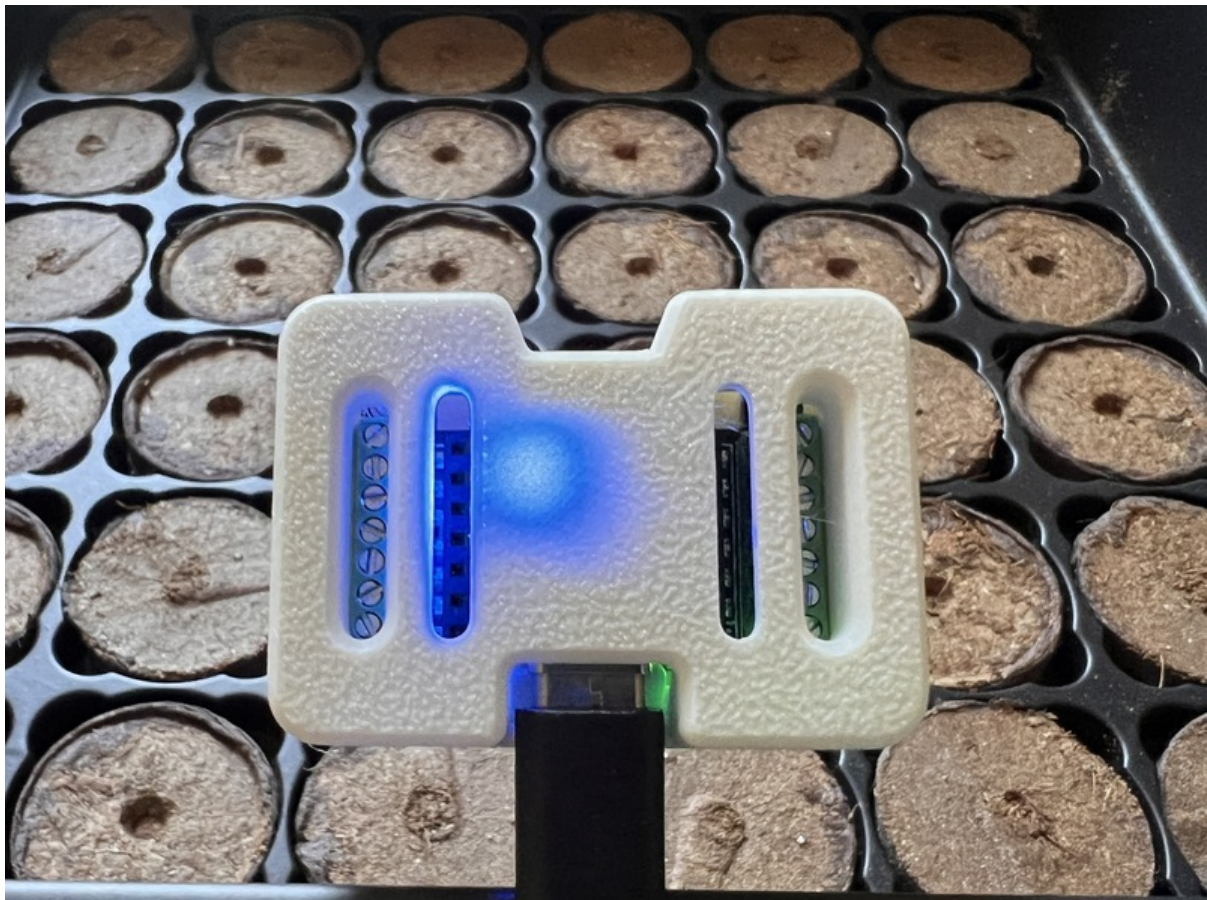




No-Code Seedling Frost Alert Device with Adafruit IO and WipperSnapper

Created by Brent Rubell



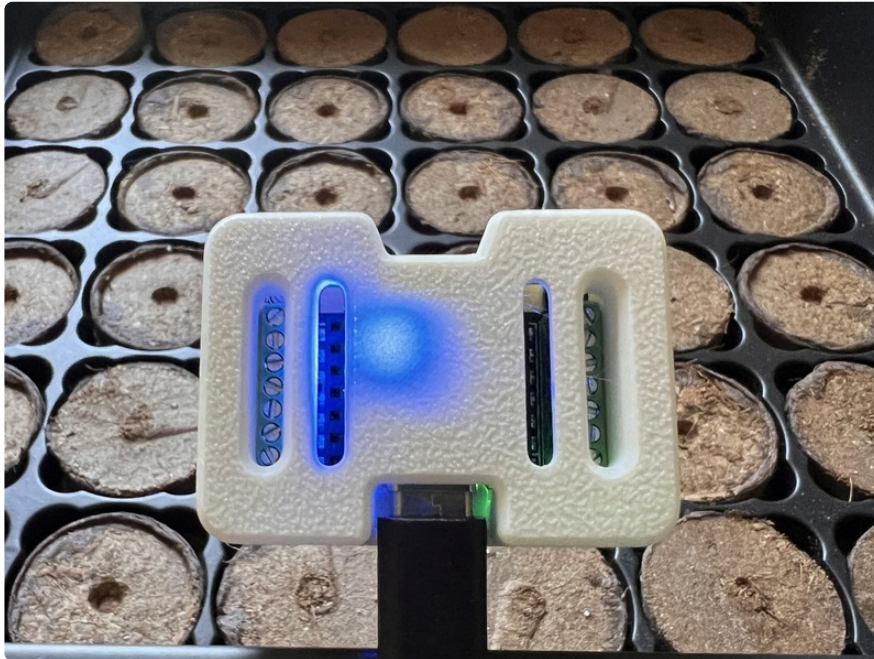
<https://learn.adafruit.com/no-code-seedling-frost-and-sun-alert-with-adafruit-io-and-wippersnapper>

Last updated on 2026-04-29 03:14:29 PM UTC

Table of Contents

Overview	3
<ul style="list-style-type: none">• What is WipperSnapper• Parts	
WipperSnapper Setup	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	
3D Printing	12
<ul style="list-style-type: none">• Parts• Build Volume• Case CAD• Design Source Files	
Assembly	14
<ul style="list-style-type: none">• Populate Terminal Block BFF• Assemble The Bottom Case• Snap-fit Case Top• Test-fit USB	
Configure QT Py	16
<ul style="list-style-type: none">• Add a NeoPixel Component	
Configure Weather Power-Up	18
<ul style="list-style-type: none">• Configure the Weather Power-Up	
Configure an Adafruit IO Action	21
<ul style="list-style-type: none">• Create a new Action• Build the Adafruit IO Action• 3. Set the NeoPixel LEDs• 4. Text Me!• Run the Action	
Usage	28

Overview



Don't lose seedlings to an overnight frost, protect them by building a frost alert device! **In this project, you'll use a QT Py ESP32-S3 and Adafruit IO to build a project which ambiantly glows blue when it is predicted to freeze overnight. When it glows blue, it's time to take the seedlings in overnight.**

This project is no-code and no-wiring. The QT Py runs Adafruit's WipperSnapper firmware. You'll use the Adafruit IO website to configure the QT Py's onboard NeoPixel. Then, **you'll configure an Adafruit IO Action to fetch the weather** for your location at the top of the hour. If the overnight forecast is freezing, a command is sent to the QT Py to turn the NeoPixel a bright blue color.

Additionally, **it will also send out a text message to your phone - reminding you to take in your seedlings overnight.**



This project utilizes the Adafruit IO Plus Weather Power-Up to fetch real-time weather data, so, an Adafruit IO Plus Subscription is required. This project **WILL NOT WORK** without an Adafruit IO Plus Subscription!

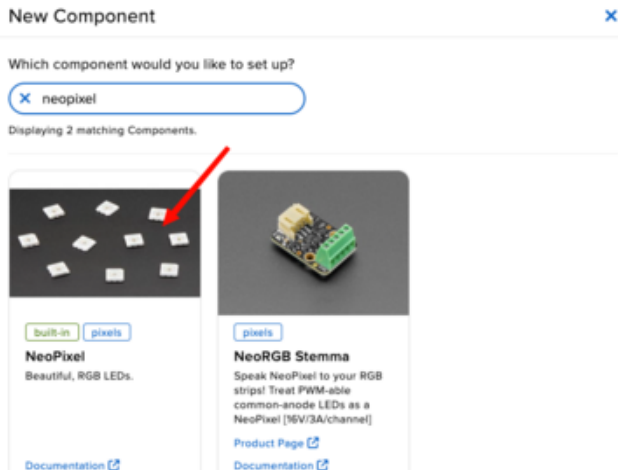
To upgrade your account, visit io.adafruit.com/plus

What is WipperSnapper

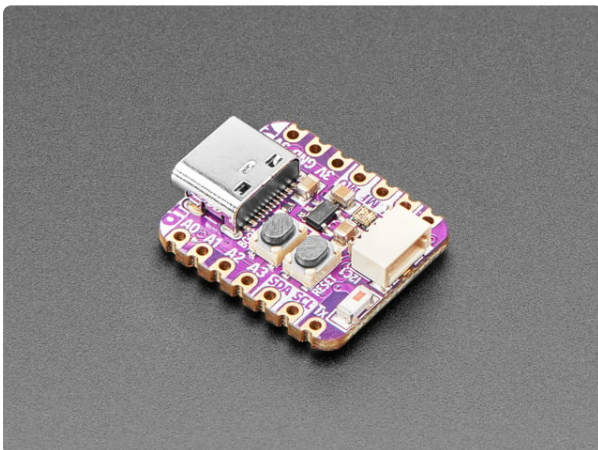
WipperSnapper is a firmware designed to turn any WiFi-capable board into an Internet-of-Things (IoT) device without programming a single line of code. WipperSnapper connects to [Adafruit IO](https://adafru.it/fsU) (<https://adafru.it/fsU>), a web platform designed (by [Adafruit!](https://adafru.it/Bo5) (<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.



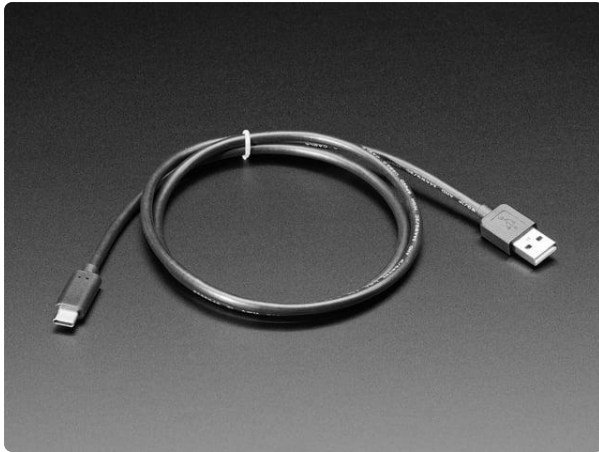
Parts



[Adafruit QT Py S3 with 2MB PSRAM WiFi Dev Board with STEMMA QT](https://www.adafruit.com/product/5700)

The ESP32-S3 has arrived in QT Py format - and what a great way to get started with this powerful new chip from Espressif! With dual 240 MHz cores, WiFi and BLE support, and native...

<https://www.adafruit.com/product/5700>



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>

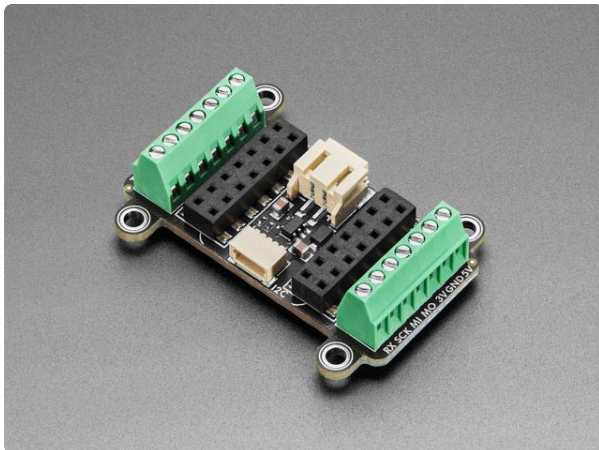


Adafruit IO+ 1 Year Subscription Card

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

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

i (Optional) The Terminal Block BFF and 3D printed case are NOT REQUIRED to complete this project. We included it to make it easy to extend the project by adding external components (i.e: adding extra LEDs, a local environmental sensor like the BME680, a servo, an irrigation pump, and more..)



Adafruit Terminal Block BFF Add-On for QT Py and Xiao

Our QT Py boards are a great way to make very small microcontroller projects that pack a ton of power - and now we have a way for you to make wiring up sensors and batteries to

<https://www.adafruit.com/product/6495>

WipperSnapper Setup



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 [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.

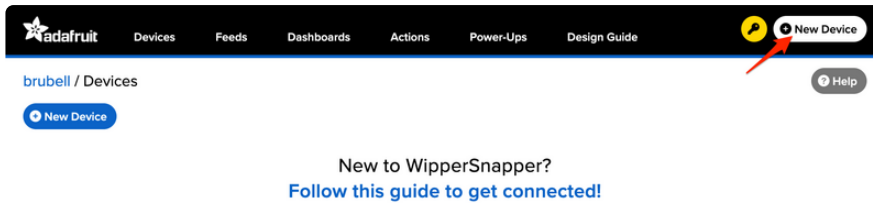
From there, you can add components to your board such as buttons, switches, potentiometers, sensors, and more! Components are dynamically added to hardware, so you can immediately start interacting, logging, and streaming the data your projects produce without writing code.

Sign up for Adafruit.io

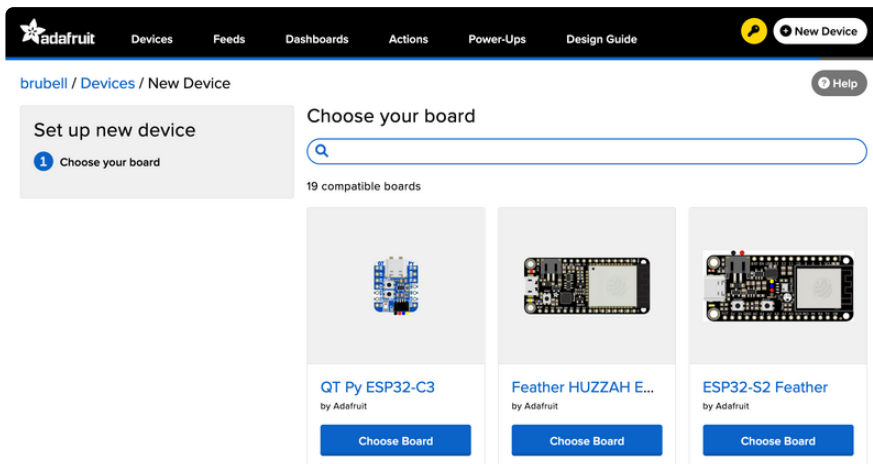
You will need an Adafruit IO account to use WipperSnapper on your board. If you do not already have one, head over to [io.adafruit.com \(https://adafru.it/fsU\)](https://adafru.it/fsU) to create a free account.

Add a New Device to Adafruit IO

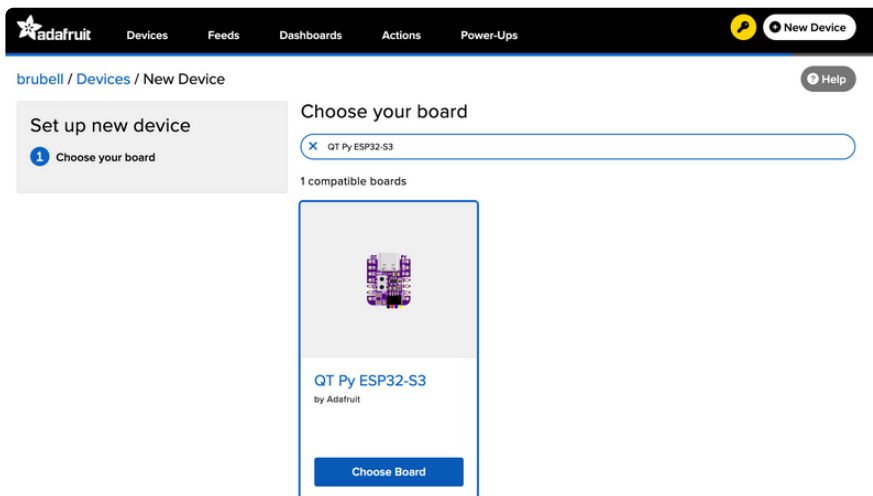
Log into your [Adafruit IO \(https://adafru.it/fsU\)](https://adafru.it/fsU) account. Click the New Device button at the top of the page.



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 the QT Py ESP32-S3. Once you've located the board you'd like to install WipperSnapper on, 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.

adafruit Devices Feeds Dashboards Actions Power-Ups [New Device](#)

brubell / Devices / New Device [Help](#)

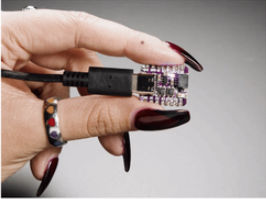
Set up new device

- ✓ Choose your board
- 2 Plug in USB**
- 3 Download WipperSnapper
- 4 Launch UF2 bootloader
- 5 Drag & drop UF2 file
- 6 Set up Secrets file
- 7 Upload Secrets file
- 8 Connect!

Plug in USB Cable

Plug your Adafruit QT Py ESP32-S3 into your computer using a known-good USB cable.

ⓘ A lot of people end up using a charge-only USB cable and it is very frustrating! Make sure you have a USB cable you know is good for data sync.



Once you've plugged your board into your computer, move on to the next step.

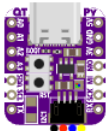
[← Back to Step 1](#)
[Next Step](#)

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".

New Device Detected! ✕

You have successfully connected a new **qtpy-esp32s3** device to Adafruit IO. It is already set up and submitting data. You can name the device here, and set up components on the device page.



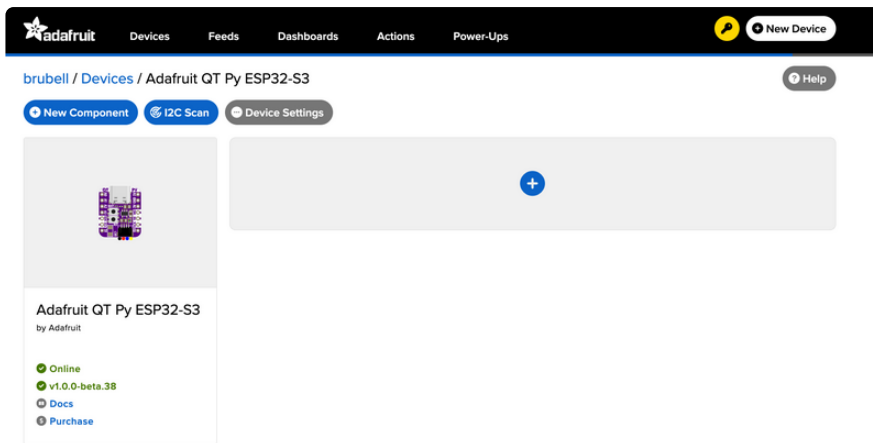
Device Name

Adafruit QT Py ESP32-S3

Firmware Version: ✔ v1.0.0-beta.38

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. \(https://adafru.it/V6a\)](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. \(https://adafru.it/V6a\)](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.

<https://adafru.it/10aU>

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

<https://adafru.it/10aV>

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

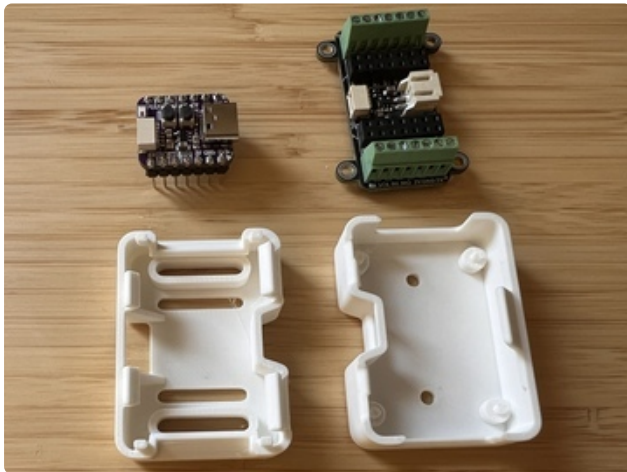
Factory Resetting a WipperSnapper Board

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.

3D Printing



(Optional) The Terminal Block BFF and 3D printed case are NOT REQUIRED to complete this project. We included it to make it easy for you to extend the project by adding external components (i.e: adding extra LEDs, a local environmental sensor like the BME680, a servo, an irrigation pump, and more..)

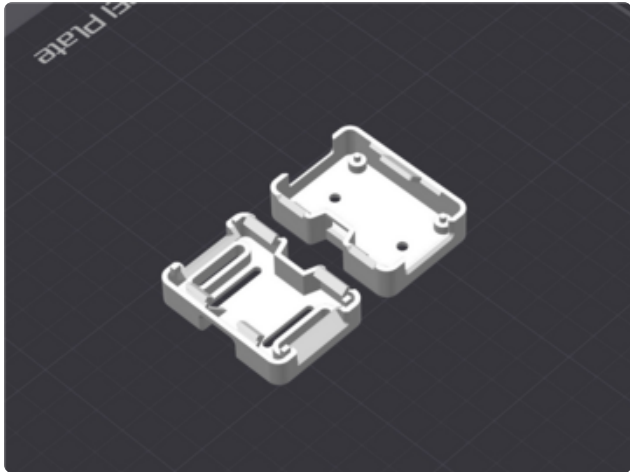


Parts

3MF files for 3D printing are oriented and ready to print on FDM machines using PLA filament. Original design source files may be downloaded using the links below.

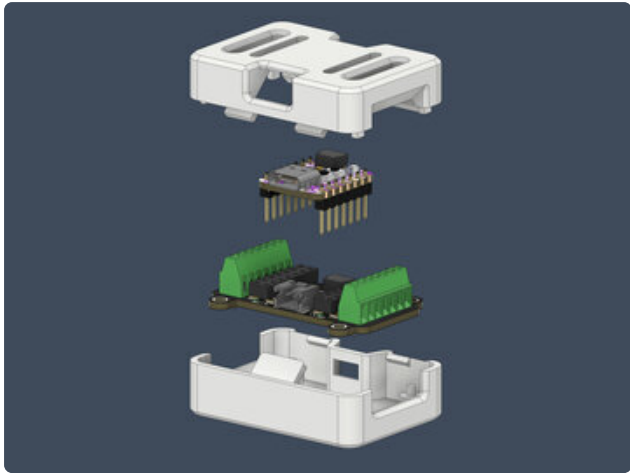
<https://adafru.it/1aD0>

<https://adafru.it/1aD1>



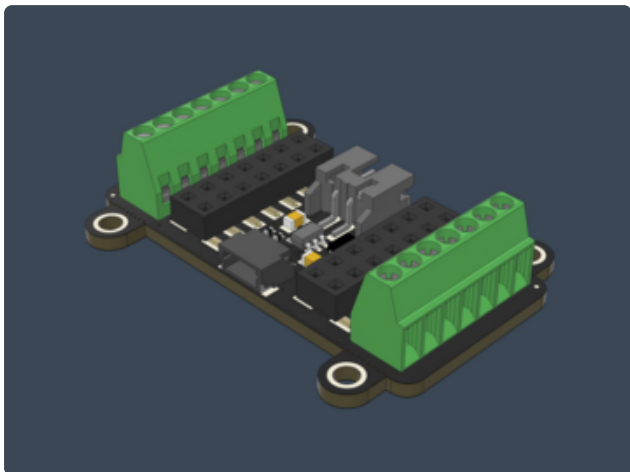
Build Volume

The parts require a 3D printer with a minimum build volume of 50 (X) x 38 (Y) x 16mm (Z).



Case CAD

The QT Py Terminal Block BFF fits into the bottom half of the enclosure. The top half of the enclosure snaps over the bottom half with access to the STEMMA QT port and the USB-C port on the QT Py.



Design Source Files

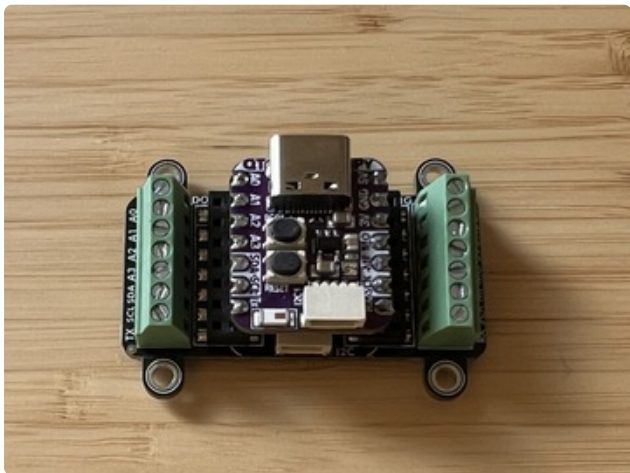
The project assembly was designed in Autodesk Fusion. STEP files are available in the CAD source zip file.

Electronic components like Adafruit's boards, displays, connectors and more can be downloaded from the [Adafruit CAD parts GitHub Repo \(https://adafru.it/RvF\)](https://adafru.it/RvF).

Assembly

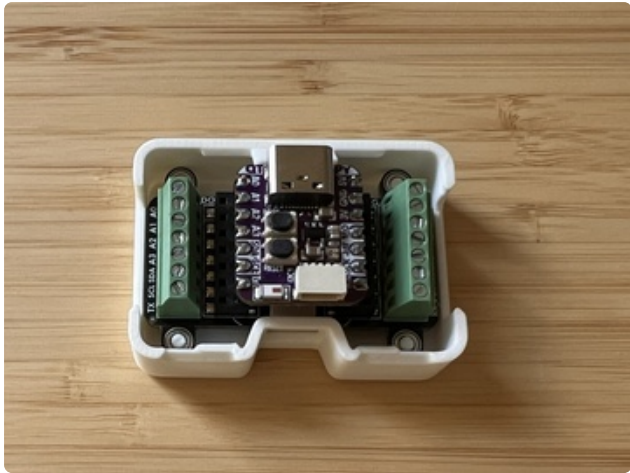
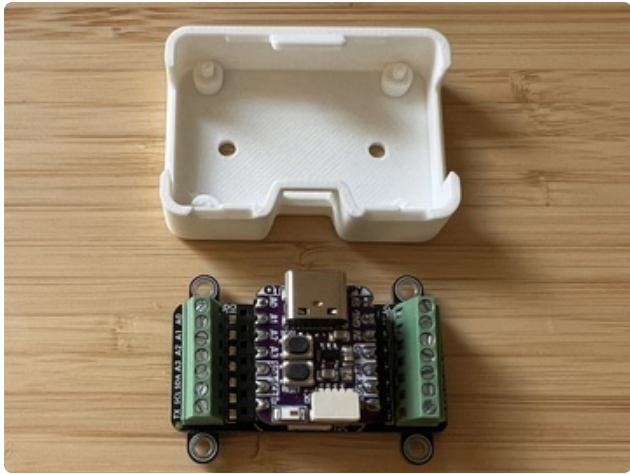


(Optional) The Terminal Block BFF and 3D printed case are NOT REQUIRED to complete this project. We included it to make it easy to extend the project by adding external components (i.e: adding extra LEDs, a local environmental sensor like the BME680, a servo, an irrigation pump, and more..)



Populate Terminal Block BFF

Gently press a QT Py into the Terminal Block BFF's socket headers. It should feel flush, do not force the parts together.



Assemble The Bottom Case

Place the QT Py and Terminal BFF into the bottom case, ensuring the case's pegs line up with the Terminal BFF's mounting holes

Snap-fit Case Top

Align the snap fit nubs to connect the top of the case to the bottom of the case.

Pressing the case together, you should hear a "snap"!

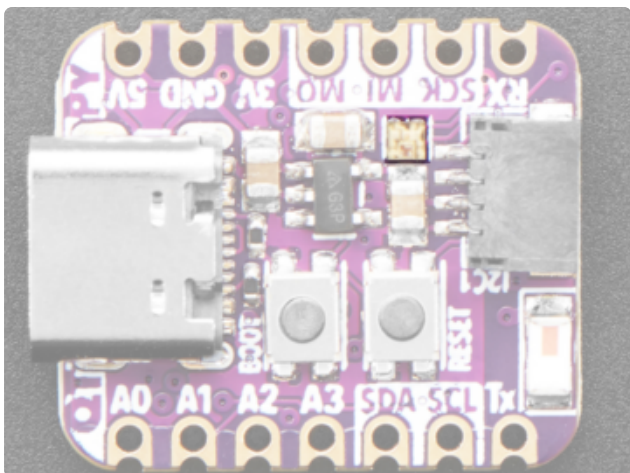


Test-fit USB

Before continuing, ensure you can plug a USB-C cable into the case.

Configure QT Py

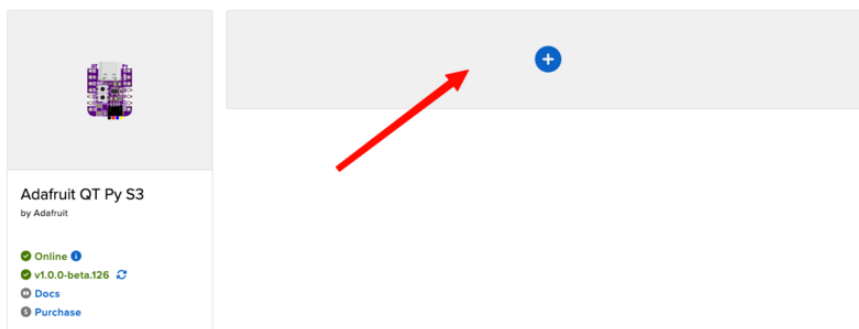
Add a NeoPixel Component



We'll use the QT Py's NeoPixel LED to visually indicate if it's freezing or scorching hot out.

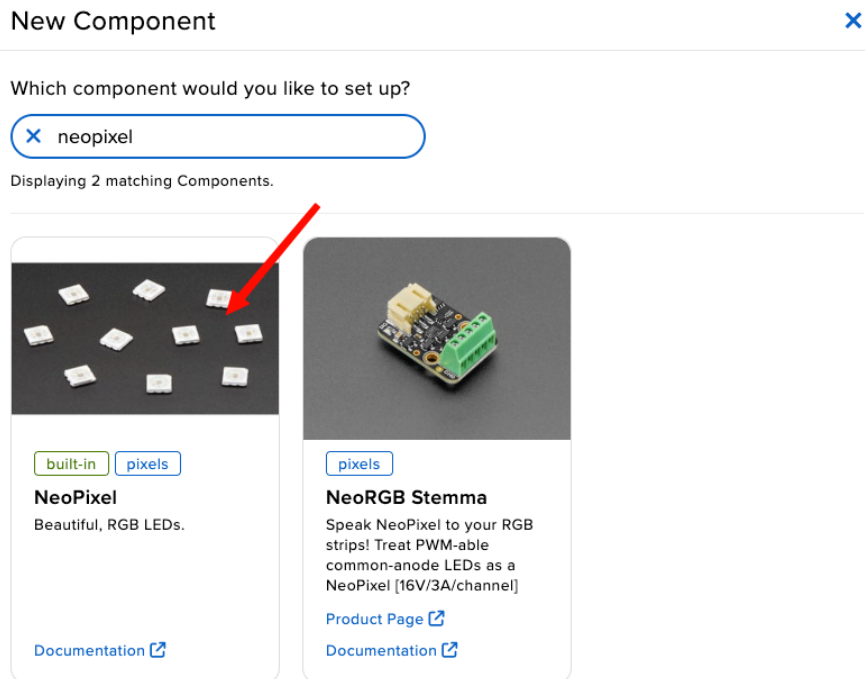
The QT Py has a NeoPixel built in, highlighted on this picture, no soldering/wiring is required to connect it.

Ensure your QT Py appears "Online" on Adafruit IO. From the QT Py's device page, click the New Component (or "+") button to open the component picker.



Search for the component name by typing **NeoPixel** into the text box on the component picker.

Click the NeoPixel component.



The QT Py has a NeoPixel built in, on Adafruit IO we pre-populate the form so it's ready for use.

Click Create Component.

Settings

NeoPixel Name

NeoPixel

NeoPixel Pin

D39 (NeoPixel)

Number of Pixels

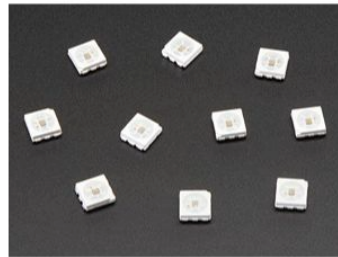
1

Color Order

GRB

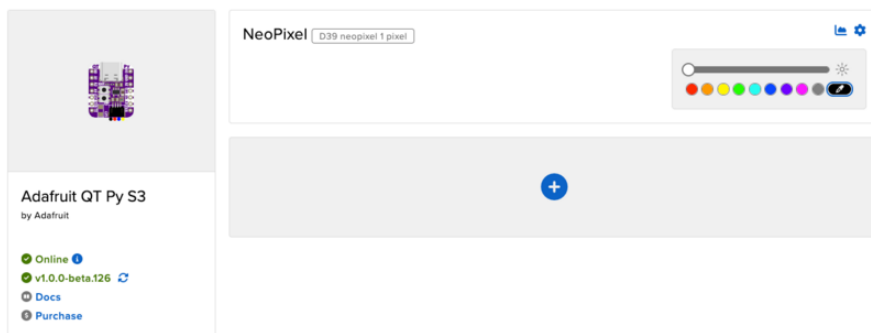
Brightness

255

[← Back to Component Type](#)[Create Component](#)

Behind the scenes, Adafruit IO sends a command to your QT Py telling it to "configure pin D39 as a NeoPixel LED".

Your board's page on Adafruit IO should display the new NeoPixel component.



Configure Weather Power-Up

This project utilizes the Adafruit IO Plus Weather Power-Up to fetch real-time weather data, so, an Adafruit IO Plus Subscription is required. This project **WILL NOT WORK** without an Adafruit IO Plus Subscription!

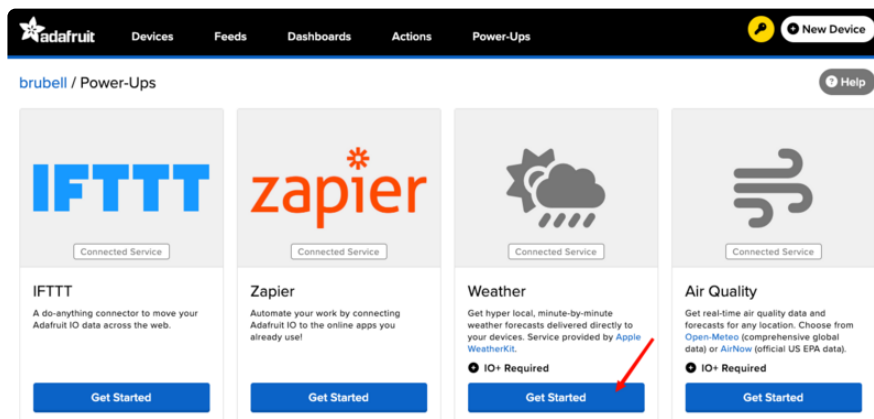
To upgrade your account, visit io.adafruit.com/plus.

Configure the Weather Power-Up

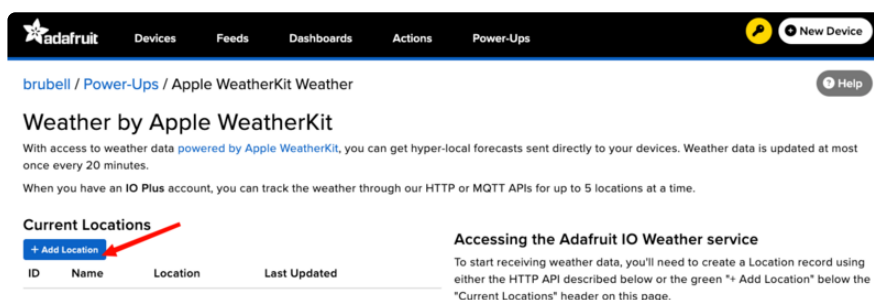
A perk of Adafruit IO Plus is the Weather Power-Up. With it, you can access hyper local, minute-by-minute weather forecasts (provided by Apple WeatherKit) and deliver it directly to your device.

We're going to use the Adafruit IO Weather power up to look at upcoming forecasts to see if seedlings need to be brought in overnight to shelter them from the frost.

First we need to set up your location in the Adafruit IO Weather Power-Up (if you haven't done this already). Visit the [Adafruit IO Power-Up page \(https://adafru.it/Eik\)](https://adafru.it/Eik) (or click "Power-Ups" on the header). Then, click on the Get Started button on the Weather Power-Up.



Under "Current Locations", click "Add Location".



Create a new Weather Service Integration ✕

Select a location from the map or enter your own `latitude,longitude` value.



Location

40.65417,-74.00874

Name

Brooklyn, NY

Enter a friendly name that will help you identify this location later.

Cancel

Create

Select a location from the map, or enter your own location in `latitude, longitude` format.

While `40.6517, -84.00874` describes an exact location for a hyperlocal forecast - we'll need to refer to the location by name later in this guide.

Give the location a name and click Create.

The new location should appear below **Current Locations**.

Current Locations

+ Add Location

ID	Name	Location	Last Updated	
2845	Brooklyn, NY	40.65417,-74.00874	n/a	View

Configure an Adafruit IO Action

Adafruit IO Actions lets you define the logic for this project without programming.



If you get stuck during this page, we have a guide all about Adafruit IO Actions: <https://learn.adafruit.com/how-to-use-blockly-for-actions-on-adafruit-io>

Create a new Action

Navigate to [the Adafruit IO Actions page \(https://adafru.it/YAT\)](https://adafru.it/YAT). Click New Action.



Give your action a name and a description.

Click Create.

What are we automating today? ✕

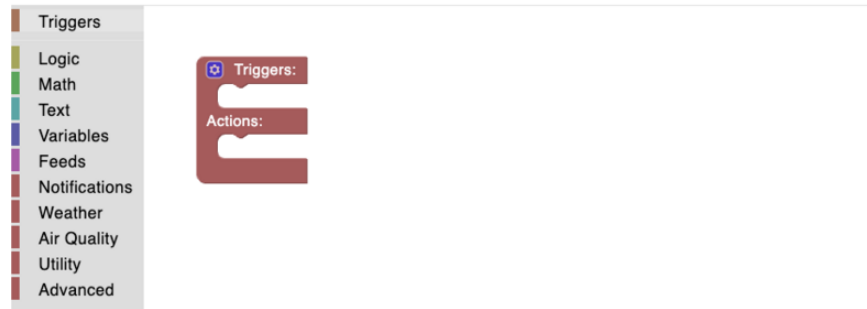
Give your Action a brief name:

...and a longer description (optional):

You should see the Actions workspace.

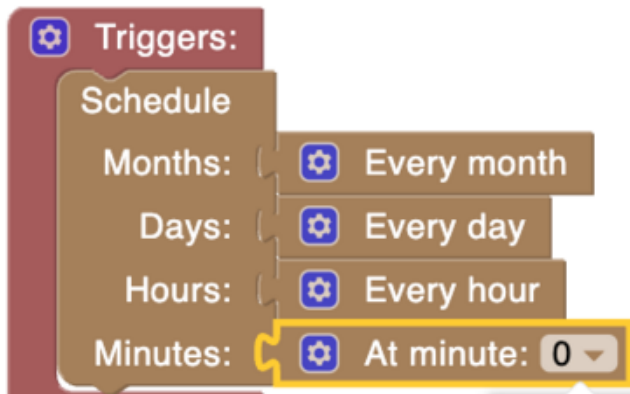
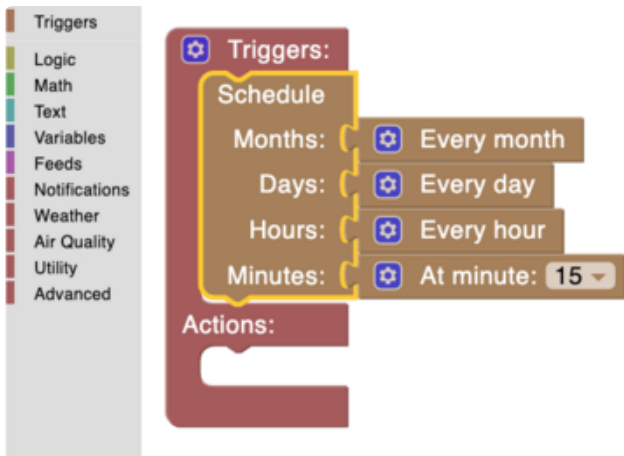
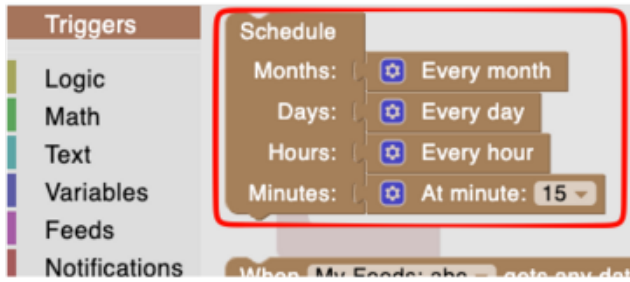


Read [the documentation](#), right-click any block and click "help", and get support or submit feedback [on the forum](#).



Build the Adafruit IO Action

Next, we'll build the Adafruit IO action for this project.



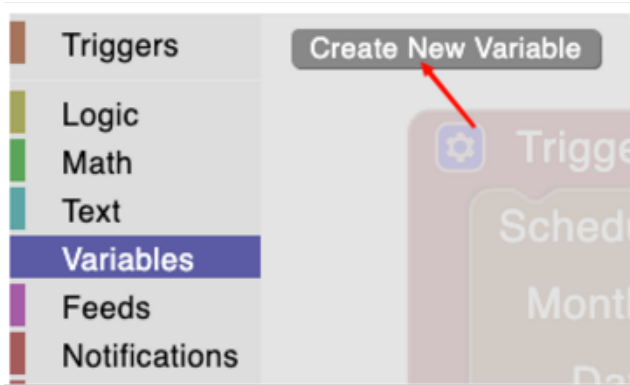
1. Set the Schedule

Every automation needs a trigger to tell it when to run.

Trigger: Choose **Schedule**.

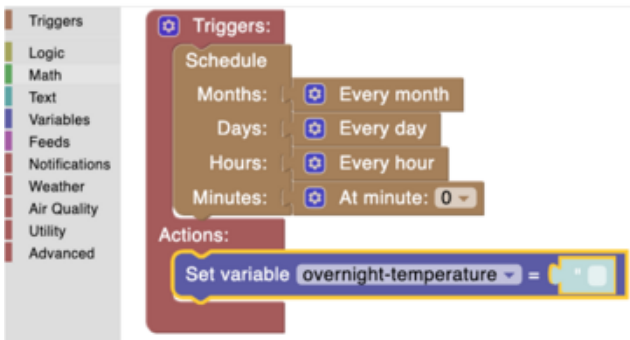
Frequency: Set it to run **Every Month**, **Every Day**, and **Every Hour**.

Minute: Set it to **At minute: 0**. This ensures your display updates at the top of every hour.



2. Check the Forecast

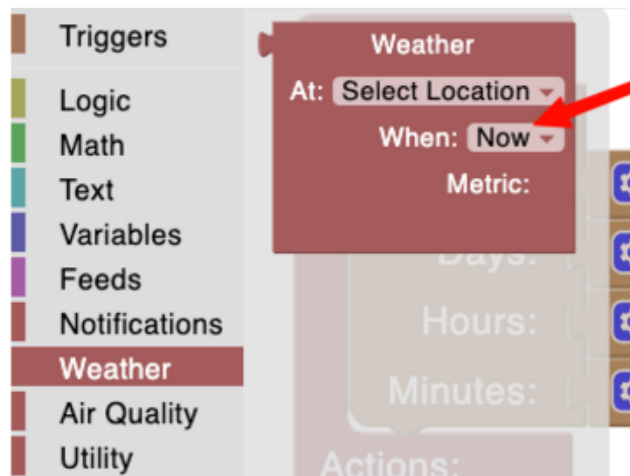
To get the minimum temperature it will be overnight, we'll first need a variable to store it.



Create a new variable: Name it overnight-temperature

Variable Block: Add an [Set Variable block](https://adafru.it/1aD2) (<https://adafru.it/1aD2>) for the variable you just created

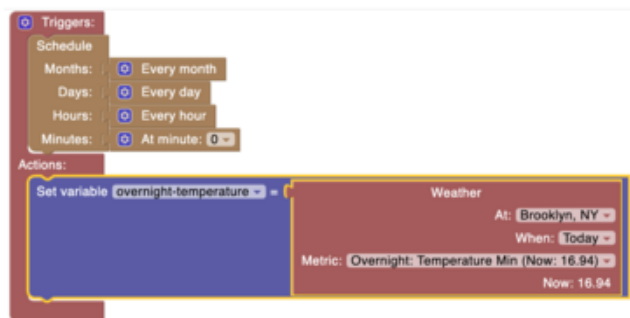
Next, set the variable to the weather forecast metric you want (overnight temperature).

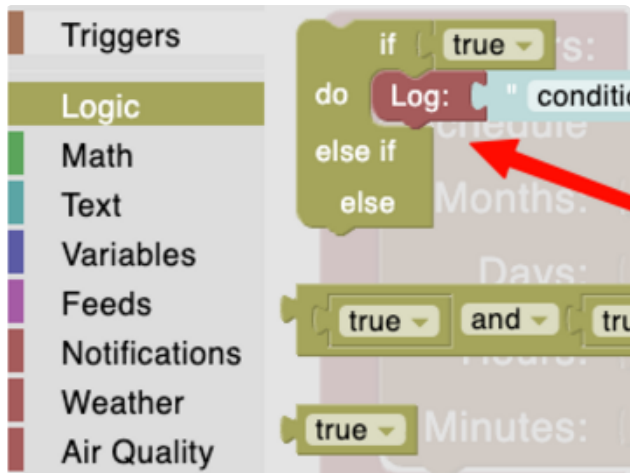


Weather Metrics: Select

[the Weather block](https://adafru.it/1aD3) (<https://adafru.it/1aD3>).

Set the location (e.g., **Brooklyn**), the time to **Today**, and the metric to **Overnight: Temperature Min**.

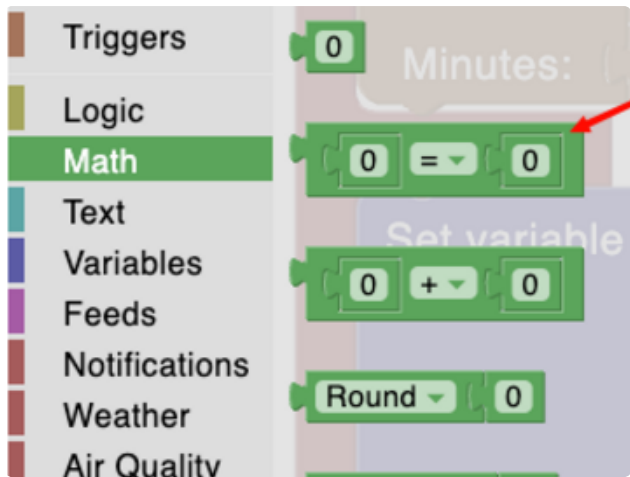




3. Set the NeoPixel LEDs

Next, to check if it is going to freeze overnight and set the NeoPixel accordingly.

Add a [conditional block](https://adafru.it/1avy) (<https://adafru.it/1avy>).



Add a [compare numbers block](https://adafru.it/1avz) (<https://adafru.it/1avz>).

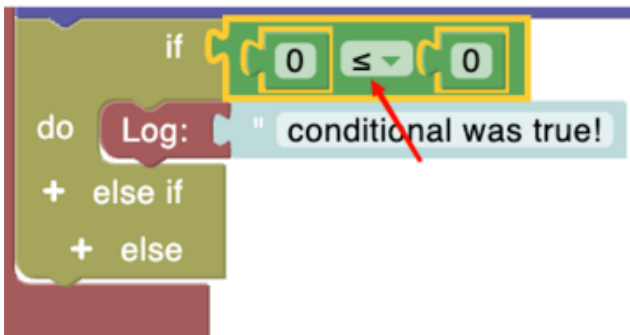
Conditional Block

If: Set the comparison to `<=`

Add a Get Variable Block and set it to the `overnight-temperature` variable.

Do: If it is going to frost overnight, use the Set block to change your QT Py NeoPixel to a blue hex color code to indicate frost, like `#0047ff`.

Else: If it's not going to frost overnight, set the NeoPixel to `#000000` (off). The light will only turn on when it's going to frost overnight.



```
if Get variable overnight-temperature ≤ 0
do Log: " conditional was true!
+ else if
+ else
```

```
Actions:
Set variable overnight-temperature = [Metric: Overnight: Temperature]
if Get variable overnight-temperature ≤ 0
do Set Adafruit QT Py ESP32-S3: NeoPixel to: "#0047ff
+ else if
+ else
```

```
if Get variable overnight-temperature ≤ 0
do Set Adafruit QT Py ESP32-S3: NeoPixel to: "#0047ff
+ else if
- else
do Set Adafruit QT Py ESP32-S3: NeoPixel to: "#000000
```



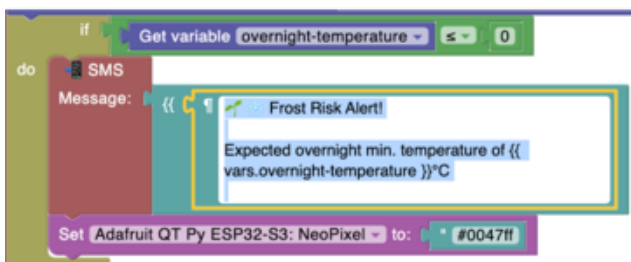
4. Text Me!

With Adafruit IO+, you can receive text messages for alerts. It's easy to add this functionality into your action.

Add a [SMS Block](https://adafru.it/1aD4) (https://adafru.it/1aD4). Set the message to something like:

☐ * Frost Risk Alert!

Expected overnight min. temperature of {{ vars.overnight-temperature }}°C



For more information about setting the variable in the text block - [check out this page](https://adafru.it/1axq) (https://adafru.it/1axq).

Run the Action

Click "Save and Run"

brubell / Actions / "🌱 Seedling Alert"



The run retrieves tonight's forecast (17.15 degrees Celsius) and shows the NeoPixel's color (#000000) in relation to the color code

Going forwards, the Action is enabled and will run at the top of every hour.

Run Report



Validation: Succeeded

(Triggers are skipped for "Run Now".)

Output:

```
[  
  17.15,  
  [  
    "#000000"  
  ]  
]
```

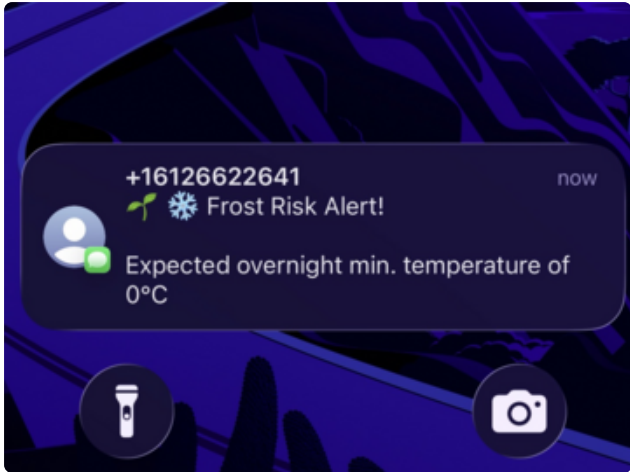
Usage



When the project is plugged into USB power, the QT Py boots up, connects to your WiFi network and syncs with Adafruit IO.



When frost is in the overnight forecast, the front panel glows blue letting you know its time to bring in your plants overnight.



In addition to glowing blue, you should receive a text message.



When the forecast does not call for frost overnight, the NeoPixels will not glow and you will not receive any text messages.