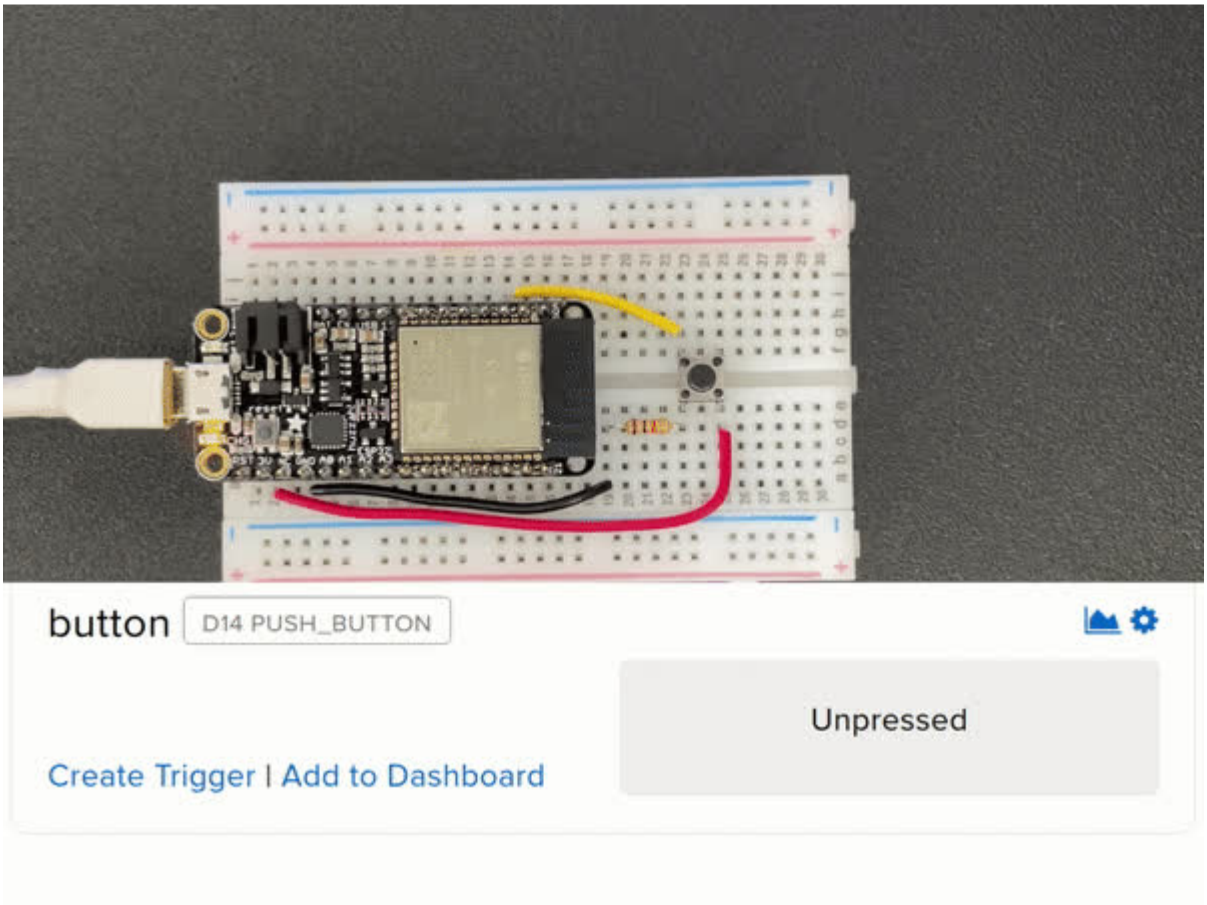
# Digital Inputs with Adafruit IO WipperSnapper

Created by Eva Herrada

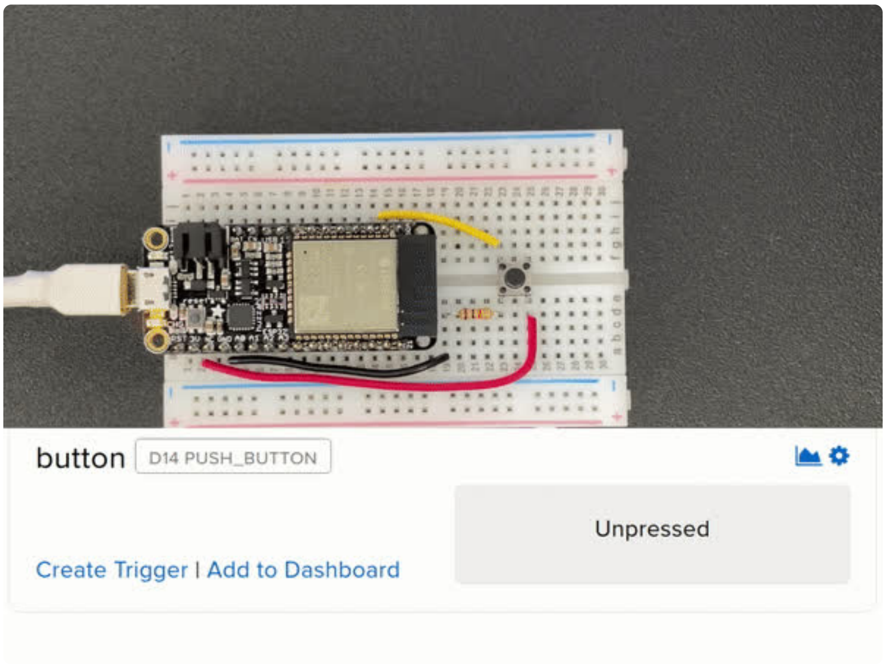


https://learn.adafruit.com/digital-inputs-with-adafruit-io-wippersnapper

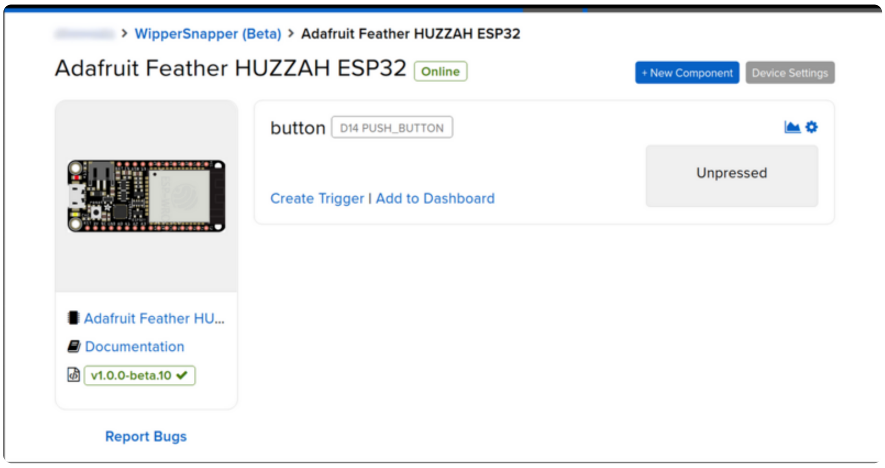
Last updated on 2024-06-03 03:28:17 PM EDT

# Table of Contents

# Overview



This guide is part of a series of guides that cover the basics of using Adafruit IO WipperSnapper. It will show you how to send push-button data to Adafruit IO, without writing a single line of code.

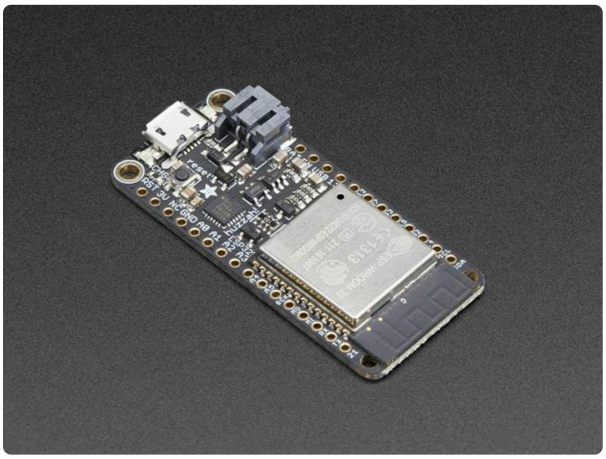


WipperSnapper is an interface for [Adafruit IO, Adafruit's incredibly easy-to-use IoT platform (https://adafru.it/V5A)](https://adafru.it/V5A), designed to turn any WiFi-capable board into an IoT device without programming a single line of code.

Simply load the WipperSnapper firmware onto your board, add credentials, and plug it into power. Your board will automatically register itself with your Adafruit IO account.

From there, you can add components to your board such as buttons, switches, potentiometers, sensors, and more! Components are dynamically added to hardware without the need for re-programming your board.

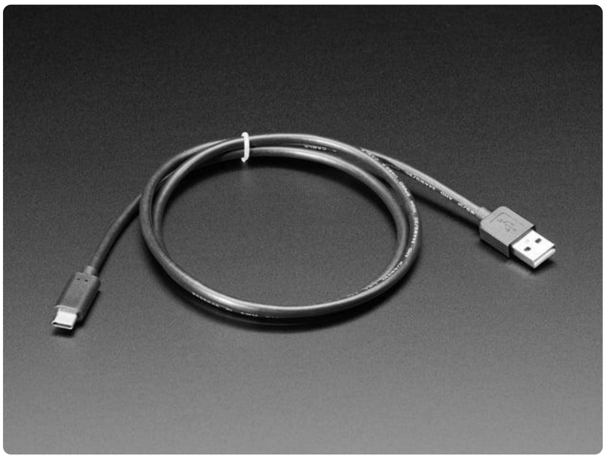
## Parts



Adafruit HUZZAH32 – ESP32 Feather Board

Aww yeah, it's the Feather you have been waiting for! The HUZZAH32 is our ESP32-based Feather, made with the official WROOM32 module. We packed everything you love...

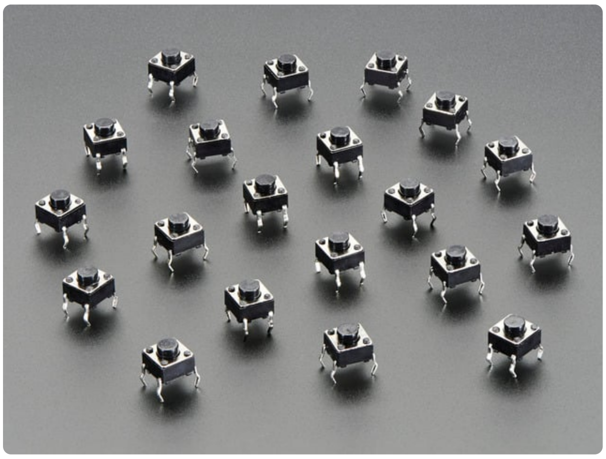
https://www.adafruit.com/product/3405



USB Type A to Type C Cable - approx 1 meter / 3 ft long

As technology changes and adapts, so does Adafruit. This  USB Type A to Type C cable will help you with the transition to USB C, even if you're still...

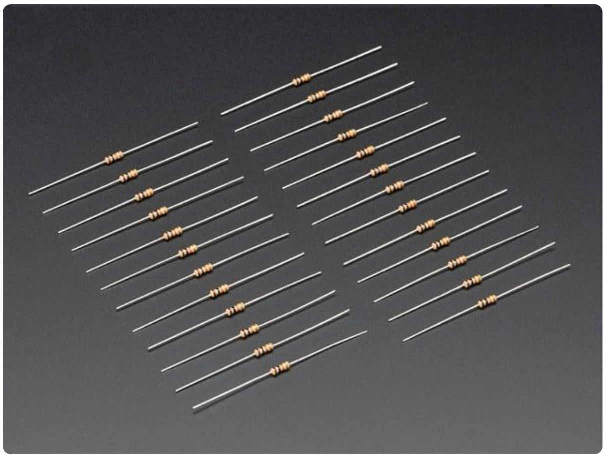
https://www.adafruit.com/product/4474



Tactile Button switch (6mm) x 20 pack

Little clicky switches are standard input "buttons" on electronic projects. These work best in a PCB but

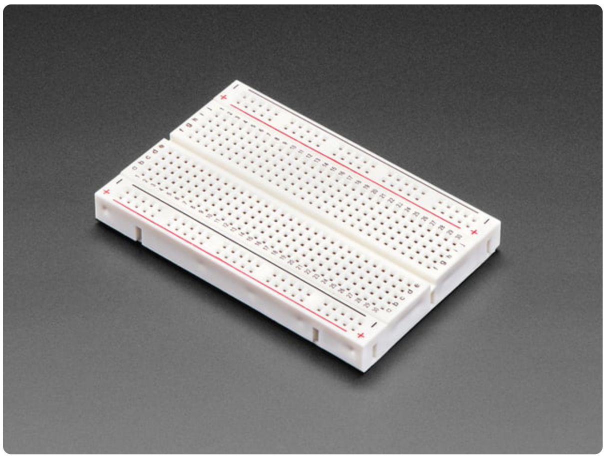
https://www.adafruit.com/product/367



Through-Hole Resistors - 10K ohm 5% 1/4W - Pack of 25

ΩMG! You're not going to be able to resist these handy resistor packs! Well, axially, they do all of the resisting for you!This is a 25 Pack of 10K...

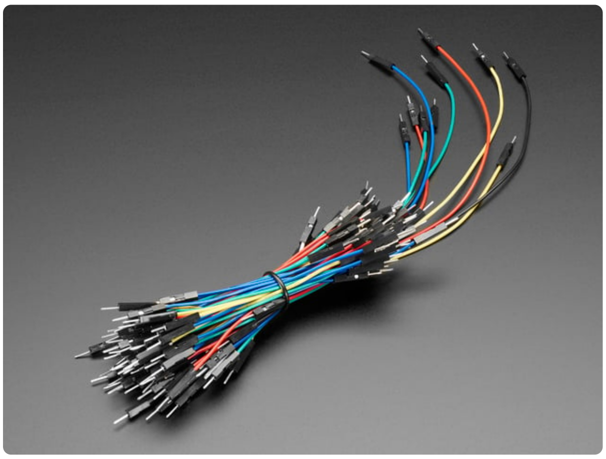
https://www.adafruit.com/product/2784



Half Sized Premium Breadboard - 400 Tie Points

This is a cute, half-size breadboard with 400 tie points, good for small projects. It's 3.25" x 2.2" / 8.3cm x 5.5cm with a standard double-strip in the...

https://www.adafruit.com/product/64



Breadboarding wire bundle

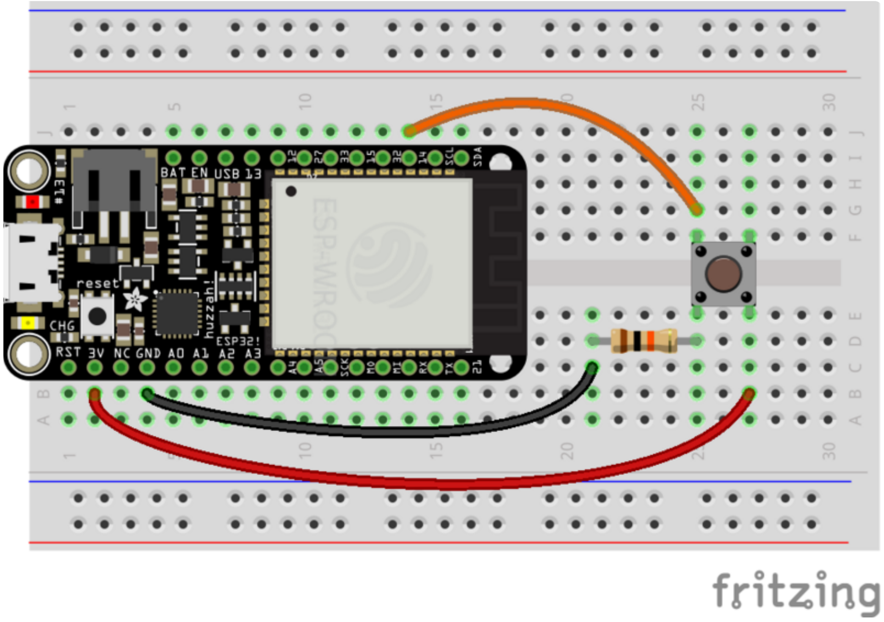
75 flexible stranded core wires with stiff ends molded on in red, orange, yellow, green, blue, brown, black and white. These are a major improvement over the "box of bent...

https://www.adafruit.com/product/153

# Wiring

There are two different ways to wire this project. The first way uses a physical pull-down resistor, and the second one uses a built-in pull-down resistor.

## Physical pull-down wiring



Connect the following wires:

- Feather **GND** to a 10K resistor to the bottom left side of button
    - The resistor is not polarized, so it can be connected in either direction.
- Feather **3V** to the bottom right side of button
- Feather **Pin 14**  to the  upper left side of button

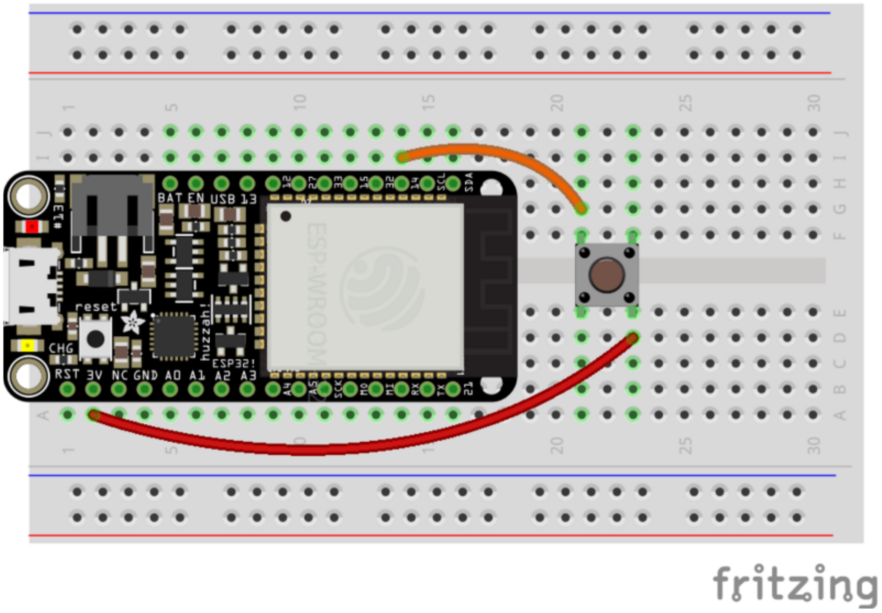
### What is a pull-down resistor?

The resistor being used in this project is really important. It's used to pull the input pin, #14 down. This makes it so that the button doesn't appear to open and close a bunch when it isn't pressed. If you were to switch the red and black wire, it would be pulled up, which would make it show up as 0 when pressed and 1 when not pressed, as opposed to a pull-down which is 0 when not pressed and 1 when pressed.

Resistors used to pull should not be too low, which would short the circuit and always show True, or too high, which would be effectively the same as no resistor at all. Generally, resistance values between 1K and 100K are used, but nominally just use 10K.

Feel free to play around with this yourself once you've set it up with WipperSnapper. If you disconnect the GND wire completely, you'll notice that the state of the button switches back and forth really fast and you get rate limited quickly. Also try switching the red and black wires so that the red one is connected to the resistor and the black one is connected to the lower right pin of the switch. You should see it show up as closed when it's open and open when it's closed.

## Built-in pull-down wiring



BAT EN USB 13
12 27 33 15 32 14 SCL SDA
RST 3V NC GND A0 A1 A2 A3 A4 A5 SCK MO MI RX TX 21
reset
CHG
huzzah!
ESP32!
ESP-WROOM
fritzing

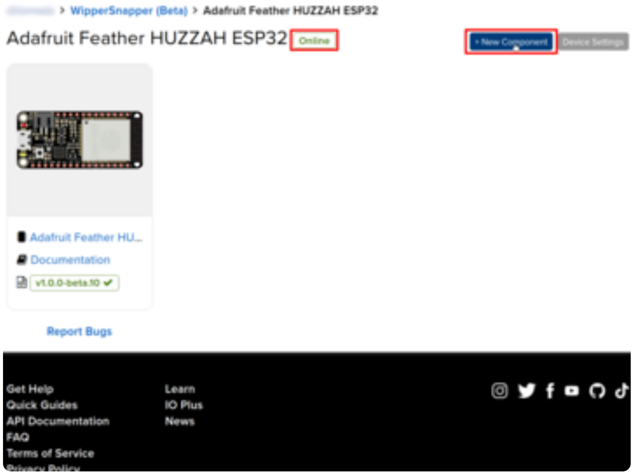
Connect the following wires:

- Feather **GND** to one side of the button
- Feather **Pin 14** to the other side of the button

# WipperSnapper Setup

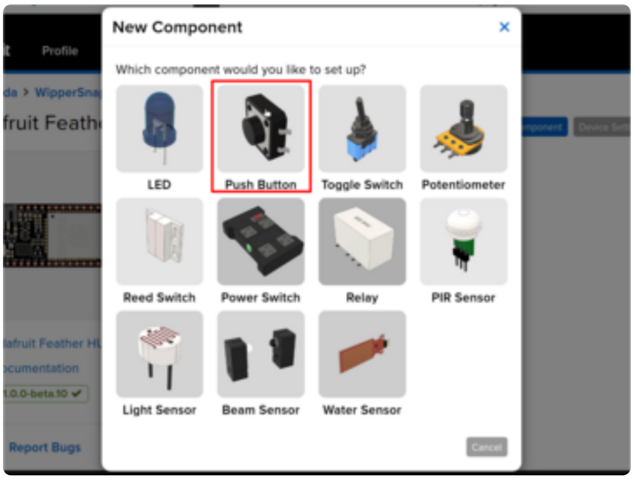
Now that you've wired everything up and added your board to WipperSnapper, it's time to set up the button itself.

Log into Adafruit.io (https://adafru.it/TAu).

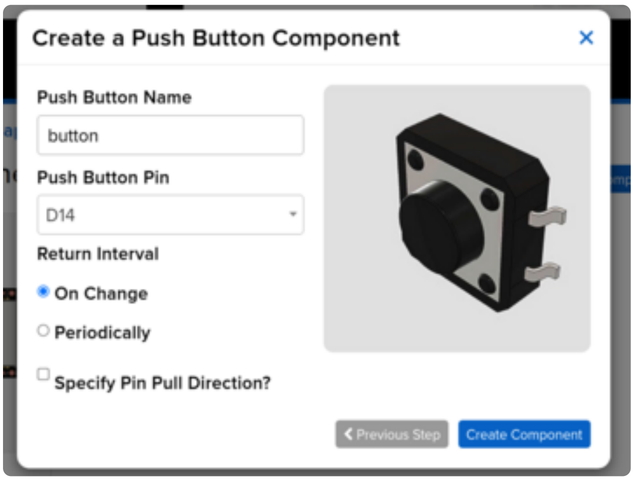


From the device page, click + **New Component**.

Make sure that your device says that it is online next to the name of the board. If it isn't but it was previously connected, try moving it closer to your router.



Then, click on **Push Button**.


Create a Push Button Component

Push Button Name: button

Push Button Pin: D14

Return Interval

● On Change

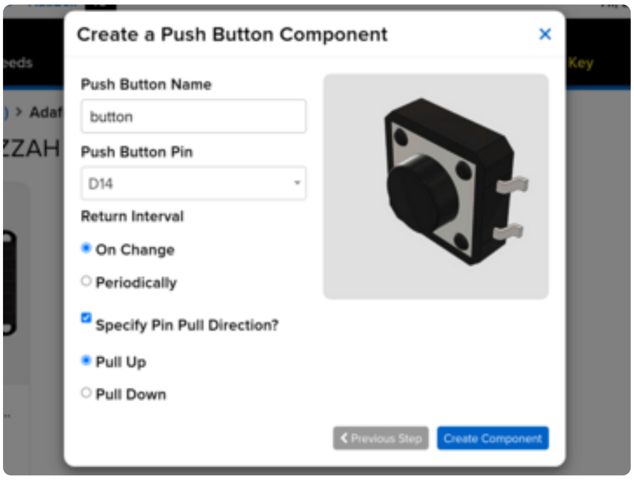
○ Periodically

□ Specify Pin Pull Direction?

Name it whatever you want. I chose to call it button, but anything else should work as well.

Set **Push Button Pin** to D14.

Finally, make sure that **Return Interval** is set to **On Change**.



Create a Push Button Component

Push Button Name: button

Push Button Pin: D14

Return Interval

● On Change

○ Periodically

☑ Specify Pin Pull Direction?

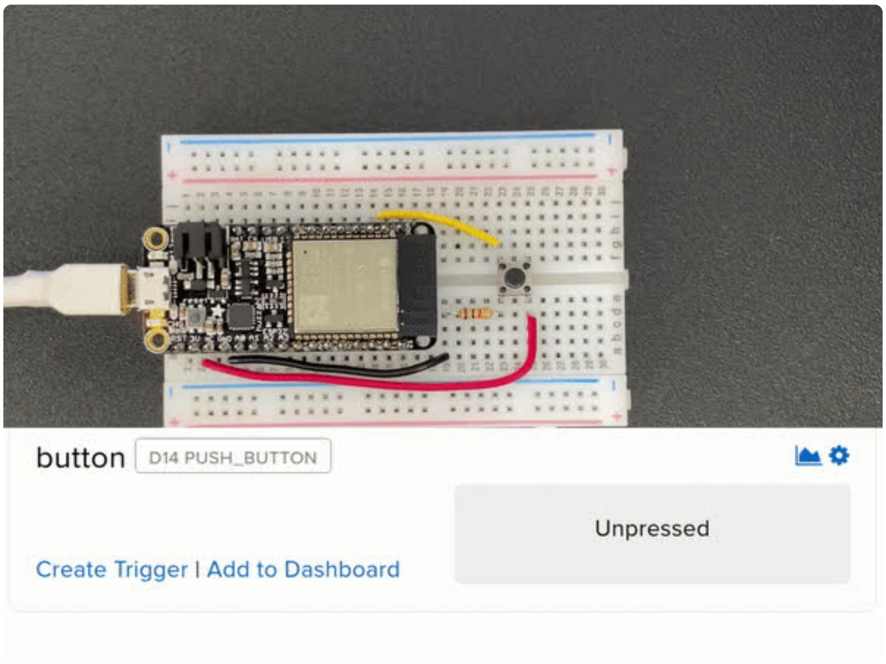
● Pull Up

○ Pull Down

If you are using an internal pull-up, check the box labeled **Specify Pin Pull Direction?** and select the box labeled **Pull Up**. If you are using a physical pull-down resistor, you don't need to do this.

## Project usage

Now, as you push the button, you should see the text in the grey sensor box go from "Unpressed" to "Pressed."



button D14 PUSH_BUTTON

Unpressed