Halloween Neon LED Signs

Created by Ruiz Brothers

https://learn.adafruit.com/halloween-neon-led-signs

Last updated on 2024-06-03 03:43:06 PM EDT

Table of Contents

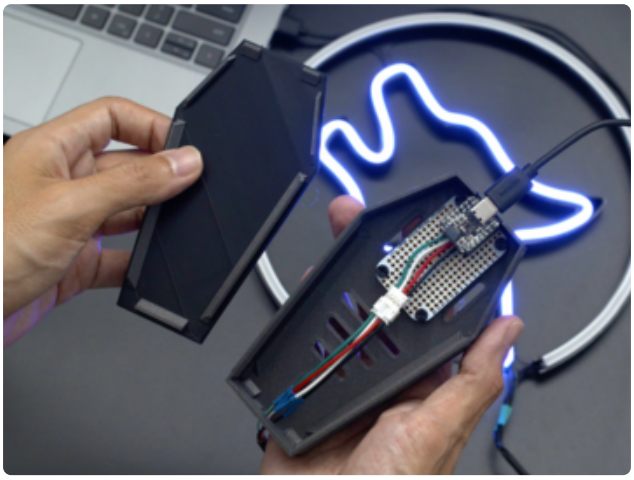
Overview

3D print your own spooky LED neon signs for Halloween!

Make a hipster ghost wearing shades, a grim reaper holding a scythe or a werewolf howling at the moon!

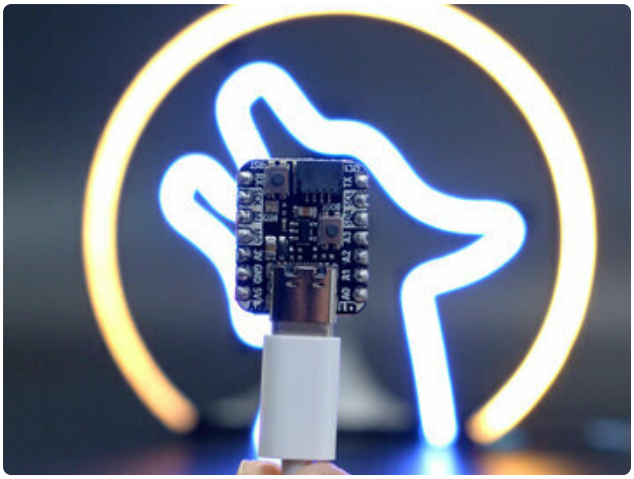
Bring life to your NeoPixels with LED animations using CircuitPython.

Use multiple NeoPixel strips to create elements with spooky animated features.



This 3D printed coffin case snap fits together and houses a small perma-proto board.

Perfect for letting your dev board slumber in peace when it's time to lay it to rest.



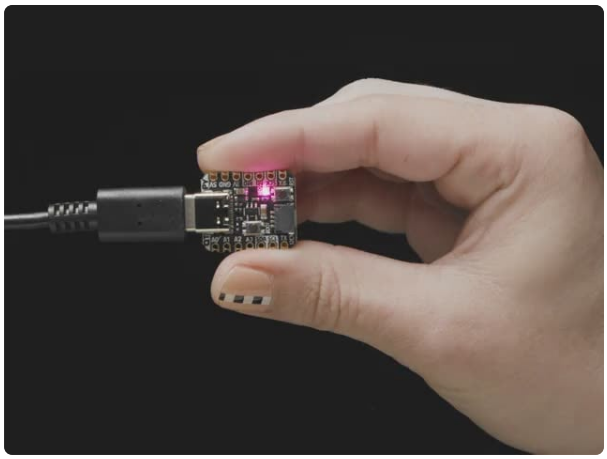
Powered by an Adafruit QT Py RP2040 microcontroller board, this little devil packs lots of treats with features that are sure to fright and delight.

Prerequisite Guides

Take a moment to review the following guides:

- CircuitPython LED Animations (https://adafru.it/LZF)
- QT Py RP2040 (https://adafru.it/RWD)

Parts



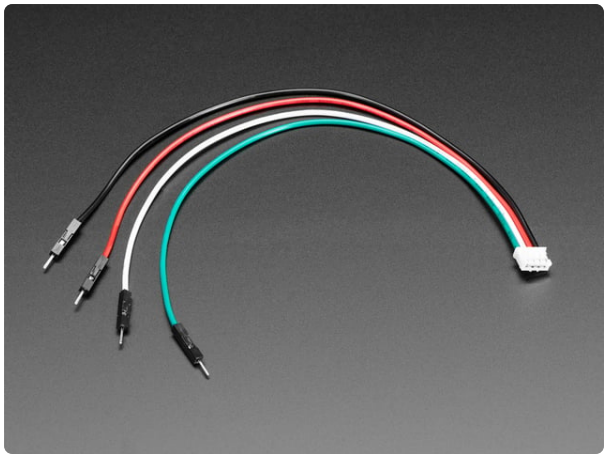
Adafruit QT Py RP2040
What a cutie pie! Or is it... a QT Py? This diminutive dev board comes with one of our new favorite chip, the RP2040. It's been made famous in the new https://www.adafruit.com/product/4900

Flexible Silicone Neon-like Skinny NeoPixel LED Strip

You love NeoPixels, and you love silicone diffusion? Peep this Flexible Silicone Neon-like Skinny NeoPixel LED Strip! OK it's a bit of a mouthful, but check...
https://www.adafruit.com/product/4310

Adafruit Perma-Proto Small Mint Tin Size Breadboard PCB - 3 pack

Making a project that will fit into an "Altoids Smalls" Mint Tin? Or maybe you just need a small amount of prototyping space and a larger breadboard size is too big? Put down...
https://www.adafruit.com/product/1214



JST PH 2mm 4-Pin to Male Header Cable - I2C STEMMA Cable - 200mm

This cable will let you turn a JST PH 4-pin cable port into 4 individual wires with high-quality 0.1" male header plugs on the end. We're carrying these to match up with any...
https://www.adafruit.com/product/3955

USB Extension Cable - 3 meters / 10 ft long

This handy USB extension cable will make it easy for you to extend your USB cable when it won't reach. The connectors are gold plated for years of reliability. We use these handy...
https://www.adafruit.com/product/993

Black – M3 Thread

1 x Nylon Machine Screw and Stand-off Set
Black – M3 Thread
https://www.adafruit.com/product/4685

1 x Ribbon Cable
Silicone Cover Stranded-Core - 10 Wire 1 Meter Long - 28AWG Black
https://www.adafruit.com/product/3890

1 x Female Header
36-pin 0.1" Short - Pack of 5
https://www.adafruit.com/product/3008

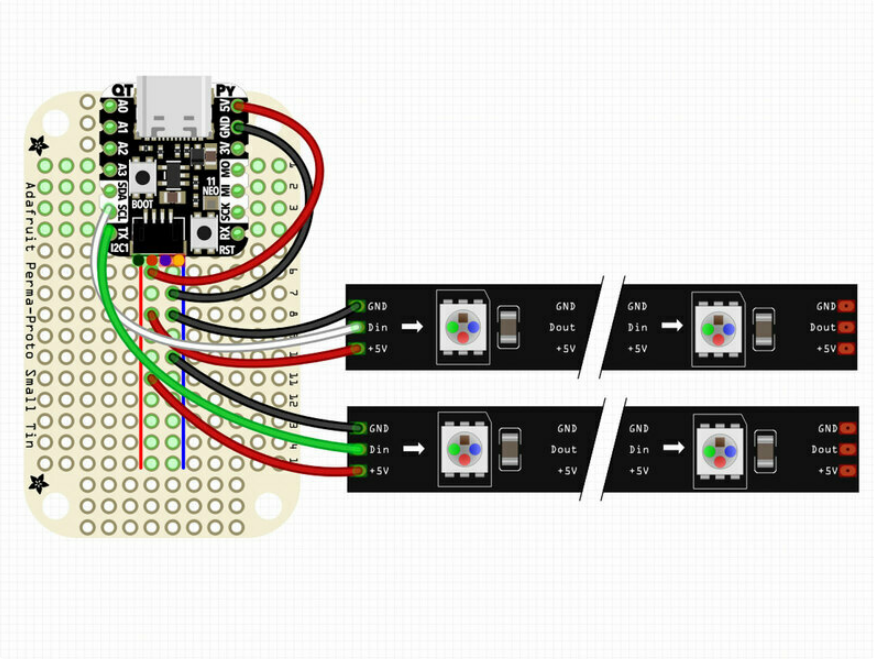
1 x JST Cable
JST PH 2mm 4-Pin Socket to Color Coded Cable - 200mm
https://www.adafruit.com/product/4045

Circuit Diagram

The diagram below provides a visual reference for wiring of the components. This diagram was created using the software package Fritzing (https://adafru.it/oEP).

Adafruit Library for Fritzing

Use Adafruit's Fritzing parts library to create circuit diagrams for your projects. Download the library or just grab individual parts. Get the library and parts from GitHub - Adafruit Fritzing Parts (https://adafru.it/AYZ).



Wired Connections

The QT Py is secured to the small Perma-Proto using socket headers.

Strip A

- 5V to 5V pin on Perma-Proto
- GND to GND pin on Perma-Proto
- Din to SCL pin on Perma-Proto

Strip B

- 5V to 5V pin on Perma-Proto
- GND to GND pin on Perma-Proto
- Din to SDA pin on Perma-Proto

CircuitPython

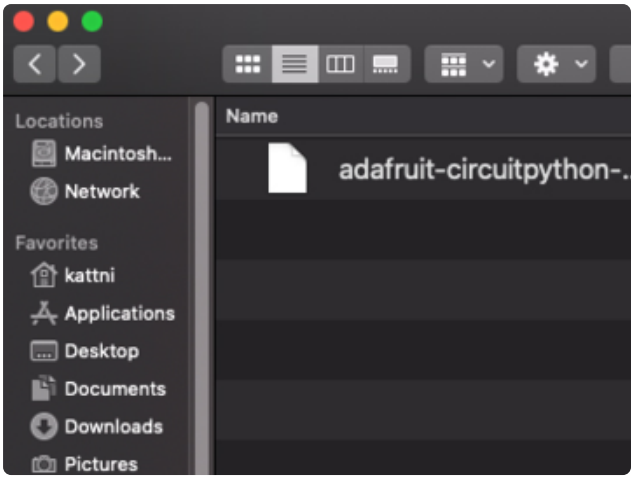
CircuitPython (https://adafru.it/tB7) is a derivative of MicroPython (https://adafru.it/BeZ) designed to simplify experimentation and education on low-cost microcontrollers. It makes it easier than ever to get prototyping by requiring no upfront desktop software downloads. Simply copy and edit files on the **CIRCUITPY** drive to iterate.

CircuitPython Quickstart

Follow this step-by-step to quickly get CircuitPython running on your board.

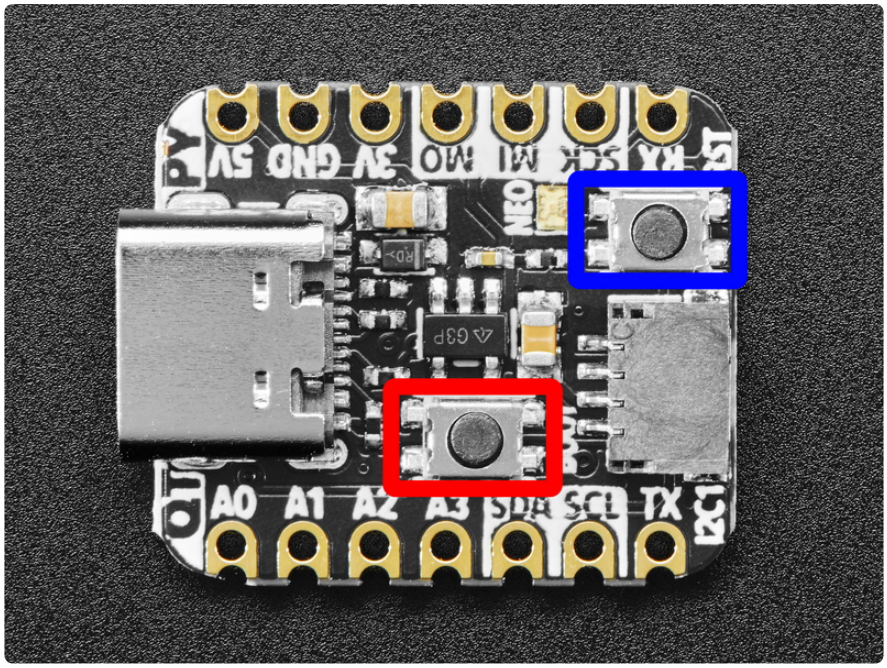
Download the latest version of CircuitPython for this board via circuitpython.org

https://adafru.it/RLD



Click the link above to download the latest CircuitPython UF2 file.

Save it wherever is convenient for you.

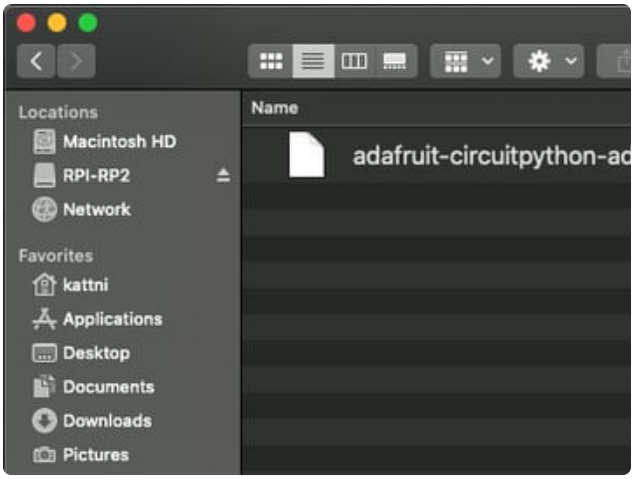


To enter the bootloader, hold down the **BOOT/BOOTSEL button** (highlighted in red above), and while continuing to hold it (don't let go!), press and release the **reset button** (highlighted in blue above). **Continue to hold the BOOT/BOOTSEL button until the RPI-RP2 drive appears!**

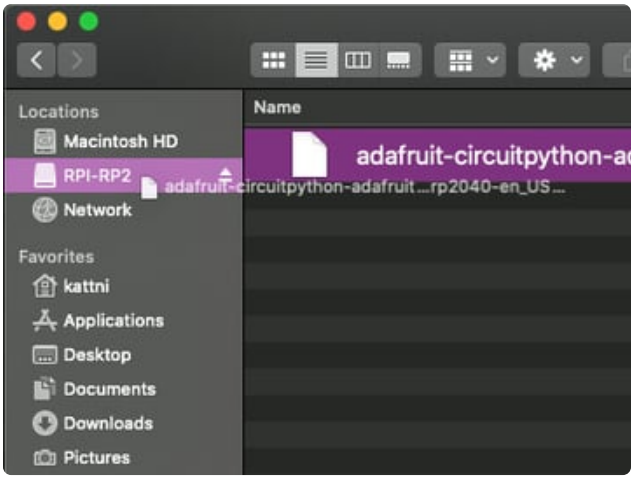
If the drive does not appear, release all the buttons, and then repeat the process above.

You can also start with your board unplugged from USB, press and hold the BOOTSEL button (highlighted in red above), continue to hold it while plugging it into USB, and wait for the drive to appear before releasing the button.

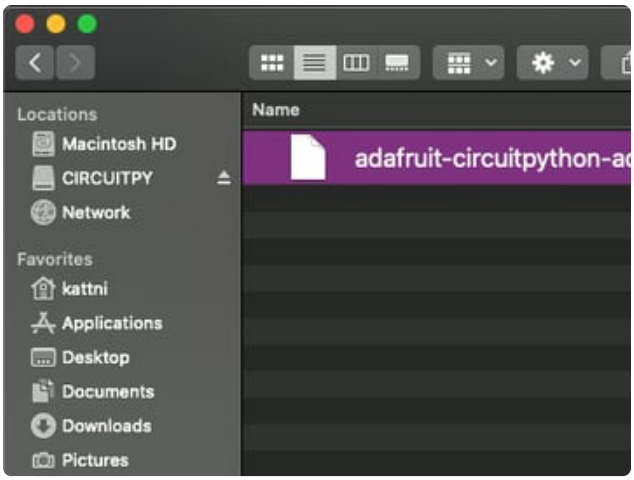
A lot of people end up using charge-only USB cables and it is very frustrating! **Make sure you have a USB cable you know is good for data sync.**



You will see a new disk drive appear called **RPI-RP2**.



Drag the **adafruit_circuitpython_etc.uf2** file to **RPI-RP2.**



The **RPI-RP2** drive will disappear and a new disk drive called **CIRCUITPY** will appear.

That's it, you're done! :)

Safe Mode

You want to edit your **code.py** or modify the files on your **CIRCUITPY** drive, but find that you can't. Perhaps your board has gotten into a state where **CIRCUITPY** is read-only. You may have turned off the **CIRCUITPY** drive altogether. Whatever the reason, safe mode can help.

Safe mode in CircuitPython does not run any user code on startup, and disables auto-reload. This means a few things. First, safe mode bypasses any code in **boot.py** (where you can set **CIRCUITPY** read-only or turn it off completely). Second, it does not run the code in **code.py**. And finally, it does not automatically soft-reload when data is written to the **CIRCUITPY** drive.

Therefore, whatever you may have done to put your board in a non-interactive state, safe mode gives you the opportunity to correct it without losing all of the data on the **CIRCUITPY** drive.

Entering Safe Mode

To enter safe mode when using CircuitPython, plug in your board or hit reset (highlighted in red above). Immediately after the board starts up or resets, it waits 1000ms. On some boards, the onboard status LED (highlighted in green above) will blink yellow during that time. If you press reset during that 1000ms, the board will start up in safe mode. It can be difficult to react to the yellow LED, so you may want to think of it simply as a slow double click of the reset button. (Remember, a fast double click of reset enters the bootloader.)

In Safe Mode

If you successfully enter safe mode on CircuitPython, the LED will intermittently blink yellow three times.

If you connect to the serial console, you'll find the following message.

```
Auto-reload is off.
Running in safe mode! Not running saved code.

CircuitPython is in safe mode because you pressed the reset button during boot.
Press again to exit safe mode.

Press any key to enter the REPL. Use CTRL-D to reload.
```

You can now edit the contents of the **CIRCUITPY** drive. Remember, your code will not run until you press the reset button, or unplug and plug in your board, to get out of safe mode.

Flash Resetting UF2

If your board ever gets into a really weird state and CIRCUITPY doesn't show up as a disk drive after installing CircuitPython, try loading this 'nuke' UF2 to RPI-RP2. which

will do a 'deep clean' on your Flash Memory. **You will lose all the files on the board**, but at least you'll be able to revive it! After loading this UF2, follow the steps above to re-install CircuitPython.

Download flash erasing "nuke" UF2

https://adafru.it/RLE

Coding Neon Signs

Once you've finished setting up your QT Py RP2040 with CircuitPython, you can access the code and necessary libraries by downloading the Project Bundle.

To do this, click on the **Download Project Bundle** button in the window below. It will download as a zipped folder.

```
# SPDX-FileCopyrightText: 2022 Noe Ruiz for Adafruit Industries
# SPDX-License-Identifier: MIT
# Ghost with Glasses Neon Sign
import board
import neopixel
from adafruit_led_animation.animation.blink import Blink
from adafruit_led_animation.animation.comet import Comet
from adafruit_led_animation.animation.chase import Chase
from adafruit_led_animation.animation.pulse import Pulse
from adafruit_led_animation.group import AnimationGroup
from adafruit_led_animation.sequence import AnimationSequence
from adafruit_led_animation import color

ghost_pixels = neopixel.NeoPixel(board.SDA, 90, brightness=0.5,
    auto_write=False, pixel_order=neopixel.RGB)
glasses_pixels = neopixel.NeoPixel(board.SCL, 33, brightness=0.5,
    auto_write=False, pixel_order=neopixel.RGB)

animations = AnimationSequence(
    # Synchronized animations
    AnimationGroup(
        Chase(ghost_pixels,speed=0.02, color=color.CYAN, size=40, spacing=5),
        Blink(glasses_pixels, speed=.4, color=color.PURPLE),
        sync=False,
```

```
    ),

    # Sequential animations
    Pulse(glasses_pixels, speed=0.01, color=color.WHITE, period=2),

    # Synchronized
    AnimationGroup(
        Pulse(glasses_pixels, speed=0.01, color=color.PURPLE, period=1),
        Comet(ghost_pixels, speed=0.01, color=color.CYAN, tail_length=50,
bounce=False),
        sync=True,
    ),

    advance_interval=4.0,
    auto_clear=True,
    auto_reset=True,
)

while True:
    animations.animate()
```

```
# SPDX-FileCopyrightText: 2022 Noe Ruiz for Adafruit Industries
# SPDX-License-Identifier: MIT
# Grim Reaper and Moon Neon Sign
import board
import neopixel
from adafruit_led_animation.animation.chase import Chase
from adafruit_led_animation.animation.pulse import Pulse
from adafruit_led_animation.group import AnimationGroup
from adafruit_led_animation.sequence import AnimationSequence
from adafruit_led_animation import color

knife_leds = neopixel.NeoPixel(board.SDA, 48, brightness=0.8,
    auto_write=False, pixel_order=neopixel.RGB)
repear_leds = neopixel.NeoPixel(board.SCL, 60, brightness=0.8,
    auto_write=False, pixel_order=neopixel.RGB)

animations = AnimationSequence(

    AnimationGroup(
        Chase(knife_leds, speed=0.02, color=color.PURPLE, spacing=12, size=40),
        Pulse(repear_leds, speed=0.01, color=color.GREEN, period=3),
        sync=True,
    ),
    advance_interval=8.0,
    auto_clear=True,
    auto_reset=True,
)

while True:
    animations.animate()
```

```
# SPDX-FileCopyrightText: 2022 Noe Ruiz for Adafruit Industries
# SPDX-License-Identifier: MIT
# Werewolf and Moon Neon Sign
import board
import neopixel
from adafruit_led_animation.animation.blink import Blink
from adafruit_led_animation.animation.comet import Comet
from adafruit_led_animation.animation.pulse import Pulse
from adafruit_led_animation.group import AnimationGroup
from adafruit_led_animation.sequence import AnimationSequence
from adafruit_led_animation import color

moon_leds = neopixel.NeoPixel(board.SDA, 60, brightness=0.8,
    auto_write=False, pixel_order=neopixel.RGB)
```

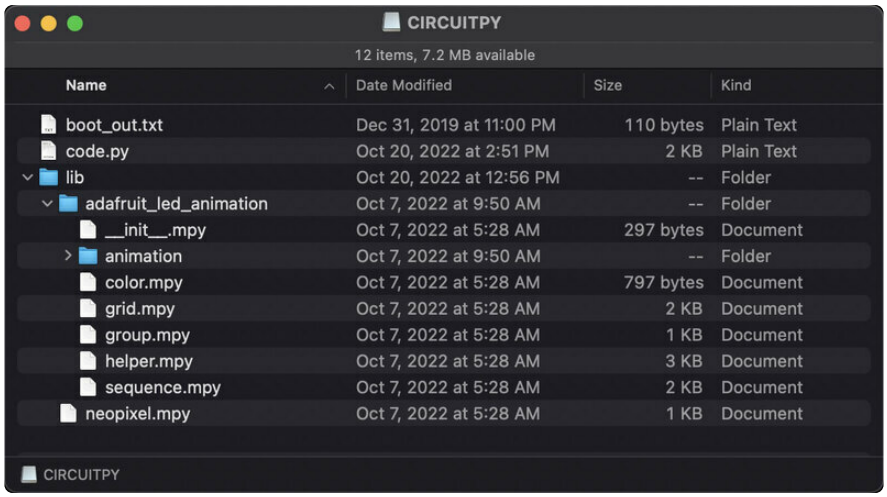
```
wolf_leds = neopixel.NeoPixel(board.SCL, 57, brightness=0.8,
    auto_write=False, pixel_order=neopixel.RGB)

animations = AnimationSequence(
    Blink(wolf_leds, speed=0.07, color=color.BLUE),
    Pulse(wolf_leds, speed=0.01, color=color.PURPLE, period=3),
    AnimationGroup(
        Pulse(wolf_leds, speed=0.01, color=color.PURPLE, period=3),
        Comet(moon_leds, speed=0.01, color=color.AMBER, tail_length=60,
reverse=True),
        sync=True,
    ),
    AnimationGroup(
        Pulse(wolf_leds, speed=0.01, color=color.PURPLE, period=3),
        Pulse(moon_leds, speed=0.01, color=color.AMBER, period=3),
        sync=True,
    ),
    AnimationGroup(
        Pulse(wolf_leds, speed=0.01, color=color.PURPLE, period=3),
        Pulse(moon_leds, speed=0.01, color=color.AMBER, period=3),
        sync=True,
    ),
    advance_interval=2.0,
    auto_clear=True,
    auto_reset=True,
)

while True:
    animations.animate()
```

Upload the Code and Libraries to the QT Py RP2040

After downloading the Project Bundle, plug your QT Py into the computer's USB port with a known good USB data+power cable. You should see a new flash drive appear in the computer's File Explorer or Finder (depending on your operating system) called **CIRCUITPY**. Unzip the folder and copy the following items to the QT Py RP20402's **CIRCUITPY** drive.

- **code.py**
- **lib**

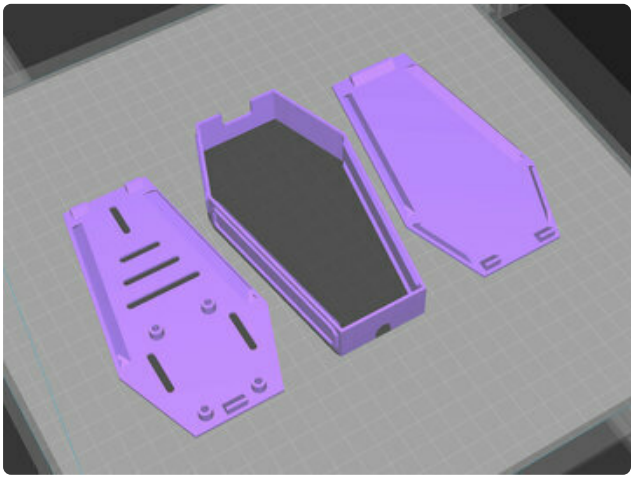
Your QT Py RP2040 **CIRCUITPY** drive should look like this after copying the **lib** folder and the **code.py** file.



3D Printing

CAD Assembly

The QT Py is mounted to the Perma-Proto with socket headers. The Perma-Proto is secured to the bottom cover using M3 screws and hex nuts. The top and bottom cover snap fits over the enclosure frame.



CAD Parts List

STL files for 3D printing are oriented to print "as-is" on FDM style machines. Parts are designed to 3D print without any support material. Original design source may be downloaded using the links below:

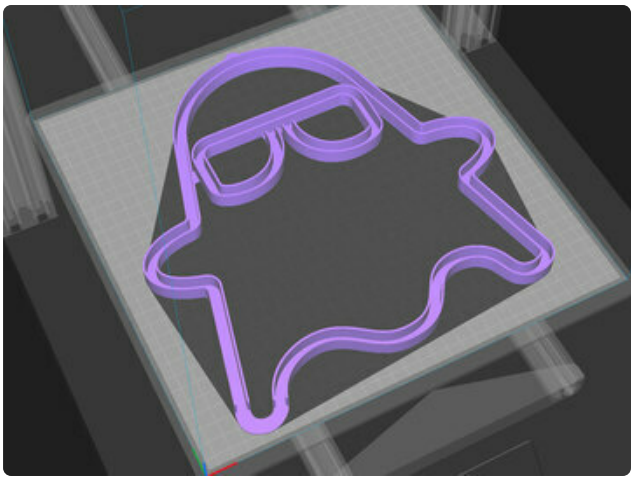
coffin-top
coffin-bottom
coffin-frame
Wolf Sign
Ghost Sign
Reaper Sign

Download STLs.zip

https://adafru.it/11Cq

Download CAD Source Files

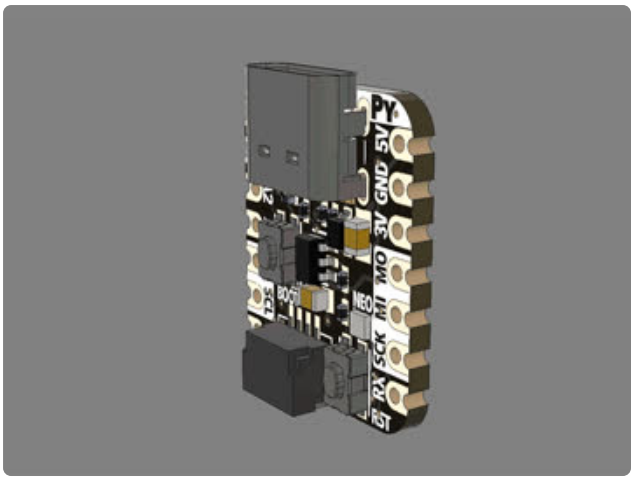
https://adafru.it/11Cr



Build Volume

The parts require a 3D printer with a minimum build volume.

300mm (X) x 300mm (Y) x 50mm (Z)

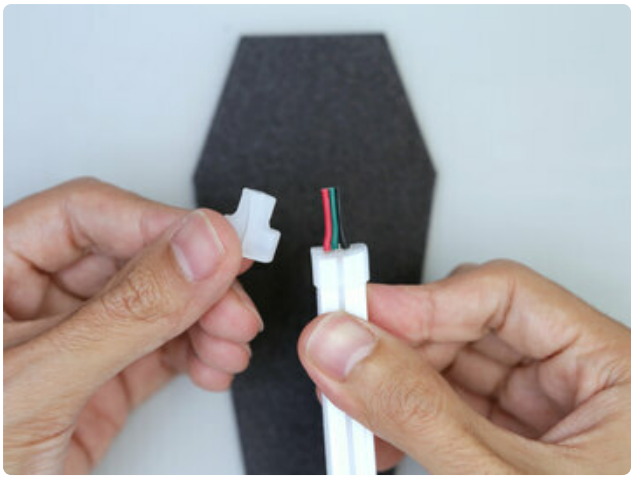


Design Source Files

The project assembly was designed in Fusion 360. This can be downloaded in different formats like STEP, STL and more. Electronic components like Adafruit's boards, displays, connectors and more can be downloaded from the Adafruit CAD parts GitHub Repo (https://adafru.it/AW8).

Wiring

Be very careful handling hobby knives. Wear safety glasses and use adult supervision and gloves.



Prepare Strips

Remove the end caps from both ends of the NeoPixel strip.

Use a hobby knife to cut through the silicone and carefully remove the end caps and bits of hot glue.

Remove Wires

Use a soldering iron to desolder the stock wires from both ends of the NeoPixel strip.

Use a hobby knife to cut an opening to better reach the solder pads.

Install Strips

Begin press fitting the NeoPixel strip into the sign.

Gauge how long each section of the sign will need to be used.

Plan a good location for the NeoPixel strips to start and end.

Locate Strip Cutoff Sections

With the NeoPixel strip installed, locate where in the strip that needs to be cut.

Cut NeoPixel Strip Sections

Use a hobby knife to cut the silicone sheathing.

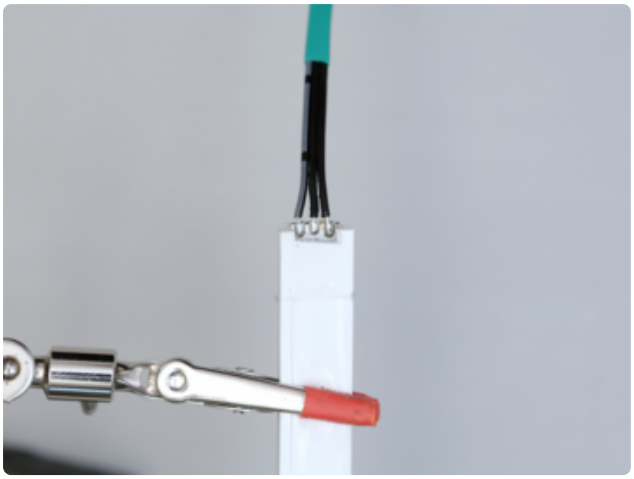
Locate Solder Pads

Reveal the solder pads in the strip. If needed, use the knobby knife to cut a larger opening.

Use flush diagonal cutters to cut off the section of NeoPixel strip.

Install Second Strip

Repeat the process to create the second section of the NeoPixel strip.



Solder Wires

Locate the end of the strip with the DATA IN pad.

Create a 3-pin wire using the silicone cover stranded ribbon cable.

Solder the 3-pin wire to the three pads on the NeoPixel strip.

Wired Strips

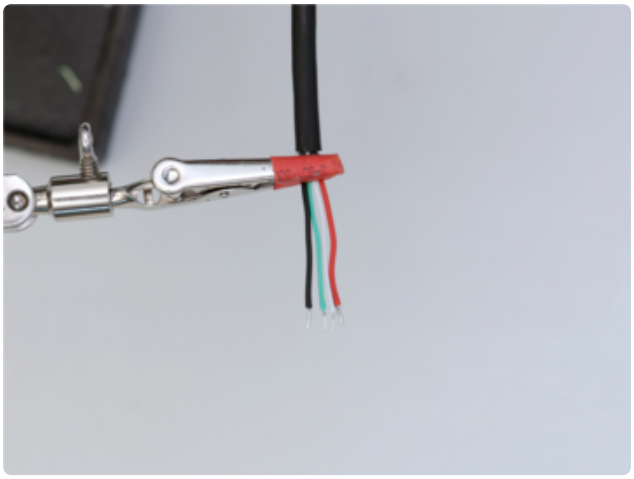
Create a second set of silicone wires for the other NeoPixel strip.

Solder the 3-pin wire to the second NeoPixel strip.

Prep USB Extension Cable

Remove the connectors from USB extension cable.

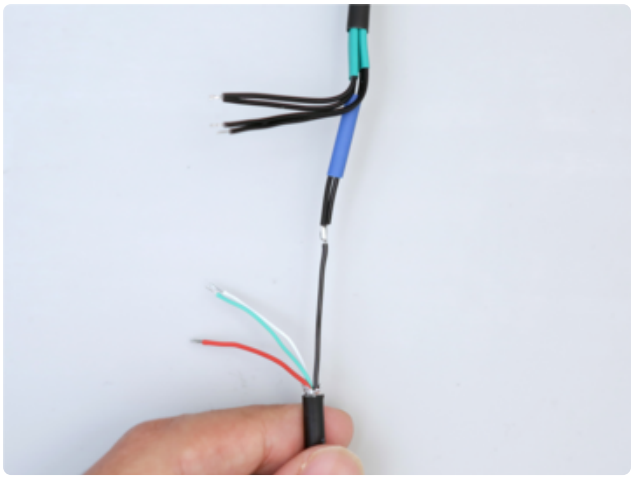
Use a hobby knife or wire strippers to remove an inch (25mm) of sheathing from the cable on both ends.

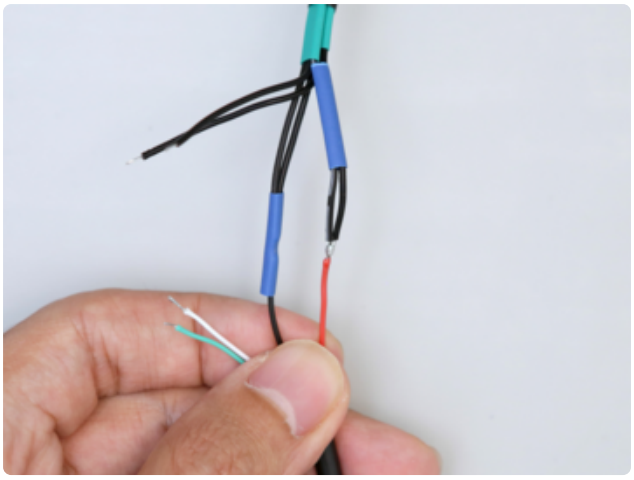


Tin Wires

Use the wire strippers to remove a small bit of insulation from each of the wires.

Use a soldering iron to tin each wire with a bit of solder.



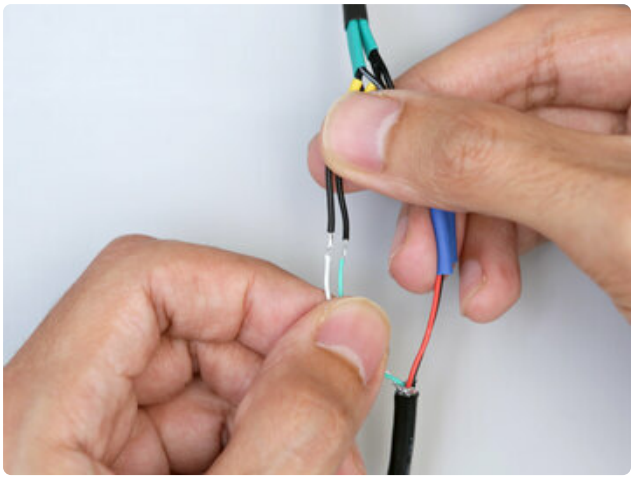


Solder Voltage and Ground

Use the soldering iron and solder to attach the black wire from the extension cable to both ground wires from the NeoPixel strips.

Solder the red wire from the extension cable to the 5V wires on the NeoPixel strips.

Use heat shrink tubing to insulate the exposed wired connections.



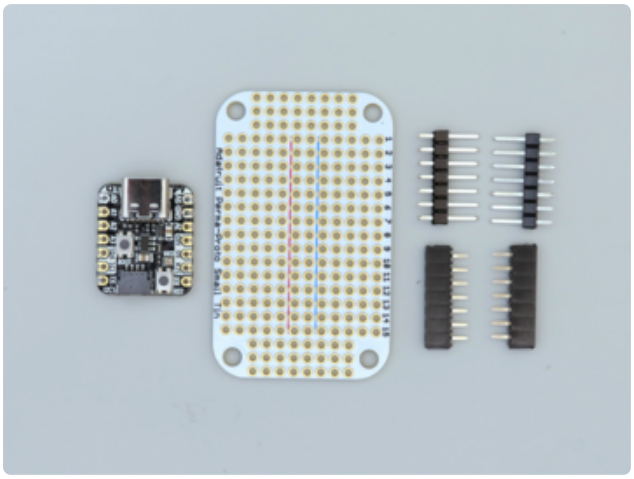
Connect Data Wires

Solder the data in wire from the first NeoPixel strip to the white wire on the extension cable.

Solder the second data in wire rom the other NeoPixel strip to the green wire on the extension cable.

Wired Cable

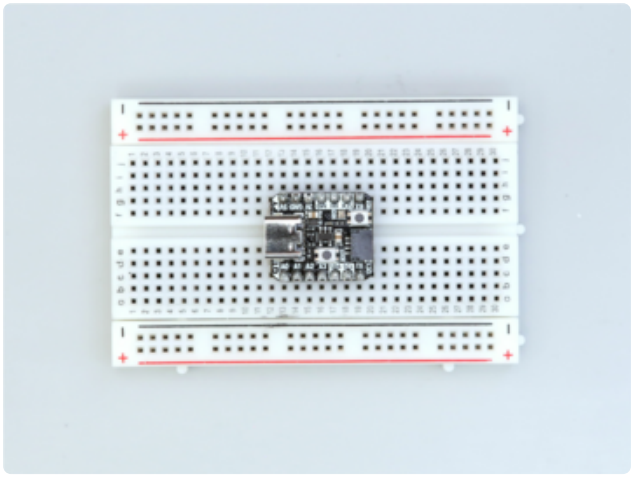
Double check the wires have been properly soldered to the corresponding pads on both NeoPixel strips.



Headers for QT Py

Use the following headers to secure the QT Py onto the Perma Proto board.

2x 1x7 male header pins
2x 1x7 short socket headers

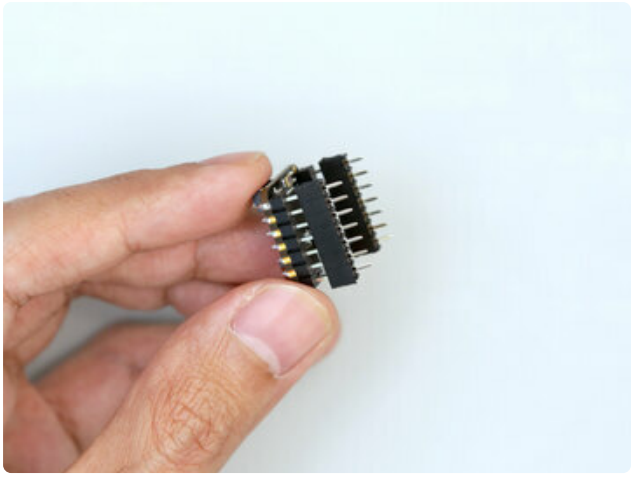


Solder Headers on QT Py

Install the two strips of headers to the pins on the QT Py.

Solder all 13 pins on the QT Py.

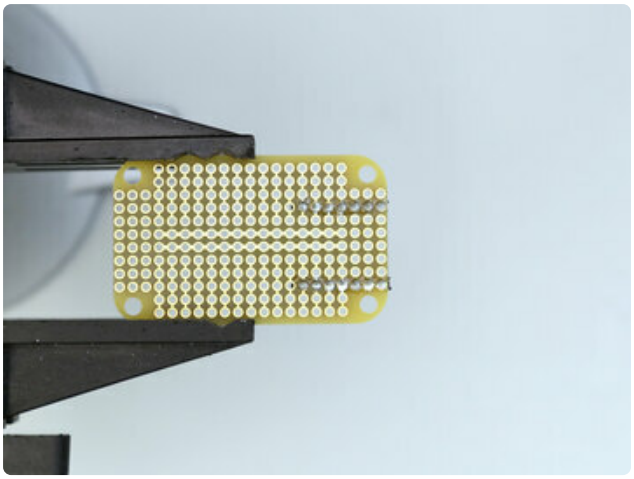
Use a breadboard to help keep the board sturdy while soldering.



Installing Socket Headers

Fit the two socket headers onto the male header pins of the QT Py.

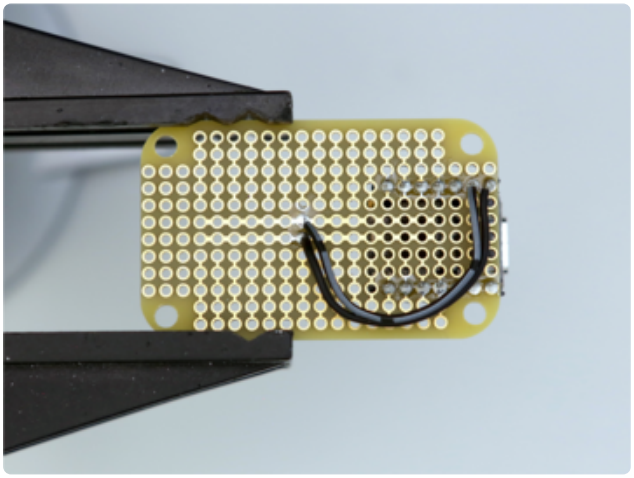
This will help keep the pins sturdy while soldering to the Perma Proto board.



Connect Socket Headers to Perma Proto

Place the QT Py with the socket headers onto the pins of the Perma Proto.

Solder all 13 pins to the Perma Proto board.

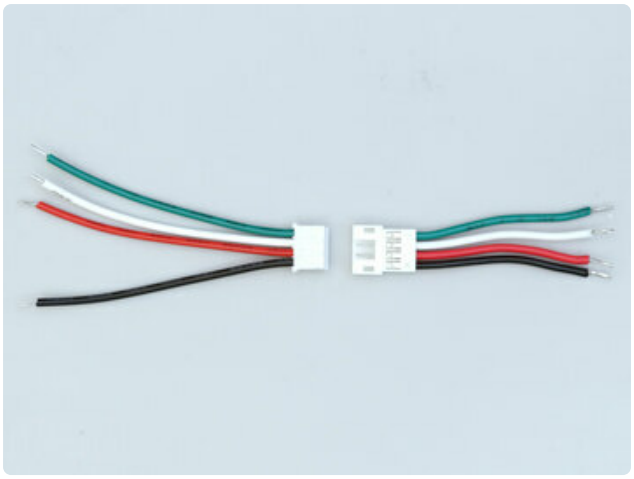


Solder Ground and Voltage Rails

Create a 2-pin wire from the silicone ribbon cable.

Solder the cable to the ground pin on QT Py and connect it to the ground rails on the Perma Proto

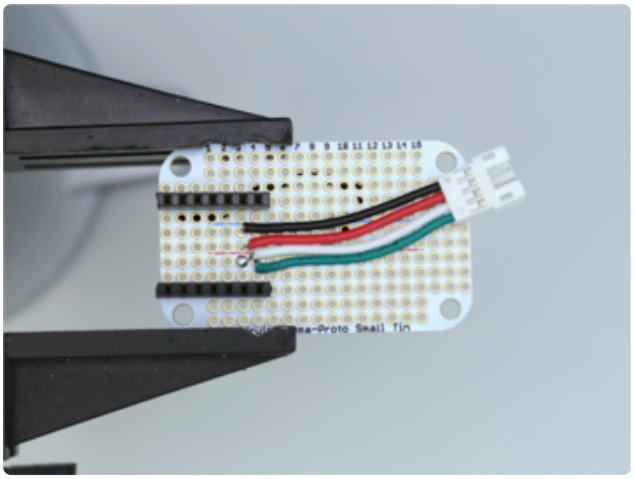
Solder the remaining wire to the 5V pin on QT Py and connect it to the voltage/power rails on the Perma Proto.

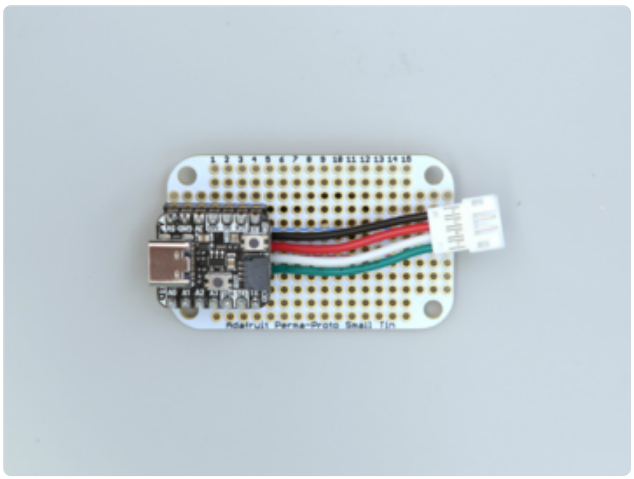


Prep Cable

Use wire cutters to shorten the lengths of the wire on both sets of the 4-pin cables.

Use soldering iron to add a bit of solder to the strips. This helps keep the strands of wires from fraying when soldering.

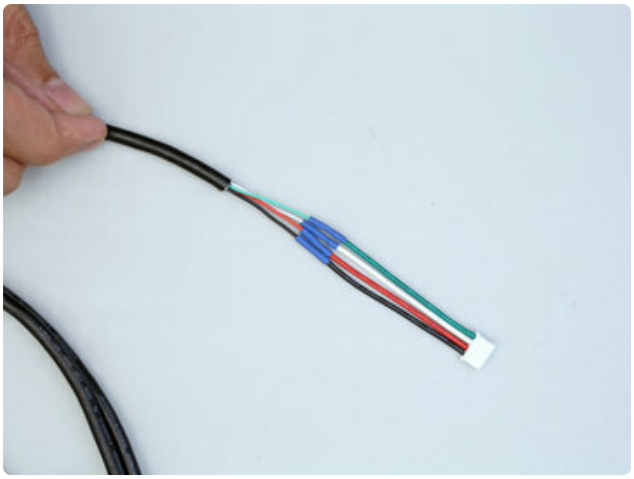




Connect Cable to Perma Proto

Remove the QT Py from the Perma Proto board and solder one set of the 4-pin cable to the 5V, GND and data pin on QT Py.

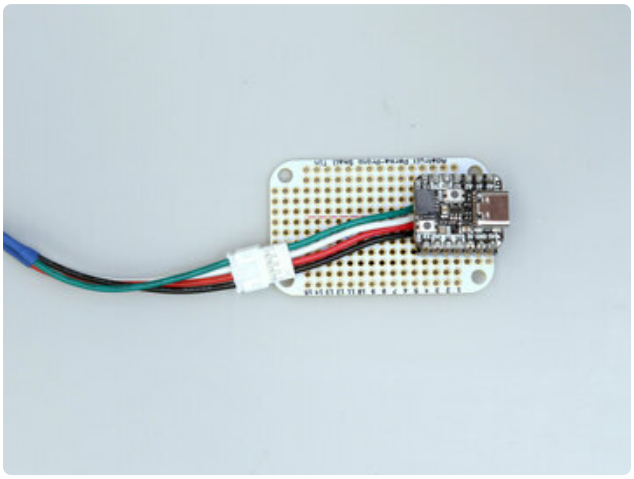
Install the QT Py back onto the Perma Proto when ready.



Attach Connector to Extension Cable

Use the second 4-pin cable and solder it to the wires on the end of the extension cable.

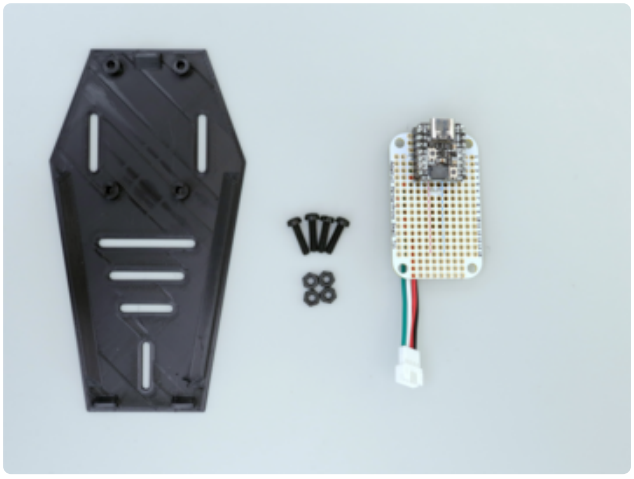
Use pieces of heat shrink tubing to insulate the exposed wires.



Connect Strips to QT Py

Plug in the 4-pin cables together.

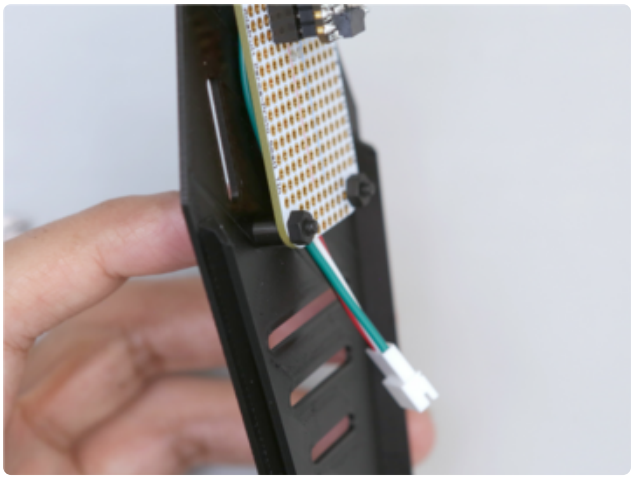
Assembly



Hardware for Perma Proto

Use the following hardware to secure the Perma Proto to the bottom cover.

4x M3 x 10mm screws
4x M3 hex nuts

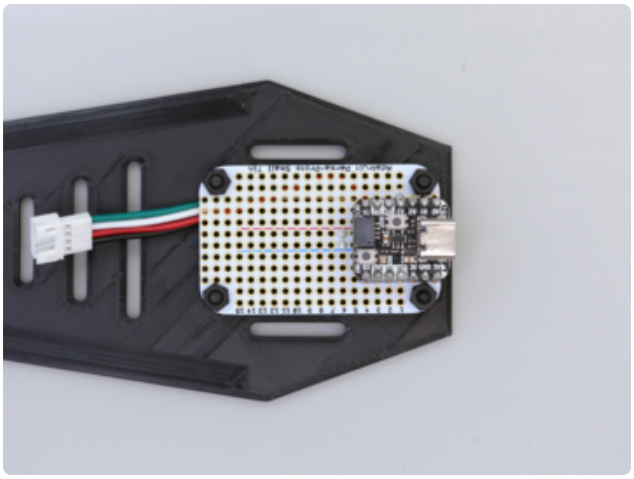


Install Screws and Nuts

Insert the M3 screws through the bottom mounting holes.

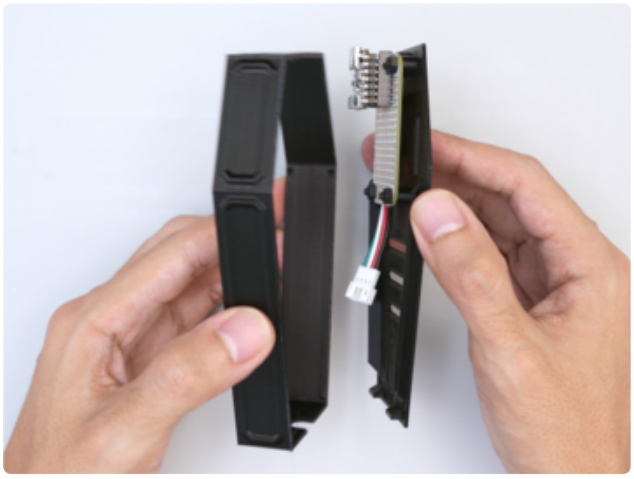
Place the Perma Proto board onto the standoffs with the thread of the screws going through the mounting holes.

Use the hex nuts to secure the Perma Proto board to the bottom cover.



Secured Perma Proto

Inspect the Perma Proto board and 4-pin cable to enclosure none of the wires are being kinked.



Install Frame

Orient the case frame with the bottom cover.

Snap fit the case frame over the bottom cover.

Connect Cables

Plug in a USB-C cable to the QT Py.

Install Top Cover

Orient the top cover with the case framing and snap fit them together.

Final Build

Plug in the USB-C cable from the QT Py to a 5V power source.

Congratulations on building the neon sign!

Additional Signs

Repeat the processes above to create the other signs.