



Sound Reactive LED Top Hat

Created by Erin St Blaine



<https://learn.adafruit.com/sound-reactive-led-top-hat>

Last updated on 2025-03-31 01:02:20 PM EDT

Table of Contents

Overview	3
<ul style="list-style-type: none">• Before you Start• Materials	
Wiring Diagram	6
WLED Software	7
<ul style="list-style-type: none">• Board Choices• Driver Update• Install WLED• Setup & Preferences• Use It	
Electronics Assembly	14
<ul style="list-style-type: none">• Power Injection• Troubleshooting	
WLED Audioreactive Setup	19
Build the Hat	21
<ul style="list-style-type: none">• Diffusion & Decoration	
WLED 2d Matrix	26
<ul style="list-style-type: none">• Matrix Setup• Matrix Effects	
Create Presets	28
<ul style="list-style-type: none">• Control Presets	
WLED Button Setup	31
<ul style="list-style-type: none">• Troubleshooting	
WLED Additional Settings	32

Overview

Welcome, intrepid creators and enthusiasts of the extraordinary, to this illuminating tutorial on crafting your very own LED top hat. In this journey, we shall delve into the harmonious blend of Victorian elegance and modern marvels, weaving together the vibrant glow of Progress with the timeless charm of Style.

Imagine, furthermore, a hat that not only dazzles with light but also dances to the rhythm of sound and responds to your commands through the marvel of WiFi control. Prepare to embark on a splendid voyage into the heart of innovation, where tradition meets the future in a dance of light and shadow.



Before you Start

This is a fairly complex build with a lot of pieces. You may want to skim through the whole guide to get a feel for the scope of the project.

I'm going to take you pretty deep into some of the cool features of WLED. If this is your first time using this software, play around with the basics first. Your hat will be very cool even without the 2d effects or sound reactivity.

This project uses **WLED v 14.4 Audioreactive**. Be sure to select the Audioreactive option when you install the software.

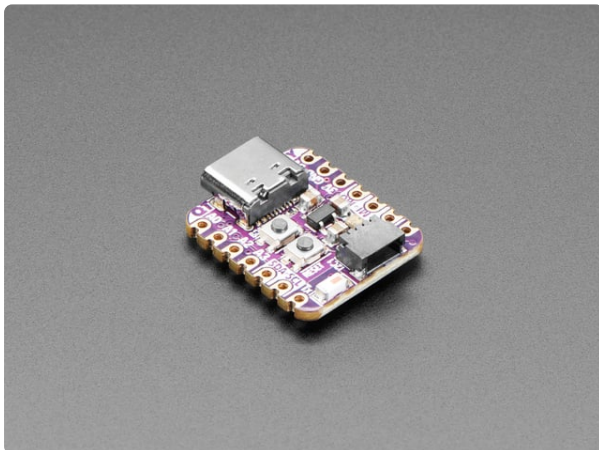
This hat features:

- Audioreactive modes using a PDM microphone breakout

- A decorative button that changes between modes, toggles audioreactive mode, and turns the LEDs on and off
- 2d Matrix setup for "mapped" animations
- A physical on/off switch
- WiFi control from any smartphone or computer

So, gather your tools, ignite your creativity, and get started on this magnificent adventure!

Materials



Adafruit QT Py ESP32 Pico - WiFi Dev Board with STEMMA QT

This dev board is like when you're watching a super-hero movie and the protagonist shows up in a totally amazing costume in the third act and you're like 'OMG! That's...

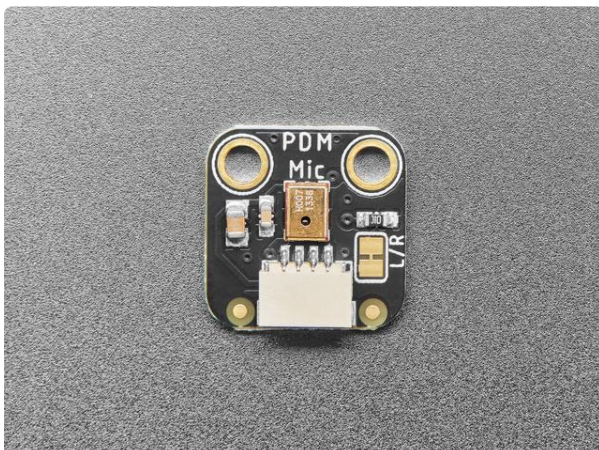
<https://www.adafruit.com/product/5395>



Adafruit NeoPixel Digital RGB LED Strip - Black 60 LED 4m

You thought it couldn't get better than our world-famous 32-LED-per-meter Digital LED strip but we will prove you wrong! You wanted...

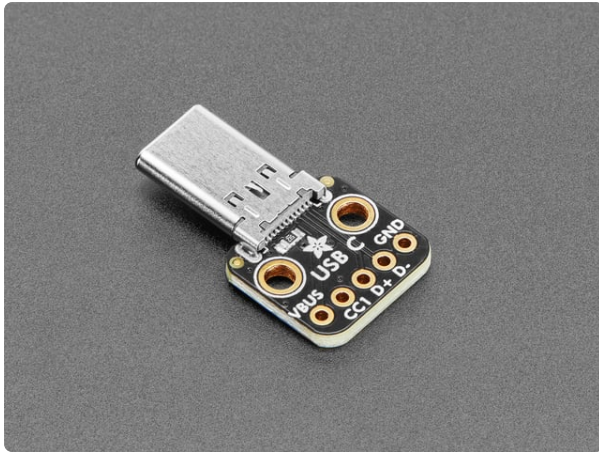
<https://www.adafruit.com/product/1461>



Adafruit PDM Microphone Breakout with JST SH Connector

An exotic new microphone has arrived in the Adafruit shop, a PDM MEMS Microphone! PDM is the 'third' kind of microphone you can integrate with electronics,...

<https://www.adafruit.com/product/4346>



Adafruit USB Type C Plug Breakout

Throw out all those Mini and Micro B USB cables you have in a plastic bin - the next generation of USB connectors is here with USB C! You've seen these...

<https://www.adafruit.com/product/5978>

1 x [Stemma Cable](https://www.adafruit.com/product/4210)

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

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

3 x [Silicone Stranded Wire](https://www.adafruit.com/product/1881)

<https://www.adafruit.com/product/1881>

Silicone Stranded Wire in Various Colors

2 x [JST Connectors](https://www.adafruit.com/product/2880)

<https://www.adafruit.com/product/2880>

2-pin JST Connectors

2880

1 x [USB C Adapter](https://www.adafruit.com/product/5328)

<https://www.adafruit.com/product/5328>

USB Type C Plug to Plug Adapter

5328

1 x [Tactile Switch Button](https://www.adafruit.com/product/1119)

<https://www.adafruit.com/product/1119>

Tactile Switch Buttons (12mm square, 6mm tall) x 10 pack

Additional Electronics

- [USB C Splitter](https://adafru.it/1acC) (<https://adafru.it/1acC>)
- [USB C Power Switch](https://adafru.it/1a5e) (<https://adafru.it/1a5e>)
- [USB C Battery Pack](https://adafru.it/1a5f) (<https://adafru.it/1a5f>)

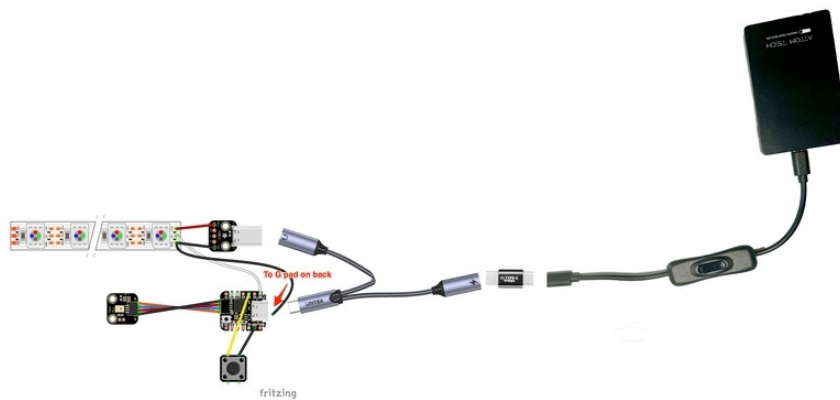
Hat Build Materials

- A Top Hat: find a nice one that fits comfortably with at least 2-3 inches of height above the crown of your head, and has fairly straight vertical sides - a funky, curvy hat will make it harder to line up the LED matrix
- 1/2" EVA craft foam for lining the hat and making an enclosure inside for the electronics
- Diffusion Material: I used [a pair of opaque tights](https://adafru.it/1a5g) (<https://adafru.it/1a5g>) to stretch over the hat
- Trim & Decoration for the hat that will hide your momentary switch button

Tools & Supplies

- Soldering iron & accessories
- Hot glue gun
- Heat gun
- E6000 glue
- Heat shrink
- Awl
- Needle & thread

Wiring Diagram



There are a lot of pieces to this project, but many of them just plug into one another so it's not quite as complex as it seems.

NeoPixel Wiring:

- **+5v** to USB Breakout **VCC**
- **G** to USB Breakout **G**
- **DI** to Qt Py **A1** (aka pin **25**)
- **G** to QT Py - (**G**) pad on the back of the USB port

PDM Microphone Wiring:

- Plug into the QT Py Stemma Port with a Stemma connector. Yellow wire will be connected to GPIO **19** and blue wire will be connected to GPIO **22**.

Momentary Switch Wiring:

- One leg to **A3** (aka GPIO **15**)

- Other leg to QT Py **G** pin

Power Wiring:

- USB battery plugs into the power switch tail
- Power switch tail plugs into the USB C male-male adapter
- USB c adapter plugs into the "power" input to the USB splitter (with the lightning bolt)
- USB splitter male connector plugs into the QT Py
- USB splitter female (headphone) connector plugs into the USB breakout/ NeoPixels

This is a little bit convoluted, but it gets things done without having to build custom USB cables, and there's plenty of room in the top hat for a few extra adapters.

WLED Software

Board Choices

WLED runs on several different boards in Adafruit's collection. There are different benefits to each, but the installation process is largely the same. This page contains instructions for multiple boards -- be sure to use the pinouts and installation instructions for the one you're using,

Sparkle Motion

This is our flagship ESP32 board, designed with WLED and Xlights in mind. It has 4 outputs and is set up to drive either 5v, 12v or 24v pixels. It's a workhorse of a board and for larger projects it's the clear winner. It has an onboard microphone for instant sound-reactive support, and an IR sensor built in, to make it easy to control your project with an infrared remote. It also has a couple stemma ports so you can add your own sensors or peripherals.

Sparkle Motion Mini

The Sparkle Motion Mini is a smaller version of the Sparkle Motion board. It has two LED outputs, a microphone, and two stemma ports that make it easy to add an IR sensor or other peripherals. It's got an onboard NeoPixel and a small footprint, making it perfect for wearables or smaller projects. It will power a whole lot of pixels through the onboard USB port: it's safe to draw up to 4A through this port, giving you plenty of power for most wearable projects.

At this time, the Sparkle Motion Mini works best with WLED 0.15.1 -- the extra GPIO for the microphone pins are not supported in WLED 0.15.0. This should be fixed with the release of version 16.

To get mic support now, the following combined .bin file can be used. Get it by downloading this zip file:

esp32_bootloader_v4_WLED_0.16.0-alpha_ESP32.zip

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

To install, extract the .bin file from the zip and then follow the same [ESB Web Flasher process used for installing CircuitPython \(https://adafru.it/1adF\)](https://adafru.it/1adF). At the "Programming the Board" step, choose the .bin file and leave offset as 0x0.

QT Py Pico ESP32

The [QT Py Pico \(http://adafru.it/5395\)](http://adafru.it/5395) is small and affordable, so usually my go-to for smaller costumes or wearables. It also has a range of BFF add-on boards that add functionality. [Here's a guide with more QT Py info \(https://adafru.it/1abD\)](https://adafru.it/1abD). The QT Py will drive up to around 30 pixels through the onboard USB port, so if you have more LEDs than that you may want to consider the Sparkle Motion Mini instead, or you can power the board through the +5v pin.

Note: WLED works on the QT Py Pico but NOT on the S2 or S3 versions, at the time of writing.

Feather Huzzah ESP32

The [Feather Huzzah ESP32 \(http://adafru.it/3405\)](http://adafru.it/3405) the top of the line. It's a great choice for projects where you want to add sensors, interaction, or drive a whole lot of LEDs. It's the most reliable as well -- I've run these for two months straight with no power cycling and they just keep on truckin. Adafruit has a very wide selection of Feather Wing boards that connect to the Feather microcontroller line. The sky is the limit with these boards.

It also comes in a version with a high-powered WiFi range extender! If you're trying to sync multiple instances across distance, check this one out. [Feather Huzzah ESP32 V2 w.FL Antenna \(http://adafru.it/5438\)](http://adafru.it/5438)

Feather Huzzah ESP8266

The [Feather Huzzah ESP8266 \(http://adafru.it/2821\)](http://adafru.it/2821) will run WLED as well, but won't drive as many pixels: the ESP32 limit on WLED is around 1000 pixels per input, but the ESP8266 tops out at around 500. It's about \$5 cheaper though, so for smaller

projects it's a great way to save a little money and still have access to all the Featherwing options in the Adafruit store.

Driver Update

Some versions of our controllers have a new serial chip which needs a driver installed before we can install WLED. Head over to our [How to Install Drivers for WCH USB to Serial Chips](https://adafru.it/-f8) (<https://adafru.it/-f8>) tutorial, and download and install the new driver.

If you have an older QT Py with CP2102 USB-to-Serial bridge, [use SiLabs' driver instead](https://adafru.it/11em) (<https://adafru.it/11em>).

Install WLED

These next steps require a **Web Serial-compatible browser**. As of this writing, that means **Google Chrome**, **Microsoft Edge** or **Opera** “desktop” browsers. Other browsers (Safari, Firefox, Explorer and anything mobile) won't work.



Visit <https://install.wled.me/> (<https://adafru.it/11dL>)

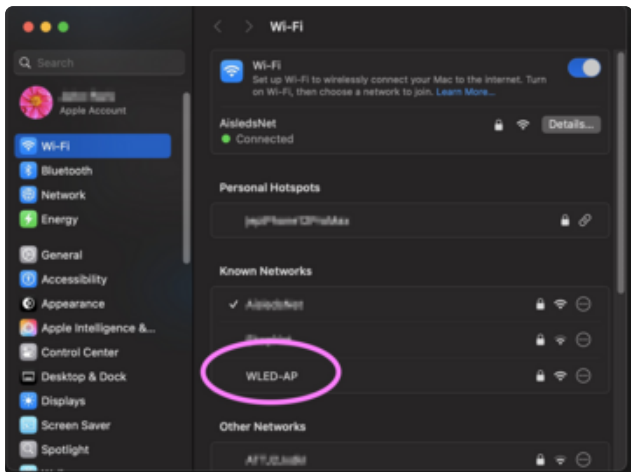
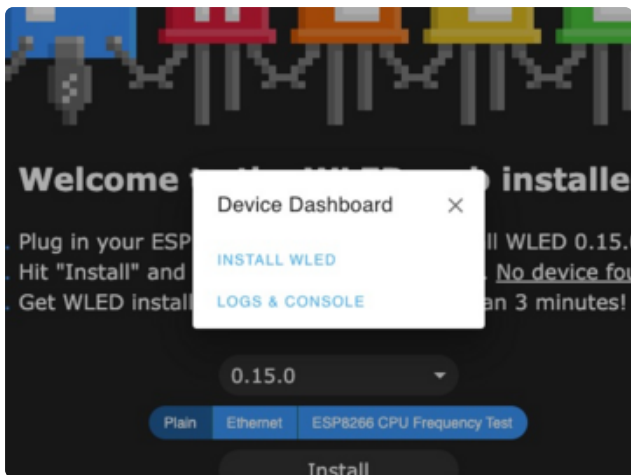
Plug your microcontroller into your computer with a known good USB cable. Click "Install" and select the port for your board.

Depending on the USB-to-serial bridge chip on the board, you might see one or two serial ports. On Mac, for instance, there might be both “/dev/cu.usbmodem[number]” and “/dev/cu.wchusbserial[number]”. Use the “wchusbserial” one.



After successful installation, enter your WiFi network name and password when prompted. This must be a **2.4 GHz** WiFi network; ESP32 does not support 5 GHz networks. If it can't connect, then as a fallback WLED will create its own 2.4 GHz WiFi access point.

Sometimes the "Connect to Wi-Fi" prompt doesn't show up. Don't panic, just see the step below on connecting your computer or mobile device to the WLED-AP access point created on the microcontroller itself!



If you don't see the "Connect to Wi-Fi" prompt, you'll need to set up your WiFi network using AP (access point) mode. Open up your WiFi settings and look for a WiFi network called **WLED-AP**. (Note, this access point can take up to 30 seconds to appear sometimes.) Connect to this network using the default password **wled1234**. The WLED interface will pop up in its own captive browser window.

From here, go into **Config/Wifi Settings** and enter your WiFi credentials for the access point you normally use near the top.

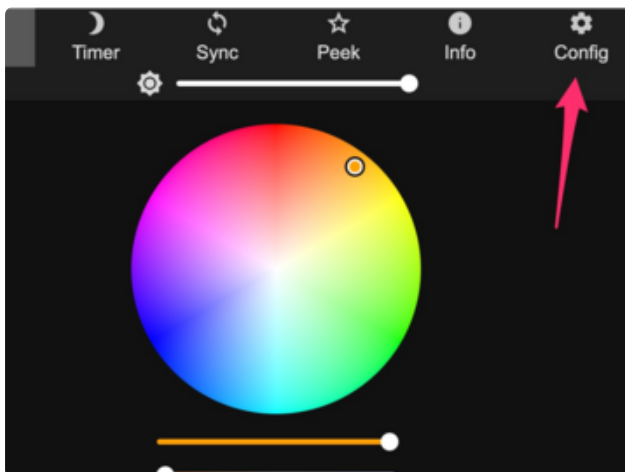
Give your project a name in the mDNS field a little further down the page. Now you can type in "projectname.local" (where "projectname" is your mDNS name) into any web browser on the same wifi network to access your microcontroller.

You can also scan the QR code below to open access point mode.

For more help and troubleshooting tips visit the [Getting Started page on the WLED knowledge base](https://adafruit.it/1acu). (<https://adafruit.it/1acu>)



Setup & Preferences



WiFi Setup

Head to the **WiFi Setup** screen under Config and create a good URL so you can control your project from any web-enabled device. Call it something you'll remember, that's easy to type into any web browser on your WiFi network in order to connect to your project.

In Safari or Chrome on your phone or computer, type in this web address to access the WLED interface: <http://projectname.local> (<https://adafru.it/1acs>) (where "projectname" is whatever you put into this field).

Check out the Additional Settings page for more info on accessing your project. WLED has an "access point mode" that doesn't require a WiFi network for when you're out on the go. It's also helpful to download one of the WLED apps to help manage and organize your projects.

LED Preferences

Next, head to the **LED Preferences** tab under the Config menu.

Scroll down to **Hardware Setup**. Put your total number of LEDs into the "Length" field, and change GPIO to the pin number associated with the pin you soldered to. Check the pinout diagram for the board you're using (it's the number in yellow).



[Sparkle Motion Pinout \(https://adafru.it/1acD\)](https://adafru.it/1acD)

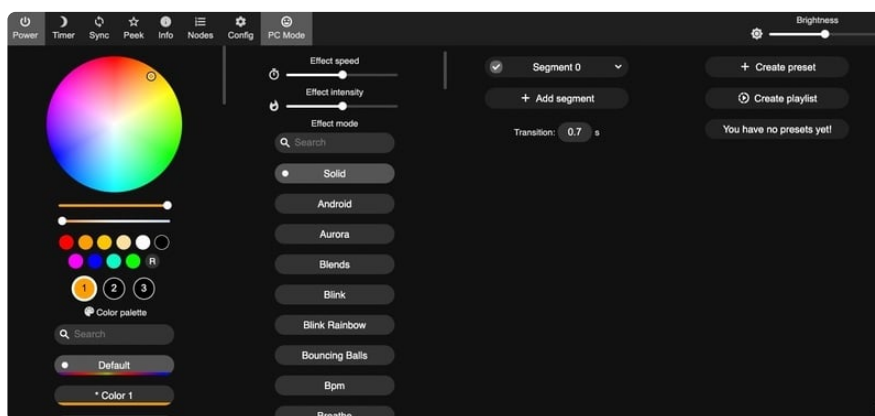
[Sparkle Motion Mini Pinout \(https://adafru.it/1acE\)](https://adafru.it/1acE)

[QT Py Pico Pinout \(https://adafru.it/11dK\)](https://adafru.it/11dK)

[Feather Huzzah ESP8266 Pinout \(https://adafru.it/1a53\)](https://adafru.it/1a53)

[Feather Huzzah ESP32 Pinout \(https://adafru.it/1aaj\)](https://adafru.it/1aaj)

Use It



Now you can use any computer or handheld device to control your LEDs.

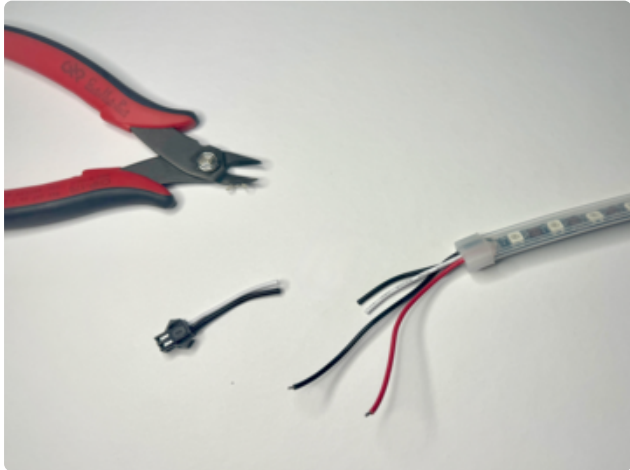
Make sure your device is on the same WiFi network as your board. Navigate to your custom URL (projectname.local/) in a web browser. You'll see a color picker above a whole bunch of color palette choices.

Choose a color, choose an effect, and watch your lights animate and glow!

Save your favorite combinations as presets, create playlists, control the speed and intensity of the animations, and lots more. This web app is incredibly intuitive and easy to use.

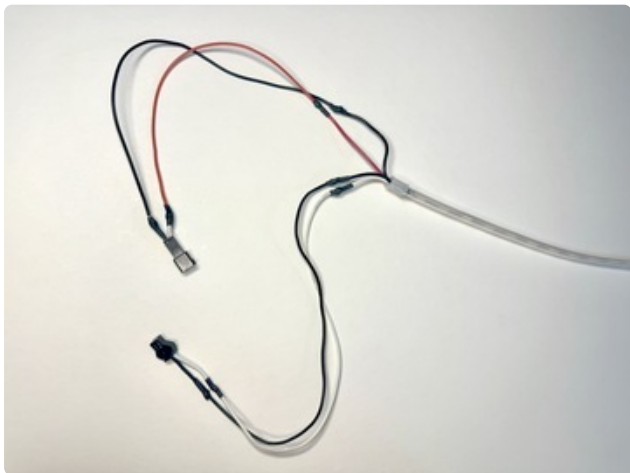
Head over to the WLED wiki at <https://kno.wled.ge/> (<https://adafru.it/11dN>) to delve into all the particulars.

Electronics Assembly

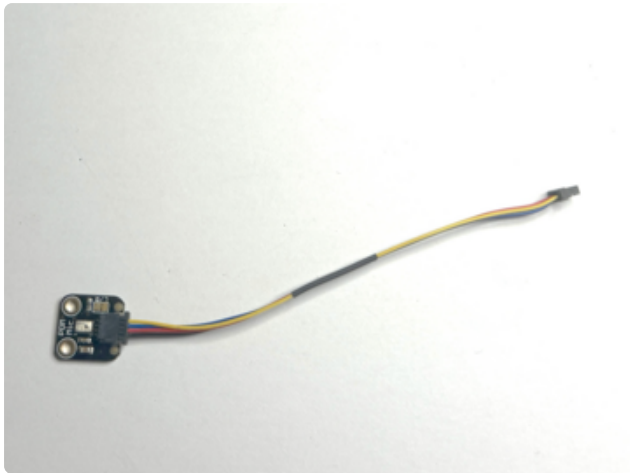
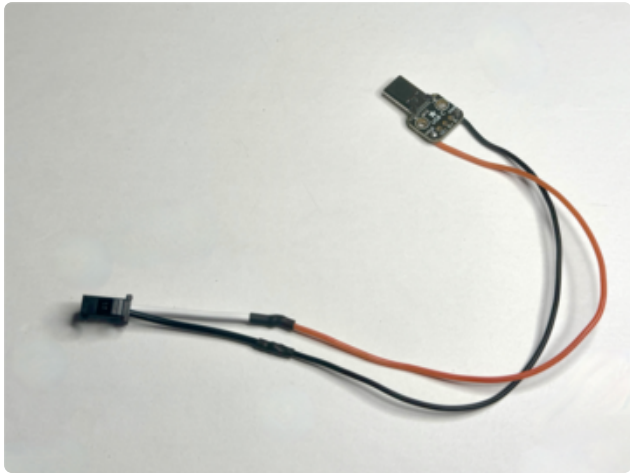
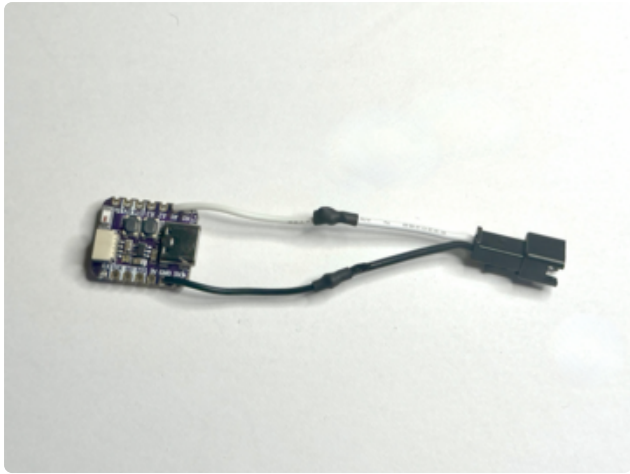


Find the IN end of your LED strip. These are directional and won't work if you solder to the wrong end. Look for the arrow to indicate the direction of data flow.

On a brand new roll of lights, you'll find some wires and a female connector at the end. Cut off the connector and extend the wires a bit and then reattach it. This will keep the connector near the top of the hat instead of pressing into your forehead.



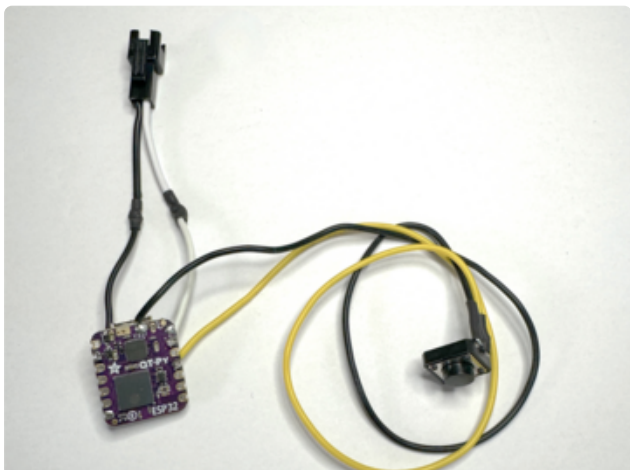
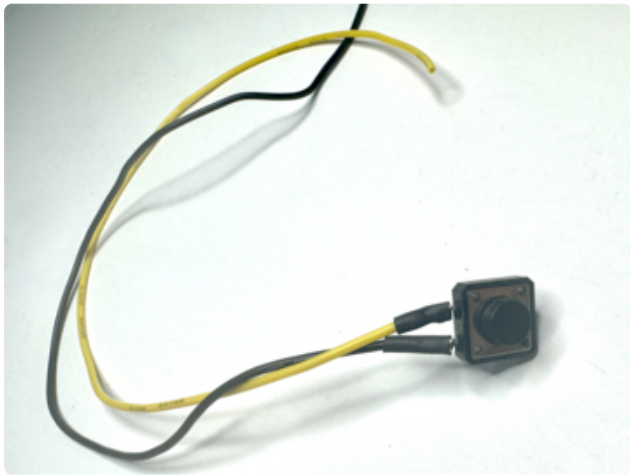
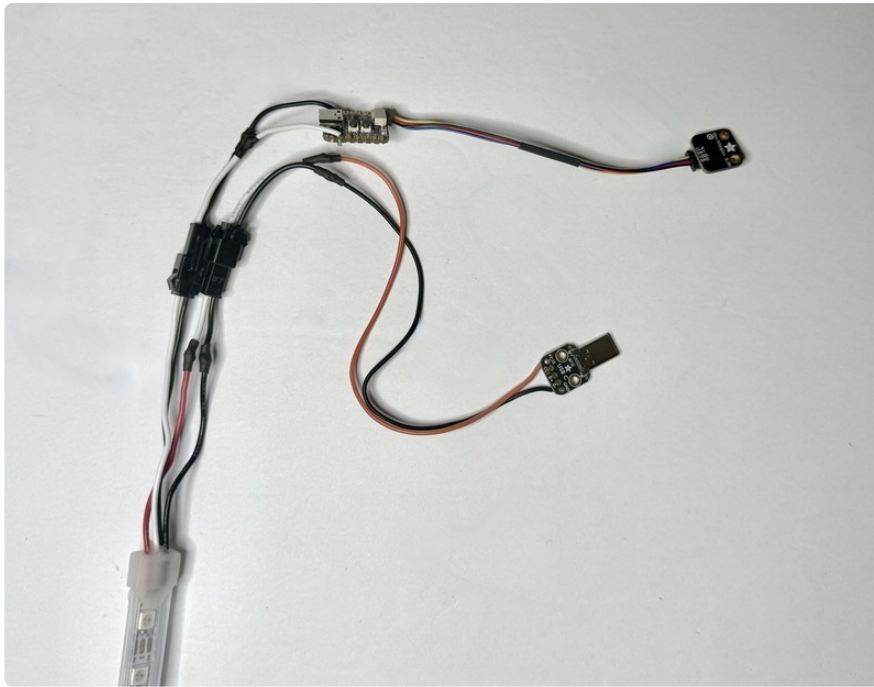
Extend the red and black wires as well and add a male connector. Using a male here instead of another female will keep you from accidentally plugging them in wrong.



Grab the male side of your connector. Solder the white wire (NeoPixel data) to a GPIO pin on the QT Py. I used **A1**. Solder the black wire (NeoPixel G) to the G pin.

Next, solder a female connector to your USB breakout using around 6-8" of lead wire. Red (NeoPixel V+) goes to VBUS and black (NeoPixel G) goes to G.

Plug your PDM Microphone breakout into the QT Py using the Stemma cable.

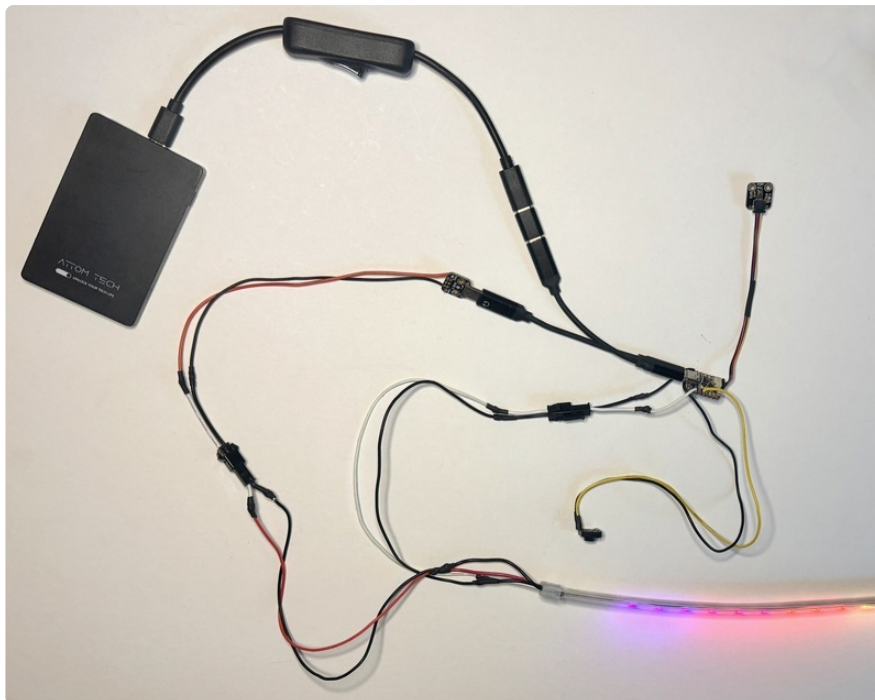


Add 8" wires to the legs of your momentary switch. I used a black and yellow wire, but would recommend using two black wires here instead, since the button goes on the outside of the hat and the wires might show.

Solder one wire to a GPIO pin (I used A3) and the other to the G pad on the back of the QT Py's USB port.



Plug the power switch tail into the USB battery, then use the adapter to plug it into the splitter on the "charge" side. Mine is marked with a lightning bolt.



Plug the USB breakout into the other side of the USB splitter -- mine is marked with a headphones icon.

Plug the male end of the splitter into the QT Py. Flip your switch, and your lights should power up. If they did, success! Do a happy little dance. If not, check the troubleshooting section below.

One 4m reel of 60/m pixels covered about half my hat. My total length ended up being close to 7.5m long, so I needed to extend my LED strip. This is definitely possible but not that easy to do, especially since it's important to keep the LED spacing consistent for this project so the LEDs line up as well as possible in the matrix layout.



I cut the last pixel off the strip so I had a clean copper pad to work with, then slipped clear heat shrink over the pixels. I carefully made three solder bridges by tinning the pads and then heating each one until it flowed together with the other strip. I was careful to check for any shorts, then tested.

When it worked, (which wasn't the first try), I slid the heat shrink over the section and filled it up with hot glue to keep it all in place.

Power Injection

My hat has over 350 LEDs on it. I was concerned that the voltage drop across that many lights would make the ones at the top noticeably dimmer than the ones at the bottom.

However, it turns out this wasn't a problem for me. I am a pretty respectful person, by nature, so I'm keeping the brightness turned way, way down on this hat so as not to blind anyone who looks at it. With the brightness at 20% or lower, I do not see a noticeable voltage drop.

If your LEDs seem to be dimming toward the end of the strand, you can add an additional power and ground connection to the pads at the very end of the strip, and splice those wires back to the power wires coming from the USB breakout. This will solve your dimming problem. But please.. be kind, don't make me blind.

Troubleshooting

If they didn't power up, here are a few things to try:

1. Plug your QT Py in directly to power with a USB cable into its port. The LEDs won't come on (they're not getting power) but you should be able to connect to the QT Py's WLED interface. If you can't connect, the trouble is with the software - try reinstalling. If you can connect, and WLED appears to be working fine, the trouble is with the wiring.
2. Be sure you soldered to the correct end of the LED strip. These strips are directional, and won't work if you connect the data wire to the wrong end. [More about how to solder NeoPixels here \(https://adafruit.it/LDV\)](https://adafruit.it/LDV)

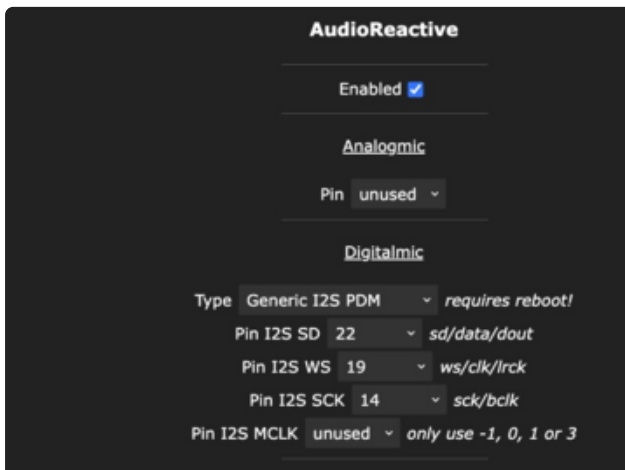
3. Open the WLED software and go to the LED Settings page. Make sure you have the correct pin (GPIO) entered here. Find this out in the [pinout diagram in the QT PY guide \(https://adafru.it/11dK\)](https://adafru.it/11dK). I soldered to **A1**, so I entered **25** here.
4. Try switching the two ends of the USB splitter. Mine only works in one configuration.
5. Try plugging in a USB cable instead of your battery, just for testing purposes. I found that some USB batteries won't power this project at all, it just.. doesn't turn on. The one I'm using works great: [USB Battery Pack \(https://adafru.it/1a57\)](https://adafru.it/1a57)
6. If your lights blink or red-out, try turning down the brightness in WLED.

WLED Audioreactive Setup



This guide uses WLED version 14.4: Audioreactive.

Check to be sure that you have the Audioreactive version of WLED installed. If you installed the plain version, no worries - head back to [install.wled.me \(https://adafru.it/183D\)](https://adafru.it/183D) and choose the Audioreactive tab, then install it again. Your settings won't be overwritten.



Click **Config** and then select the **Usermods** tab. Scroll down a bit and you'll find the AudioReactive section.

Click the box to enable, then enter the settings and the Digitalmic section as follows:

Type: **Generic I2S PDM**

Pin I2S SD: **22**

Pin I2S WS: **19**

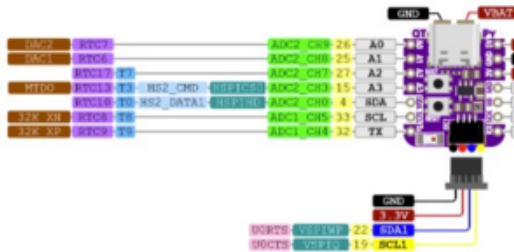
The other pins are unused.

Reboot your QT Py for changes to take effect.



How to figure this out? The back of the microphone breakout is labeled: the clock line is plugged into the yellow wire and data is plugged into blue.

Adafruit QT Py ESP32 Pi
<https://www.adafruit.com/product/5>



Look at at the [pinout diagram \(https://adafru.it/11dK\)](https://adafru.it/11dK) to find out which pins are assigned to each wire on the stemma port. The blue wire from the Stemma port goes to pin 22, and the yellow wire is pin 19.

There are a lot of other settings you can adjust in WLED. I found that the default settings for this mic seem to work perfectly, but you can delve into the particulars at <https://kno.wled.ge/advanced/audio-reactive/> (<https://adafru.it/1a56>)

Build the Hat



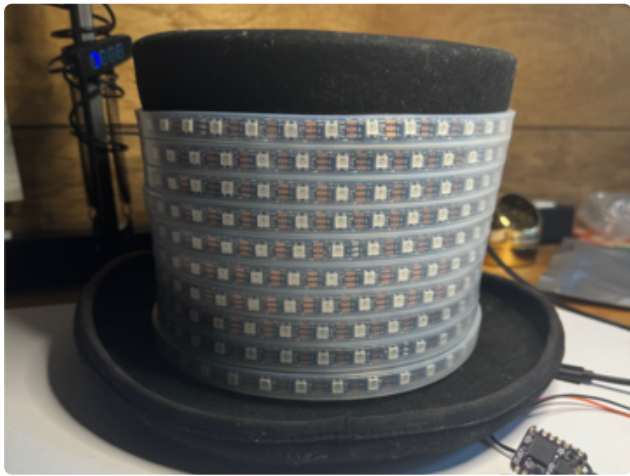
Find the center back of your hat and make a hole with an awl about 3/4" to the left of center, right along the base. The idea here is to get the first LED in the strip to be right in the center, so make the hole off-center enough to account for the wires and endcap of the strip.



Also, double check and make sure you're starting in the back of the hat and not the front! I made this mistake and had to redo the whole thing -- the matrix effects will look better centered on the front, which means starting in the back.

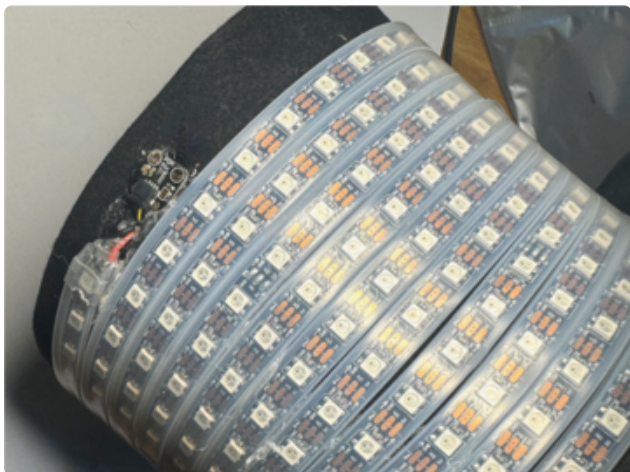


Start winding the lights around the hat, making sure the lights are lining up in straight vertical lines, as best you can. My hat's sides are not perfectly vertical, so the bottom strip is a little bit off, but the rest are pretty good. I stopped before running into this same alignment problem at the top where it flares out a bit again.



It might be helpful to have some shims to slip in between the hat and the LEDs if your hat is just the wrong size to get the lights to line up.

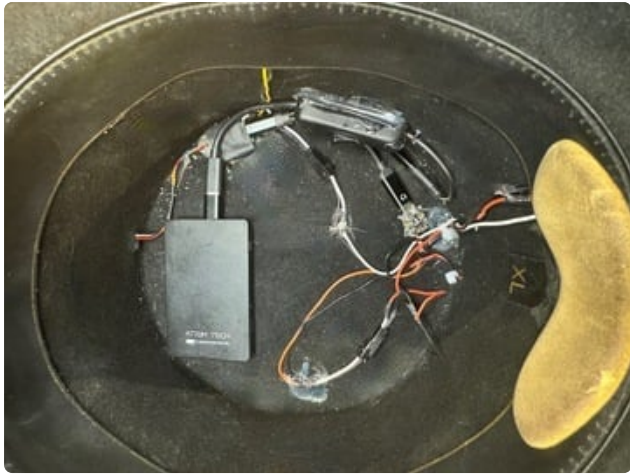
Secure each strip in place with a bit of hot glue to hold it. The hot glue will hold but not super well so it's useful for prototyping and adjusting the wire.



Once you get to the far end and you're happy with placement, secure it permanently to the hat using silicone glue or a physical connection like a zip tie or sewing thread.

Decide where you want to place your microphone. It needs to be on the outside of the hat to pick up enough sound. I put it on the front of the hat, so it will pick up the voice of the person I'm talking to, which is pretty fun.

Poke a hole for the Stemma connector and thread it to the inside, and then glue the microphone down securely.



Decide where you want to place your button. I put mine on the left side at the brim, slightly forward, right where many hats have feathers or decoration.

Slip the button through the hole so it's on the outside.



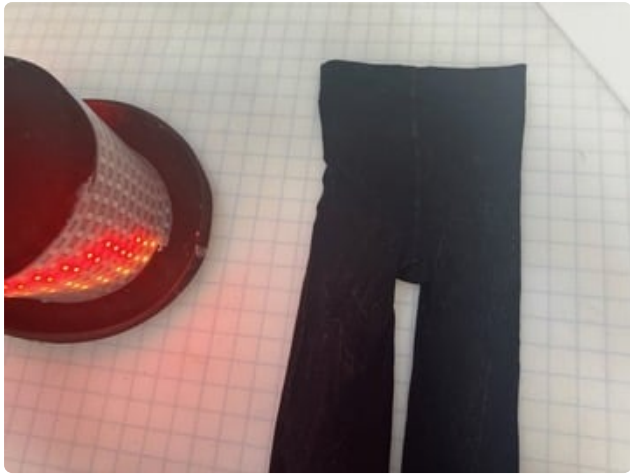
Plug everything in and verify it's all working. Once I was happy with placement, I glued the electronics securely to the "roof" of the hat. For the battery, I used velcro to stick it in place so it doesn't slide around, but so that I can remove it easily for charging.



To finish the enclosure I cut a circle of 1/2" EVA foam so that it fits snugly into the top of the hat. It covers everything except for the on/off switch, which I glued securely to the side of the hat about an inch and a half from the top. The EVA foam covers all the electronics but fits behind / above the switch, so it's easy to turn on and off.

Diffusion & Decoration

I tried a few different materials fabrics for diffusion. There are a lot of options - check out our tutorial on [NeoPixel Diffusion](https://adafru.it/19f9) (<https://adafru.it/19f9>) for some ideas.



I used the upper portion of a pair of [black opaque tights](https://adafru.it/1a58) (<https://adafru.it/1a58>). They are already sewn into a circle and will stretch to cover the sides of the hat perfectly.



I cut off the legs and used some [FrayCheck](https://adafru.it/1a58) (<https://adafru.it/1a58>) to seal the cut-off end and keep the tights from getting runs when I stretched it. I also used my serger and sewing machine to make a narrow hem along the cut edge.



I glued the waistband of the tights to the top and bottom brim of the hat using E6000 glue.



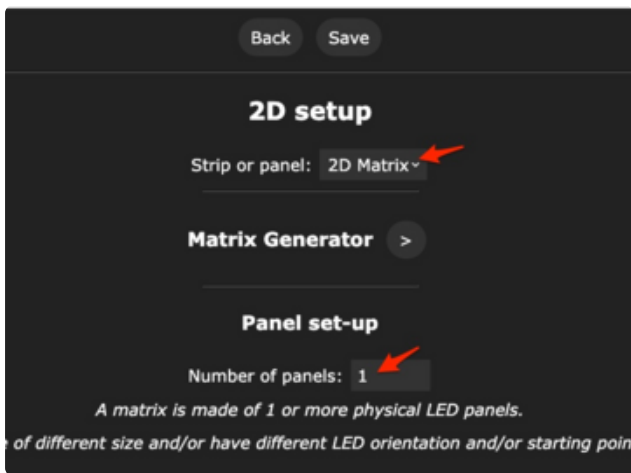
I added some lace to cover the transitions so the edges of the LED strip is less obvious. For the button, I attached it to the outside of the hat and then sewed / glued a 3d printed Auryn decoration over it, placing it so that pressing on the Auryn activates the button.

WLED 2d Matrix

Matrix Setup

WLED has a handy 2d-matrix setup feature that will take care of the mapping for you on a rectangular matrix. Our matrix is not quite rectangular, but we can pretend it is by "unwrapping" it with our minds. Let's pretend we cut the LEDs in a straight line up the back of the hat, and then laid all the strip flat on the table. (Don't do this for real!) Now we have a rectangle we can use as a matrix.

Go back to Config and choose the **2D Configuration** tab.



Change the dropdown to read 2d Matrix, and additional options will appear. If you want to sync more than one panel, you can do it here.



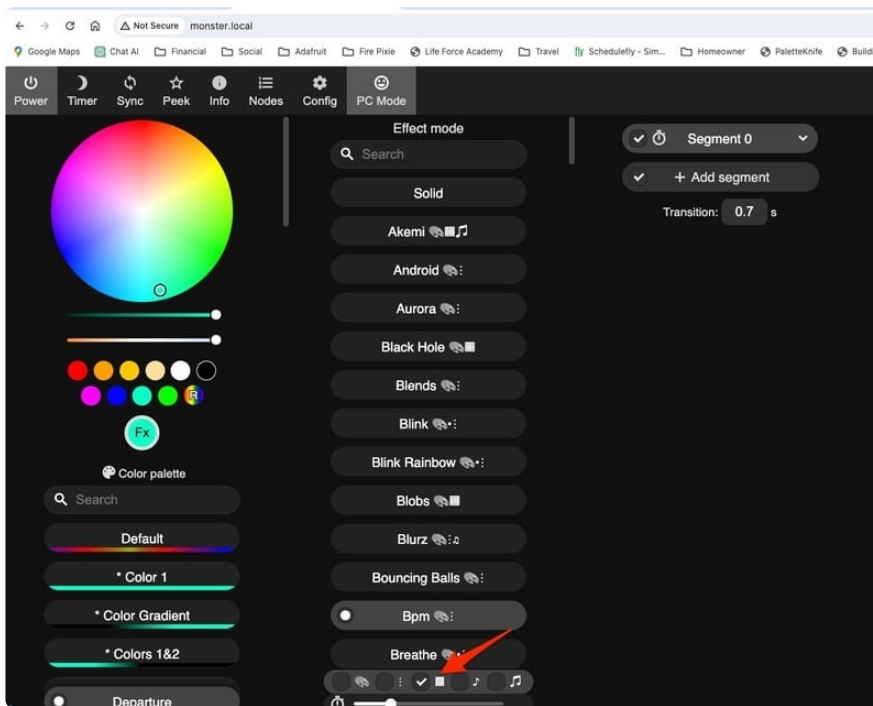
Set up your layout numbers to match the number of rows and columns in your project. You can also change orientation here - my pixels start in the lower left corner and finish in the upper right.

Count the number of pixels around the hat in one revolution to get your horizontal number, and then count the number of strips stacked up to get your vertical number.

If the effects don't look right, try adjusting these numbers until your animations "snap" into view. When the mapping is correct, it's like your project suddenly comes into focus.

Matrix Effects

WLED has a hefty number of matrix effects that appear in the list when you've got your 2d matrix set up. Click the little square in the "filters" popup to see only the 2d matrix effects. Many of them can be customized with speed and intensity sliders, or different color palettes. Go wild!



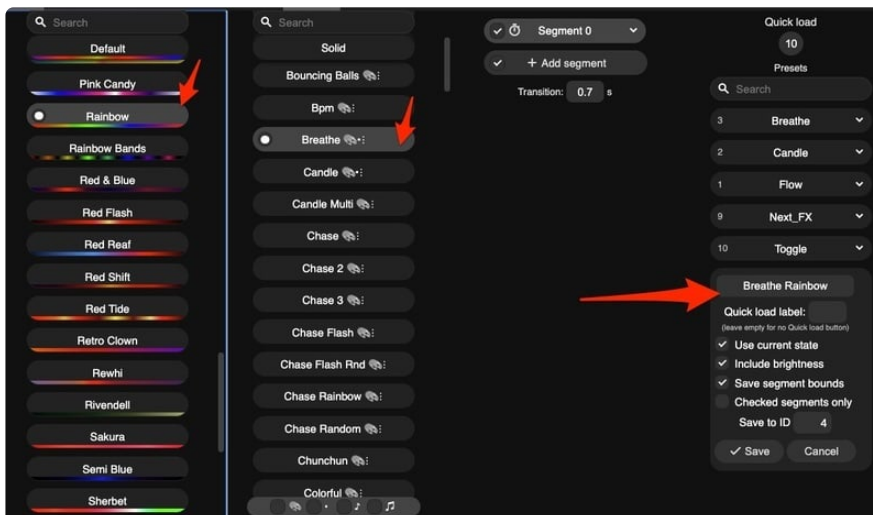
Create Presets

Animation Presets

Once your lights are working, it's time for the fun part: creating light animations using WLED presets. This is easy and intuitive in WLED. Choose a color palette and an effect, then play with the sliders at the bottom of the Effect Mode column to customize the effect.

When you find something you like, click the +Preset button in the Presets tab and give your preset a name, then save it. You can reorder them using the "Save to ID" field.

You can create hundreds of presets using the preprogrammed effects, so take some time to play around and see what looks good on your LED strip.



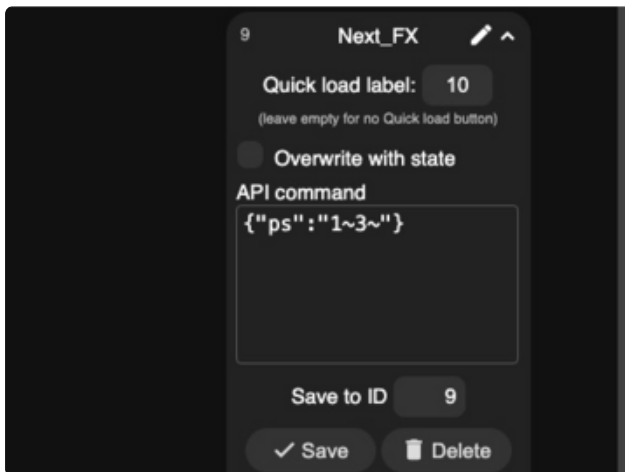
Create at least 3 presets, and be sure they are saved to ID numbers 1-3. Our next step will be to set up the switch to toggle between these presets.

Control Presets

The WLED preset interface can also be used to send control commands to the LED strip. Once you've set up a button or switch, it can be used for a wide variety of control functions: change modes, change brightness, toggle the lights on and off, or a whole host of other features.

Each button function in WLED has 3 options: a short press, a long press, or a double press. I want my lights to cycle through presets with a short press, and to turn on or off with a long press. Here's how I set up a control preset for each of these features.

Cycle through Presets

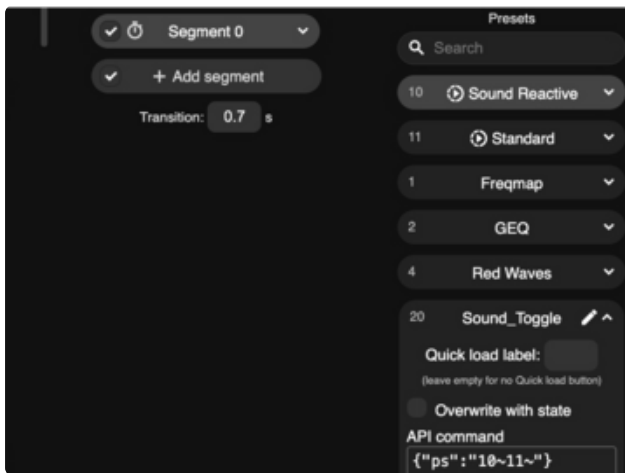


Click +Preset and then uncheck the "Use current state" checkbox. This will open up an API command window within the preset module.

Call the effect "Next_FX" and type {"ps": "1~3~"} into the API command box. This tells WLED to cycle through presets 1-3. If you'd like to cycle through more presets, change the 3 to the number of presets you have.

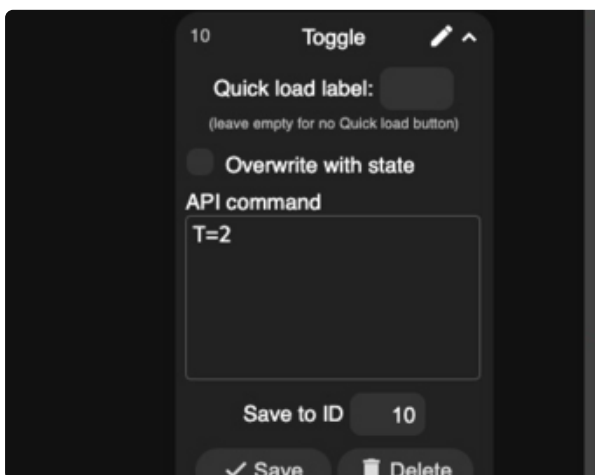
Be sure your preset IDs include all the numbers -- skipping a number will break this command.

Give your preset an ID number that's above the range of your preset numbers -- I called mine 9.



It's also possible to cycle between playlists. I made a playlist of sound reactive effects and a separate playlist of "standard" effects. I set up a preset that toggles between the two playlists -- effectively turning "sound reactive" mode on and off.

Toggle On/Off



Create another preset and call it "Toggle". Uncheck the "use current state" checkbox and enter **T=2** into the API command box. This will tell the LEDs to toggle between on and off. Save this one to ID 10.

On the next page we'll connect these control presets to our button.

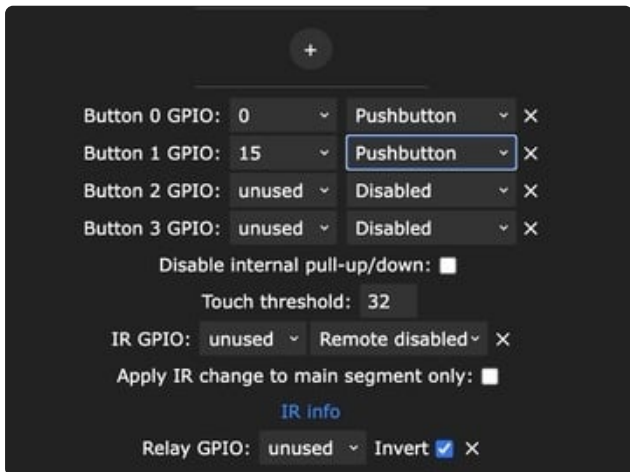
Find out more about creating these control presets here:

<https://kno.wled.ge/features/presets/> (<https://adafru.it/1a4F>)

You can enter either JSON commands or HTTP commands into this command box, giving you a lot of control over your WLED project.

<https://kno.wled.ge/interfaces/json-api/> (<https://adafru.it/1a4G>)

WLED Button Setup

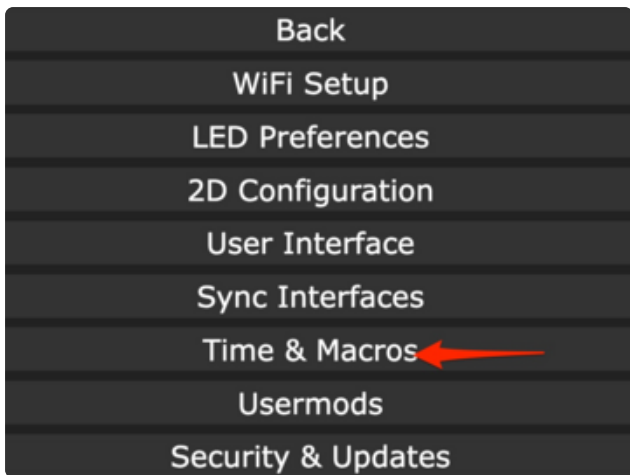


I soldered my button touch wire to pin **A3**, which is also known as **GPIO 15** (see the [QT Py Pico pinout diagram \(https://adafruit.it/11dK\)](https://adafruit.it/11dK), it's the number in yellow).

In the LED Settings screen, scroll to the button setup section and set button 1 to use pin 15, with the type set as "Pushbutton".

I advise against using button 0. WLED's buttons have some default behaviors written in, and one of button 0's default behaviors is that when it's pressed for more than a few seconds, it resets your microcontroller to factory settings. I originally had button 0 selected, and I held the button down a bit too long while I was sewing it onto the hat, and .. POOF. All my settings and presets were wiped out.

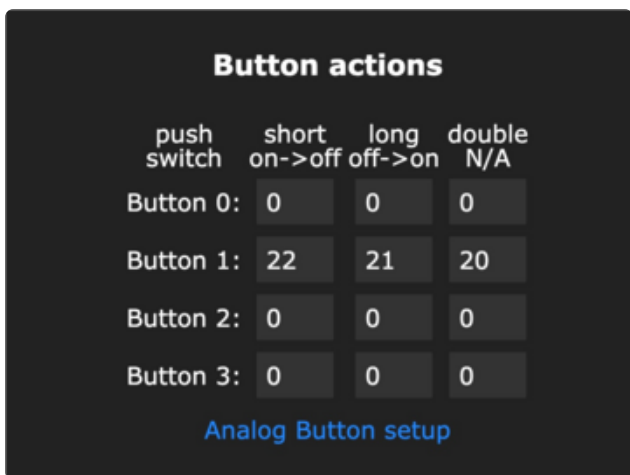
Check out the "Backup" section under the Additional Settings page to learn to back up your configuration and presets, in case this kind of thing happens to you.



Next, head to the **Time & Macros** config screen. Scroll down to the Button actions area.

For button 1, enter the number assigned to the two control presets you made on the last page. My Next_FX preset is number 22, so I entered 22 under "short press", and my Toggle preset is number 21, so I've got that set up as a "long press".

I also added my playlist toggle to preset 20, so a double-click of my button will turn sound reactive mode on or off.



Troubleshooting

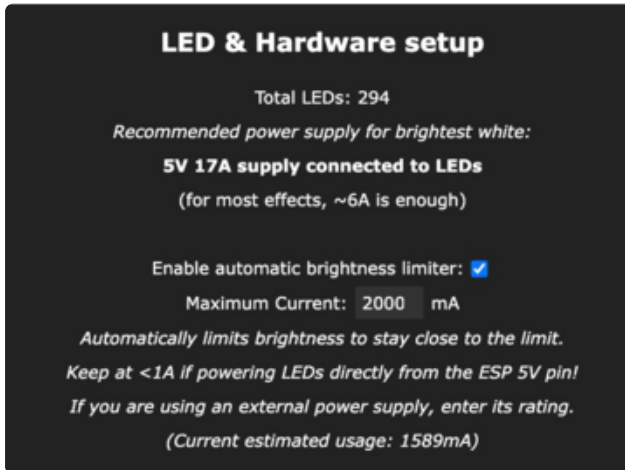
If your button isn't working, here are some things to try:

1. Double check both the LED preferences page and the Time & Macros page to be sure your settings are correct, and have saved.
2. Be sure your presets are correctly numbered. WLED gets confused if the presets have non-sequential IDs (so make sure they're numbered 1, 2, 3 rather than 2, 5, 7).
3. Try soldering to a different GPIO pin, and make sure to update the pin number in LED Preferences. I originally soldered to A1, but there was a short or some interference from A0 so giving a little space between the soldered wires on the board seemed to help.

WLED Additional Settings

Brightness Limiter

Find this on the LED Settings screen.

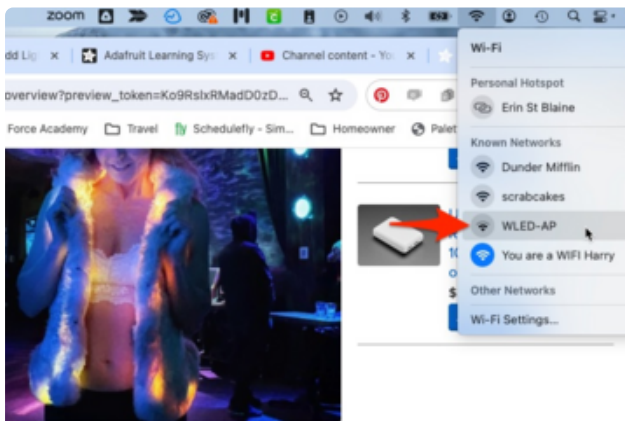


WLED automatically limits the brightness of your project so your power supply doesn't get over-taxed. The default amperage is 850mA, which is a little low for most power supplies.

For some projects, especially battery-powered projects, having the limiter turned on is a good idea. But if you're not getting the brightness you expect, try adjusting this number to match the amperage of your power supply or battery pack.

Access Point (AP) Mode

While you're home, it's easy to control your project over your local WiFi network. But when you're out at a festival you probably don't have WiFi access. It's still possible to connect to your project and control it using WLED's Access Point Mode.

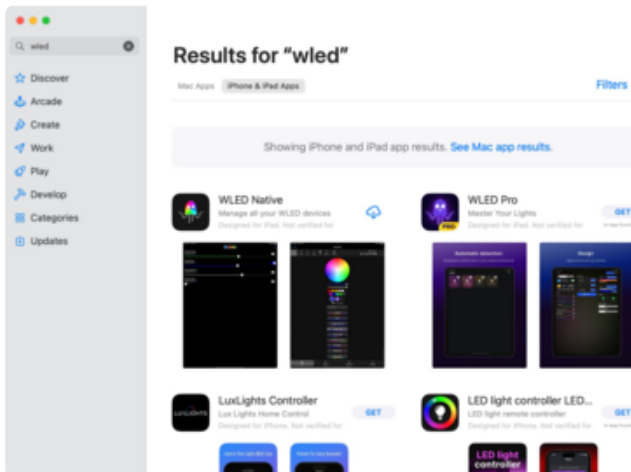


Turn your project on and give it a minute or two to start broadcasting. Look in your WiFi networks and find **WLED-AP** - this is a mini WiFi network being broadcast by the Feather. Connect to it - the default password is "wled123". An instance of WLED will automatically pop up and you can control your project from anywhere.

If you're putting your lights up in public, it's a good idea to change the AP Mode default password so strangers can't log in and control your lights. This could be a security risk.

AP Mode only broadcasts for a few minutes after you boot up the board so if you don't see the WLED-AP network try rebooting.

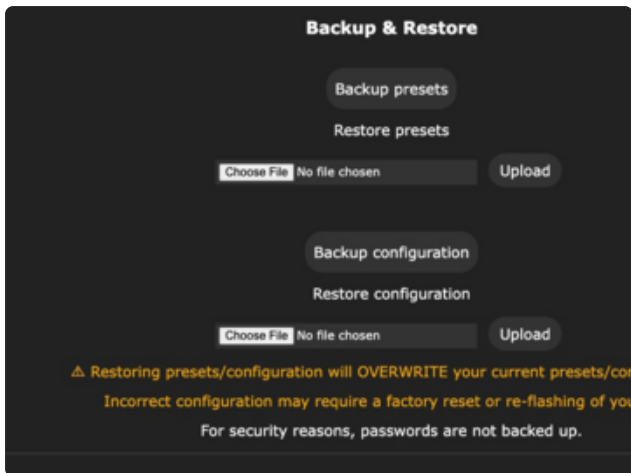
WLED App



There are a couple different apps available to manage your WLED projects. Name and organize your projects, and find them quickly without having to type in a URL. Check the Apple or Android store for downloads.

My favorite is "WLED Native". It allows you to organize multiple instances and easily switch between devices without having to remember any URLs.

Backup Config & Presets



Under Config / Security & Updates you will find a place to back up your data. It's a good idea to back up your config file as soon as you're happy with the settings. Save it as a .json file on your computer. Now you can prototype and experiment to your heart's content, and if everything breaks, just re-upload this file. Or, if you're doing another build you can use this feature to copy all your settings from one board to another.