



## ESP32-Powered Tabletop Kinetic Sand Drawing Robot



by aidenvigue

In 2020, back when I had just started my sophomore year of high school, I built a very basic sand plotter robot that drew patterns in sand. Three years later, I wanted to use the skills that I have taught myself to revisit the design in order to be able to take it to college with me to Purdue in a couple of weeks! You can read more about the original robot [here](#)

The original inspiration came when I saw an Instagram ad for the SANDSARA Indiegogo campaign back in late 2019/early 2020.

This redesign has been in the making for the past six months. I recently wrote it up on my main blog [here](#), however it was more of a showcase. I will attempt to explain in detail how to construct it, but feel free to leave a comment or contact me if you would like any more information :)

### Supplies:

3D Printer

ESP32 DevKit V1

2x TMC2208 -or- TMC2209 stepper drivers

2x mini nema 17 pancake steppers

2x hall effect endstop (generic ones will work fine)

TSL2561 light sensor

Panel mount USB-C and DC power jack

Custom PCB

4 inch lazy susan bearing

Assorted countersunk M3 screws

SK6812 RGBW led strip (1m) I used 144 leds/m

M3 heat set inserts (4.5mm OD)

Assorted 2,3,4,5 pin JST connectors and SMT sockets.

400MM GT2 timing belt w/ pulleys.

<https://vimeo.com/844274357?share=copy>









---

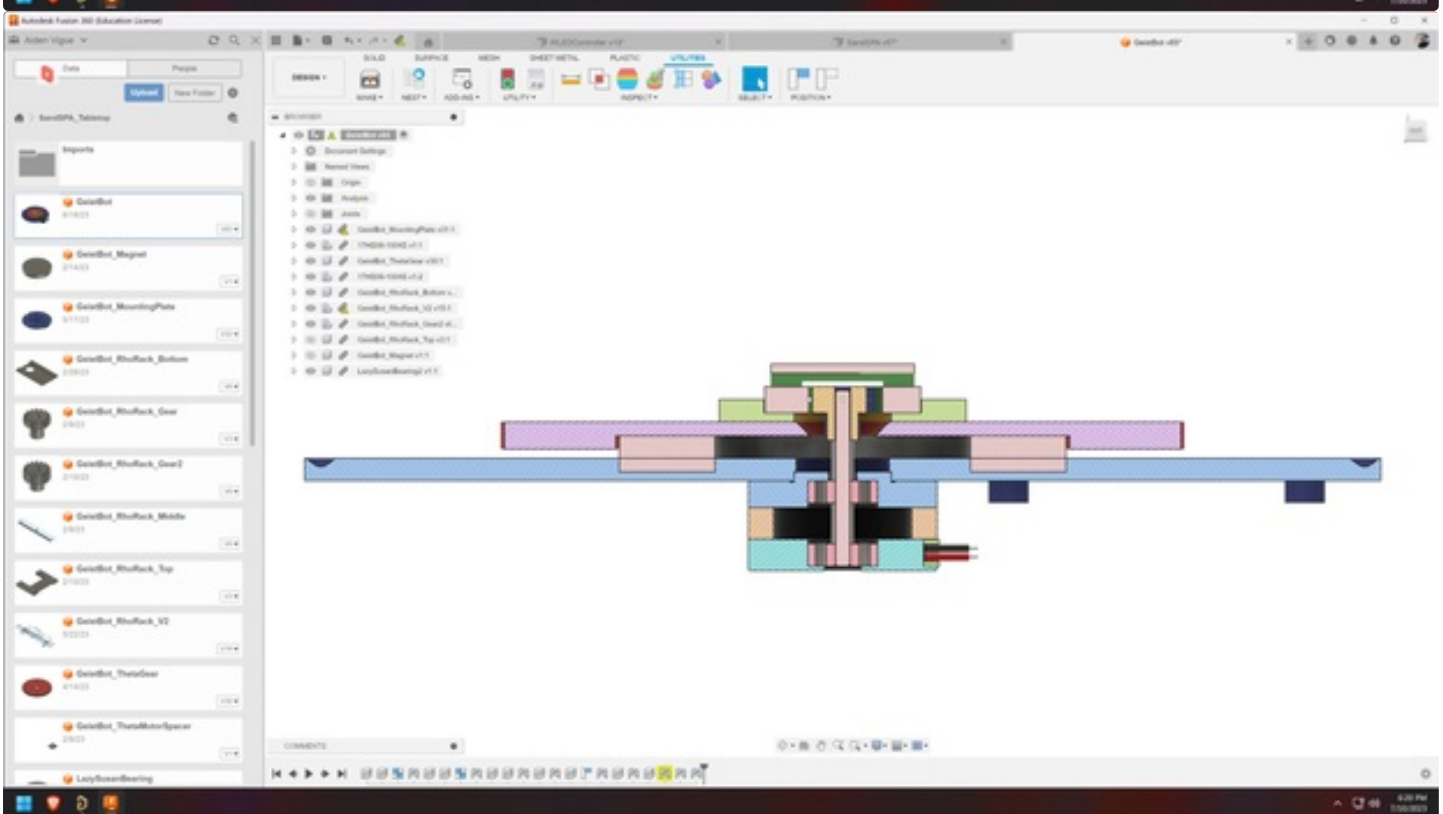
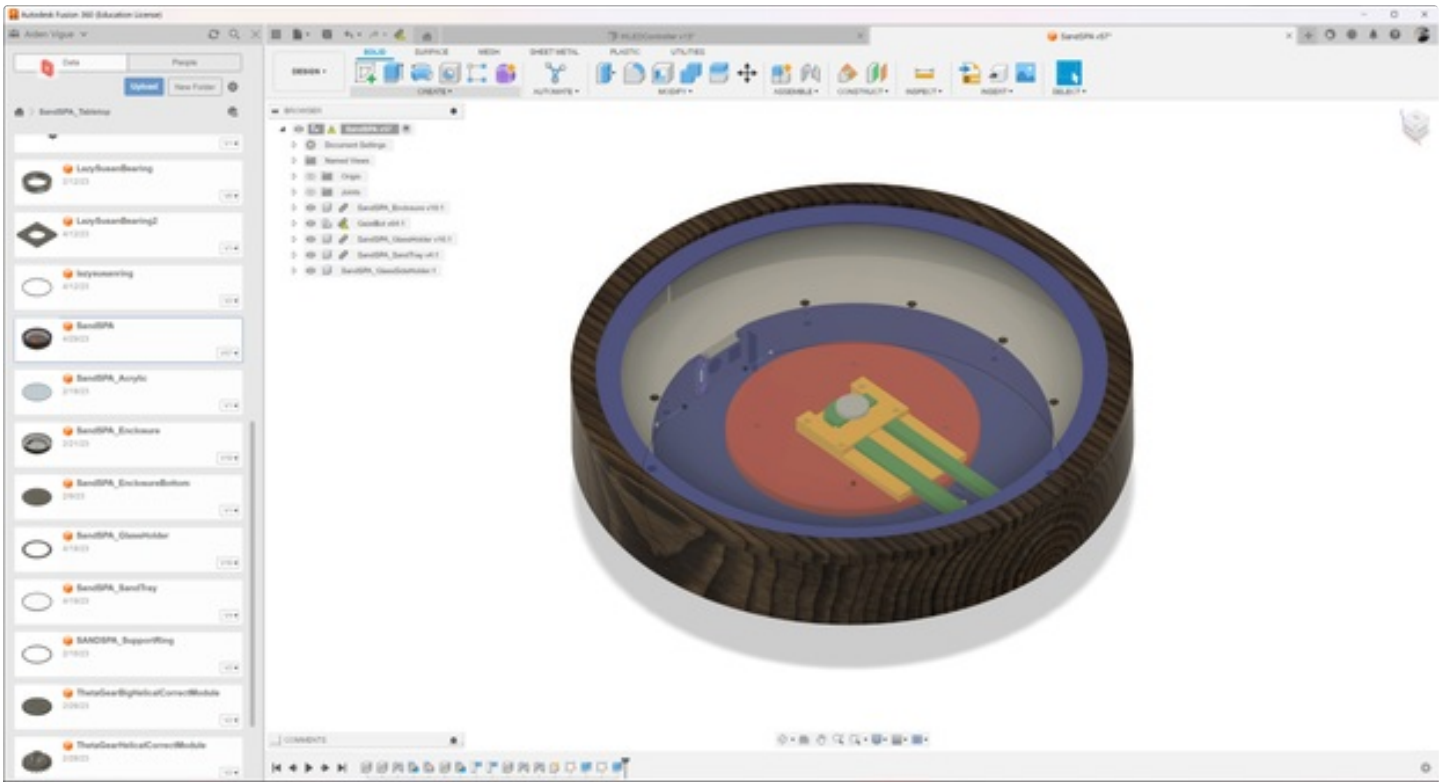
## Step 1: Fusion 360

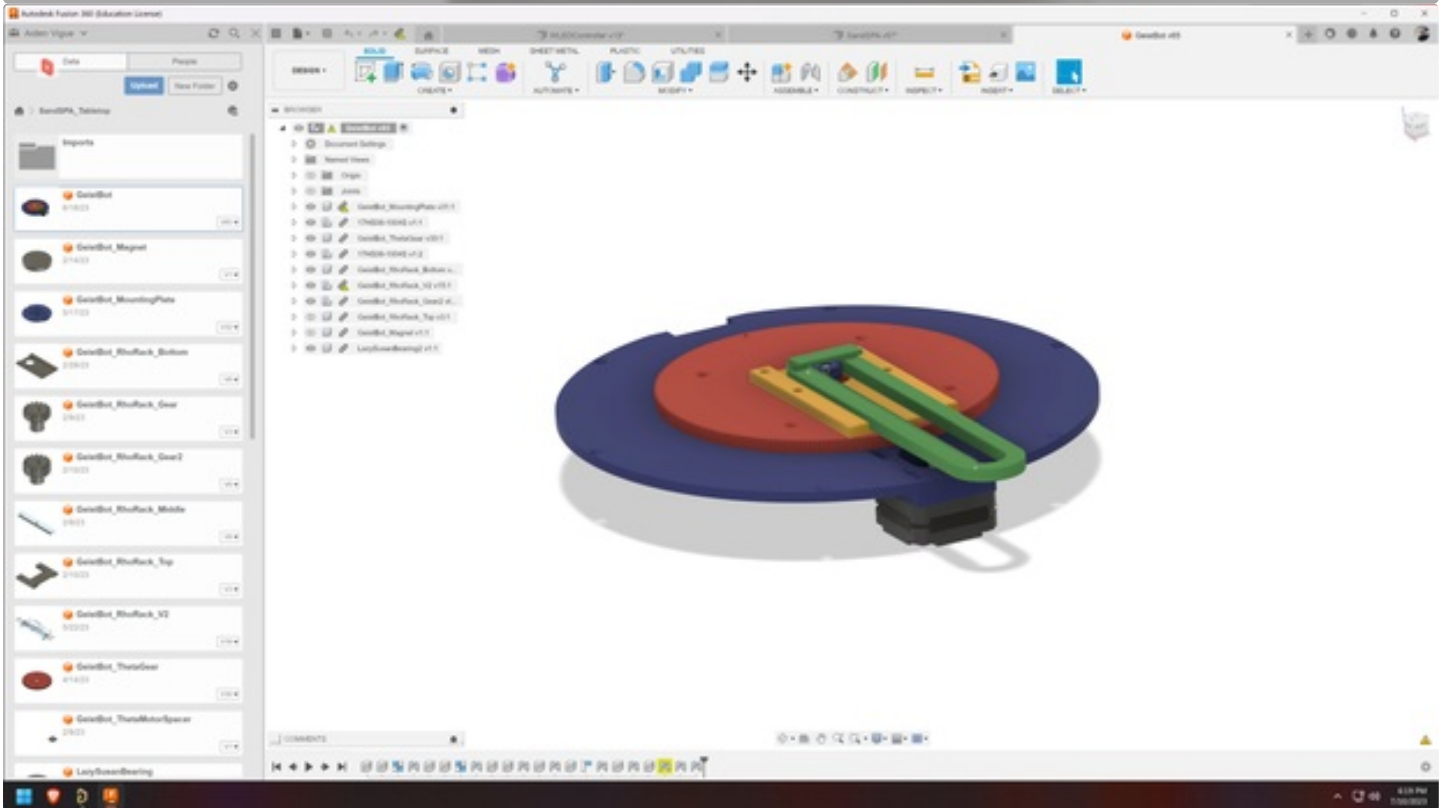
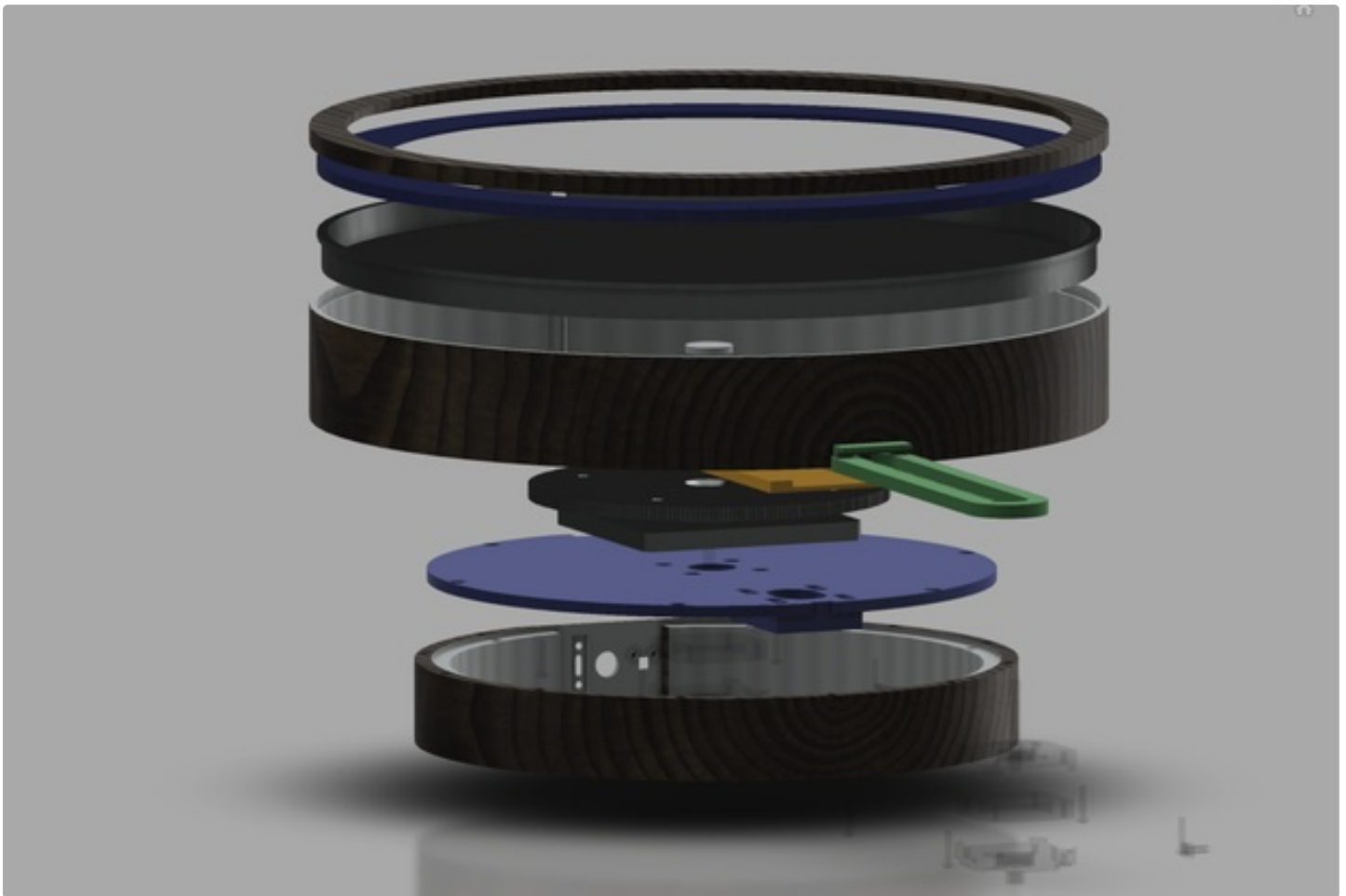
I designed this kinetic sand robot to be extremely slim, and with everything printed, it is under 3in tall! There are a lot of overlapping parts, mostly around where the large theta gear attaches to the base plate. The entire robot was designed from scratch in Fusion 360, and I've shared them online through Fusion Team ([here](#))

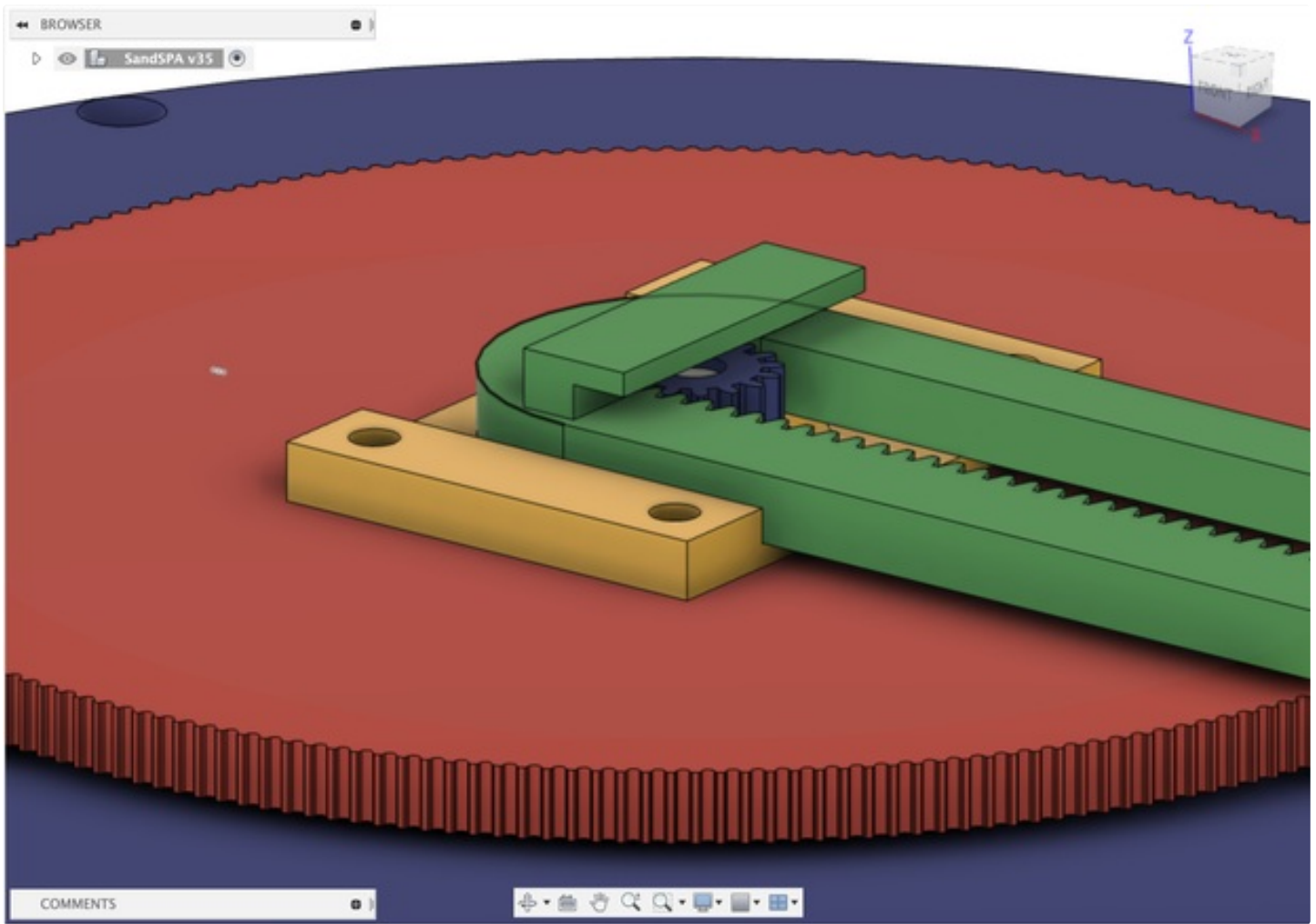
Yes, Fusion 360 does say that this is version 55 of the mechanism... I spent longer than I want to admit during this phase of design :)

I started by sketching the rough shape of the enclosure, and then split it up along the large overhang to ensure printability. I then built the robot baseplate, modeled the lazy susan bearing, and the central gear stackup.

I've attached both a cutaway view and an exploded view of the robot assembly for more detail into how it goes together.







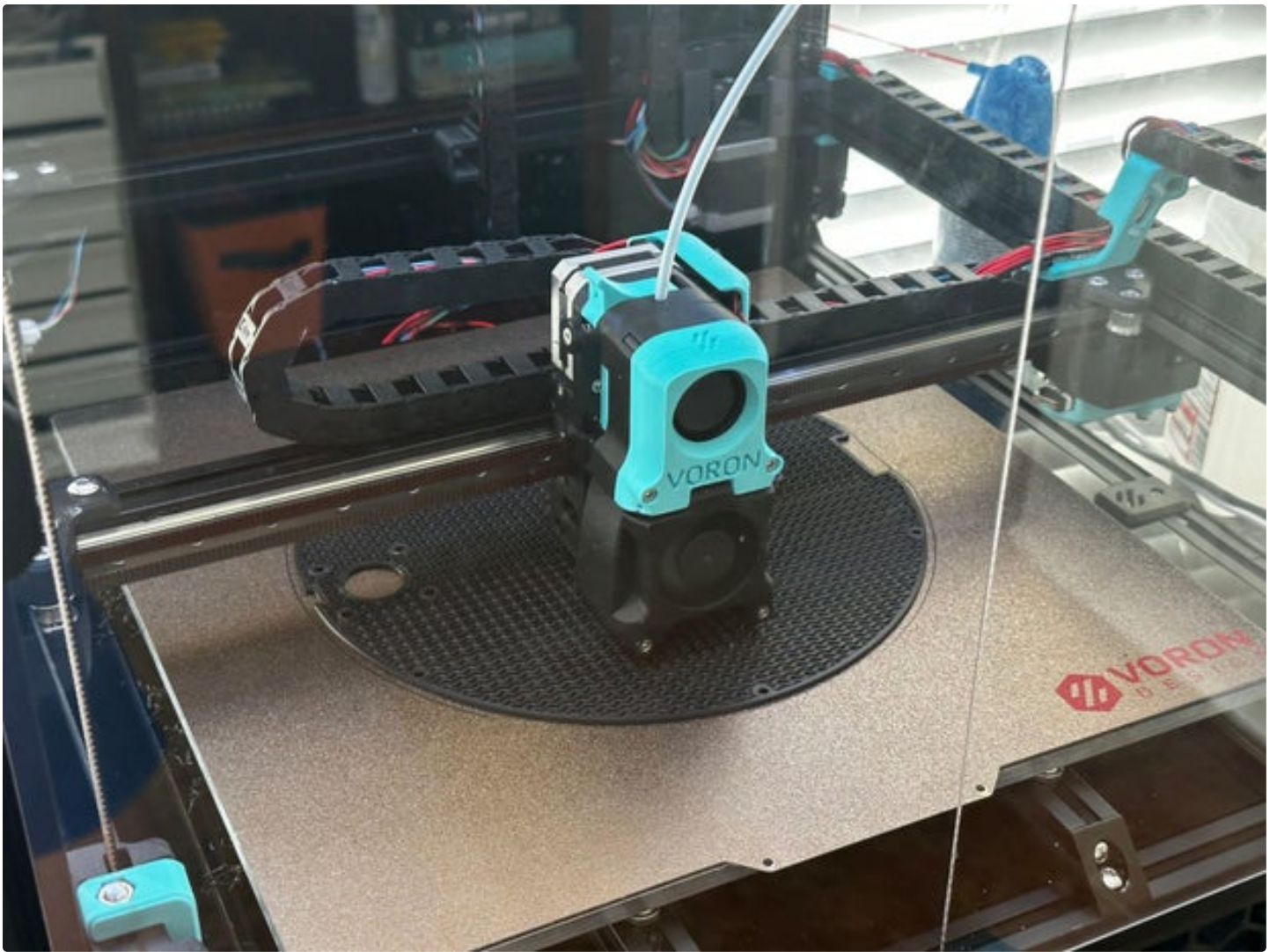
## Step 2: 3D Printing!

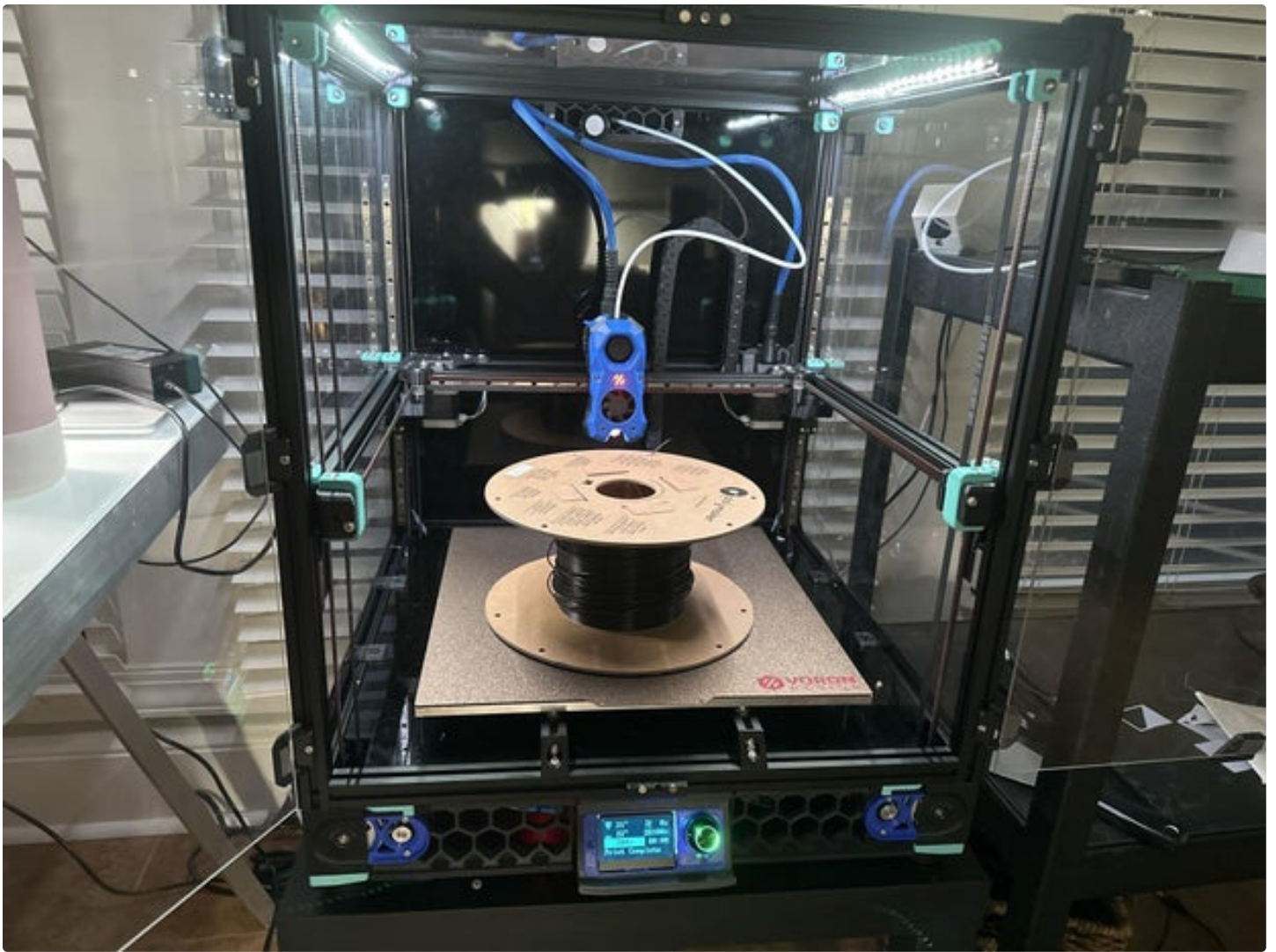
I printed everything in black Polymaker ABS using 30-50% gyroid infill depending on the part. Usually, the internal components got 50%, and I just printed the enclosure with enough shells to make it solid. I'd estimate that I used almost all of the 3kgs of filament I bought for this project. I did a few printer upgrades during this project as well :p

PLA would've fared better in terms of less warping, however I wanted the structural properties of ABS.

I've attached a photo of me test-fitting the linear axis and it's guide rails.









---

### Step 3: Robot Assembly

I added 4-pin JST connectors to the stepper motors, a 5-pin connector to the TSL2561 light sensor, and 3-pins to both magnetic endstops and the LED strip.

I then installed the theta stepper motor with countersunk M3 screws to the robot base plate. Then, I put the GT2 pulley on the motor, and screwed the lazy susan bearing into both the large central theta gear and the lower base plate using M3 screws. The bearing snaps right into place and shouldn't wobble. In the Fusion assembly, there is an optional spacer that can be used if it does. The linear axis and guide rails screws directly into the top of the theta gear with some heat-set inserts.





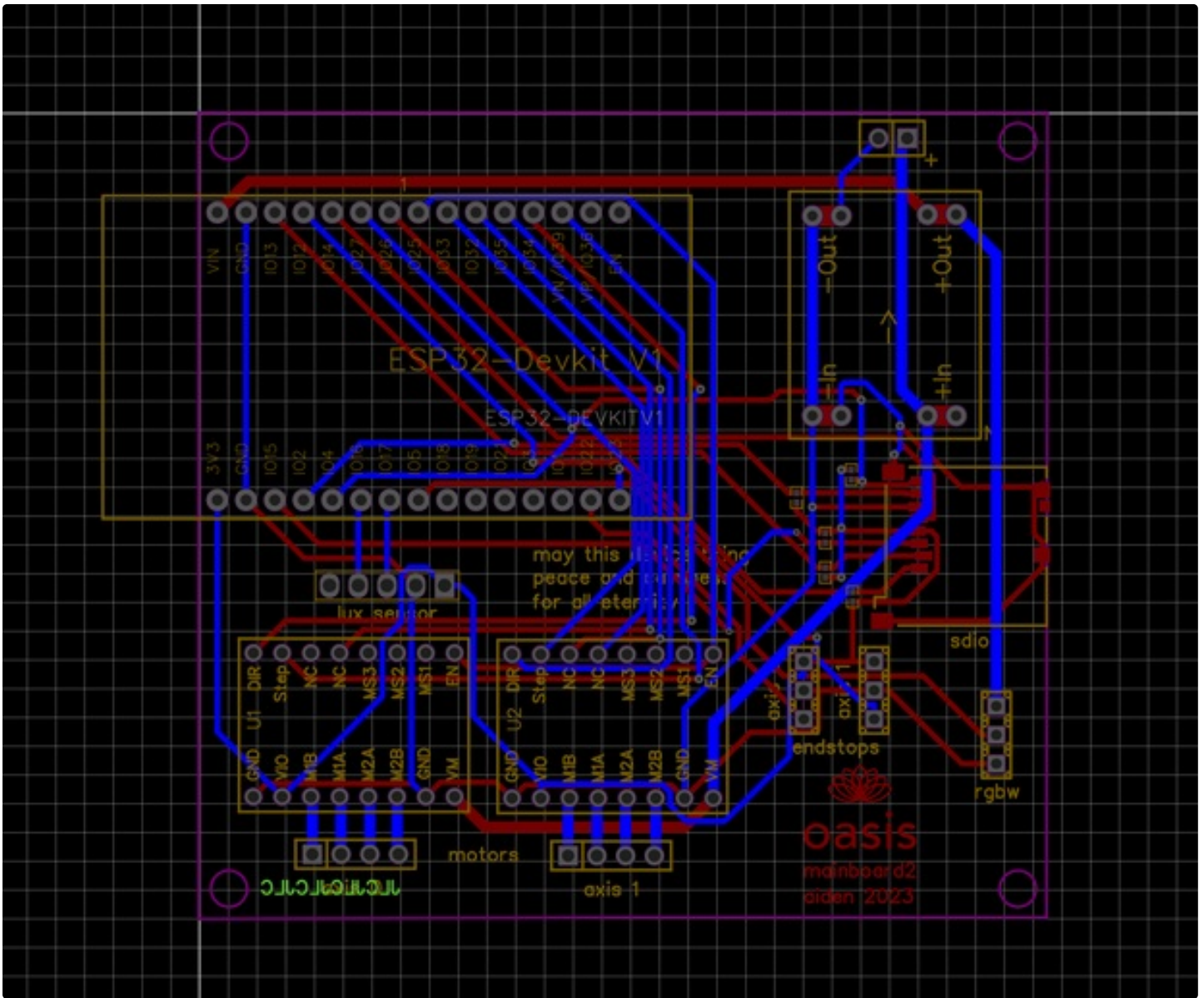
---

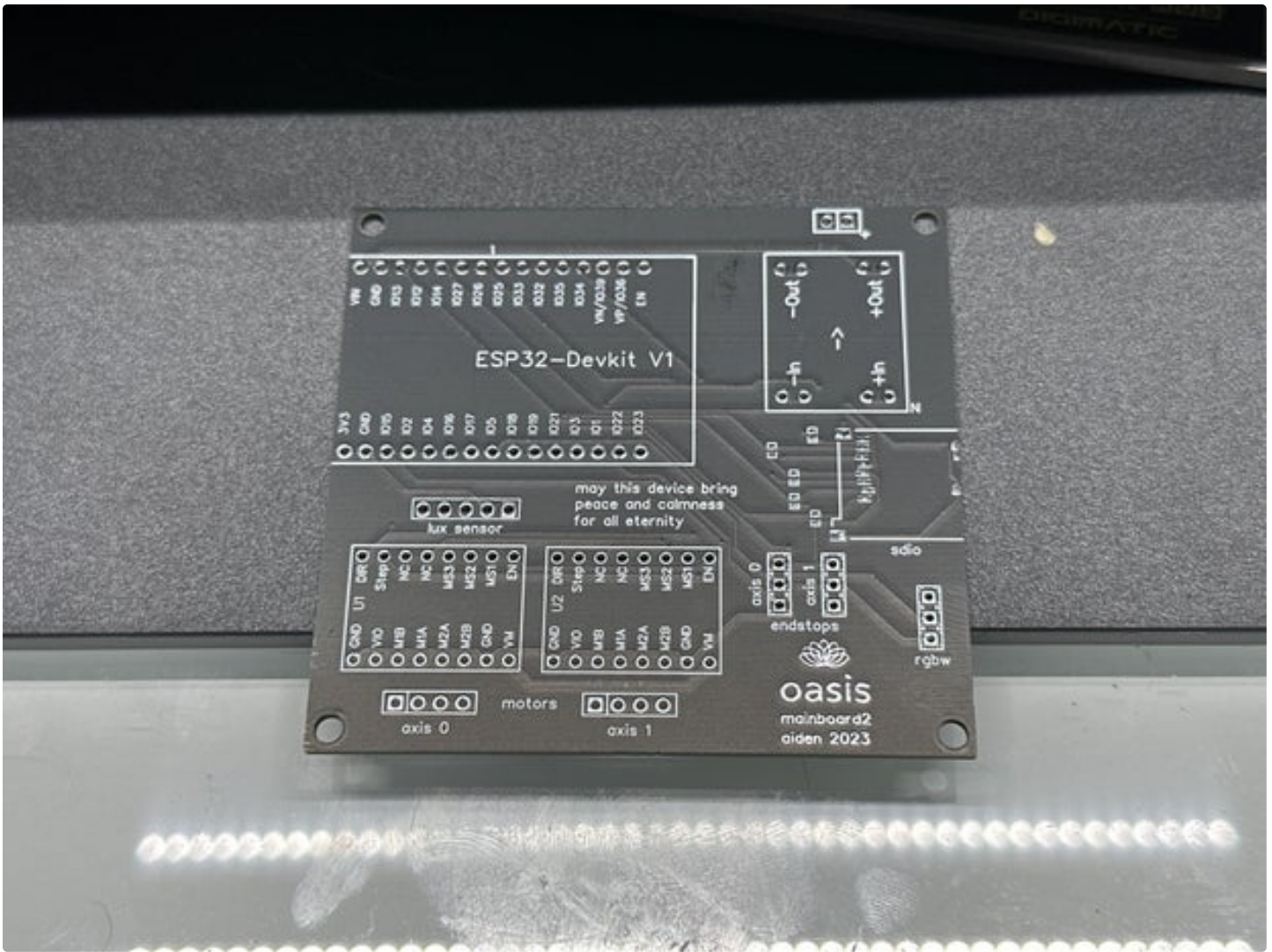
## Step 4: PCB Design

I designed a very basic breakout board for all of the components using the standard version of EasyEDA and had it manufactured with a black soldermask.

I actually ended up changing the name to "Tranquil" when I was writing the web interface, I just never really wanted to re-order the boards just for some text.

The design is available [here](#)



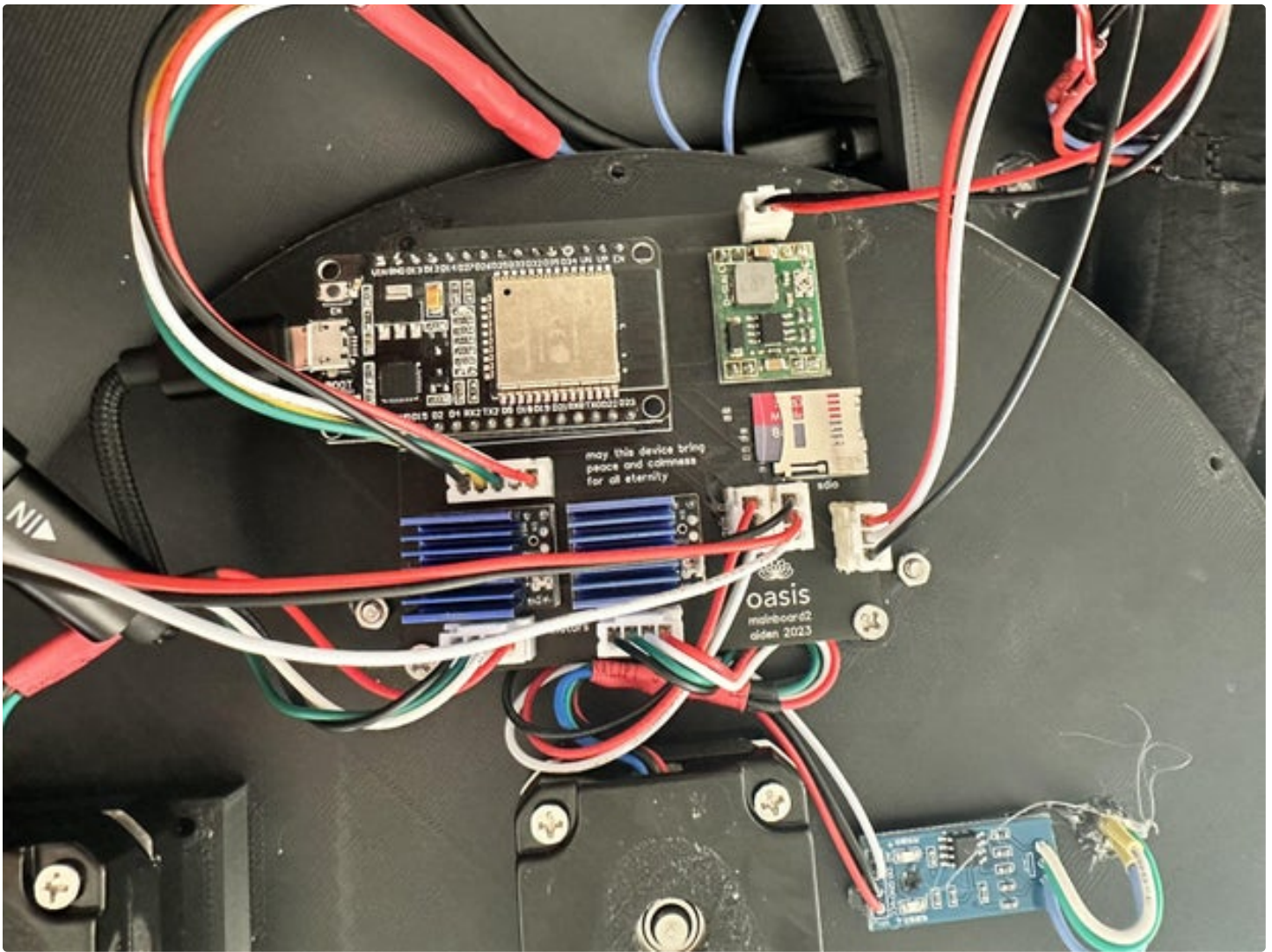


## Step 5: PCB Assembly

I simply soldered down the motor drivers, ESP32, and the JST connectors. I added an optional SD card slot that is compatible with MPN#47352-1001 ([Mouser](#)) I hand-soldered the connector with a fine point tip and some no-clean flux.



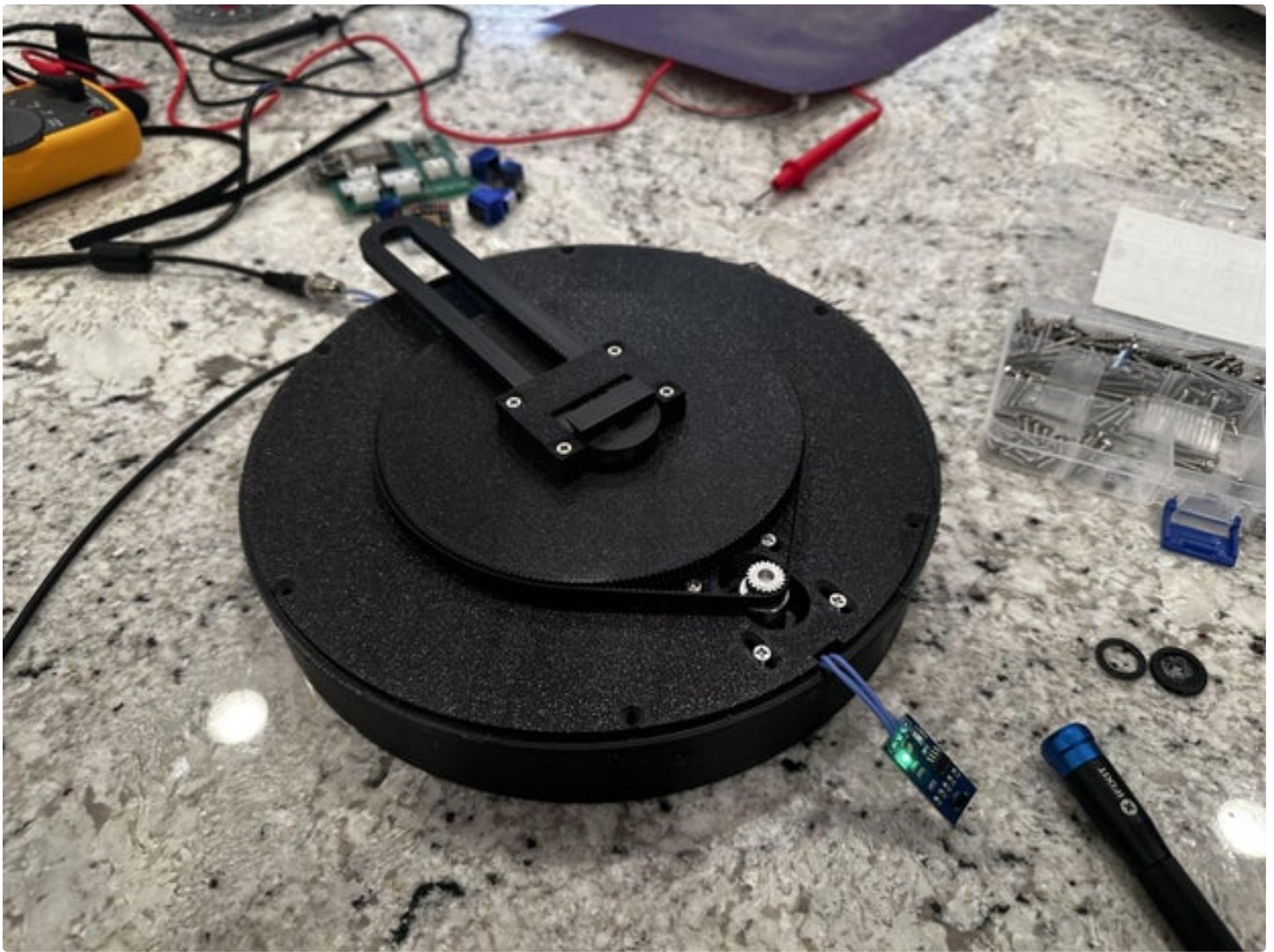




## Step 6: Connect It All Up

I installed one of the magnetic endstop sensors right into the tiny hole offset from the center to sense when the theta gear arrived at a known position

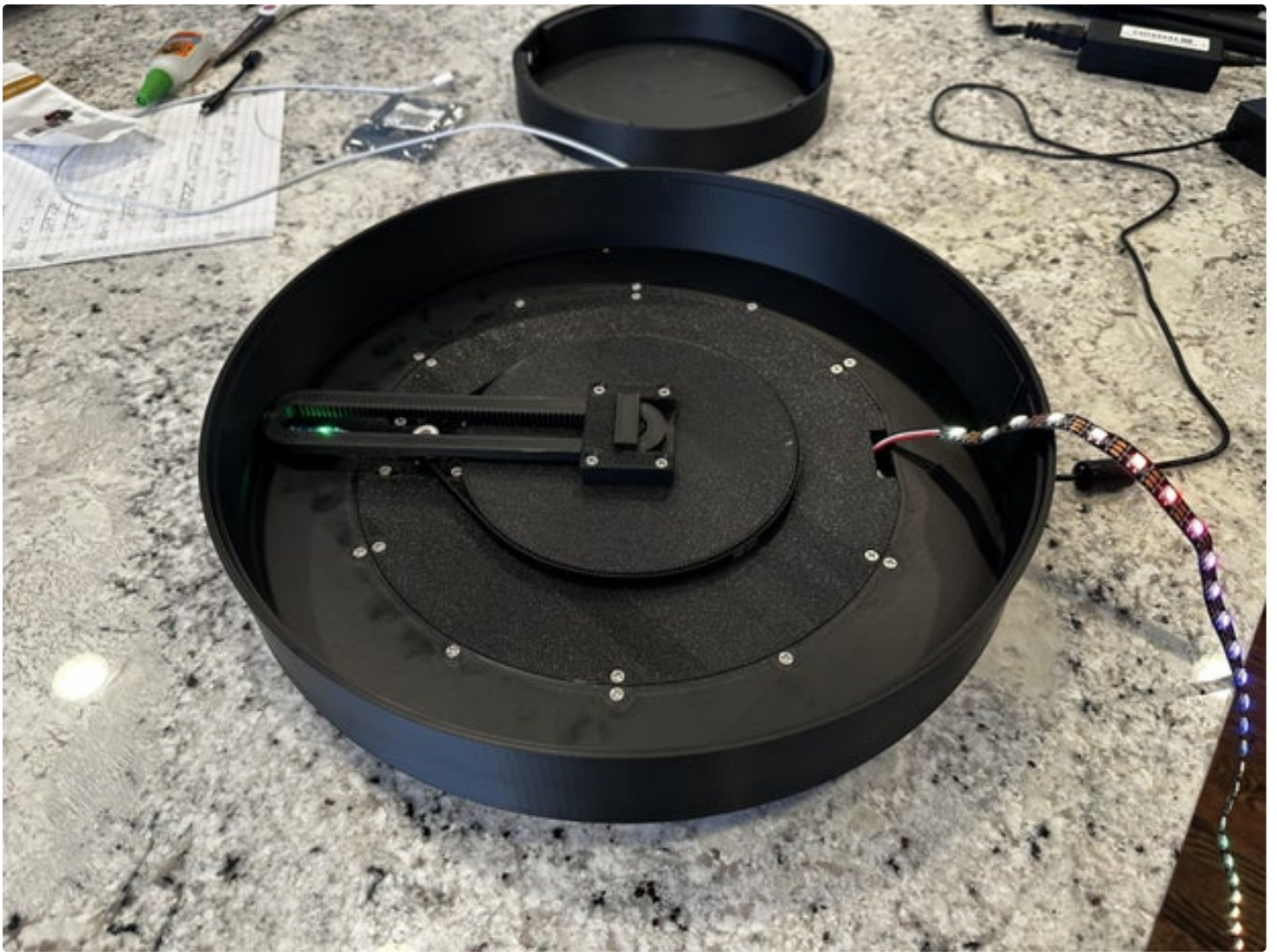
The fully assembled robot base plate simply slides into the center of bottom half of the enclosure.



---

## Step 7: Assemble Enclosure

Place top half of enclosure on bottom half. Screw down with heat set inserts & 12 10mm M3 screws.





---

### **Step 8: Add Sand**

I took some generic craft felt I had at home and spray glued it to the bottom of the sand bed to make it super quiet. Then, I just dumped some Home Depot sand and spread it around with a butter knife.



---

### Step 9: Lights! Camera? No, Just Lights..

I stuck the LED strip on the sides of the sand bed with some double sided foam tape, and ran the 3 wires (5V data and ground) through the tiny slit on the side, down to the main control board.



---

## Step 10: Compiling the Web Interface

The web interface was written from scratch with a mix of TypeScript and Vue 3. State management is done with Pinia and data fetching is done with a set of Axios request interceptors.

The GitHub repository is [here](#).

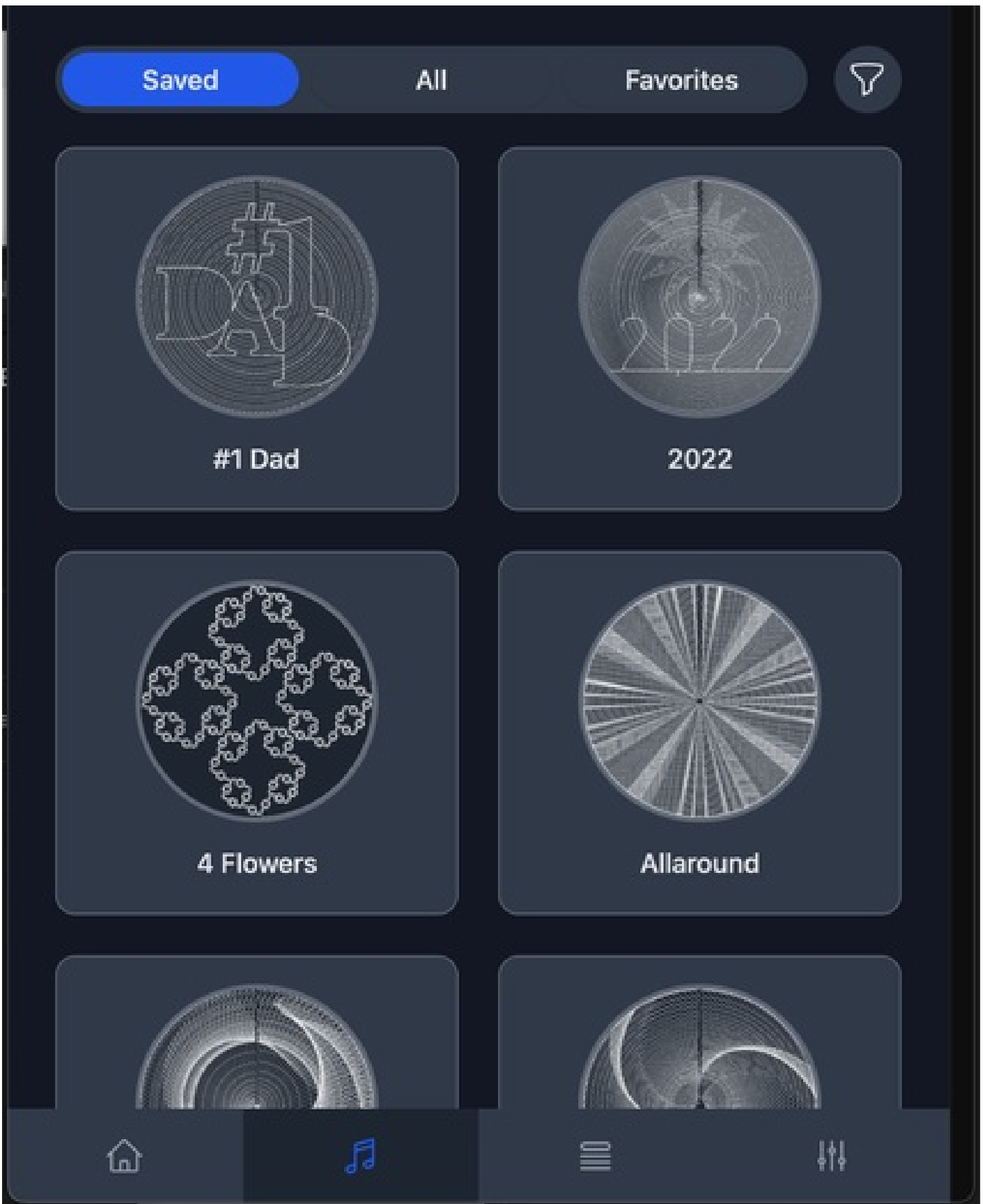
Simply clone and build the project to continue,

You can build the project on a \*nix machine by running (change command for equivalent on Windows machine)

```
yarn && yarn build
```

Save the generated build files that are output in the ./dist directory.



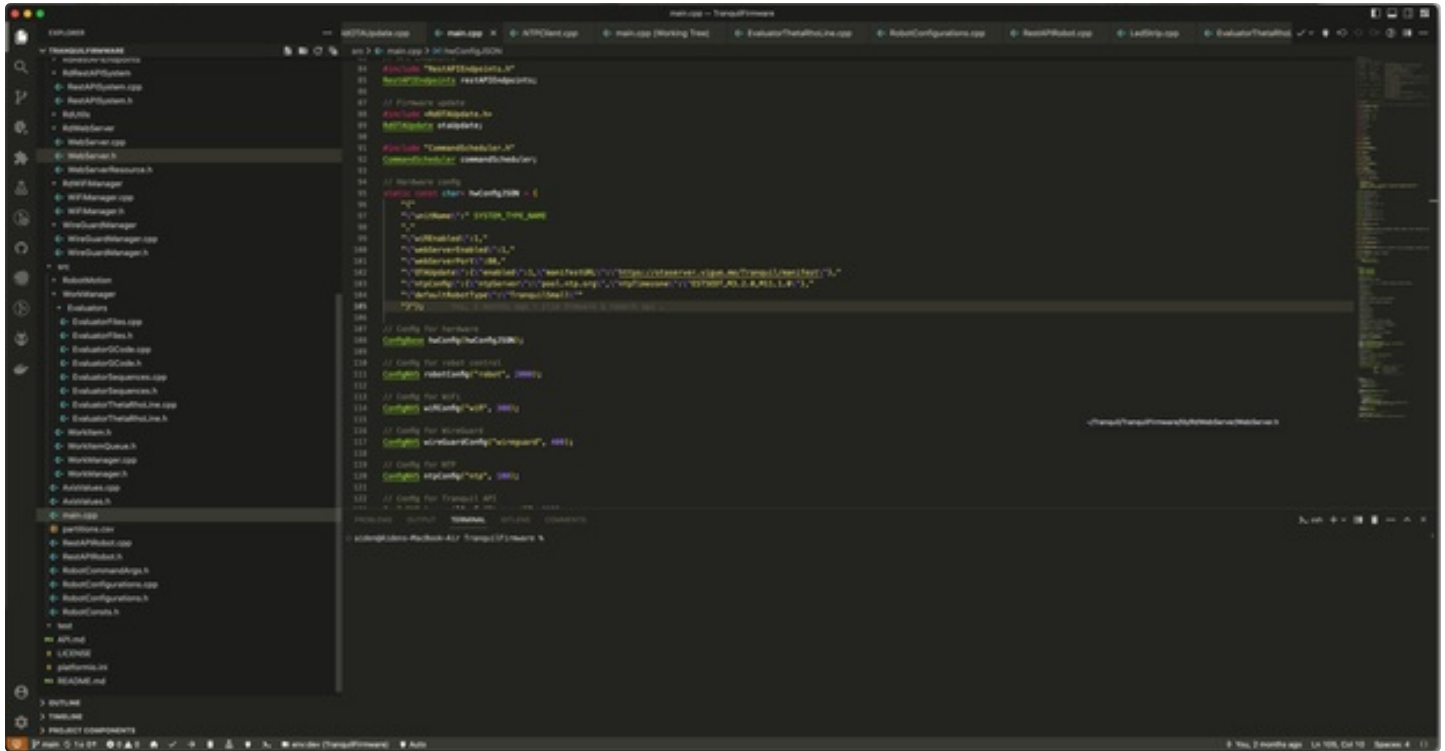


## Step 11: Compiling the Device Firmware



The firmware has been re-written from almost the ground up and has been in the works for at least a year now. It's a normal PlatformIO project and can be downloaded from [here](#). Drag the built web assets to this directory and PlatformIO will automatically move them to the ESP32's SPIFFS partition.

The patterns featured are not provided and can be downloaded from the open internet. The web interface simply lists what patterns are installed on the device.



---

## Step 12: Configuring Device

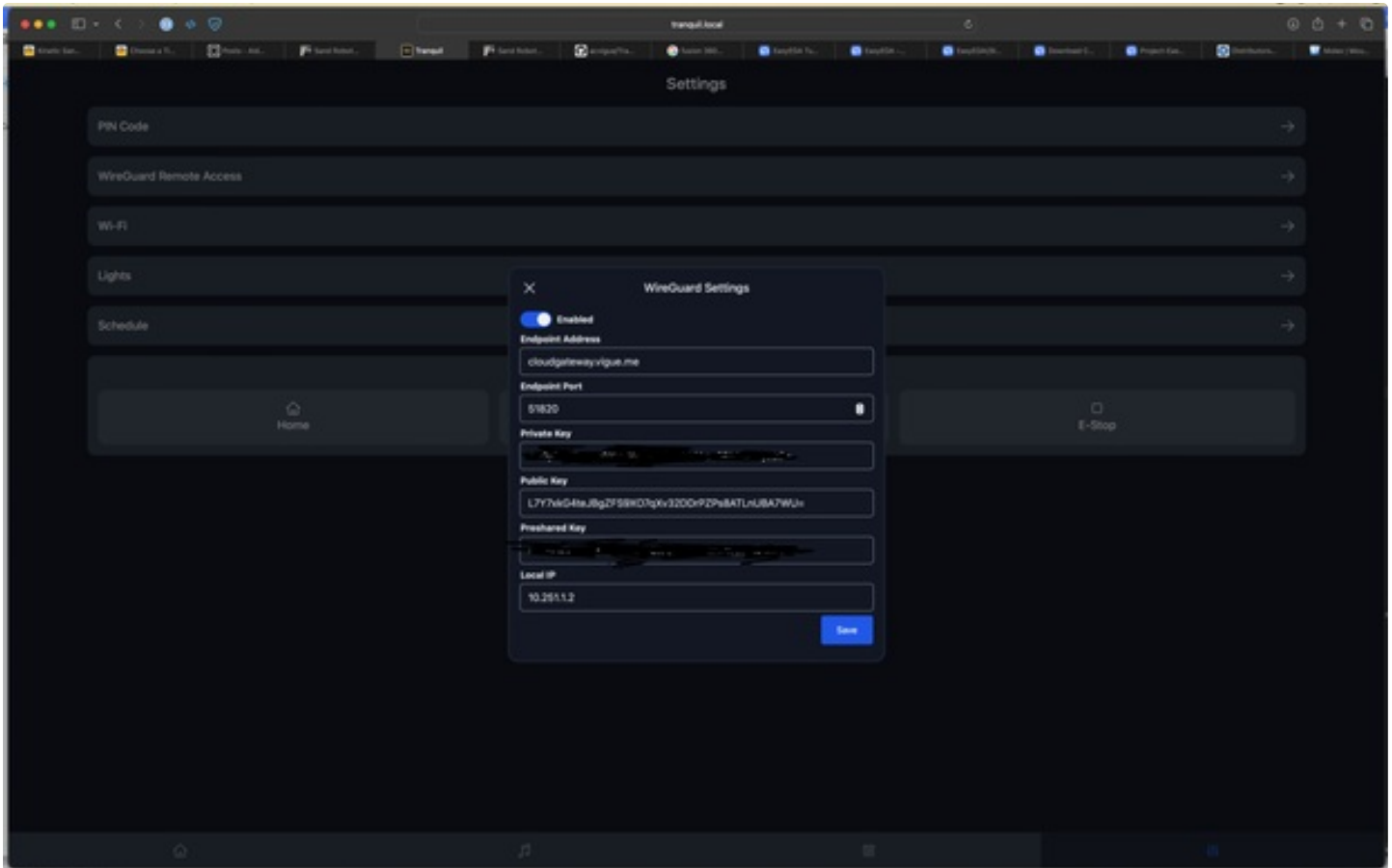
Once the firmware is installed, the device will setup it's own access point named "Tranquil" where all initial configuration is done. You can connect it to your Wi-Fi, setup OTA updates, and even (in my case) set it up as a WireGuard client. (see next step)

---

## Step 13: Wireguard Client

You can go to the settings tab of the device and insert the relevant keypair and endpoint information for the robot to join a remote WireGuard subnet.

I make use of this as the college that I will be attending in the fall implements client isolation on their network, and without this, I would have no way to control the robot.



## Step 14: Enjoy!

Thank you so much for reading! If you want to build your own, but are stuck somewhere, please let me know in the comments and I will definitely try my best to help!

I've now built two sand drawing robots and have gained so much experience in designing for printability and ensuring that the firmware that I write is stable!

You can see more of my projects on my blog: <https://vigue.me>, or follow me on GitHub [@acvigue](#)







Wow, I like this project. I especially like that it's unusual and so original.  
Thank you for sharing this project.



Thank you!