



LED Matrix Scoreboard

Created by Ruiz Brothers



<https://learn.adafruit.com/led-matrix-scoreboard>

Last updated on 2024-06-03 03:09:53 PM EDT

Table of Contents

Overview	5
<ul style="list-style-type: none">• LED Pixel Hoops• IR Sensing Hoop• Prerequisite Guides• Adafruit Parts• Black LED Acrylic• Hardware Screws	
Circuit Diagram	9
<ul style="list-style-type: none">• Adafruit Library for Fritzing• Wired Connections• Powering	
3D Printing	10
<ul style="list-style-type: none">• Parts List• Build Plate• CAD Assembly• Slicing Parts	
Using Feather M4 Express	12
CircuitPython on Feather M4 Express	12
<ul style="list-style-type: none">• Set up CircuitPython Quick Start!• Further Information	
Using MatrixPortal	15
Install CircuitPython	16
<ul style="list-style-type: none">• Set up CircuitPython Quick Start!• Further Information	
Prep the MatrixPortal	18
<ul style="list-style-type: none">• Power Prep• Power Terminals• Panel Power• Dual Matrix Setup• Board Connection	
Coding the LED Matrix Scoreboard	22
<ul style="list-style-type: none">• Installing Project Code	
LED Matrix Scoreboard Code Walkthrough	25
Hoop Assembly	29
<ul style="list-style-type: none">• Wiring IR breakbeam receiver• Wiring IR breakbeam transmitter• IR breakbeam Hoop Wiring• Secure IR breakbeam receiver• Thread Wiring• Hole Relief for Wiring• Exit Hole	

- Installed IR receiver
- Solder JST cable
- Soldered JST cable
- Secure IR Transmitter

Wiring Feather

32

- Matrix Power Cable
- Feather & FeatherWing Setup
- JST Cables
- Feather & FeatherWing Connections
- Solder Cable – IR breakbeam receiver
- Solder Cable – IR breakbeam TX
- FeatherWing Connections
- Button Wiring
- Solder Signal – IR transmitter
- Wiring STEMMA Speaker
- Wired Feather & FeatherWing
- Wire Check

Matrix Assembly

36

- Install Corner Brackets
- Insert Hex Nuts
- Fitting Corner Brackets
- Installing Feet
- Secure Feet
- Installed Feet
- Button Wiring
- Install Button Mounting Plate
- Secure Button Mounting Plate
- Panel Mount Arcade Button
- Secure Arcade Button
- Install Hoop Brackets to Feet
- Secure Hoop Brackets
- Secured Brackets
- Second Hoop Bracket
- Install Hoop
- Secure Hoop
- Secured Hoop
- Installing Feather & FeatherWing
- Installed Feather & FeatherWing
- STEMMA Speaker
- Mount Stemma Speaker
- Secure Speaker Mount
- Plug In Connections
- Final Build

Overview



LED Pixel Hoops

Build an LED scoreboard with mini IR sensing hoop using an RGB Matrix and CircuitPython! This indoor game lets you shoot hoops and display your score on a 64x32 LED Matrix.



IR Sensing Hoop

The hoop houses an IR breakbeam sensor that detects when a ball has gone through it. The 3D printed parts hold everything together and are designed to be easy to build. There's an arcade button on the side that's used for starting a new game and resetting the score. This build is powered by an Adafruit Feather M4 Express and the RGB matrix FeatherWing.

Prerequisite Guides

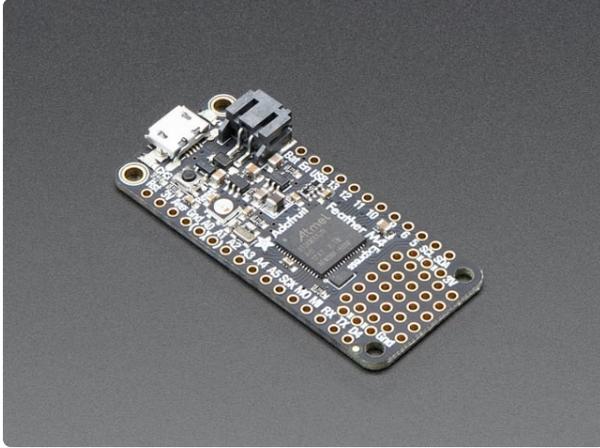
Take a moment to walk through the following guides to get better understanding of the parts used in this project.

- [RGB LED Matrix Guide \(https://adafru.it/Me5\)](https://adafru.it/Me5)
- [Adafruit Feather M4 Guide \(https://adafru.it/CJN\)](https://adafru.it/CJN)
- [RGB Matrices with CircuitPython \(https://adafru.it/L7b\)](https://adafru.it/L7b)

Black LED acrylic is used to diffuse the LEDs that are otherwise harsh and over exposed. This material features a matte finish and has the ability to transmit any color of light. That makes it perfect for bringing out the vibrance in the colors of the LEDs.

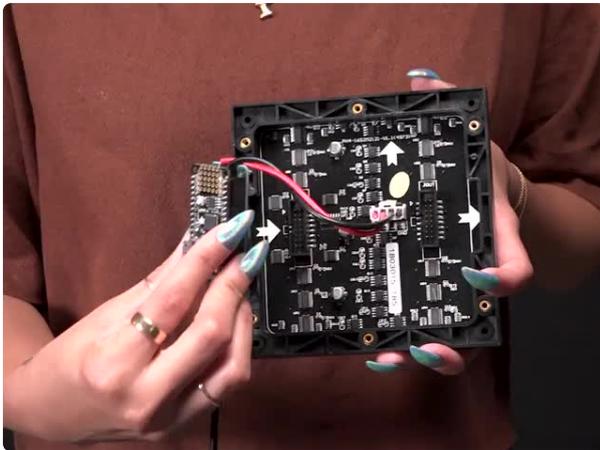
The 3D printed frame is designed with a slot on the side so it's easy to insert a sheet of Black LED acrylic.

Adafruit Parts



[Adafruit Feather M4 Express - Featuring ATSAM51](https://www.adafruit.com/product/3857)

It's what you've been waiting for, the Feather M4 Express featuring ATSAM51. This Feather is fast like a swift, smart like an owl, strong like a ox-bird (it's half ox,... <https://www.adafruit.com/product/3857>



[Adafruit RGB Matrix Featherwing Kit](https://www.adafruit.com/product/3036)

Ahoy! It's time to create a dazzling light up project with our new RGB Matrix FeatherWing. Now you can quickly and easily create...

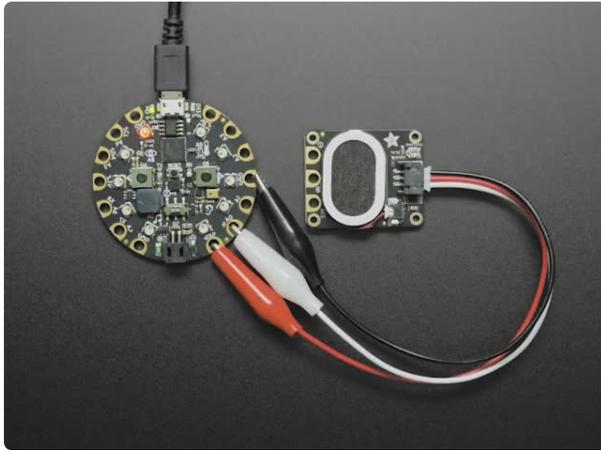
<https://www.adafruit.com/product/3036>



[64x32 RGB LED Matrix - 4mm pitch](https://www.adafruit.com/product/2278)

Bring a little bit of Times Square into your home with this sweet 64 x 32 square RGB LED matrix panel. These panels are normally used to make video walls, here in New York we see them...

<https://www.adafruit.com/product/2278>



Adafruit STEMMA Speaker - Plug and Play Audio Amplifier

Hey, have you heard the good news? With Adafruit STEMMA boards you can easily and safely plug sensors and devices together, like this Adafruit STEMMA Speaker - Plug and Play...

<https://www.adafruit.com/product/3885>



Mini LED Arcade Button - 24mm Translucent Clear

A button is a button, and a switch is a switch, but these translucent arcade buttons are in a class of their own. Particularly because they have LEDs built right in!...

<https://www.adafruit.com/product/3429>

1 x IR Break Beam Sensor
5mm LEDs

<https://www.adafruit.com/product/2168>

1 x IR Break Beam Sensor
3mm LEDs

<https://www.adafruit.com/product/2167>

1 x 2-pin JST Cable
Female Connector 100mm

<https://www.adafruit.com/product/261>

1 x 2-pin JST Cable
Male Connector 200mm

<https://www.adafruit.com/product/3814>

1 x 3-pin JST Cable
Male Connector 200mm

<https://www.adafruit.com/product/4046>

1 x 3-pin JST Cable
female socket cable

<https://www.adafruit.com/product/3894>

1 x 10-wire Ribbon Cable
Silicone Cover Stranded-Core 1m long

<https://www.adafruit.com/product/3890>

Female Header Set

1 x [Header kit for Feather](#)

<https://www.adafruit.com/product/2886>

Female Header Set

1 x [M2.5 Screw Kit](#)

<https://www.adafruit.com/product/3299>

Black Nylon M2.5

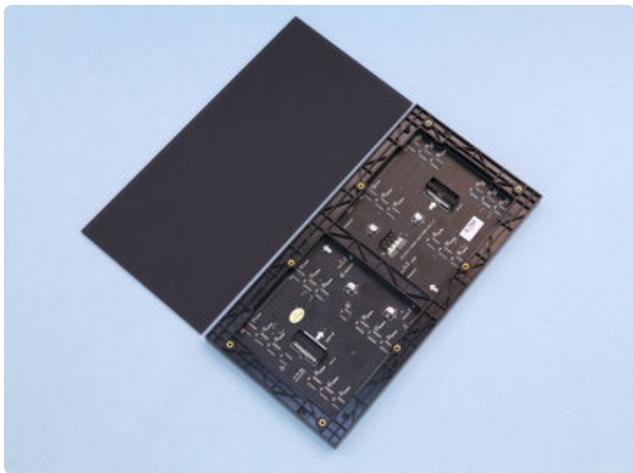
1 x [5V 4A Wall Adapter](#)

<https://www.adafruit.com/product/1466>

switching power supply

Black LED Acrylic

A sheet of black LED acrylic used to diffuse the RGB LED Matrix. From their website:



It looks like a matte black acrylic when it is not lit, but add a light behind it and watch it transform. This fantastic material has the ability to transmit light of any color.

Recommended Cut To Size Sheet: 257mm x 128mm (10.12in x 5.04in)

1 x [Black LED Acrylic](#)

https://www.tapplastics.com/product/plastics/cut_to_size_plastic/black_led_sheet/668

Black LED Acrylic from TAP Plastics

Hardware Screws

Fasteners used to secure 3d printed parts together.

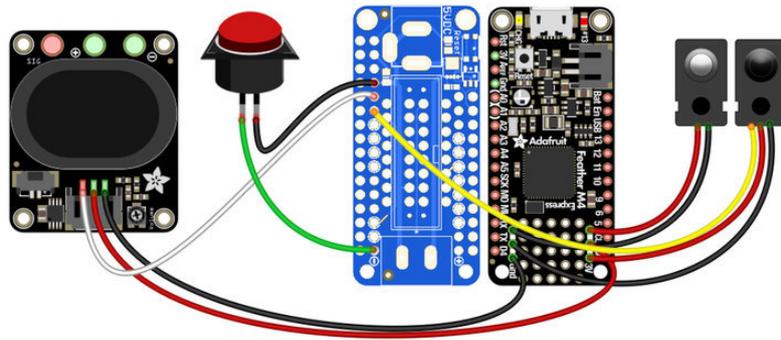
- 8x M3 x 10mm pan head screws
- 4x M3 x 6mm pan head screws
- 6x M2.5 x 10mm pan head screws
- 4x M2.5 hex nuts

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) (<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) (<https://adafru.it/AYZ>).



fritzing

Wired Connections

- VCC from Speaker to 3V on Feather
- GND from Speaker to GND on Feather
- Signal from Speaker to A0 on FeatherWing
- Button to GND on Feather
- Button to D4 on FeatherWing
- IR break beam (2-wire) to GND on Feather
- IR break beam (2-wire) to 3V on Feather
- IR break beam (3-wire) to GND on Feather
- IR break beam (3-wire) to 3V on Feather
- IR break beam (3-wire) to A1 on FeatherWing

Powering

The RGB Matrix will need to be powered by a [5V 4A wall adapter \(http://adafruit.com/1466\)](http://adafruit.com/1466). 5V USB cable is not enough to fully power the display – Glitches and artifacts are displayed when RGB matrix is only powered by USB cable.



[5V 4A \(4000mA\) switching power supply - UL Listed](http://adafruit.com/1466)

Need a lot of 5V power? This switching supply gives a clean regulated 5V output at up to 4 Amps (4000mA). 110 or 240 input, so it works in any country. The plugs are "US..."

<https://www.adafruit.com/product/1466>

3D Printing



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.

hoop-190.stl
hoop-204.stl
foot a.stl
foot b.stl
hoop bracket a.stl
hoop bracket b.stl
2x corner-a.stl
2x corner-b.stl
speaker mount.stl
button mount.stl

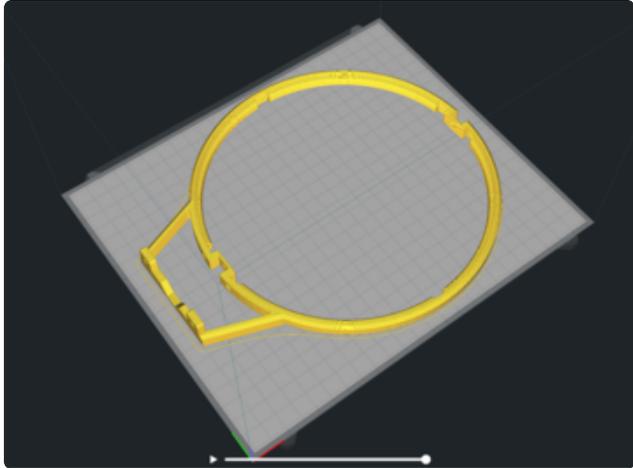
The corner mounts are printed twice. Make two copies of each corner.

Download CAD files from
Thingiverse

<https://adafru.it/MdR>

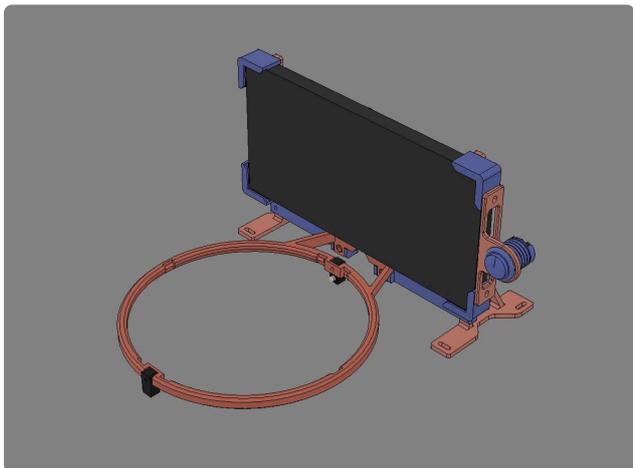
Download CAD files from
PrusaPrinters

<https://adafru.it/MdS>



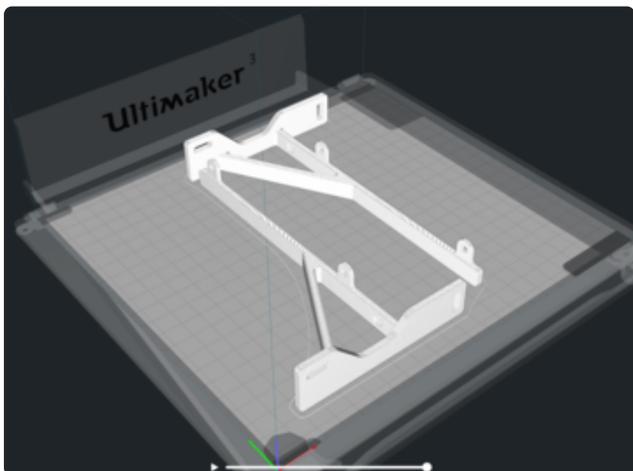
Build Plate

The hoop requires a minimum build area of 250mm x 210mm x 200mm. Two sizes are provided. hoop-190.stl has an outer diameter of 190mm. hoop-204.stl has an outer diameter of 204mm. Choose the one that best fits on your printers build plate.



CAD Assembly

The corner mounts are fitted over the RGB matrix display and features a spot for additional black LED acrylic. The two foot mounts are placed over the corners and secured with screws. A button mounting plate is secured to the side of the display. The hoop is attached to a set of brackets that are secured to the two mounting feet. The Stemma speaker is attached to a mounting plate and secured to the back of the display.



Slicing Parts

No supports are required. Slice with settings for PLA material.

The parts were sliced using CURA using the slice settings below.

PLA filament 220c extruder
0.2 layer height
10% gyroid infill
60mm/s print speed
60c heated bed

We suggest using PETG filament for stronger more durable parts.

Using Feather M4 Express

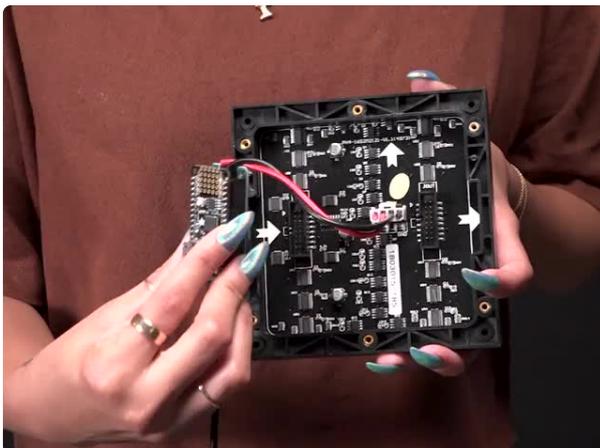
You can build this project using the Feather M4 Express. For a detailed list of all parts needed for this project, check out the Overview.



[64x32 RGB LED Matrix - 6mm pitch](#)

Bring a little bit of Times Square into your home with this sweet 64x32 square RGB LED matrix panel. These panels are normally used to make video walls, here in New York we see them on...

<https://www.adafruit.com/product/2276>



[Adafruit RGB Matrix Featherwing Kit](#)

Ahoy! It's time to create a dazzling light up project with our new RGB Matrix FeatherWing. Now you can quickly and easily create...

<https://www.adafruit.com/product/3036>

CircuitPython on Feather M4 Express

[CircuitPython \(https://adafru.it/tB7\)](https://adafru.it/tB7) is a derivative of [MicroPython \(https://adafru.it/BeZ\)](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.

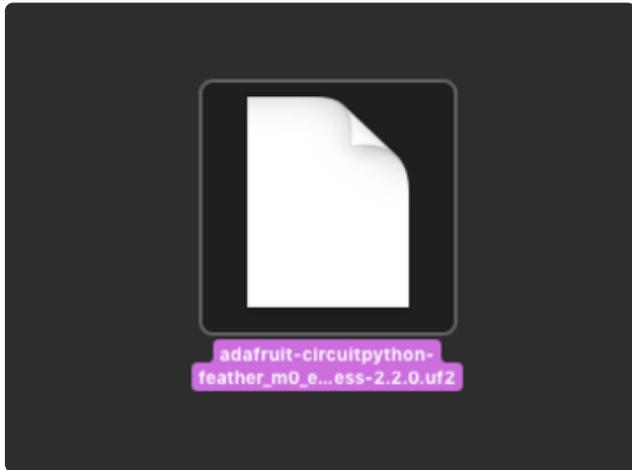
The following instructions will show you how to install CircuitPython. If you've already installed CircuitPython but are looking to update it or reinstall it, the same steps work for that as well!

Set up CircuitPython Quick Start!

Follow this quick step-by-step for super-fast Python power :)

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

<https://adafru.it/Emh>

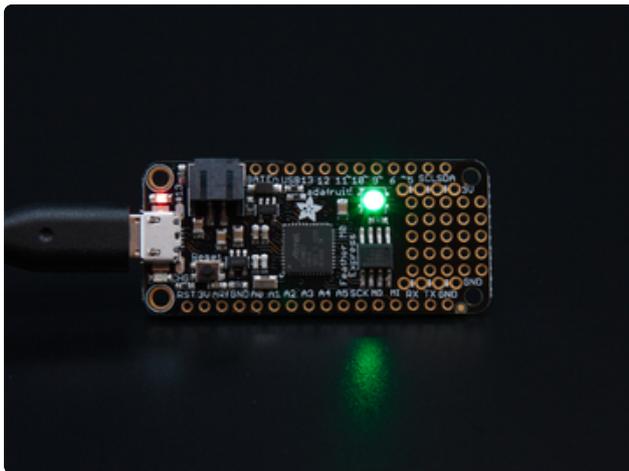


Click the link above and download the latest UF2 file.

Download and save it to your desktop (or wherever is handy).

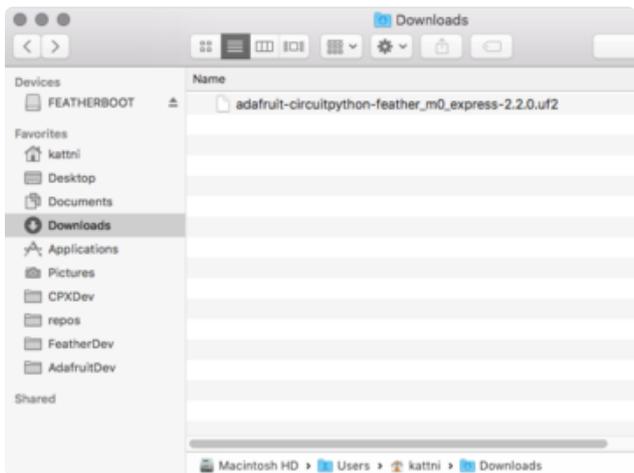
Plug your Feather M4 into your computer using a known-good USB cable.

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

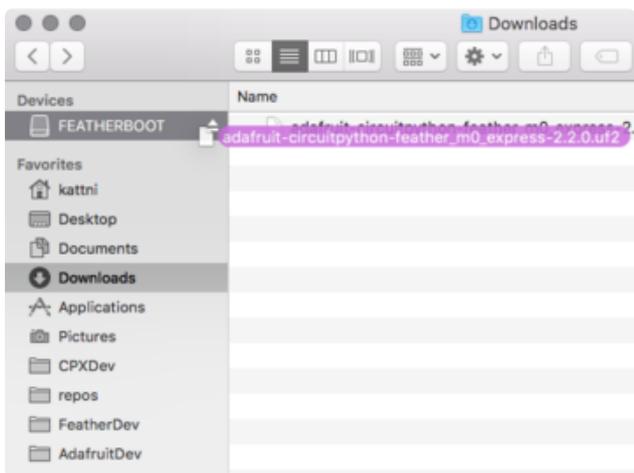


Double-click the **Reset** button next to the USB connector on your board, and you will see the NeoPixel RGB LED turn green. If it turns red, check the USB cable, try another USB port, etc. **Note:** The little red LED next to the USB connector will pulse red. That's ok!

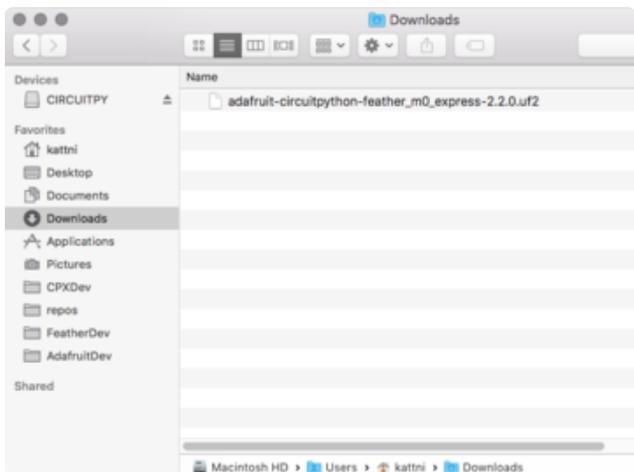
If double-clicking doesn't work the first time, try again. Sometimes it can take a few tries to get the rhythm right!



You will see a new disk drive appear called **FEATHERBOOT**.



Drag the `adafruit_circuitpython_etc.uf2` file to **FEATHERBOOT**.



The LED will flash. Then, the **FEATHERBOOT** drive will disappear and a new disk drive called **CIRCUITPY** will appear.

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

Further Information

For more detailed info on installing CircuitPython, check out [Installing CircuitPython \(https://adafru.it/Amd\)](https://adafru.it/Amd).

Using MatrixPortal

You can build this project with an all-in-one Matrix Portal board, its definitely the easiest and least-expensive way to go about it.

You will need a matrix portal, matrix, and USB C power/data cable



[Adafruit Matrix Portal - CircuitPython Powered Internet Display](https://www.adafruit.com/product/4745)

Folks love our wide selection of RGB matrices and accessories, for making custom colorful LED displays... and our RGB Matrix Shields...

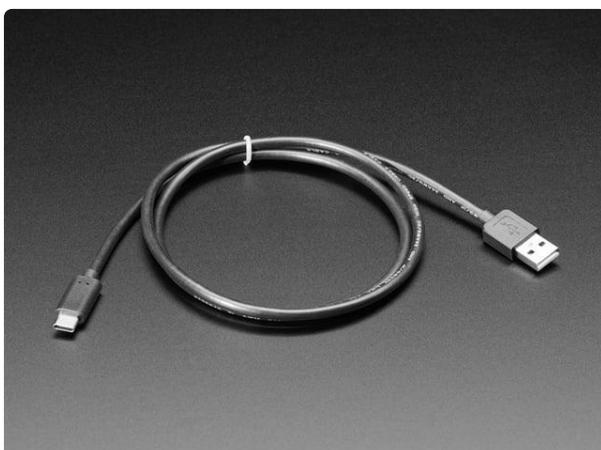
<https://www.adafruit.com/product/4745>



[64x32 RGB LED Matrix - 4mm pitch](https://www.adafruit.com/product/2278)

Bring a little bit of Times Square into your home with this sweet 64 x 32 square RGB LED matrix panel. These panels are normally used to make video walls, here in New York we see them...

<https://www.adafruit.com/product/2278>



[USB Type A to Type C Cable - approx 1 meter / 3 ft long](https://www.adafruit.com/product/4474)

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>

Install CircuitPython

[CircuitPython \(https://adafru.it/tB7\)](https://adafru.it/tB7) is a derivative of [MicroPython \(https://adafru.it/BeZ\)](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.

Set up CircuitPython Quick Start!

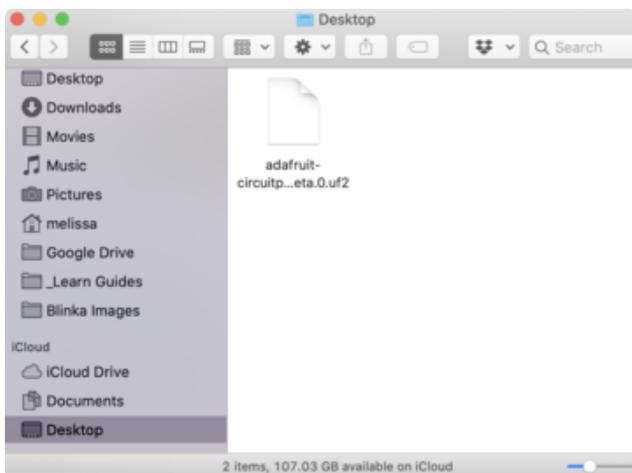
Follow this quick step-by-step for super-fast Python power :)

Download the latest version of
CircuitPython for this board via
[circuitpython.org](https://adafru.it/Nte)

<https://adafru.it/Nte>

Further Information

For more detailed info on installing CircuitPython, check out [Installing CircuitPython \(https://adafru.it/Amd\)](https://adafru.it/Amd).



Click the link above and download the latest UF2 file.

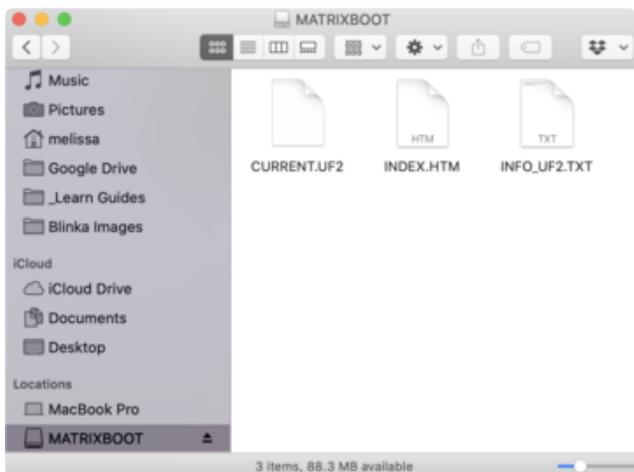
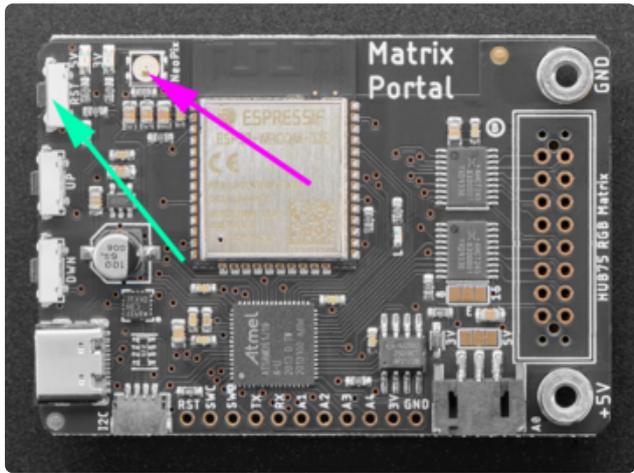
Download and save it to your desktop (or wherever is handy).

Plug your MatrixPortal M4 into your computer using a known-good USB cable.

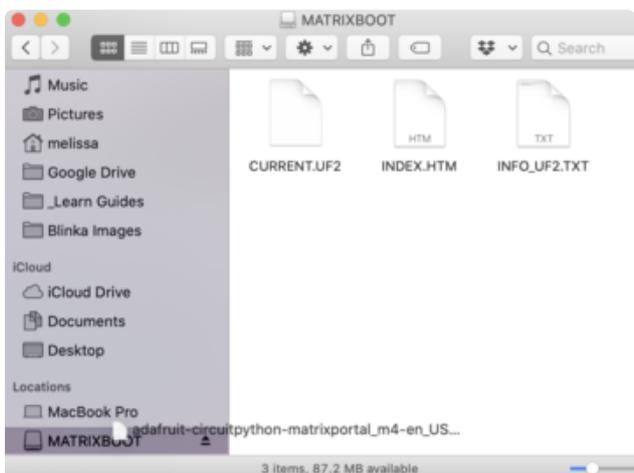
A lot of people end up using charge-only USB cables and it is very frustrating! So make sure you have a USB cable you know is good for data sync.

Double-click the **Reset** button (indicated by the green arrow) on your board, and you will see the NeoPixel RGB LED (indicated by the magenta arrow) turn green. If it turns red, check the USB cable, try another USB port, etc.

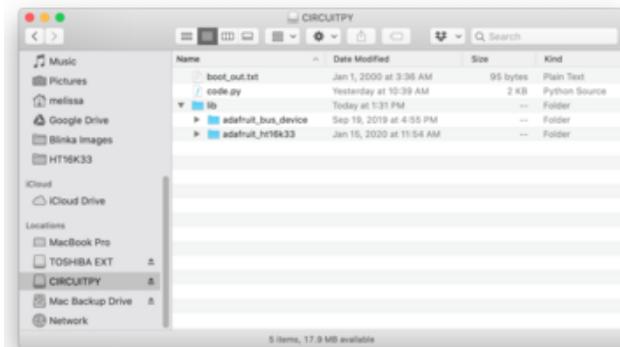
If double-clicking doesn't work the first time, try again. Sometimes it can take a few tries to get the rhythm right!



You will see a new disk drive appear called **MATRIXBOOT**.



Drag the `adafruit_circuitpython_etc.uf2` file to **MATRIXBOOT**.



The LED will flash. Then, the **MATRIXBOOT** drive will disappear and a new disk drive called **CIRCUITPY** will appear.

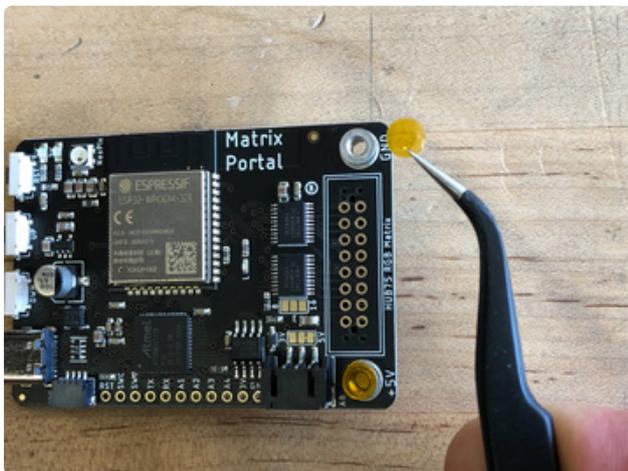
That's it, you're done! :)

Prep the MatrixPortal

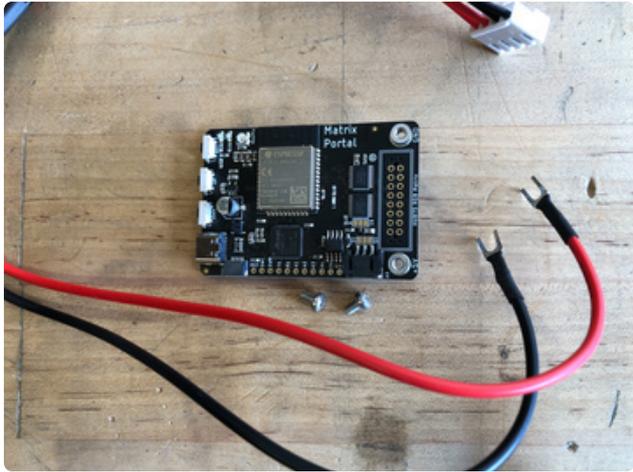


Power Prep

The MatrixPortal supplies power to the matrix display panel via two standoffs. These come with protective tape applied (part of our manufacturing process) which **MUST BE REMOVED!**



Use some tweezers or a fingernail to remove the two amber circles.

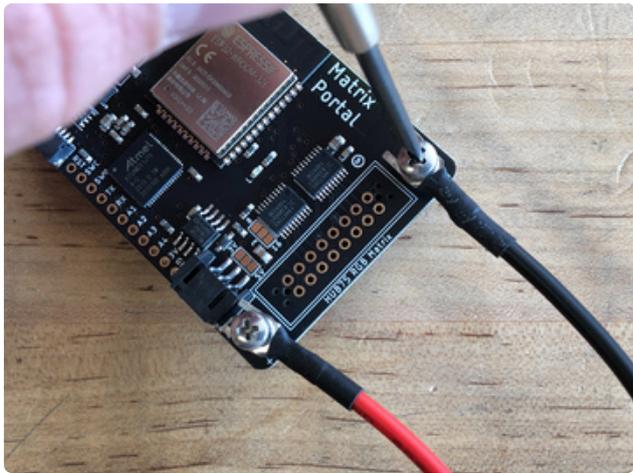


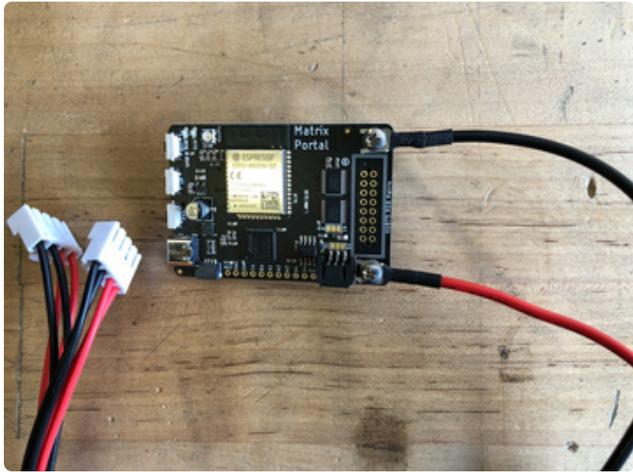
Power Terminals

Next, screw in the spade connectors to the corresponding standoff.

red wire goes to **+5V**

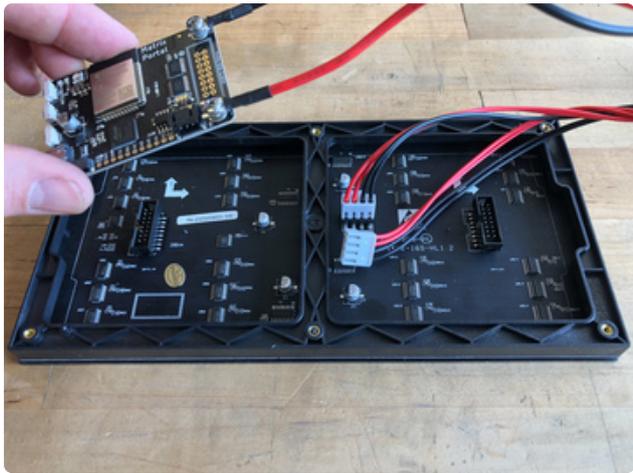
black wire goes to **GND**





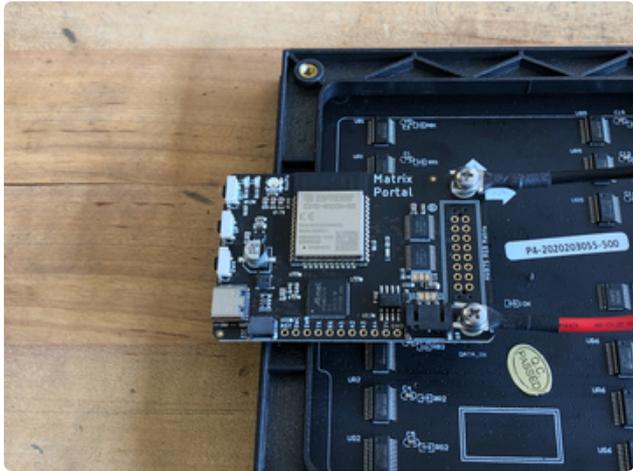
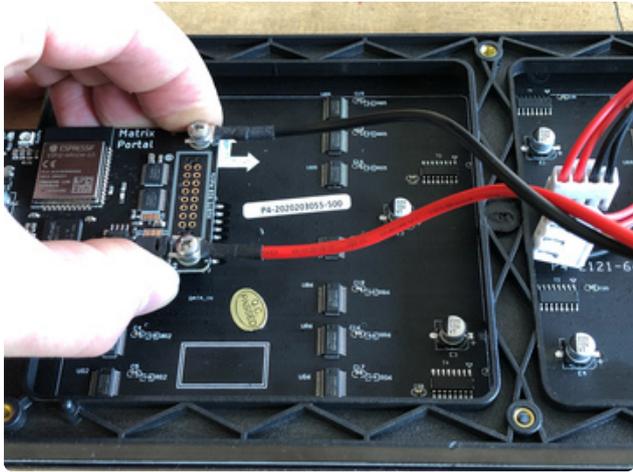
Panel Power

Plug either one of the four-conductor power plugs into the power connector pins on the panel. The plug can only go in one way, and that way is marked on the board's silkscreen.



Dual Matrix Setup

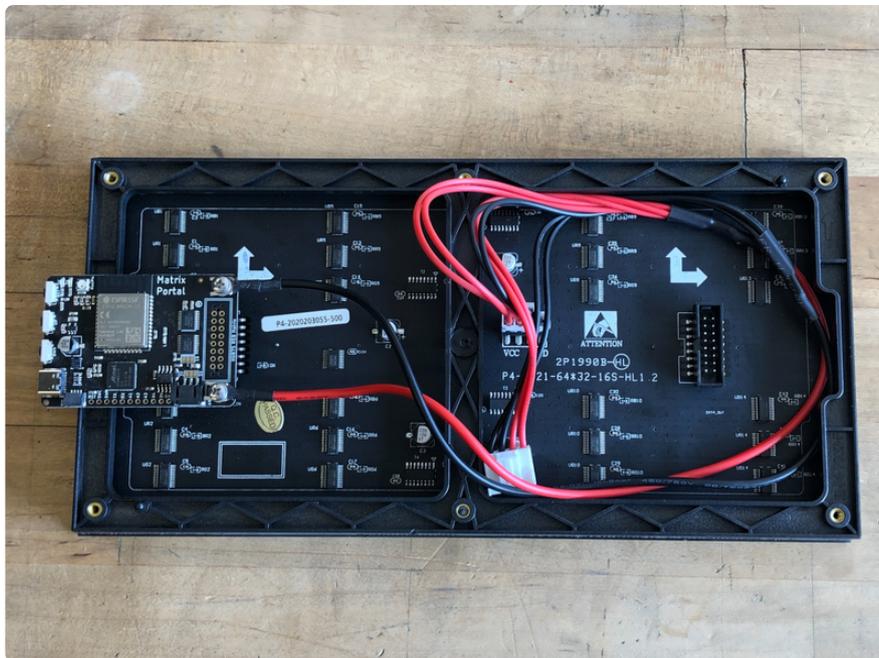
If you're planning to use a 64x64 matrix, [follow these instructions on soldering the Address E Line jumper \(https://adafru.it/OdJ\)](https://adafru.it/OdJ).



Board Connection

Now, plug the board into the left side shrouded 8x2 connector as shown. The orientation matters, so take a moment to confirm that the **white indicator arrow on the matrix panel is oriented pointing up and right** as seen here and the MatrixPortal overhangs the edge of the panel when connected. This allows you to use the edge buttons from the front side.

Check nothing is impeding the board from plugging in firmly. If there's a plastic nub on the matrix that's keeping the Portal from sitting flat, cut it off with diagonal cutters





For info on adding LED diffusion acrylic, see the page [LED Matrix Diffuser](#).

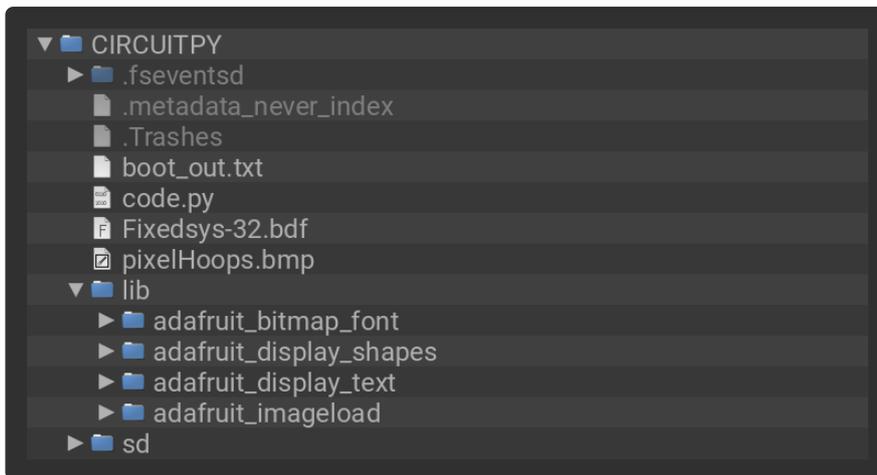
Coding the LED Matrix Scoreboard

Installing Project Code

To use with CircuitPython, you need to first install a few libraries, into the lib folder on your **CIRCUITPY** drive. Then you need to update **code.py** with the example script.

Thankfully, we can do this in one go. In the example below, click the **Download Project Bundle** button below to download the necessary libraries and the **code.py** file in a zip file. Extract the contents of the zip file, open the directory **LED_Matrix_Scorecard/** and then click on the directory that matches the version of CircuitPython you're using and copy the contents of that directory to your **CIRCUITPY** drive.

Your **CIRCUITPY** drive should now look similar to the following image:



```
# SPDX-FileCopyrightText: 2020 Liz Clark for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import board
import audioio
import audiomp3
import framebufferio
import rgbmatrix
import displayio
import adafruit_imageload
import digitalio
from adafruit_display_shapes.rect import Rect
from adafruit_bitmap_font import bitmap_font
from adafruit_display_text import label

# matrix setup
displayio.release_displays()
matrix = rgbmatrix.RGBMatrix(
    width=64,
    height=32,
    bit_depth=4,
    rgb_pins=[board.D6, board.D5, board.D9, board.D11, board.D10, board.D12],
    addr_pins=[board.A5, board.A4, board.A3, board.A2],
    clock_pin=board.D13,
    latch_pin=board.D0,
    output_enable_pin=board.D1,
)
display = framebufferio.FramebufferDisplay(matrix)

# display groups
start_group = displayio.Group()
score_group = displayio.Group()

# text & bg color setup for scoreboard
score_text = "    "
font = bitmap_font.load_font("/Fixedsys-32.bdf")
yellow = (255, 127, 0)
pink = 0xFF00FF

score_text = label.Label(font, text=score_text, color=0x0)
score_text.x = 23
score_text.y = 15

score_bg = Rect(0, 0, 64, 32, fill=yellow, outline=pink, stroke=3)

# start splash screen graphic
start, start_pal = adafruit_imageload.load("/pixelHoops.bmp",
                                          bitmap=displayio.Bitmap,
```

```

palette=displayio.Palette)

start_grid = displayio.TileGrid(start, pixel_shader=start_pal,
                                width=64, height=32)
# adding graphics to display groups
start_group.append(start_grid)
score_group.append(score_bg)
score_group.append(score_text)

# start by showing start splash
display.root_group = start_group

# setup for break beam LED pin
break_beam = digitalio.DigitalInOut(board.A1)
break_beam.direction = digitalio.Direction.INPUT
break_beam.pull = digitalio.Pull.UP

# setup for button pin
button = digitalio.DigitalInOut(board.D4)
button.direction = digitalio.Direction.INPUT
button.pull = digitalio.Pull.UP

# setup for speaker pin
speaker = audioio.AudioOut(board.A0)

# mp3 decoder setup
file = "/hoopBloop0.mp3"
mp3stream = audiomp3.MP3Decoder(open(file, "rb"))

# state machines used in the loop
score = 0
hoops = False
button_state = False
beam_state = False
sample = 0

while True:
    # button debouncing
    if not button.value and not button_state:
        button_state = True
    # debouncing for break beam LED
    if not break_beam.value and not beam_state:
        beam_state = True
    # if a game hasn't started and you press the button:
    if not button.value and not hoops:
        # game starts
        hoops = True
        button_state = False
        # display shows scoreboard
        display.root_group = score_group
        print("start game!")
        time.sleep(0.5)
    if hoops:
        # if the break beam LED detects a hoop:
        if not break_beam.value and beam_state:
            # score increase by 2 points
            score += 2
            # an mp3 plays
            file = "/hoopBloop{}.mp3".format(sample)
            mp3stream.file = open(file, "rb")
            speaker.play(mp3stream)
            print("score!")
            # resets break beam
            beam_state = False
            # increases mp3 file count
            # plays the 3 files in order
            sample = (sample + 1) % 3
            # score text x pos if 4 digit score
            if score >= 1000:

```

```

        score_text.x = -1
    # score text x pos if 3 digit score
    elif score >= 100:
        score_text.x = 7
    # score text x pos if 2 digit score
    elif score >= 10:
        score_text.x = 16
    elif score >= 0:
        score_text.x = 23
    # updates score text to show current score
    score_text.text = score
    time.sleep(0.01)
# if a game is in progress and you press the button:
if not button.value and hoops:
    # game stops
    hoops = False
    button_state = False
    # score is reset to 0
    score = 0
    score_text.text = score
    # display shows the start splash graphic
    display.root_group = start_group
    print("end game!")
    time.sleep(0.5)

```

LED Matrix Scoreboard Code Walkthrough

The code begins by importing the CircuitPython libraries.

```

import time
import board
import audioio
import audiomp3
import framebufferio
import rgbmatrix
import displayio
import adafruit_imageload
import digitalio
from adafruit_display_shapes.rect import Rect
from adafruit_bitmap_font import bitmap_font
from adafruit_display_text import label

```

The RGB matrix is setup using the **rgbmatrix** library.

```

# matrix setup
displayio.release_displays()
matrix = rgbmatrix.RGBMatrix(
    width=64,
    height=32,
    bit_depth=4,
    rgb_pins=[board.D6, board.D5, board.D9, board.D11, board.D10, board.D12],
    addr_pins=[board.A5, board.A4, board.A3, board.A2],
    clock_pin=board.D13,
    latch_pin=board.D0,
    output_enable_pin=board.D1,
)
display = framebufferio.FramebufferDisplay(matrix)

```

Then, the two display groups are created. There will be a group to show the start-up graphic and a second one to display the score when you're playing the game.

```
# display groups
start_group = displayio.Group(scale=1)
score_group = displayio.Group(scale=1)
```

The text (`score_text`) and colorful background (`score_bg`) are setup next for the scoreboard. The text is using a bitmap font.

```
# text & bg color setup for scoreboard
score_text = "      "
font = bitmap_font.load_font("/Fixedsys-32.bdf")
yellow = (255, 127, 0)
pink = 0xFF00FF

score_text = label.Label(font, text=score_text, color=0x0)
score_text.x = 23
score_text.y = 15

score_bg = Rect(0, 0, 64, 32, fill=yellow, outline=pink, stroke=3)
```

Following the score text setup, the start-up graphic bitmap is setup using the `adafruit_imageload` library.

```
# start splash screen graphic
start, start_pal = adafruit_imageload.load("/pixelHoops.bmp",
                                          bitmap=displayio.Bitmap,
                                          palette=displayio.Palette)

start_grid = displayio.TileGrid(start, pixel_shader=start_pal,
                                width=64, height=32)
```

The start-up bitmap and score text graphics are added to their respective display groups. The start-up bitmap is shown first using `display.root_group = start_group`.

```
# adding graphics to display groups
start_group.append(start_grid)
score_group.append(score_bg)
score_group.append(score_text)

# start by showing start splash
display.root_group = start_group
```

After the display is setup, the I/O pins are setup for the break beam LED, push button and speaker. The LED and button are setup as inputs and the speaker is setup as an `AudioOut` using the `audioio` library.

```
# setup for break beam LED pin
break_beam = digitalio.DigitalInOut(board.A1)
break_beam.direction = digitalio.Direction.INPUT
break_beam.pull = digitalio.Pull.UP

# setup for button pin
button = digitalio.DigitalInOut(board.D4)
button.direction = digitalio.Direction.INPUT
button.pull = digitalio.Pull.UP
```

```
# setup for speaker pin
speaker = audioio.AudioOut(board.A0)
```

The `MP3Decoder` is setup before the loop, similar to how display objects are setup early in the code. This helps to save memory when playing multiple .mp3 files. Notice how you aren't playing an .mp3 in this section, it's purely for setup.

```
# mp3 decoder setup
file = "/hoopBloop0.mp3"
mp3stream = audiomp3.MP3Decoder(open(file, "rb"))
```

The final portion of the setup creates that state machines that will be used in the loop. Their functions are commented below.

```
# state machines used in the loop
score = 0 # holds your score for the game
hoops = False # tracks if a game is active or not
button_state = False # button debouncing
beam_state = False # break beam LED debouncing
sample = 0 # file count for the mp3 files
```

The loop begins with debouncing setup for the button and break beam LED. This eliminates any glitches in receiving data from their inputs.

```
while True:
    # button debouncing
    if not button.value and not button_state:
        button_state = True
    # debouncing for break beam LED
    if not break_beam.value and not beam_state:
        beam_state = True
```

The button has a different function depending on whether or not a game is in progress. If a game has not started, then pressing the button begins a new game and also changes the display to show the scoreboard graphic.

```
# if a game hasn't started and you press the button:
if not button.value and not hoops:
    # game starts
    hoops = True
    button_state = False
    # display shows scoreboard
    display.root_group = score_group
    print("start game!")
    time.sleep(0.5)
```

A game runs if the `hoops` state is `True`. Every time the break beam LED detects a break, your score increases by two points and an .mp3 file plays.

There are three different .mp3 files and they play in order as the game goes on. The code is able to iterate through the files easily because they are all named **hoopBloop***. By naming them like this, you can use `"/hoopBloop{}.mp3".format(sample)`, to iterate through them with `sample` holding the number.

The code is referencing the earlier `mp3stream` that was setup before the loop to decode and play the .mp3 files.

Additionally, there is some text formatting that takes place depending on how many digits your score is. This allows for the score to remain centered during game play.

```
if hoops:
    # if the break beam LED detects a hoop:
    if not break_beam.value and beam_state:
        # score_increase by 2 points
        score += 2
        # an mp3 plays
        file = "/hoopBloop{}.mp3".format(sample)
        mp3stream.file = open(file, "rb")
        speaker.play(mp3stream)
        print("score!")
        # resets break beam
        beam_state = False
        # increases mp3 file count
        # plays the 3 files in order
        sample = (sample + 1) % 3
    # score text x pos if 4 digit score
    if score >= 1000:
        score_text.x = -1
    # score text x pos if 3 digit score
    elif score >= 100:
        score_text.x = 7
    # score text x pos if 2 digit score
    elif score >= 10:
        score_text.x = 16
    # score text x pos if 1 digit score
    elif score >= 0:
        score_text.x = 23
    # updates score text to show current score
    score_text.text = score
    time.sleep(0.1)
```

If a game is in progress and you press the button, it changes the `hoops` state to `False` and ends the game. The score is also reset to `0` and the display shows the start-up bitmap.

```
# if a game is in progress and you press the button:
if not button.value and hoops:
    # game stops
    hoops = False
    button_state = False
    # score is reset to 0
    score = 0
    # display shows the start splash graphic
    display.root_group = start_group
```

```
print("end game!")  
time.sleep(0.5)
```

Hoop Assembly



Wiring IR breakbeam receiver

Use a 2-wire ribbon cable and measure to 40cm in length. This wire is long in order to wrap around the 3D printed hoop. The ground and voltage wires from the IR sensor are extended by slicing them with the 2-wire ribbon cable. Use heat shrink tubing to insulate exposed wire.



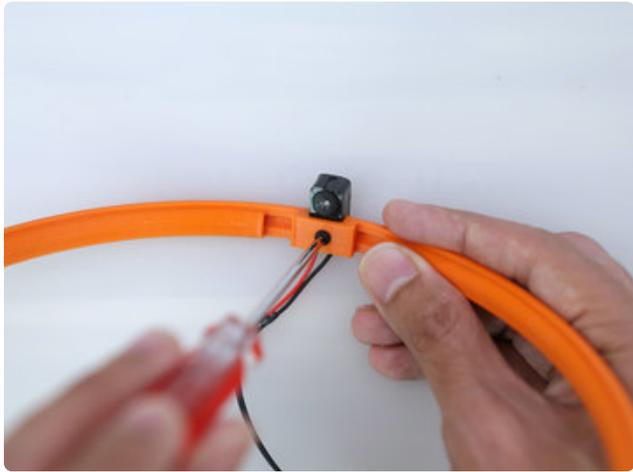
Wiring IR breakbeam transmitter

A 3-wire JST cable is soldered to the three wires on the IR transmitter. A JST connector allows the sensor to be easily connected and disconnected. Use heat shrink tubing to insulate exposed wire.



IR breakbeam Hoop Wiring

Place the IR breakbeam receiver inside the hoop and gauge the length of wire. It should be long enough to wrap around half of the hoop with extra slack at the end for connecting to a 2-wire JST connector.



Secure IR breakbeam receiver

Use an M2.5 x 10mm long screw to secure the IR receiver to the hoop. Press the IR sensor into the recess on the end of the hoop. Hold in place while fastening M2.5 screw. The LED should be orientated with the mounting tabs facing up.



Thread Wiring

The 3D printed hoop features a channel for housing the wiring. This keeps the ribbon cable from coming out during gameplay. Insert and thread the wiring through one of the channels. Doesn't matter which side as they're both symmetrical.



Hole Relief for Wiring

The channels have an opening nearly half way through the hoop to allow access to the wiring. This can help provide a hold on the wiring to further thread the wiring half-way around the hoop.



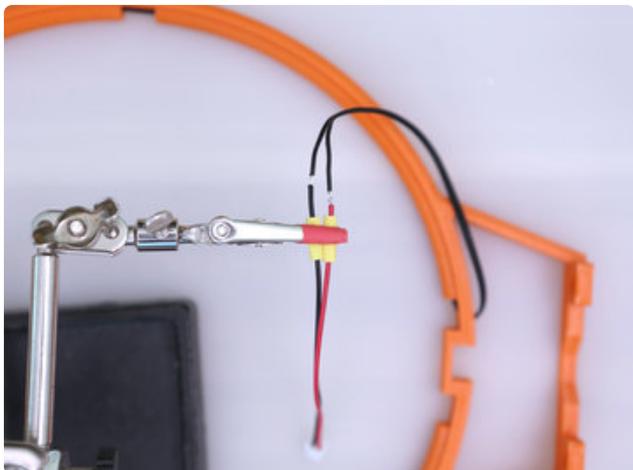
Exit Hole

Continue to thread the wiring from the IR receiver until it's poking through the other end. Carefully insert the wiring through the hole closest to the channel exit.



Installed IR receiver

Double check the wiring has been pulled all the way through. The wiring should wrap around half-way through the hoop. The IR LED should be orientated with the mounting tabs pointing up.



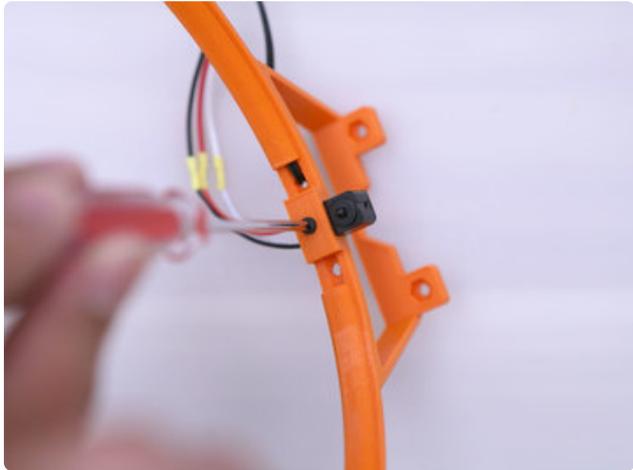
Solder JST cable

A 2-wire JST female cable is soldered to the wiring on the IR receiver. Double check the polarity is matching with the ground connecting to the black wire and voltage to the red wire. Use heat shrink tubing to insulate the exposed wire.



Soldered JST cable

The hoop now has the IR receiver installed and wired up.



Secure IR Transmitter

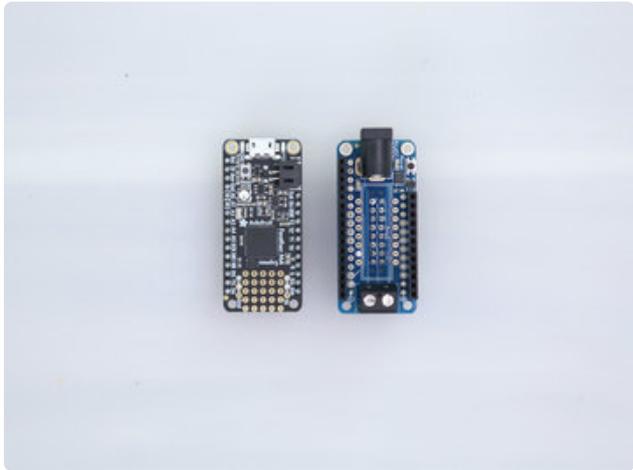
The IR transmitter is installed to the opposite end of the the hoop near the mounting tabs. Place and insert the IR transmitter into the recess in the middle. Hold in place while fastening M2.5 x 10mm long screw.

Wiring Feather



Matrix Power Cable

The 64x32 RGB matrix display features a power cable. The wiring has been shortened and simplified to make the wiring more manageable during assembly.



Feather & FeatherWing Setup

The Feather M4 Express has a 12-pin and 16-pin strip of male header pins soldered to the bottom of the PCB. The RGB Matrix FeatherWing has 12-pin and 16-pin female headers soldered to the top of the PCB. The 2x10 female socket soldered is to the bottom of the FeatherWing.



JST Cables

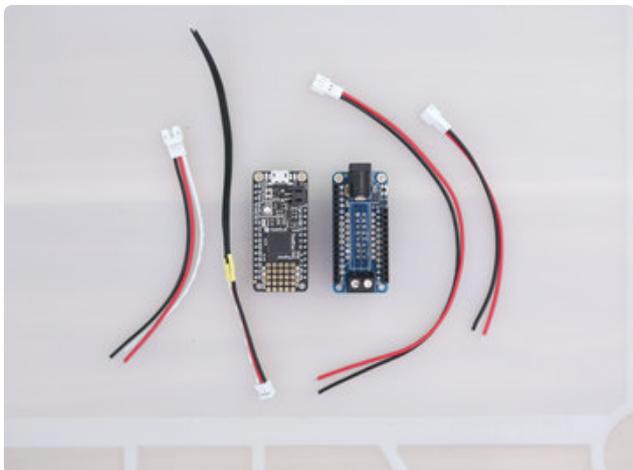
Four JST cables are used to connect the various components to the Feather and FeatherWing. Here's a list of the cables and where they're connected to.

2-wire male JST: Arcade Button

3-wire male JST: STEMMA Speaker

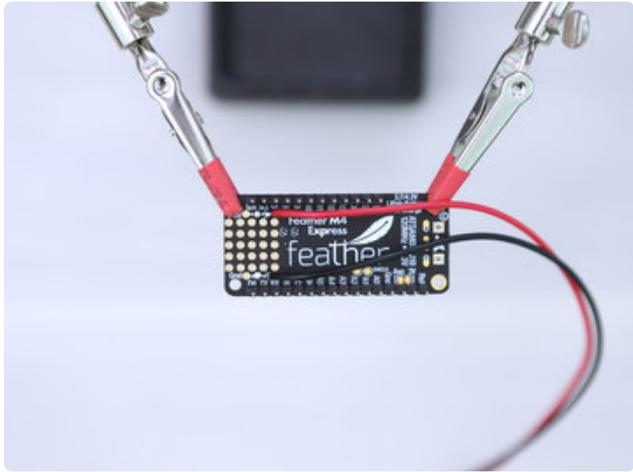
3-wire male JST: IR transmitter

2-wire male JST: IR receiver



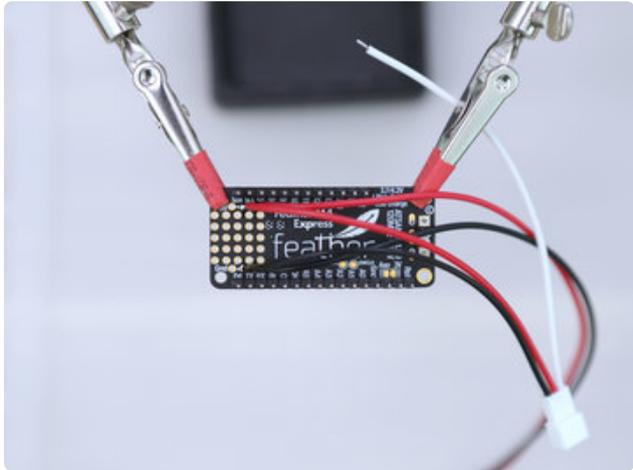
Feather & FeatherWing Connections

Double check the wires and cables are ready for soldering to the PCBs.



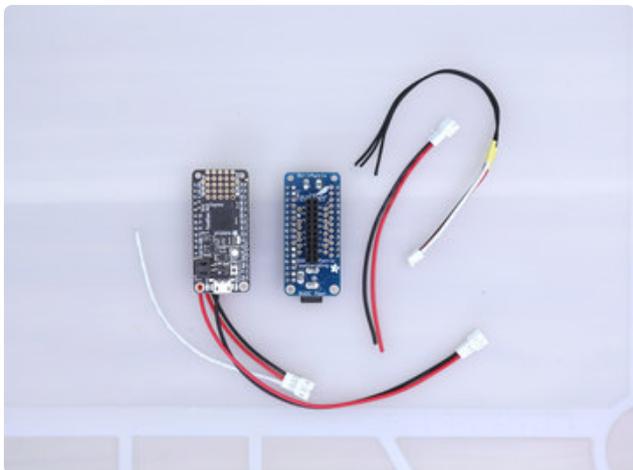
Solder Cable – IR breakbeam receiver

Begin by soldering the 2-wire JST cable to the ground and 3V pins on the Feather. This cable will connect to the IR receiver. The wiring is soldered to the bottom of the Feather.



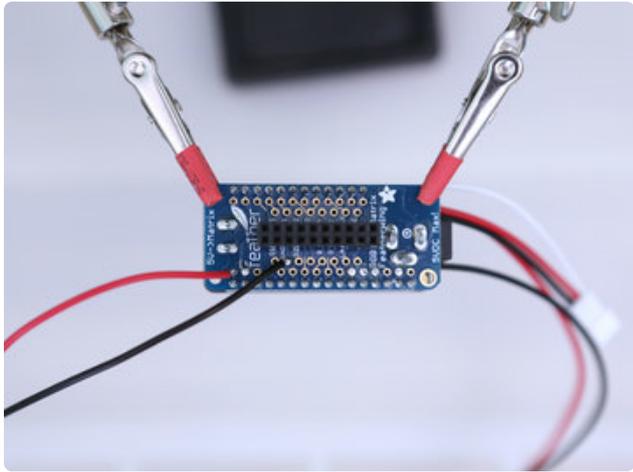
Solder Cable – IR breakbeam TX

Solder the 3-wire JST cable to available ground and 3V pins on the Feather. This cable will connect to the IR transmitter. The white wire will be connected to the A1 pin on the RGB Matrix FeatherWing.



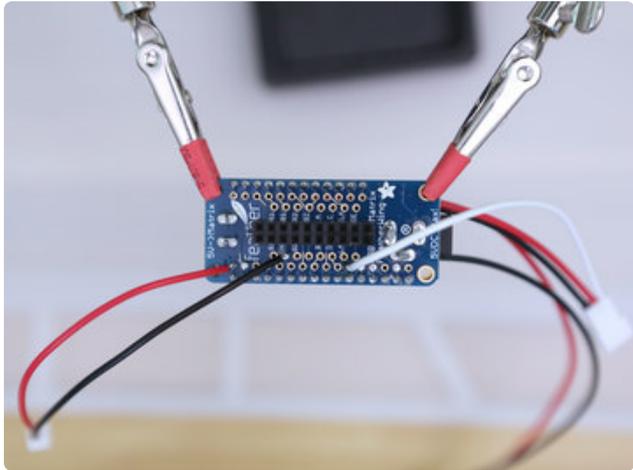
FeatherWing Connections

The remaining JST cables will be soldered to the RGB Matrix FeatherWing.



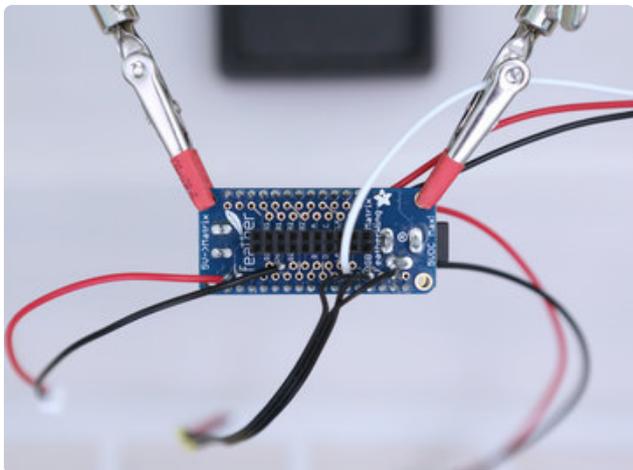
Button Wiring

Soldered the second 2-wire JST cable to the ground and D4 pins on the bottom of the FeatherWing. Reference the pins on the Feather to ensure the pins are correct.



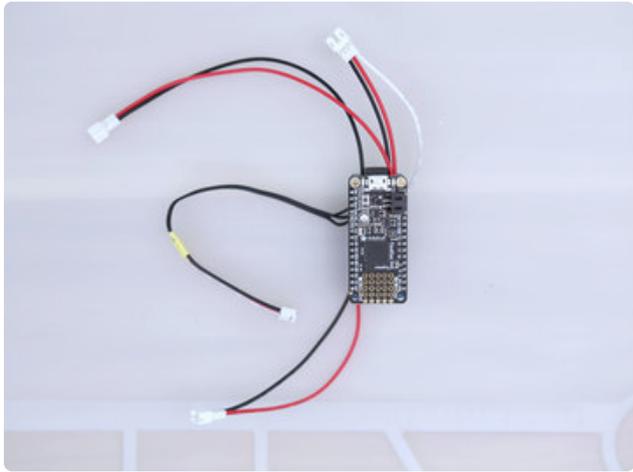
Solder Signal – IR transmitter

Solder the white wire from the 3-wire JST cable to the A1 pin on the bottom of the FeatherWing. This cable will be used to connect the IR transmitter.



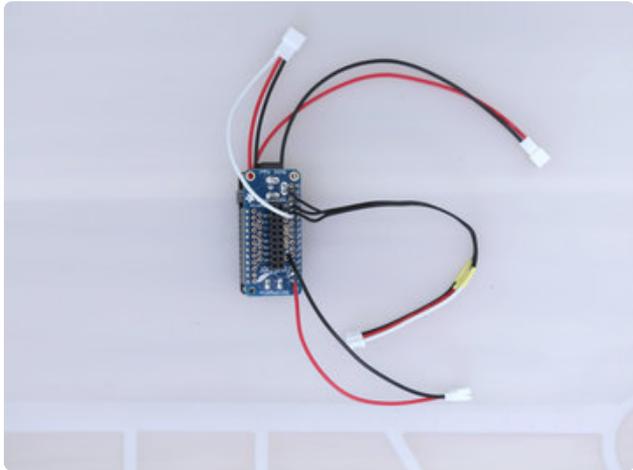
Wiring STEMMA Speaker

Solder the wires from the second 3-wire JST cable to the following pins on the FeatherWing. White wire to A0, black wire to ground and red wire to 3V.



Wired Feather & FeatherWing

The wiring is complete and ready for plugging in the various components.



Wire Check

Double check the wiring is correct.

Matrix Assembly



Install Corner Brackets

Corner brackets are secured to the four corners on the RGB matrix display. These brackets are secured to the mounting holes. They also fit a sheet of black LED acrylic.



Insert Hex Nuts

Grab two of the four corner brackets and insert M3 hex nuts into the recesses. The two corner brackets are used to secure the button mount to the side of the display. Choose which side you'd like to have the button mounted.



Fitting Corner Brackets

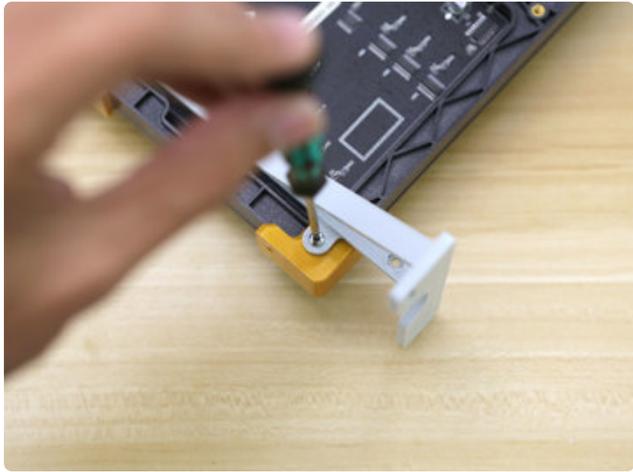
Place the brackets over the corners and orient them so the mounting holes are properly lined up. The side of display should have the brackets with visible mounting holes. These holes are used for securing the button mounting plate.

Optional: Insert the sheet of Black LED Acrylic by sliding into the corner brackets.



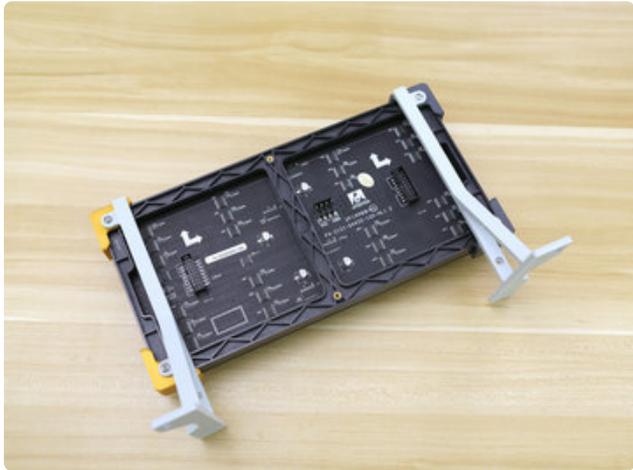
Installing Feet

Place the foot over the corner brackets and line up the mounting holes. Reference the photo for correct placement.



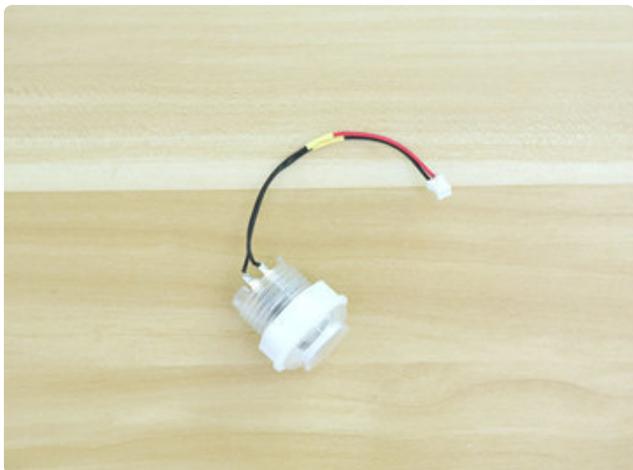
Secure Feet

Use two M3 x 10mm long screw to secure the feet to the corner brackets. The screws will also fasten into the threaded inserts that are on the display.



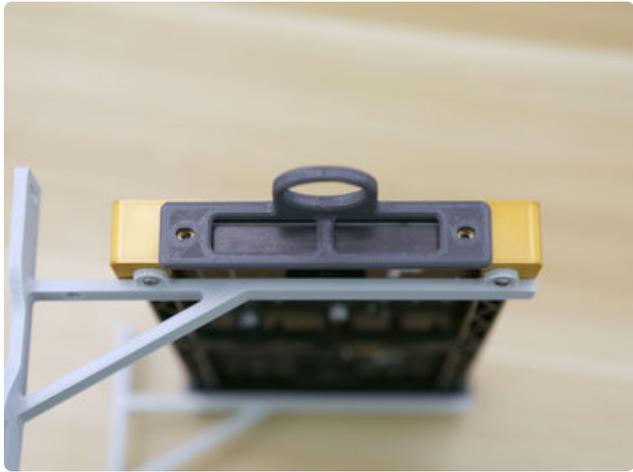
Installed Feet

Repeat this process for the second foot.



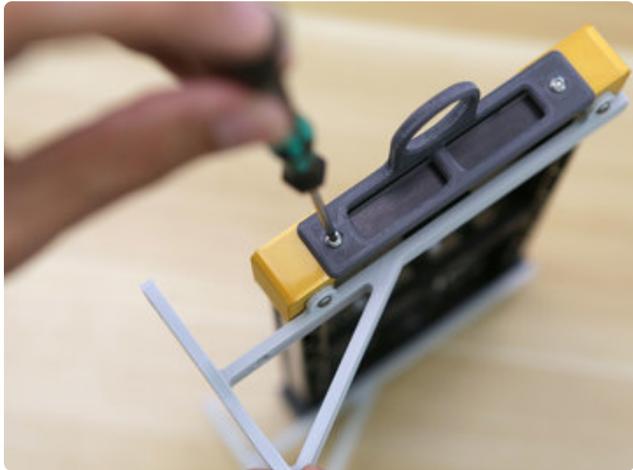
Button Wiring

A short 2-wire JST cable is soldered to the arcade button. This is connected to the JST cable that is soldered to the D4 pin on the FeatherWing.



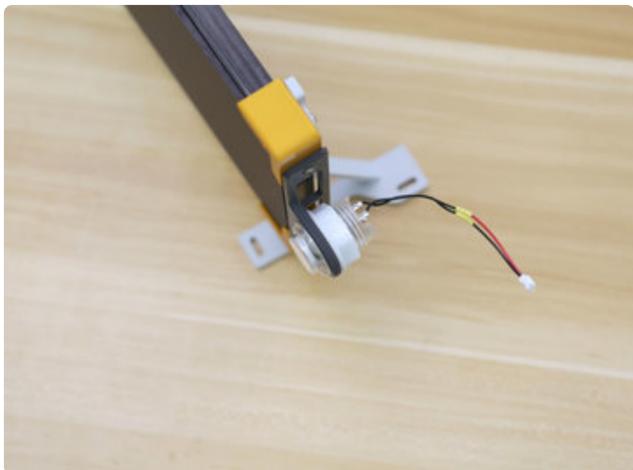
Install Button Mounting Plate

Place the button mounting plate over the corner brackets with the hex nuts you installed earlier. Line up the mounting holes.



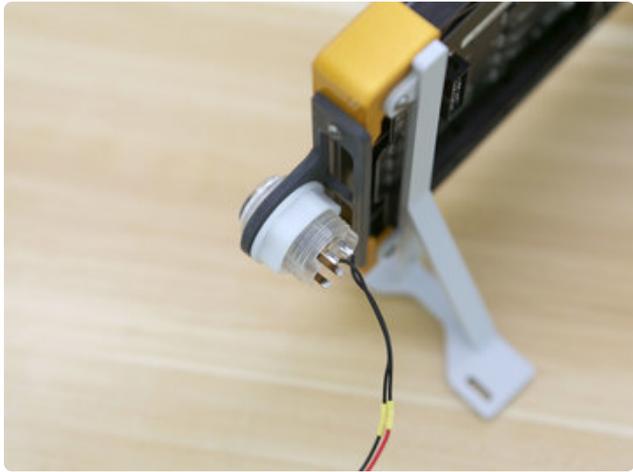
Secure Button Mounting Plate

Insert and fasten two M3 x 6mm long machine screws through the mounting holes. Reference the photo for best placement.



Panel Mount Arcade Button

Insert the arcade button to the button mounting plate. Make sure the button is facing in the correct orientation.



Secure Arcade Button

Use the screw-on sleeve to secure the arcade button to the button mounting plate.



Install Hoop Brackets to Feet

Place the two hoop brackets onto the feet. Line up the mounting holes. Reference the photo for correct placement and orientation.



Secure Hoop Brackets

Insert and fasten an M3 x 10mm long screw through the mounting holes on the hoop brackets.



Secured Brackets

Use an M3 hex nut to secure the screws. Insert the M3 hex nut into the recess on the back of the foot.



Second Hoop Bracket

Repeat this process for the other foot.



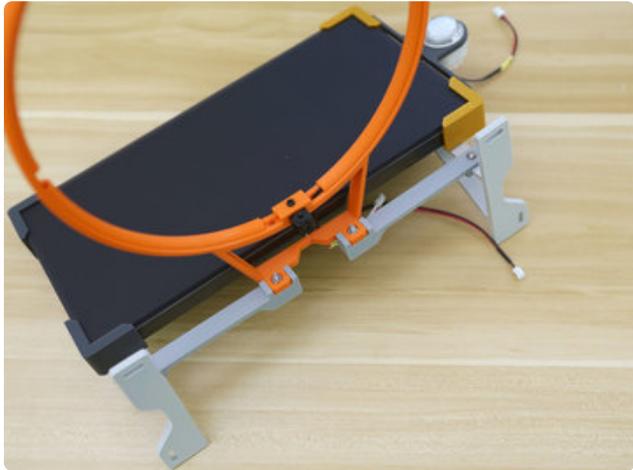
Install Hoop

Place the mounting tabs from the hoop into the two brackets on the feet. The brackets feature walls for press fitting the mounting tabs. The mounting holes should be lined up properly.



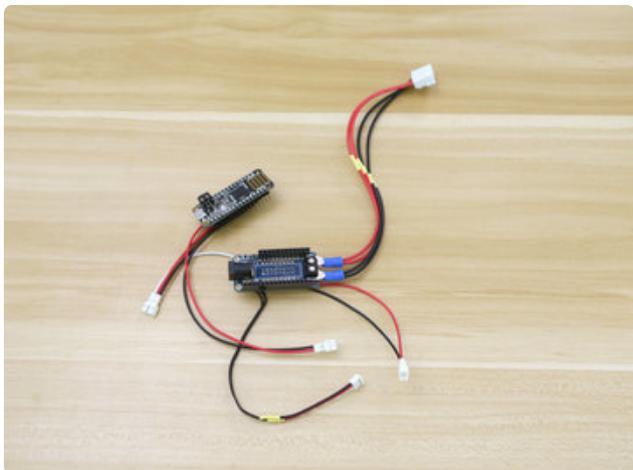
Secure Hoop

Insert and fasten M3 x 10mm long screws.



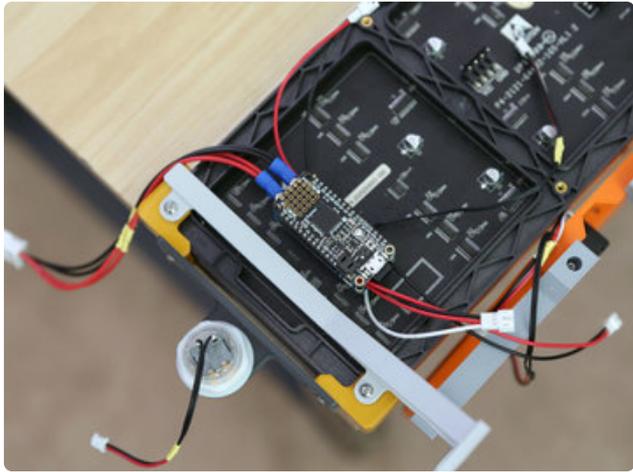
Secured Hoop

The mounting tabs on the hoop feature recesses for fitting M3 hex nuts. Insert the hex nuts to secure the M3 screws.



Installing Feather & FeatherWing

Get the PCBs ready to fit onto the back of the RGB matrix display.



Installed Feather & FeatherWing

The Feather M4 express is fitted on top of the RGB matrix FeatherWing. The FeatherWing is press fitted on top of the 2x8 IDC header on-board the display. Make sure it is placed on the correct header. Use the arrows to reference the input socket.



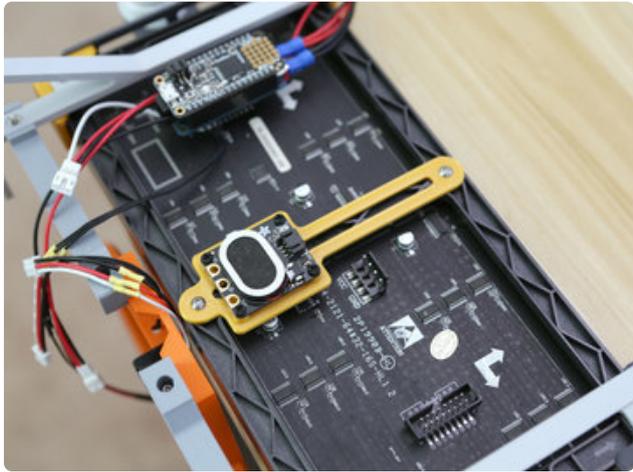
STEMMA Speaker

The Stemma speaker is secured to the speaker mount using 4x M2.5 x 10mm long machine screws.



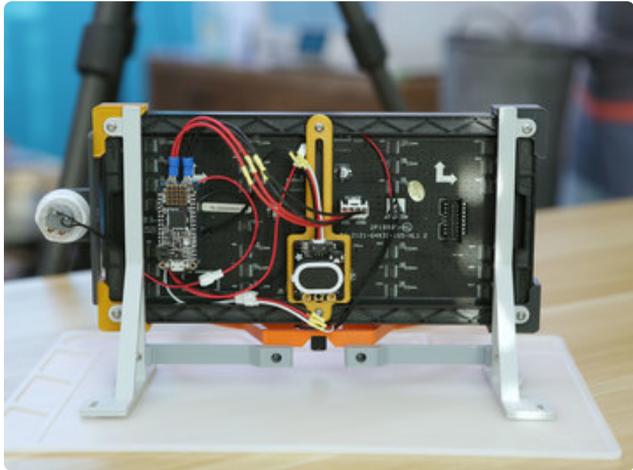
Mount Stemma Speaker

Place the Stemma speaker over the built-in stand-offs on the speaker mount. Insert M2.5 screws and use hex nuts to secure the PCB to the mount.



Secure Speaker Mount

Place the speaker mount over the mounting holes in the middle on the back of the display. Line up the mounting holes and insert two M3 x 6mm long screws to secure the speaker mount to the display. Reference the photo for best orientation.



Plug In Connections

Carefully make connections by plugging in JST cables to their respective counterparts. Take your time and make sure the connections are correct.



Final Build

And there we have it! The build is ready for testing. We suggest using mounting tack to temporarily secure the feet to a table top or work surface. The feet feature various mounting holes for securing to a frame or table top with hardware screws.

Take caution and be careful during gameplay. You are throwing a ball at a display!