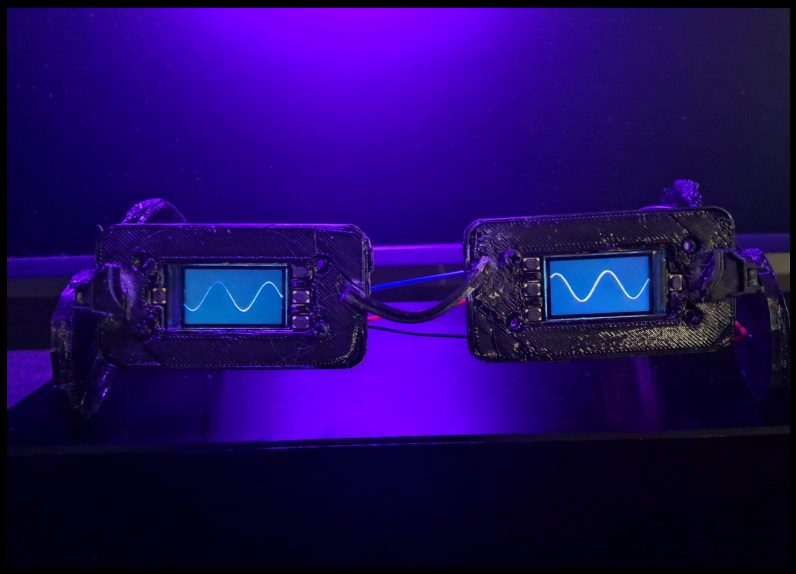
SINE WAVE GLASSES

Pierre H

Prototype_001

08/19/2025

Table Of Contents

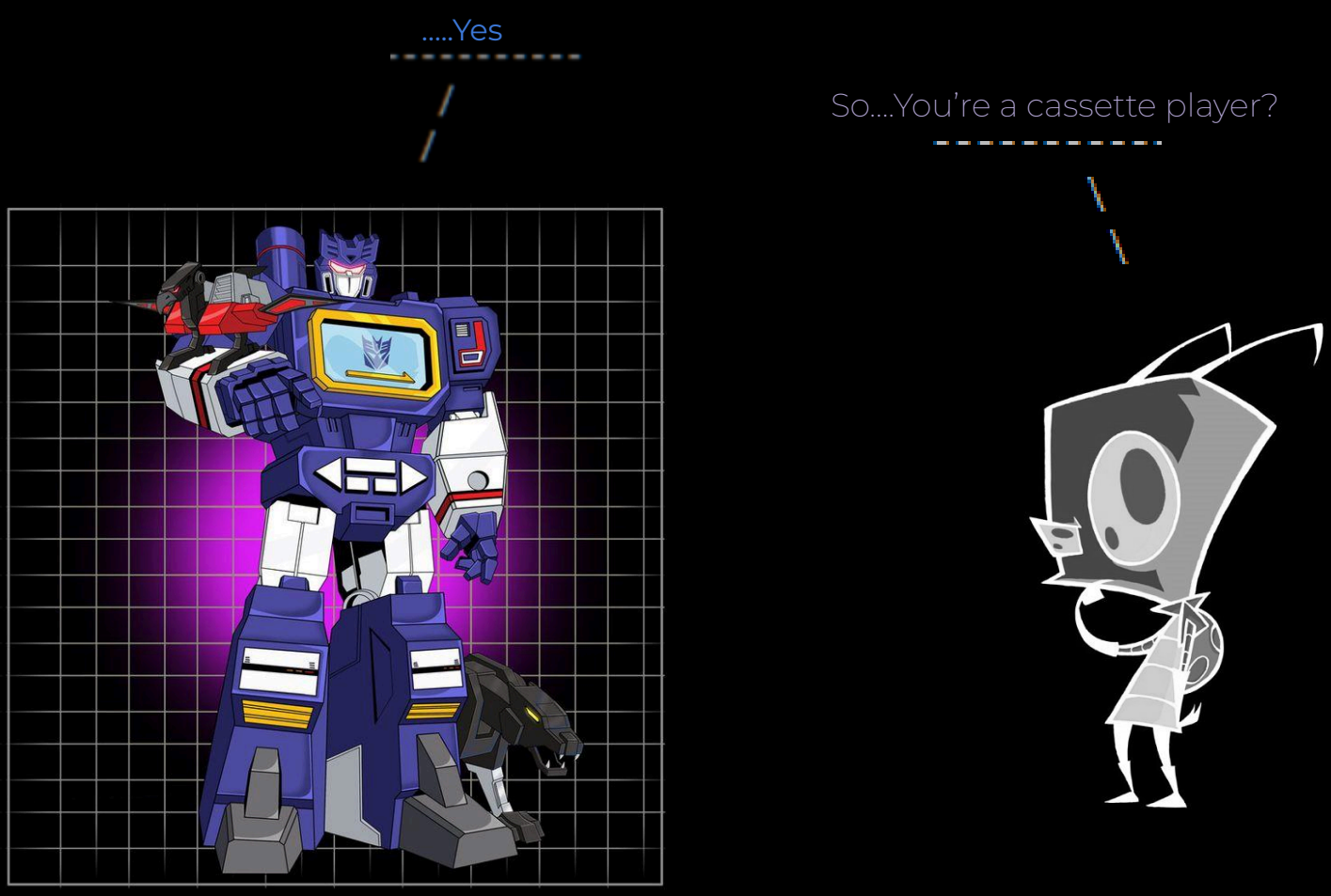


DISCLAIMER

You may or may not be visited by autobots. Make at your own risk.

Overview

This project came about as I was thinking of ways to cosplay as the Transformers character, Soundwave. In short, Soundwave is a Decepticon whose main role is Espionage and communications, largely due to his mastery of radio & electronic signals. These glasses are designed to display an oscillating sine wave. On the right side of the glasses, there is a sliding potentiometer you can adjust, controlling the amplitude of the wave. This project was a really new experience for me and I learned a lot, so I wanted to share my process for others that are interested. Through this guide, I hope you will build upon this idea and create your own work!



Parts Required

Adafruit ESP32-S3 Reverse TFT (2x)

https://www.adafruit.com/product/5691

Sliding Potentiometer 10 Ω

https://www.adafruit.com/product/4272

Lithium Ion Battery 3.7V

https://www.adafruit.com/product/3898

On/Off Switch

https://www.adafruit.com/product/805

Nylon M2.5 Screws

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

Movie Theater 3d Glasses

Soldering Iron

I tried to keep this relatively low cost, so there aren't that many parts

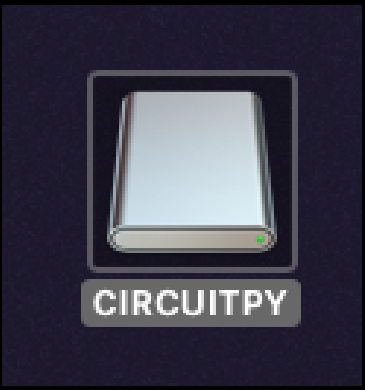
ESP32 Circuit Python Setup

To start, we need to install a TinyUF2 Bootloader on the ESP32 onto both boards, and then install the Circuit Python editor “Mu” for coding. The bootloader will change the flash storage to allow for Circuit Python files. The ESP32 is also a native USB! So when connected to your computer with a Type-C Data cable it will show up as it’s own drive.

Here is the link to the bootloader (it’ll be the .UF2 Download):

https://circuitpython.org/board/adafruit_feather_esp32s2_tft/

Once you’ve downloaded the file, double press the reset button on the ESP32 to enter boot mode, the drive should now say “FTHRSB3OOT”. From there you can drag the .UF2 onto the drive, after a few seconds the drive will disappear and then show up as, “CIRCUITPY”.

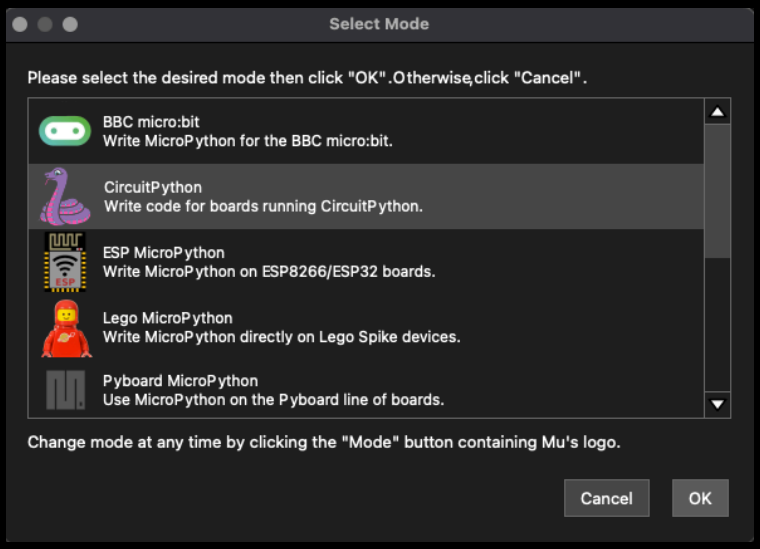


ESP32 Circuit Python Setup

Now to install the Circuit Python Editor. Download Mu from the link below

https://codewith.mu/

Once you've downloaded Mu, you'll see a window pop up with different modes, select the "CircuitPy" mode



IMPORTANT:

In Circuit Python, codes will run automatically when saved onto the "CircuitPy" drive. There are 4 different ways to save files, Code.txt, Code.py, Main.txt, and Main.py

When the main editing window loads, you're ready to code!

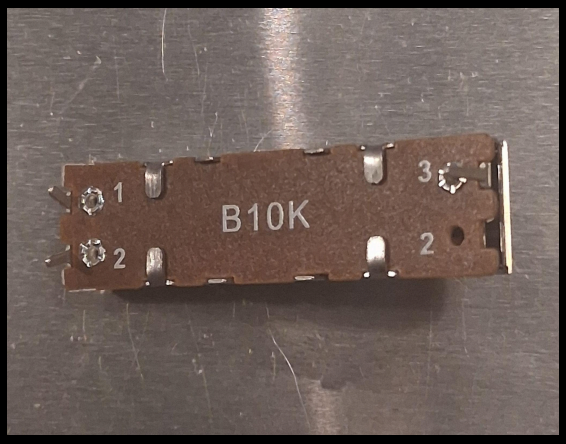
Potentiometer Wiring

Before we upload any code, we need to specify which Pin the sliding pot will be connected to on the ESP32 (A0-A5). This signal pin will control the size of the Sine Wave, in the code I chose Pin A4. Below is how to wire the potentiometer,

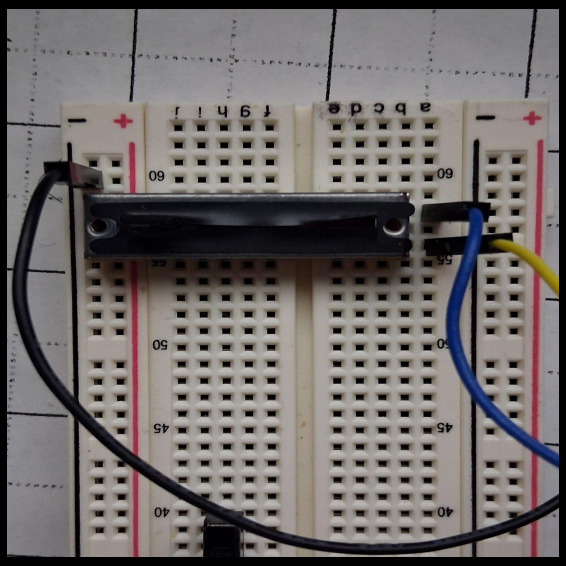
1: This is 3v/VCC

2: This is the signal pin

3: This is ground



Once you have determined which pin to use, place the potentiometer into the breadboard and set it aside.



Sine Wave Code

Copy and paste this code into the Circuit Python editor. Once it saves, you should see a small flickering, moving sine wave (yay!). If you want to change the color, simply go to this website and copy the hex code

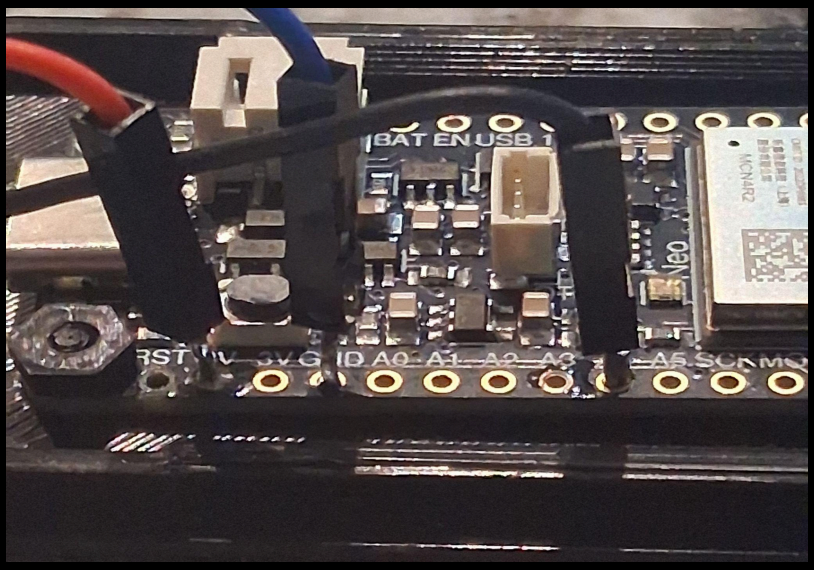
https://rgbcolorpicker.com/565/table

```
import time
import board
import digitalio
import displayio
import math
import analogio

potentiometer_pin = analogio.AnalogIn(board.A4)

display = board.DISPLAY
display.rotation = 0
splash = displayio.Group()
display.root_group = splash

# Set BOOT button on ESP32-S2 Feather TFT to stop wave
button = digitalio.DigitalInOut(board.BUTTON)
button.switch_to_input(pull=digitalio.Pull.UP)

# Create a display group to hold items
main_group = displayio.Group()
display.root_group = main_group

# Define sine wave parameters
offset_y = display.height // 2
frequency = 0.05  # Adjust for more or less waves on screen
phase_shift = 0

# Create a bitmap and palette for drawing
bitmap = displayio.Bitmap(display.width, display.height, 2)
```

```
palette = displayio.Palette(2)
palette[0] = 0x000000  # Black screen
palette[1] = 0xad61cf  # Light Purple Wave

tile_grid = displayio.TileGrid(bitmap, pixel_shader=palette)
main_group.append(tile_grid)

while True:

    # change amplitude according to pot value
    pot_value = potentiometer_pin.value
    amplitude = (pot_value / 65535) * 35  # map pot to a different range
    # time.sleep(0.05)

    # Clear the bitmap
    for x in range(display.width):
        for y in range(display.height):
            bitmap[x, y] = 0

    # Draw the sine wave
    for x in range(display.width):
        y = int(amplitude * math.sin(frequency * x + phase_shift) + offset_y)
        if 0 <= y < display.height:
            bitmap[x, y] = 1  # Set pixel to white

    phase_shift += 1  # Animate the wave
    time.sleep(0.01)  # Control animation speed
```

ESP32 Wiring

Unplug the ESP32 and prepare your soldering iron. You don't need the pin headers that come with the display so you can just store those somewhere. We'll begin by soldering the 3V, Ground, and Signal pins. Insert your wire into the pin so it's standing straight up, then apply solder until the connection is secure. Once you're done it should look like the photo below,



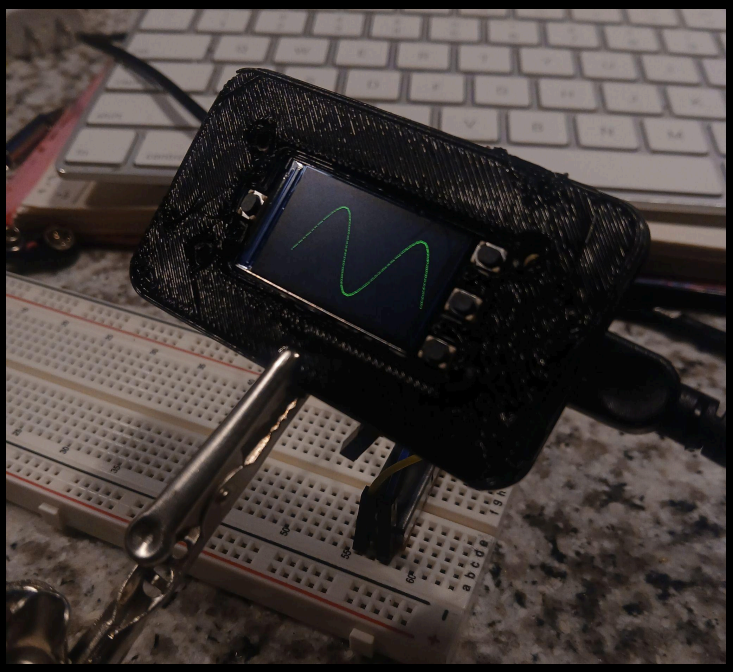
Now plug the wires from the esp32 into your breadboard,

ESP32 Ground —> Ground Pin of Potentiometer

ESP32 3V —> Positive pin of Potentiometer

ESP32 A4 —> Signal Pin of Potentiometer

Now as you adjust the potentiometer, the wave should change on screen!

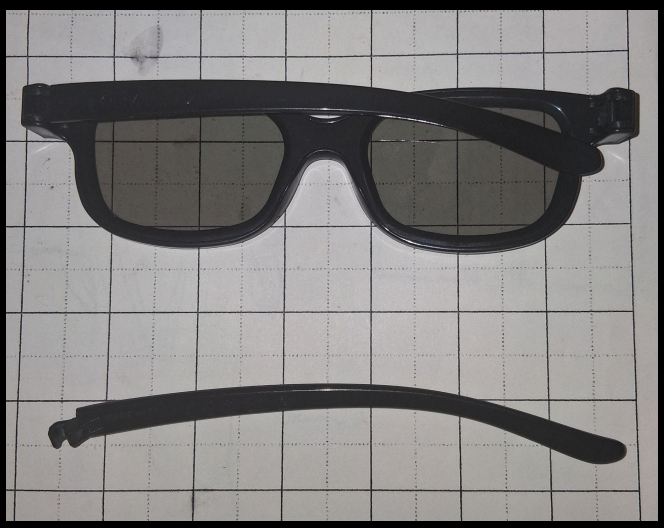


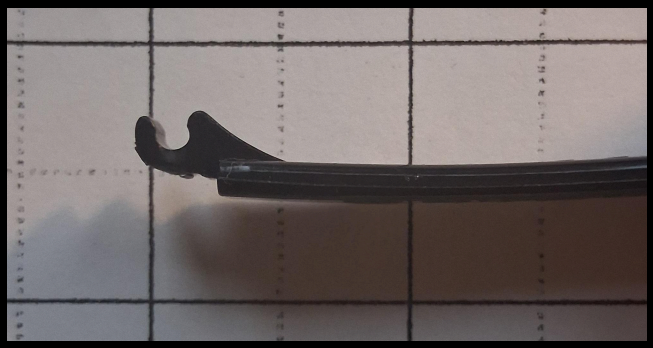
3D Printing/Assembly

The 3d files for the glasses are linked below:

https://github.com/Riff-RXff/Sine-Wave-Glasses

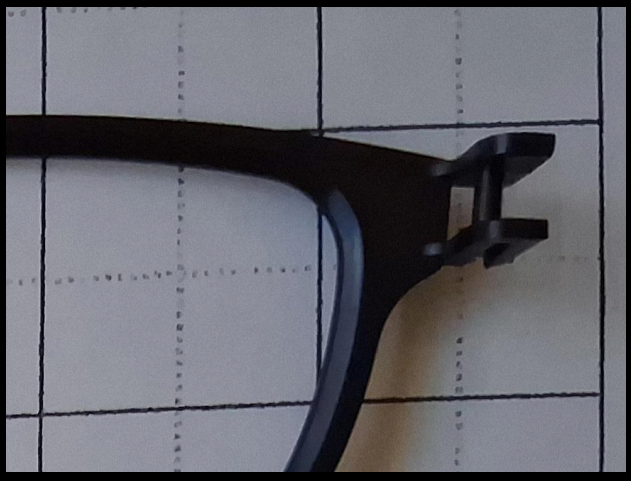
Once all parts are printed, the assembly process is pretty simple. Place the ESP32 Into the frame so that the USB-C port is facing the small opening, and secure it in place with the M2.5 screws (if your file didn't print right, you may only be able to attach 2 screws, which happened to me).

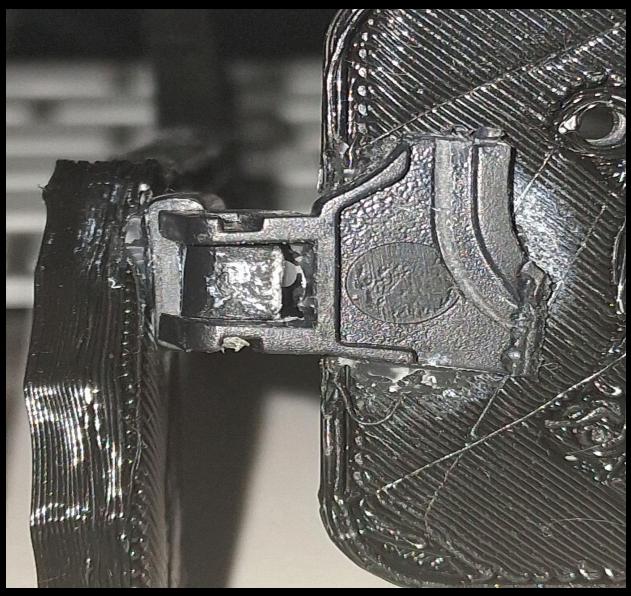




Next we will take the 3d glasses apart and salvage the hinges. Carefully rip the lens arms off of the frame, cut an inch of the connector piece off (it looks like a tiny hand), and then superglue it onto the music note arms.

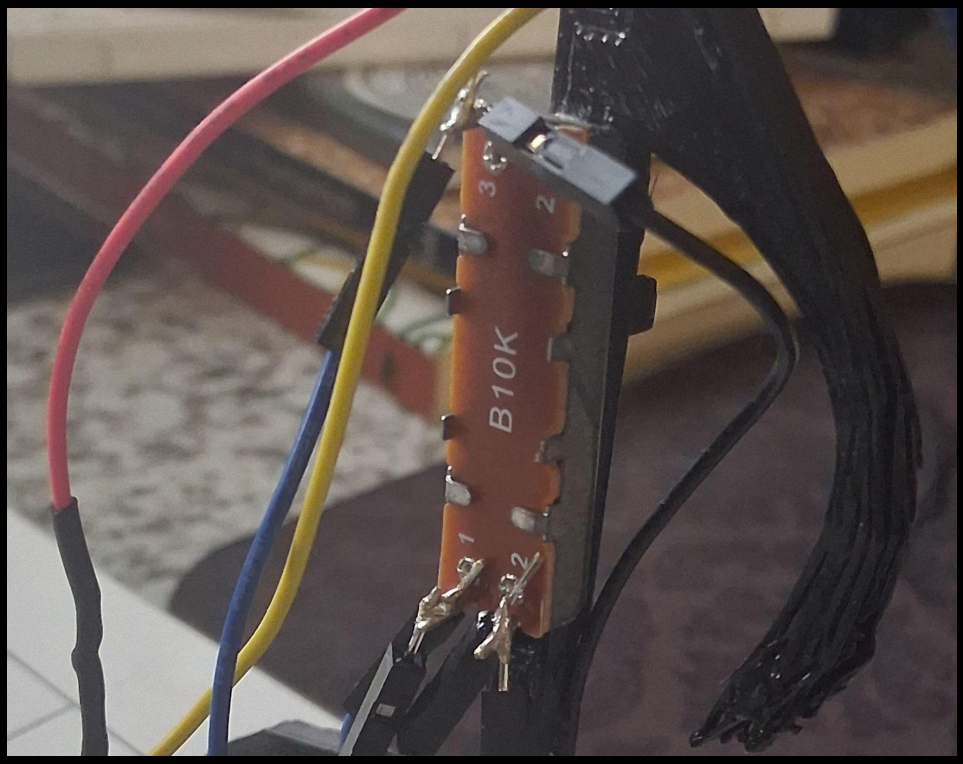
Next cut each hinge off (make sure to keep the right angle intact) and superglue it onto the middle of the frame as shown below



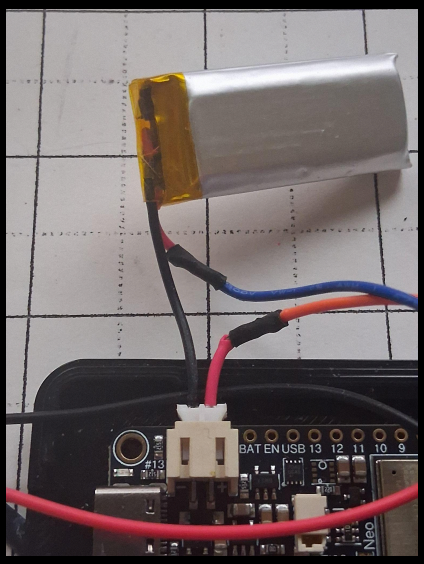


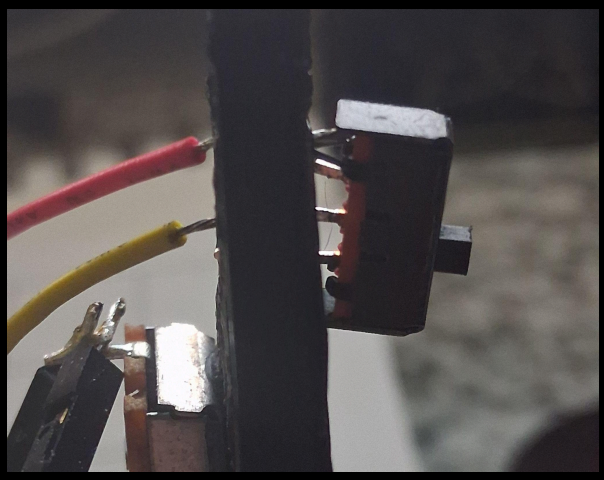
Now you can easily attach the arms

After that we can move on and begin wiring the rest of the components



Superglue the sliding pot onto the frames, then solder each GND, 5V, and A4 wire to it’s respective area on the sliding pot, now the sliding pot will affect both screens. After that we can move on to wiring the on/off switch





(I wired my switch backwards on accident, just flip the orientation)

Carefully cut the positive wire, then solder wires to the first 2 pins of the switch

Final Thoughts

Now that everything is assembled, power it on and vibe out.

Is it perfect? Not in the slightest, I wasn't productive every day and even set this project aside for 2 weeks to work on something else. During the design and building process of this project, I kept reminding myself the most important aspect of any idea is to make it exist first, as long as it's tangible I can always make it pretty later. That guiding thought kept me focused more than anything else, without it I would've been at the whim of my fickle motivation and abandoned this project at the first instance of discomfort. All in all, I hope this guide helps!