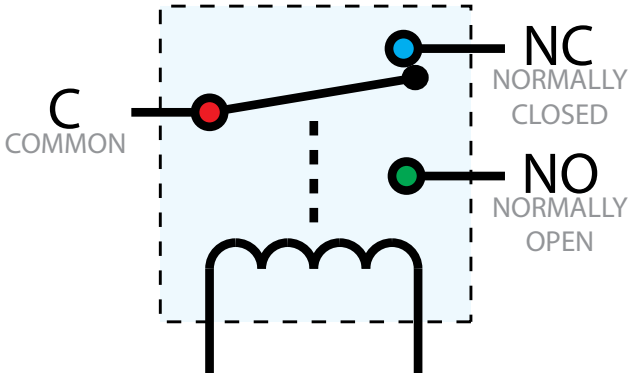
Coding for your Sensor

Phidget analog sensors do not have their own API, but instead their output is a voltage that is converted to a digital value and accessed through the SensorValue properties and events on a PhidgetInterfaceKit. It is not possible to programmatically identify which sensor is attached to the Analog Input. Your application will need to apply any formulas from this manual to the SensorValue to translate it into usable data.

See the PhidgetInterfaceKit product manual for an overview of its API and a description of our architecture.

Relays

A relay is an electrically-controlled switch. Although many types of electrical switches exist, a relay's mechanical nature gives it the advantage of reliability and current-switching capacity. The main disadvantage to using mechanical relays is their limited life-span, as opposed to solid state relays who do not suffer from this drawback.



Using a Digital Output Relay

Relays have a connection scheme determined by the arrangement of contacts within the relay. Because relays are a type of switch, they are defined in the same way other electromechanical switches are defined.

In switch schemes, the number of poles represents the number of common terminals a switch has, and the number of throws represents the number of switchable terminals that exist for each pole. The relays used in the Dual Relay Board are SPDT relays: single pole, double throw. The internal construction of this type of relay is depicted in the diagram above. Many other types of relays exist: SPST, DPDT, and DPST, to name a few.

In an SPDT relay, one of the throw terminals is labelled Normally Closed (NC), and the other is labelled Normally Open (NO). As the name indicates, the normally closed terminal is the terminal connected to common when the relay coil is not powered. When the relay coil is energized by the relay control circuit, the electromagnetic field of the coil forces the switch element inside the relay to break its contact with the normally closed terminal and make contact with the normally open terminal. The switch element would then connect the normally open terminal and the common terminal.

Wetting Current

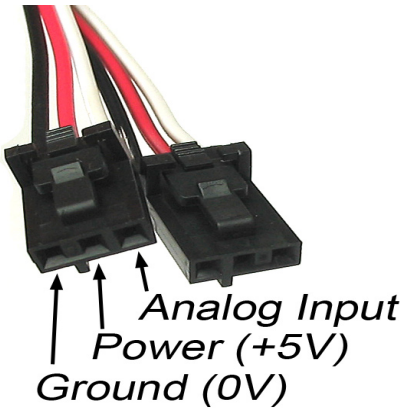
When a relay is in one switch position for a period of time, oxidation of the open contact(s) can occur. Depending upon the internal coating material of the contacts, oxide films of varying density will be displaced upon the surface of open contacts; this film acts as an insulator to current flow. When the relay is switched, a certain amount of current flowing through the contacts, known as the wetting current, is required to remove the film of oxides and ensure proper conduction. Because of this requirement, these relays are not reliable for signal switching. See the device specification on page 10 for detailed requirements.

Load Noise

If highly inductive loads are used with the Dual Relay Board, it is recommended that a noise limiting component be used to prevent damage to the device. An MOV, TVS diode, or kickback diode (for DC applications) shunted across the load will assist in dissipating voltage transients.

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 labeled. The connectors are commonly available - refer to the Table below for manufacturer part numbers.

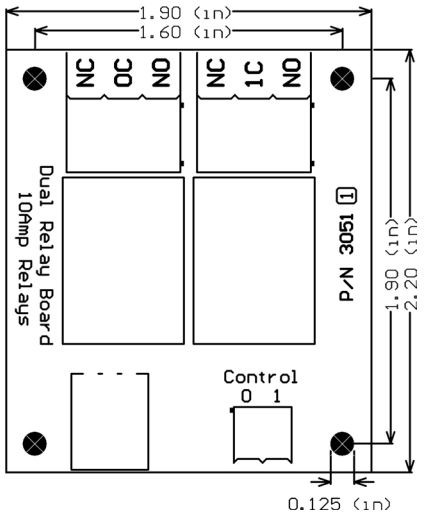


Cable Connectors		
Manufacturer	**Part Number**	**Description**
Molex	50-57-9403	3 Position Cable Connector
Molex	16-02-0102	Wire Crimp Insert for Cable Connector
Molex	70543-0002	3 Position Vertical PCB Connector
Molex	70553-0002	3 Position Right-Angle PCB Connector (Gold)
Molex	70553-0037	3 Position 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 www.digikey.com

Mechanical Drawing

1:1 scale



Note: When printing the mechanical drawing, "**Page Scaling**" in the Print panel must be set to "**None**" to avoid re-sizing the image.

Device Specifications

Characteristic	Value
Contact Resistance (max)	0.1 ohms
Minimum Switching Current (Wetting Current)	100 mA @ 5 V DC
Maximum DC Switching Voltage	100 V DC
Maximum DC Switching Current	5 A
Maximum AC Switching Voltage	250 V AC
Maximum AC Switching Current	10 A
Minimum Switching Current (Wetting Current)	100 mA @ 5 V DC
Maximum Operating Speed (Contacts Per Minute)	20 cpm
Operate time	10 ms
Recommended Terminal Wire Size	12 - 24 AWG
Terminal Wire Strip Length	5 - 6mm (0.196” - 0.236”)

Product History

Date	Board Revision	Comment
October 2007	0	Product Release
September 2008	1	Bigger connectors, Bigger board

Support

Call the support desk at 1.403.282.7335 8:00 AM to 5:00 PM Mountain Time (US & Canada) - GMT-07:00

or

E-mail us at: support@phidgets.com