

Touch Wheel Phidget

Contents

1 GETTING STARTED

2 USING THE HIN1001

2.1 Phidget Control Panel

2.1.1 Windows

2.1.2 macOS

2.2 First Look

2.3 Capacitive Touch

3 TECHNICAL DETAILS

3.1 Interaction with Data Interval

3.2 Re-Calibration

3.3 Relative Change in Position

4 WHAT TO DO NEXT

Getting Started

- HIN1001 Touch Wheel Phidget
- VINT Hub
- USB cable and computer
- Phidget cable

Next, you will need to connect the pieces:



1. Connect the VINT Hub to your computer with a USB cable.
2. Connect the HIN1001 to the VINT Hub using the Phidget cable.

Using the HIN1001

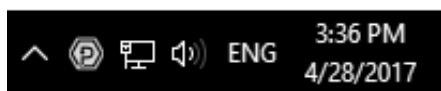
Phidget Control Panel

In order to demonstrate the functionality of the HIN1001, 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.

Windows

To open the Phidget Control Panel on Windows, find the  icon in the taskbar. If it is not there, open up the start menu and search for Phidget Control Panel



macOS

To open the Phidget Control Panel on macOS, open Finder and navigate to the Phidget Control Panel in the Applications list. Double click on the  icon to bring up the Phidget Control Panel.

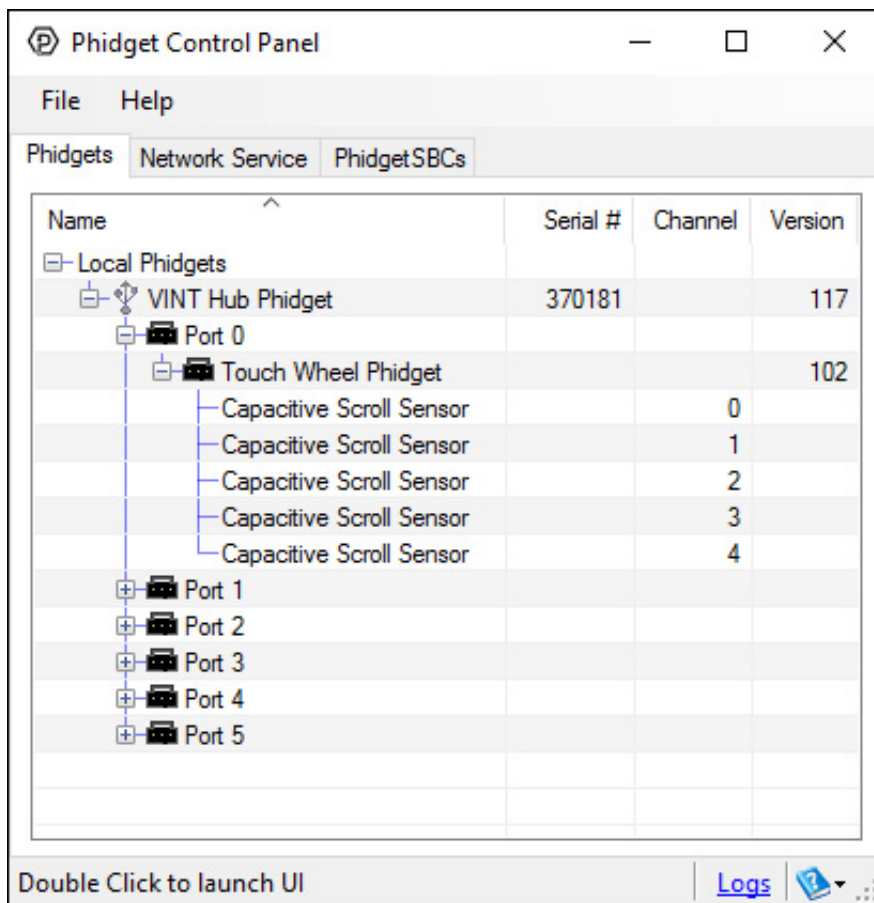
For more information, take a look at the getting started guide for your operating system:

- 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 HIN1001.

First Look

After plugging the HIN1001 into your computer and opening the Phidget Control Panel, you will see something like this:

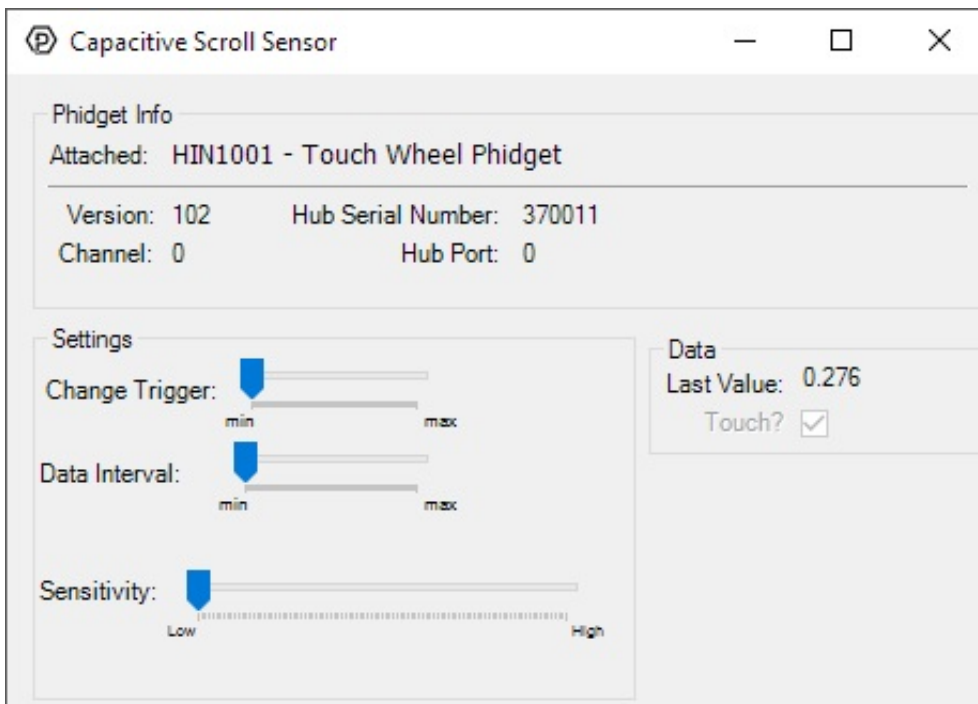


The Phidget Control Panel will list all connected Phidgets and associated objects, as well as the following information:

- **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.

Capacitive Touch



When you double click on a Capacitive Touch object, a window like the one pictured will open.

- At the top of the window, information about your device and the properties of this particular channel will be listed.
- On the left, change trigger and/or data interval can be changed. For more information on these settings, see the Data Rate/Change Trigger page. In addition, the sensor's sensitivity can be adjusted. The higher the sensitivity, the more susceptible the sensor will be to sensing touch.
- On the right, the **Touch?** check box will be checked if an object is touching or nearly touching the surface associated with this capacitive touch object. For capacitive touch objects that have a linear or circular touch region, the **Last Value** corresponds to the location on the region that the current or most recent touch occurred.

Technical Details

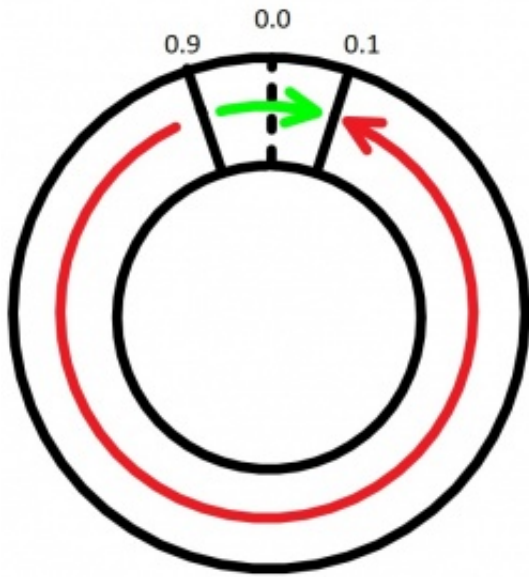
Interaction with Data Interval

Unlike most other Phidgets, setting a data interval on the HIN1001 affects its initial response time, rather than an overall data rate. After the first touch has been detected, the device will stream data as fast as possible (up to every 16ms, as data becomes available). This means that if setting a data interval of 250ms, the first touch may need to be held up to 250ms to wake the device, after which subsequent touches on all channels will be detected much faster, until no touch is detected for a few seconds. Once the device has been left alone for a few seconds, it will resume its low-power state and wait for the next touch.

Re-Calibration

Since capacitive touch sensors rely on checking capacitance against a threshold to detect when they are touched, they need to keep a good baseline of the capacitance of their environment. The HIN1001 will re-calibrate its baseline levels if it detects a constant touch for more than 50 seconds, in order to prevent getting stuck in a touched state. Similarly, it will re-calibrate its baseline capacitance to adjust to falling capacitance (such as when it is moved away from a surface) in order to prevent getting stuck in a not-touched state. The latter adjustment happens as soon as the baseline is detected to have fallen.

Relative Change in Position



Since the touch wheel is circular, calculating the change in position of a finger over the wheel can be more involved than simply subtracting one position from another. In cases where the value crosses the zero point, a simple subtraction would indicate a large movement in the opposite direction. For example a movement from 0.9 to 0.1 would appear as a movement of -0.8, when a movement of +0.2 is much more likely. As such, a change in position should be calculated taking this into account. Here is an example of how to do the calculation in C:

```
static void CCONV
onTouchListener(PhidgetCapacitiveTouchHandle ch, void *ctx, double touchValue) {

    double change = (touchValue - oldPos);
    static double oldPos;

    if (change > 0.5)
        change -= 1;
    else if (change < -0.5)
        change += 1;

    oldPos = touchValue;

    ...
}
```

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.