4x Relay Phidget
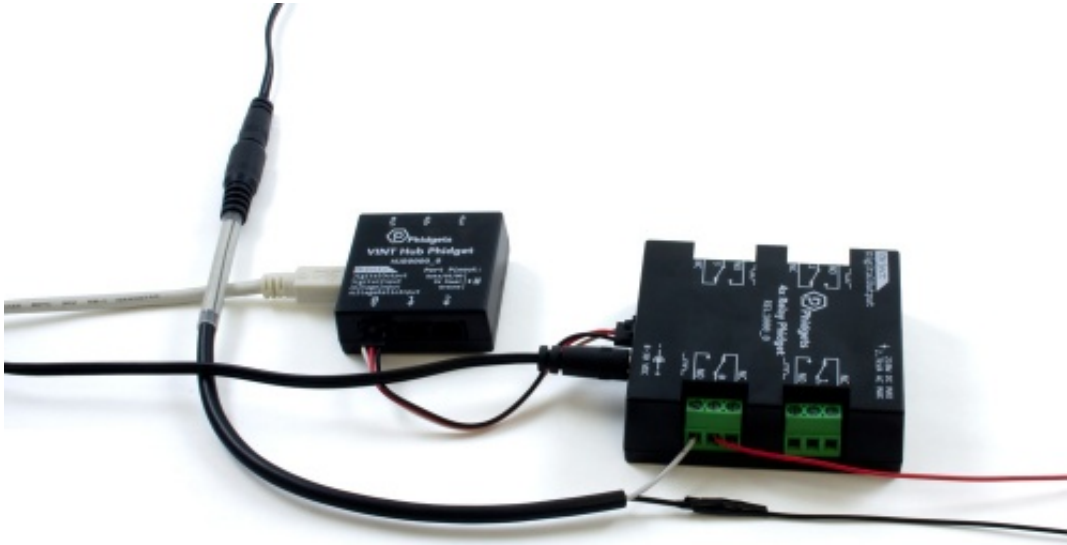
Contents

Getting Started

Welcome to the REL1000 user guide! In order to get started, make sure you have the following hardware on hand:

- REL1000 - 4x Relay Phidget
- VINT Hub
- Phidget cable
- USB cable and computer
- Power supply
- something to use with the REL1000 (e.g. a series circuit)

Next, you will need to connect the pieces:

1. Connect the REL1000 to the VINT Hub using the Phidget cable.
2. Connect the VINT Hub to your computer with a USB cable.
3. Connect the circuits you're switching to the terminal blocks on the REL1000. If you want the circuit to be **closed** by default when the relay is not energized, connect the two ends of the circuit to the **NC** and **COM** terminals. If you want the circuit to be **open** by default, connect it to the **NO** and **COM** terminals instead. It doesn't matter which end of the circuit is connected connected to each of the two terminals in either case.
4. Connect the power supply to the barrel jack on the REL1000.

Now that you have everything together, let's start using the REL1000!

Using the REL1000

Phidget Control Panel

In order to demonstrate the functionality of the REL1000, 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 REL1000.

First Look

After plugging the REL1000 into your computer and opening the Phidget Control Panel, you will see something like this:
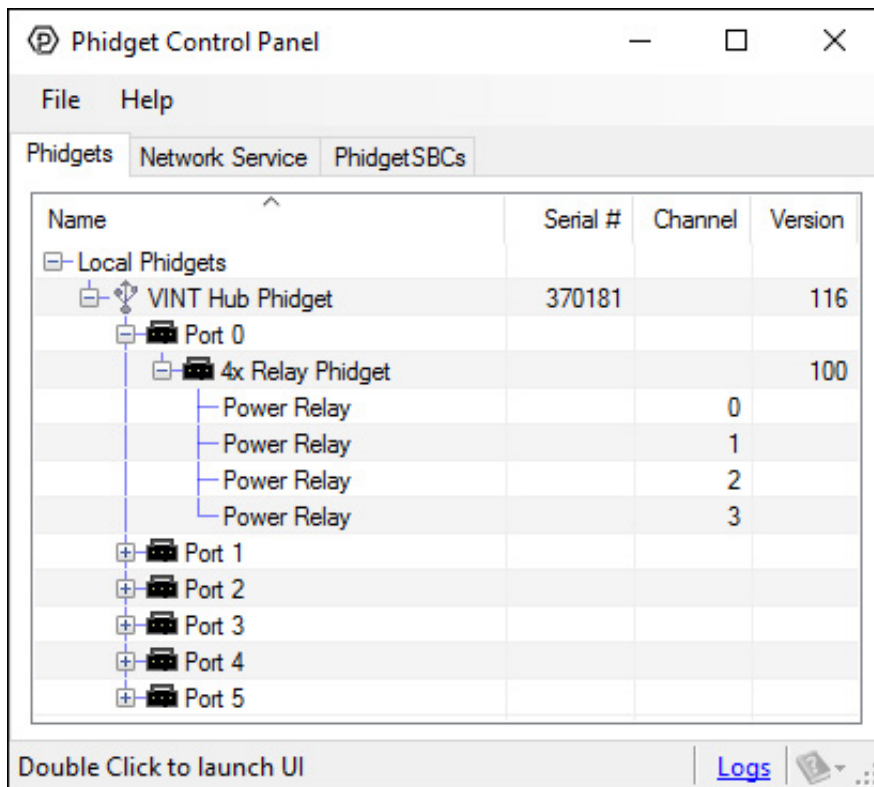

The Phidget Control Panel will list all connected Phidgets and associated objects, as well as the following information:

- **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.

Relay Output

Double-click on a Digital Output object labelled Power Relay in order to run the example:
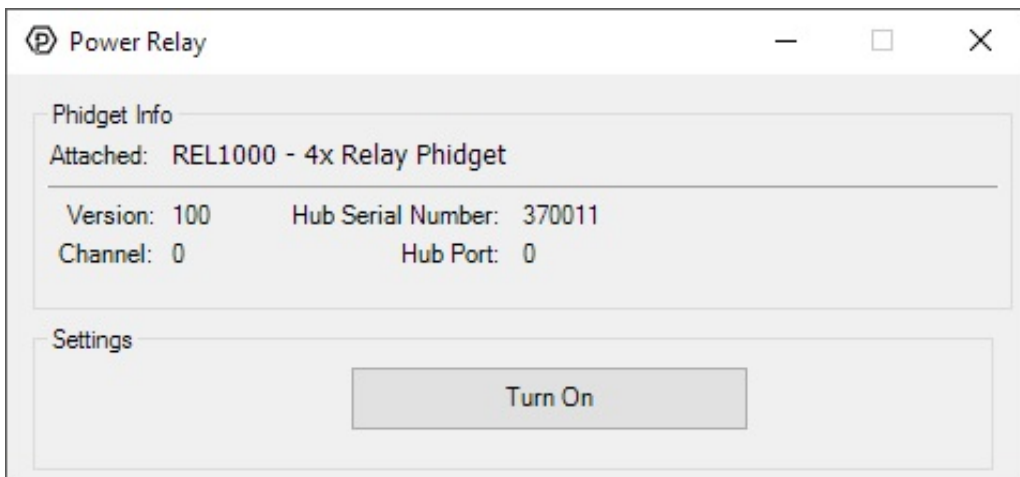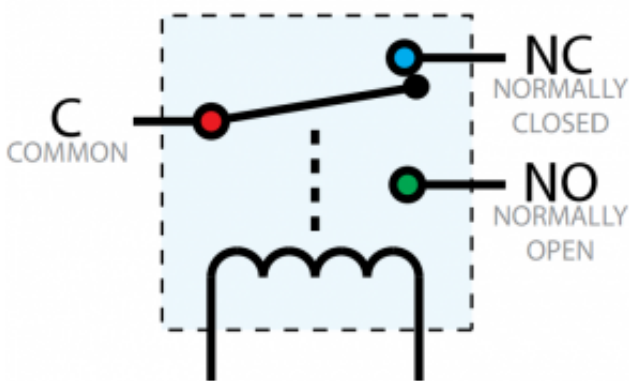
General information about the selected object will be displayed at the top of the window. You can also experiment with the following functionality:

- Toggle the state of the relay by pressing the button.

Technical Details

Safety

The relays on the REL1000 are SPDT (Single pole, double throw). This means there is a common pin (C), a normally open pin (NO) and a normally closed pin (NC). When the relay is unpowered, the switch will be resting in the NC position, as seen in the diagram. When the relay's Digital Output object is toggled in software, it will switch to the NO position. If the Digital Output object is closed (using the `Close()` method), the relay will always return to the NC position. For this reason, it is considered a best practice to call `Close()` at the end of your program on in the 'closing' portion of your project.

If communication between the REL1000 and your computer is broken (e.g. if the Phidget cable is unplugged), the relay will not change state. You will, however, get a detach event for the attached channels of that Phidget, so you may want to handle this case in your Digital Output detach handler. Since communication is already interrupted at this point, you can't tell the Digital Output to return to the NC position, but you can set a warning in your program to notify someone to come and reset the system manually.

Further Reading

You can read more about how mechanical relays work on our Mechanical Relay Primer.

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.