Design Milestone 4c: Please Work's Final Report

Hoang Tran, Elena Leslie, and Jacqueline Brienza

Department of Engineering, Union College

ESC - 100: Exploring Engineering

Professor Traver

16 November 2023

Table of Contents

1. Introduction

The problem that we wanted our device to address was the lack of accessible and reliable methods of helping blind people avoid obstacles while they walk. Typically they use a cane while they walk, but this creates the issue that some obstacles in the blind person's way can still become undetected because they are in difficult to reach places.

The goal we created from this problem was to construct a device that would help blind people avoid obstacles as they walk through the use of vibrations. By using vibrations, these people might have a better understanding of their surroundings and will be able to walk without as much worry of hitting obstacles they hadn't detected before.

2. Design Requirements

This device had to have a footprint that was 30" by 30" or less, used at least one RedBoard for control, used one 9V battery per RedBoard and up to 8 AA batteries, perform all of its functions for 20 to 60 seconds, have movement that is visible from at least 10' away, and utilize a combination of audio/visual sensors and electric actuators. In addition, the device needed to be still before its operation was initiated by a single start button, and its operation must be able to be repeated with only one minute of setup in between each run. Our professor had to approve of our project idea, and the design had to be none damaging to any room it is in as well as cost less than or equal to $65 dollars to build. Our device met all of these requirements when we finished its construction.

3. Conceptual Design

3.1 Designs (Alternative Concept Combinations)

In our ABC Matrix (Table 1) we came up with several ideas for our three subfunctions which could potentially work. We then each selected some of those ideas from each subfunction and combined them to make a design. Some of the ideas for the same subfunction could overlap, such as the type of sensor and the way it would account for multiple heights, so sometimes multiple ideas would be used in one design.

	1	2	3	4	5
A. Detect objects in front at both foot and heat levels	Tall device with sensors at top and bottom	A sensor that can change the angle of scanning itself	Use ultrasonic sensors and speaker for echolocation	Use a camera and ai to identify objects	Device is a drone that moves up and down
B. Detect objects moving towards person from either side	Have lots of sensors on each side the don't rotate	One sensor on each side that can change its angle	Detect noises that moving things make	Use motion sensors	
C. Provide feedback to user that communicates location of the object	Vibrations of different lengths depending on whether object is to the side or the front	Vibrations coming from different parts of the device	Signals get stronger for closer objects	Make sounds communicating distance and location	Certain number of vibrations for different locations and distances

Table 1: ABC Matrix

The first alternate design (Figure 1) combined 2A, 2B, and 5C: sensors that changed their angles, two sensors on the side and on in front, and a different number of vibrations for different locations and distances. The design was a device that would roll in front of a person in order to detect objects ahead. The vibrations were provided by a bracelet that connected to the main device using bluetooth.

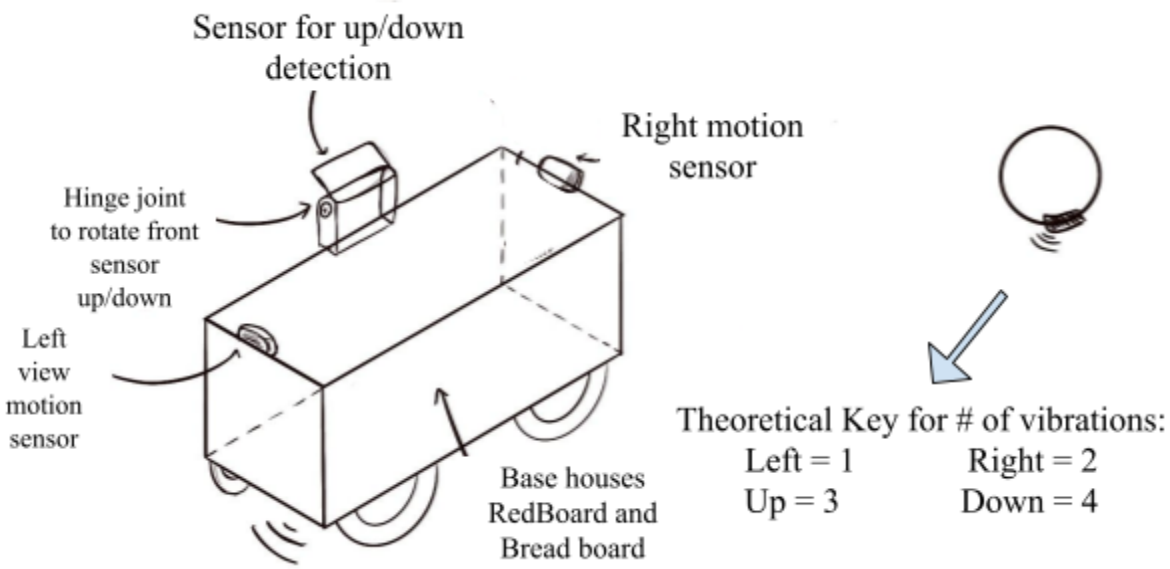

Figure 1: Alternate Design A - Jacqueline Brienza

Design B (Figure 2) combined 1A, 3A, 2B, and 2C. It was a tall structure with ultrasonic sensors at both the top and bottom, on the sides it had two sensors with changing angles, and the vibrations would come from different parts of the device to convey the direction they were coming from. It also was a device that would roll in front of the user while they walked. The user would hold a handle that vibrations would come from that was connected to the device with a wire.

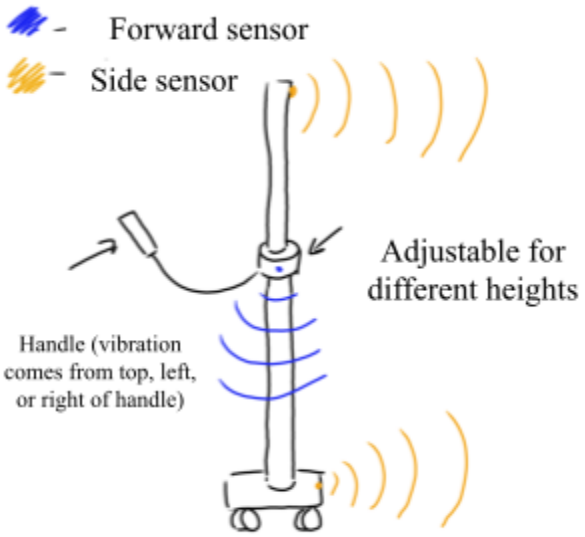

Figure 2: Alternate Design B - Elena Leslie

The third alternate design we evaluated (Figure 3) combined 2A, 2B, 2C, and 3C. The three sensors, one in front and two on the sides, all changed their angles. It used an antenna and a receiver to convey information about nearby objects through vibrations. Different parts of the receiver would vibrate depending on the direction and the vibration would grow stronger for closer objects. The device would move in front of the user with a remote control.

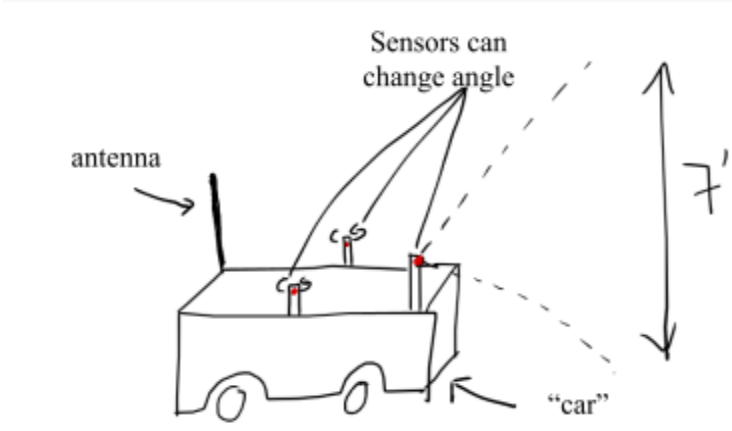

Figure 3: Alternate Design C - Hoang Tran

3.2 Evaluation of Designs (Pugh Matrix)

To compare Alternate Designs A, B, and C and to figure out which design would be best to go forward with we used a Pugh Matrix (Table 2). Functionality and durability were the most important criteria for this device because a blind person would need to be able to rely on the device so that they could gain an understanding of their environment while they walk if they were to use the device. Aesthetics were the least important to us because, while they do make devices more appealing, they do not do much to help with functionality or make sure that the device can be comfortably used. In the end, Design C got the highest score, so we concluded it was the best of our three designs. Design A got a pretty close score, but we decided it might have issues with functionality if we used the system of vibrations included within its design. the person using the device could miss one of the vibrations, which could cause confusion about which sensor detected an object, thus reducing functionality. Design B got the least number of points because we thought its tall structure might cause energy inefficiency and make it less flexible for different environments because it would likely fall over.

Criteria	Weight	A-value	A-v*w	B- value	B- v*w	C- value	C- v*w
Durability	0.2	4	0.8	2	0.4	3	0.6
Flexibility (different environments)	0.15	4	0.6	2	0.3	4	0.6
Energy Efficiency	0.1	4	0.4	3	0.3	4	0.4
Functionality: detects objects and delivers vibrations	0.35	3	1.05	4	1.4	4	1.4
Affordability	0.15	4	0.6	3	0.45	4	0.6
Aesthetics	0.05	5	0.25	2	0.1	3	0.15
Total Values	1.0	24	3.7	16	2.95	22	3.75

Table 2: Pugh Matrix to compare Alternate Designs A, B, and C

3.3 Final Selection

After using a Pugh Matrix for evaluation, we determined that a hybrid of Designs A, B, and C (Figure 4) would be the most effective device for assisting visually impaired individuals. This combined design integrates the best features of each, ensuring high functionality and durability, crucial for users navigating their environment. Despite aesthetics being a lesser priority, this integrated approach leverages the strengths of each design while mitigating individual shortcomings, resulting in a well-rounded, reliable device.

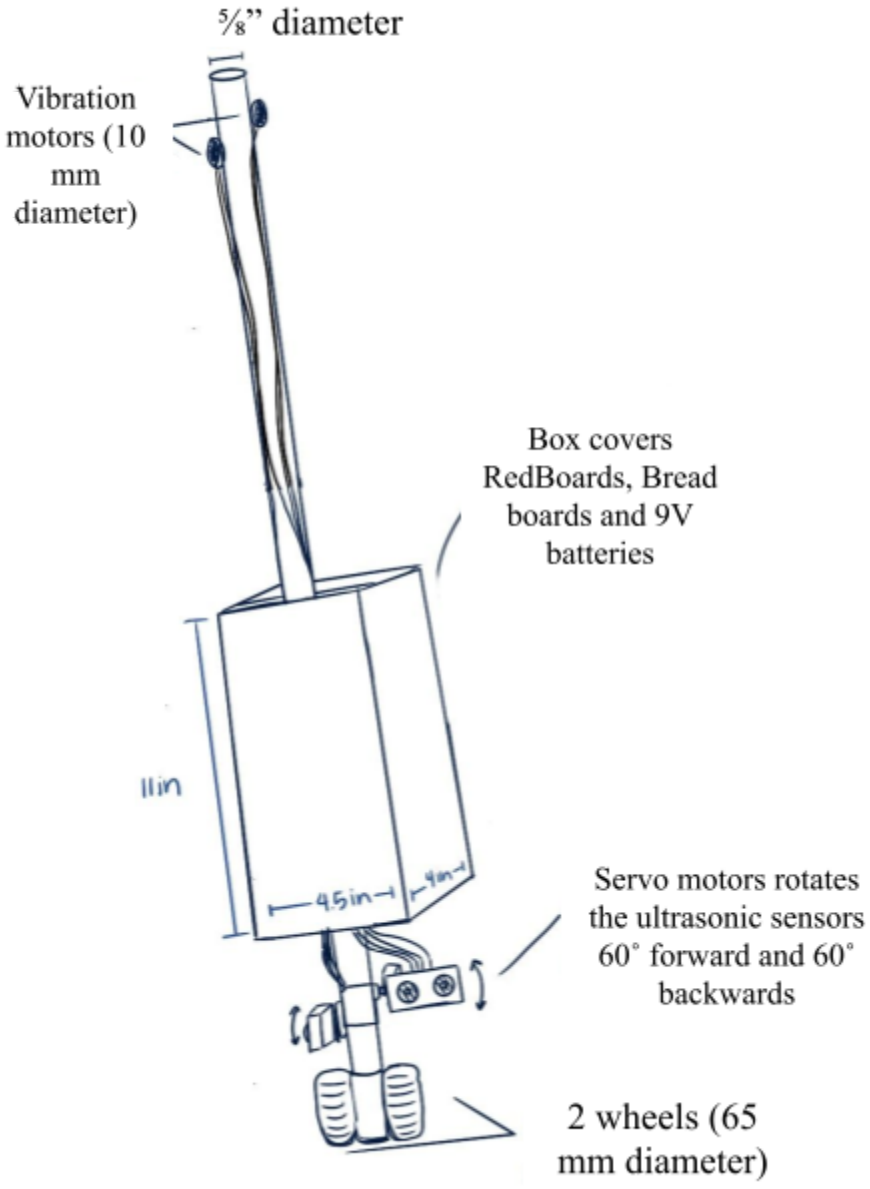

Figure 4: Final Detailed Sketch

4. Final Design

4.1 Overview of Final Design

Two different RedBoards are used in order for our design to work and are each powered by their own 9V batteries (Figure 5). The first RedBoard (the higher one) is used to control the servo motors, which the ultrasonic sensors are attached to. Once the button on the connected breadboard is pressed, each of the three servo motors are coded to turn sixty degrees forward and sixty degrees backwards continuously with a delay for 15 ms in-between the forward and backward rotations. This rotation makes it so that the attached ultrasonic sensors can detect objects that are at different angles. The second (lower) RedBoard is used to control both the vibration motors and the ultrasonic sensors. In order to have the ultrasonic sensors detect objects at about the same time, we incorporated a 200 ms delay between each of the ultrasonic sensors detection. In the code, the left ultrasonic sensor first records a distance, and if that distance is less than 30 cm, the left vibration motor vibrates. The right ultrasonic sensor then records a distance after the 200 ms delay, and if that value is less than 30 cm, the right vibration motor vibrates. The middle ultrasonic sensor is the last to record a distance, and if that value is less than 30 cm, both of the vibration motors will vibrate. The device is able to move on two wheels that are connected to a DC motor (that is not plugged in) which is used as an axle.

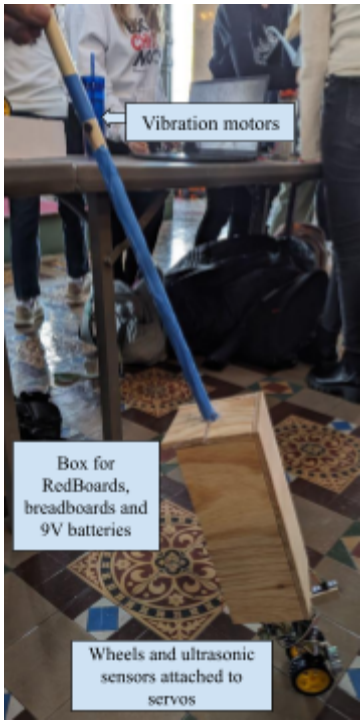

Figure 5: Final device construction

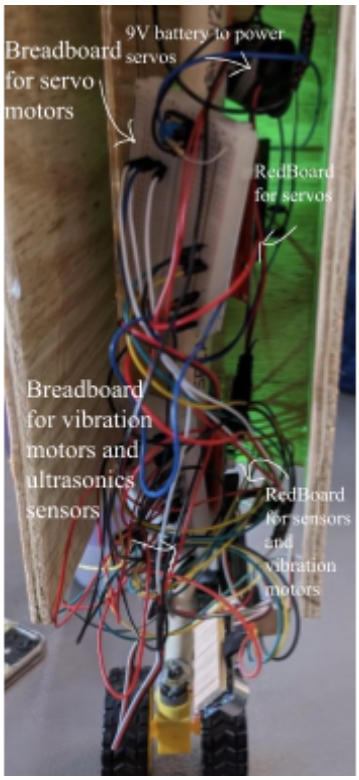

Figure 6: Breadboard, RedBoards, and 9V batteries in the box of the final constructed design

4.2 Overview of the Electronics Design

For our electronics we used two redboards. One of them controlled the sensors and vibration motors while the other controlled the servo motors. We were not able to have the code that moved the servo motors work simultaneously with the code for the sensors while in the same program, so we split them into two programs with two separate breadboards (Figure 7). Then we added another breadboard to keep the systems separate and organized. We powered each RedBoard with a 9 volt battery.

Since the servo motors all did the same thing, they could all use the same code. For simplicity, we made that code only affect one pin on the RedBoard and then wire that pin to the three servo motors. We connected pin 9 to the breadboard and then wired the signal of the servo motors into the same row so they would all be connected.

The sensors all needed two pins, one for echo and one for trigger. We wired the echos to pins 4, 7, and 11 and the triggers to pins 3, 8, and 12. The vibration motors did not need a motor driver so we could wire them directly to ground and their power directly to pins 9 and 10.

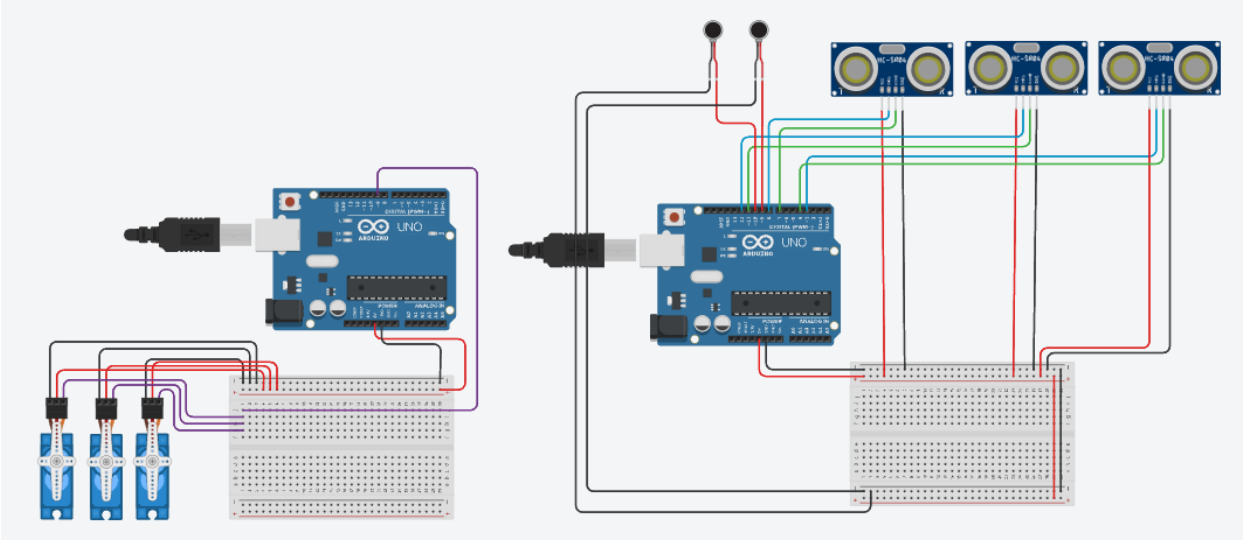

Figure 7: Electronics diagram

4.3 First Subsystem - Getting Feedback From Sensors

To detect objects within 30 cm of the person using our device, we used ultrasonic sensors. Having the ultrasonic sensors run at about the same time was a difficult task with our coding experience, so we had to research how to make three of these sensors run while being controlled by only one RedBoard (DIY GUY Chris, 2017). Through the use of 200 ms delays, we were able to get each of the ultrasonic sensors (first the left, then the right, then the middle) to record distances in the serial print view. We then included in the code that if each of the values recorded by each ultrasonic sensor was less than 30 cm, then one or both of the vibration motors would vibrate. If the left sensor detected an object, the left vibration motor would vibrate, and if the right ultrasonic sensor detected an object, the right vibration motor would vibrate. If the middle ultrasonic sensor returned a value less than 30 cm, both of the vibration motors would vibrate.

4.4 Second Subsystem - Movement (via the wheels)

For the wheels, we used a DC motor from the Sparkfun Inventor's Kit as an axle connected to the two wheels from the kit (Figure 8). The wires were disconnected so instead of being turned by the motor, the wheels would be turned by the friction of the ground as the device was pushed. The motor was attached to the back of the tube using glue so that there would be a large enough surface area that would not fall off. This system reliably allowed the device to move forward and make turns in the direction it was pushed.

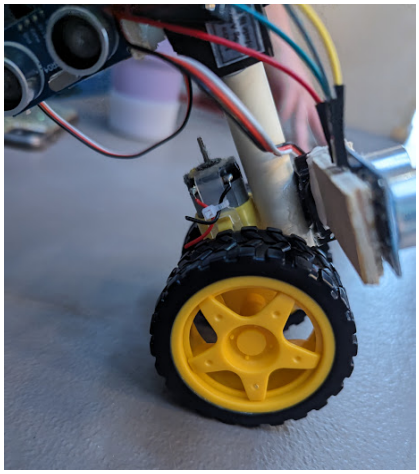

Figure 8: Wheels

4.5 Third Subsystem - Sensor Rotation

We attached the ultrasonic sensors to servo motors (via wood pieces and plastic attachment pieces found in the Sparkfun Inventor's Kit) so that they could rotate and detect objects in a larger range of angles (Figure 9). The rotation of the servo motors is initiated by the press of a button (Figure 10), then the servo motors turn sixty degrees forward. After the forward rotation, there is a fifteen second delay, which accounts for the time the servo needs to rotate forward. After this delay, the servo turns backwards sixty degrees and there is another fifteen second delay to account for that rotation. This action is then looped and can be stopped when the reset button on the RedBoard is pressed.

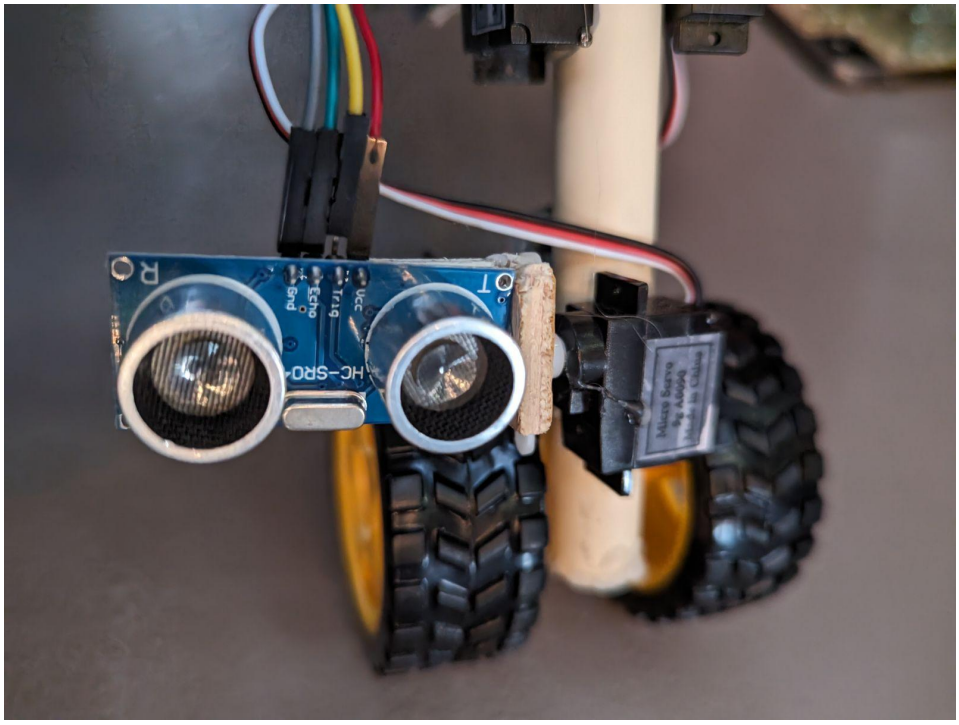

Figure 9: Front ultrasonic sensor attached to servo motor

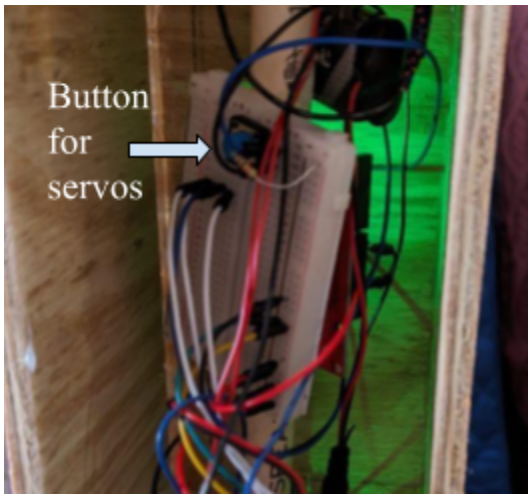

Figure 10: Button for starting the rotation of the servo motors, attached to the higher breadboard

5. Final Project Evaluation Results

The final project evaluation results for our device have yielded promising outcomes, indicating that it is performing according to expectations. The core functionalities are consistent

with our objectives, demonstrating reliability and efficiency in its operations. However, the timeline constraints have posed a significant challenge. Despite the success in the initial performance metrics, there isn't ample opportunity to enhance the device further as initially planned (i.e. isolating the vibration motors further).

6. Conclusions and Lessons Learned

One of the main things we learned throughout the project was the design process and how it could help us complete the project. It allowed us to know what we needed to work on, both by knowing the design requirements and by giving us an idea of what the next step in the project would be. Knowing the design process made it much easier to complete the project one step at a time rather than if we had tried to do it all at once.

We also learned how to work as a team on a long term project where every member of the group has different skills. Sometimes we discovered it was easier to split up the work and have each person work on one part that they knew how to do. Other times we needed to solve a problem together using all of our skills and perspectives.

Throughout the process we discovered the importance of organization of information. Our diagrams for the design needed to be clear, detailed, and easy to read so that we could look back on them later. We also needed clarity and organization in parts of the project such as the code and electrical systems so that we could more easily find problems. Without organization it would be very difficult to see what we had done and what we could do to improve it.

7. References

Arduino Forum User. (2015, August 12). *Servo Motor for 180 Degrees Rotation [Online forum post]*. Arduino Forum. Retrieved November 15, 2023, from https://forum.arduino.cc/t/servo-motor-for-180-degrees-rotation/369591/6

Cox, C. (n.d.). *How to Control Servos With the Arduino*. Circuit Basics. Retrieved November 15, 2023, from https://forum.arduino.cc/t/servo-motor-for-180-degrees-rotation/369591/6

DIY GUY Chris. (2017, April 3). *How to use two Ultrasonic sensors with Arduino*. YouTube. Retrieved November 15, 2023, from https://www.youtube.com/watch?v=Lxh7hh_gl8Q

Isaac100. (2017, August 5). *Getting Started with the HC-SR04 Ultrasonic sensor*. Arduino Project Hub. Retrieved November 16, 2023, from https://projecthub.arduino.cc/Isaac100/getting-started-with-the-hc-sr04-ultrasonic-sensor-7cabe1

Appendix A: Arduino Code

Ultrasonic Sensor & Vibration Motor Code:

```
#define echoPin 4
#define trigPin 3

#define echoPin01 7
#define trigPin01 8

#define echoPin02 11
#define trigPin02 12

long duration, duration01, duration02;
```

```
int distance, distance01, distance02;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(trigPin01, OUTPUT);
  pinMode(echoPin01, INPUT);
  pinMode(trigPin02, OUTPUT);
  pinMode(echoPin02, INPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(13, OUTPUT);

  Serial.begin(9600);
  Serial.println("Ultrasonic Sensor HC-SR04 Test");
  Serial.println("with Arduino Nano");
}

void loop() {
  digitalWrite(trigPin, LOW);
  digitalWrite(trigPin01, LOW);
  digitalWrite(trigPin02, LOW);

  delay(200);

  digitalWrite(trigPin, HIGH);
  digitalWrite(trigPin01, HIGH);
  digitalWrite(trigPin02, HIGH);

  delay(1000);

  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
  if(distance < 30) {
    digitalWrite(9, HIGH);
  }
  delay(200);
  if(distance > 30) {
    digitalWrite(9, LOW);
  }

  Serial.print("Distance:");
  Serial.print(distance);
  Serial.println(" cm");
```

```
  digitalWrite(trigPin01, LOW);
  duration01 = pulseIn(echoPin01, HIGH);
  distance01 = duration01 * 0.034 / 2;
  if(distance01 < 30) {
    digitalWrite(10, HIGH);
  }
  delay(200);
  if(distance01 > 30) {
    digitalWrite(10, LOW);
  }

  Serial.print("Distance01:");
  Serial.print(distance01);
  Serial.println(" cm");



  digitalWrite(trigPin02, LOW);
  duration02 = pulseIn(echoPin02, HIGH);
  distance02 = duration02 * 0.034 / 2;
  if(distance02 < 30) {
    digitalWrite(9, HIGH);
    digitalWrite(10, HIGH);
  }
  delay(200);
  if(distance02 > 30) {
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
  }
  Serial.print("Distance02:");
  Serial.print(distance02);
  Serial.println(" cm");
  }
}
```

Servo Motor Code:

```
#include <Servo.h>
const int TRIG_PIN = 8;
const int ECHO_PIN = 7;
int check = 0;

// Anything over 400 cm (corresponding to a 2320 us pulse) is “out of range”
const unsigned int MAX_DIST = 23200; // an unsigned int only stores positive values
```

```
Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;  // variable to store the servo position

void setup() {
  pinMode(11, OUTPUT);
  pinMode(2, OUTPUT);
  // The Trigger pin will tell the sensor to range find
  pinMode(TRIG_PIN, OUTPUT);
  digitalWrite(TRIG_PIN, LOW);

  // We'll use the serial monitor to view the sensor output
  Serial.begin(9600);

  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop() {
  if(digitalRead(A0) == LOW) {
      check++;
      Serial.println(check);
  }
  if(check % 2 == 1) {
      // the following code is servo motor

    for (pos = 0; pos <= 60; pos += 1) { // goes from 0 degrees to 60 degrees
      // in steps of 1 degree
      myservo.write(pos);              // tell servo to go to position in variable 'pos'
      delay(15);                       // waits 15ms for the servo to reach the position
    }
    for (pos = 60; pos >= 0; pos -= 1) { // goes from 60 degrees to 0 degrees
      myservo.write(pos);              // tell servo to go to position in variable 'pos'
     delay(15);                       // waits 15ms for the servo to reach the position
   }
  }
}
```