NYU | TANDON SCHOOL OF ENGINEERING

General Engineering Department

SmartBrella

Project Requirements Specifications

Version 2.0

Project Team Members: Selena Cheung, Kai Shinozaki-Conefrey, Erica Wang, Jack Yang

1. Introduction

1.1 Purpose of Project

The purpose of this project is to provide an accessible, convenient, and smart innovation on the classic umbrella design. While umbrellas are very useful in protecting people from the rain, they can be cumbersome to open and close, especially for those with disabilities, arthritis, amputations, or blindness. Around 15% of people in the world have disabilities, 28% are blind or have vision impairment and 17 million are amputees. This causes many inconveniences when trying to perform simple actions. The umbrella, an item everyone has at home, needs to be manually opened and closed, requiring a lot of strength to open and close. Modifying the umbrella to automatically open and close will help a lot of people who have trouble performing these actions.

By accomplishing this project, we hope to create convenience and improve daily lives with an umbrella made of higher quality. Not only does this umbrella offer greater protection against harsh weather compared to the common alternative, but it also creates a world better suited for people with disabilities. By creating an app that controls the umbrella, we hope to make bigger strides toward building a more friendly environment for the disabled.

1.2 Background of the Experiment

The umbrella is paired with an application that allows the remote opening and closing of the umbrella, as well as the ability to connect with weather APIs and remind the user to bring their umbrella if rain is predicted. The app and the umbrella are connected through a Bluetooth chip, which allows opening and closing messages to be sent to the umbrella. The umbrella also allows those with disabilities or inconveniences to open and close the umbrella through a sound sensor. The sensor is connected to motors that will pull the umbrella to open or close it and is coded through Arduino, a platform that enables the creation of interactive electronic objects.

2. Requirements

2.1 Physical Components

For the product, a standard umbrella was used to make changes. To make the umbrella automatically open and close, a pulley was required. The pulley consisted of two VEX motors, a LEGO gear, a metal axle, a long yarn string, and two shaft collars. The axle is used to connect the two motors together, while also having the gear placed on the center of the axle with shaft collars on each side of the gear to keep it in place. One end of the string is tied onto the gear and the other end is tied to the inside of the umbrella. A KY-038 sound sensor, wires, an Arduino UNO board, a breadboard, and a DC motor driver controller board was used to make the umbrella open and close when the sound sensor detects a sound. To let the motor move when a sound is detected, the motor would have wires connected to the controller board. Other wires were then used to connect the controller board to the breadboard, which would connect to the Arduino UNO board and the sound sensor. A Bluefruit Bluetooth module was also wired onto the breadboard to allow Bluetooth to send messages to the motors. Duct tape was used to put all the previously listed materials together and attached to the umbrella.

Table 1: Cost Estimate Table

Resource	Cost Per Unit	Quantity	Cost ($)
Umbrella	12	1	12
Arduino Kit	20	1	20
Motors	17	2	34
Sound Sensor	10	1	10
BlueFruit Bluetooth Chip	18	1	18
Miscellaneous	-	-	30
Total			$124

The Arduino Kit consists of an Arduino UNO board, wires, a breadboard, and a screwdriver.

2.2 Software Components

The software components of this project consist of the usage of Fusion360, Figma, Arduino IDE and Kivy. Fusion360 is a 3D modeling software platform for product designing and manufacturing. Fusion360 was used to create 3D design prototypes for how the umbrella will look like. Figma is an app interface prototyping tool, it was used to design the different pages of the app and display what the app notification would look like if the app was used. The Arduino IDE, also known as the Arduino Integrated Development Environment, is a software that contains a text editor for inputting code, a text console, a toolbar menu, and a message area. The Arduino IDE was used to code the sound sensor so that whenever the sensor would detect sound, it would move the motors. Kivy is a software platform that allows users to use Python to create apps for mobile devices and desktops. Kivy was used to create our SmartBrella app that would be paired with our umbrella.

3. Procedures

3.1 Physical Construction

To open the particular umbrella used in this prototype, traditionally one would slide upwards a hollow rod that contains a non-moving central rod. The moving rod slides upwards until it reaches the top of the central rod, and this movement extends the collapsed frame of rods holding up the umbrella fabric. To automate this process, a string was used to pull the moving rod upwards. To do so, this string was pushed through a pulley at the top of the central rod and tied around the moving rod so that pulling down on the string would pull the moving rod upwards. The other end of the string was tied around an axle rotated by two motors. Turning the motor would wind the string around the axle, creating tension in the string and creating force redirected by the pulley into pulling the moving rod upwards. Turning the motor the other way would unwind the string, releasing tension and allowing the moving rod to fall downwards, collapsing the umbrella.

The motors were wired to a DC motor driver controller board, allowing the motors to turn in both directions. The DC motor driver controller board was wired to a breadboard along with the microphone and a Bluetooth sensor. Power and grounding were provided to the breadboard by an Arduino UNO powered by a battery.

These components were attached to the handle of the umbrella using duct tape and cardboard. The SmartBrella logo was added on top of this fixture.

3.2 Software Setup

Arduino IDE was used to code the Arduino UNO unit used in the SmartBrella. The libraries used for the Arduino code included the Adafruit BLE library and the standard Arduino library. The script was programmed to recognize the sound of a clap through a microphone, signified by a LOW signal transmitted from the microphone's input pin, and decide whether to stop the motor or begin turning it clockwise or counterclockwise. The script kept track of the last direction turned.

To code the app interface, Python was used with Kivy, a plugin geared towards developing mobile application GUIs. Adafruit's Bluetooth plugins were used to manage sending data between the app and the Arduino UNO in the umbrella.

```
//
// VARIABLES
//

int motor2pin1 = 3;
int motor2pin2 = 4;

int motor1pin1 = 5;
int motor1pin2 = 6;

int windstate = 0;
int incomingByte = 0;
unsigned long last_event = 0;
int microphonePin = 8;


// MAIN CODE

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);

  pinMode(motor1pin1, OUTPUT);
  pinMode(motor1pin2, OUTPUT);
  pinMode(motor2pin1, OUTPUT);
  pinMode(motor2pin2, OUTPUT);
  pinMode(microphonePin, INPUT);

}
```

```
void loop() {

  //microphone detection
  int output = digitalRead(microphonePin);
  if (output == LOW) {
    if (millis() - last_event > 25) {
      switch(windstate) {
        case 0:
          startClockwise();
          Serial.println("Clockwise");
          windstate = 1;
          break;
        case 1:
          stopAll();
          Serial.println("Stop");
          windstate = 2;
          break;
        case 2:
          startCounterClockwise();
          Serial.println("CounterClockwise");
          windstate = 3;
          break;
        case 3:
          stopAll();
          Serial.println("Stop");
          windstate = 0;
          break;
      }
      Serial.println("Clap");
    }
    last_event = millis();
  }

  //Debug from log
  if (Serial.available() > 0) {
      // read the incoming byte:
      incomingByte = Serial.read();

      // say what you got:
      if(incomingByte == 48) {
        stopAll();
        Serial.println("Stop");
      }
      if(incomingByte == 49) {
        startClockwise();
        Serial.println("Clockwise");
      }
      if(incomingByte == 50) {
        startCounterClockwise();
        Serial.println("CounterClockwise");
      }
  }

}
```

```
// FUNCTS

void startCounterClockwise() {
  digitalWrite(motor2pin1, HIGH);
  digitalWrite(motor2pin2, LOW);
  digitalWrite(motor1pin1, HIGH);
  digitalWrite(motor1pin2, LOW);
}

void startClockwise() {
  digitalWrite(motor2pin1, LOW);
  digitalWrite(motor2pin2, HIGH);
  digitalWrite(motor1pin1, LOW);
  digitalWrite(motor1pin2, HIGH);
}

void stopAll() {
  digitalWrite(motor2pin1, LOW);
  digitalWrite(motor2pin2, LOW);
  digitalWrite(motor1pin1, LOW);
  digitalWrite(motor1pin2, LOW);
}
```

Figure 1: Arduino Code

3.3 Software Troubleshooting

The most difficult part of the software was attempting to set up the Bluetooth connections between the application as a central unit and the Arduino UNO peripheral. The particular Bluetooth chip used for the umbrella had limited online documentation, making it difficult to take advantage of the Bluetooth functionality.

4. Milestone and Final Product Requirements

4.1 Benchmark A Requirements

The preliminary CAD model in Fusion/Solidworks for the umbrella was required. The umbrella did not need to include sensors yet. However, at least some sort of plan for attaching sensors/microcontroller to the umbrella as well as a system for automatic opening and closing of the umbrella was required. An .STL and a .gcode file of the team logo needed to be submitted through the 3D Printing Submission portal on the EG website.

The Preliminary Design Investigation, which was an outline of the project idea, inspiration, and goals, needed to have a cover page, project overview, goals and objectives, design and approach, cost estimate, project schedule, and relevant pictures. The engineering notebook needed to have a record of work done up to this point, as well as future plans and goals.

The flowchart of code for the entire project was also required. When to open and close the umbrella, when to report information to the app, and any additional code for the sensors needed to be shown in the flowchart of the code. It did not need to include plans of the specific sensors for the umbrella but a general idea of how to code the sensor into the umbrella was needed. Soldering training must have been completed and checked off.

4.2 Benchmark B Requirements

The CAD model in Fusion/Solidworks for the umbrella needed to be updated. It needed to include more detail on automatic opening and closing of the umbrella, as well as any sensors that would have been used. The engineering notebook needed to be updated to accurately depict work done since Benchmark A. The submissions of the .STL and .gcode files of the team logo needed to be approved through the 3D Printing Submission portal on the EG website.
The front-end of the app for SmartBrella needed to be started. While the app did not need to be completely coded, there should have been multiple images of the different usages of the app, such as the main screen, the page for closing/opening the umbrella, and showing the weather.
The circuit for the chosen sensor and motor was required to be successfully wired on Tinkercad.

4.3 Final Submission Requirements

Our final submission requirements were to have the final CAD model completed in Fusion 360 with all its necessary components. In the 3D model, the attached motor and sensor needed to be shown on the umbrella as well as the pulley system. Additionally, our team engineering notebook needed to be updated and finished. As for the physical parts of our umbrella, the motor and sensor needed to be coded, powered and wired correctly for our umbrella’s pulley system to work. The app component for our product needed to have a fully designed front-end and a semi-functional back-end that included the code for the navigation buttons for the app’s user interface.

4.4 Human Resources and Training

Each individual team member of our RAD team had successfully completed the required Makerspace assessment and training orientation. Further, we had all completed the necessary 3D printer training to use the Ultimaker and the soldering training, which proved helpful when putting together the electronic components for our umbrella. In addition to the training we received, we had also utilized the resources (e.g., duct tape, twine, metal rings and dowels) in Open Lab and gained valuable insight from our RAD mentor via weekly meetings on how we could approach our project feasibly. During sessions with our mentor, what was accomplished included discussions around prototyping, important deadlines, new ideas and how to implement them to our mechanized umbrella.

5. Results

5.1 Benchmark A Results

The preliminary CAD model (Figure 1) was created in Fusion.

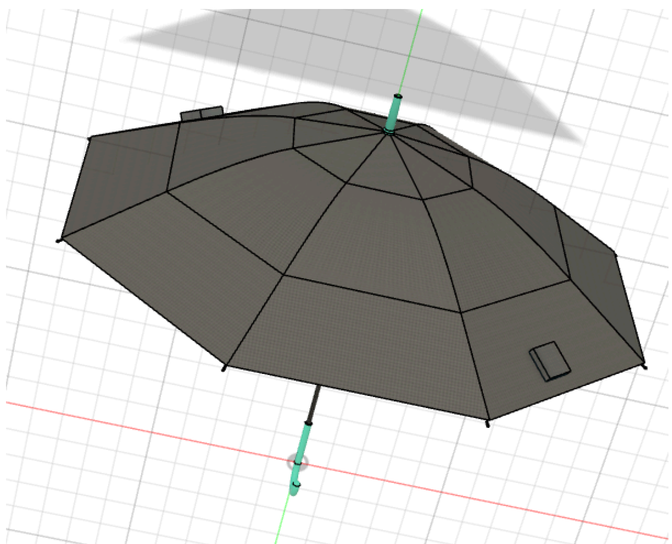

Figure 2: Preliminary CAD model

An .STL and a .gcode file of the team logo (Figure 2) were created and submitted through the 3D Printing Submission portal on the EG website.

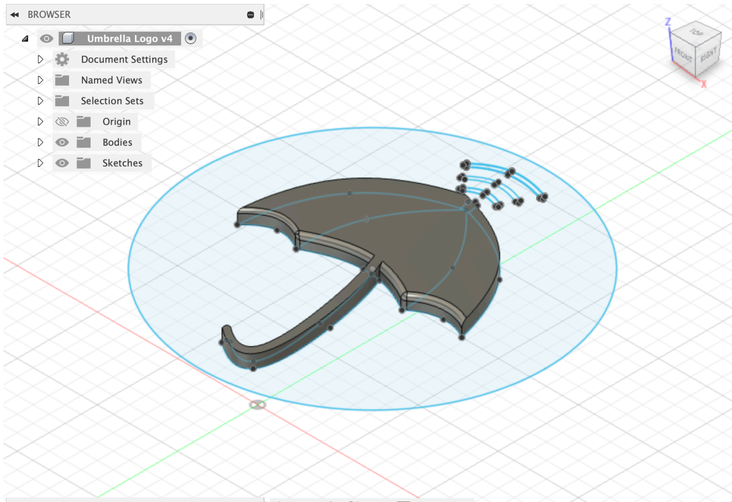

Figure 3: SmartBrella Logo

The cover page, project overview, goals and objectives, design and approach, cost estimate, project schedule, and relevant pictures of the Preliminary Design Investigation were written. The engineering notebook was updated.

The flowchart of code (Figure 3) for the umbrella's desired function was created.

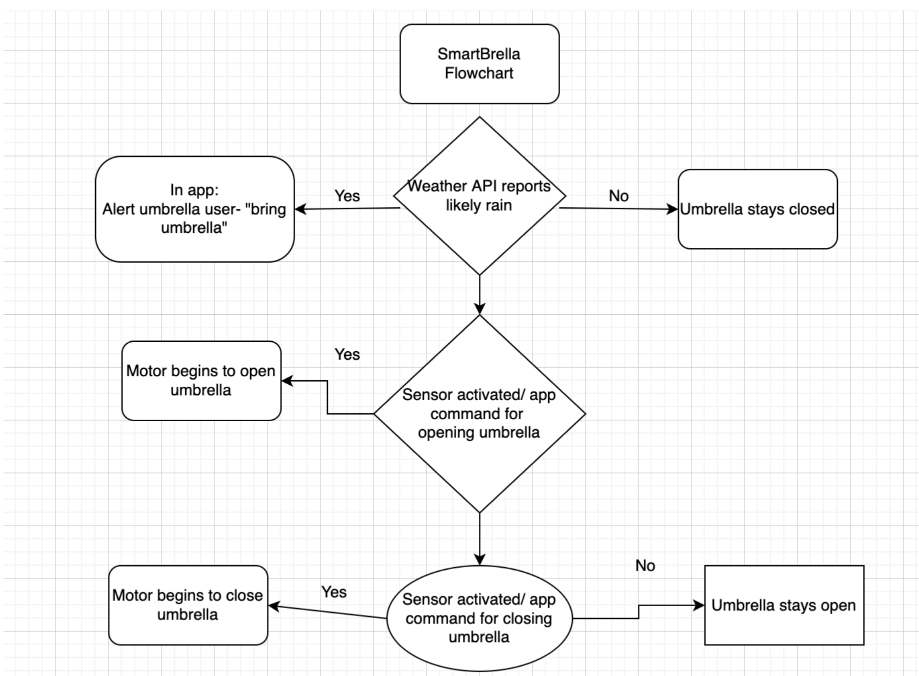

Figure 4: Flowchart of SmartBrella

When to open and close the umbrella, when to report information to the app, and any additional code for the sensors needed were shown in the flowchart of the code. It did not need to include plans of the specific sensors for the umbrella but a general idea of how to code the sensor into the umbrella was illustrated.

Soldering training was completed and checked off.

5.2 Benchmark B Results

The CAD model for SmartBrella (Figure 4) was updated in Fusion.

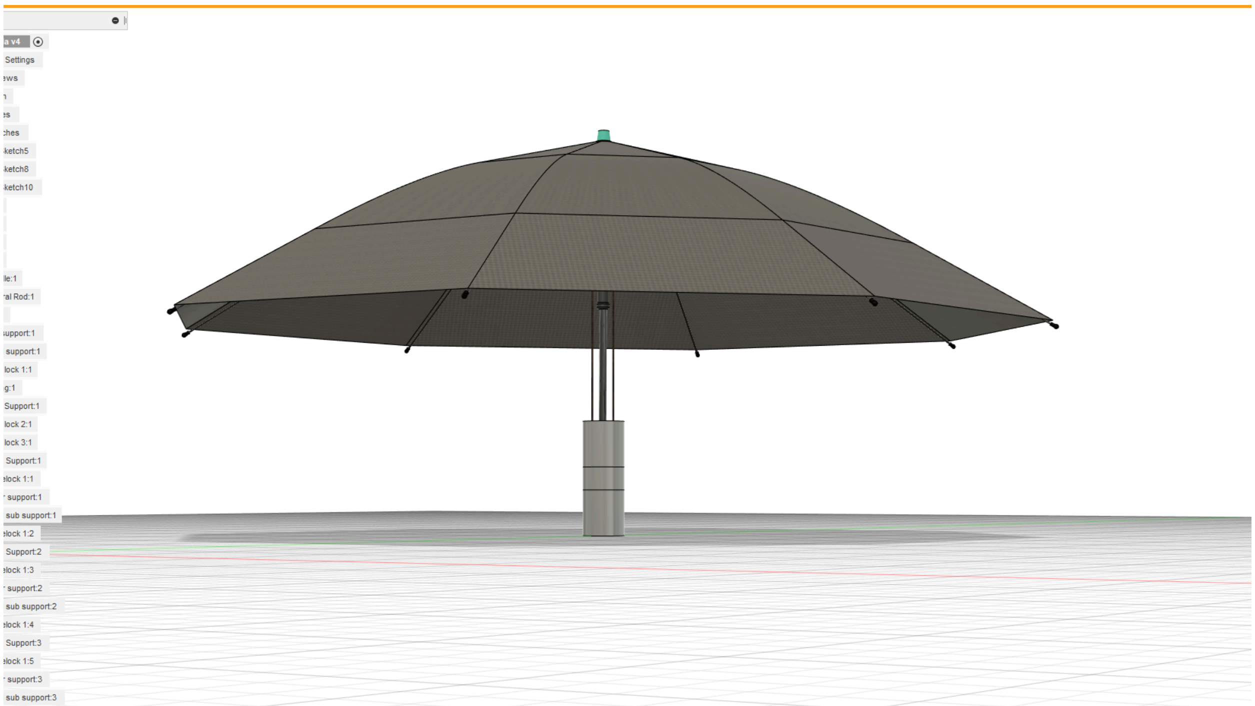

Figure 5: Updated CAD Model for SmartBrella

It included more detail on the automatic opening and closing of the umbrella. The two cables in the center of the umbrella, as well as the bigger handle to contain the sound sensors and motor were the new updates.

The engineering notebook was updated to accurately depict work done since Benchmark A. The submissions of the .STL and .gcode files of the team logo (Figure 5) were approved and 3D printed.

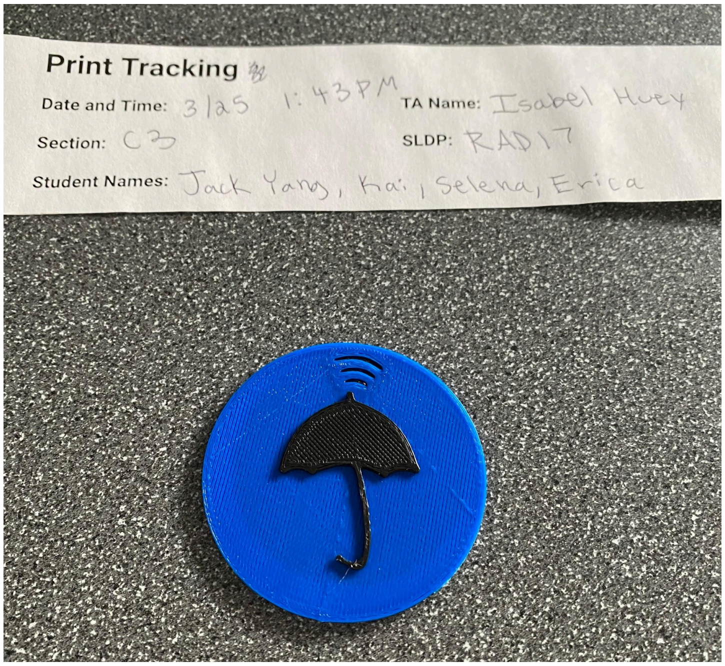

Figure 6: 3D Printed Logo

The front-end of the app for SmartBrella (Figure 7) was formulated.

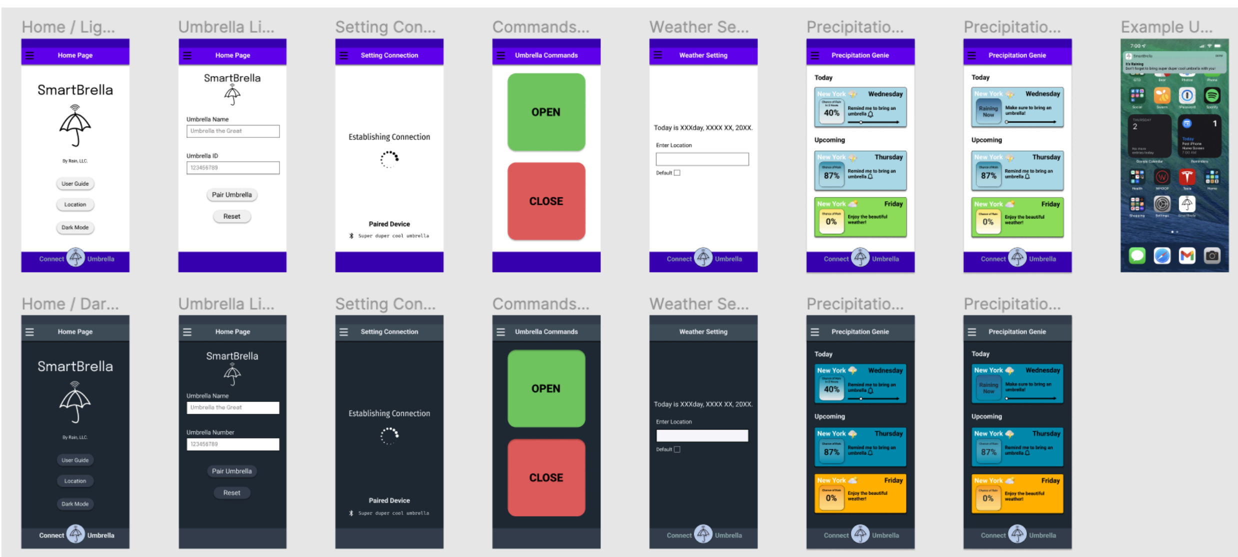

Figure 7: Light and Dark Mode for SmartBrella App

The app had a main screen, a page for closing and opening the umbrella, notifications, a user guide, and the weather.

The circuit for the motor (Figure 8) was successfully wired on Tinkercad.

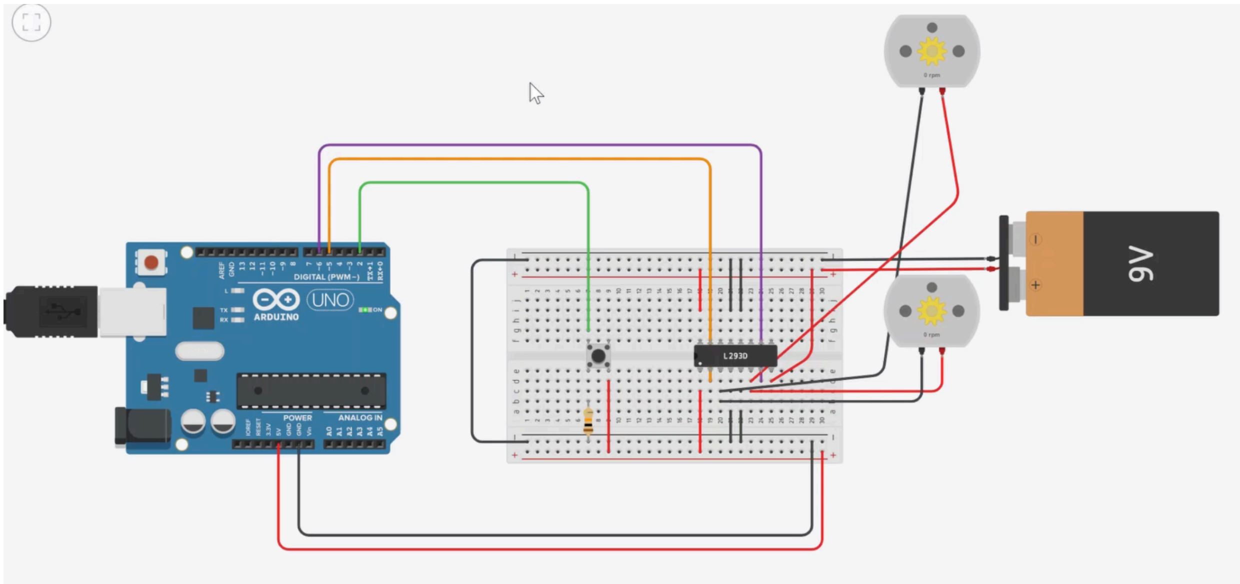

Figure 8: TinkerCAD of the Motors

5.3 Difficulties Experienced

The VEX motors used in the SmartBrella had difficulties with the force needed to wind up the string and pull down the umbrella due to the strength of the springs holding the umbrella. When the force required to pull down the string became larger than the motors could withstand, the motors would simply lock up and refuse to move further. In order to get more force out of the motors, two motors were connected with one axle in between. Difficulties were also experienced in the sound sensor, as the sensor was extremely sensitive to being touched and would react as if a loud sound was emitted near it when touched, however actual loud sounds such as claps nearby were not consistently recognized. Attempting to increase the sensitivity of the sound sensor alleviated the issue somewhat. In creating the handle, difficulties were encountered in trying to find a way to package the mechanical pieces and wires. 3D modeled or laser-cut containers were considered, but ultimately duct tape and cardboard were used to package the materials.

6. Conclusion

6.1 Results of Project

Through our team's hard work and tenacity, we have successfully prototyped, designed, and built a multifunctional smart umbrella. Our product aims to make umbrellas accessible for paraplegic individuals, amputees, or others with similar impairments like Parkinson's disease and blindness (accommodated by SmartBrella's sound recognition capabilities). The result of our innovation is an umbrella that is able to automatically open and close based on sound recognition and through a smartphone device with Bluetooth capabilities (the app component provides functionality for the user). The app component is an important aspect of our smart umbrella, as it not only reminds the user to bring an umbrella when it is raining but also enables the user to open and close the umbrella with a push of a button.

6.2 Future Improvements

However, there still are some aspects of our product that can be significantly improved and features that could potentially be added. For example, a base that could be fixed to a wheelchair arm to attach a SmartBrella could be built as an add-on to our product. Other improvements could be made to our current features; for example, we could focus on implementing an improved version of SmartBrella's sound sensor to allow for reliable voice detection and include speed adjustment settings for the opening and closing functions of the umbrella. In addition, we hope to utilize better electronic components and materials to create a higher-quality product in the future. Last but not least, reducing the bulk of our motorized base will be without a doubt beneficial to the appearance and useability of the SmartBrella.