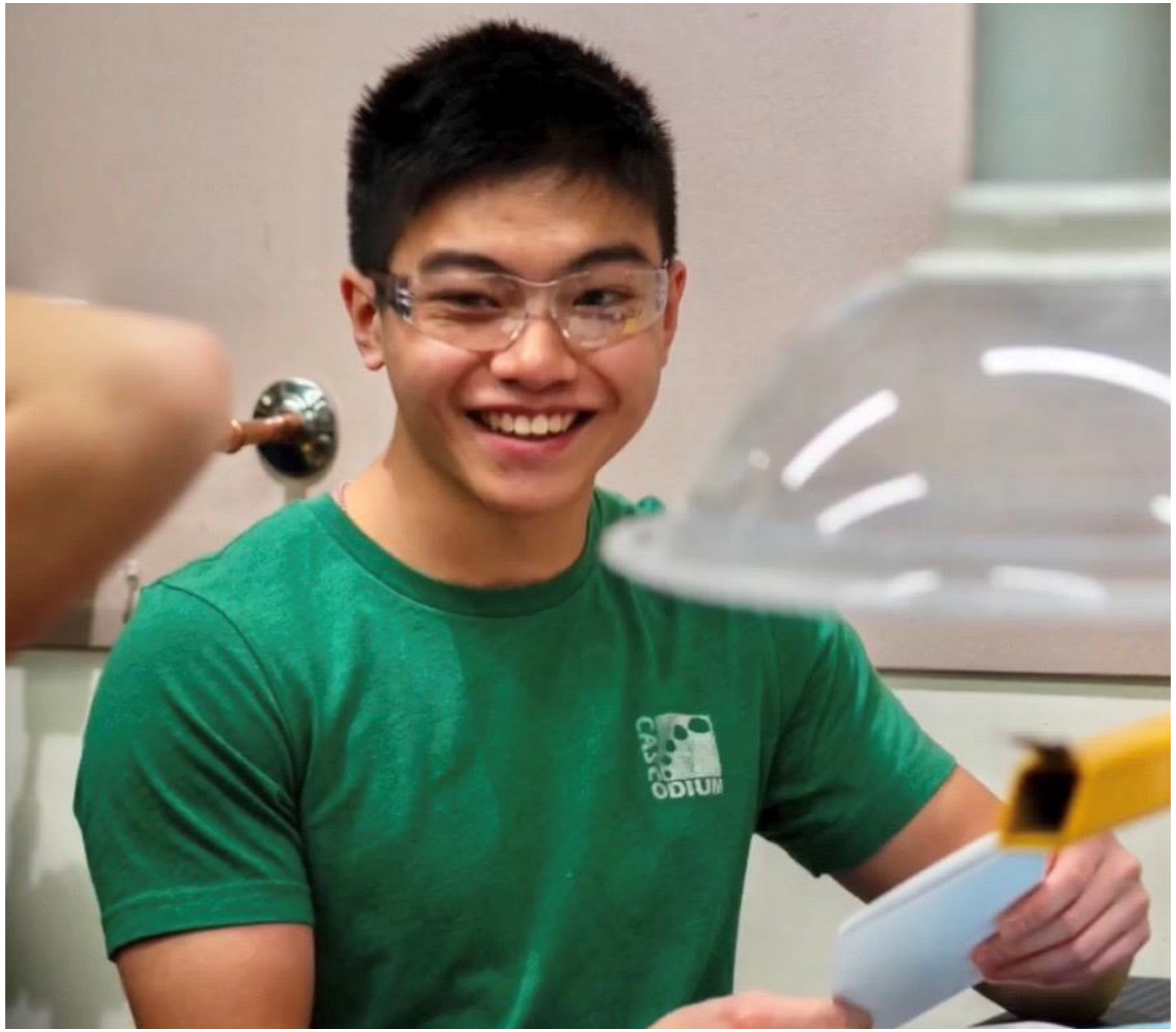
Michael Lu

EECS @ MIT ‘23

github.com/michaellu2019

mlu0708@mit.edu

linkedin.com/in/michael-lee-lu



Hi, I’m Michael!

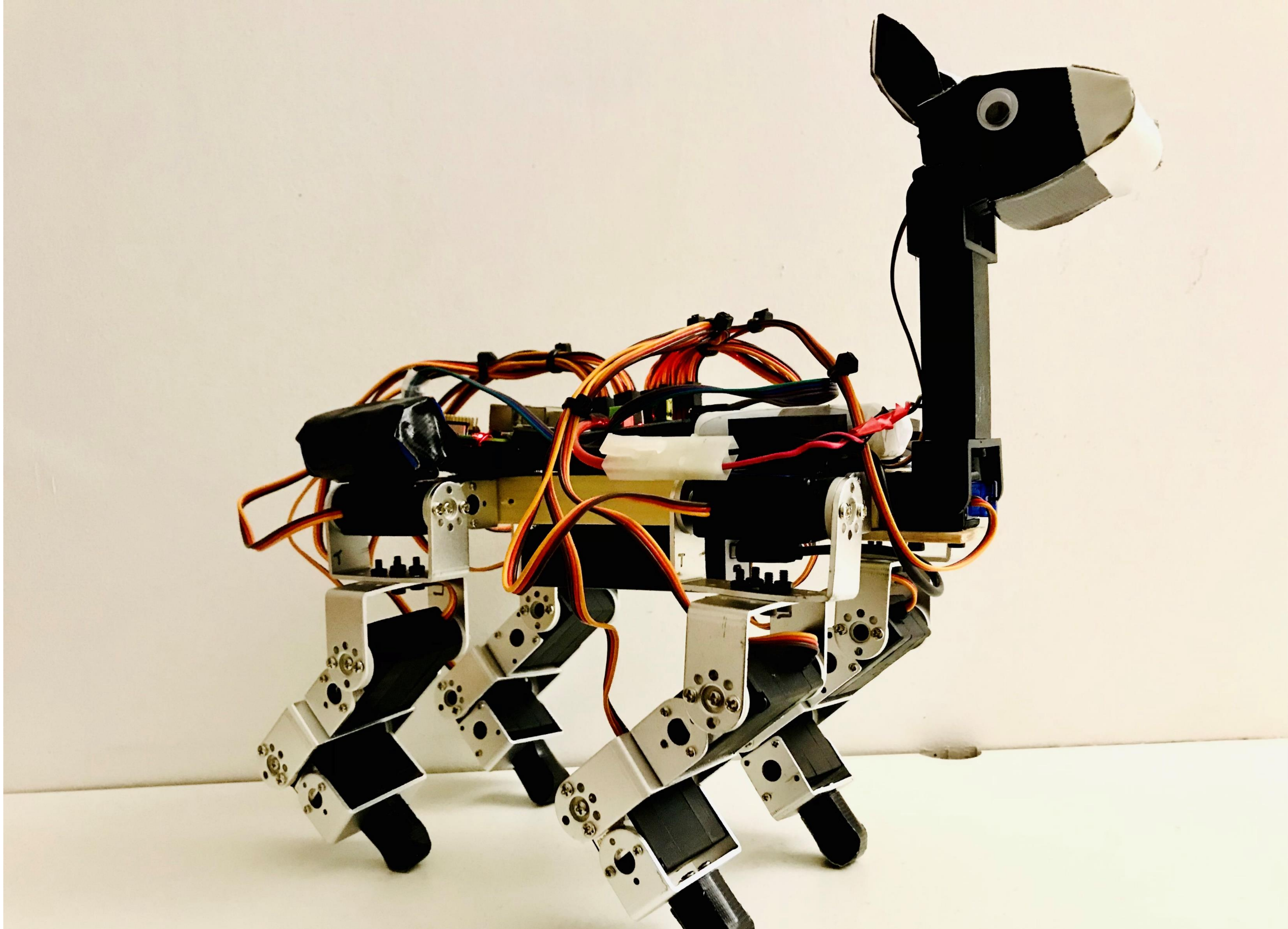
Just as painting is the creative outlet of an artist, engineering is my way to translate the imaginative ideas of my mind into the inventive work of my hand. This perspective is the foundation of my work as a junior at MIT studying Electrical Engineering and Computer Science.

Robot Donkey

Personal Project
January – June 2021

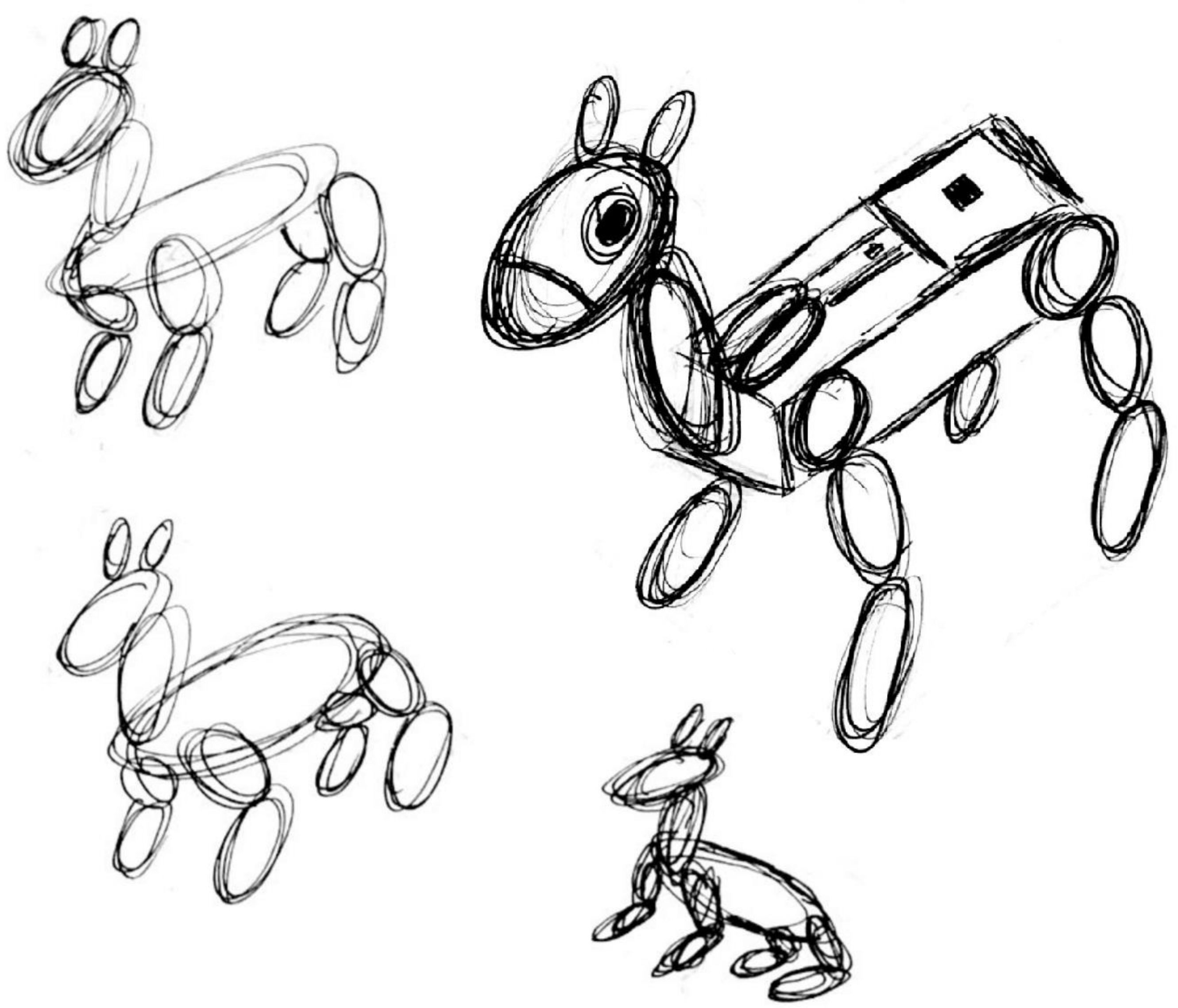
A WiFi-controlled robotic donkey that walks using inverse kinematics. The robot has four 3-degrees-of-freedom (DOF) legs, a 2-DOF neck/head, and two mini speakers for audio output.

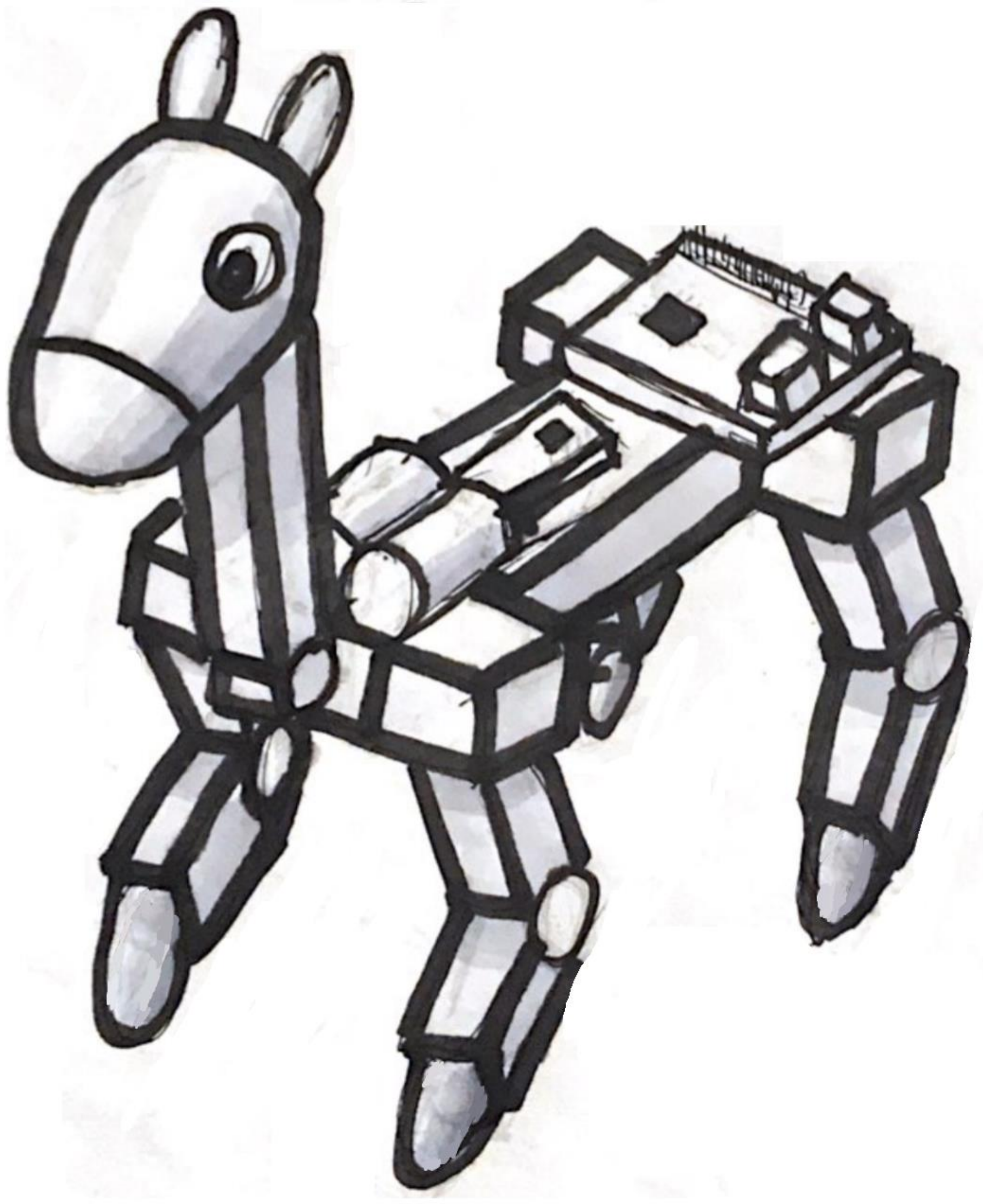
Demo:
https://youtu.be/aikBnfc7GT0
Code:
https://github.com/michaellu2019/dog

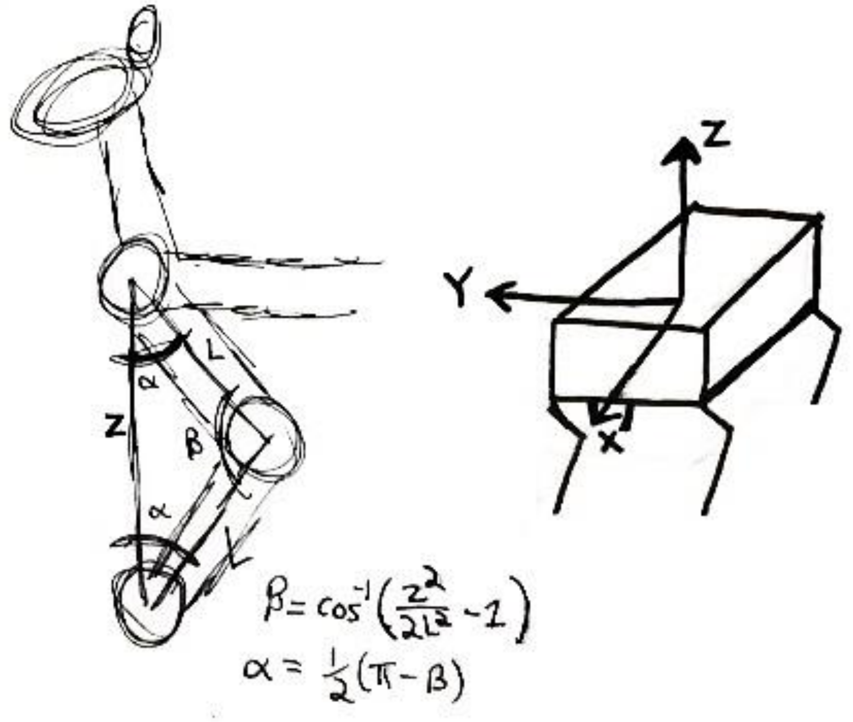


Ideation and Sketching

To build a 12-DOF quadruped and manipulate its orientation and position, I had to figure out how to pass in a 3D position and translate it into a set of angles for the servos to rotate to. Inverse kinematics would provide an easy way to calculate these values.

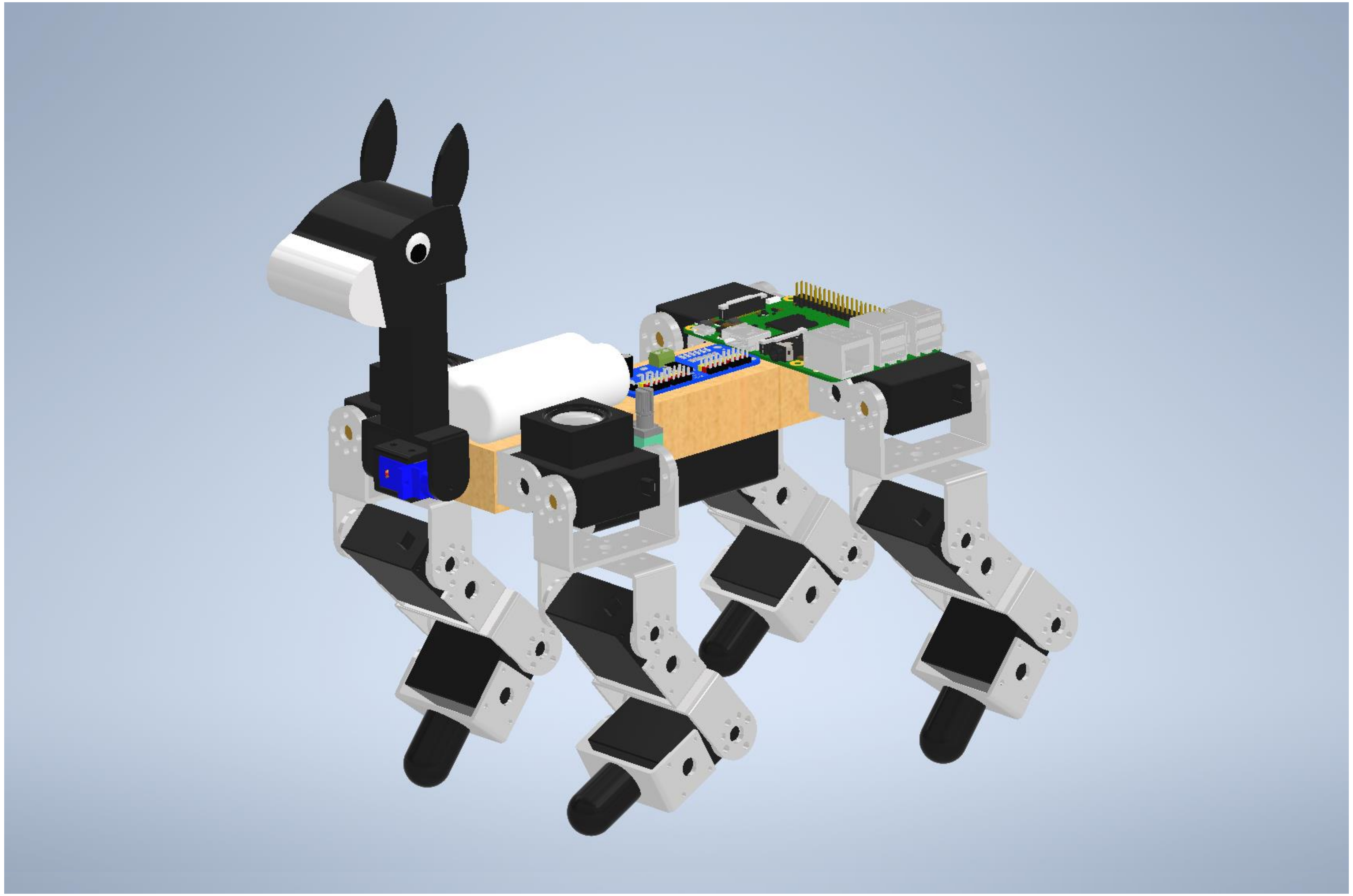


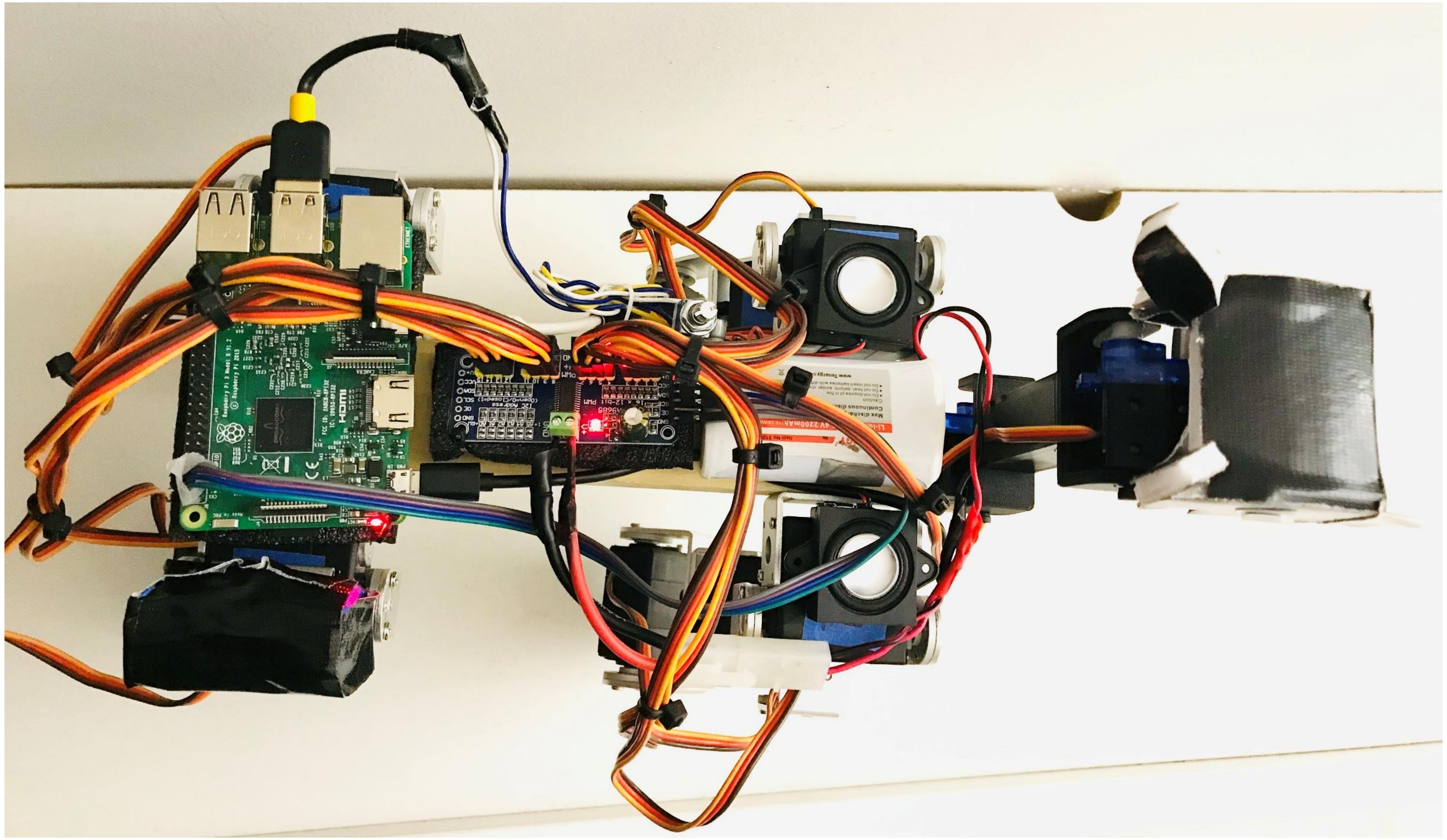




3D Modeling

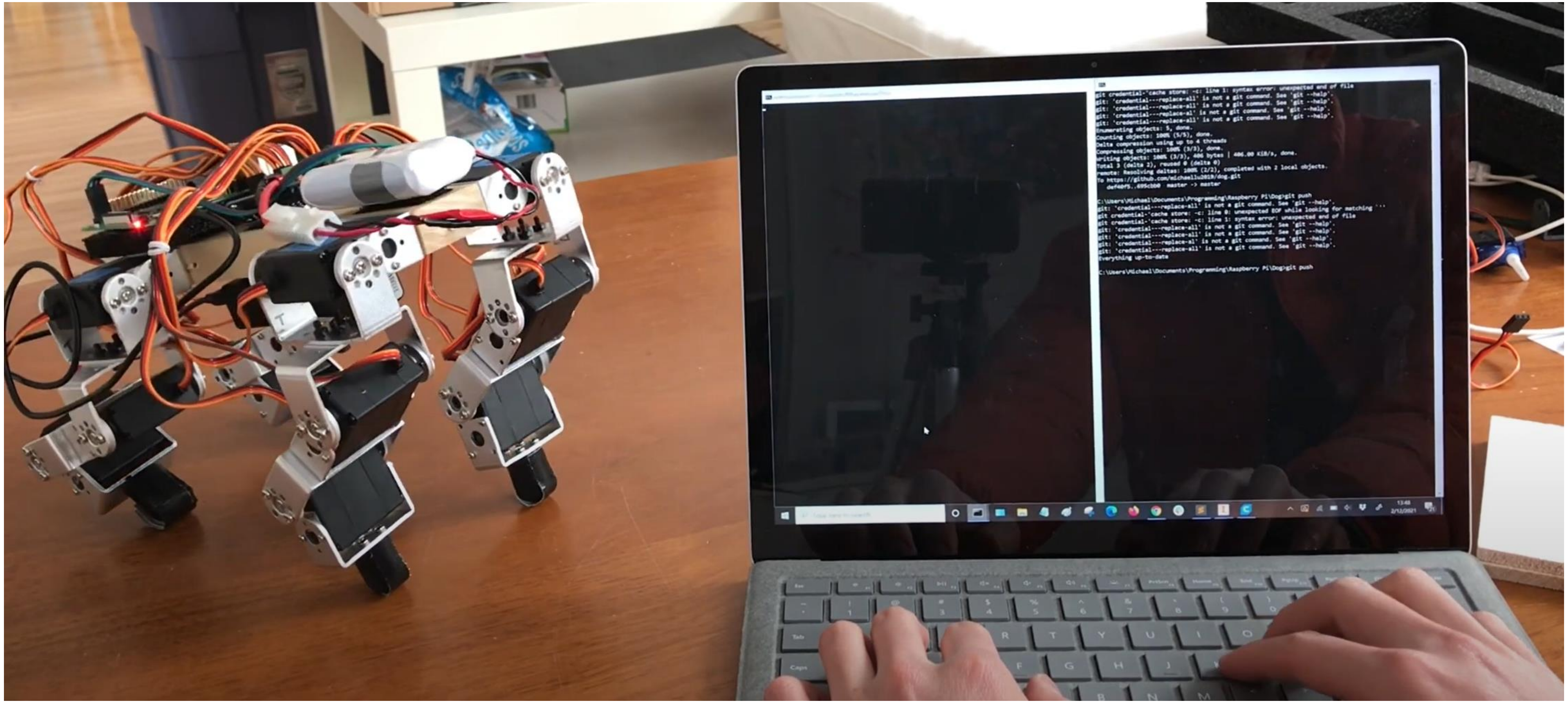
Robot Donkey's HobbyPark servos came with multiple aluminum mounting brackets, which made it easy to build four 3-DOF legs around a main wooden body. For the robot's feet, I used Autodesk Inventor to design four 3D-printed ball-point hooves that could be fastened to the servo brackets and were covered with rubber tips for more traction with the ground.





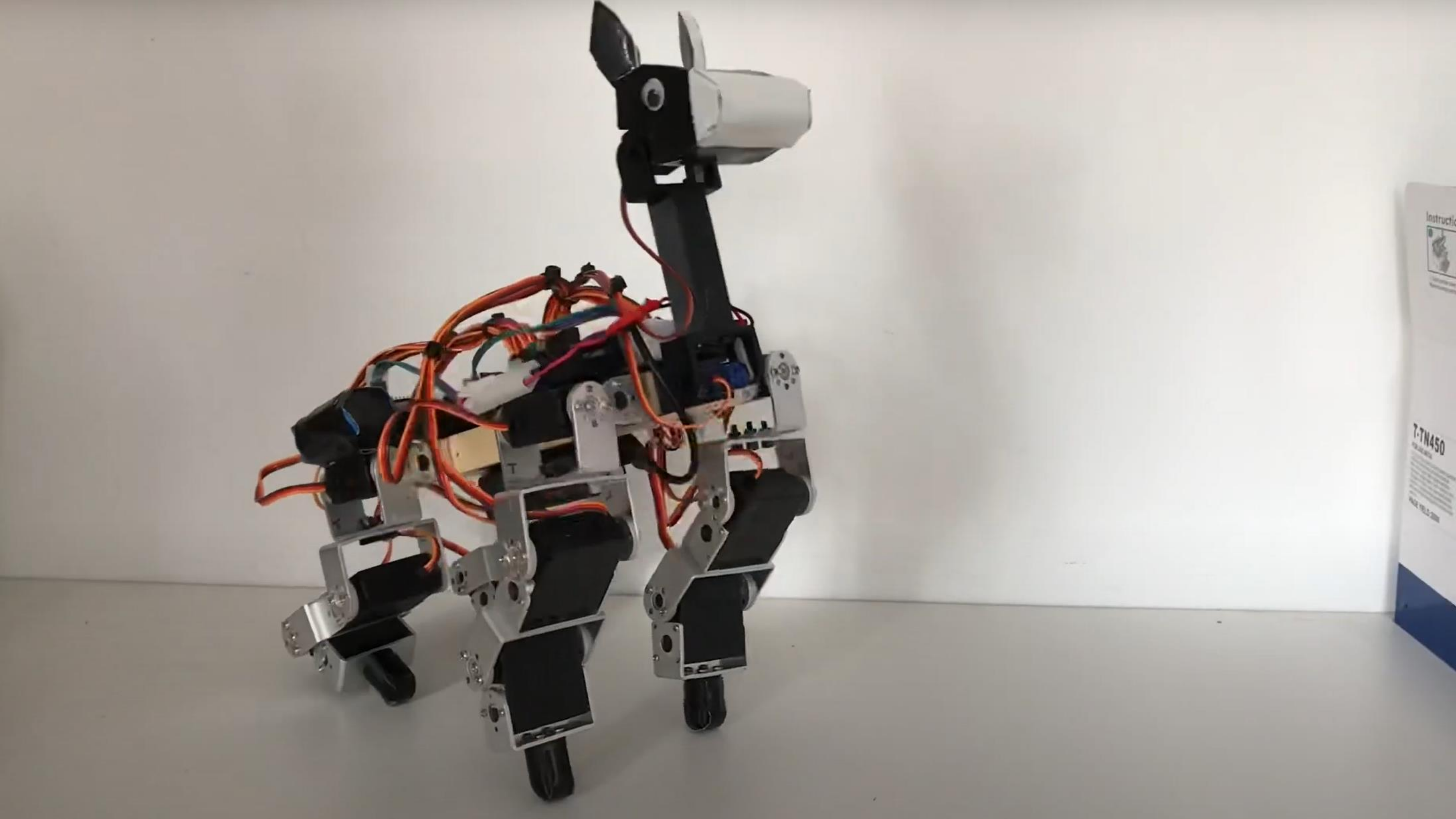
Prototyping

I built Robot Donkey with an Adafruit PCA9695 Servo Driver, 12 HobbyPark Servos for the robot legs, 2 SG90 Servos for the head and neck, a Super Mini PAM8403 Audio Amplifier and 2 MakerHawk Speakers for sound, a Raspberry Pi to control all these components, and a USB power bank and 7.4V LiPo battery for power.



Programming

Robot Donkey was controlled by a Raspberry Pi Python script that rotated the leg joints based on inverse kinematics to move the robot to precise positions and orientations. Once the Raspberry Pi connected to the Internet after powering up, I could SSH into the board and control the robot from the command line on my laptop over WiFi.

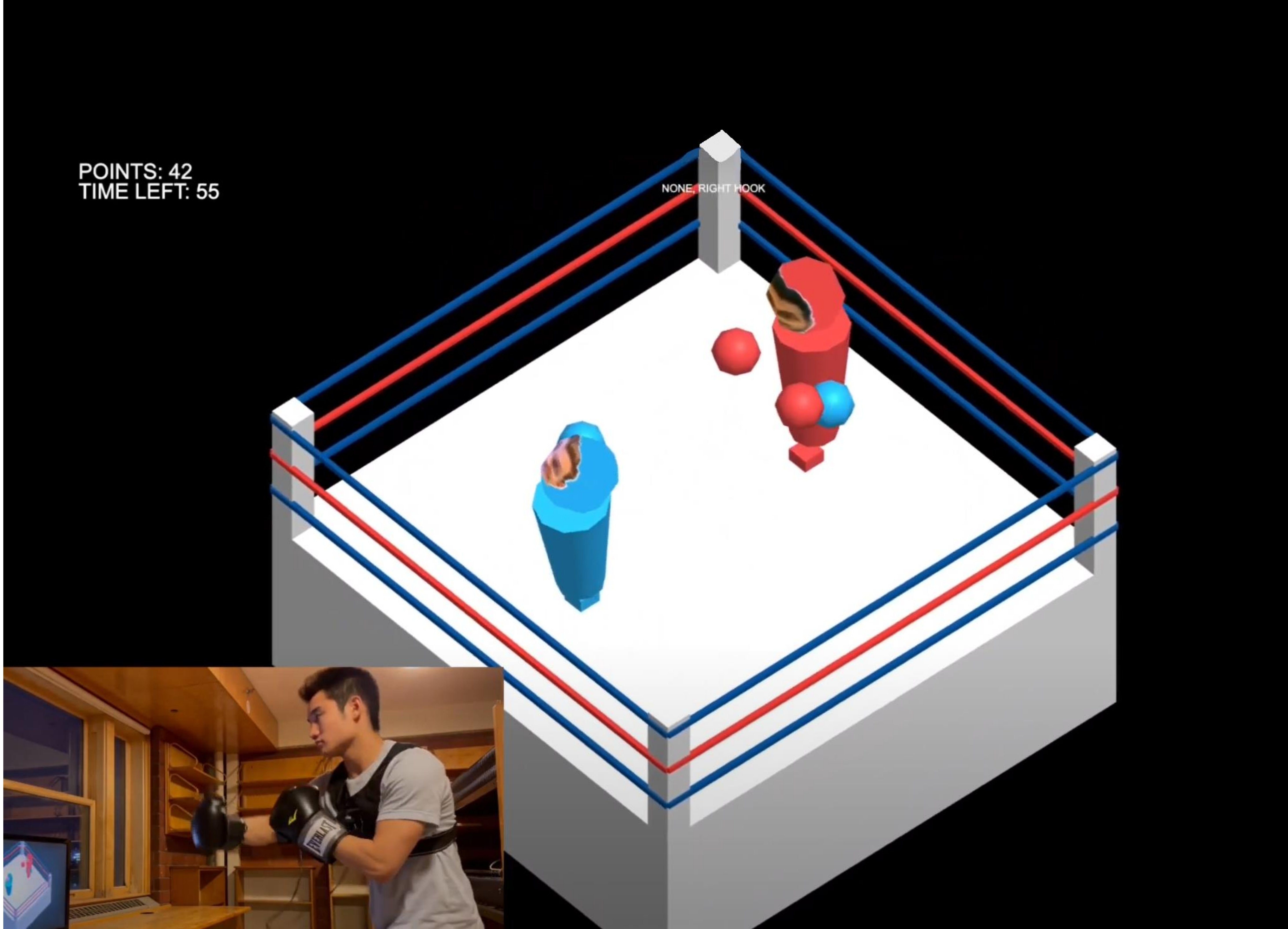

T-TN450

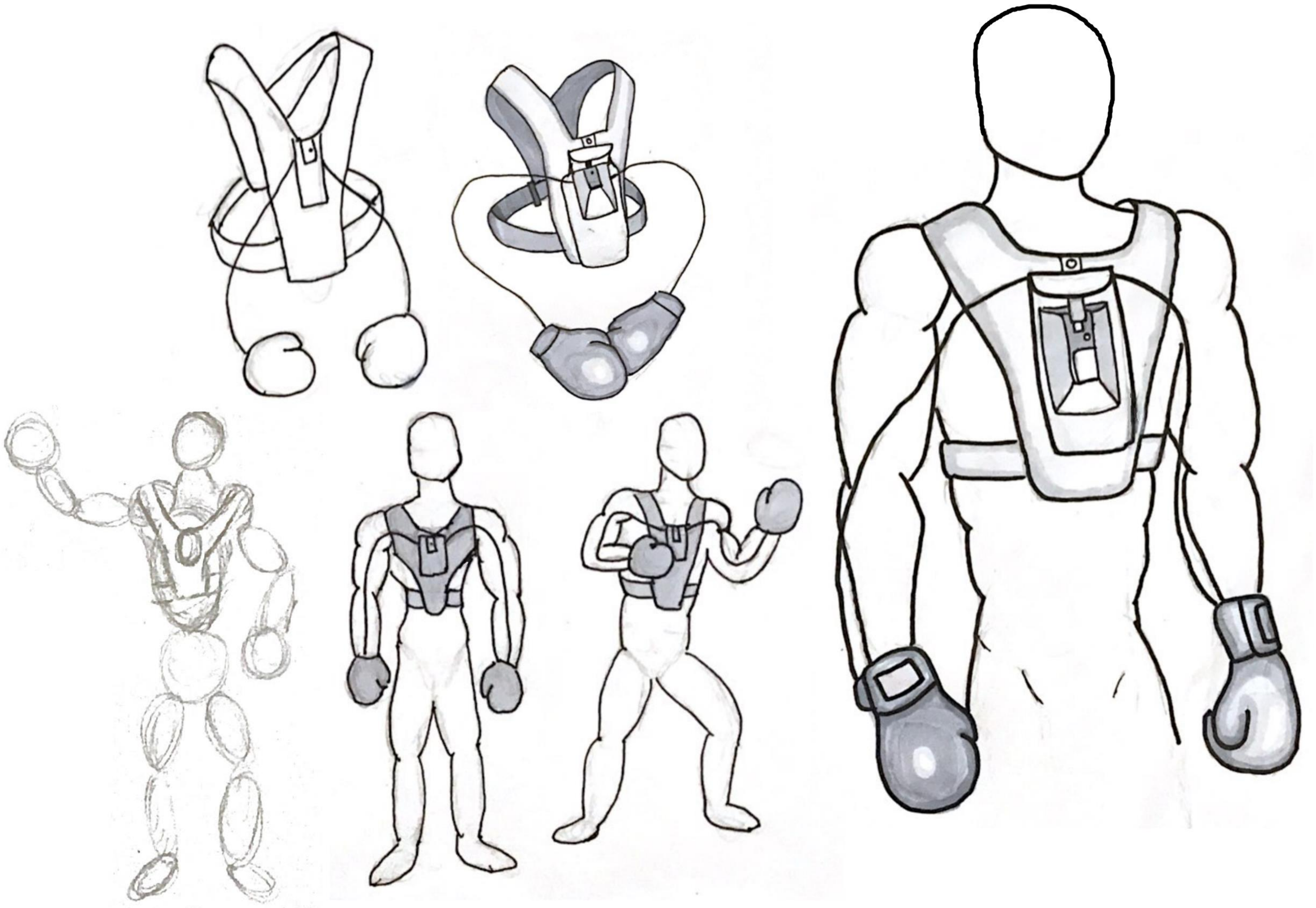
IoT Boxing

6.08 – Embedded Systems
April – May 2021

An IoT version of the iconic Wii Sports Boxing game that uses an ESP32 WiFi MCU and accelerometer to send real time punch motion data over WebSockets to an online 3D boxing game.

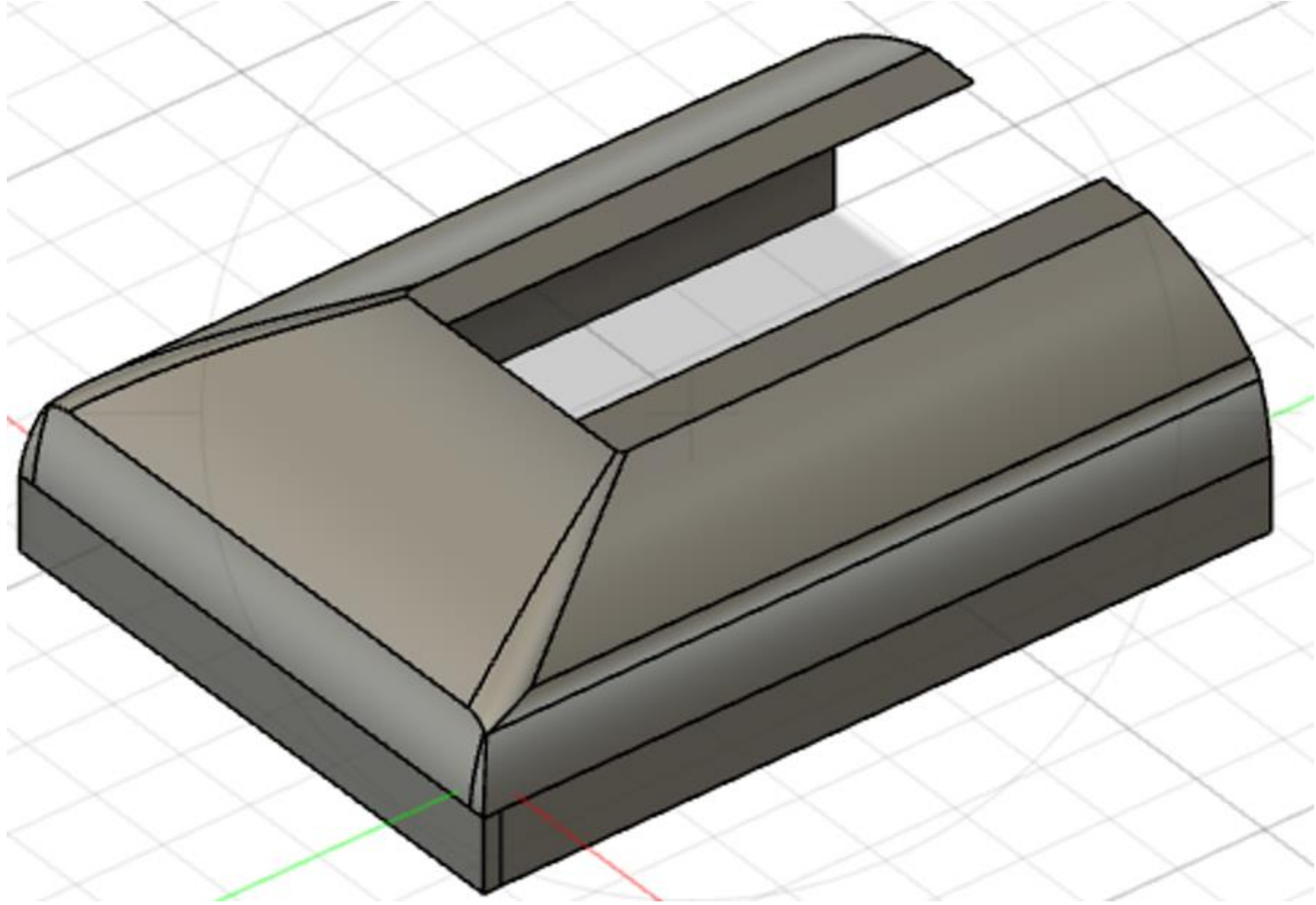
Demo:
https://youtu.be/XnNphWLp-24
Code:
https://github.com/michaellu2019/ragdoll-boxing

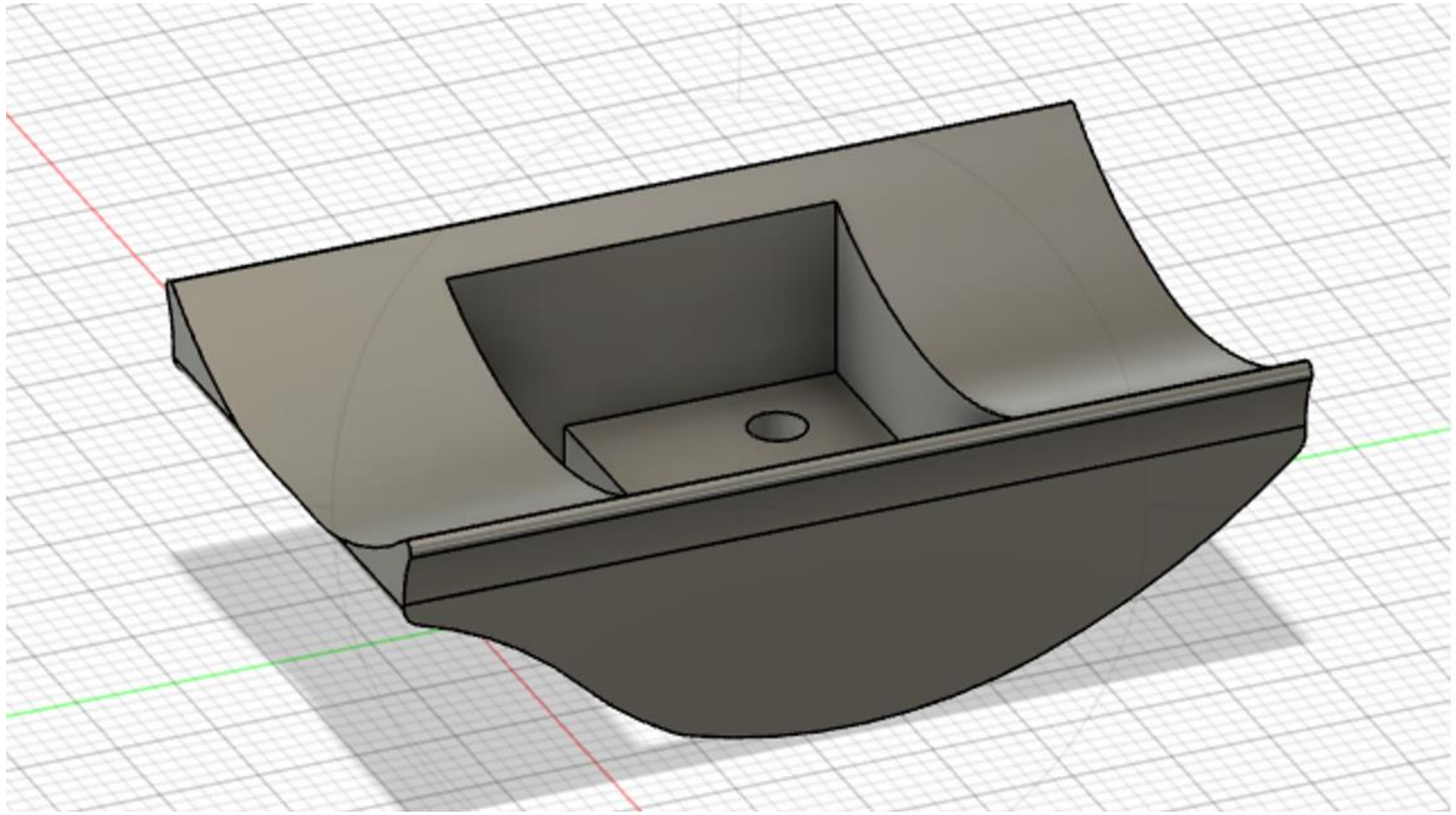




Ideation and Sketching

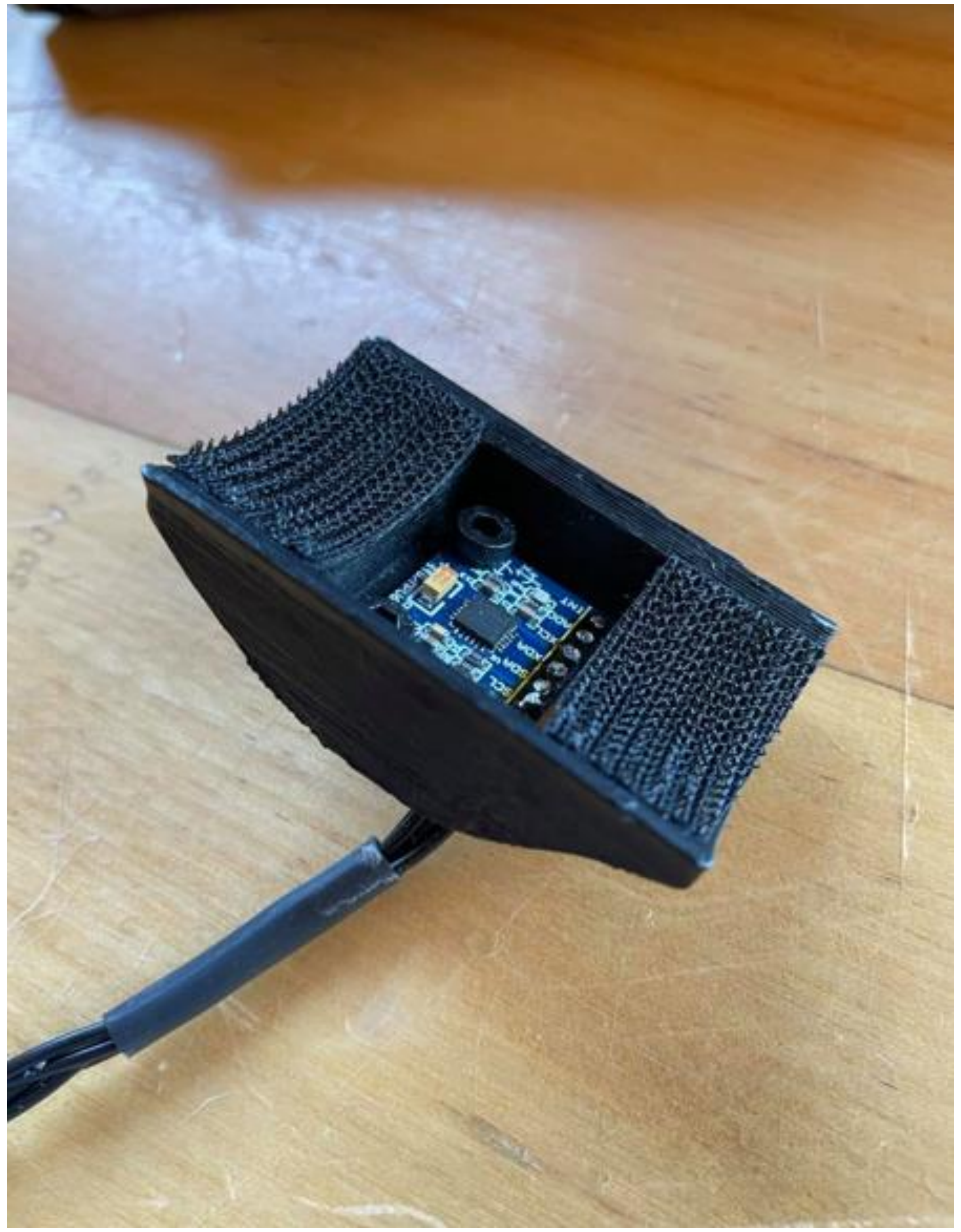
To give the player the most immersive boxing experience, I decided to embed the electronics inside a pair of boxing gloves and a sports vest. The boxing gloves would hold the MPU6050 accelerometers to detect punch movements in real time, and the main electronics, such as the ESP32 and TNTOR Ultra Thin Power Bank, would be mounted inside the sports vest.





3D Modeling

In order to give the wearable game accessories a polished look, the electronics had to blend in with the boxing gloves and sports vest, so custom mounts for the MPU6050 and ESP32 were designed in Autodesk Fusion360 to be 3D printed.



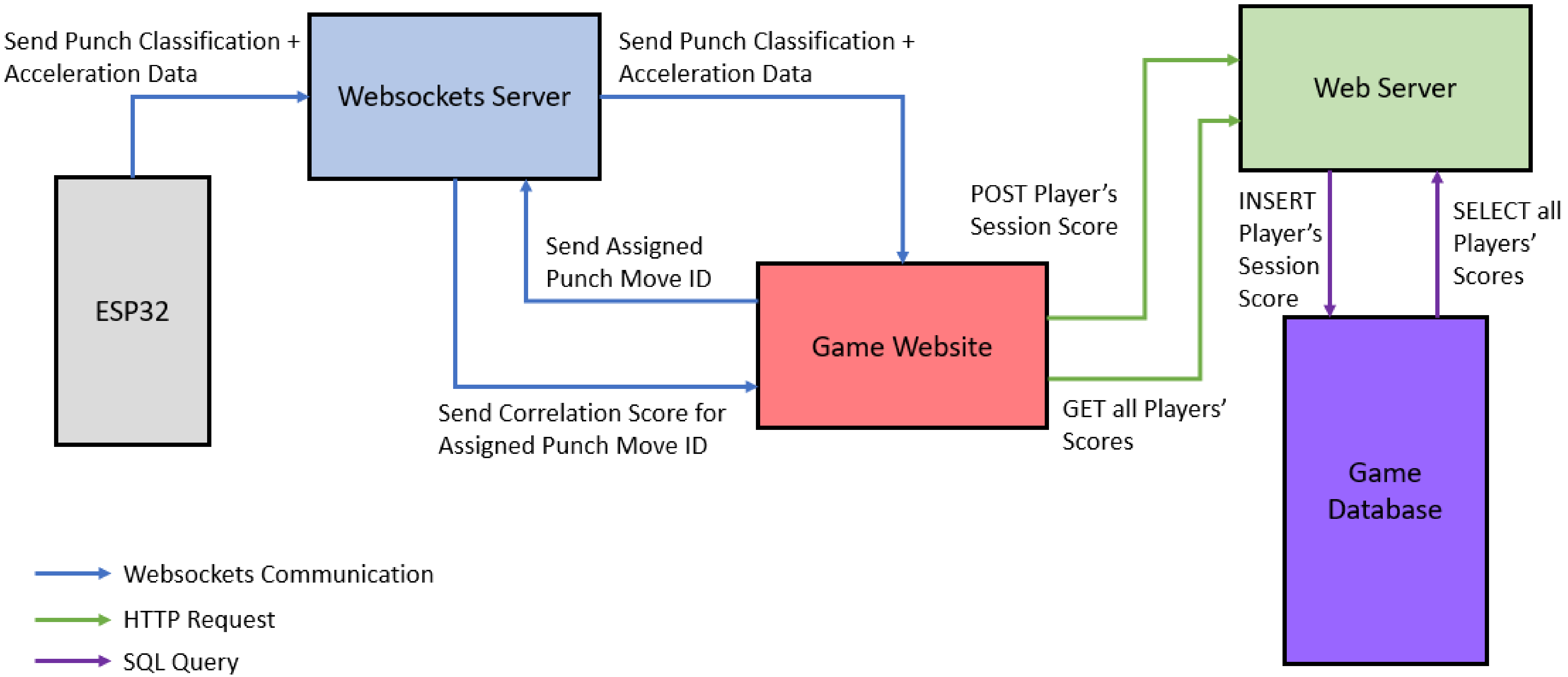
Prototyping

I mounted the 3D-printed electronics housings to the wearable game accessories using Velcro, which would be relatively inconspicuous. In addition, using Velcro would allow the electronics to be easily removed in case they needed to be replaced or modified.

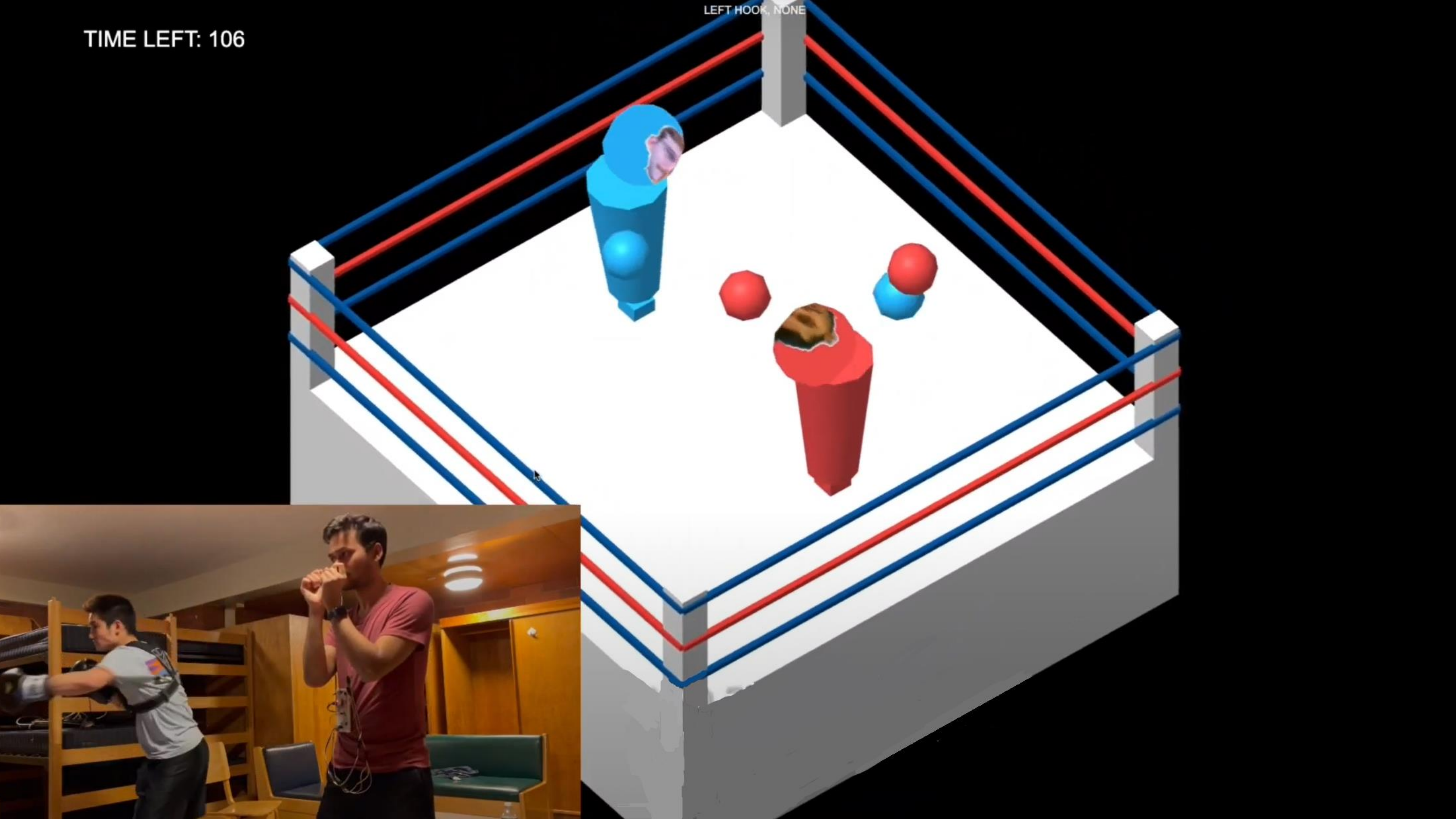
Prototyping

To minimize the electronics and weight in the boxing gloves, the MPU6050s in the boxing gloves were wired to the heavier components—the ESP32 and TNTOR Ultra Thin Power Bank—in the sports vest. This would allow the player to throw punches without feeling the added weight of the electronics.

Programming



The ESP32 ran C code to process the acceleration data from the player's punches and sent them to a Python WebSockets server, which forwarded it to the game website. Written in HTML, CSS, and JavaScript (using Three.js for graphics and Cannon.js for physics), the game website would control a 3D model of a boxer based on the received movements and would also save player scores in a backend database.

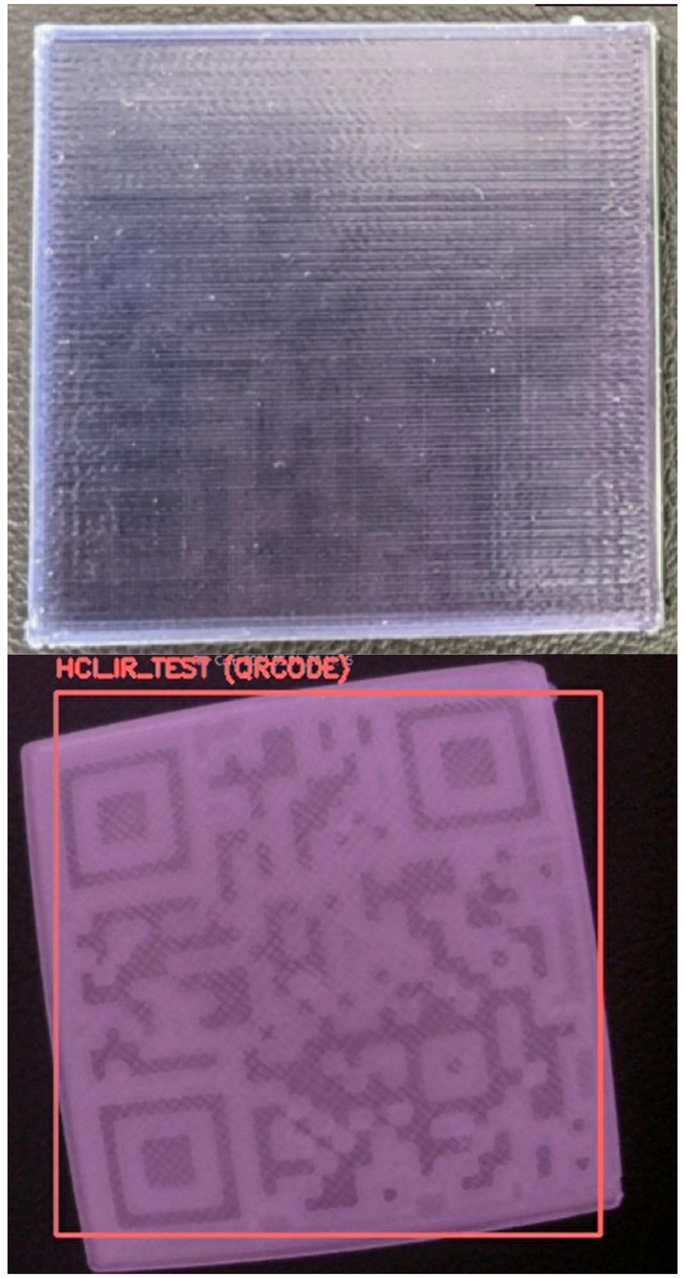

TIME LEFT: 106
LEFT HOOK, NONE

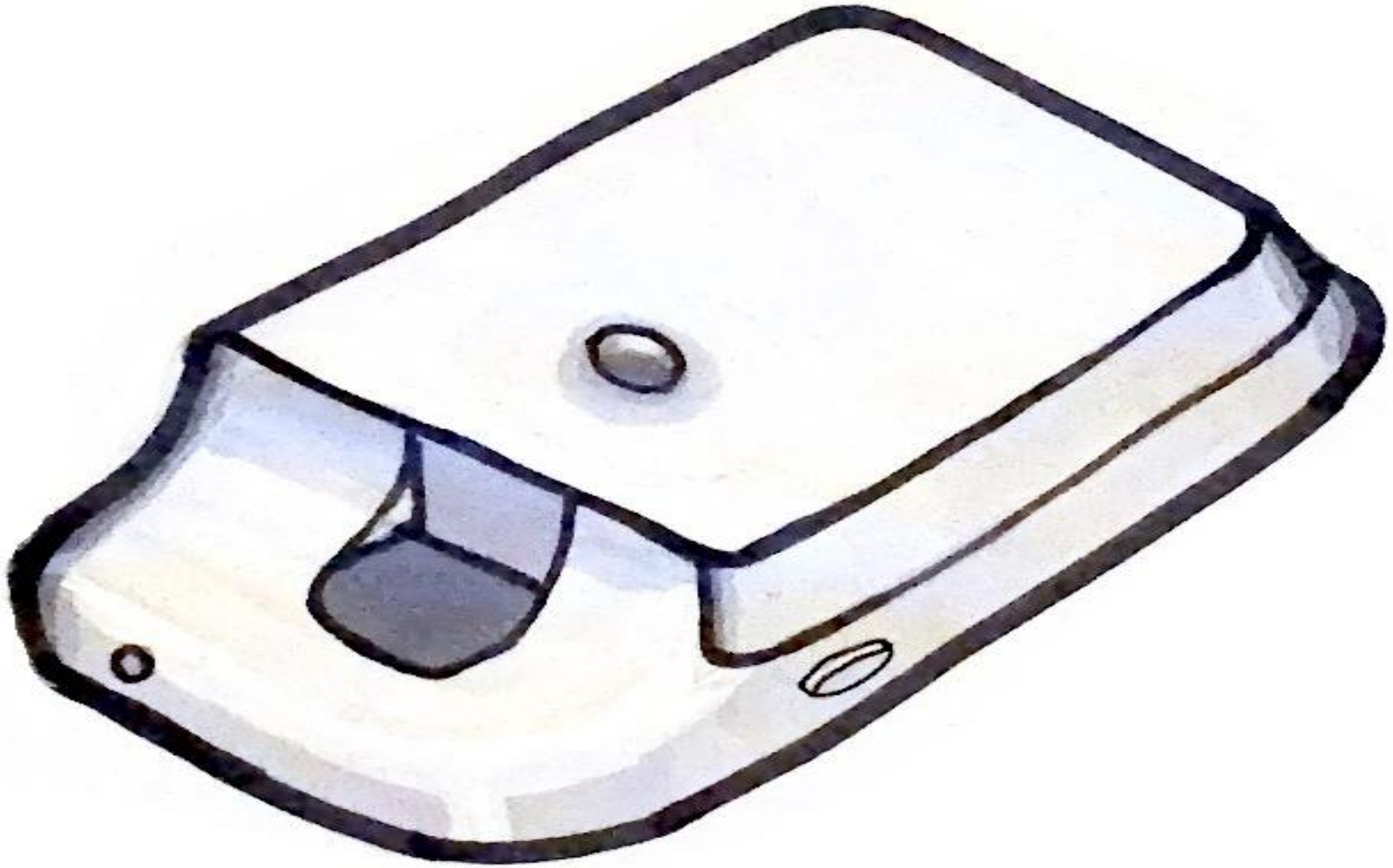
IR Camera Module

MIT CSAIL HCIE Research Project

February – May 2021

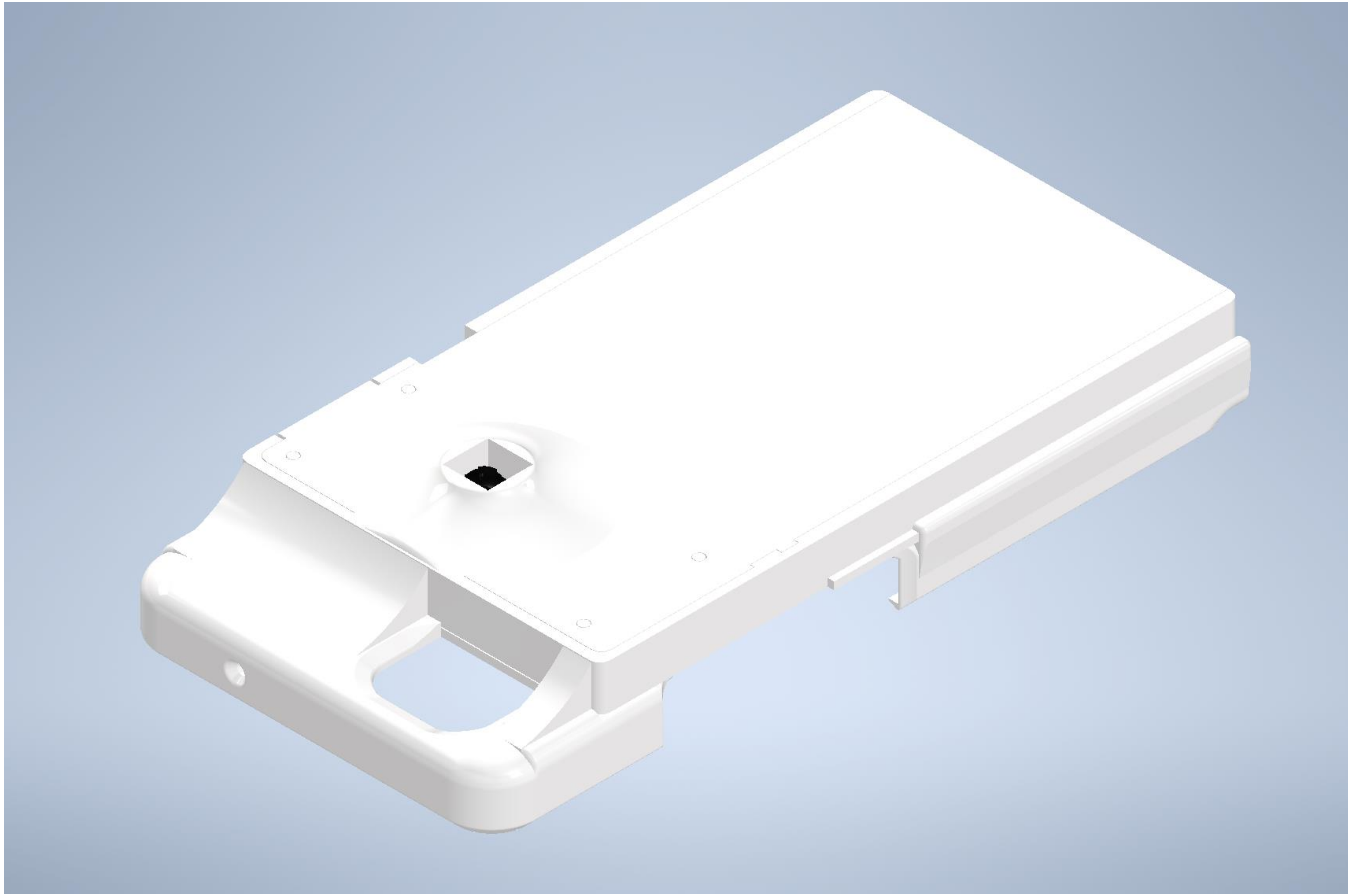
A 3D printed phone case with a Raspberry Pi Zero W infrared camera module that can read QR codes hidden behind infrared-transparent plastic.





Ideation and Sketching

With the ability to use infrared-transparent filament to 3D print hidden QR codes that could only be decoded with an IR camera, I wanted to design a lightweight, thin, removable phone case module that would house an IR camera that could read these hidden QR codes and stream them to my phone.

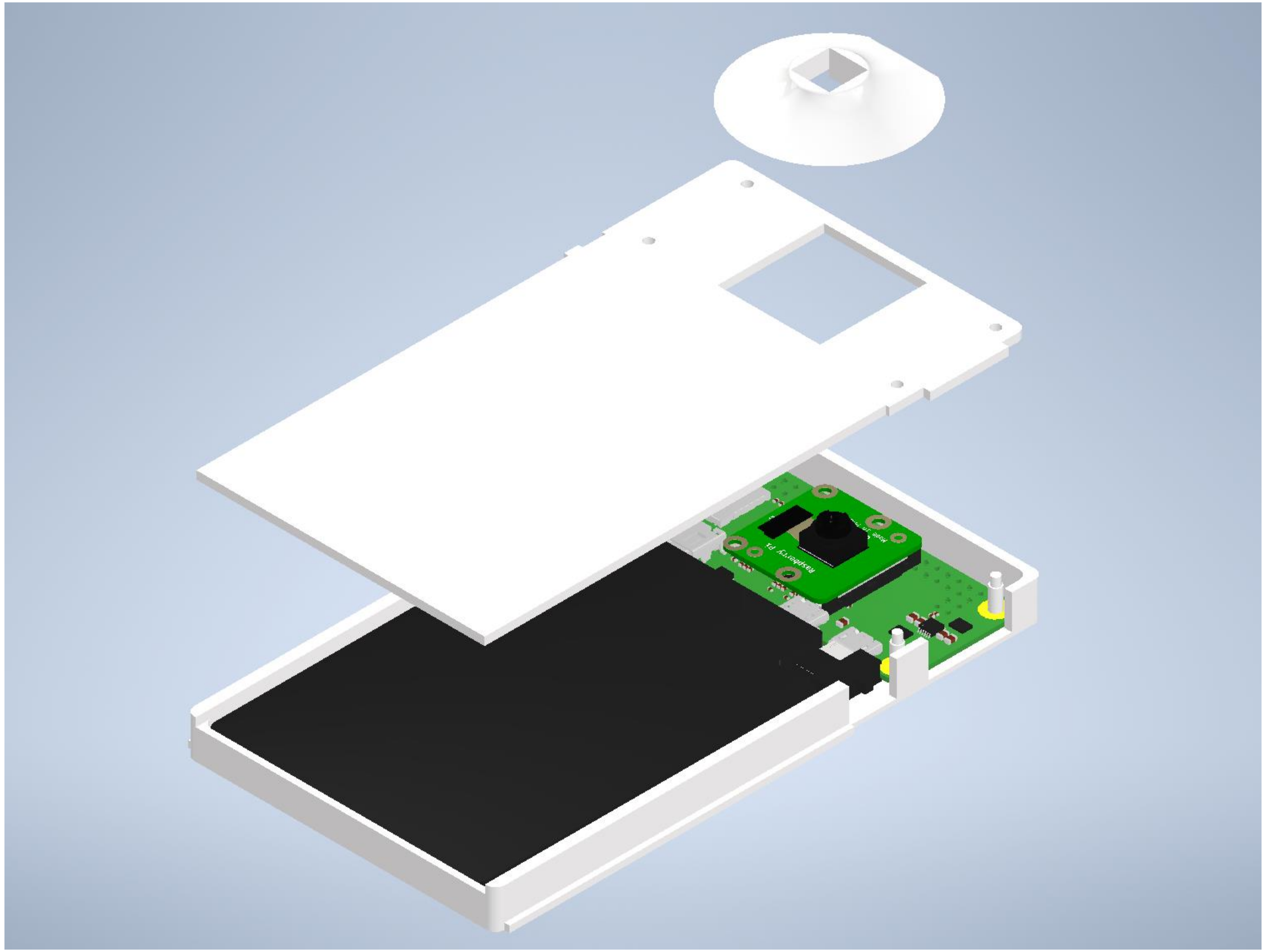


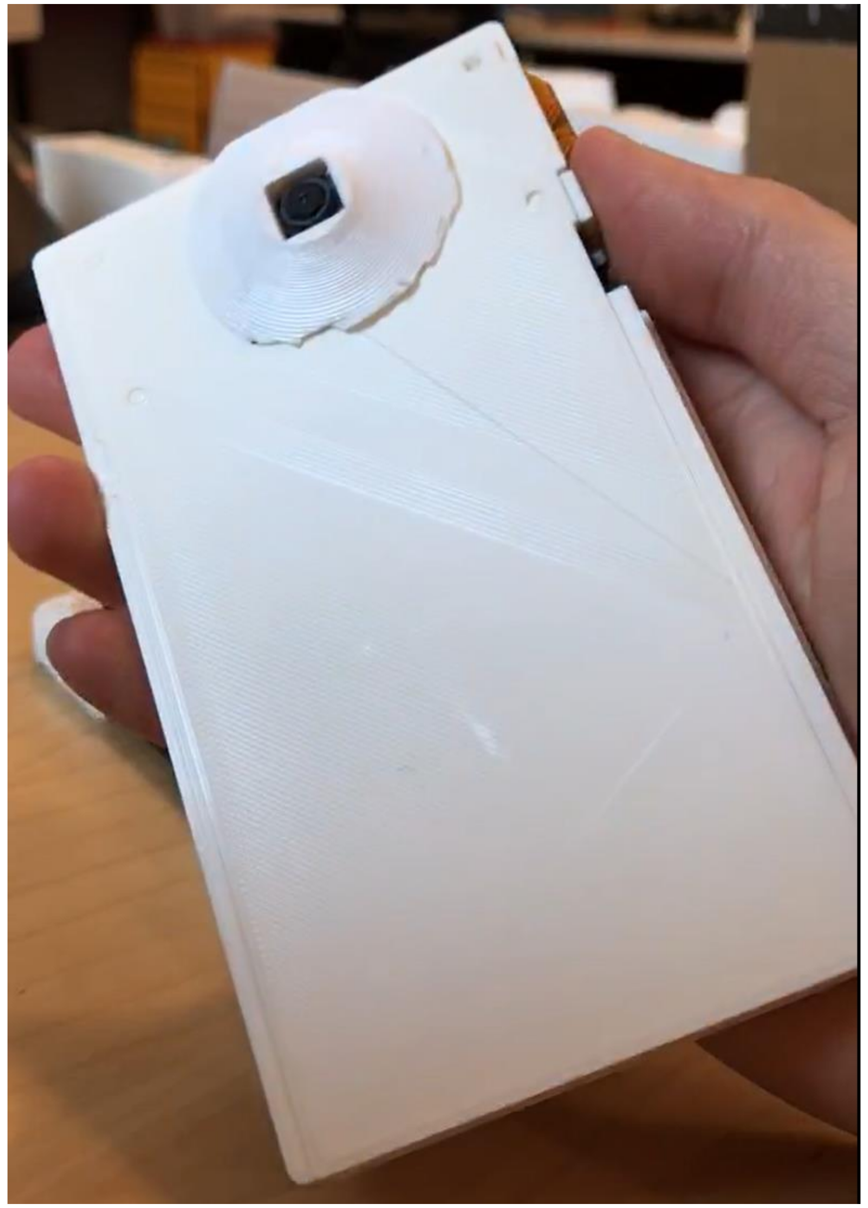
3D Modeling

The IR camera housing would be a removeable module that could slide into small grooves on a custom 3D-printed phone case. To maintain the slim form factor of a smartphone phone, I designed the IR camera module housing to be just a few millimeters thick.

3D Modeling

To keep the IR camera module as small and simple as possible, the main electronic components—Raspberry Pi Zero W, NoIR V2 Camera, and TNTOR Ultra Thin Power Bank—were fit tightly together in a case with connecting pegs to secure the module cover, removing the need for any fasteners.



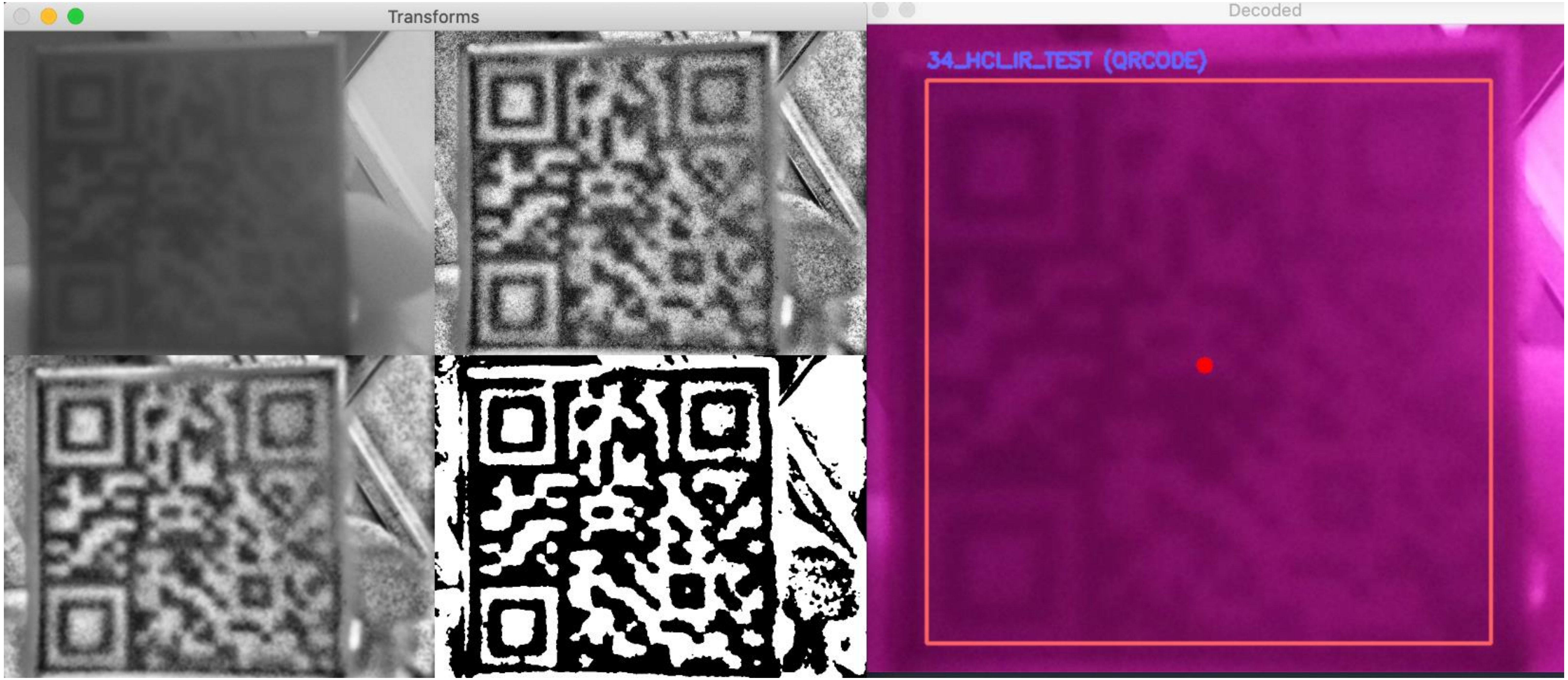


Prototyping

The IR camera module was printed with PLA to give it sufficient rigidity. To achieve a tight fit inside the module, I replaced the TNTOR Ultra Thin Power Bank charging cable with a compact Micro USB and switch assembly. The NoIR camera would fit into the square hole of the camera cap above the Raspberry Pi Zero W.

Prototyping

The IR camera module was able to easily slide into the custom phone case, which was 3D printed with TPU so that it was flexible enough to fit around the phone and camera module.



Programming

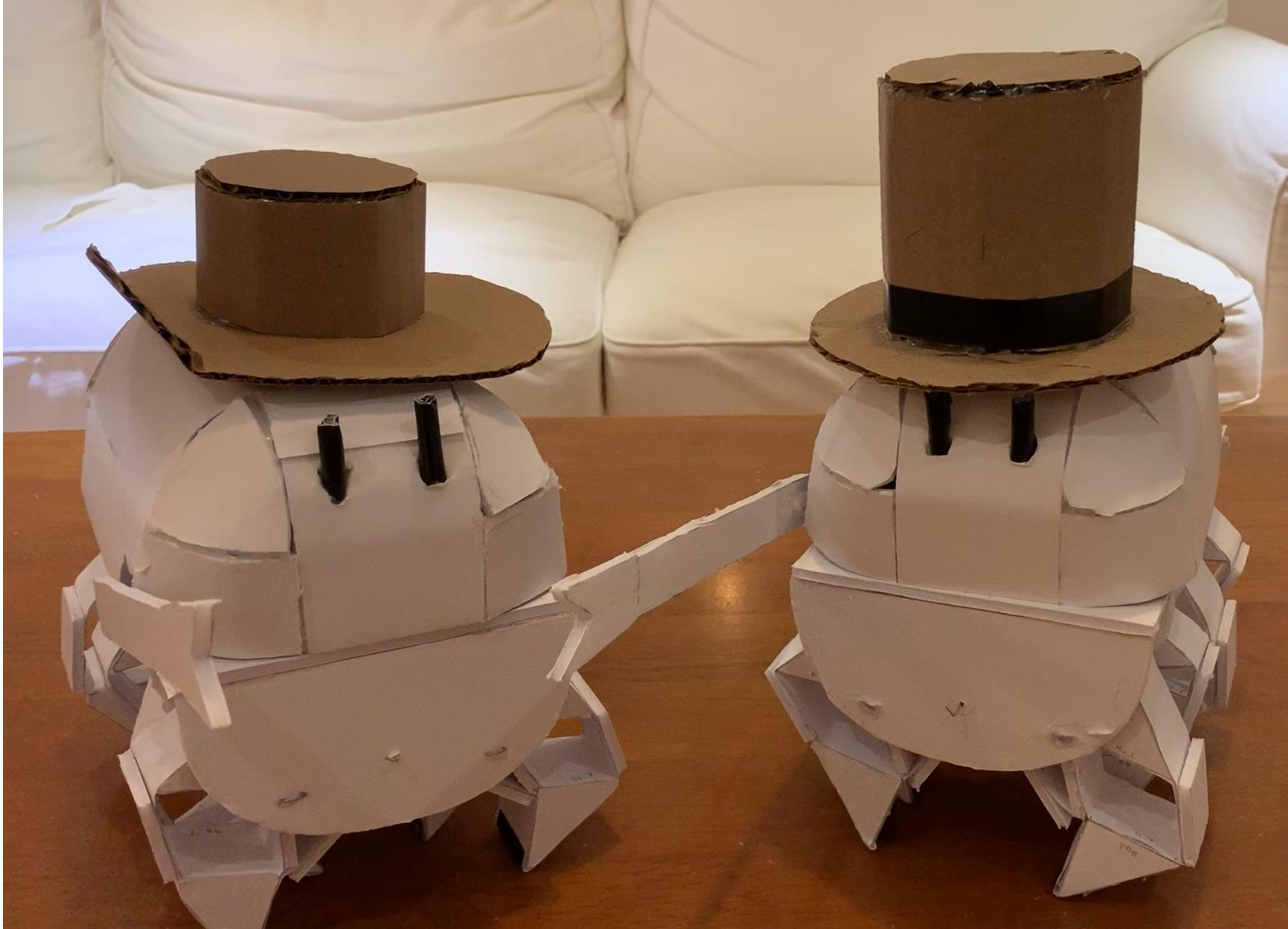
An OpenCV image processing script ran on the Raspbery Pi Zero W, which would take the IR camera output and apply several image transform techniques, including grayscale, Gaussian blur, and contrast limited adaptive histogram equalization (CLAHE), so that it was easier to read the QR code. The results would be streamed in real time to a smartphone app built with the cross-platform Python framework Kivy.

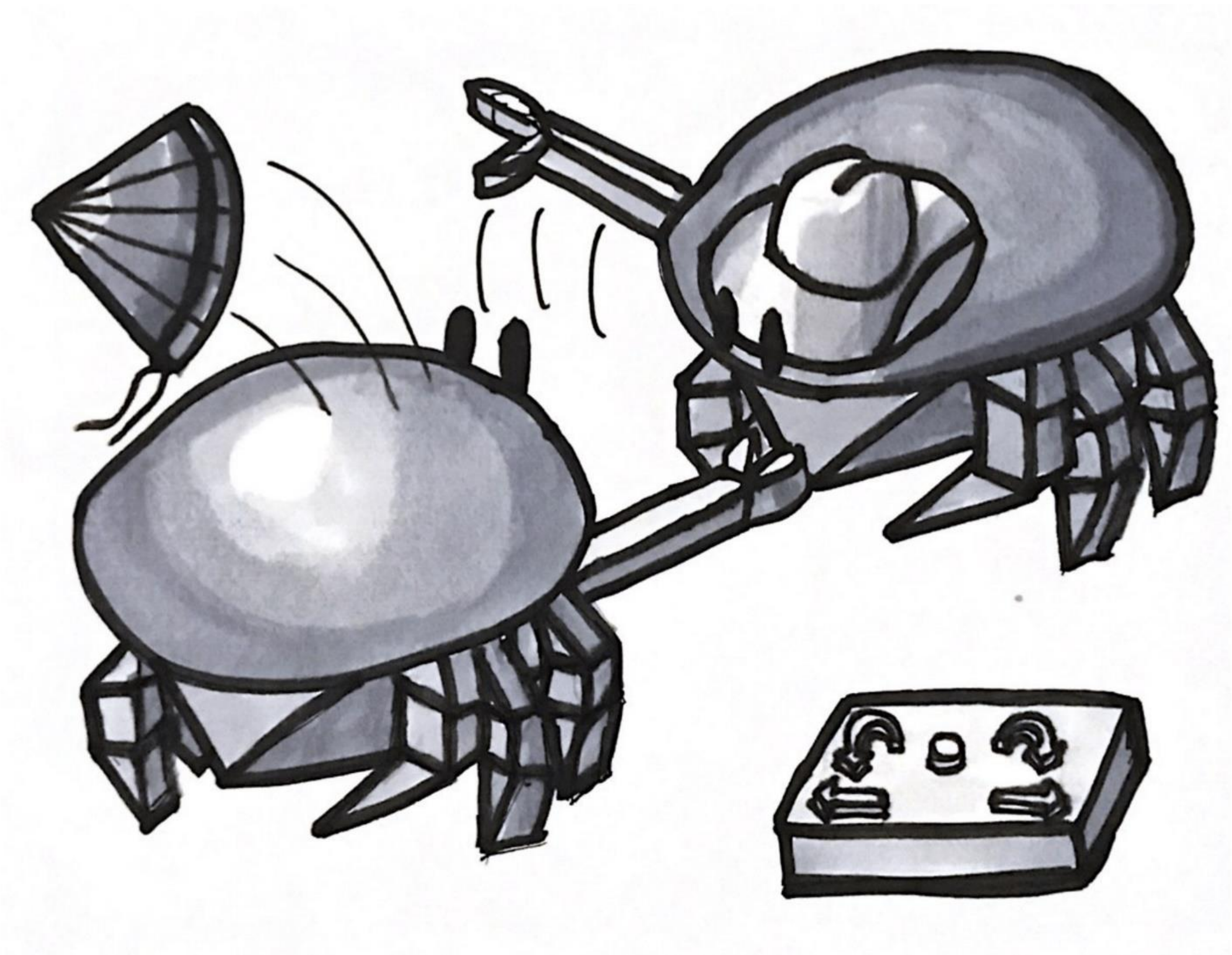
Kombat Krabs

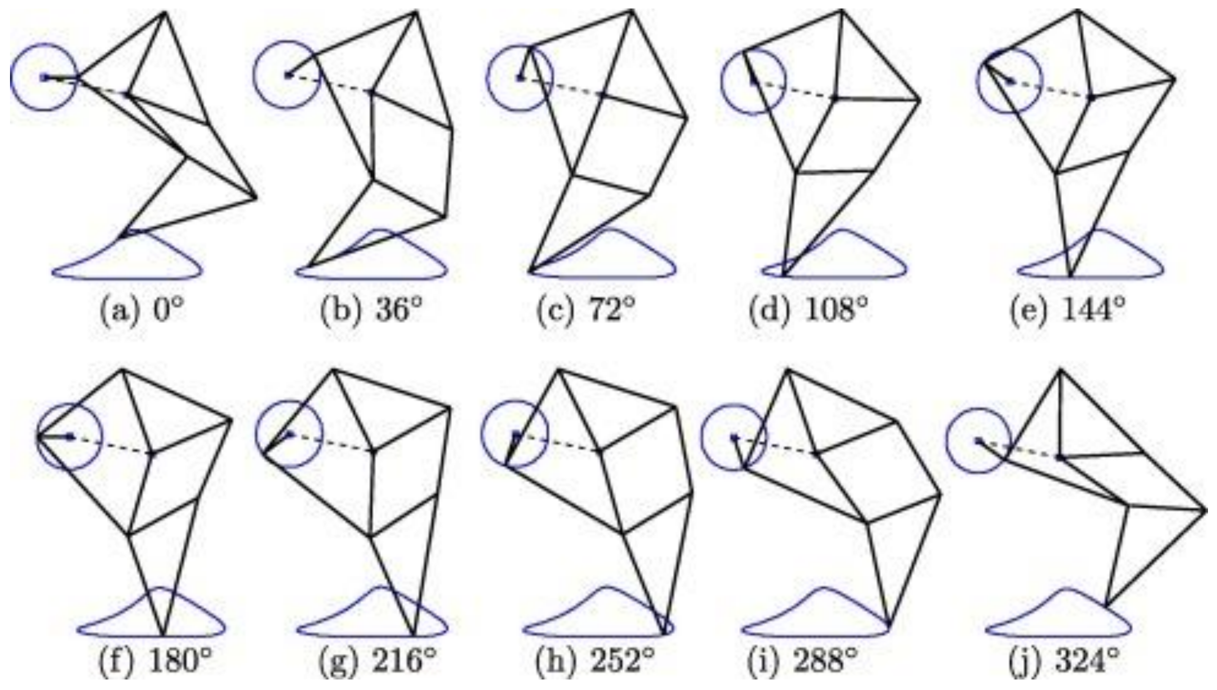
2.00B – Toy Product Design
April – May 2020

Remote-controlled toy robot crabs that walk using the famous leg linkage of kinetic sculptor Theo Jansen. These were built primarily out of foam core, paper, and cardboard.

Demo:
https://youtu.be/xg4couBU9fM
Code:
github.com/michaellu2019/kombat-krabs
CAD:
a360.co/3ePCxNx

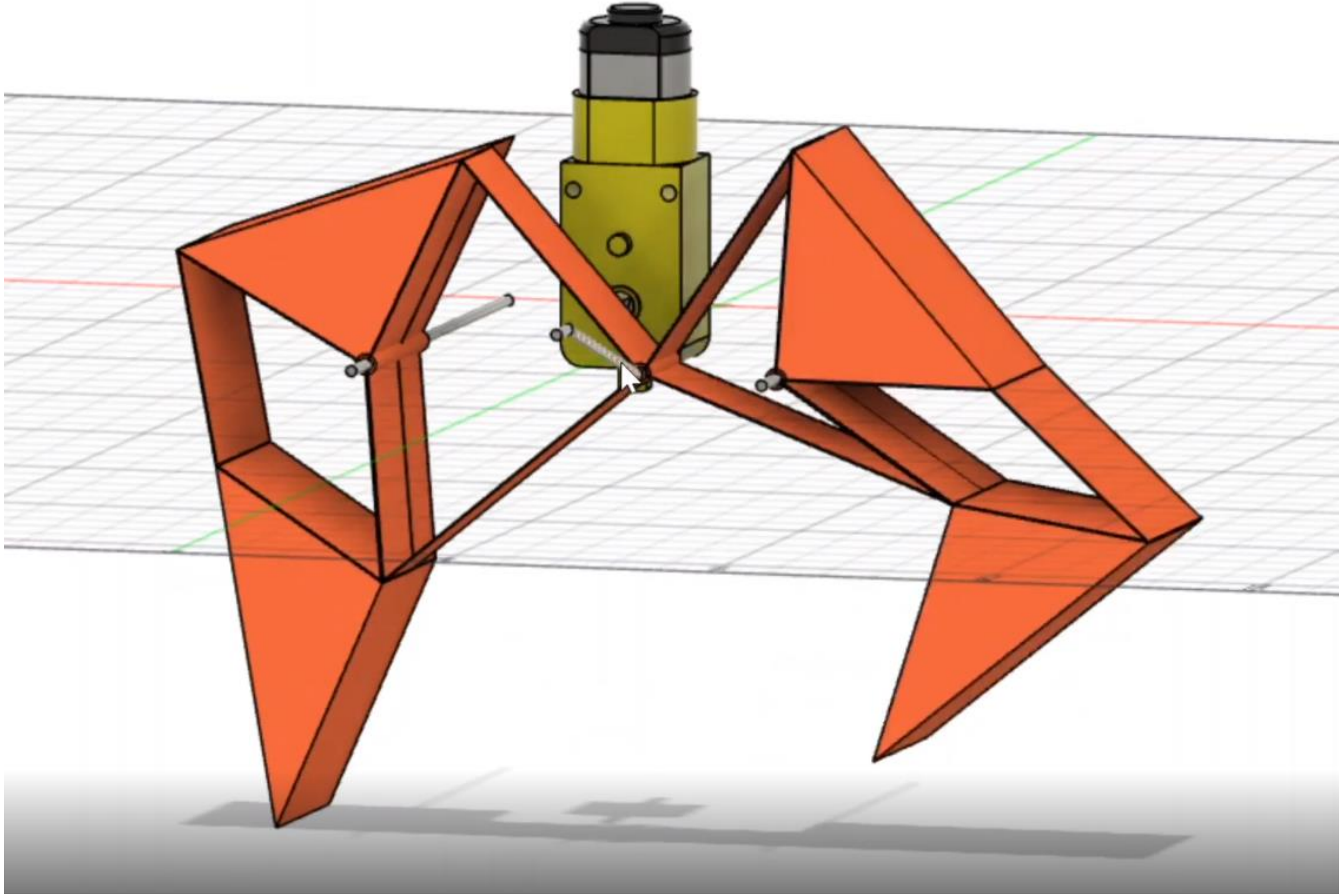






Ideation and Sketching

The biggest challenge of Kombat Krabs was designing a mechanism for the crabs to walk on eight legs. I decided to use Jansen’s Linkage, a clever walking mechanism composed of several connected segments that would trace out a smooth walking curve when driven by a circular rotational shaft.

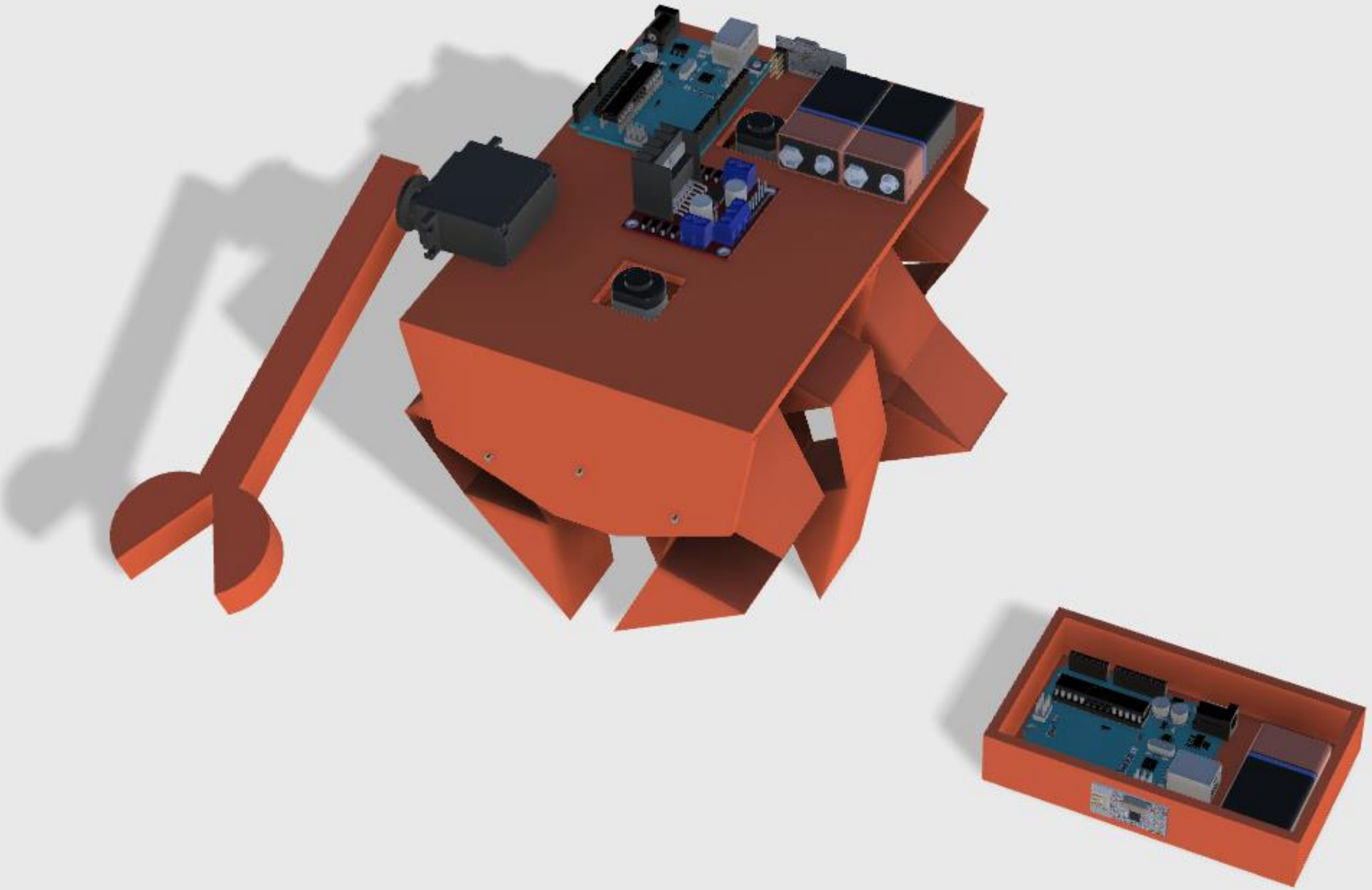


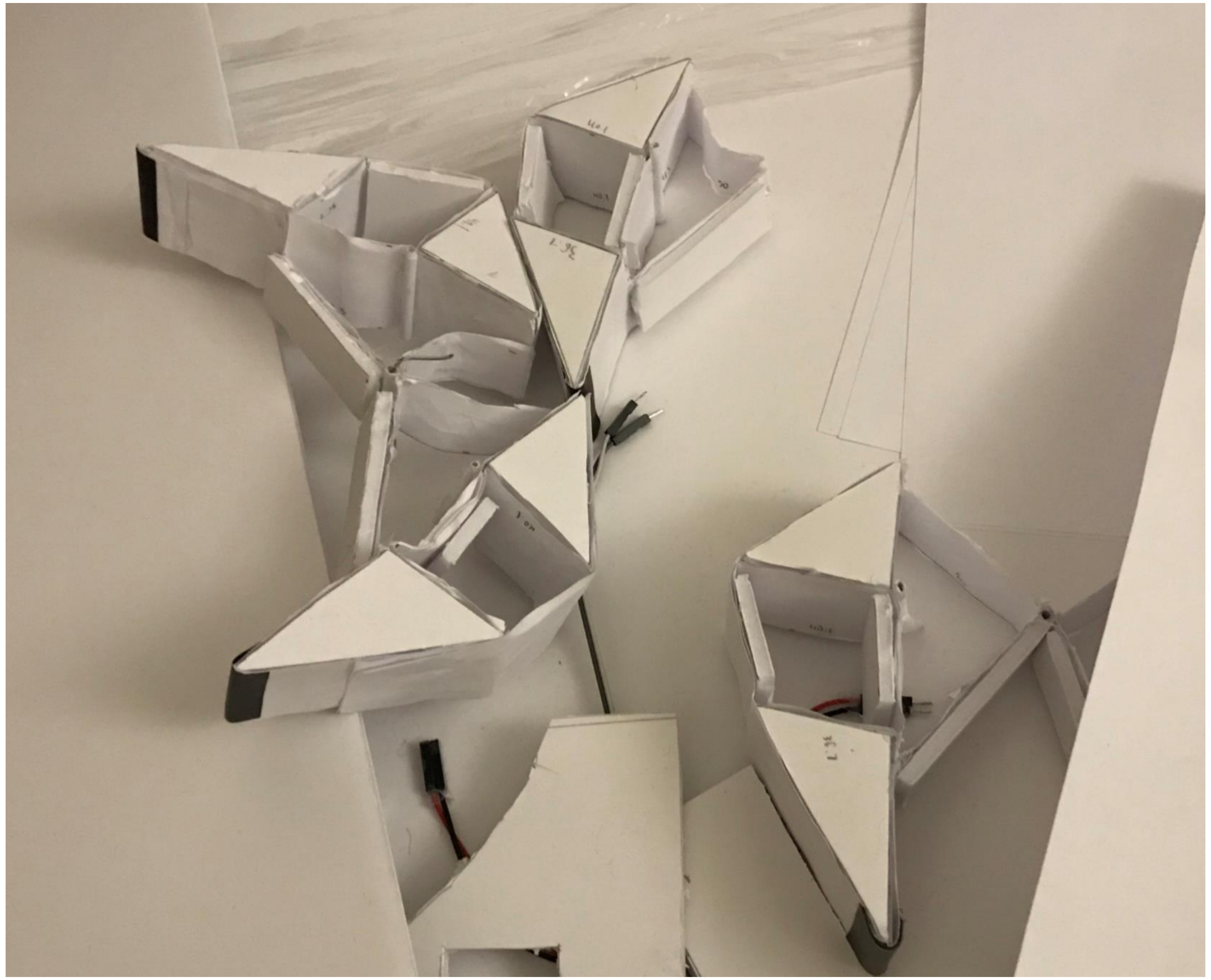
3D Modeling

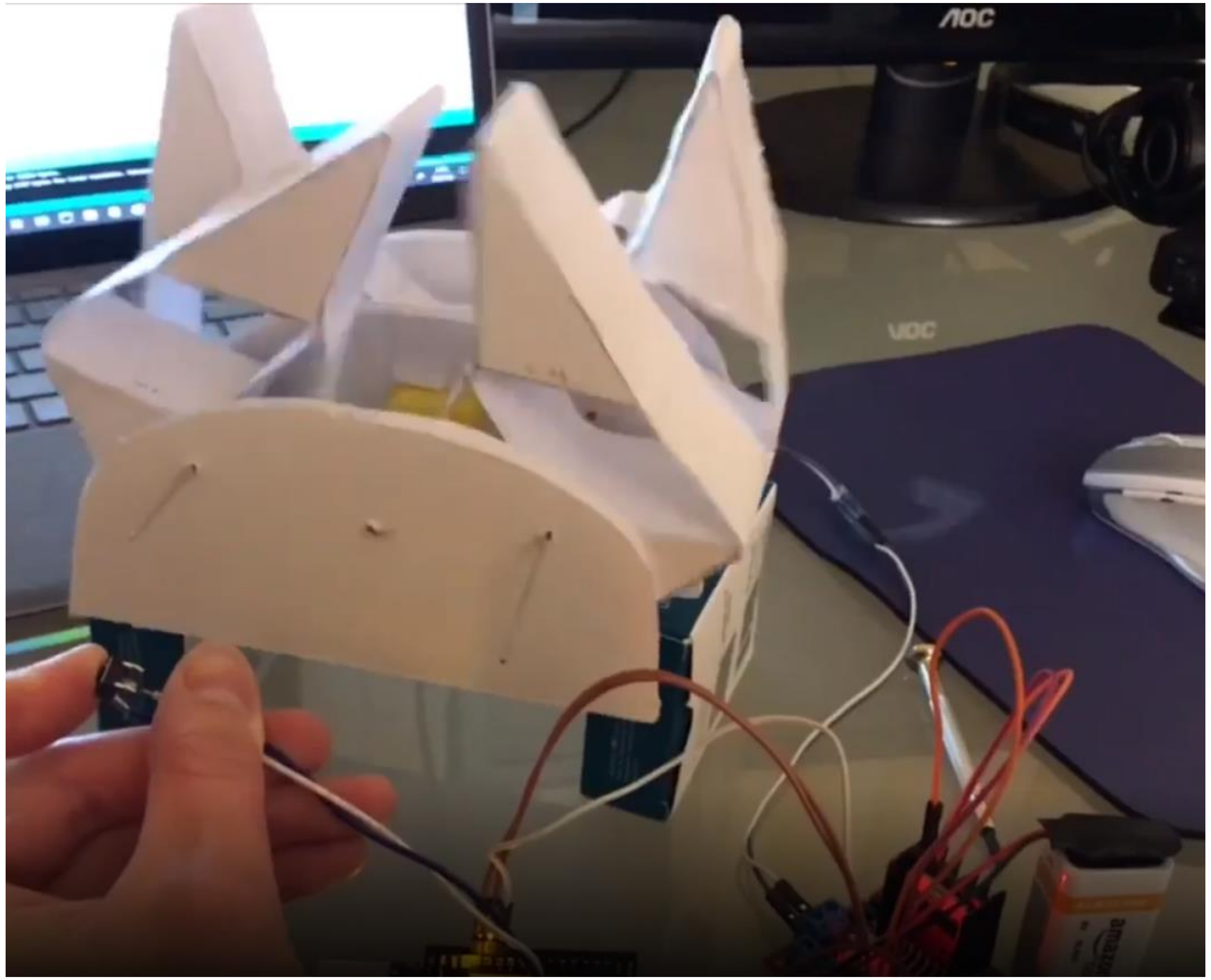
Using Jansen's Linkage, multiple legs could be driven by one motor. Thus, a crab with eight legs would have two motors, each controlling four legs. This would allow crabs to move straight and turn.

3D Modeling

The entire crab, everything from the leg mechanism to the crab shell to the electronics, was modeled in Autodesk Fusion360.

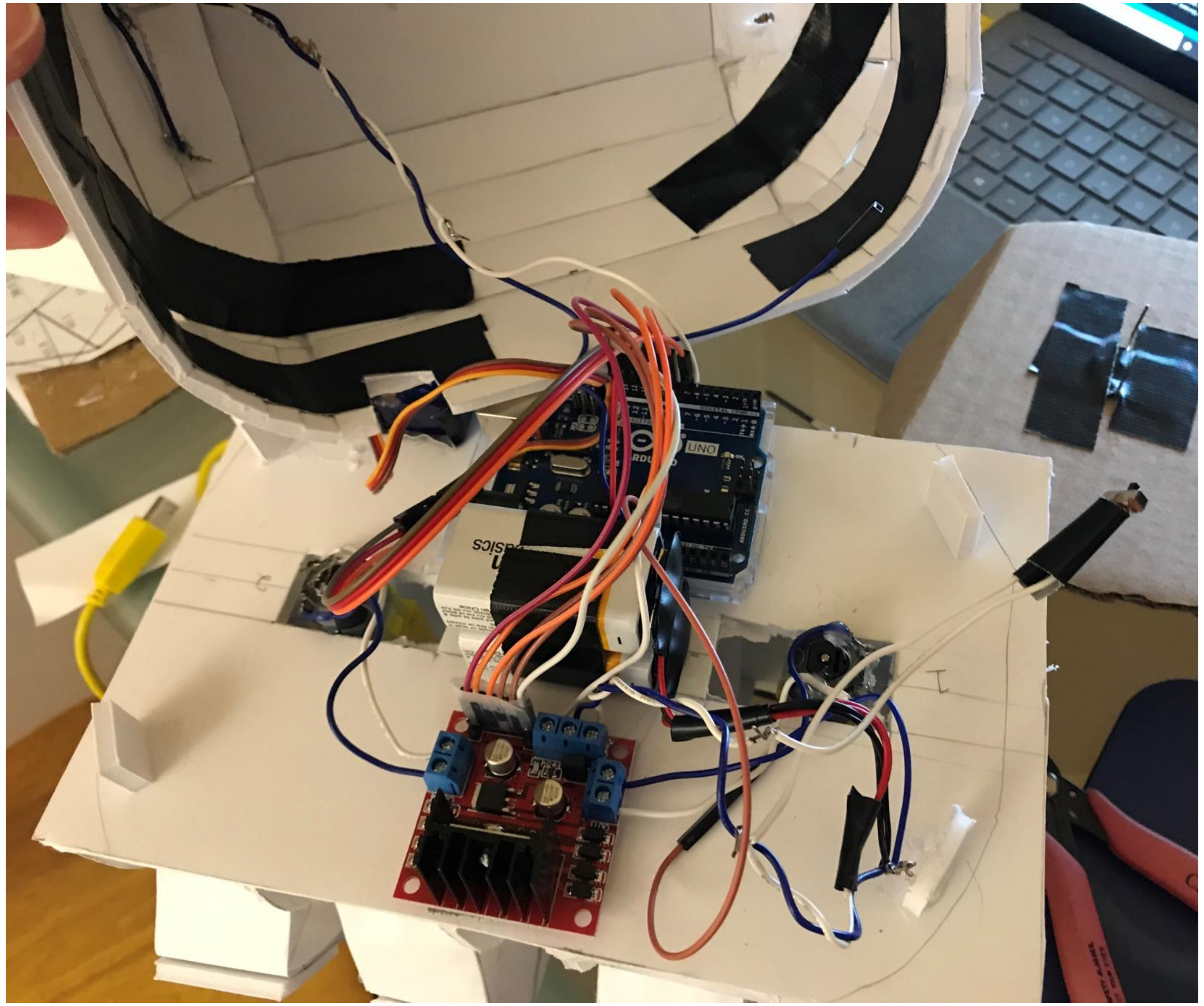


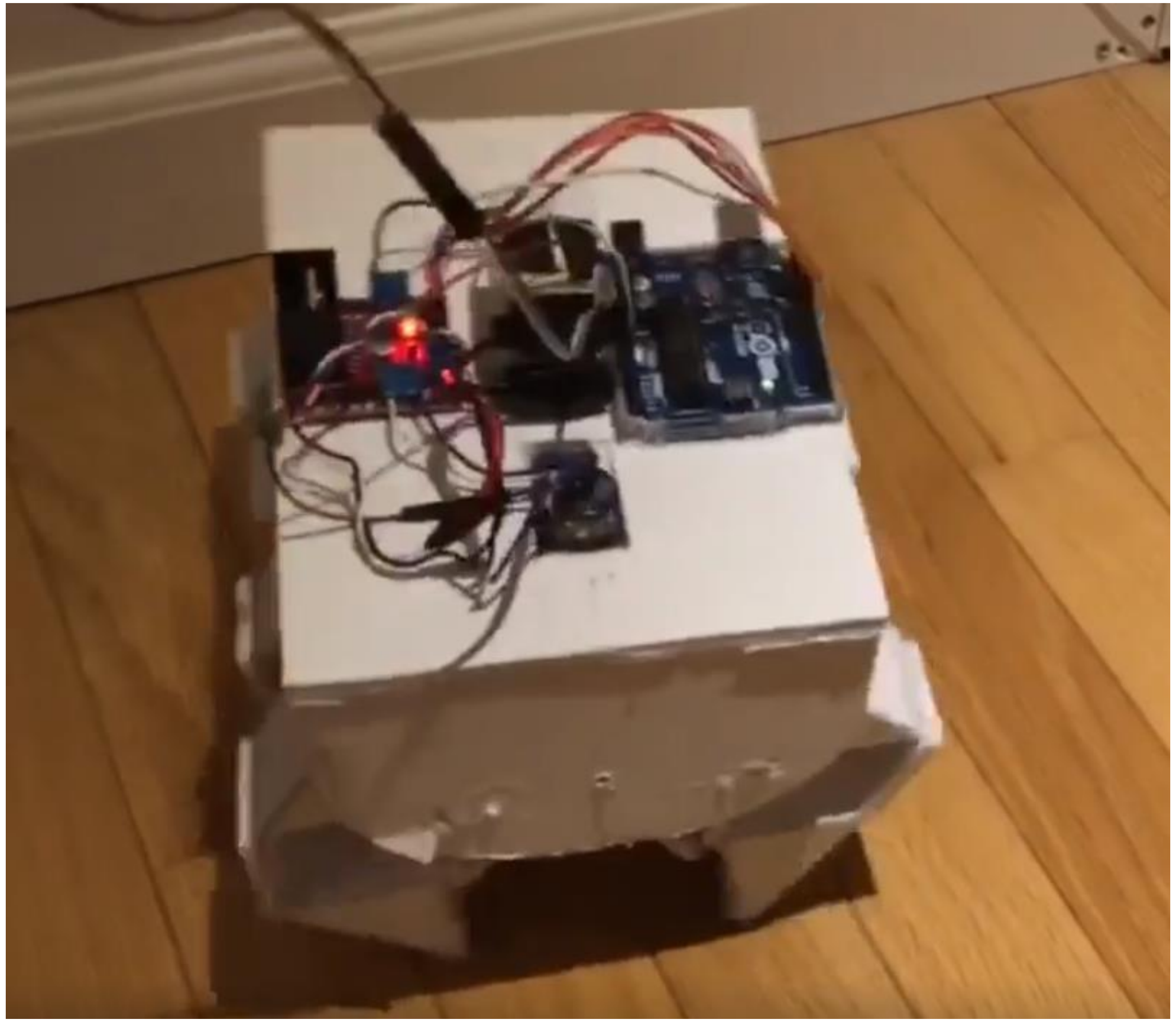




Prototyping

The biggest challenge of prototyping was optimizing the crab's walking curve by tweaking the dimensions of Jansen's linkage for the crab legs. Some linkage segments were shortened to reduce the size of the crab's walking curve, which was important so that the crab could maintain its balance, especially with a high center of gravity.





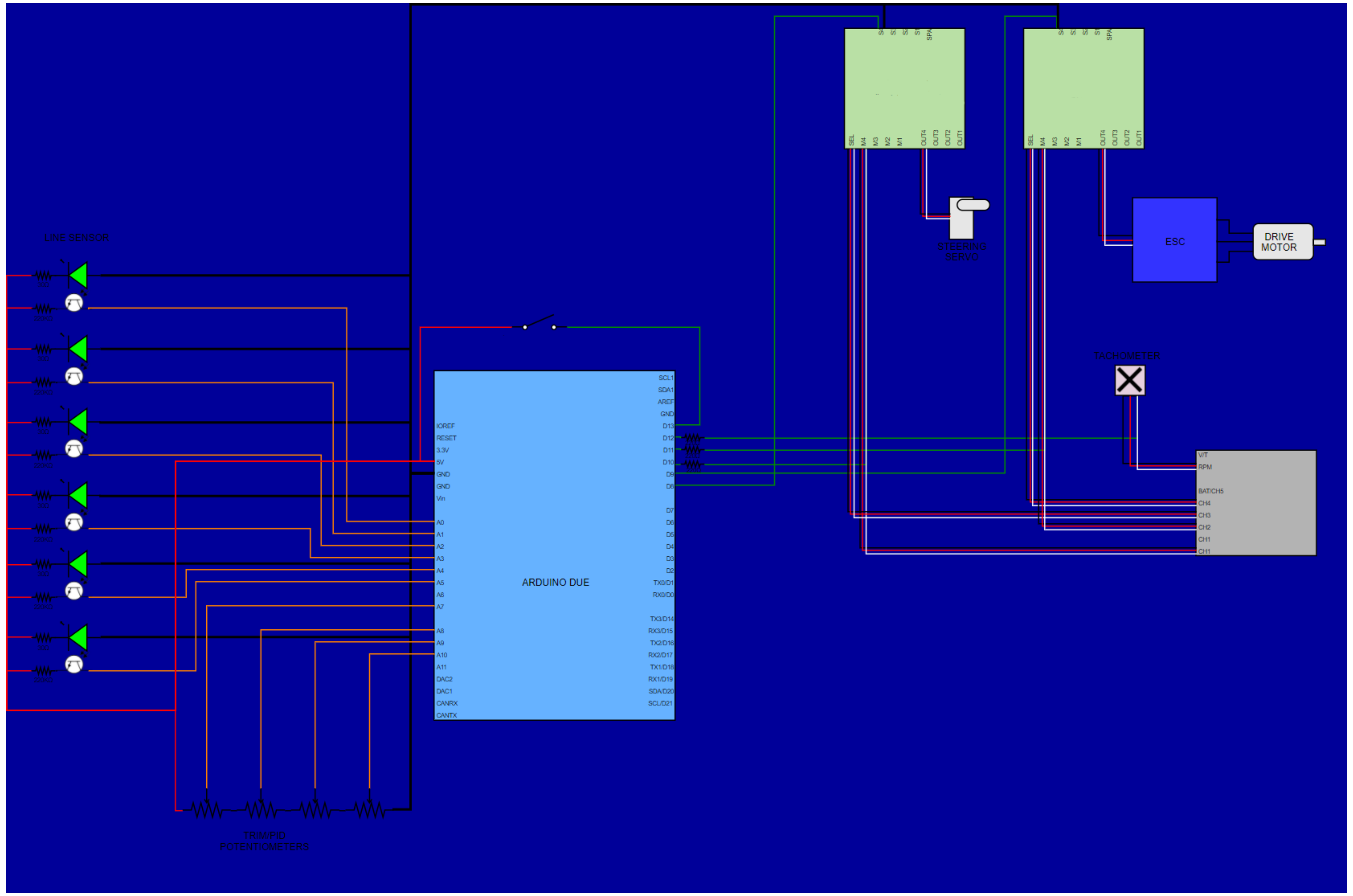
Prototyping

The main crab electronics included an Arduino Uno R3, L298N Motor Driver, NRF24L01 Transceiver, SG90 Micro Servo Motor, two Antrader Dual Shaft Gear Motors, and two 9V batteries.

BeatBot

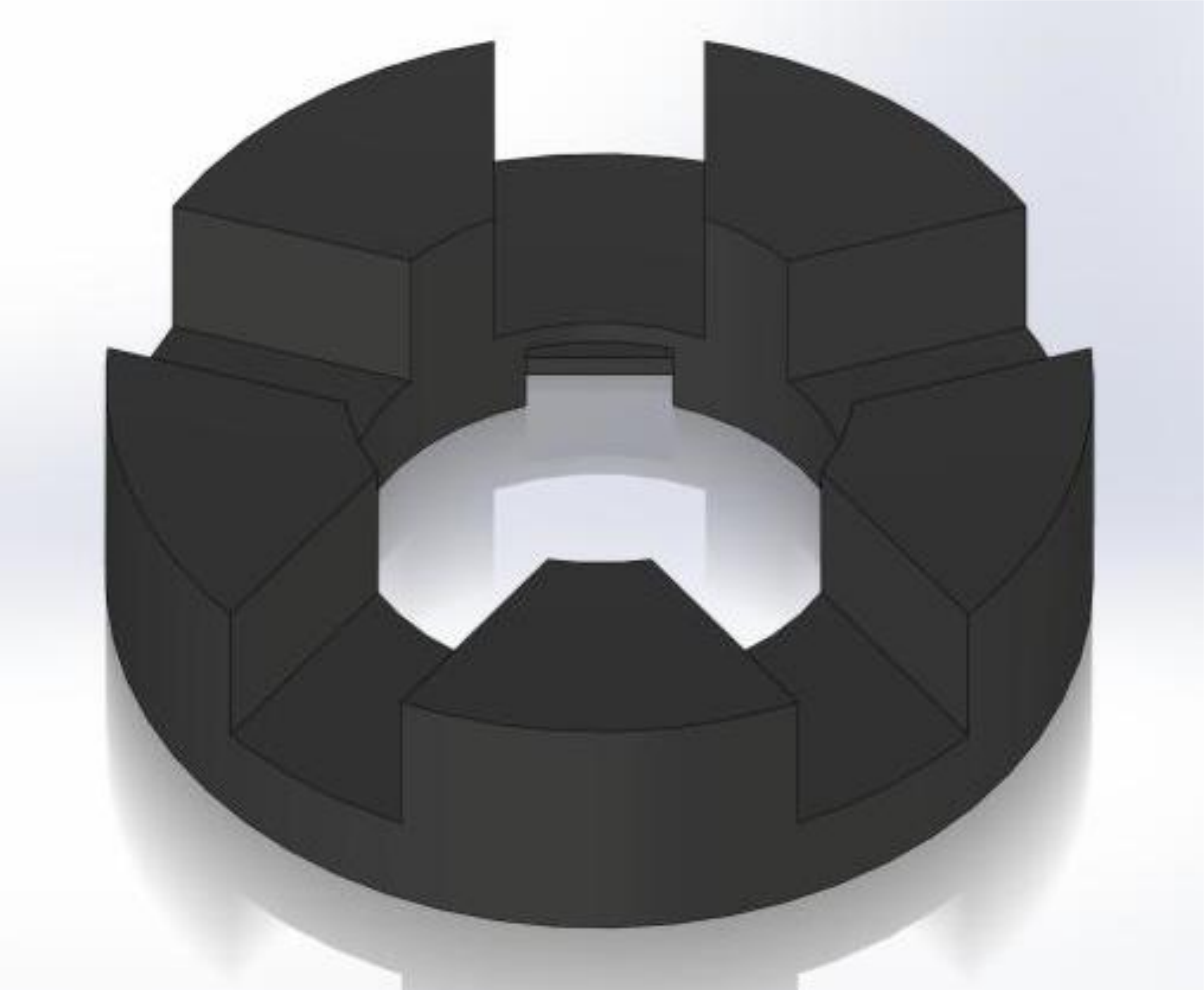
10XBeta
October 2018 – June 2019

An autonomous line-following car that paces long distance runners along a running track. This was a prototype of the second version for the original PUMA BeatBot built by 10XBeta, JWT, and PUMA in 2016.



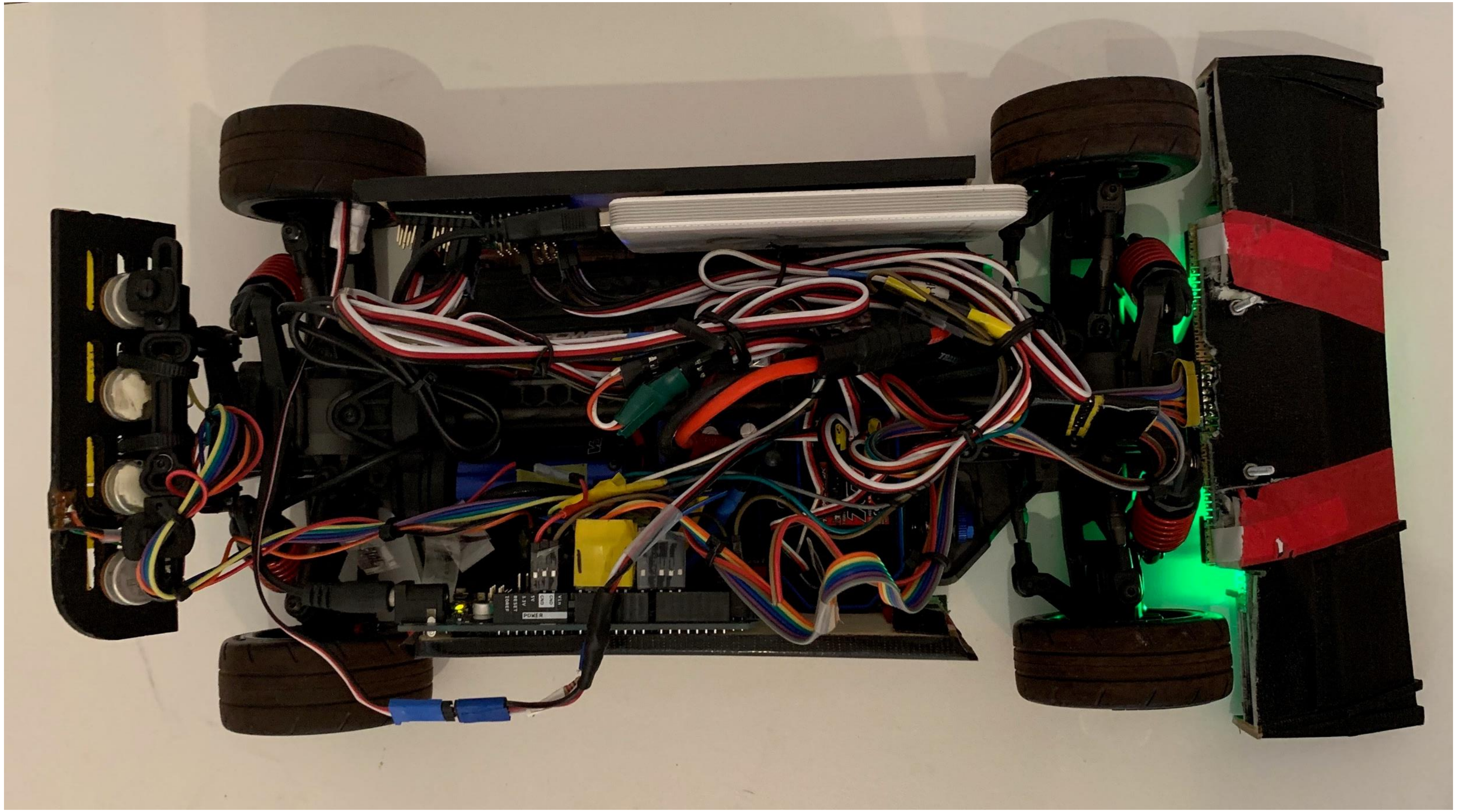
Ideation and Designing

The previous BeatBot had been built on a large, heavy remote-controlled car chassis, which was limiting its maneuverability at high speeds. For the new BeatBot, I decided to go with a smaller and lighter chassis. The challenge thus became fitting the numerous electronic components on a smaller chassis.



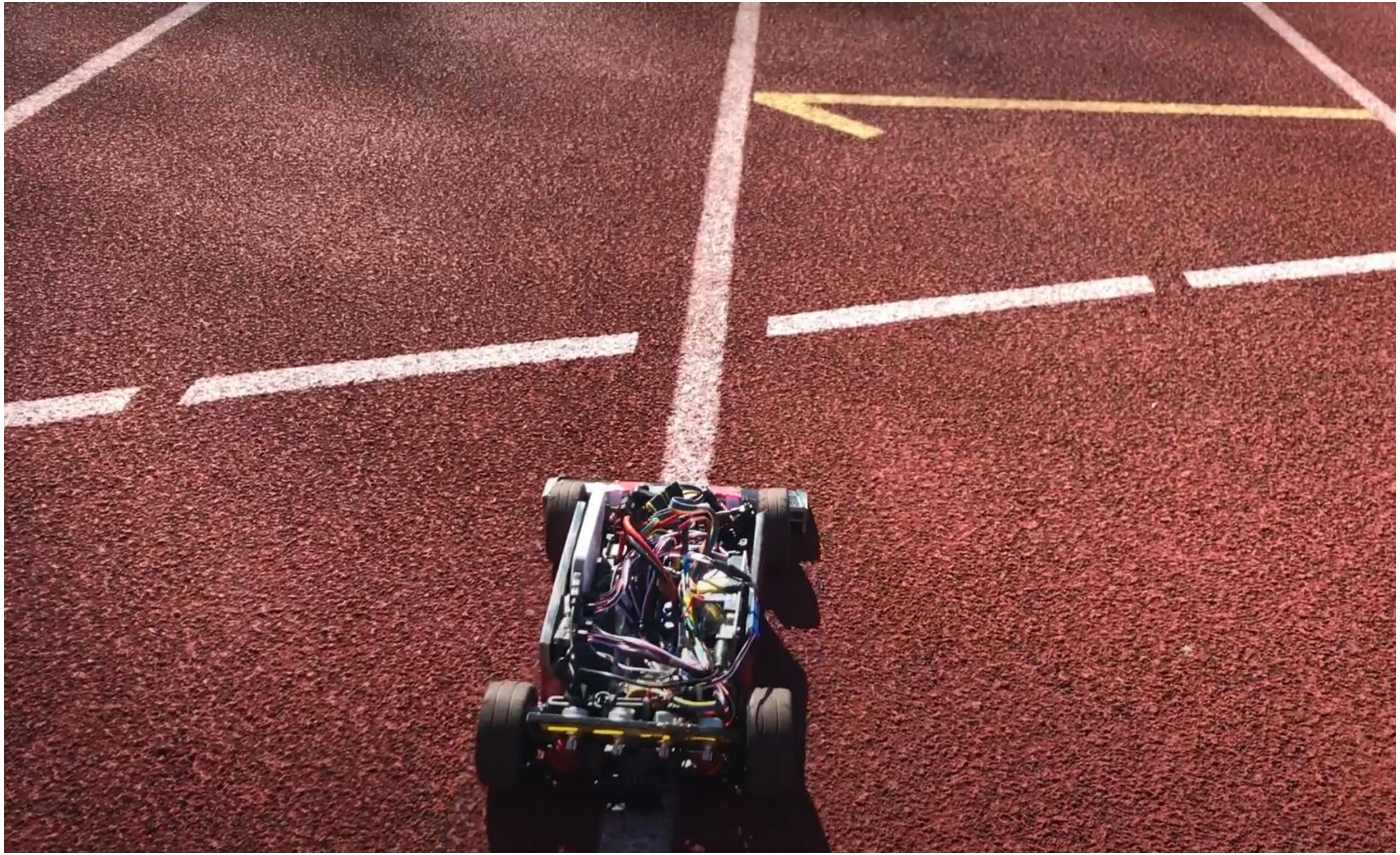
3D Modeling

Because a new chassis was being used, I had to redesign many sensor mounts in Solidworks. For instance, the tachometer magnets used to measure wheel rotations would be housed in a new custom SLA 3D-printed wheel mount that would clip onto the car wheel spokes.



Prototyping

In addition to changing the chassis, I switched many metal pieces of the old BeatBot to 3D-printed plastic pieces that were strong yet light. This change ultimately reduced the prototype weight by 34% and size by 40%.



Programming

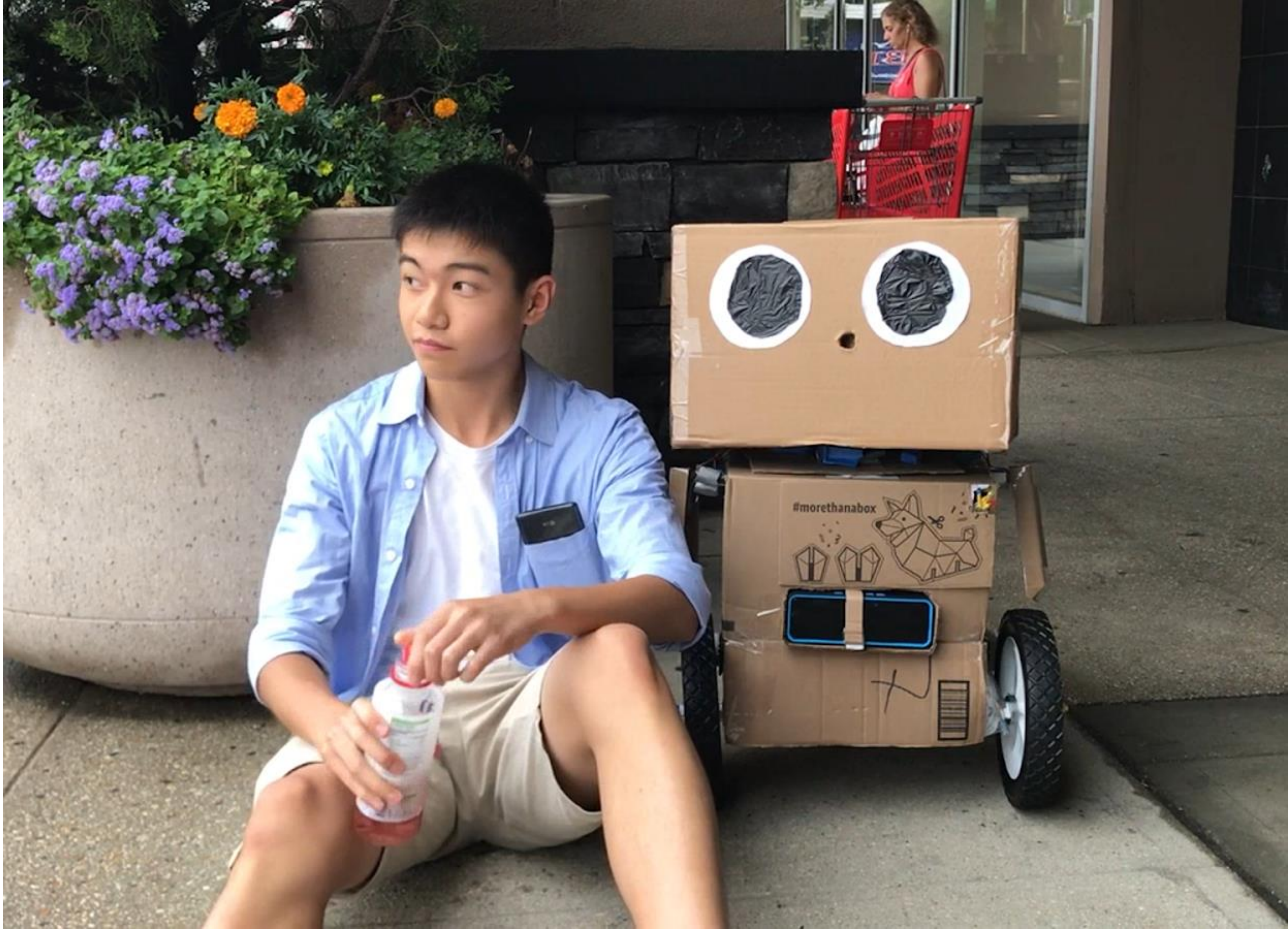
The line-following feature of the BeatBot relied on a PID loop in its software. A large portion of the prototype development cycle was dedicated to retuning the PID values for the new car chassis.

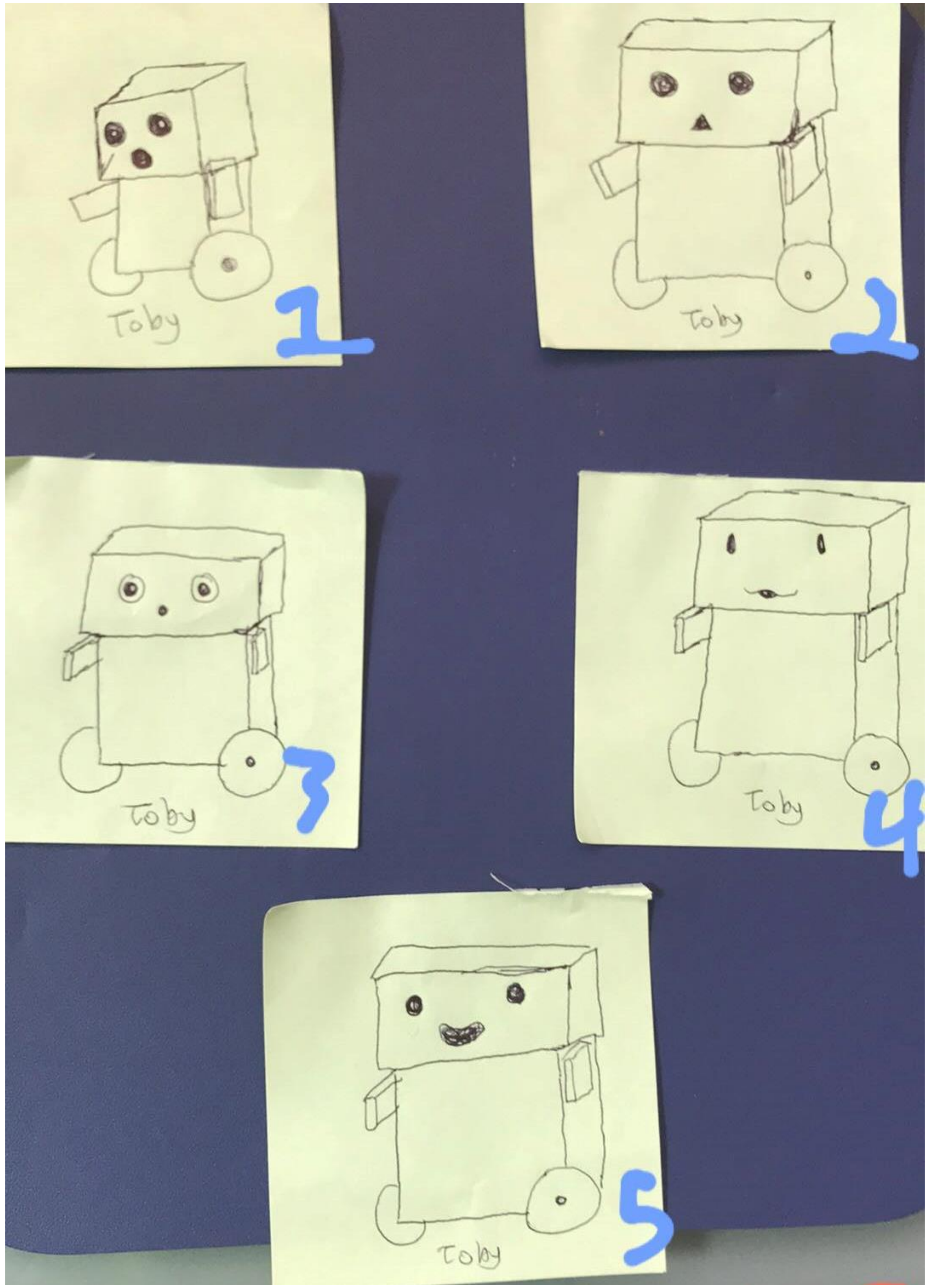
Toby

Personal Project
July – September 2018

A homemade robot built primarily out of ordinary items: cardboard, paper, Legos, cordless drills, and a smartphone.

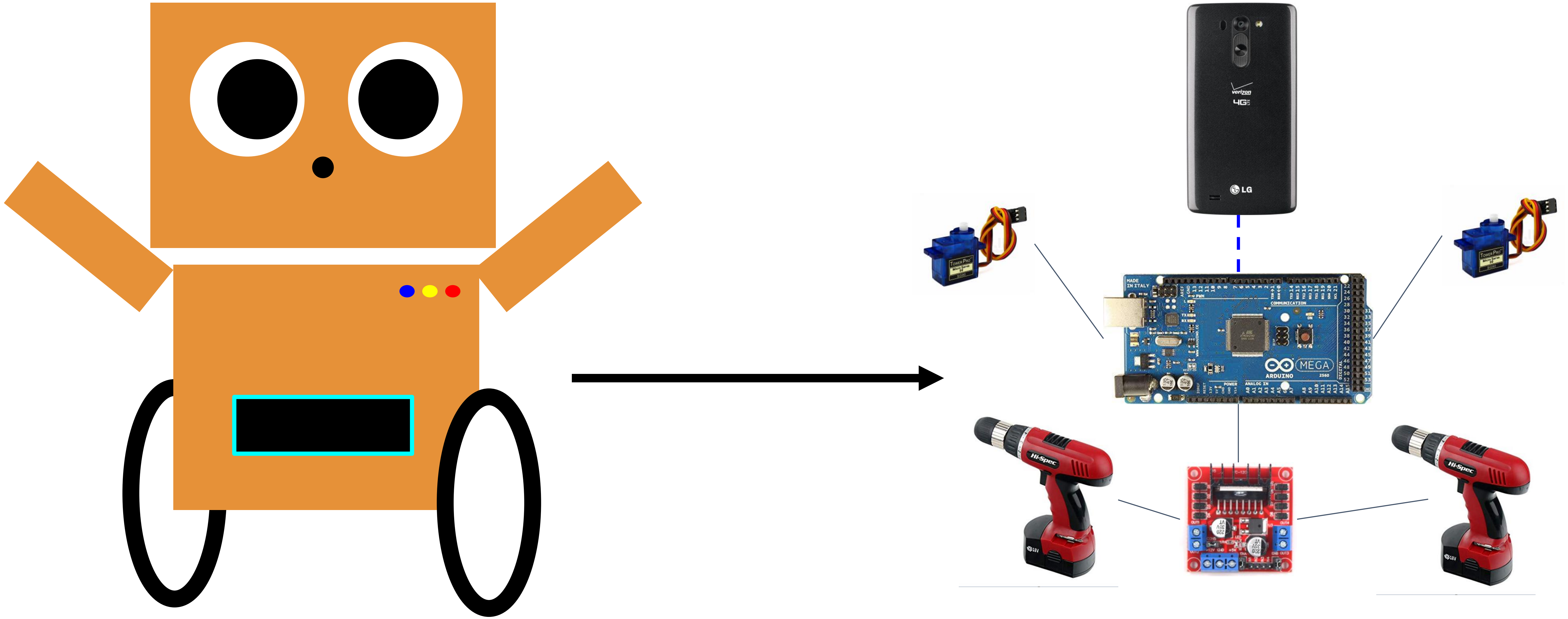
Demo:
youtube.com/watch?v=3ILHPyHL_tg
Arduino Code:
github.com/michaellu2019/Toby
Android Code:
github.com/michaellu2019/Annice-Brain-App





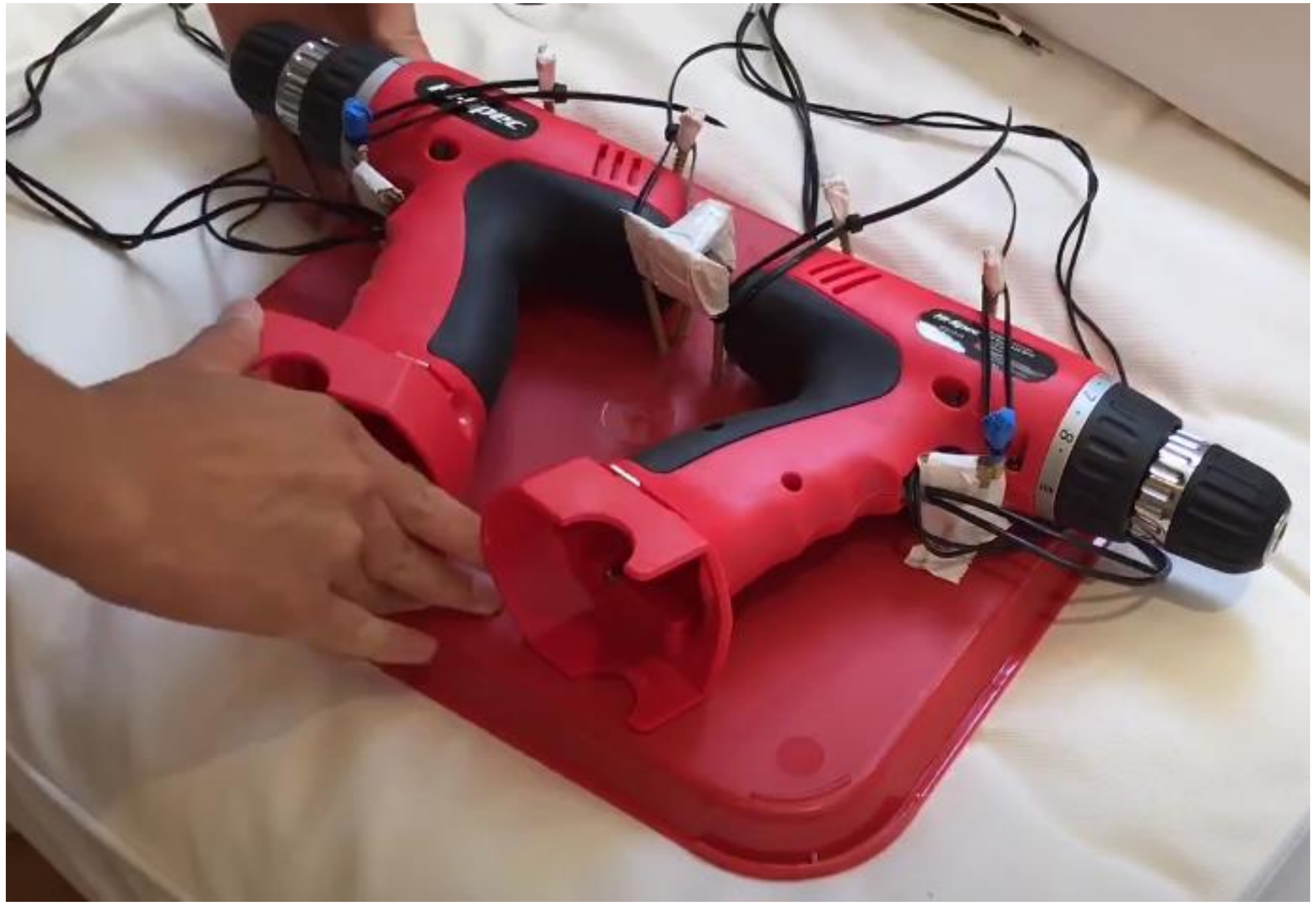
Ideation and Sketching

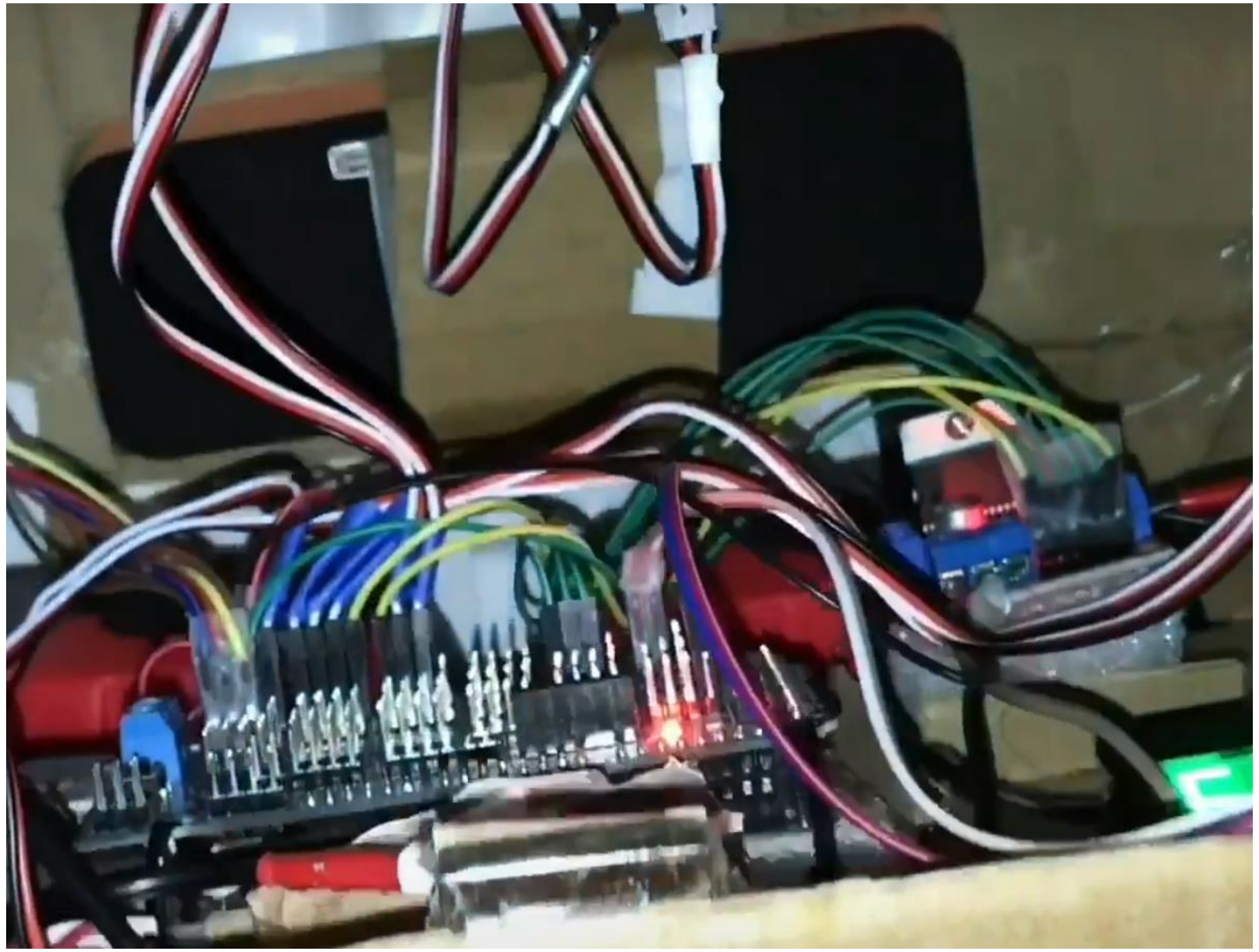
I noticed this #morethanabox illustration on one of my Amazon packages over the summer, and it inspired me to push the limits of what a simple cardboard box could become, so I decided to create a giant robot named Toby out of ordinary items. I especially explored through many post-it sketches how to make Toby seem as friendly as possible.



Ideation and Designing

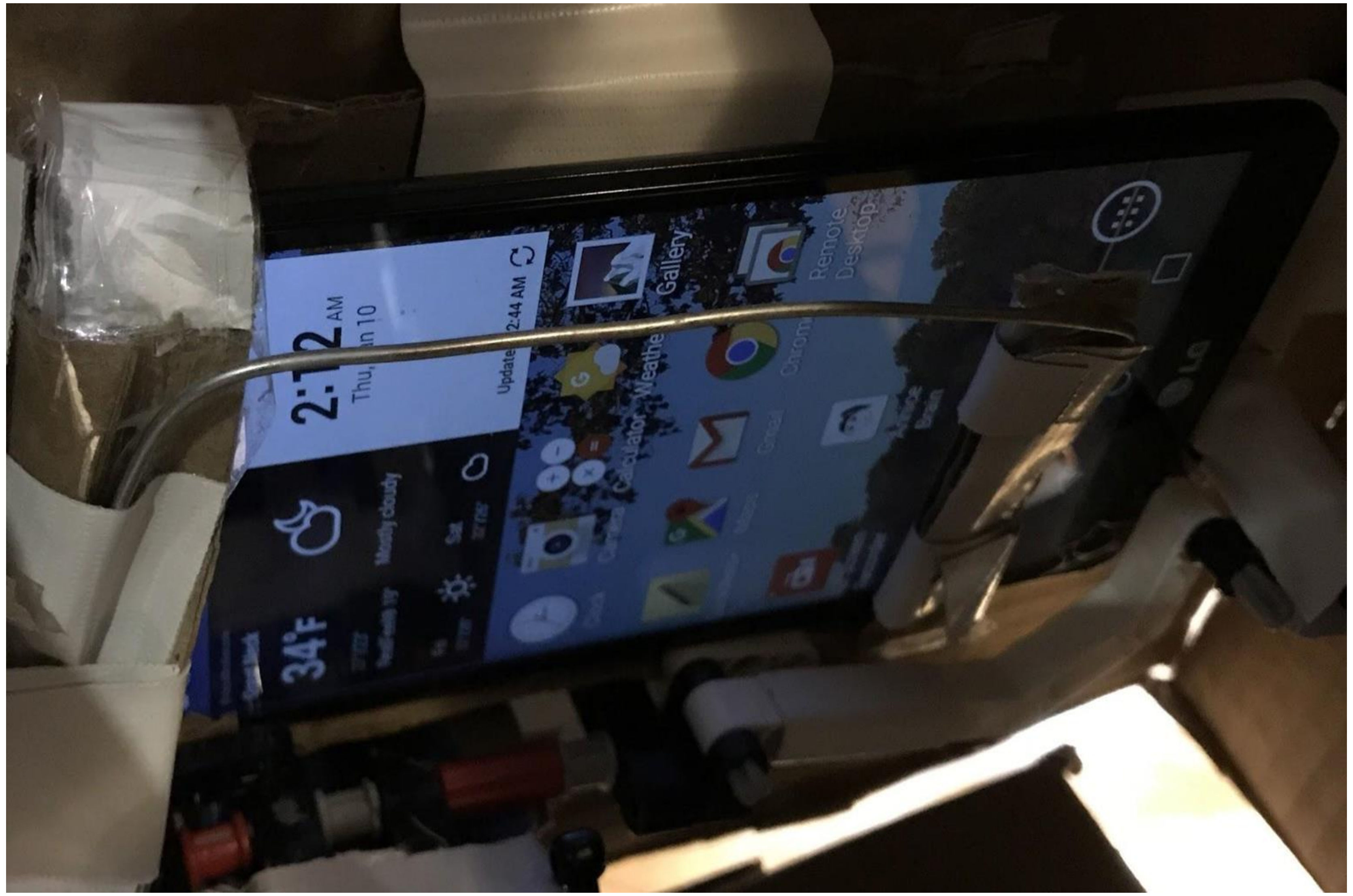
The crux of Toby’s design was that computationally-heavy processes would be handled by a smartphone, which would communicate over Bluetooth with an Arduino Mega to control the movement of Toby with servo motors that controlled his arms and cordless drill motors that controlled his wheels.





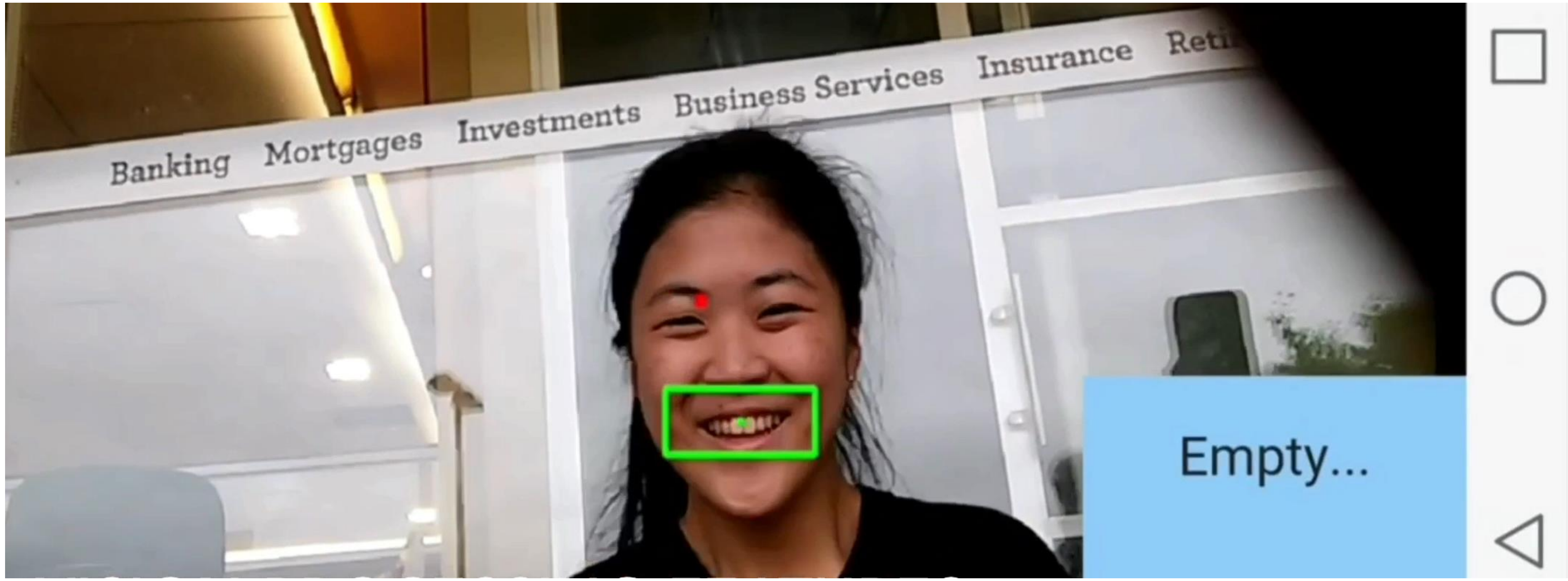
Prototyping

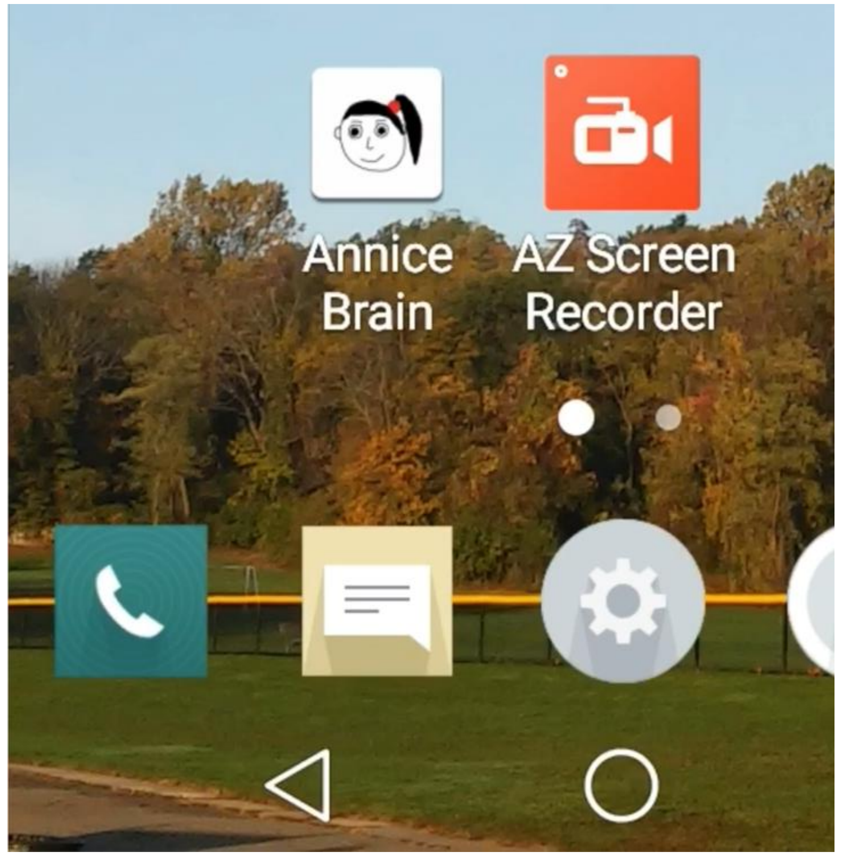
The drive train chassis for Toby consisted of two cordless drills bolted to a plastic lunch tray. The main electronics included an Arduino Mega and Servo Shield, L298N Motor Driver, HC-06 Bluetooth Module, SG90 Micro Servo Motors, and two Hi-Spec Cordless Drills.



Prototyping

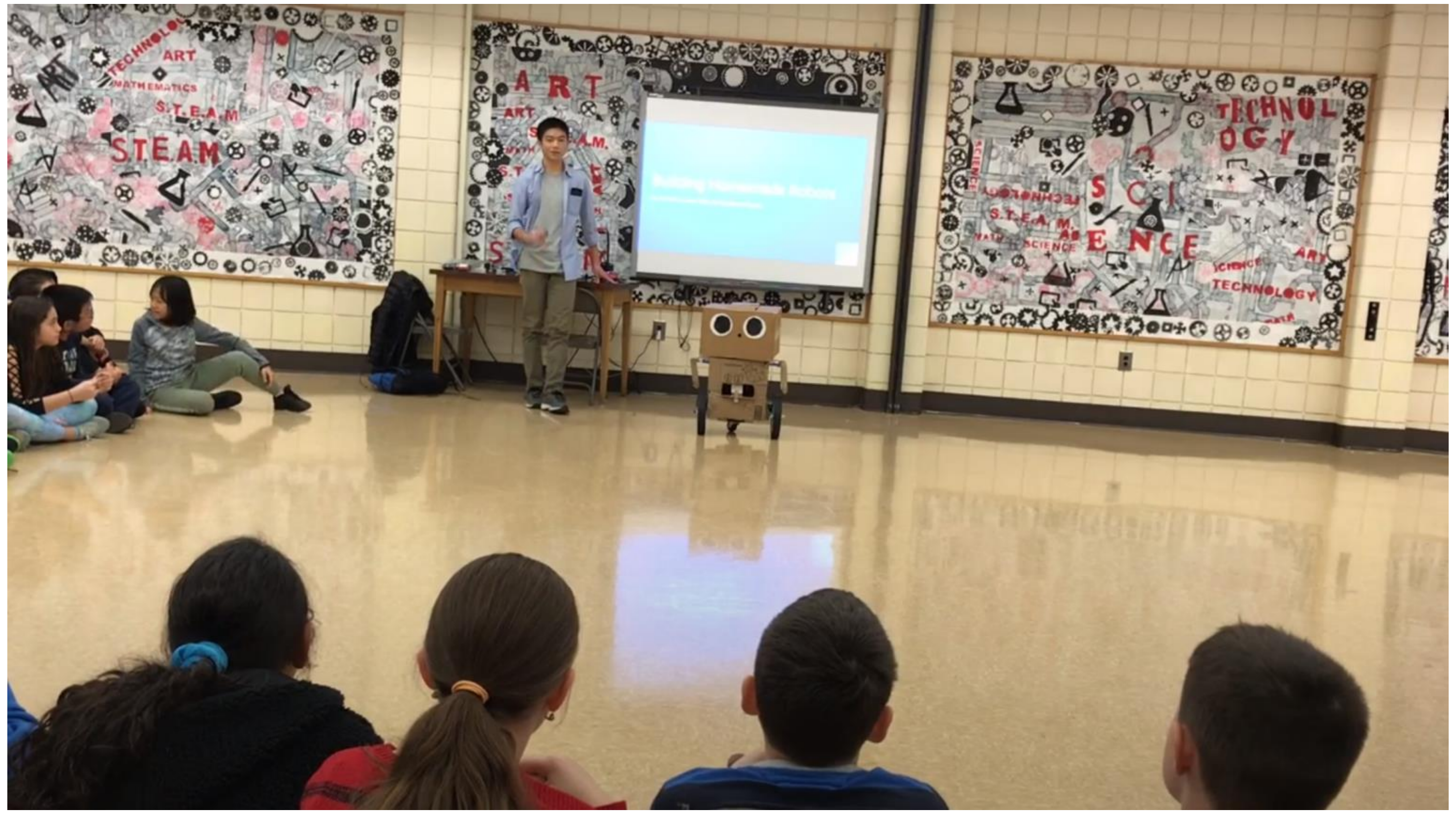
The smartphone was mounted inside Toby’s head and the camera was put behind Toby’s little mouth opening. This would allow Toby to observe his surroundings with the smartphone’s built-in camera and to run computer vision algorithms to respond to his environment.

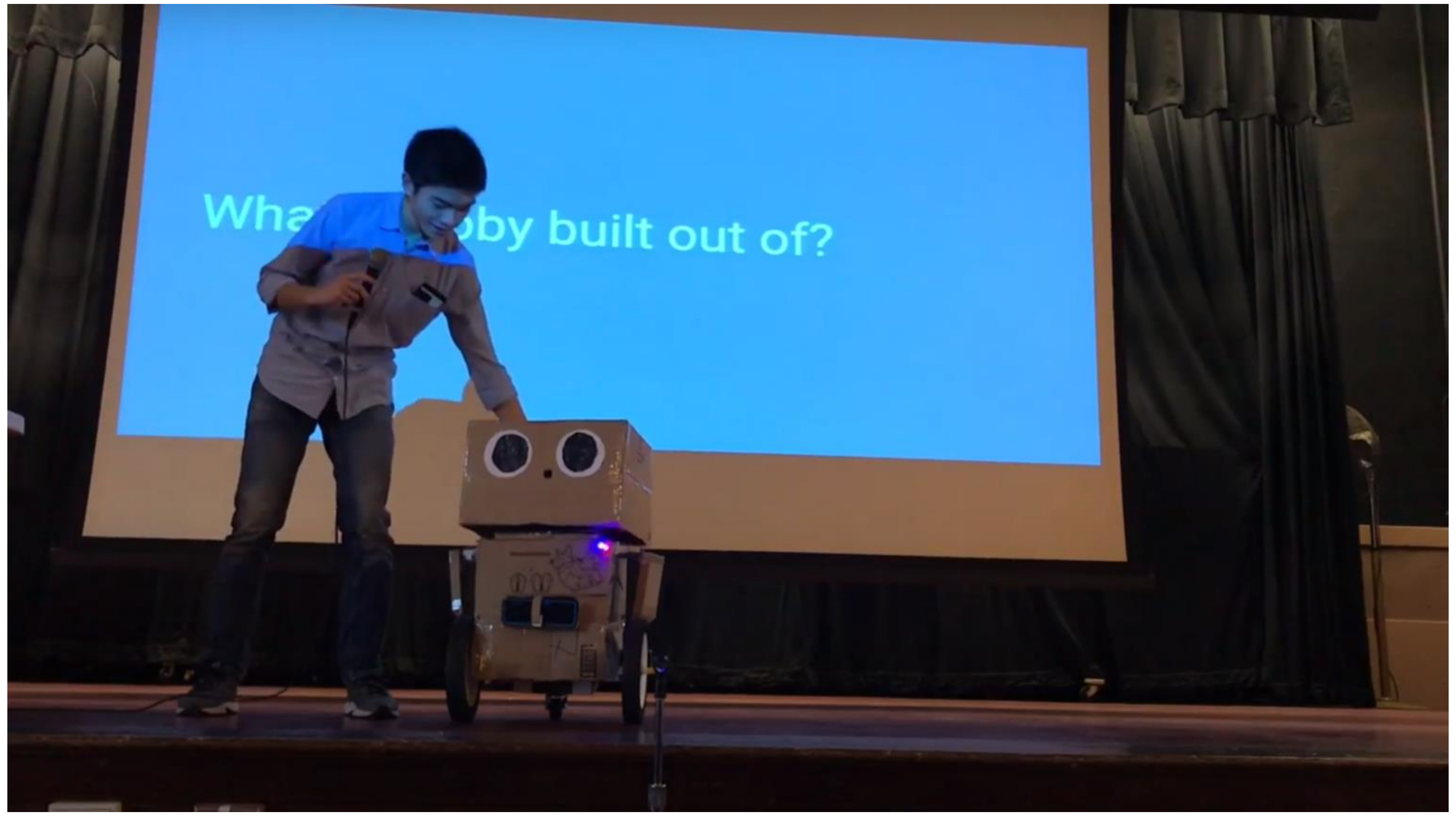




Programming

The smartphone inside Toby's head ran a custom Android app that I wrote called "Annice Brain" so that Toby could carry out intensive computer vision (OpenCV Haar Cascade Classifiers) and speech recognition (CMU PocketSphinx) algorithms.





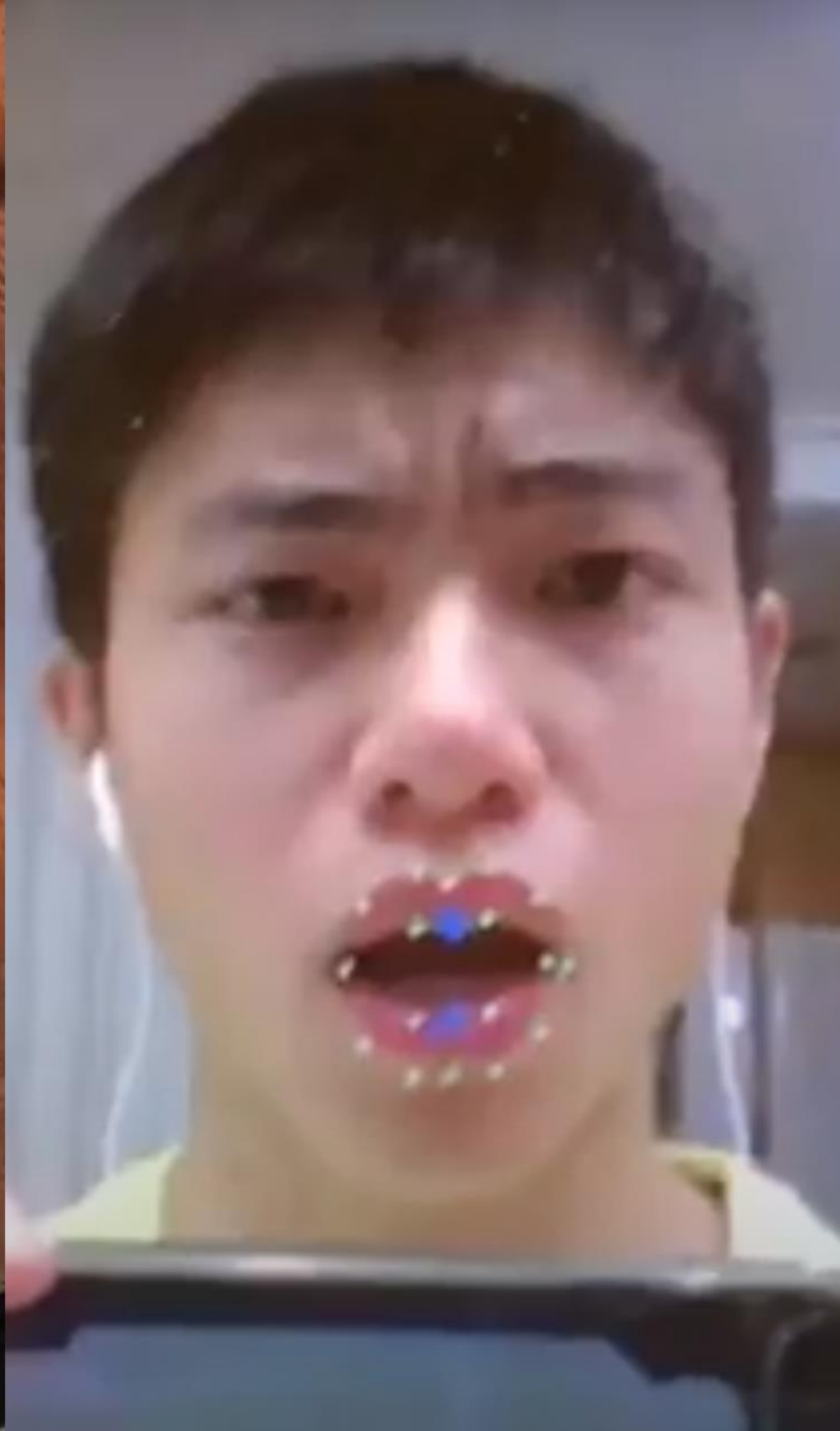
Presenting

Because of the relatively simple materials used for Toby, multiple local elementary schools asked me to present my project to their students to pique their interest in robotics. This was a great way to reconnect with my local community before I left for MIT and to share with other students my passion for STEM.

Robot Horse Head

Personal Project
August 2018

An animatronic horse head that mimics the movements of my mouth based on real-time data from OpenCV algorithms that would track my mouth position.



RC Tank

Personal Project
May 2018

A remote-controlled Arduino toy tank built out of cardboard.

Trash Mobile

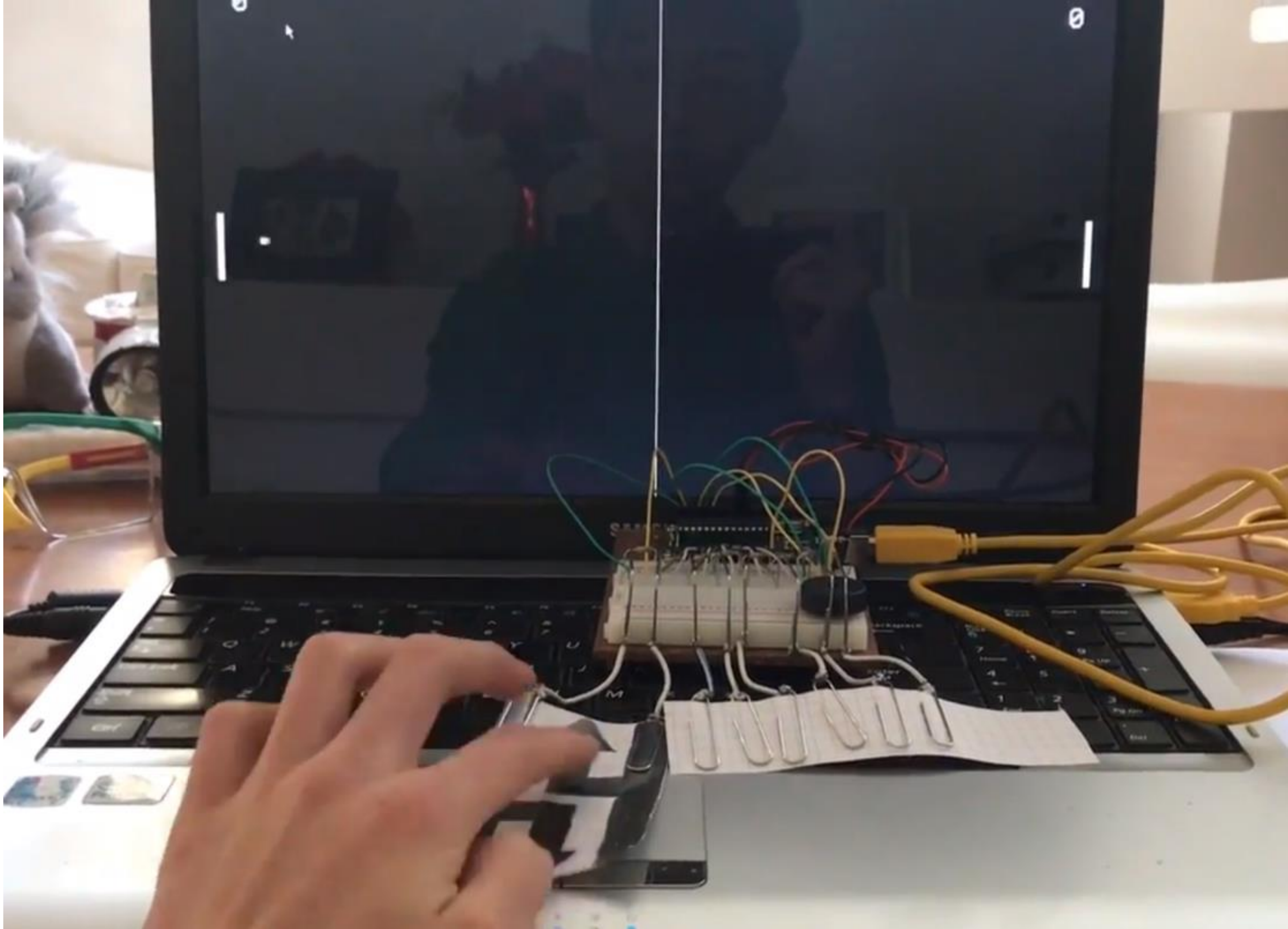
Personal Project
July – August 2017

A homemade electric trash can vehicle built out of "trash": scrap wood, cardboard, old cordless drills, and wheelbarrow wheels.

Capacitive Touch Controller

Personal Project
February 2017

A circuit that used capacitive touch to control a computer. A user would draw a custom controller on paper with a pencil and hook it up to the circuit with several paper clip connectors. They would touch the pencil-shaded sections of their controller to send signals to control the computer.



Thank You

github.com/michaellu2019 mlu0708@mit.edu linkedin.com/in/michael-lee-lu