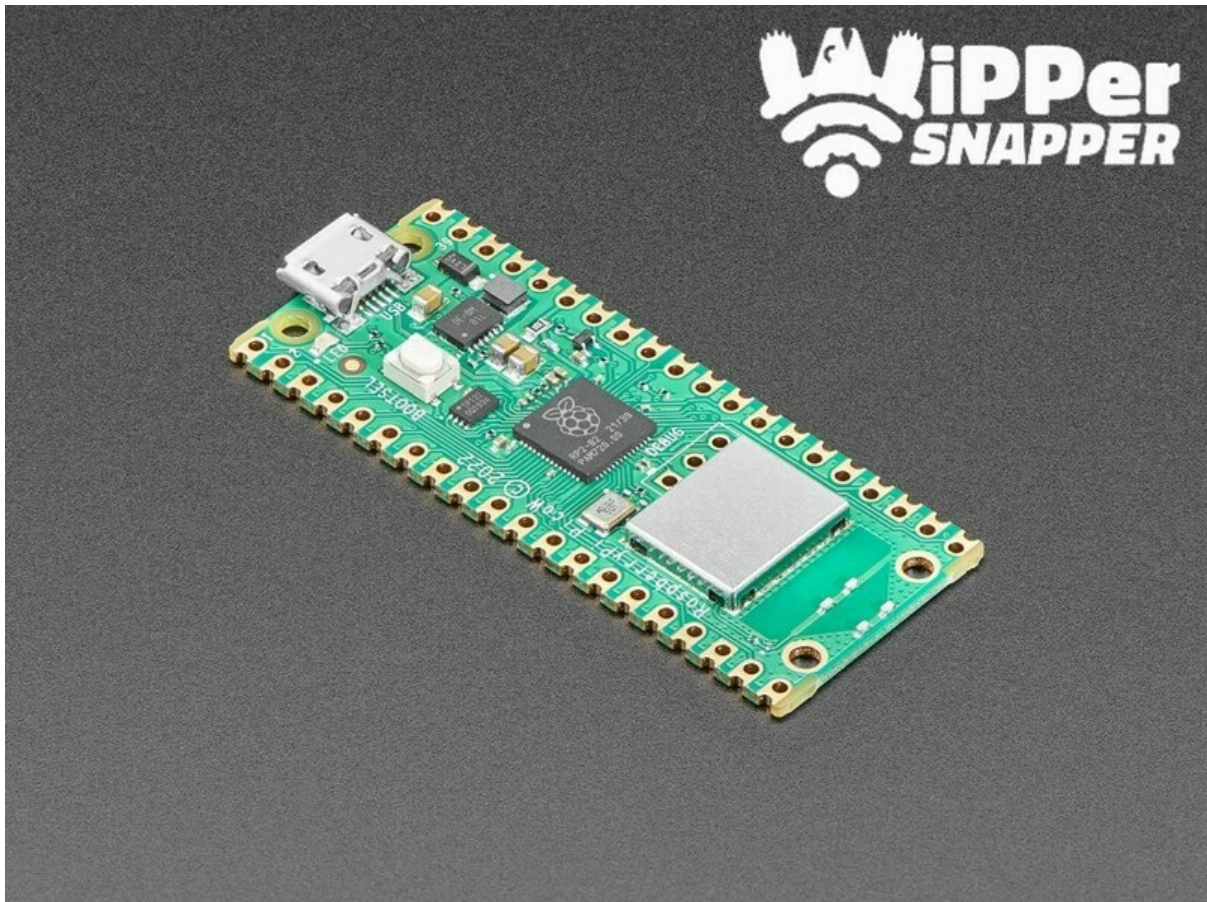




Quick Start: Pico W / 2W with WipperSnapper

Created by Brent Rubell



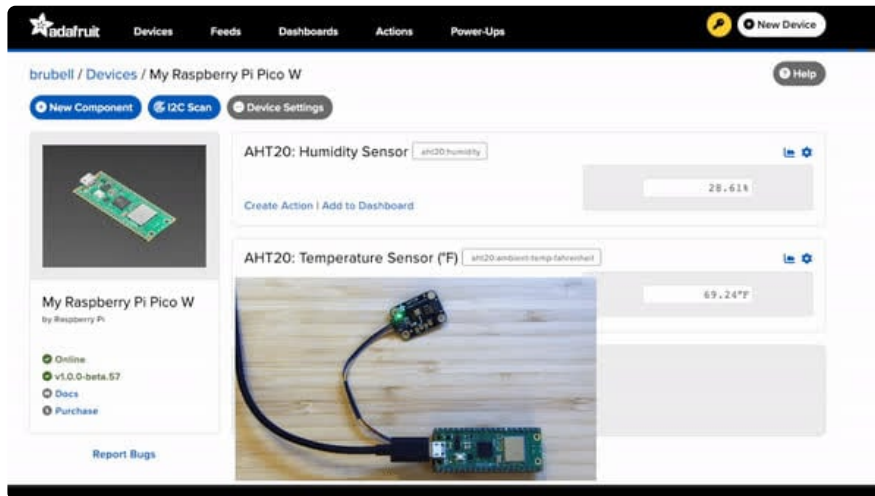
<https://learn.adafruit.com/quick-start-the-pico-w-with-wippersnapper>

Last updated on 2025-06-12 03:04:54 PM EDT

Table of Contents

Overview	3
<ul style="list-style-type: none">• What is WipperSnapper• Parts	
Install WipperSnapper	6
<ul style="list-style-type: none">• What is WipperSnapper• Sign up for Adafruit.io• Add a New Device to Adafruit IO• Feedback• Troubleshooting• "Uninstalling" WipperSnapper	
Usage	11
<ul style="list-style-type: none">• Blink a LED• Read a Push-Button• Read an I2C Sensor• Going Further	

Overview



The Raspberry Pi Foundation changed single-board computing when they released the Raspberry Pi computer. Then they did the same for microcontrollers with the release of the **Raspberry Pi Pico W**. This low-cost microcontroller board features their powerful new chip, the **RP2040**, and it has all the fixin's to get started with IoT embedded electronics projects at a stress-free price.

Now in 2024 they've released an upgraded version, pin compatible, called the **Pico 2 W**, using the brand new **RP2350** chip. It's a faster microcontroller containing more RAM and Flash storage space.

Installing WipperSnapper firmware on a Raspberry Pi Pico W / 2W allows you to create a Wi-Fi-enabled electronics project without any coding knowledge.

This guide covers installing WipperSnapper firmware on a Pico W, connecting the Pico W to our cloud platform (Adafruit IO), and configuring the Pico W to report button press data, toggle a LED, and send temperature measurements to the cloud. If you have a Pico 2W then usage with WipperSnapper is identical to the Pico W.

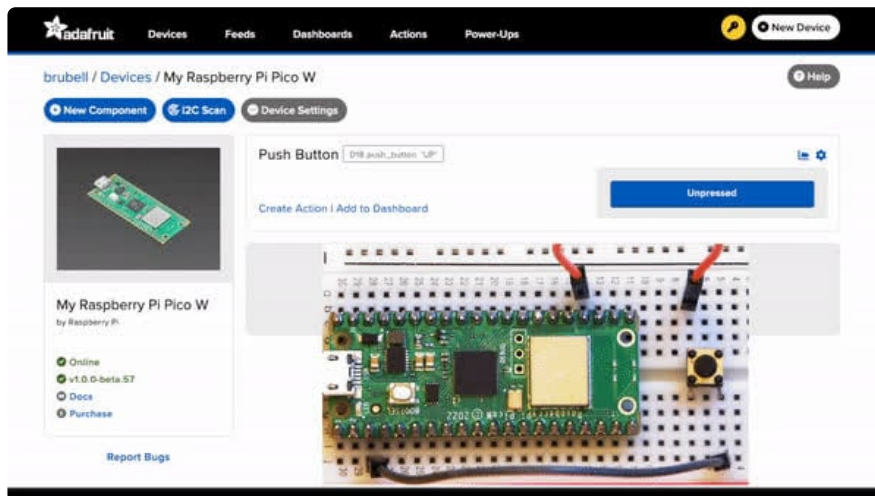
What is WipperSnapper

WipperSnapper is firmware designed to turn any WiFi-capable board into an Internet-of-Things device without programming a single line of code. WipperSnapper connects to [Adafruit IO \(https://adafru.it/fsU\)](https://adafru.it/fsU), a web platform designed (by [Adafruit! \(https://adafru.it/Bo5\)](https://adafru.it/Bo5)) to display, respond, and interact with your project's data.

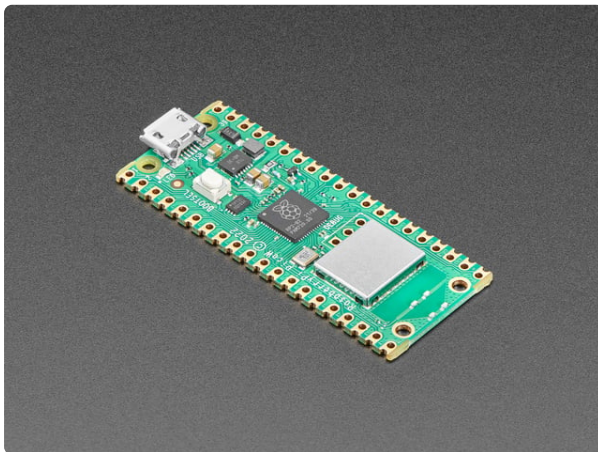
Simply load the WipperSnapper firmware onto your board, add credentials, and plug it into power. Your board will automatically register itself with your Adafruit IO account.

From there, you can add components to your board such as buttons, switches, potentiometers, sensors, and more! Components are dynamically added to hardware,

so you can immediately start interacting, logging, and streaming the data your projects produce without writing code.



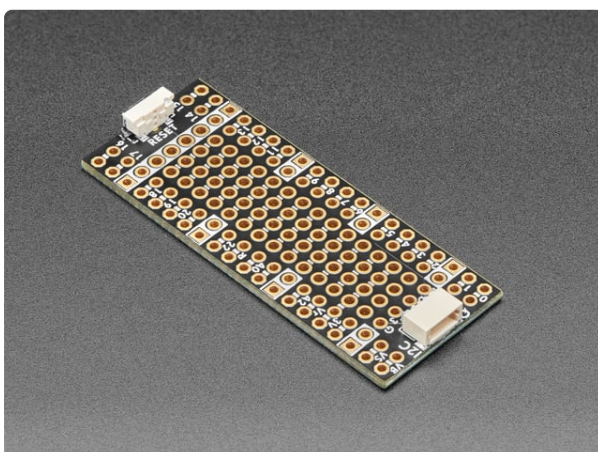
Parts



Raspberry Pi Pico W

The Raspberry Pi foundation changed single-board computing when they released the Raspberry Pi computer, now they're ready to...

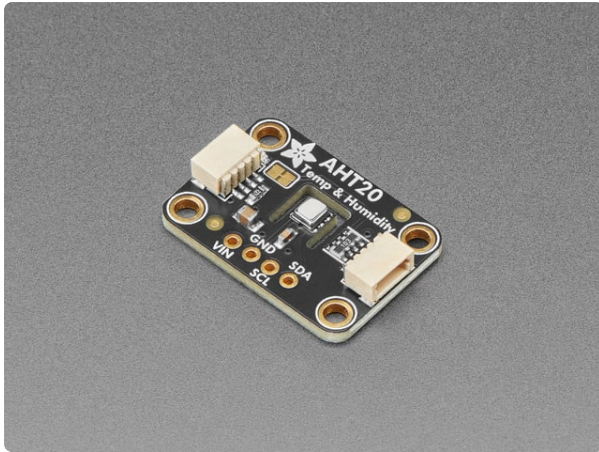
<https://www.adafruit.com/product/5526>



Adafruit PiCowbell Proto for Pico - Reset Button & STEMMA QT

Ding dong! Hear that? It's the PiCowbell ringing, letting you know that the new Adafruit PiCowbell Proto is finally in stock and ready to assist your

<https://www.adafruit.com/product/5200>



Adafruit AHT20 - Temperature & Humidity Sensor Breakout Board

The AHT20 is a nice but inexpensive temperature and humidity sensor from the same folks that brought us the DHT22. You can take...

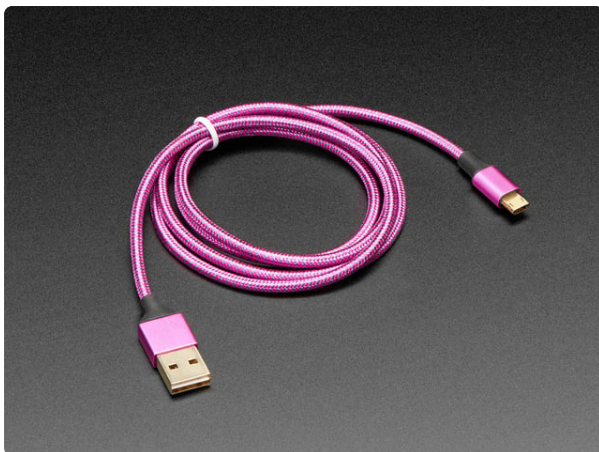
<https://www.adafruit.com/product/4566>



STEMMA QT / Qwiic JST SH 4-pin Cable - 100mm Long

This 4-wire cable is a little over 100mm / 4" long and fitted with JST-SH female 4-pin connectors on both ends. Compared with the chunkier JST-PH these are 1mm pitch instead of...

<https://www.adafruit.com/product/4210>



Fully Reversible Pink/Purple USB A to micro B Cable - 1m long

This cable is not only super-fashionable, with a woven pink and purple Blinka-like pattern, it's also fully reversible! That's right, you will save seconds a day by...

<https://www.adafruit.com/product/4111>



Adafruit IO+ Subscription Pass – One Year

The all-in-one Internet of Things service from Adafruit you know and love is now even better with IO+. The 'plus' stands for MORE STUFF! More feeds, dashboards,...

<https://www.adafruit.com/product/3792>



Adafuit IO+ 1 Year Subscription Card

It's the Internet of the Things! Adafuit IO+ is the easiest way to stream, log, and interact with your data. Whether you're...

<https://www.adafruit.com/product/3980>

Install WipperSnapper

The WipperSnapper firmware and ecosystem are in BETA and are actively being developed to add functionality, more boards, more sensors, and fix bugs. We encourage you to try out WipperSnapper with the understanding that it is not final release software and is still in development. If you encounter any bugs, glitches, or difficulties during the beta period, or with this guide, please contact us via <http://io.adafruit.com/support>

What is WipperSnapper

WipperSnapper is a firmware designed to turn any WiFi-capable board into an Internet-of-Things device without programming a single line of code. WipperSnapper connects to [Adafuit IO \(https://adafru.it/fsU\)](https://adafru.it/fsU), a web platform designed ([by Adafuit! \(https://adafru.it/Bo5\)](https://adafru.it/Bo5)) to display, respond, and interact with your project's data.

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

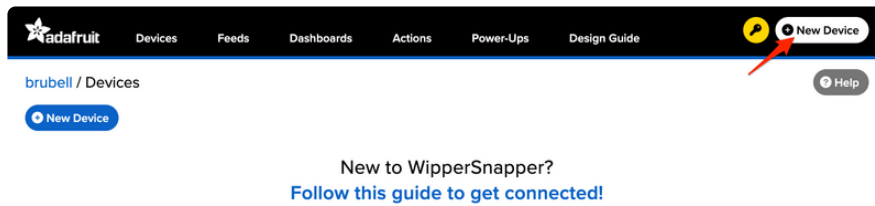
From there, you can add components to your board such as buttons, switches, potentiometers, sensors, and more! Components are dynamically added to hardware, so you can immediately start interacting, logging, and streaming the data your projects produce without writing code.

Sign up for Adafuit.io

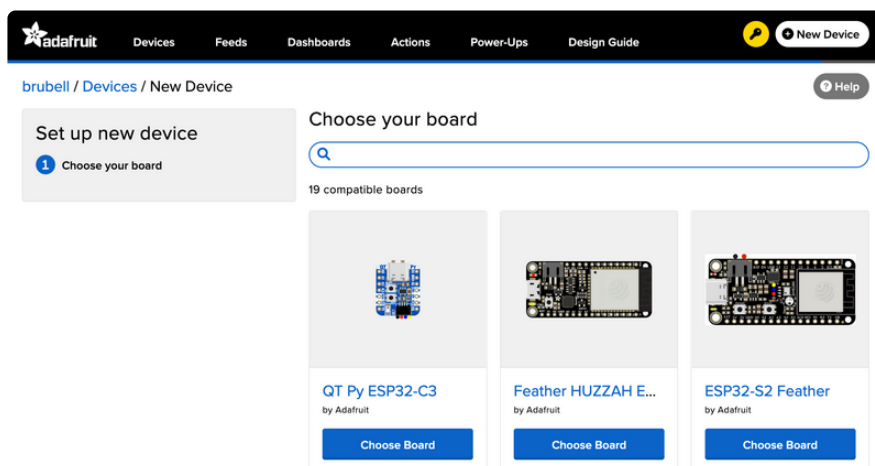
You will need an Adafuit IO account to use WipperSnapper on your board. If you do not already have one, head over to [io.adafruit.com \(https://adafru.it/fsU\)](https://adafru.it/fsU) to create a free account.

Add a New Device to Adafruit IO

Log into your [Adafruit IO \(https://adafru.it/fsU\)](https://adafru.it/fsU) account. Click the New Device button at the top of the page.

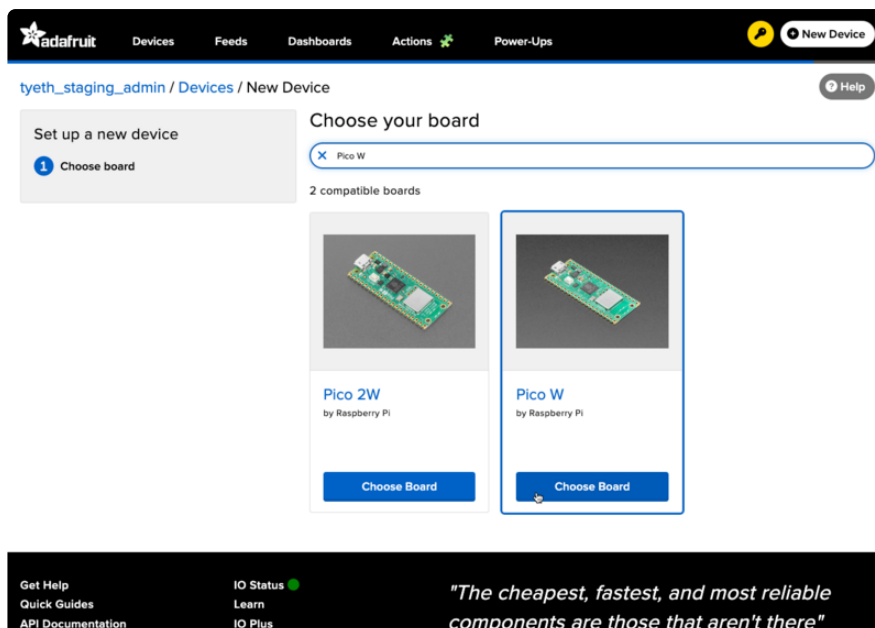


After clicking New Device, you should be on the board selector page. This page displays every board that is compatible with the WipperSnapper firmware.

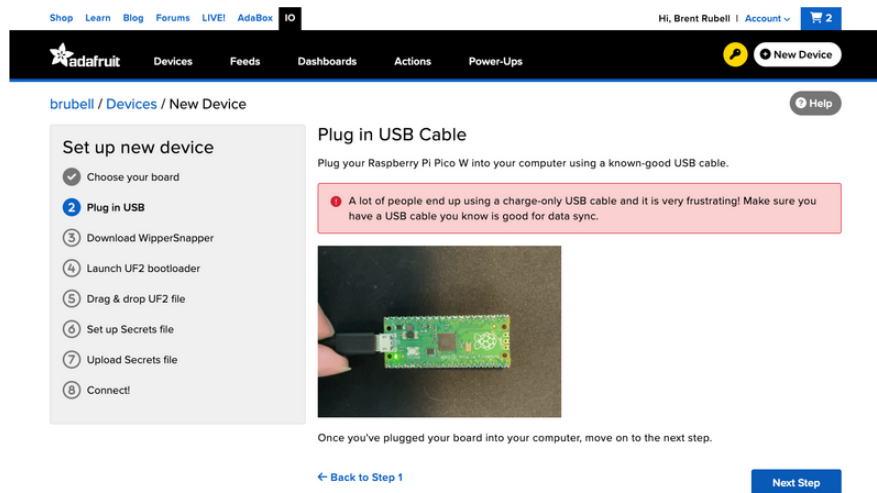


In the board selector page's search bar, search for Pico W. Pick either the Pico W or Pico 2W as needed.

Click the Choose Board button to bring you to the self-guided installation wizard.

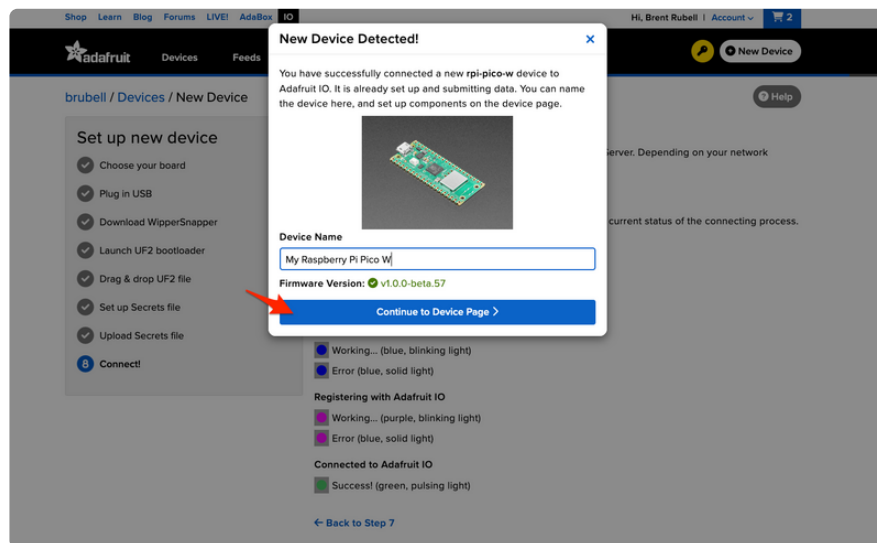


Follow the step-by-step instructions on the page to install Wippersnapper on your device and connect it to Adafruit IO.

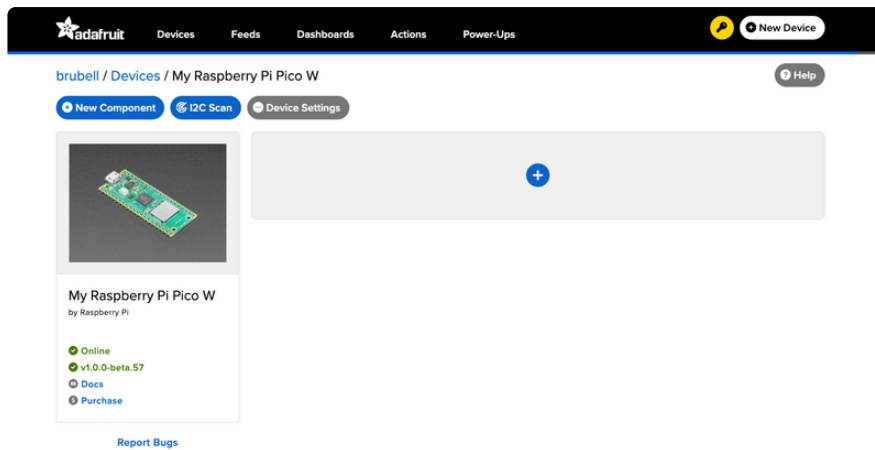


If the installation was successful, a popover should appear displaying that your board has successfully been detected by Adafruit IO.

Give your board a name and click "Continue to Device Page".



You should be brought to your board's device page.



Feedback

Adafruit.io WipperSnapper is in **beta** and you can help improve it!

If you have suggestions or general feedback about the installation process - visit <https://io.adafruit.com/support> (<https://adafru.it/Sgb>), click "Contact Adafruit IO Support" and select "I have feedback or suggestions for the WipperSnapper Beta".

Troubleshooting

If you encountered an issue during installation, please try the steps below first.

If you're still unable to resolve the issue, or if your issue is not listed below, get in touch with us directly at <https://io.adafruit.com/support> (<https://adafru.it/Sgb>). Make sure to click "Contact Adafruit IO Support" and select "There is an issue with WipperSnapper. Something is broken!"

? I don't see my board on Adafruit IO, it is stuck connecting to WiFi

First, make sure that you selected the correct board on the board selector.

Next, please make sure that you entered your WiFi credentials properly, there are no spaces/special characters in either your network name (SSID) or password, and that you are connected to a 2.4GHz wireless network.

If you're still unable to connect your board to WiFi, please [make a new post on the WipperSnapper technical support forum with the error you're](#)

experiencing, the LED colors which are blinking, and the board you're using. (<https://adafru.it/V6a>)



I don't see my board on Adafruit IO, it is stuck "Registering with Adafruit IO"

Try hard-resetting your board by unplugging it from USB power and plugging it back in.

If the error is still occurring, please [make a new post on the WipperSnapper technical support forum with information about what you're experiencing, the LED colors which are blinking \(if applicable\), and the board you're using. \(<https://adafru.it/V6a>\)](#)

"Uninstalling" WipperSnapper

WipperSnapper firmware is an application that is loaded onto your board. There is nothing to "uninstall". However, you may want to "move" your board from running WipperSnapper to running Arduino or CircuitPython. You also may need to restore your board to the state it was shipped to you from the Adafruit factory.

Moving from WipperSnapper to CircuitPython

Follow the steps on the [Installing CircuitPython page \(<https://adafru.it/Amd>\)](https://adafru.it/Amd) to install CircuitPython on your board running WipperSnapper.

- If you are unable to double-tap the RST button to enter the UF2 bootloader, follow the "Factory Resetting a WipperSnapper Board" instructions below.

Uploading this sketch will overwrite WipperSnapper. If you want to re-install WipperSnapper, follow the instructions at the top of this page.

Moving from WipperSnapper to Arduino

If you want to use your board with Arduino, you will use the Arduino IDE to load any sketch onto your board.

First, follow the page below to set up your Arduino IDE environment for use with your board.

Setup Arduino IDE for Pico W

<https://adafru.it/UPF>

Connect your Pico W to Arduino IDE

<https://adafru.it/18IE>

Then, follow the page below to upload the "Arduino Blink" sketch to your board.

Upload a sketch to Pico W from Arduino IDE

<https://adafru.it/18ma>

Uploading this sketch will overwrite WipperSnapper. If you want to re-install WipperSnapper, follow the instructions at the top of this page.

Factory Resetting a WipperSnapper Board

Sometimes, hardware gets into a state that requires it to be "restored" to the original state it shipped in. If you'd like to get your board back to its original factory state, follow the guide below.

Resetting RP2040 Flash Memory

<https://adafru.it/18mc>

Usage

This page assumes that you have installed WipperSnapper on your board and registered it with the Adafruit.io website. If you have not done this yet, please go back to the previous page in this guide and connect your board.

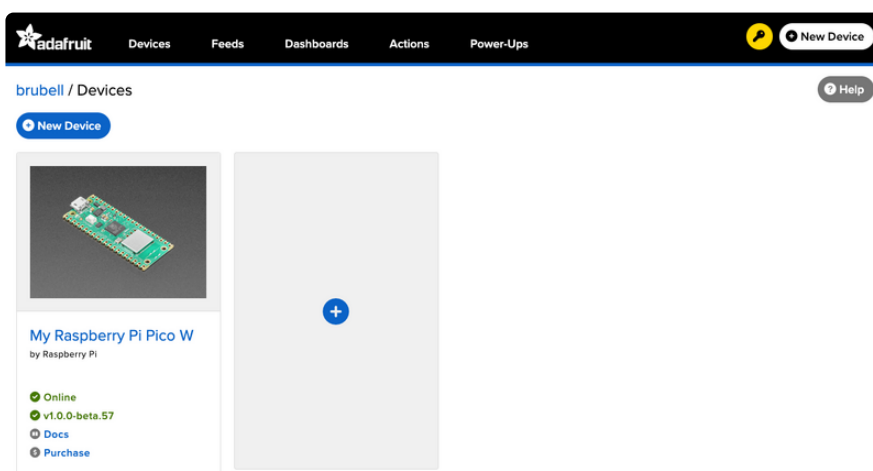
Blink a LED

One of the first programs you typically write to get used to embedded programming is a sketch that repeatably blinks an LED. IoT projects are wireless so after completing this section, you'll be able to turn on (or off) the LED built into your board from anywhere in the world.

In this demo, we show controlling an LED from Adafruit IO. But the same kind of control can be used for relays, lights, motors, or solenoids.

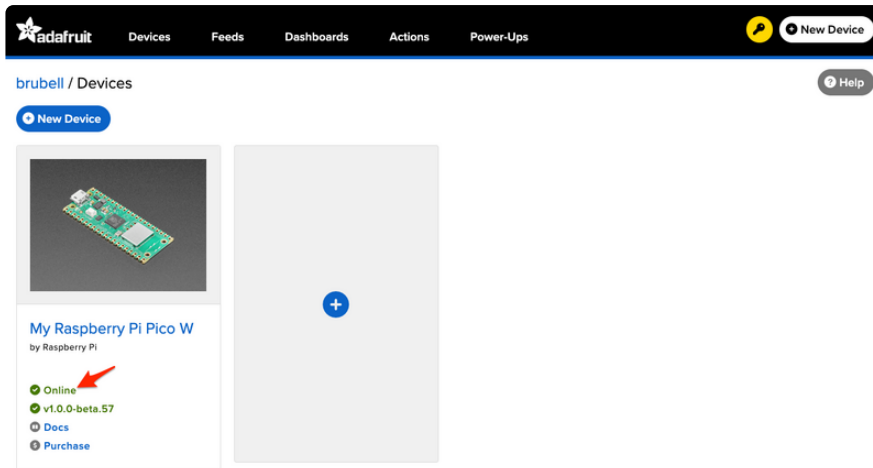
Navigate to io.adafruit.com/wippersnapper (<https://adafru.it/TAu>). You should see the board you just connected to Adafruit IO listed on this page.

- If you do not see your board - go back to the previous setup page and ensure you have registered it with Adafruit IO

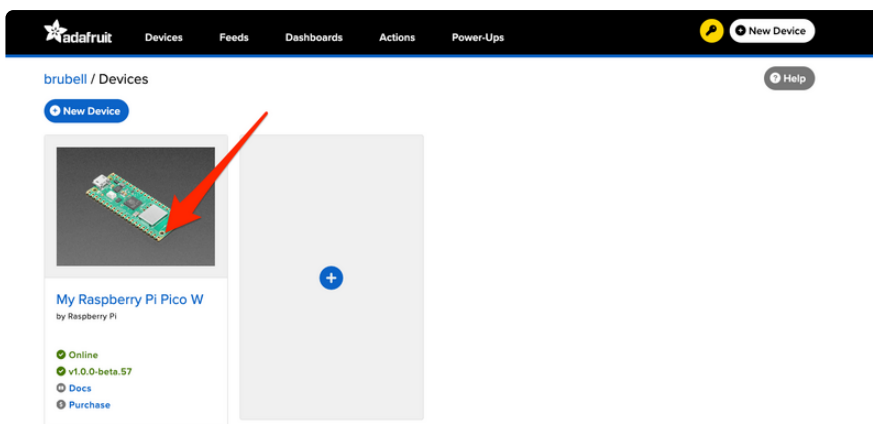


Make sure the board's tile says Online in green text, indicating that it's online and communicating with Adafruit IO.

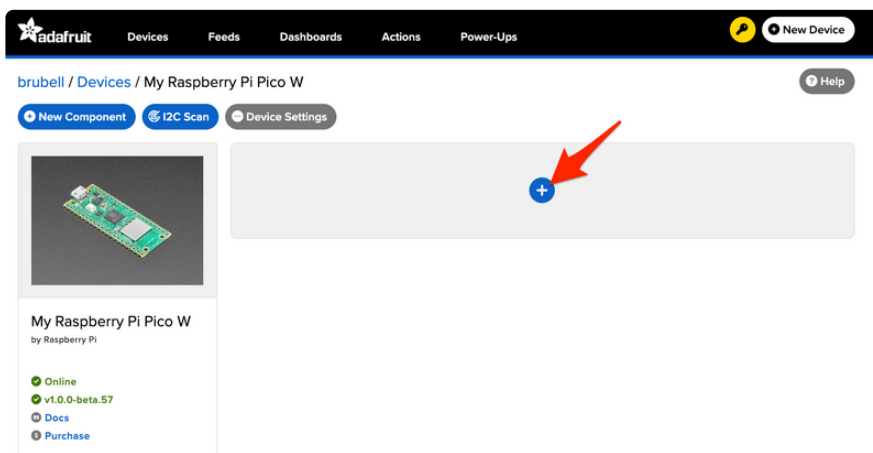
- If the board appears offline on the website but was previously connected, press the Reset (RST) button (or unplug the USB, and plug it back in) to force the board to reboot.



Click the board to navigate to its page.

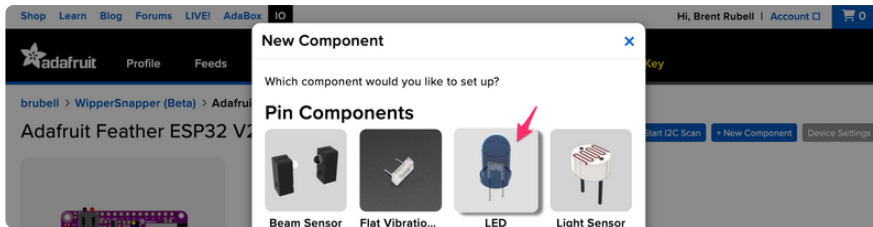


From the device page, click the + New Component (or +) button.



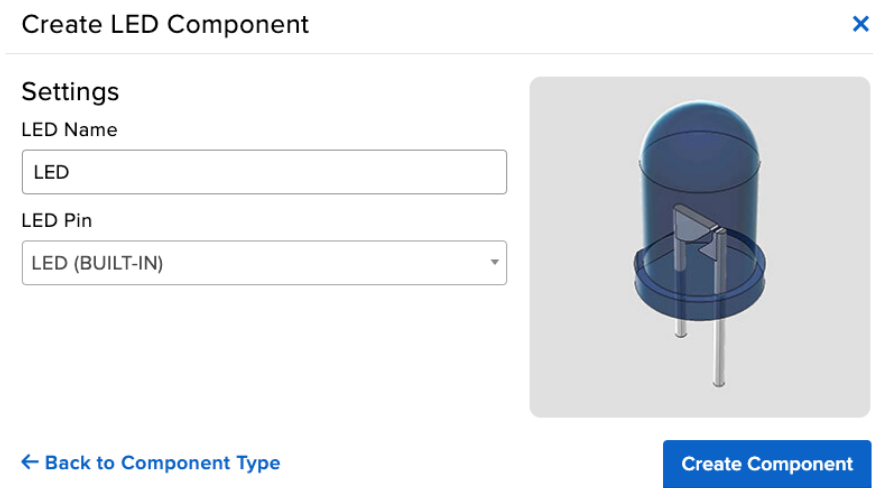
The Component Picker lists all the sensors and actuators which can be used with the WipperSnapper firmware.

Click the LED icon.



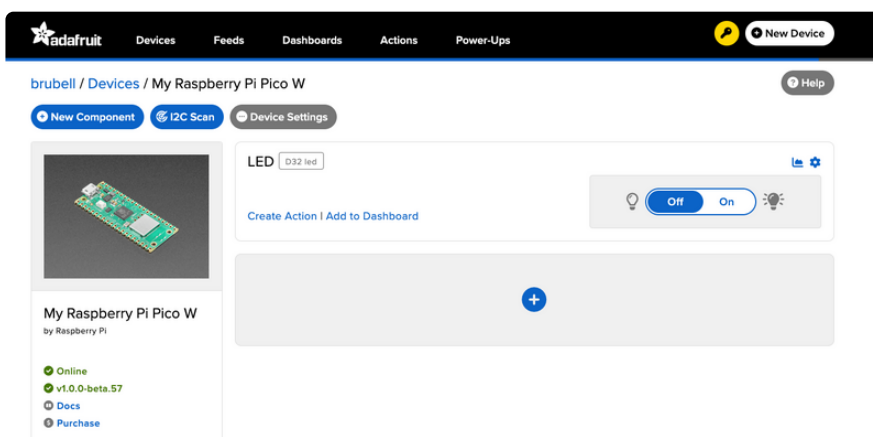
Microcontroller boards contain GPIO pins that can be configured either as an input or an output. The "Create LED Component" screen tells WipperSnapper to configure a general-purpose output pin connected to the LED on your board as a digital output so you can turn the LED on or off.

The Raspberry Pi Pico W has a built-in LED. Select this pin as the LED Pin and click **Create Component**.

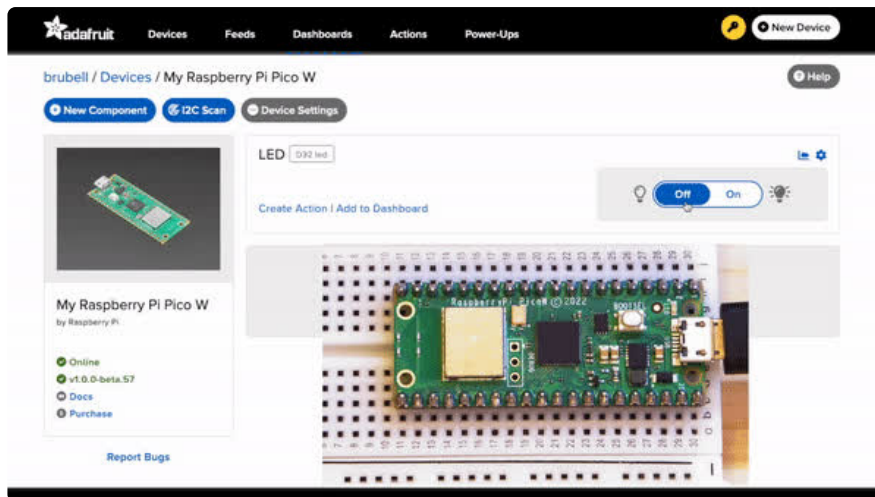


Behind the scenes, Adafruit IO sends a command to your board running WipperSnapper telling it to configure the GPIO pin as a digital output.

Your board's page on Adafruit IO shows a new LED component.



On the board page, **toggle the LED component by clicking the toggle switch**. This should turn your board's built-in LED on or off.



Read a Push-Button

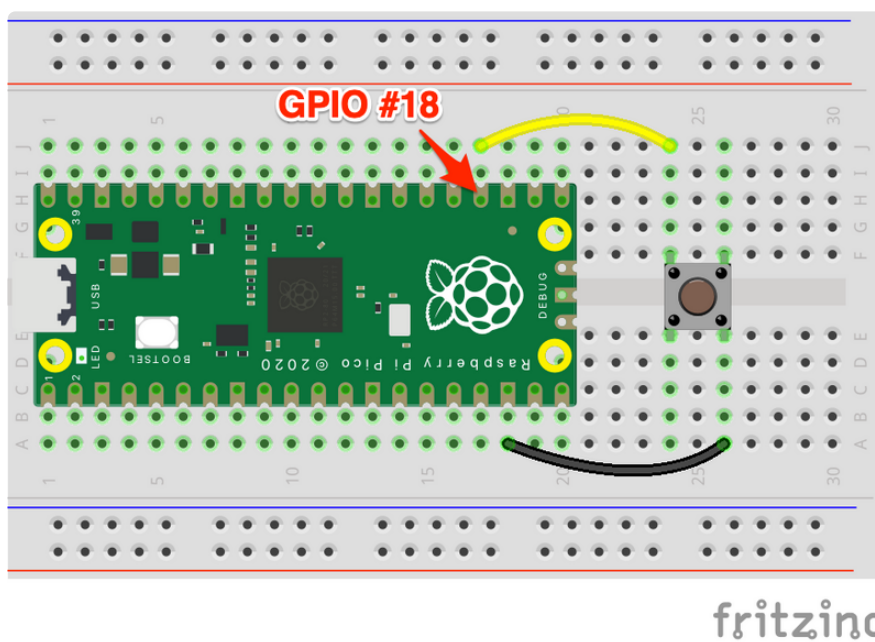
You can also configure a board running WipperSnapper to read data from standard input buttons, switches, or digital sensors, and send the value to Adafruit IO.

Let's wire up a push button to your board and configure it to publish a value to Adafruit IO when the button has been pressed or released.

In this demo, we show reading the state of a push-button from WipperSnapper. But the same kind of control can be used for reading switches, break beam sensors, and other digital sensors.

Wiring

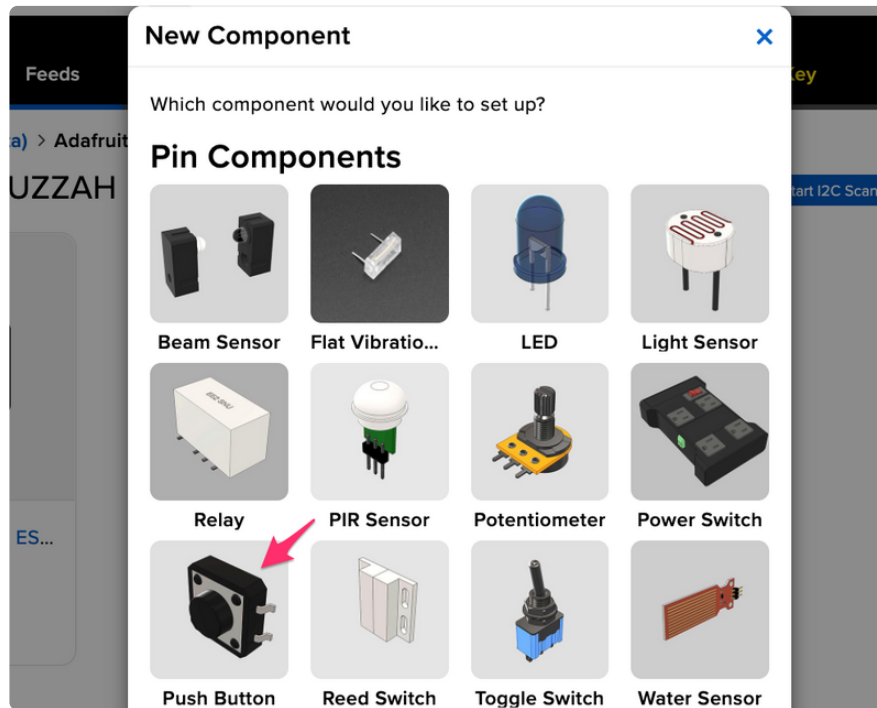
We'll be using the board's internal pull-up resistors instead of a physical resistor.



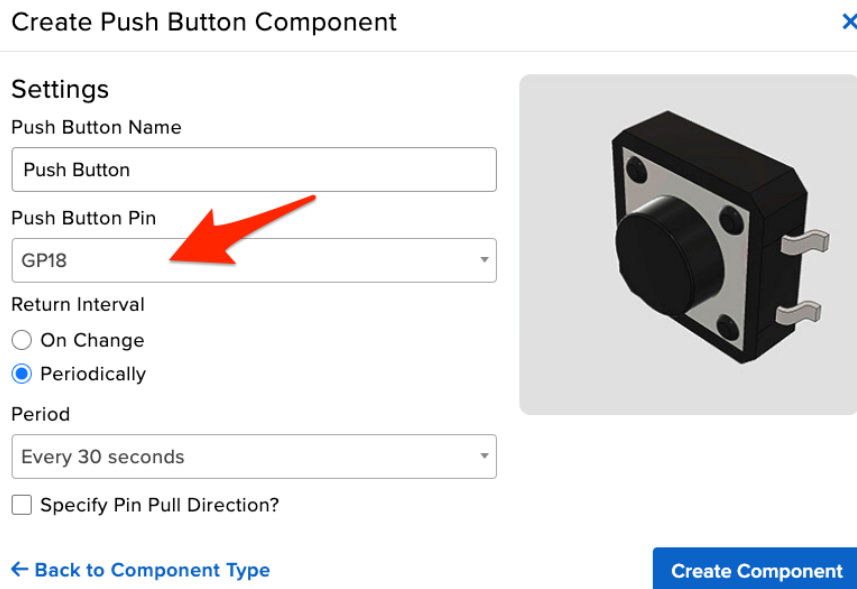
Usage

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

From the component picker, select the **Push Button**.



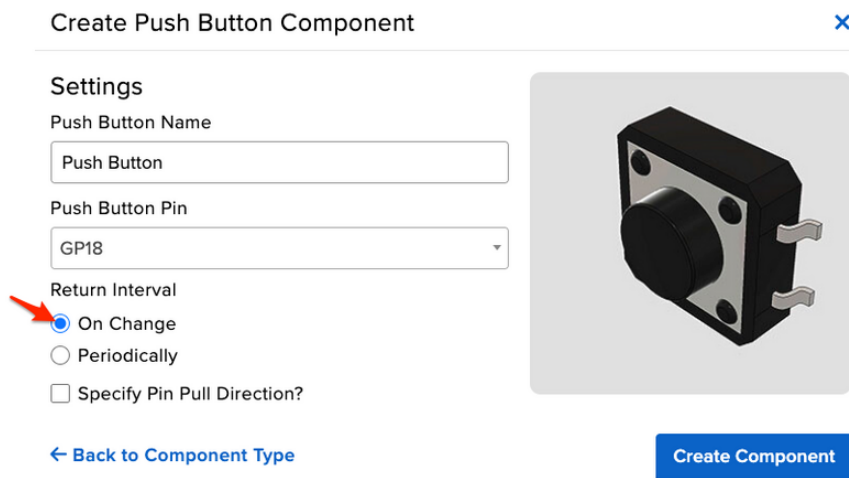
The next screen presents you with options for configuring the push button. Start by selecting the board's pin you connected to the push button.



The Return Interval dictates how frequently the value of the push-button will be sent from the board to Adafruit IO.

For this example, you will configure the push button's value to be only sent when the value changes (i.e: when it's either pressed or depressed).

Select On Change

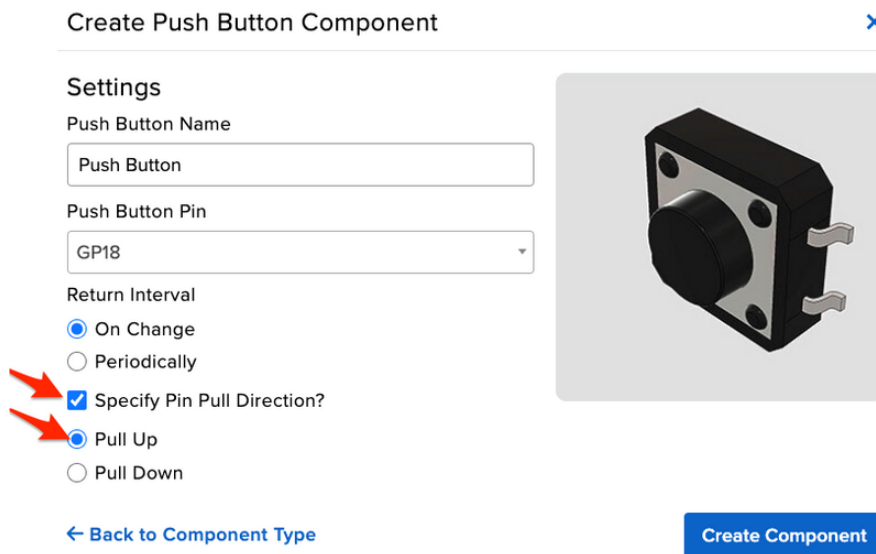


The screenshot shows the 'Create Push Button Component' form. The 'Settings' section includes:

- Push Button Name: Push Button
- Push Button Pin: GP18
- Return Interval: On Change, Periodically
- Specify Pin Pull Direction?

A red arrow points to the 'On Change' radio button. At the bottom, there is a blue 'Create Component' button and a blue link 'Back to Component Type'.

Finally, check the **Specify Pin Pull Direction** checkbox and select **Pull UP** to turn on the pin's internal pull-up resistor.



The screenshot shows the 'Create Push Button Component' form with the following settings:

- Push Button Name: Push Button
- Push Button Pin: GP18
- Return Interval: On Change, Periodically
- Specify Pin Pull Direction?
 - Pull Up
 - Pull Down

Two red arrows point to the 'Specify Pin Pull Direction?' checkbox and the 'Pull Up' radio button. At the bottom, there is a blue 'Create Component' button and a blue link 'Back to Component Type'.

Make sure the form's settings look like the following screenshot. Then, click **Create Component**.

Create Push Button Component

Settings

Push Button Name

Push Button Pin

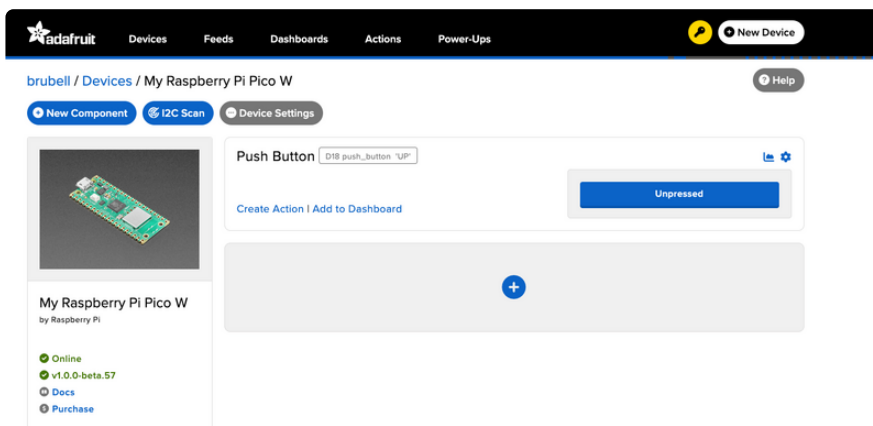
Return Interval

 On Change Periodically Specify Pin Pull Direction? Pull Up Pull Down

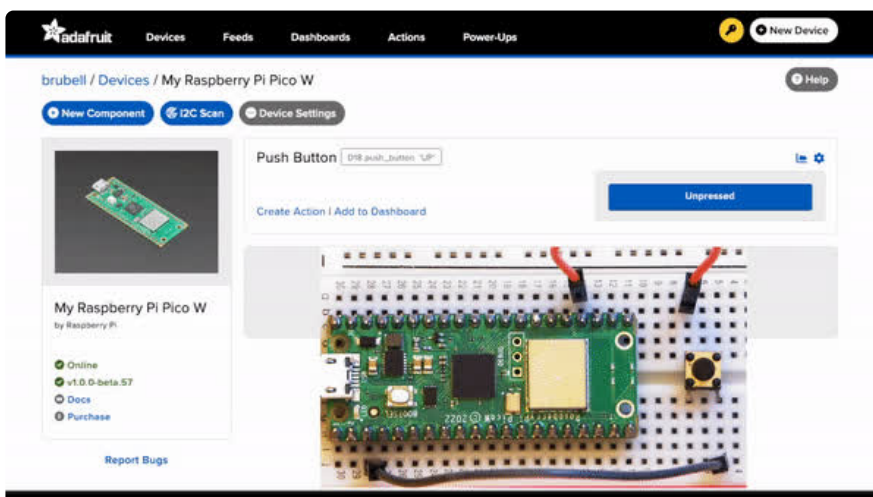
[← Back to Component Type](#)

[Create Component](#)

Adafruit IO sends a command to your WipperSnapper board, telling it to configure the GPIO pin you selected to behave as a digital input pin and to enable it to pull up the internal resistor. Your board's page should also show the new push-button component.



Push the button to change the value of the component on Adafruit IO.



Read an I2C Sensor

Inter-Integrate Circuit, aka **I2C**, is a two-wire protocol for connecting sensors and "devices" to a microcontroller. A large number of sensors, including the ones sold by Adafruit, use I2C to communicate.

Typically, using I2C with a microcontroller involves programming. Adafruit IO and WipperSnapper let you configure a microcontroller to read data from an I2C sensor and publish that data to the internet without writing code.

The WipperSnapper firmware supports a number of I2C sensors, [viewable in list format here \(https://adafru.it/Zbq\)](https://adafru.it/Zbq). If you do not see the I2C sensor you're attempting to use with WipperSnapper - [we have a guide on adding a component to Adafruit IO WipperSnapper here \(https://adafru.it/Zbr\)](https://adafru.it/Zbr).

The process for adding an I2C component to your board running WipperSnapper is similar for most sensors. For this section, we're using the [Adafruit AHT20 \(http://adafru.it/4566\)](http://adafru.it/4566), an inexpensive sensor that can read ambient temperature and humidity.

Wiring

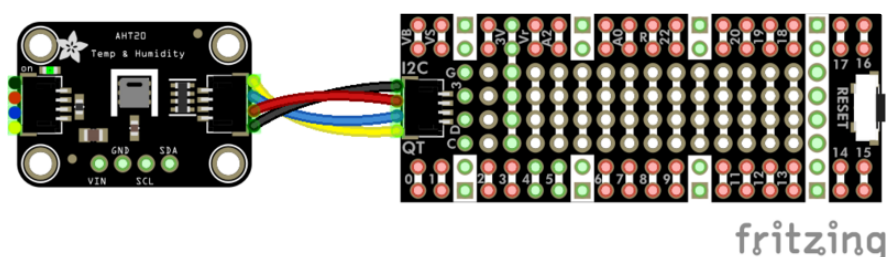
First, wire up an AHT20 sensor to your board exactly as follows. Here is an example of the AHT20 wired using I2C [with a STEMMA QT cable \(no soldering required\) \(http://adafru.it/4209\)](http://adafru.it/4209).

While the Raspberry Pi Pico is able to define any two pins as an I2C bus, only an I2C bus using pins GPIO 4 and GPIO 5 is supported while using WipperSnapper.

To simplify wiring, we highly suggest using an [Adafruit PiCowbell Proto for Pico \(http://adafru.it/5200\)](http://adafru.it/5200) since it includes a STEMMA QT/Qwiic connector for I2C.

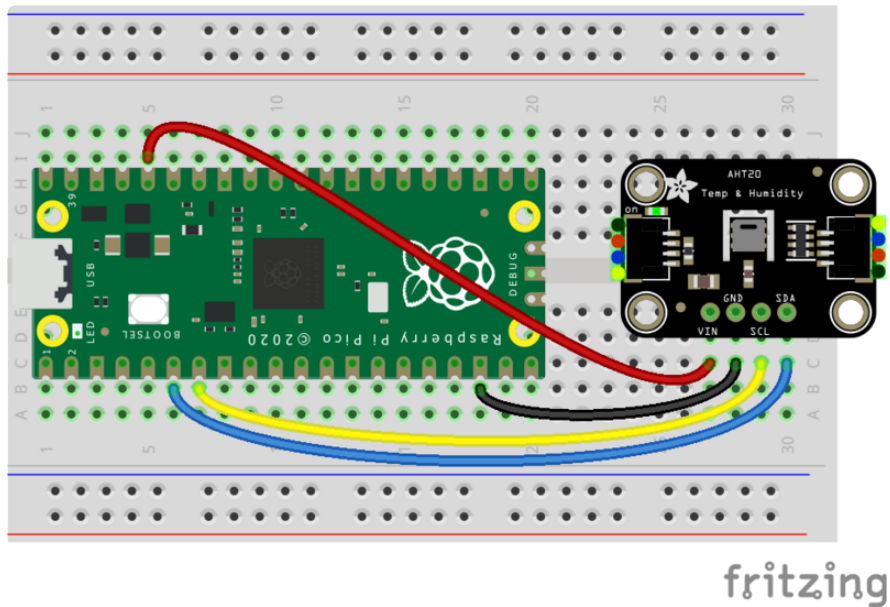
- Before wiring everything up, follow [the assembly page here \(https://adafru.it/18mA\)](https://adafru.it/18mA) to connect the PiCowbell to the Raspberry Pi Pico W.

Connect a "STEMMA QT to STEMMA QT" cable from the STEMMA QT port on the PiCowbell to a STEMMA QT port on your sensor.



If you do **not** have a PiCowBell, make the following connections between the Pico W and the AHT20:

- Pico 3.3V (Physical Pin 36) to AHT20 VIN
- Pico Ground to AHT20 GND
- Pico SDA (GPIO 4 - Physical Pin 6) to AHT20 SDA
- Pico SCL (GPIO 5 - Physical Pin 7) to AHT20 SCL

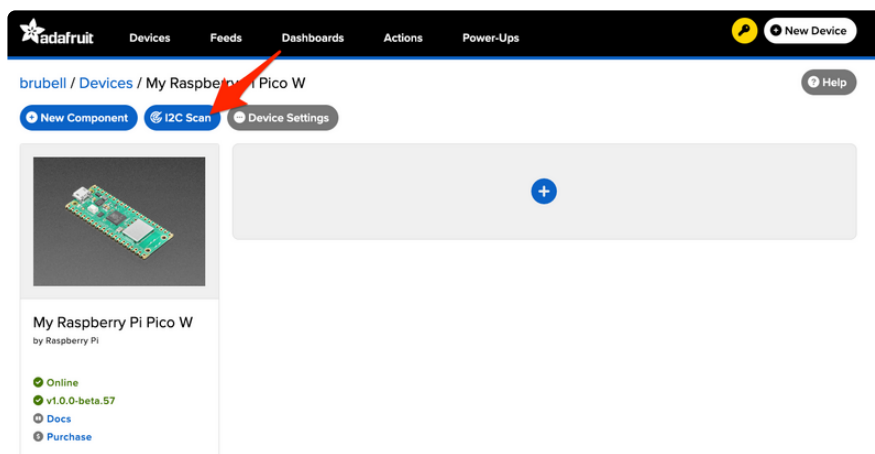


Scan I2C Bus

First, ensure that you've correctly wired the AHT20 sensor to your board by performing an I2C scan to detect the I2C device on the bus.

On the board page, click **Start I2C Scan**.

- If you do not see this button, double-check that your board shows as Online.



You should see a new pop-up showing a list of the I2C addresses detected by an I2C scan. If wired correctly, the AHT20's default I2C address of **0x38** appear in the I2C scan list.

I2C Scan Complete ✕

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00								--	--	--	--	--	--	--	--	--
10	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
20	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
30	--	--	--	--	--	--	--	38	--	--	--	--	--	--	--	--
40	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
50	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
60	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
70	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Close Scan Again

? I don't see the I2C sensor's address in the list

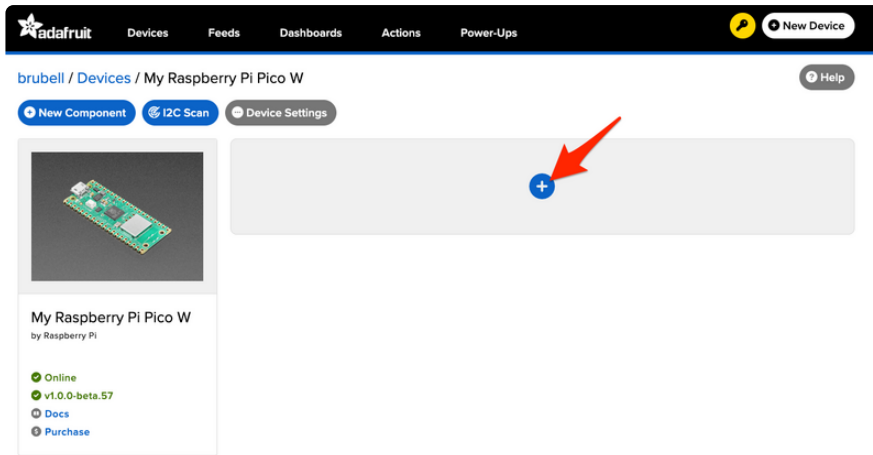
First, double-check the connection and/or wiring between the sensor and the board.

Then, reset the board and let it re-connect to Adafruit IO WipperSnapper.

Create the Sensor Component

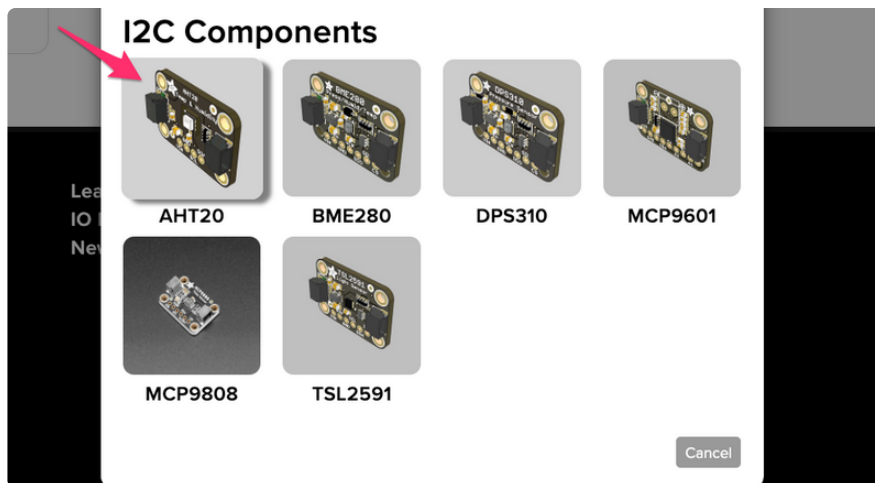
Now that you know the sensor can be detected by the board, it's time to configure and create the sensor on Adafruit IO.

On the board page, add a new component to your board by clicking the + **button** or the + **New Component** button.



The Component Picker lists all the available components, sensors, and parts that can be used with WipperSnapper.

Under the **I2C Components** header, click **AHT20**.



On the component configuration page, the AHT20's I2C sensor address should be listed along with the sensor's settings.

The AHT20 sensor can measure ambient temperature and relative humidity. This page has individual options for reading the ambient temperature, in either degree Celsius or degree Fahrenheit, and the relative humidity. You may select the readings which are appropriate to your application and region.

The **Send Every** option is specific to each sensor measurement. This option will tell the Feather how often it should read from the AHT20 sensor and send the data to Adafruit IO. Measurements can range from every 30 seconds to every 24 hours.

For this example, set the **Send Every** interval for both seconds to **Every 30 seconds** and click **Create Component**.

Create AHT20 Component



Select I2C Address:

0x38

Enable AHT20: Temperature Sensor (°C)?

Enable AHT20: Temperature Sensor (°F)?

Name:

AHT20: Temperature Sensor (°F)

Send Every:

Every 30 seconds

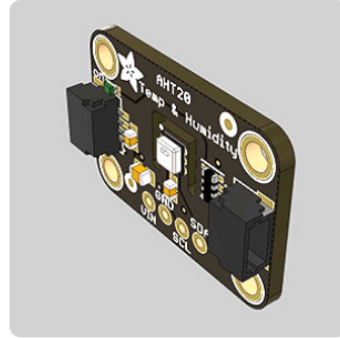
Enable AHT20: Humidity Sensor?

Name:

AHT20: Humidity Sensor

Send Every:

Every 30 seconds

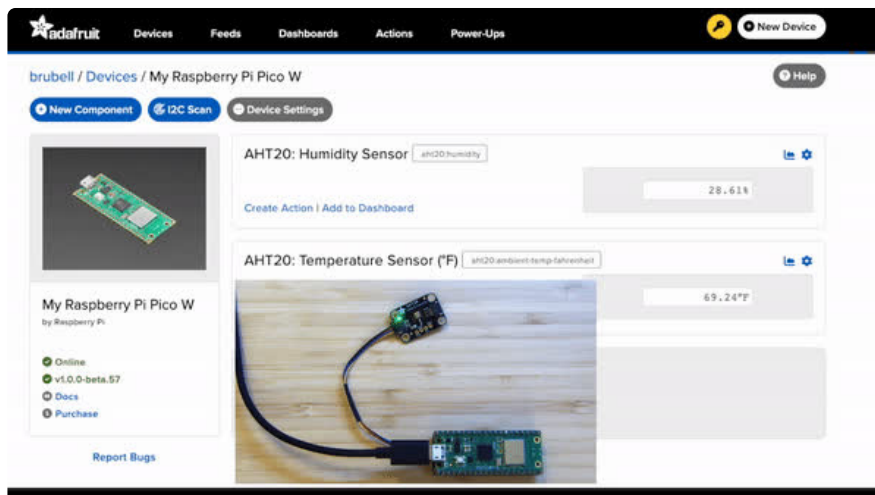


[← Back to Component Type](#)

Create Component

The board page should now show the AHT20 component you created.

After the interval you configured elapses, WipperSnapper automatically reads values from the sensor and sends them to Adafruit IO.



Going Further

Want to learn more about Adafruit IO WipperSnapper? We have [more complex projects on the Adafruit Learning System \(https://adafru.it/Yyd\)](https://adafru.it/Yyd).