Sensing Instrument for Generating Haptic Touch – SIGHT

Team 30 – John Lee, Dip Patel, Jamiel Abed

ECE 445 Final Paper – Spring 2023

TA - Dushyant Singh Udawat

Abstract

Our project aims to aid visually impaired people detect objects further and higher up than a normal walking stick would. Our device uses 2 sensors, a doppler and an ultrasonic sensor, to determine how far away an object is from the user and alert the user at varying haptic intensities. This means that if a user is approaching an object (or vice versa) at a fast speed, our project will output an intense haptic buzz. On the converse, this means that our project will buzz lightly if you're approaching objects (or vice versa) at a slow speed. In the event that the user or object is stationary or is moving in the opposite direction, the device does not buzz. By providing this type of feedback, our device can help users more easily navigate their environments.

Contents

1 Introduction

1.1 Problem

Globally, there are currently around 2.2 billion people that are visually impaired [5] who may face hardships related to sensing objects near them. Common solutions available to visually impaired people are the following: a walking cane, a guide dog, or a human guide. A walking cane requires the user to constantly sweep for obstacles all while having a limited range. The problem with guide dogs and human guides is that not everyone has access to those resources. Our project introduces a device that can alleviate some of the struggles that come with being visually impaired while being easily accessible.

1.2 Solution

Our project will allow the user to receive haptic feedback to directionally warn them about obstacles that could possibly impede their movement. The way we will do this is by providing the user with a modular system that can detect objects moving closer and/or farther away and alert the user if objects are approaching the user at an abnormally fast speed or if objects are within a certain distance. Using a complementary filter, we will take a sensor fusion approach to be able to detect a more accurate distance signal from combining the data coming from our ultrasonic and doppler modules. Additionally, once we have the distance, we will alert the user via haptic feedback by using tiny pager motors.

1.3 Visual Aid

Figure 1: SIGHT being utilized to detect and warn user of an incoming person

1.4 High-Level Requirements

- The ultrasonic and doppler sensors must, in conjunction, produce a signal that can reliably measure the distance from the user to the nearby object. The distance we will need to reliably measure is anything between 1/2 – 2 meters. Reliability is determined by a <2% error in real life distance versus measured distance.
- The haptic pad must work in conjunction with the filtered sensor data and be able to turn on only if the object is detected to be moving closer. It must be able to gradually adjust intensity (from 0 – 100%), scaling linearly with distance to the object.
- The processing must be able to discriminate objects that are stationary relative to you as well objects moving away and towards you by using the distance data that was produced by the complementary filter.

2 Design

2.1 Block Diagram

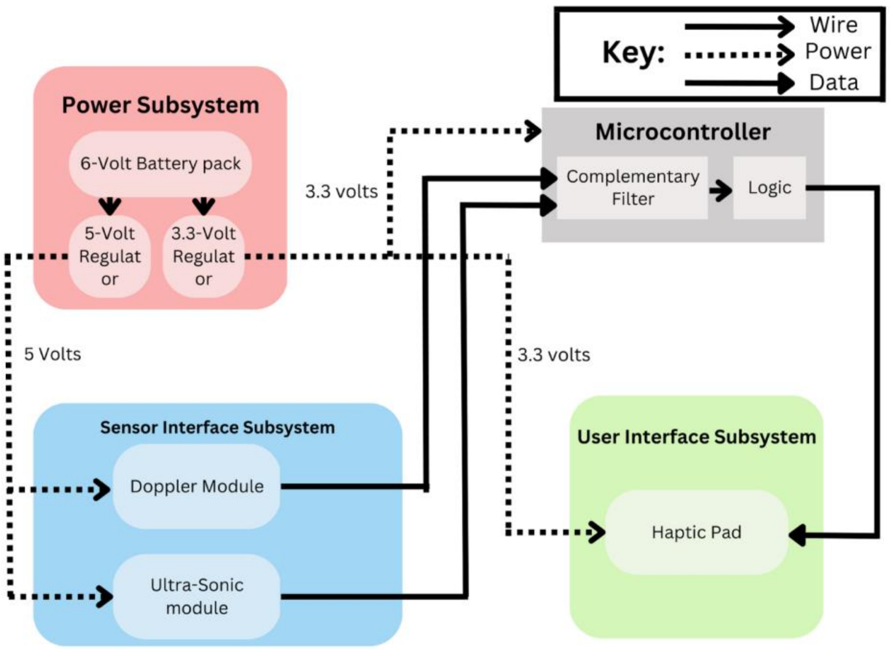

Figure 2: SIGHT Block Diagram

Figure 2 shows the block diagram for the system. The microcontroller takes in sensor data from the sensor interface subsystem to process them and decides whether to trigger the haptic pads in the user interface subsystem. The power subsystem is responsible for providing adequate power to all components in our system.

2.2 Physical Design

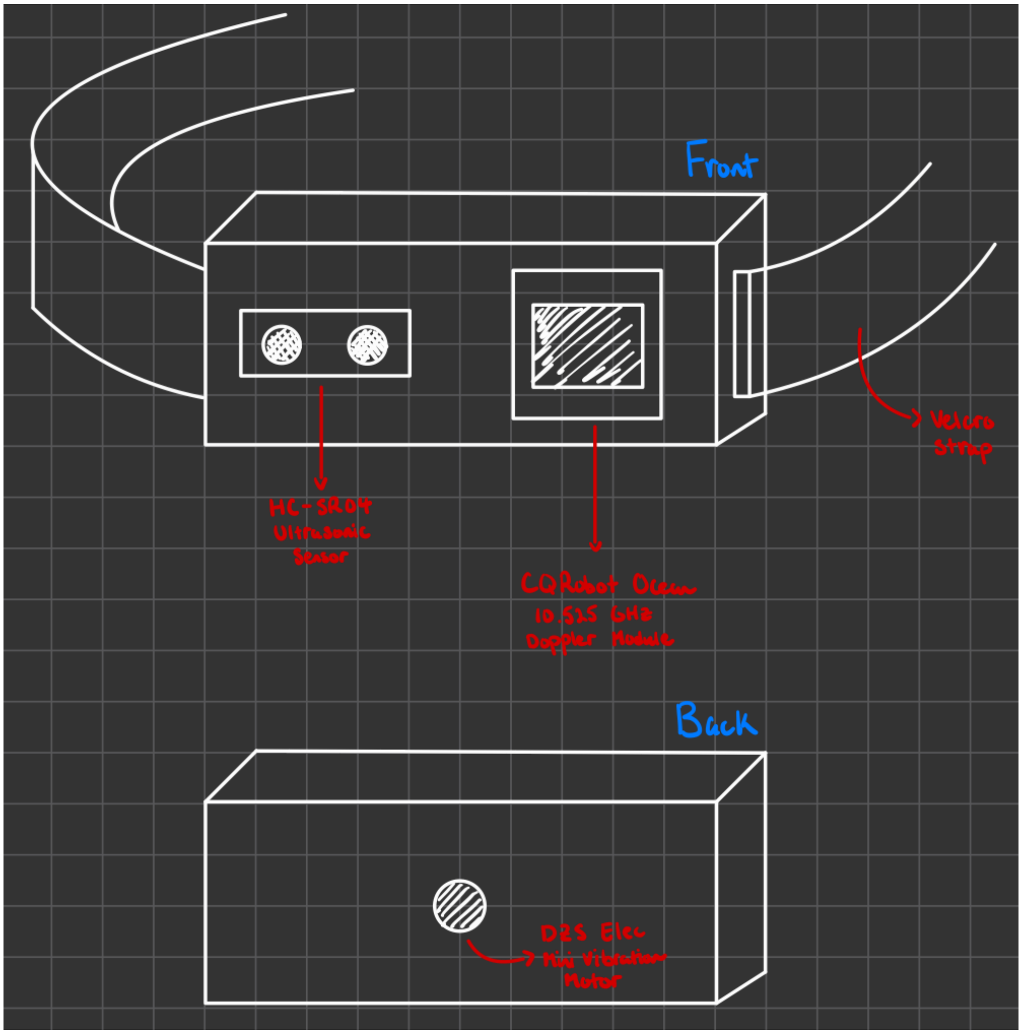

Figure 3: Physical design for SIGHT

2.3 Block Descriptions

2.3.1 Power Subsystem

The power subsystem consists of a 12-volt battery pack and two voltage regulators, a 5-volt and 3.3 voltage regulators. This subsystem is capable of providing power to various modules in other subsystems.

The LAMPVPATH 12-volt battery pack serves as the input voltage into our voltage regulators, therefore, making it our primary source of power.

The BD4 5-volt regulator provides power to the doppler module and the ultrasonic module in the sensor interface subsystem.

The TPS7 3.3-volt regulator provides power to the haptic pads in the user interface subsystem.

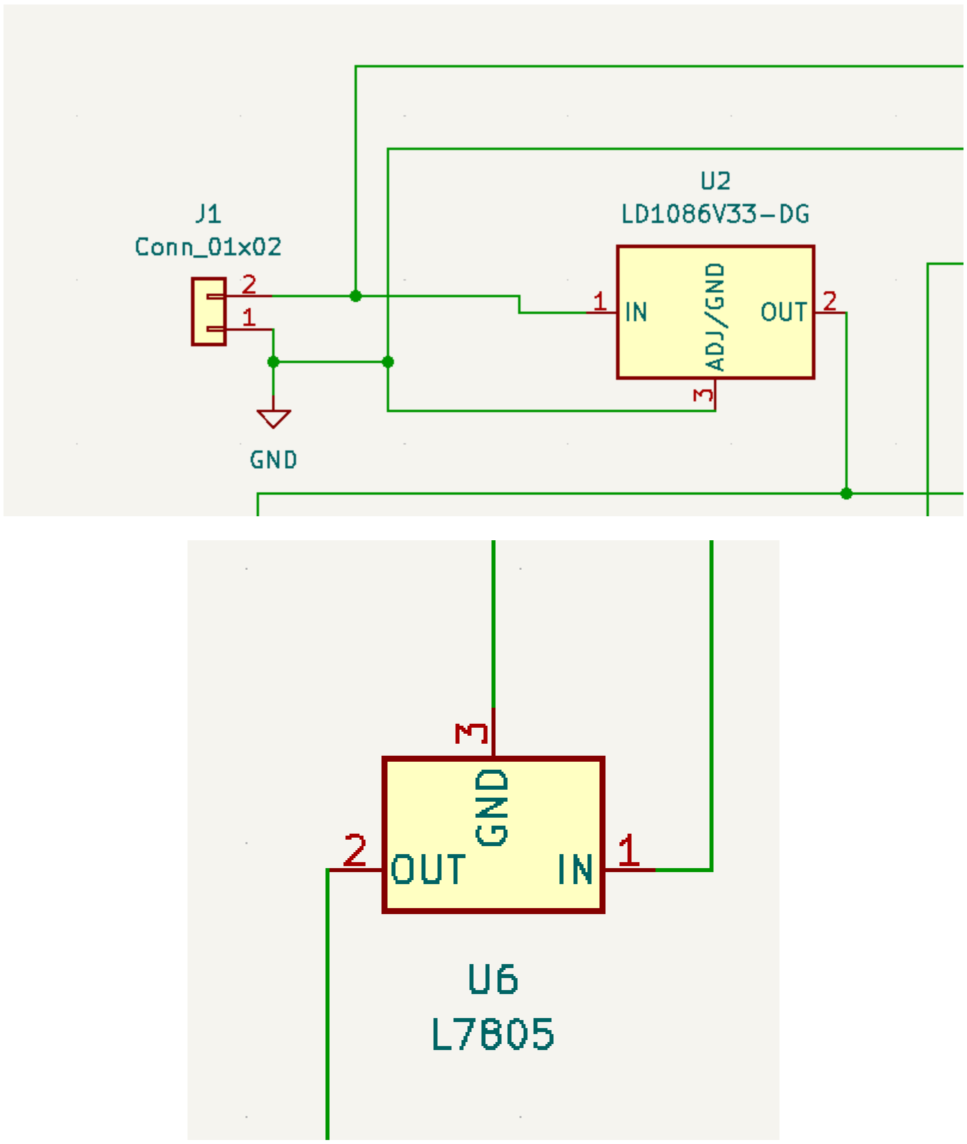

Figure 4: Schematic for voltage regulators and battery pack connections

Additionally, care was taken to include a voltage divider for the GPIO pin on the microcontroller as the line coming from the ultrasonic was 5 V while the STM32F303K8's GPIO pins were rated for 3.3 V [4].

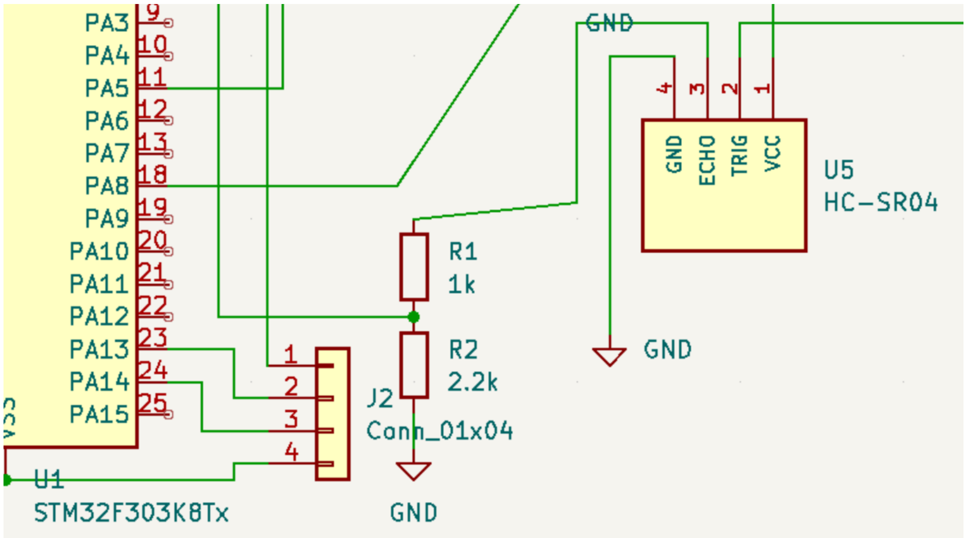

Figure 5: Voltage divider between the GPIO pin and the HC-SR04

The power subsystem requirements were verified with the procedures described in Appendix C.

2.3.2 Sensor Interface Subsystem

The sensor interface subsystem consists of two sensors, the doppler radar module and the ultrasonic sensor. This subsystem is responsible for collecting and transferring distance and speed data to the microcontroller for further processing.

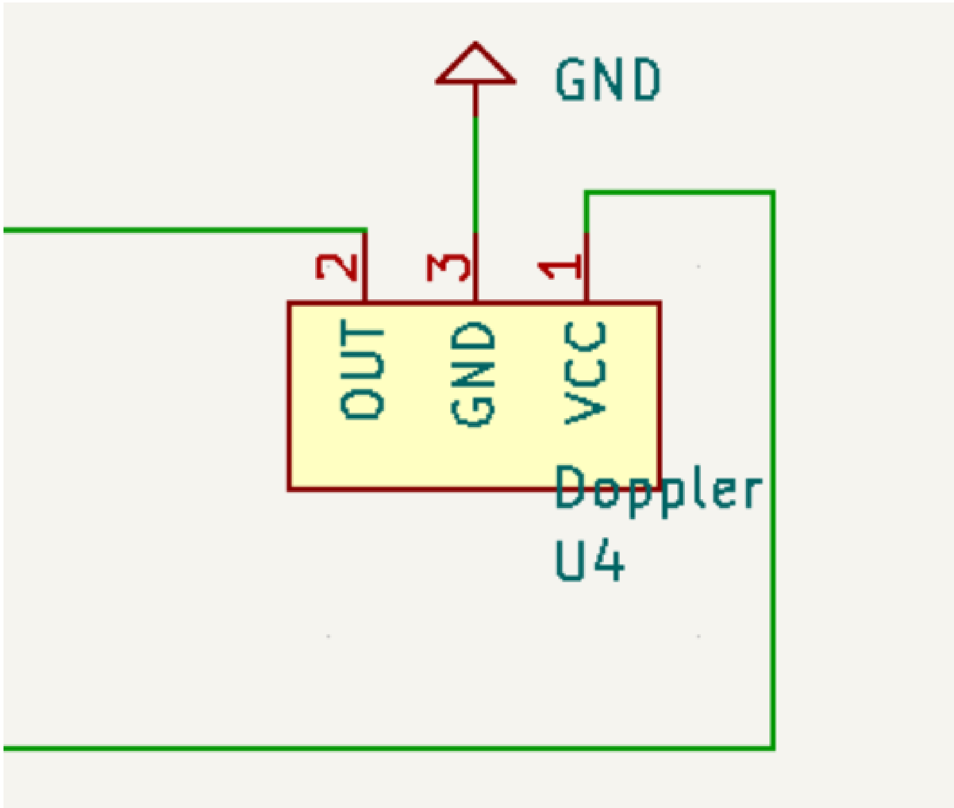

Figure 6: Schematic for the doppler module

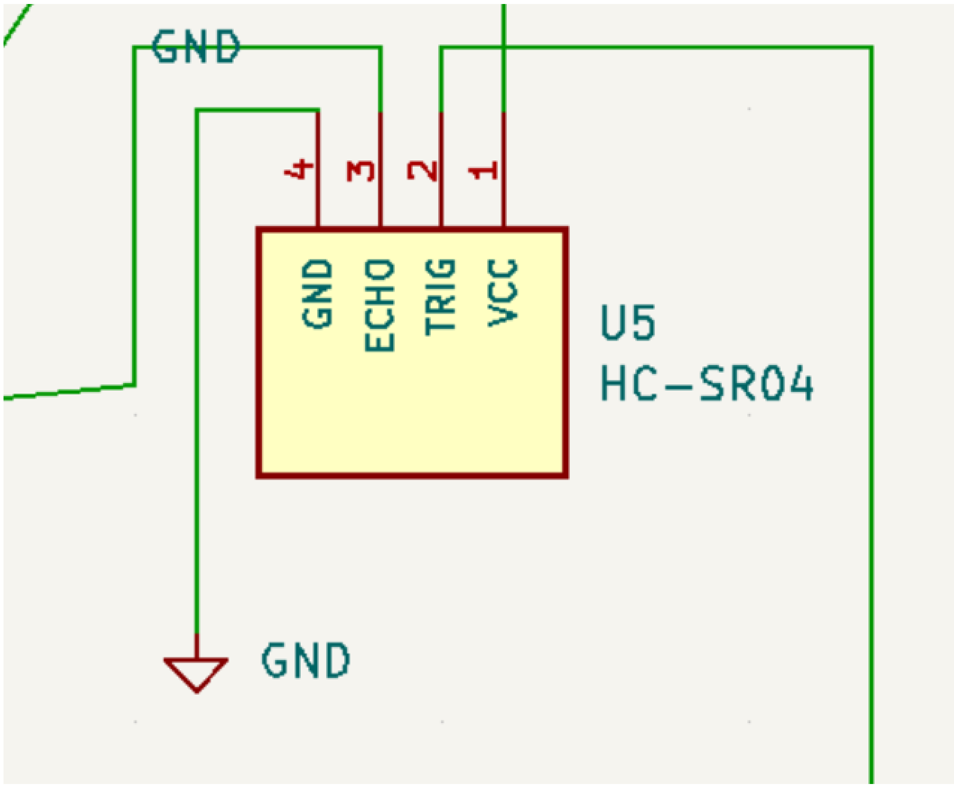

Figure 7: Schematic for the ultrasonic module

The HB100 doppler module is a CW-monostatic microwave transceiver which transmits and receives frequencies at 10.525 GHz in the X-Band. The module relies on the doppler effect to be able to detect objects moving relative to it. The module uses two patch antennas each, for both the TX and RX which have a beam pattern that is directional with a wide main lobe that has an HPBW of around 80 degrees in the azimuth direction and 40 degrees in the elevation direction.

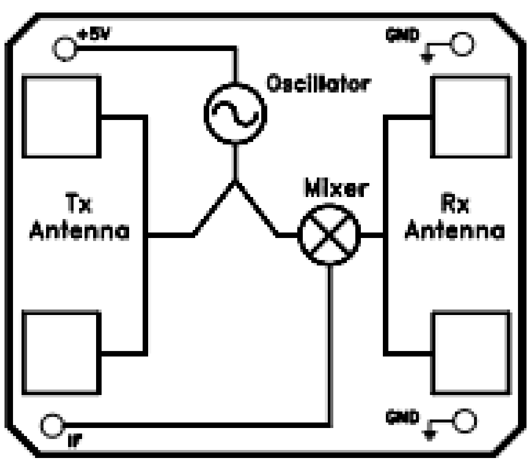

Figure 8: Doppler Block Diagram and Connections [2]

The HC-SR04 ultrasonic distance sensor can give distance data from 2 cm to 400 cm with a ranging accuracy up to 3 mm. It utilizes a pulse train of eight 40 kHz ultrasonic waves and detects if a pulse signal comes back. If the module detects a return signal the time it takes for the signal to return is the time of flight which then we can use to get distance.

Overall, this sensor subsystem takes in data produced by both the doppler and ultrasonic sensors and provides them to the microcontroller for further processing.

The sensor subsystem requirements were verified with the procedures described in Appendix C.

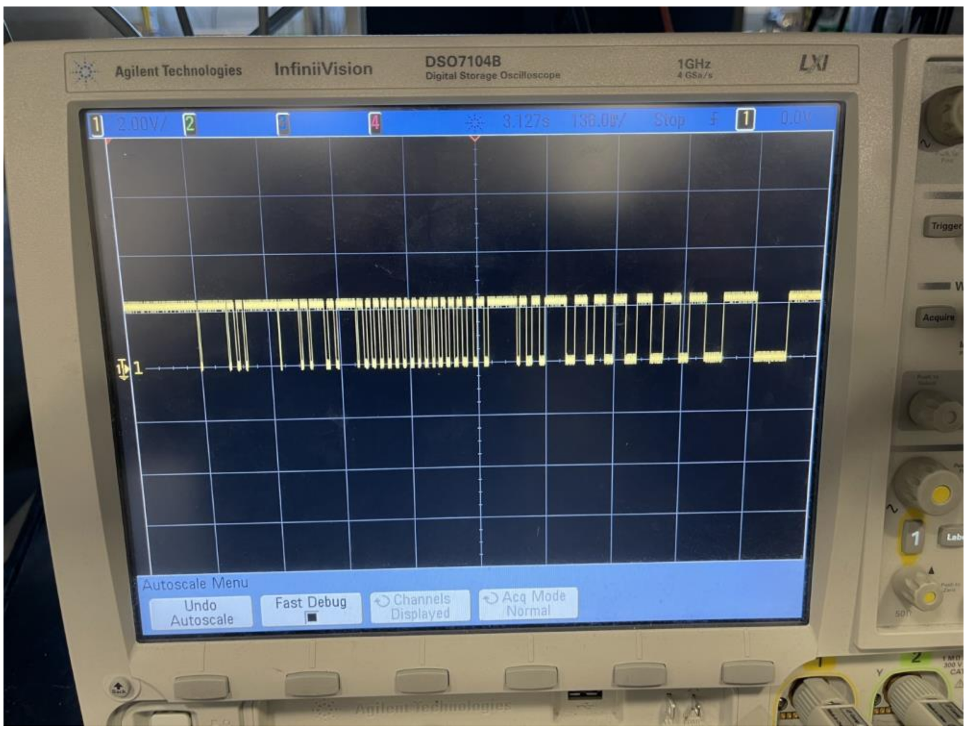

Figure 9: Oscilloscope output for the doppler walking test

2.3.3 Microcontroller Subsystem

For the microcontroller we chose to use the STM32F3 series, given their powerful mix of computational power as well as good connectivity features [4]. The software consists of a complementary filter and logic which will determine based on the filtered distance signal whether to activate the haptic pad.

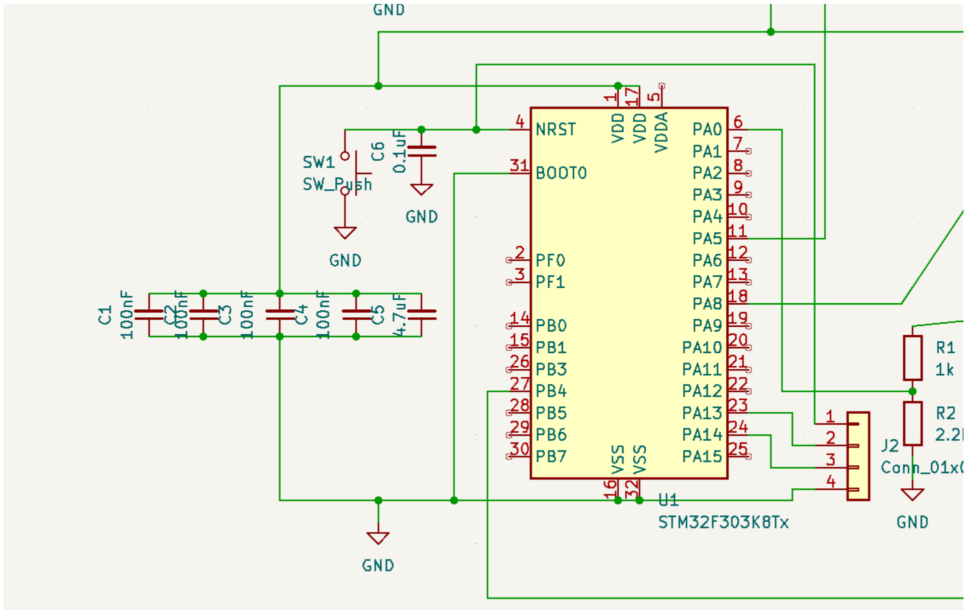

Figure 10: Schematic for the STM32F303K8 microcontroller

The complementary filter takes the preprocessed ultrasonic and doppler signals from the sensor module to create a more consistent and reliable signal. This is done by passing the doppler signal through a low pass filter and adjacently passing the ultrasonic signal first through a discrete integrator then a high pass filter. Adding the resulting two signals together will create a much more error prone signal.

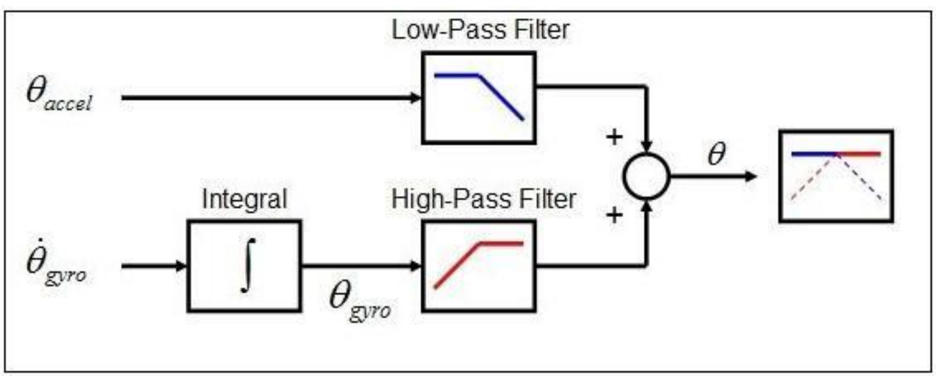

Figure 11: Complementary Filter [6]

Using the improved signal created by the complementary filter, we used some logic to determine when the haptic pads should be active. In our case, we would like the haptic pads to only be active if an object is relatively approaching the user. Therefore, every instruction cycle in the microcontroller we compare the last average distance with the distance produced by the complementary filter to determine if an object is moving away, towards, or stationary relative to us. Using this information, we then only triggered the haptic pad if we determined that the object was to be moving away from us at a large enough delta.

We opted to use a linear scale to determine how the haptic intensities should be sent according to how much of a distance delta there was between samples. We experimented with an exponential scale as well but found that a linear scale was the simplest and best option for this project. The equation used was,

$$Intensity = 35\left(\frac{I_{max}\Delta D}{D_{previous}}\right)$$

Where $\Delta D = |D_{current} - D_{previous}|$ is the change in distance calculated for each instruction cycle. And $I_{max} = 100$ was the maximum intensity percentage which corresponded to the duty cycle of the pulse width modulated (PWM) signal that controlled the haptic pad.

The microcontroller subsystem requirements were verified with the procedures described in Appendix C.

2.3.4 User Interface Subsystem

The user interface subsystem consists of the haptic pad that is used to alert the user if an object is approaching.

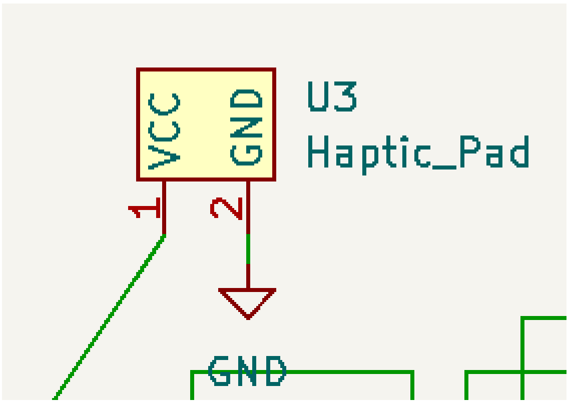

Figure 12: Schematic for the haptic pad

The haptic pad is a coin sized, flat coreless vibration motor that produces a vibration sensation when 3.0 volts is applied to the leads.

The user interface subsystem requirements were verified with the procedures described in Appendix C.

3 Design Verification

The verification process for our project consisted of two main things. The first being filtering. We needed to find a way to reduce the noise caused by the raw ultrasonic sensor data. The second was the intensity of the haptic feedback. We needed the haptic feedback to vary based on the intensity of the change in distance.

3.1 Filtering

3.1.1 Choosing a Filter

To test filtering, we produced two types of filters. Both of which utilized a sliding window of data values. The first type of filter was an average filter. This filter took all the values in the window and averaged them. The algorithm is shown below:

```
// Calculate average of current window and return it
float avg = 0.0;
for (size_t i=0; i<filter->size; i++)
{
    avg += filter->window[i];
}
avg /= filter->size;

return avg;
```

Figure 13: Algorithm for average filter

The second type of filter was a median filter. This filter chose the median value amongst the window of values. The algorithm for this filter is shown below:

```
// Calculate median of current window and return it
float tmp_window[filter->size];

for (size_t i=0; i<filter->size; i++)
{
    tmp_window[i] = filter->window[i];
}

qsort(tmp_window, filter->size, sizeof(*tmp_window), qsort_compare);

return tmp_window[filter->size/2];
```

Figure 14: Algorithm for median filter

For our final product, we decided to go with the median filter because in practice, it produced less noise than the average filter. This is due to the average filter’s tendency to spike more drastically than the median filter.

An example of distance data we received before and after applying the median filter is shown below:

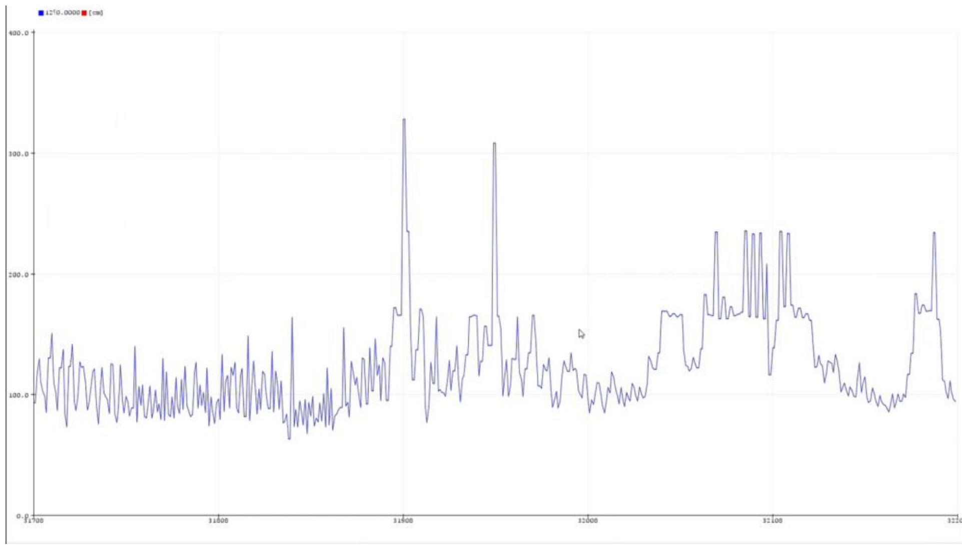

Figure 15: Ultrasonic distance data without median filter

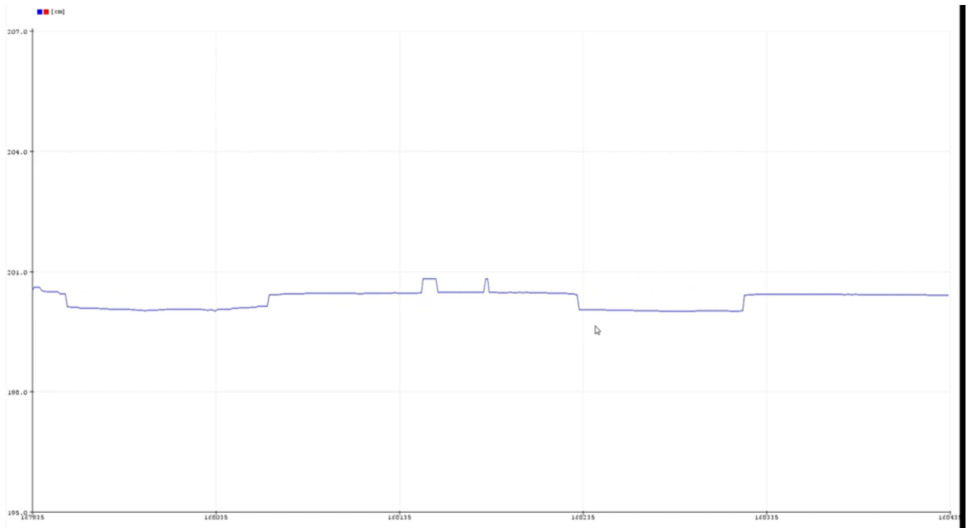

Figure 16: Ultrasonic distance data with median filter

Based on the data, there is a drastic difference in noise when the median filter is applied. Although the average filter also produced significantly less noise, through our testing, we concluded that the median filter was producing overall better results than the average filter.

3.1.2 Standard Walking Test

After deciding the optimal filter to minimize noise, we developed a standardized test to use for all our measurements and accuracies. We set up our test by marking 2 meters away from a wall with 50 cm intervals in between. For the actual test, we walk towards the wall, stopping at around the 50 cm mark and walking back to the 2-meter mark. This test was used to test how accurately our sensors were picking up distances.

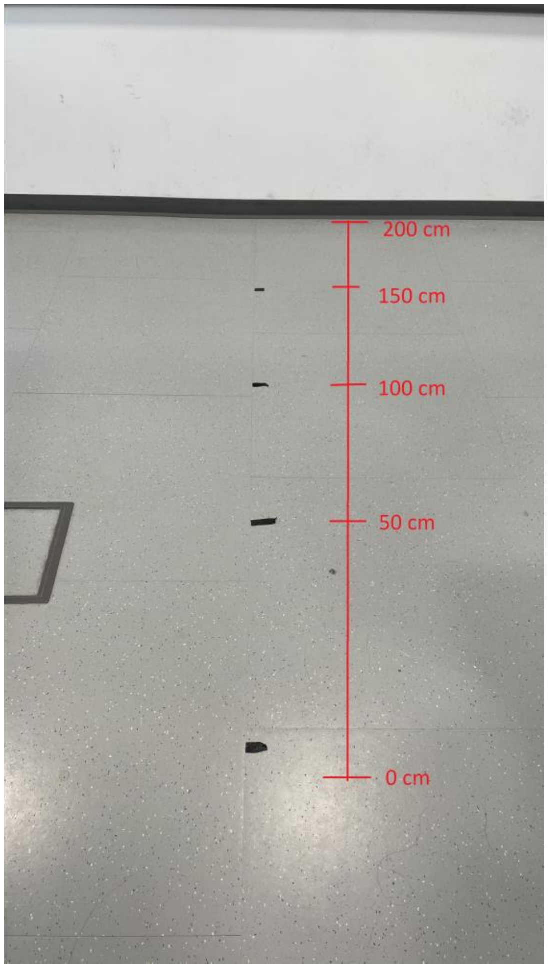

Figure 17: Standard test setup

Using this testing setup, we produced these results for our distances:

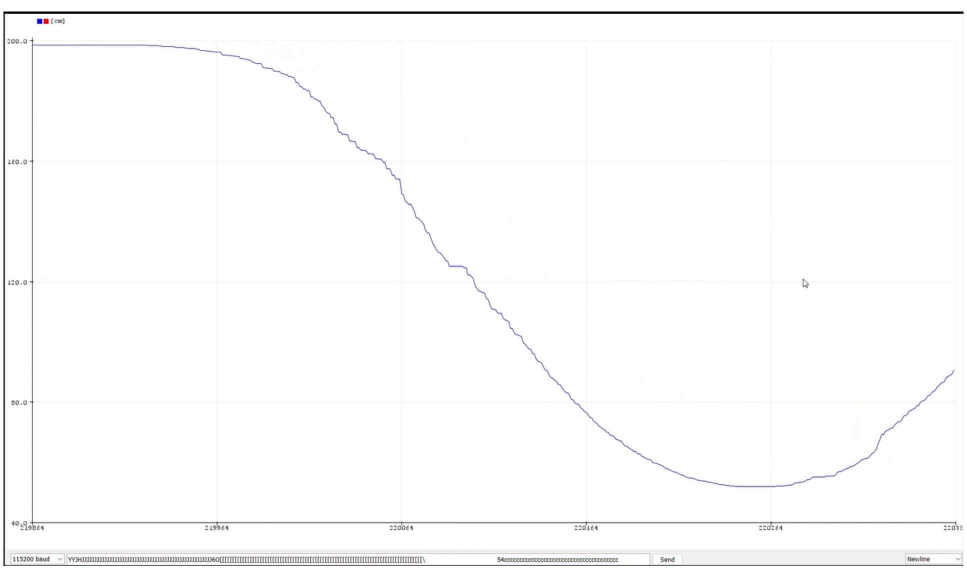

Figure 18: Distance plot using standard test

The average distance values we recorded while standing at the 200 cm (2 meter) mark was 199.1 cm. To maintain within 2% error, our results must have been between 196 – 204 cm. Since 199.1 cm is within this range, we stayed true to our intended accuracy.

3.2 Haptic Intensity

A vital element of our project was for our device to be able to produce haptic feedback that would scale based on how quickly something was approaching relative to the user. The algorithm that we used to calculate the intensities based on the distances is show below:

```
if (avg < *last_distance - .5)
{
    intensity = (uint32_t) ((*last_distance - avg) * (MAX_INTENSITY/ *last_distance) * 35);
    if ((*last_distance) > 75 ){
        intensity *= 1.5;
    }


    comp_struct.distance_sign = -10.0;

}
```

Figure 19: Intensity algorithm

The intensity function calculates the intensity value based on the changes in distance between the current and previous distance values. If the current distance is less than the previous one (object is relatively approaching), it calculates the intensity.

To make sure that this part of our project functioned as intended, we used the same standard test mentioned above. However, in this case, we expect higher intensities the closer the sensors get to the wall.

```
0
0
6
7
17
32
51
72
86
95
51
100
100
100
100
100
91
53
26
0
0
0
```

Figure 20: Intensity values using standard test (serial port data)

At the start the intensity is zero, meaning that there was no haptic feedback since the user is not moving. Then, as the user walks, the intensity ramps up to 100 and once the user is no longer getting closer to the wall, or the user starts walking backwards, the intensity goes back to zero.

3.3 Requirements and Verifications

Overall, we fully met all the requirements that were laid out in our requirements and verification tables (see Appendix C for more details). The only major point to note would be that our vision for the doppler module was that we would use the ultrasonic and doppler signals equally to create a new signal via the complementary filter. In our final product however, we ended up using an 85/15 split where 85% of the signal came from the ultrasonic sensor and 15% came from the doppler module. This action was necessary to maintain the 2% error that we had promised. Increasing the percentage for the doppler module would cause too much noise and would break out of that 2% error window.

4 Cost

4.1 Cost Breakdown

In our team we have

Cost of Labor:

$$\$35/\text{hour} \times 2.5 \times 10\ \text{hours/week} \times 10\ \text{weeks} \times 3 = \$26{,}250$$

5 Conclusion

We were able to successfully complete our project as promised, meeting all of our high-level requirements. We were able to achieve a <2% error on distance measurements, our haptic pad gave outputs from 0-100%, and our device was able to distinguish stationary from non-stationary objects. During our final demonstration, we showed how to wear our device, how it interacts in a real-life environment, and how it produces the outputs. We then showed how our device reacts to unorthodox obstacles approaching from different angles.

5.1 Accomplishments

Our process of development was very methodical in that we started with the theory before embarking on physical breadboarding, software development, and schematic design. This theoretical foundation gave us success in all the proceeding stages of project building. We were also able to customize an enclosure to give a realistic feel to our project and allow us to be able to test it in the real world.

5.2 Uncertainties

The bulk of our uncertainties lie in how our project will perform to scale. Although we were able to meet our high-level requirements, these requirements don't ensure that our project will reliably work in a busy, noisy environment. What this means is that since our project uses relatively cheap sensors, it has a hard time blocking out noise and sometimes produces false positives or negatives when detecting objects. So, in a real-life environment our project will unnecessarily buzz because of all the various detected noise.

5.3 Ethics and Safety

Our group acted in accordance with the IEEE code of ethics [insert footnote]. We recognize that breaching this code of ethics would be an ethics infringement. Because of this, we held ourselves to the highest ethical standards in our teamwork. Such standards include:

1. Protecting the privacy of others [1]
2. Being open to honest criticism of our technical work, to acknowledge and correct errors, and to be as honest as possible when making claims [1]
3. Avoiding unlawful conduct in professional activities [1]

Refer to Appendix A for full details of Ethics and Safety

5.4 Future Work

In the future we'd like to be able to get better hardware to ensure a higher quality sensor and more accurate results. Also, we'd re-think our sensing algorithm because the complementary filter makes it difficult to work with the doppler sensor.

The point of focus would be to extend the detection range to beyond 2 meters (with high accuracy). This will help us be able to work in more noisy environments and not always having to be in a controlled environment to produce the best results. Essentially, we want to be able to test our project on a large scale. We’d want this project to be officially approved by the NIH and the FDA so that it can be deemed safe for visually impaired people to use.

Appendix A: Ethics and Safety

The main concern associated with our project is the intention and real-life usage of our project. Although the intention of our project is to aid visually impaired people, we have not (and most likely will not) tested our device to meet the safety standard required to legally aid visually impaired people [1]. Our project is a recreational solution out of our own curiosity. This will give us a more holistic understanding of visual impairment in the hopes that one day we can provide society with a better understanding of emerging technologies like our project [1].

In section I.1 of the IEEE Code of Ethics, it also mentions the protection of privacy of others. Although our project will be used to detect the position of objects around the user, the geographical location is not tracked and thus the location of the user will always be anonymous. This way, we are doing our best to uphold the ethical standards laid out by IEEE in all respects, but especially privacy.

In order to put out the best quality project that we can, we as a group have committed to being open to honest criticism of our technical work, to acknowledge and correct errors, and to be as honest as possible when making claims [1].

In terms of possible safety hazards within the project itself, battery usage and touch sensitivity to the haptic pads are the only areas for potential concern. In terms of power, we power our system with a 12-volt battery pack and use two regulators to adjust the volage to 5V and 3.6V respectively. We've ensured that the batteries we're using aren't lithium-ion, so it'll be safe to stay near human skin for extended periods of time. Furthermore, none of the features or modules of our project will be exposed to the user's skin as the entire system will be encapsulated in a pouch. If someone is extremely sensitive to haptic vibrations, it is best to stay away from using our product. However, the haptic vibrations aren't extremely strong and don't pose a legitimate physical threat to the user.

Appendix B: Parts & Costs

Description	Part Number	Quantity	Cost
STM32F303K8 Development Board	497-12848-ND	1	$10.99
Ultrasonic Sensor HC-SR04	347-45418-ND	1	$4.50
Haptic Pads (Mini Vibration Motors)	694-14818-ND	1	$4.99
6V Battery Pack	WM4151-ND	1	$7.49
Doppler Radar	341-47818-ND	1	$18.99
4.7 uF Ceramic Capacitor	1276-1045-2-ND - Tape & Reel (TR)	5	$0.09

0.1 uF Ceramic Capacitor	1276-1043-2-ND - Tape & Reel (TR)	6	$0.09
1 kΩ Resistor	311-1KMTR-ND - Tape & Reel (TR)	4	$0.10
2.2 kΩ Resistor	311-2.20KHRTR-ND - Tape & Reel (TR)	4	$0.10
4-Pin Male Header Connectors	WM4202-ND	4	$0.32
3-Pin Male Header Connectors	WM4201-ND	2	$0.25
2-Pin Male Header Connectors	900-0022232021-ND	4	$0.19
STM32F3 Microcontroller	497-15828-ND	2	$6.17
3.3V Voltage Regulator	497-17820-ND	2	$1.54
5V Voltage Regulator	497-1443-5-ND	3	$0.62
STLINK V3 Mods	497-STLINK-V3MODS-ND	1	$8.78
STLINK V2 USB	3647-ST-LinkV2USB-ND	1	$8.43
Total Cost			$85.78

Appendix C: Requirements and Verifications Tables

Table 1: Power Subsystem – Requirements & Verification Pt. 1

Requirements	Verifications
1. The 5-volt regulator must sufficiently power both the doppler and ultrasonic modules 2. The 12-volt battery pack must be able to sufficiently supply power to both 5-volt and 3.3-volt regulators 3. The 3.3-volt regulator must sufficiently power all the haptic pads	1. Using the Oscilloscope, measure the output voltage of the regulator. (Must be between 5V +/- 2%) 2. Using the Oscilloscope, measure the output voltage of the battery pack. (Must be between 12V +/- 2%) 3. Using the Oscilloscope, measure the output voltage of the regulator. (Must be between 3.3V +/- 2%)

Table 2: Sensor Interface Subsystem – Requirements & Verification Pt. 2

Requirements	Verifications
1. The doppler module must be able to detect objects moving relative to it and	1. Using the Oscilloscope, we set up the doppler module to be still and proceed

be able to send the doppler shift signal for further preprocessing. It must be able to capture speeds between 3 km/hr and 14 km/hr. 2. The ultrasonic module must be able to detect moving objects in front of it and relay the distance information to the microcontroller. The distance captured must be between ½ and 2 meters	to walk/run at it to measure the frequencies. This allows us to compare the theoretical frequency with the actual frequency of a human moving towards the doppler module at speeds between 3- 14 km/hr. The actual frequency should be within 20% of the theoretical. See figure 5 for frequency test on a walking person. 2. Using the Oscilloscope, we can set up the ultrasonic sensors to test how far away certain objects are from it based on the signal. The closer we move the object the smaller the square wave should become.

Table 3: Microcontroller Subsystem – Requirements & Verification Pt. 3

Requirements	Verifications
1. The complementary filter should successfully create a new signal from the two given signals (ultrasonic and doppler). 2. The filtered signal must be more accurate (>2% accurate) than each of the original signals. 3. The logic we use must allow the haptic pads to only activate under the specific circumstances that we predetermine (approaching objects only). It must be able to adjust intensity (from 0 – 100%) based on the incoming signal.	1. In software, we can analyze the data and verify that the distance data received correlates with the objects that are nearby. (If objects get closer, the distances should decrease etc.). 2. Using the oscilloscope, we can check to see if the filtered signal is producing less low frequency noise than both original signals as well as less high frequency noise than both original signals. 3. We can move objects away from the user and keep them stationary to see if the haptic pads will stay off. Then we will test the intensity of the haptic pads based on how fast/close an object approaches the user. (Increasing in intensity as it gets closer).

Table 4: User Interface Subsystem – Requirements & Verification Pt. 4

Requirements	Verifications
1. The haptic pad must be able to produce a powerful enough sensation to be felt by the user when activated. It must also be able to scale in intensity depending on the distance from the object to the user.	1. Using the waveform generator, we will replicate the expected signals that the haptic pads will receive. If the vibrations are felt by the user from the lowest threshold (63 Hz) to the highest threshold (286 Hz), then the haptic pads are powerful.

References

1. IEEE Code of Ethics, IEEE, 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html
2. “10.525GHz Doppler Effect Microwave Motion Sensor Sku: CQRSENWB01.” CQRobot, http://www.cqrobot.wiki/index.php/10.525GHz_Doppler_Effect_Microwave_Motion_Sensor_SKU:_CQRSENWB01.
3. Ultrasonic Ranging Module HC - SR04 - SparkFun Electronics. https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf.
4. “STM32F303K8.” STMicroelectronics, https://www.st.com/en/microcontrollers-microprocessors/stm32f303k8.html#documentation.
5. “Vision Impairment and Blindness.” World Health Organization, World Health Organization, https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment#:~:text=Prevalence,near%20or%20distance%20vision%20impairment.
6. S. Kandhasamy, Sensor fusion with complementary filters. 2015.