

Haptic Horizon

Sean Balbale,
Johanna Woldeher,
Crystal Kim,
Donovan Chang

12/04/2024

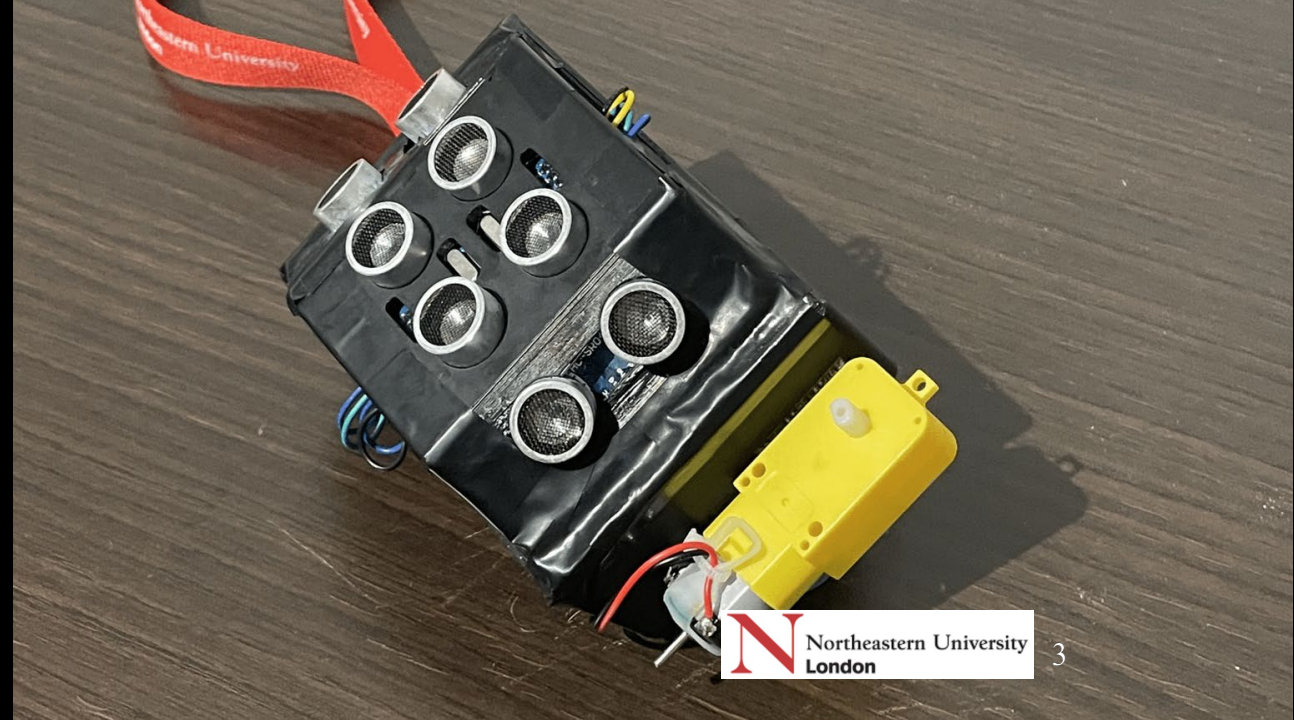
AGENDA

Background
Manufacturing
Technical
Improvements



Background

- We aim to produce a wearable haptic alarm system designed to help the visually impaired.
- The closest already existing product to ours is BuzzClip.
- We aim to improve the experience by giving the user better resolution through the usage of multiple sensors.



Technical

How does it work?

Technical Section Breakdown

Parts Used

- 4 Ultrasonic Sensors
- 1 Hobby Gearmotor

Code Explanation

- The purpose of the code is to spin the motor when an object gets within a certain distance of an ultrasonic sensor.
- The code is designed to be adjustable so it can be tailored to the individual user

Code Explanation Continued

Variables Used

- Pins 2 through 9 are used for the ultrasonic sensors.
- Pins 11 through 13 are for the motor controller.
- The final variables are for the alarm distance and for the distance measurements from the ultrasonic sensor.

```
Northeastern-GE1501-Cornerstone-of-Engineering-2 - HapticWalkingStick_V2.ino

4 // variables
5 // pins
6 // Ultrasonic sensor pins
7 static int echoPin1 = 2; // Echo pin for sensor 1
8 static int trigPin1 = 3; // Trigger pin for sensor 1
9
10 static int echoPin2 = 4; // Echo pin for sensor 2
11 static int trigPin2 = 5; // Trigger pin for sensor 2
12
13 static int echoPin3 = 6; // Echo pin for sensor 3
14 static int trigPin3 = 7; // Trigger pin for sensor 3
15
16 static int echoPin4 = 8; // Echo pin for sensor 4
17 static int trigPin4 = 9; // Trigger pin for sensor 4
18
19 // Motor driver pins
20 const int AIN1 = 13; //control pin 1 on the motor driver for the motor
21 const int AIN2 = 12; //control pin 2 on the motor driver for the motor
22 const int PWMA = 11; //speed control pin on the motor driver for the motor
23
24 // distances
25 // Variables to store the distance measured by each sensor
26 long distance1; // Distance measured by sensor 1
27 long distance2; // Distance measured by sensor 2
28 long distance3; // Distance measured by sensor 3
29 long distance4; // Distance measured by sensor 4
30
31 // Alarm distance threshold
32 static int alarmDistance = 20; // Set the alarm distance to 20 inches
```

Code Explanation Continued

getDistance Function

- This function calculates the distance an object is away from the ultrasonic sensor.
- It pulses sound and records the time it takes to bounce back. Then it divides by a constant.
- Finally it returns the distance.

```
Northeastern-GE1501-Cornerstone-of-Engineering-2 - HapticWalkingStick_V2.ino

89 //RETURNS THE DISTANCE MEASURED BY THE ULTRASONIC SENSOR
90 float getDistance(int funcTrigPin, int funcEchoPin)
91 {
92     float echoTime;           //variable to store the time it takes for
                                //a ping to bounce off an object
93     float calculatedDistance; //variable to store the distance calculat
                                //ed from the echo time
94
95     //send out an ultrasonic pulse that's 10ms long
96     digitalWrite(funcTrigPin, HIGH);
97     delayMicroseconds(10);
98     digitalWrite(funcTrigPin, LOW);
99
100    echoTime = pulseIn(funcEchoPin, HIGH); //use the pulseIn command to se
                                //e how long it takes for the
101                                           //pulse to bounce back to the senso
102    r
103    calculatedDistance = echoTime / 148.0; //calculate the distance of the obj
                                //ect that reflected the pulse (half the bounce time multiplied by the speed of
                                //sound)
104
105    return calculatedDistance; //send back the distance that was c
                                //alculated
106 }
```

Code Explanation Continued

spinMotor Function

- This function spins the motor a specified speed.
- It checks to see if the motorSpeed is positive or negative. Then it spins in the associated direction.

```
Northeastern-GE1501-Cornerstone-of-Engineering-2 - HapticWalkingStick_V2.ino

108 //SPINS THE MOTOR THE SPECIFIED SPEED AND DIRECTION
109 void spinMotor(int motorSpeed) //function for driving t
    he right motor
110 {
111     if (motorSpeed > 0) //if the motor should d
        rive forward (positive speed)
112     {
113         digitalWrite(AIN1, HIGH); //set pin 1 to high
114         digitalWrite(AIN2, LOW); //set pin 2 to low
115     }
116     else if (motorSpeed < 0) //if the motor should d
        rive backward (negative speed)
117     {
118         digitalWrite(AIN1, LOW); //set pin 1 to low
119         digitalWrite(AIN2, HIGH); //set pin 2 to high
120     }
121     else //if the motor should s
        top
122     {
123         digitalWrite(AIN1, LOW); //set pin 1 to low
124         digitalWrite(AIN2, LOW); //set pin 2 to low
125     }
126     analogWrite(PWMA, abs(motorSpeed)); //now that the motor di
        rection is set, drive it at the entered speed
127 }
```


Code Explanation Continued

setup Function

- It opens a serialport so that the program can communicate to the user.
- Then it sets input and output pins for the ultrasonic sensors.
- Finally, it sets output pins for the motor driver.

```
Northeastern-GE1501-Cornerstone-of-Engineering-2 - HapticWalkingStick_V2.ino
35 void setup()
36 {
37     Serial.begin (9600);           //set up a serial connection with the computer
38
39     pinMode(trigPin1, OUTPUT);    //the trigger pin will output pulses of electricity
40     pinMode(echoPin1, INPUT);    //the echo pin will measure the duration of pulses coming back from the distance sensor
41
42     pinMode(trigPin2, OUTPUT);    //the trigger pin will output pulses of electricity
43     pinMode(echoPin2, INPUT);    //the echo pin will measure the duration of pulses coming back from the distance sensor
44
45     pinMode(trigPin3, OUTPUT);    //the trigger pin will output pulses of electricity
46     pinMode(echoPin3, INPUT);    //the echo pin will measure the duration of pulses coming back from the distance sensor
47
48     pinMode(trigPin4, OUTPUT);    //the trigger pin will output pulses of electricity
49     pinMode(echoPin4, INPUT);    //the echo pin will measure the duration of pulses coming back from the distance sensor
50
51     pinMode(AIN1, OUTPUT);        //set the pin that controls the motor to output
52     pinMode(AIN2, OUTPUT);        //set the pin that controls the motor to output
53     pinMode(PWMA, OUTPUT);        //set the pin that controls the motor to output
54 }
```

Code Explanation Continued

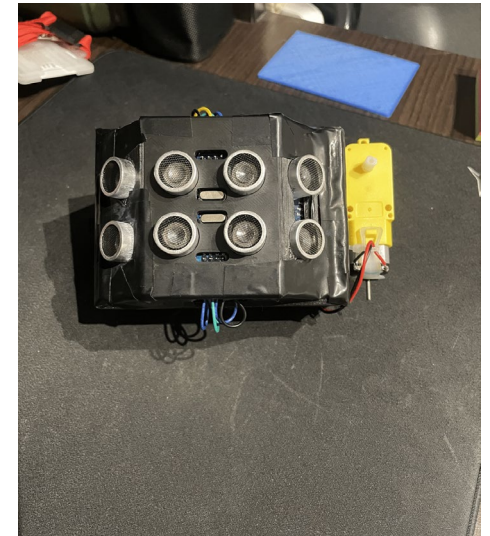
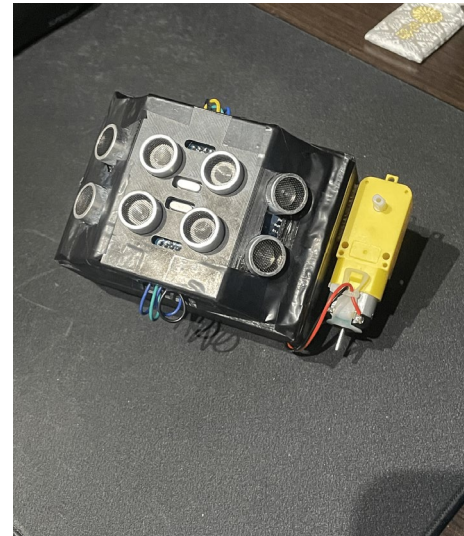
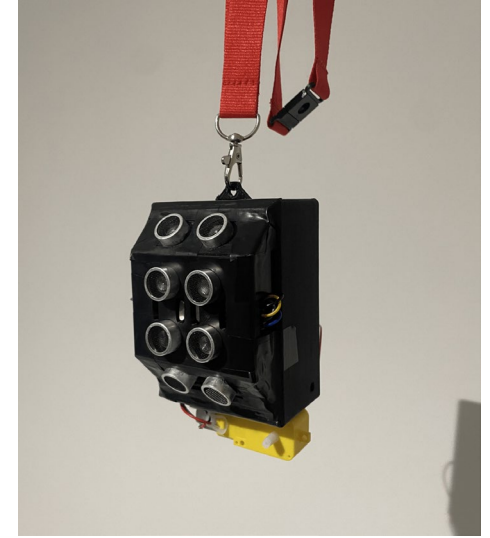
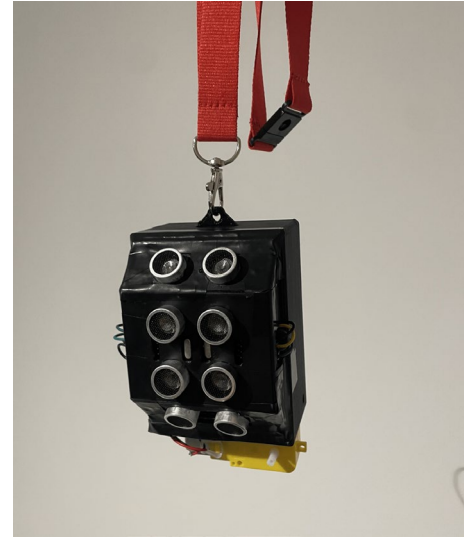
loop Function

- These are the functions that loop automatically.
- For each sensor it calls `getDistance` and then saves the distance to a variable and prints out the distance to the serial monitor.
- Then it does some logic to see if it needs to spin the motor.

```
Northeastern-GE1501-Cornerstone-of-Engineering-2 - HapticWalkingStick_V2.ino
56 void loop()
57 {
58     distance1 = getDistance(trigPin1, echoPin1); //variable to store the distance measured by the sensor
59     Serial.print("Sensor 1: ");
60     Serial.print(distance1); //print the distance that was measured
61     Serial.println(" in"); //print units after the distance
62
63     distance2 = getDistance(trigPin2, echoPin2); //variable to store the distance measured by the sensor
64     Serial.print("Sensor 2: ");
65     Serial.print(distance2); //print the distance that was measured
66     Serial.println(" in"); //print units after the distance
67
68     distance3 = getDistance(trigPin3, echoPin3); //variable to store the distance measured by the sensor
69     Serial.print("Sensor 3: ");
70     Serial.print(distance3); //print the distance that was measured
71     Serial.println(" in"); //print units after the distance
72
73     distance4 = getDistance(trigPin4, echoPin4); //variable to store the distance measured by the sensor
74     Serial.print("Sensor 4: ");
75     Serial.print(distance4); //print the distance that was measured
76     Serial.println(" in"); //print units after the distance
77
78     if (distance1 <= alarmDistance || distance2 <= alarmDistance || distance3 <= alarmDistance || distance4 <= alarmDistance) { //if the object is close
79         spinMotor(255);
80     }
81     else {
82         spinMotor(0);
83     }
84 }
```

SUMMARY

- We have created a wearable haptic device that helps the visually impaired avoid obstacles.
- The device meets these three subtasks.
 - It must detect objects in front of it
 - It must vibrate when an object is detected
 - It must be wearab





THANK YOU

Sean Balbale, Johanna Woldeher, Crystal Kim, Donovan Chang

References

- [1] Mali, Arjun. “The BuzzClip: Wearable Mobility Tool for the Blind.” Indiegogo, 2022, www.indiegogo.com/projects/the-buzzclip-wearable-mobility-tool-for-the-blind#/. Accessed 8 Apr. 2024.
- [2] “GP II Unified Listing.” Gpii.net, 2024, ul.gpii.net/content/imerciv-buzzclip-wearable-mobility-clip-blind. Accessed 8 Apr. 2024.