# HUB0000 User Guide

From Phidgets Support

## Getting Started

Before you get started with plugging in and setting up your VINT Hub, we recommend downloading our libraries from here.

## Phidget Control Panel

In order to demonstrate the functionality of the HUB0000, the Phidget Control Panel running on a Windows machine will be used.

The Phidget Control Panel is available for use on both macOS and Windows machines. If you would like to follow along, first take a look at the getting started guide for your operating system:

- Getting started with Windows
- Getting started with macOS

Linux users can follow the getting started with Linux guide and continue reading here for more information about the HUB0000.

## First Look

After plugging the HUB0000 into your computer and opening the Phidget Control Panel, you will see something like this:
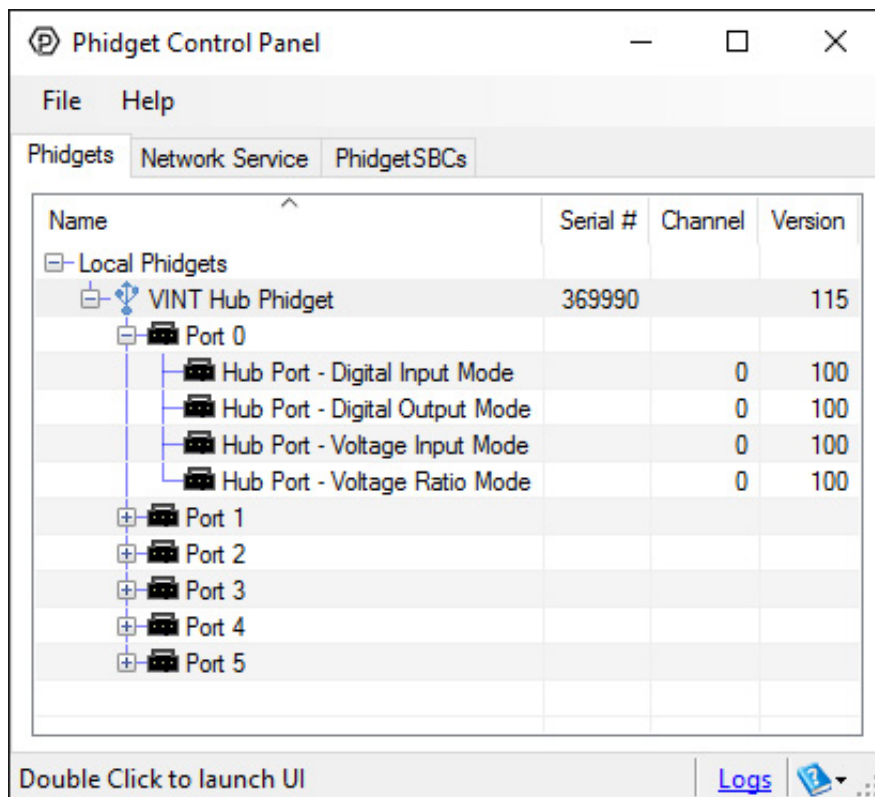


The Phidget Control Panel will list all connected Phidgets and associated objects, as well as the following information:

- **Serial number:** allows you to differentiate between similar Phidgets.
- **Channel:** allows you to differentiate between similar objects on a Phidget.
- **Version number:** corresponds to the firmware version your Phidget is running. If your Phidget is listed in
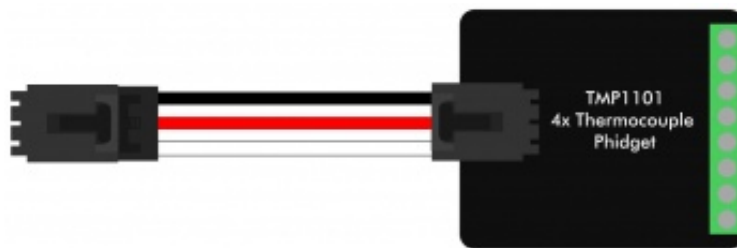
red, your firmware is out of date. Update the firmware by double-clicking the entry.

The Phidget Control Panel can also be used to test your device. Double-clicking on an object will open an example.

# Using the HUB0000

This table will help you decide where to look next to get your devices plugged in and running:

## VINT Device Mode



This mode is for connecting to a VINT device. These devices differ from analog sensors because instead of just reporting 0 to 5 volts, they communicate with the VINT Hub. The communication is digital, therefore it is immune to electrical interference between the hub and the device.

## Digital Input Mode
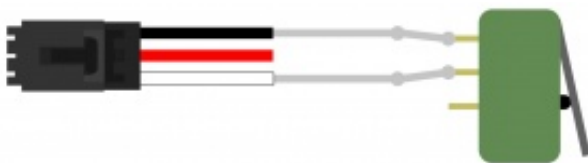


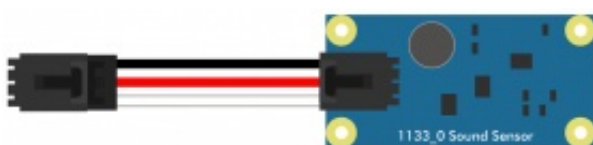In digital input mode, the VINT Hub port can act as an active-low digital input. This mode is great for reading the state of buttons and switches.

## Voltage Ratio Input Mode



In voltage ratio input mode, the VINT Hub port will read the voltage on the white wire and compare it to the voltage being supplied on the red wire. This mode will let you read any ratiometric Phidgets sensor.

## Voltage Input Mode



In voltage input mode, the VINT Hub port will read the voltage on the white wire. This can be used to interface a non-ratiometric sensor, or

## Digital Output Mode



In digital output mode, a VINT Hub port can behave like a 3.3V digital output. You could use this mode to blink an LED or switch on a MOSFET. And don't worry if

to measure the voltage in a 5V digital circuit.

you still need a 5V digital output; there are VINT modules available that provide multiple 5V digital outputs on a single VINT port.

Now that you have everything together, let's start using the HUB0000!

## VINT Device

A VINT Device is any Phidget that must be controlled by a VINT Hub instead of plugging directly into your computer via USB. You can find a complete list of VINT devices here.

When you double click on an VINT Device in the control panel, a window will open with controls and readouts for that specific class of Phidget. For instructions on how to use this example, find the **user guide** for that Phidget.
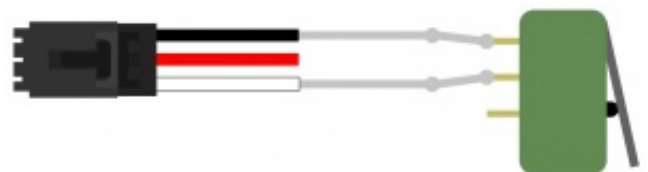


## Digital Input

### *Using the Digital Inputs*

Here are some circuit diagrams that illustrate how to connect various devices to the digital inputs on your Phidget.
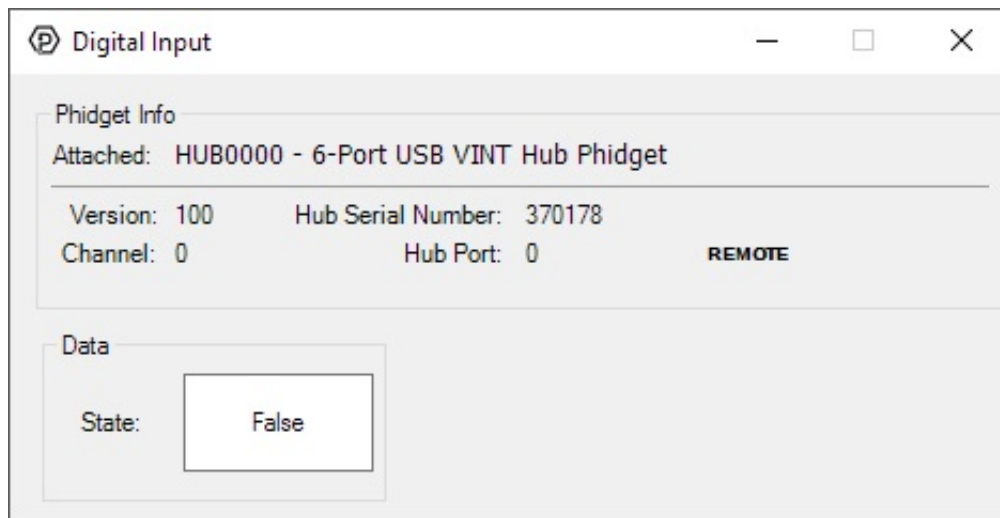
**Wiring a switch to a Digital Input**

Connect the switch between the white and black wires of a Phidget cable, and plug the cable into a VINT port. Wiring the switch this way will make the input TRUE when the switch is closed, and FALSE when the switch is open.



### *Using the Control Pannel*

Double-click on a Digital Input object in order to run the example:

Digital Input — □ ✕

Phidget Info
Attached:  HUB0000 - 6-Port USB VINT Hub Phidget

Version: 100          Hub Serial Number:  370178
Channel: 0                        Hub Port:  0          **REMOTE**

Data

State:      False

General information about the selected object will be displayed at the top of the window. You can also experiment with the following functionality:

- This is an active-low device, therefore, it will be *true* when connected to ground, and *false* when connected to a high voltage.

For more information about Digital Inputs, take a look at the Digital Input Primer

### *Next Step: Finding The Addressing Information*

To use Phidgets in programs other than the Control Panel, you will need to take note of the Addressing Information for the device. Continue to the Finding The Addressing Information section for more information and to get started with your own programs.

## Digital Output

### *Using the Digital Outputs*

Here are some diagrams that illustrate how to connect various devices to the digital outputs on your Phidget.
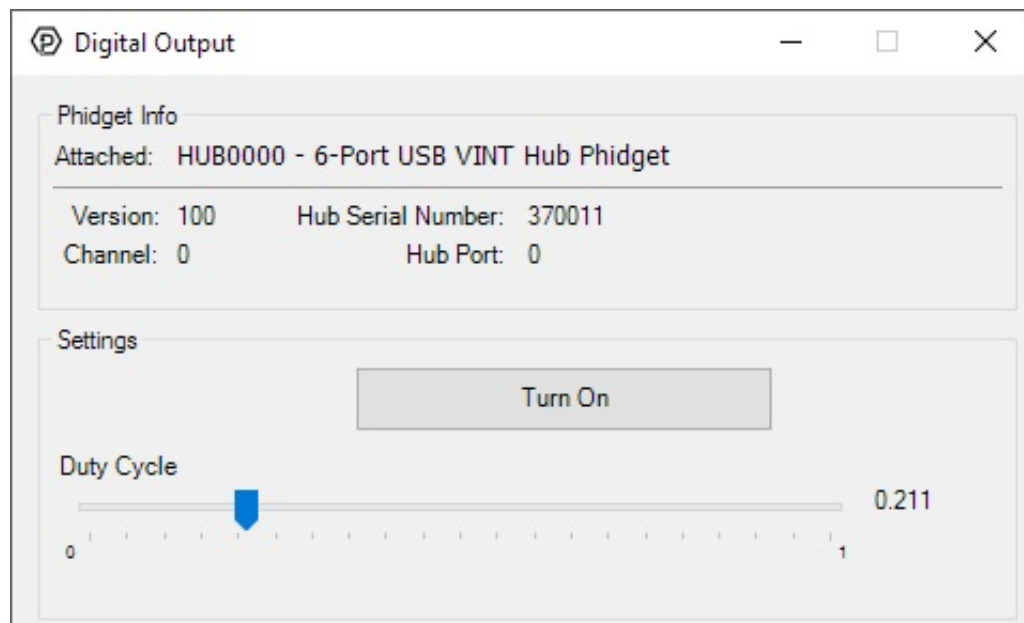
**Driving an LED**

Connecting an LED to a digital output is simple. Wire the anode (long side) to the white wire on the Phidget cable, and the cathode to the black wire of the Phidget cable. If you wire it backwards, the LED will not light but no harm will come to the system.



### *Using the Control Pannel*

Double-click on a Digital Output object in order to run the example:



General information about the selected object will be displayed at the top of the window. You can also experiment with the following functionality:

- Toggle the state of the digital output by pressing the button.
- Use the *Duty Cycle* slider to start using pulse-width modulation. Rather than simply switching on or off, you can select a *Duty Cycle* to limit the power being supplied to a specific level. This allows you to control devices like a dimmer would.

For more information about Digital Outputs, check out the Digital Output Primer.

### *Next Step: Finding The Addressing Information*

To use Phidgets in programs other than the Control Panel, you will need to take note of the Addressing Information for the device. Continue to the Finding The Addressing Information section for more information and to get started with your own programs.

## Voltage Input

### *Using the Voltage Inputs*

Here are some examples of how you can connect various devices to the voltage inputs on your Phidget:
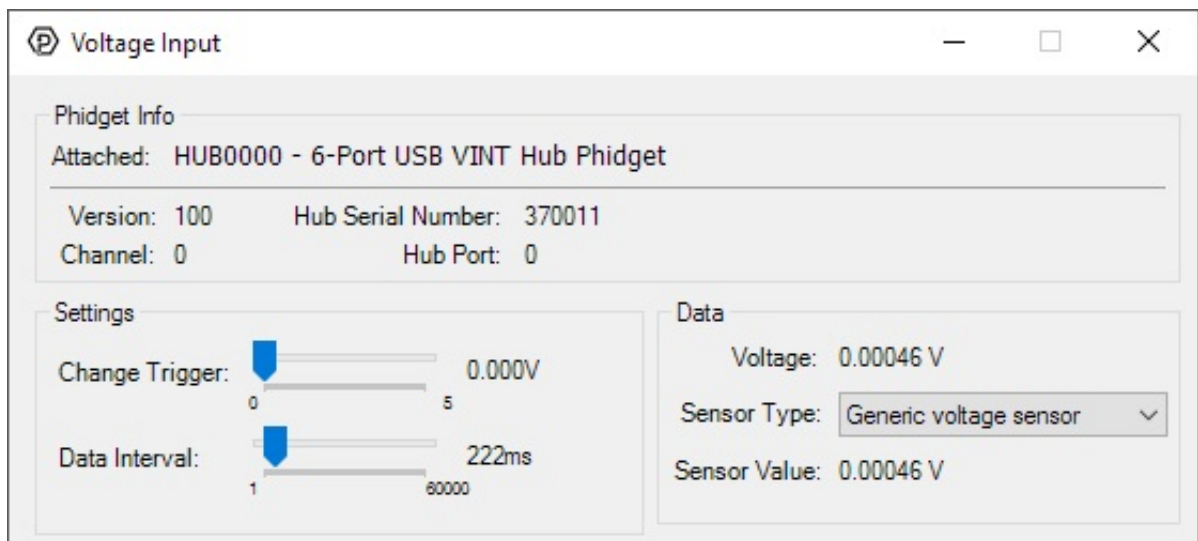
**Connect a Sensor**

Connecting to a Phidget sensor is as simple as plugging it into the VINT Port with a Phidget cable.

*Using the Control Pannel*

Double-click on a Voltage Input object in order to run the example:



General information about the selected object will be displayed at the top of the window. You can also experiment with the following functionality:

- Modify the change trigger and/or data interval value by dragging the sliders. For more information on these settings, see the data interval/change trigger page.
- If you have an analog sensor connected that you bought from us, you can select it from the *Sensor Type* drop-down menu. The example will then convert the voltage into a more meaningful value based on your sensor, with units included, and display it beside the *Sensor Value* label. Converting voltage to a *Sensor Value* is not specific to this example, it is handled by the Phidget libraries, with functions you have access to when you begin developing!

For more information about Voltage Inputs, check out the Voltage Input Primer.

*Next Step: Finding The Addressing Information*

To use Phidgets in programs other than the Control Panel, you will need to take note of the Addressing Information for the device. Continue to the Finding The Addressing Information section for more information and to get started with your own programs.

# Voltage Ratio Input

## *Using the Voltage Ratio Inputs*

Here are some examples of how you can connect various devices to the voltage ratio inputs on your Phidget:
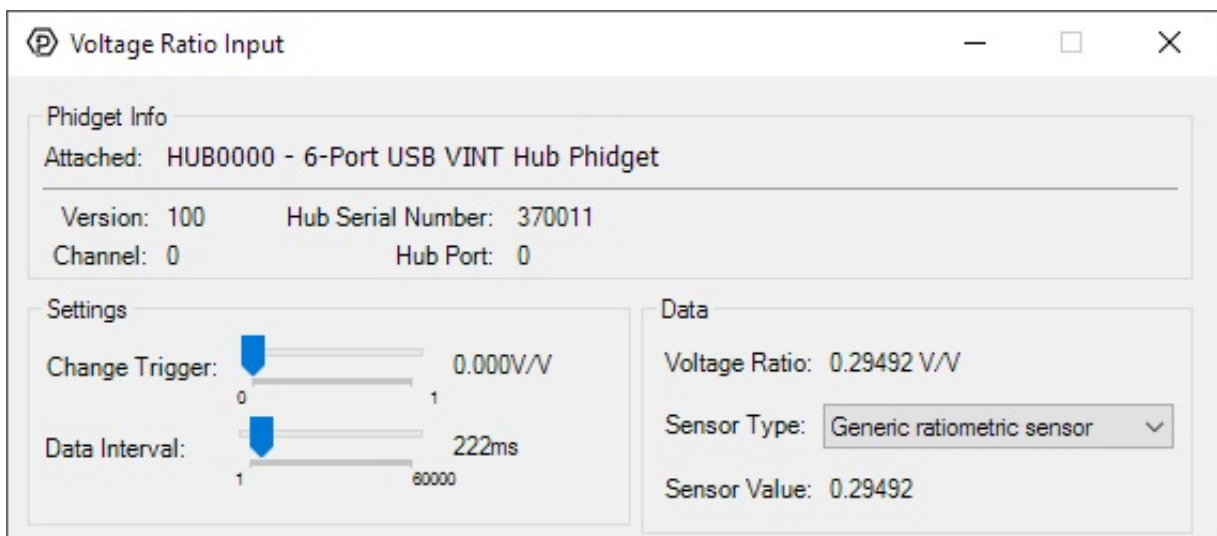
**Connect a Sensor**

Connecting to a ratiometric sensor is as simple as plugging it into the VINT Port with a Phidget cable.

## *Using the Control Pannel*

Double-click on a Voltage Ratio Input object in order to run the example:

General information about the selected object will be displayed at the top of the window. You can also experiment with the following functionality:

- The voltage ratio is reported in Volts per Volt. For example, if the Phidget is providing 5V and the sensor is sending back 2.5V, the ratio will be 0.5V/V.
- Modify the change trigger and/or data interval value by dragging the sliders. For more information on these settings, see the data interval/change trigger page.
- If you have an analog sensor connected that you bought from us, you can select it from the Sensor Type drop-down menu. The example will then convert the voltage into a more meaningful value based on your sensor, with units included, and display it beside the Sensor Value label. Converting voltage to a Sensor Value is not specific to this example, it is handled by the Phidget libraries, with functions you have access to when you begin developing!

For more information about Voltage Ratio Inputs, check out the Voltage Ratio Input Primer.

### *Next Step: Finding The Addressing Information*

To use Phidgets in programs other than the Control Panel, you will need to take note of the Addressing Information for the device. Continue to the Finding The Addressing Information section for more information and to get started with your own programs.

## Finding The Addressing Information

Before you can access the device in your own code, and from our examples, you'll need to take note of the addressing parameters for your Phidget. These will indicate how the Phidget is physically connected to your application. For simplicity, these parameters can be found by clicking the button at the top of the Control Panel example for that Phidget.
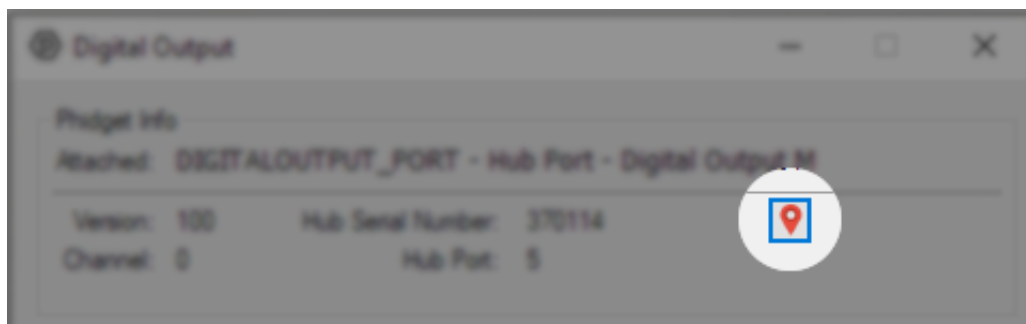


In the Addressing Information window, the section above the line displays information you will need to connect to your Phidget from any application. In particular, note the *Channel Class* field as this will be the API you will need to use with your Phidget, and the type of example you should use to get started with it. The section below the line provides information about the network the Phidget is connected on if it is attached remotely. Keep track of these parameters moving forward, as you will need them once you start running our examples or your own code.

## Using Your Own Program

You are now ready to start writing your own code for the device. The best way to do that is to start from our examples:

The the HUB0000 can be used by itself with the DigitalInput Examples, the DigitalOutput Examples, the VoltageInput Examples, and the VoltageRatioInput Examples. To find examples for a Phidget connected to the HUB0000, look at its **User Guide** for more information.

Once you have your example, you will need to follow the instructions on the page for your programming language to get it running. To find these instructions, select your programming language from the Software Overview page.

# Technical Details

## VINT Ports

For more information on the capabilities of the ports on the VINT Hub, see the VINT Primer.

## Setting the Device Label

When opening channels with Phidgets, you can set a number of properties to make sure you match the precise channel you want. The HUB0000 has a unique serial number which can be used to find its ports or any VINT devices connected to them. If you want something more customizable and human-readable, you can use the device label instead. You can customize the device label by using `writeDeviceLabel` on the `Hub` object after it's been opened.

Once a device label has been written, you can use it to address any of the VINT Hub's channels, or the channels of any VINT device connected to the hub.

# What to do Next

- Software Overview - Find your preferred programming language here to learn how to write your own code with Phidgets!
- General Phidget Programming - Read this general guide to the various aspects of programming with Phidgets. Learn how to log data into a spreadsheet, use Phidgets over the network, and much more.
- Phidget22 API - The API is a universal library of all functions and definitions for programming with

Phidgets. Just select your language and device and it'll give you a complete list of all properties, methods, events, and enumerations that are at your disposal.

---