OcuFeel Project Report

Callum Little, EE, Matthew Corcoran, CE, Pradeep Manivannan, EE, and Jon McDonald, CE

***Abstract*—According to an Association for Research in Vision and Ophthalmology (ARVO) study, they have estimated that more than 200 million people currently have some form of visual impairment [16]. As a result, the visually impaired must rely on some sort of detection method so they can go about their lives normally. More times than not, the method they use limits other body parts, so our team has come up with a solution. *OcuFeel is* a haptic device that is worn, and it can detect the distance of objects during natural walking motion by utilizing head movements. This device comes with two pieces, a headset that detects objects in front, above and below the user in the direction they are facing with the potential to detect 108° each way from the direction the waist is pointing. This prototype is rechargeable, provides the user with audio cues, provides feedback in real time, (<50 ms) does not impede innate motor functions, and has a weight similar to VR headsets.**

I. Introduction

The blind and the visually impaired predominantly utilize the white cane, a device that allows the user to detect objects in front of them and allow them to navigate the world. However, this tool does not have the ability to help detect distance (outside the length of the white cane), depth of objects in the environment without physical contact and objects in motion.

A. Significance

Whether you are born with or developed your impairment over time, the loss of vision "leads to disability, morbidity, and loss of productivity" [14]. Our goal is to help and lessen the load of visually impaired users such that they can be happier, less obstructed, and more productive to society. Technological advances to assist the visually impaired can additionally help reduce the strain of the annual economic impact (Currently $51.4 Billion) [14]. Furthering their contributions is a win-win situation for everybody, it will help the users, and will eventually lead to more government spending in other important sectors once this technology progresses enough to make them more self-sufficient.

B. Context and Competing Solutions in Marketplace

Current solutions to the problem only exist in academia with no known solutions on the market. The most prominent solution is the use of the white cane, and/or a seeing-eye dog, however both of those limit the use of hands [5]. An overview of these solutions involves solving the problem of assisting a blind person by helping them navigate their surroundings by using other senses. One solution utilized GPS and GSM to notify a caretaker on their mobile phone regarding the user's location in addition to providing the user directions via audio feedback [3]. The user was given directional instruction via a haptic feedback glove. However, it should be noted that gloves can hinder the user's motor functions and require them to hold their hands out continuously to sense the surroundings. Another solution uses a medical walker and a laser scanner to map the environment around the user [2]. This solution uses a haptic waistband as well, however the walker restricts the use of their hands. Another solution, created by a research group at MIT, used time-of-flight distance sensors worn around the user's waist with haptic motors located on the upper abdomen that provided feedback regarding the user's surroundings [1]. Other solutions that were interesting that we came across were proof-of-concepts, such as using a Xbox Kinect [4] and a The idea of using a waistband for haptic feedback seemed to be a viable solution since it would be out of the way for a user. Our proposed solution uses time-of-flight sensors as well, rather on a headset which enables the user to look around as one would normally do. Haptic feedback is provided on the waistband subsystem with the two systems communicating via the Enhanced ShockBurst (ESB) protocol.

C. Societal Impacts

As mentioned in our introduction, our system is primarily designed for those who are completely, visually impaired. We hope that our system can become a possible aid to those who are affected by blindness. With this being a wearable device, the comfort of our users was taken into great consideration. This was a factor in our weight specification on our headset system. We also wanted to design the belt so that it was comfortable to wear throughout the day. Throughout the process of our design, we also realized that our device can also be used in low-light environments so a user can still navigate their surroundings. With the current prototype we have created, we hope that it will somewhat benefit the visually impaired, but it is certain that there are much more needed tests that must be ran to confirm working and non-working features. We are hopeful that in the future, another group would be able to make our system more robust, smaller, and allow it to help the visually impaired traverse more precisely, quicker, and safe.

D. System Requirements and Specifications

Specification	Goal	Actual/Note
Forward Range	> 8 m	~12 m
Detects low clearing hazards	50 cm away from headset (1 motor)	Successful
Detects Tripping Hazards	5 cm tall obstacle from ground (1 motor)	Successful
Haptic motor angular resolution	27-degree intervals over 10 motors.	Triggers at 10 g
Response time	< 500 ms from sensor reading to motor output	~40 ms
Error checking Methods	Can determine: Loss of wireless communication, system malfunction, data corruption	Successful: Audio Cues
Battery Management	Rechargeable and replaceable	Successful
Battery Life	> 3 hr.	~ 9 hr.
Weight	< 600 g	522 g

Table 1. Technical Specifications

System requirements and specifications were decided to keep the system simple enough for the user to navigate in normal indoor conditions such as a classroom or office. We have determined that an 8-meter range should suffice to detect objects in such environments. For detecting low clearing and tripping hazards, the system should be able to detect objects within 50 cm for low clearing objects and 5 cm tall objects off the ground for tripping hazards. As for the haptic waistband design, the angular resolution for each motor should be over twelve, 27-degree intervals. Since both systems communicate between each other wirelessly, we set a response time goal from sensor reading to waistband motor output to be less than 500 ms. Since the system is meant to be worn on a person, we designed it to be battery operated with a rechargeable and replaceable battery with a life of greater than 3 hours. The weight of the headset subsystem should also be less than 600 grams.

II. Design

A. Overview

The main design of our system consisted of two major systems, the headset and waistband. The headset system collects data, processes data, sends data and the waistband system receives that data and provides haptic feedback to the user. We settled on this setup for our design because of how it could perform the way we wanted it to without hindering any of the user's other senses except for eyesight. With this specific headset sensor method to collect our data, it takes advantage of the user 'looking around' in a natural way.

In the beginning stages of our project, we went through a few different iterations of how the feedback should be provided to the user. We eventually settled with the following waistband system. With the surface area that the waist area provides, we can provide feedback to a wide range of angles relative to the center of the body. With this large surface area, it is also easier to differentiate between two different angular locations than it would be in a small area. As the head turns, the waist stays in the same orientation, so it is practical to provide location data relative to the orientation of the head rather than having any feedback move with the sensors.

Other alternatives we went through when determining the optimal way to provide feedback to the user was providing feedback on the hands or on the headset itself. With providing feedback on the hands, we found that the hands will never be in a constant orientation, and there for be impossible to provide consistent and accurate feedback on object locations. With providing feedback on the headset itself, we felt that the surface area that we had to work with was not large enough accurately show different angular locations. We also felt that constant vibrational feedback on the head would not be as comfortable for the user as it would be around the waist.

B. Headset Subsystem

The headset subsystem provides distance and angle readings which are sent via Enhanced ShockBurst (ESB) to the waistband. There are a total of four distance sensors and one gyroscope/accelerometer package. The two TFMini Plus LiDAR time-of-flight sensors [11] face forwards for distance detection directly in front of the user. One TFMini-S LiDAR time-of-flight sensor [12] is angled 45 degrees downward for trip detection and one Sharp Short-Range IR [15] is angled 45 degrees upwards for low clearance hazards. The LiDAR sensors were chosen for forward distance measurement since these would be the primary distance readings and we wanted extreme precision to determine the intensity of the motors. The TFMini Plus had 12m ranging capabilities with 5mm precision. Other forward distance sensors were considered such as sonar and IR, but these did not provide as much accuracy or range. Originally a Sharp Long-Range IR (the GP2Y0A710YK0F) was chosen for trip detection, but its output was too noisy, and a third LiDAR (the TFMini-s) replaced it. For orientation, the headset utilizes the LSM6DSO 6-DoF gyroscope and accelerometer [13].

With our custom PCB design, the gyroscope/accelerometer package is placed near the center of the board. The PCB is controlled using the Raytac MDBT50Q-1MV21 containing the nrf52840 as the MCU along with a 3D ceramic antenna [17]. There is also a speaker connected which enables audio feedback signaling low battery. The three LiDARs receive continuous data readings approximately every 10ms but goes through a multiplexor due to there being two UARTs instead of three. The Short IR receives data into the analog pins every 15ms and the gyroscope/accelerometer uses the I2C to receive data. The sensor information is forwarded to the MCU to process, then the motors and corresponding intensity is chosen and fit into a 3-byte packet, which is transmitted to the waistband via ESB. It should be noted that the MCU is connected to the power booster which is powered by the 3.7V battery.

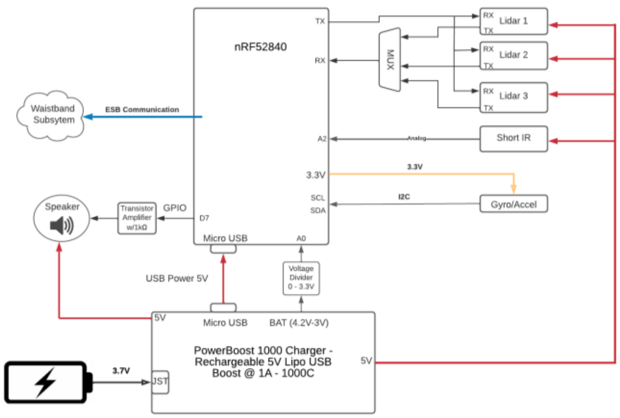

Figure 1. Headset Block Diagram

C. Waistband Subsystem

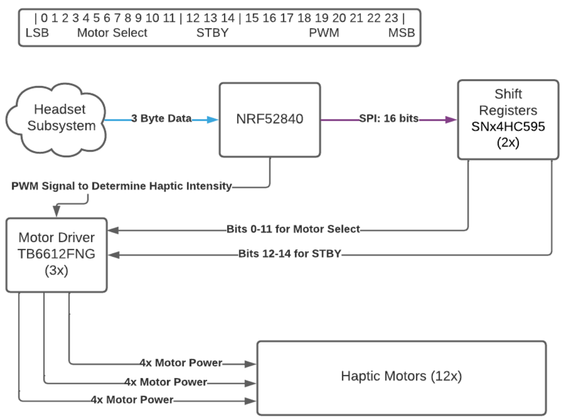

Figure 2 Waistband System Block Diagram

The waistband system of OcuFeel provides the physical feedback to the user based on the user's surroundings. The two main components of the waistband system are the waistband PCB and the haptic motors. Our waistband PCB is controlled by a Bluetooth capable MCU, the nrf52840. This MCU is on a Raytac module, MDBT50Q-1MV2l, which includes a chip antenna. The waistband PCB also includes 2 SNx4HC595 shift registers, 3 TB6612FNG motor drivers, a power booster/charging system, as well as connectors for 12 haptic motors. The high-level block diagram of the waistband system is shown in Figure 2.

The software for this system was contained within the nrf52840 MCU and is described in more detail in section II-D. Once the data received from the headset is decoded on the waistband MCU, the MCU outputs the corresponding PWM signals to the 3 motor drivers. It also outputs 16 bits of motor select and standby data via SPI to the 2, cascaded 8-bit shift registers. From the shift registers comes signals to the specific motor driver corresponding with the SPI data they received. On the motor drivers, the combination of the PWM signal, which determines the strength of the haptic response, and the motor select signal, which determines the location of the haptic response, a haptic feedback response is created for the user based on the initial data collected by the headset system.

D. Software

Software was distinctly divided over the two subsystems. Code ran on top of the Zephyr RTOS [6] [7], which the nRF52 line of chips supports. Zephyr, through its device tree, allowed for low level driver access and APIs to be used effectively to drive the connected peripherals. Development was facilitated using the nRF Connect for VS Code extension, which allowed for easy development and prototyping. Debugging was done using a SEGGER J-Link EDU.

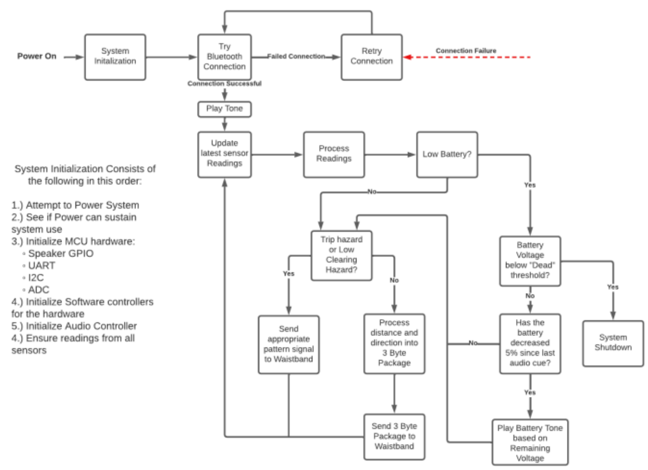

Figure 3: Headset software diagram

1) Headset Software

The headset software takes an object-oriented approach and is written in C++. The object-oriented approach allowed for better organization and easy modification of the code. Every peripheral had an object (typically referred to as "controller") that was responsible for keeping track of the data and contained methods that interacted with the peripherals.

The software follows the diagram in Figure 3. The program starts by initializing an 'OcuFeel Headset' Object, which in turn contains the controller for each of the peripherals. Each peripheral initializes their appropriate hardware drivers and verifies that they are working appropriately. Afterwards, ESB initialization is started, which requires the initialization of high-frequency hardware clocks and ESB protocol specific configurations.

Once initialization is complete, a connection attempt to the Waistband subsystem is made. If a connection is not established, the program tells the Audio controller to play a no-connection sound. The system continues to do this until a connection is established.

Once communication between the subsystems is established, the system begins to poll all peripherals and obtains raw values for each sensor. Once this is done, processing of the raw values begins in the OcuFeel object. The first things processed are the readings of the downward-facing LiDAR and the upward facing IR sensors. The readings of these sensors are more critical than the forward-facing LiDARs, since they indicate that an immediate obstacle is present to the users. In software, the IR's value is calculated through an exponentially weighted moving average which acts as a low-pass filter for the sensor. The LiDAR's output is accurate and precise to within 5mm, and since the processing times of each LiDAR is the longest compared to all other sensors on the Headset (each taking about 10ms), the moving average was not deemed necessary for the LiDAR. If either the IR or downward LiDAR value was within a user defined threshold, then a hazard was determined, and notification is immediately sent to the waistband system.

If a trip of low-clearance hazard is not determined to be present, then the forward-facing LiDAR's distance values are processed. The average of the two distance sensors is taken and this average determines the strength of the motors, as defined in Figure 4.

The LSM6DSO's values, which include a MEMS gyroscope and accelerometer, are then processed. The Headset system keeps track of the angle at which it deviates from zero degrees, defined as in front of the person. This is done by integrating the angular rotation provided by the gyroscope over time from sample to sample. To reduce the angular drift over time, the integrated value is also put through a complementary filter with the accelerometer values [9]. The current angle from zero was used to determine which motor to actuate, as shown in Figure 5.

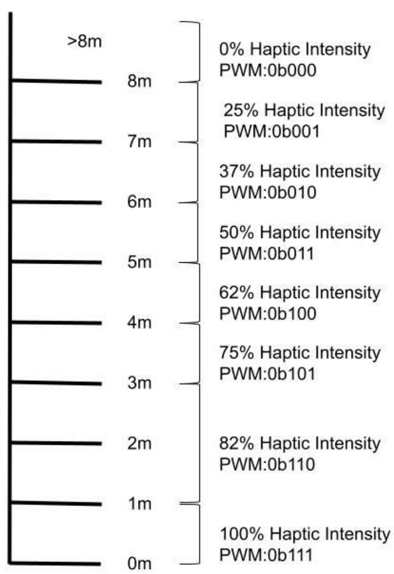

Figure 4: Waistband motor distance-to-intensity breakdown

The choice of motor and the vibration strength of the motor is condensed into the three-byte packet that is detailed in Section F. The packet is then sent to the Waistband system.

After a certain number of processing loops, the OcuFeel system will measure the voltage of the battery. The voltages that define a "Low Battery" status is compared against the measured voltage, and in this case an audio cue is played to indicate that the battery is low on energy. If the battery decreases another 5%, the audio cue is repeated. Once the voltage on the battery is determined to be at a point where it can no longer supply 5 volts through the power booster, a shutdown audio cue is played, and the system shuts down.

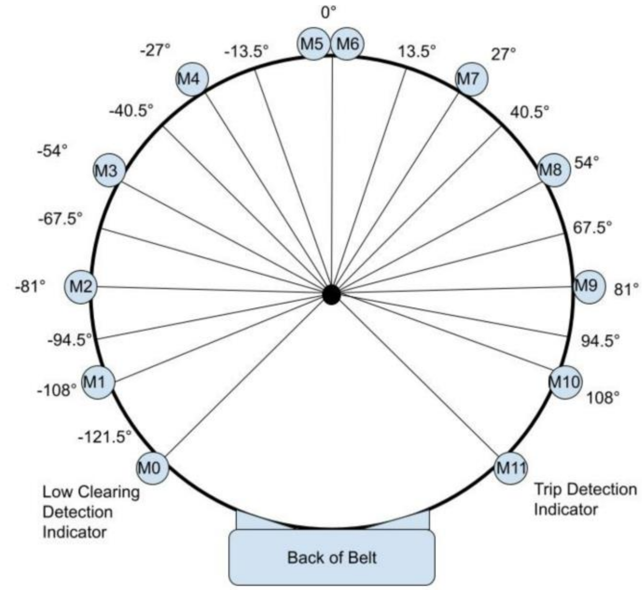

Figure 5: Waistband Motor angular breakdown

If a packet fails to be sent after processing, the headset indicates as such and plays an audio cue. It then will attempt to reconnect to the Waistband system. This should coincide with the Waistband also being unable to communicate.

2) Waistband Software

Code on the Waistband was written in traditional C. Drivers that are required for the waistband are the PWM, SPI, and GPIO drivers. ESB communication overhead is defined in section F.

The Waistband software accomplishes what is set forth in our waistband software diagram, Figure 6. Initialization consists of initializing the ESB into RX mode, high frequency clocks, configuration of PWM and SPI drivers. The software main loop then starts by waiting for an ESB receiving message from the headset. Once established, the waistband decodes the packet as outlined in Section F and makes decisions as to which motor to actuate and at what strength. The PWM is driven by the MCU and outputs into the motor drivers, while the actuation is sent to two eight-bit shift registers to determine which motor to actuate.

When a connection loss occurs, the Waistband MCU will wait for a reconnection. The software, depending on what data is sent, also decides if there is a tripping hazard or low-clearing hazard if the headset sends such information.

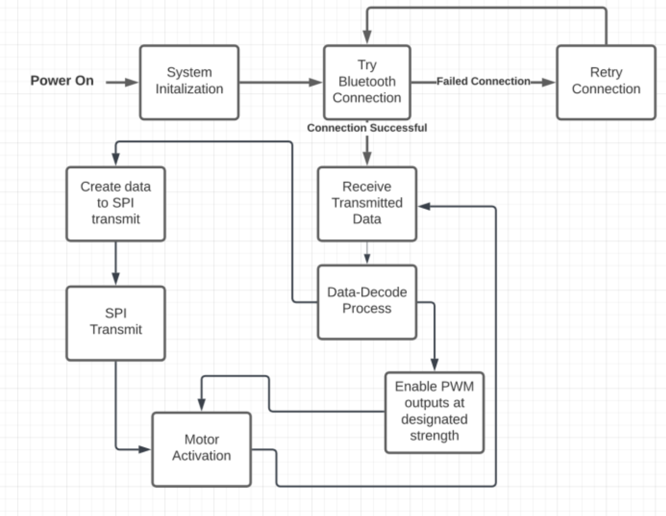

Figure 6. Waistband Software Diagram

E. Communication

Communication between the Waistband and Headset is facilitated by the Enhanced ShockBurst (ESB) radio communication protocol [8] [10]. ESB is a proprietary protocol by Nordic Semiconductor, the maker of the two MCUs OcuFeel utilizes, the nRF52840. ESB allows for a simple two-way communication with minimal overhead between the two microcontrollers.

The headset is responsible for processing data from its sensors into a three-byte sequences that corresponds to which motor is actuated, motor driver enables (STBY), and each of the three-motor driver's strength via a 3-bit PWM value. This data fusion and packaging is then sent via ESB from the headset to the waistband. The waistband takes the three-byte sequence and appropriately actuates the wanted motor at the wanted strength.

F. Physical Design

As our system's specifications highlight our project as being a wearable device, the physical design of our device was very important. Our device is split into two main physical components: the headset and the waistband. Both systems need to be able to contain the PCB and a battery for power. Both systems will also have physical indicators on them that a visual impaired user can use to determine the correct orientation of the headset and the waistband.

The main purpose of the headset is to act as the "eyes" of the device. To accomplish this goal, the headset houses our four distance sensors, and our gyroscope is located on the headset PCB. In order to house all of these components as well as others and be worn comfortably on the head, an ergonomic and lightweight design was a important specification. The headset physical design was designed using TinkerCad and 3D printed. With the capabilities of 3D printing our design, very specific details to accurately mount our distance sensors at the right angles and location. The specification for the weight of the headset was less than 500g in order for the headset to still be comfortable to wear by the user. The 3D design of the headset is shown in Figures 7 and 9.

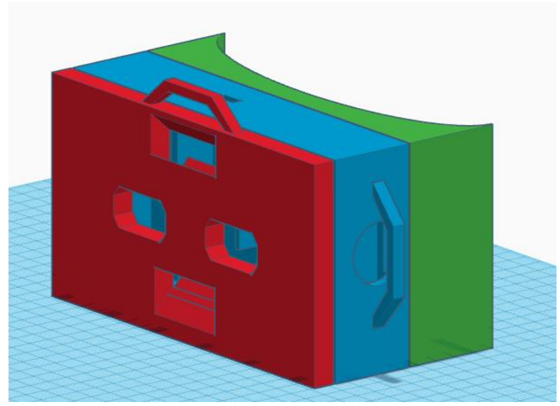

Figure 7. Headset 3D Design

The headset also has places for the speaker, microUSB charging, and a power switch. The waistband physical design is much simpler than the headset as it only has to house the battery and PCB. The waistband also has openings for the microUSB charging, power switch, as well as the openings to feed the connections to the haptic motors. The haptic motors will be placed around the belt that will be attached to the housing.

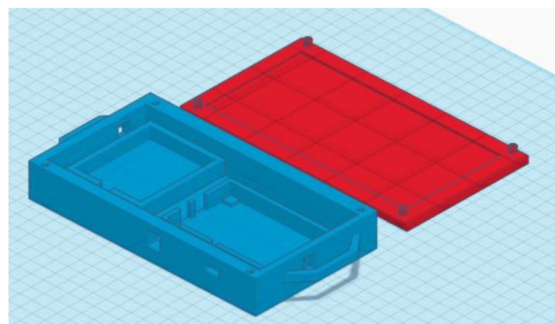

Figure 8. Waistband Housing 3D Design

The headset has three handles that allow for a strap to be passed through so that the user can secure the headset to their head. An elastic strap used for headlamps was used and was attached to the two handles on the side of the headset.

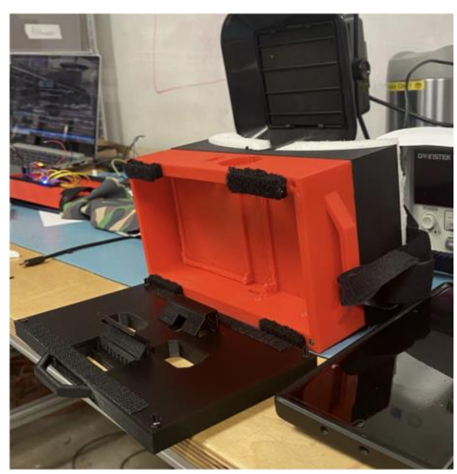

Figure 9. Inside of Headset Housing

The top of the strap goes over the user's head to secure the headset in place. The waistband was made using a stretchy belt. Fabric was placed around the outside of the belt which housed the motors and all the wiring. The 3D modle of the waistband housing (sans belt) is in Figure 8

Figure 10. Physical Waistband System

Figure 10 shows the physical waistband prototype. The 3D printed housing in the center holds the PCB and serves as a hub where all the motor wires are connected. Extending out each side of the housing are the belt straps that wrap around the waist of the user with the housing being at the back center of the waist. The straps hold each of the haptic motors spread out to match the designated angles that they represent. The straps also hold all the wiring that is routed from the housing at the back to each of the motors. The wiring includes a power wire for each of the 12 original motors as well as a joined ground route for each of the motors to connect to that was routed to the PCB. These straps were overlayed onto a normal belt with a traditional belt buckle that the user could use to connect the belt together in the front.

III. The Refined Prototype

A. Prototype Overview

Our final prototype was the physical implementation of our block diagrams for both the headset and waistband systems. In our prototype the general idea of providing haptic feedback to users based on input data from the headset was successfully implemented. The user can wear the system. With the headset the user can 'look around' and collect data using our sensors about distance and orientation. This data is then successfully communicated to the waistband system that they are also wearing. The waistband can provide haptic feedback that is easily tangible based on the data collected by the headset. The main functions shown in our block diagram were able to be displayed in our prototype. Explained in the sections below are how specific items in each of the systems block diagrams were implemented in the prototype.

B. List of Hardware and Software

All all-major hardware and software used in the OcuFeel system are noted below:

Headset Hardware:

- LiDAR (3x)
- Short IR (1x)
- 6DOF Gyroscope/Accelerometer
- Speaker
- Adafruit Express nrf52840
- 4400 mAh Rechargeable Battery
- Adafruit PowerBooster 1000C

Waistband Hardware:

- Shift Registers (2x)
- Motor Drivers (3x)
- Haptic motors (12x)
- nrf52840
    - Raytac Wireless Comm. Modu
- 2500 mAh Rechargeable Battery
- PowerBoost Charger

Software:

- nRF Connect for VS Code IDE
- J-Link RTT Viewer (debugger)
- Altium (PCB Design)
- TinkerCad (design 3D model)
- Cura (print 3D model)

C. Custom Hardware

In our project there are two different PCBs. One for the headset and one for the waistband and they communicate wirelessly with each other. With the wearable nature of our device, we wanted the PCBs to be as small and condensed as possible while still being accessible to us as developers. However, the headset PCB had power issues when adding and removing ICs from breakouts. This was due to the chip shortage and chips being unavailable unless the breakout was purchased. In the final prototype, the headset used a protoboard using the Adafruit Express nrf54280, Adafruit Powerboost 1000C, Sparkfun 6 DOF Accelerometer/Gyroscope breakouts. The multiplexer used the SMT variant but placed on a DIP adapter and soldered to the protoboard. All connections were soldered, and the sensors were connected using header connectors. The battery was then connected directly to the power booster breakout to power the system. A second revision PCB could have been made but due to problems with PCB manufacturing and interest of time, the protoboard was the most viable option.

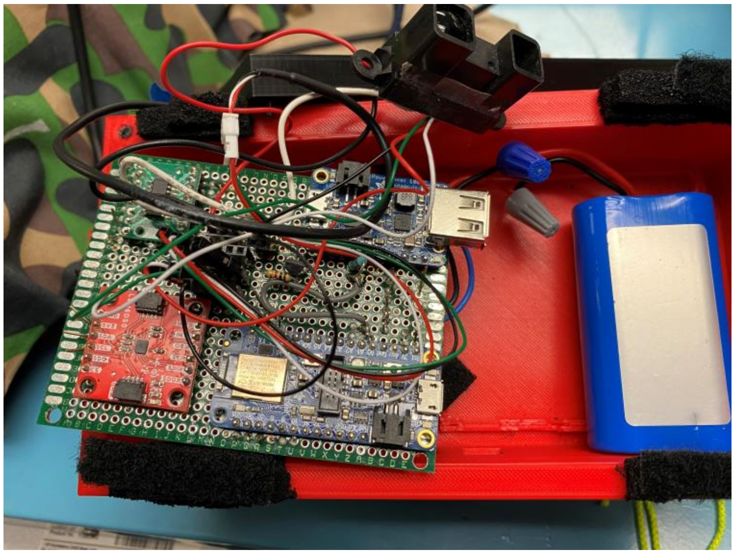

Figure 11. Headset Protoboard

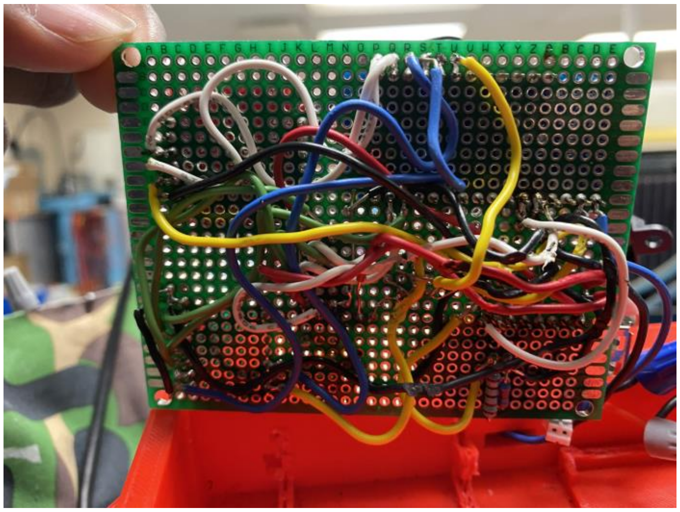

Figure 12. Headset Protoboard back side

The waistband PCB consists of two main ‘sections.’ The power booster section and the motor control section. The power booster section of our PCB allows our system to not only be battery powered, but also microUSB powered. It is the same as one the headset PCB. The power booster provides our 5v supply voltage that powers our MCU as well our shift registers. From the MCU a 3.3v supply is also used that powers our motor drivers. The MCU and the rest of the board other than the power booster section is dedicated to the haptic motor control. The motor control outputs to two headers on either side of the headset. Each header will be connected to six motors that span the right and left sides of the belt. We assembled the waistband PCB first and went through extensive testing of the power system as it was having issues. The headset PCB uses a similar power setup, so we wanted to have it working on one PCB before the same issue arose on the headset PCB. Due to our powering issues we resorted to assembling a new board. In our final prototype we also had issues with the shift registers that were supposed to be implemented on the PCB so in the end they were externally connected to the waistband PCB.

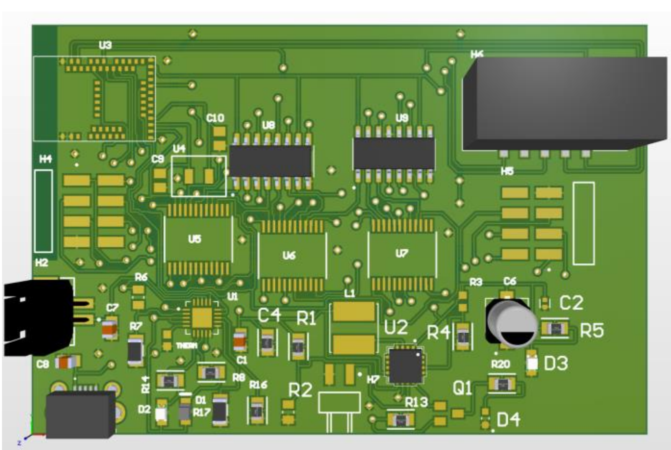

Figure 13. Top View of Waistband PCB

Figure 14. Bottom View of Waistband PCB

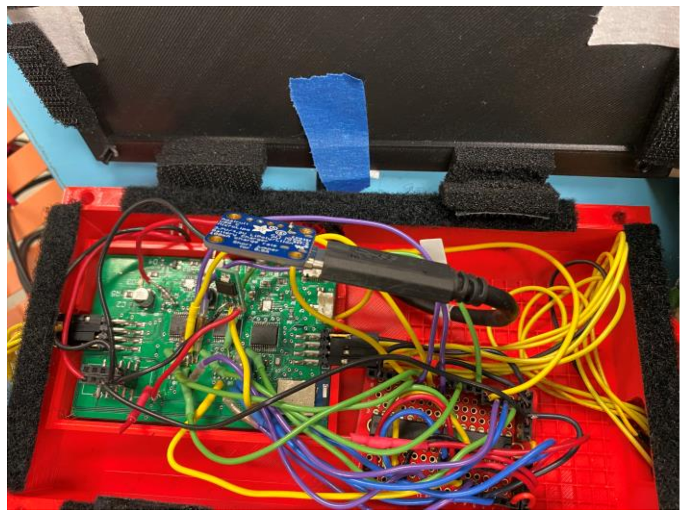

Figure 15. Actual Implementation of Waistband PCB

D. Prototype Functionality

Using your block diagram and software diagrams, discuss what elements of your prototype were functioning and those that were not functioning. Provide evidence that your prototype worked and discuss the issues faced in troubleshooting the non-functioning elements.

The headset was able to use all four distance sensors to acquire distance and angle readings and then send the 3-byte information packet to the waistband via ESB. If a low clearing or tripping hazard occurred, that information would take priority and be sent to the waistband. The audio feedback subsystem was not present in the full prototype due to soldering issues on the protoboard causing substantial noise to the speaker. The waistband was able to receive the 3-byte data within less than the specified 500ms. The headset was also comfortable enough to wear for a user and weighed less than the specified 600g. The battery was able to exceed the 3-hour specification.

E. Prototype Performance

Based on our specifications, the prototype was able to detect objects past the 8-meter distance with 5mm accuracy for forward detection. Low clearing hazards within the 50cm threshold were also successfully detected. As for tripping hazards, objects at 5cm were not tested but rather common objects found in an office were used such as chairs and tables. In our demo, the legs of nearby people also tripped potential tripping hazards. The gyroscope/accelerometer provided accurate angle measures that would turn on the corresponding haptic motors. And when turning corners, the gyroscope would recalibrate the angle measure such that if the user is facing forward, the angle is zero degrees. The actual processing time averaged at 34ms with a worst case of 40ms, far exceeding the specification of 500ms. The haptic motor angular resolution changed from 27 degrees since four of the motors on a PWM did not work. M5 spanned 54 degrees, M2, M3, M8, and M9 were all 40.5 degrees. M11 was used for tripping hazards and M0 for low clearing hazards. The

IV. Conclusion

Upon Friday April 29, 2022, the ECE department's Senior Design Project Demo Day, we were able to successfully demonstrate a prototype of our system to the attendees of the event. Throughout the process of completing our prototype we ran into many issues that we have talked about in this report but in the end, we were able to complete and demonstrate a working and functional system. In our demonstrations, group member Pradeep Manivannan was able to walk around our area using our system and based on his reports was able to successfully navigate his surroundings without hitting any obstacles. We were also able to let other people try on our system and feel the haptic feedback that our system provides. As a team we felt like we performed a successful demonstration despite setbacks that we experienced throughout the process of creating a prototype.

Acknowledgment

Team 17 would like to thank Prof. Michael Zink for his unwavering and constant support throughout our design process. We would also like to thank Prof. Amir Arbabi and Prof. Hossein Pishro-Nik, our evaluators, for their helpful criticism, support, and advice throughout the two semesters.

We would also like to thank the SDP22 course coordinators, Prof. Christopher Hollot, Prof. Baird Soules, Prof. Shira Epstein, and Dr. Charles Malloch, for their immense help and support throughout the two semesters.

We would like to particularly thank Christopher Caron, the teaching assistant for SDP22, for his dedication and perspectives that he shared with our team as we continued with our design.

We would also like to thank the staff of M5, the ECE makerspace at UMass Amherst, the staff, and members of the All-Campus Makerspace, and the many, many faculty and staff of the ECE department.

References

[1] H. Wang, R. K. Katzschmann, S. Teng, B. Araki, L. Giarré and D. Rus, "Enabling independent navigation for visually impaired people through a wearable vision-based feedback system," 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 6533-6540, doi: 10.1109/ICRA.2017.7989772.

[2] A. Wachaja, P. Agarwal, M. R. Adame, K. Möller, and W. Burgard, "A Navigation Aid for Blind People with Walking Disabilities," in IROS Workshop on Rehabilitation and Assistive Robotics: Bridging the Gap Between Clinicians and Roboticists, Chicago, USA, 2014

[3] S. T. H. Rizvi, M. J. Asif and H. Ashfaq, "Visual impairment aid using haptic and sound feedback," 2017 International Conference on Communication, Computing and Digital Systems (C-CODE), 2017, pp. 175-178, doi: 10.1109/C-CODE.2017.7918924.

[4] J. R. Alayon, V. G. D. Corciega, N. M. L. Genebago, A. B. A. Hernandez, C. R. C. Labitoria and R. E. Tolentino, "Design of Wearable Wrist Haptic Device for Blind Navigation using Microsoft Kinect for Xbox 360," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2020, pp. 1005-1010, doi: 10.1109/ICOEI48184.2020.9143005.

[5] M. Bousbia-Salah, M. Bettayeb, and A. Larbi, (May, 2011) A Navigation Aid for Blind People. Journal of Intelligent Robot Systems [Online]. 64, 387–400. Available: https://link.springer.com/article/10.1007%2Fs10846-011-9555-7

[6] "Zephyr project," *Zephyr Project*, 22-Feb-2022. [Online]. Available: https://www.zephyrproject.org/. [Accessed: 04-Jan-2022].

[7] "Zephyr project documentation," *Zephyr Project Documentation - Zephyr Project Documentation*, 04-May-2022. [Online]. Available: https://docs.zephyrproject.org/latest/. [Accessed: 02-Feb-2022].

[8] "Nordic Semiconductor Infocenter," *Nordic Semiconductor infocenter*. [Online]. Available: https://infocenter.nordicsemi.com/index.jsp. [Accessed: 21-Jan-2022].

[9] P.-J. Van de Maele, "Reading a IMU without Kalman: The complementary filter," *Pieter-Jan*, 26-Apr-2013. [Online]. Available: https://www.pieter-jan.com/node/11. [Accessed: 13-Feb-2022].

[10] D. Veilleux, "Intro to ShockBurst/enhanced shockburst," *Blogs - Nordic Blog - Nordic DevZone*, 08-Nov-2015. [Online]. Available: https://devzone.nordicsemi.com/nordic/nordic-blog/b/blog/posts/intro-to-shockburstenhanced-shockburst. [Accessed: 04-May-2022].

[11] Benewake LiDAR Team, *Product Manual of TFmini Plus*, Benewake (Bejing) Co. Ltd.

[12] Benewake LiDAR Team, *Product Manual of TFmini*-S, Benewake (Bejing) Co. Ltd.

[13] STMicroelectronics, "iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope", LSM6DSO datasheet, 2019. Available: STMicroelectronics, https://www.st.com/en/mems-and-sensors/lsm6dso.html#documentation [Accessed: May 3, 2022]

[14] "Economic studies - vision health initiative (VHI)," *Centers for Disease Control and Prevention*, 10-Aug-

2021. [Online]. Available: https://www.cdc.gov/visionhealth/projects/economic_studies.htm. [Accessed: 04-May-2022].

[15] SHARP, “GP2Y0A02YK0F”, datasheet. Available: SHARP, https://media.digikey.com/pdf/Data%20Sheets/Sharp%20PDFs/GP2Y0A02YK0F.pdf [Accessed: May 4, 2022]

[16] R. R. A. Bourne, J. Adelson, S. Flaxman, P. Briant, M. Bottone, T. Vos, K. Naidoo, T. Braithwaite, M. Cicinelli, J. Jonas, H. Limburg, S. Resnikoff, A. Silvester, V. Nangia, and H. R. Taylor, “Global prevalence of blindness and distance and near Vision Impairment in 2020: Progress towards the Vision 2020 targets and what the future holds.,” *Investigative Ophthalmology & Visual Science*, 10-Jun-2020. [Online]. Available: https://iovs.arvojournals.org/article.aspx?articleid=2767477. [Accessed: 04-May-2022].

[17] “MDBT50Q-1MV2,” *Raytac MDBT50Q-1MV2*. [Online]. Available: https://www.raytac.com/product/ins.php?index_id=24. [Accessed: 04-May-2022].

APPENDIX

A. Design Alternatives

Other designs were considered for both the sensing and feedback subsystems. Placing distance sensors on the body was considered however we found this to be intrusive when wearing normal clothing. It also limited the user’s movement since they would have to turn to face the direction of an object. We also considered placing sensors on the hands, but the user would lose mobility and functionality of their hands. Also, the user would have to keep their hands up to “look” around which would be uncomfortable for long periods of time. As for the feedback, we considered having feedback on the user’s face since they would be able to easily feel the vibrations but decided against it since it would not be something people would want to wear. We also considered feedback on the hands since the hands had the most nerve endings, but it would also limit the user’s hand functionality. The waist was chosen because it was out of the way and could provide directional movement the best. The waist could also hold more weight than other body parts so it would not fatigue the user after prolonged use. The head was decided for the sensors since it mimicked normal functions of a person looking around.

B. Testing Methods

So far testing has been limited to sensor and haptic motor testing rather than the full system, which is going to be added. Accuracy tests were performed for the LiDARs which resulted in accuracy within 5cm. An object was placed at a known distance and the distance was measured. The average distance for each interval and standard deviation were calculated.

Distance Intervals (cm)	Avg. Distance (cm)	St. Dev
200	198.9736842	1.237042765
300	304.6216216	1.523358786
400	403.9722222	2.843399486
505	505.2105263	1.685854461
510	510.6666667	0.887625365
600	603.0196078	1.870723882
705	705.9491525	1.718911689

Table 2: LiDAR Precision Measurements

The other major test that was performed was for the gyroscope/accelerometer. The system was placed on a servo motor that was programmed to turn to the degree intervals as they would appear on the waistband. The gyroscope starts facing directly forwards (0 degrees), and then turns to the specified angle. Meanwhile the start and ending angles are being read. The difference is then taken to compare it to the true angle value. The measured angles were found to be within 0.5 degrees of the true angle. Resulting in >2% error.

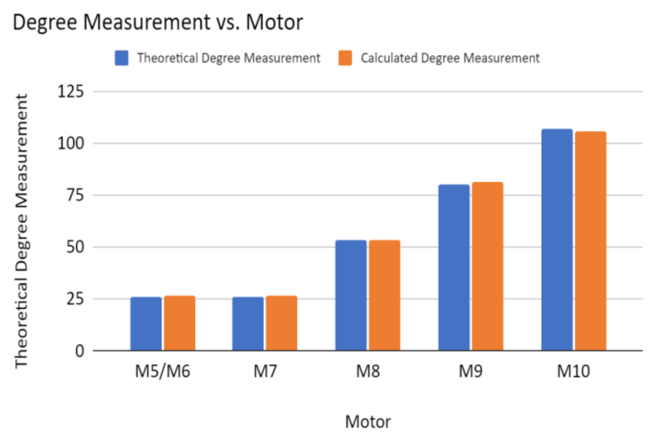

Figure 16. Degree Measurement vs. Motor

A test for low clearance hazards was also performed using the Short-Range IR. An object was placed further than 50 cm away from the sensor and a spot was marked at 50 cm indicating the true distance. Once the object passed 50 cm, an LED flashed indicating that there was a detection. The test was performed a total of twenty times with twenty successes.

C. Project Expenditures

Part	Price per Unit	Quantity	Total Price
Vibration Motor	$2.15	12	$25.80
SparkFun Triple Axis Accelerometer Breakout - ADXL335	$14.95	1	$14.95
TFMini Plus Lidar	$50.00	2	$100.00
Adafruit Feather nRF52840 Express	$24.95	1	$24.95
SHARP Long Range IR (100-550cm)	$9.85	1	$9.85
SparkFun 6 Degrees of Freedom Breakout - LSM6DSO (Qwiic)	$11.95	1	$11.95
Breadboard to JST-ZHR Cable - 5-pin x 1.5mm Pitch	$2.95	1	$2.95
JST to Breadboard Jumper (3-pin) (basic 2mm pitch)	$1.50	1	$1.50
5V 2.5A Switching Power Supply with 20AWG MicroUSB Cable	$8.25	1	$8.25
Lithium Ion Polymer Battery - 3.7v 2500mAh	$14.95	1	$14.95
Waistband PCB Rev 1	$15.10	1	$37.20
Waistband PCB Assembly Rev 1	$50.46	1	$67.85
PCB Headset Materials : Revision 1	$15.11	1	$35.47
CDR → FPR Additional Headers: 14.03 + 45.88	------------	------------	$59.91
Total Parts	------------	---------------	**$485.2**

Table 33. Project Expenditures

In addition to using the allocated $500 budget, our group spent $150 out of pocket for additional parts such as headers and additional MCU's as we were having PCB voltage problems. In conclusion, we spent a total of $650.

D. Project Management

Throughout our year of SDP, our team worked extremely well together. Going into our senior year we already had our team organized and thought that it would be best to have two electrical engineers and two computer engineers. This way we could have team members who expertise in different aspects of any project we decided to work on. Our team was fortunate enough to have worked together on multiple things since freshman year so we were well acquainted with each other and how each of us work on tasks. Throughout the two semesters we did not end up having any issues in team organization or group chemistry.

Group members Callum L. and Pradeep M. were the electrical engineers of the team. These members focused on the hardware portion of the project including circuitry, signal analysis, and PCB design and build.

Group members Matthew C. and Jon M. were the computer engineers of the team. These members focused on the software development for our system. Both members were heavily involved in debugging both hardware and software issues using the involved software.

All team members were able to help out each other in many different aspects of the project. As described in our presentations throughout the year, each team member focused on somethings while also aiding teammates on other tasks. Throughout the year all of our team members made the effort to work together in person and we felt like this was the best way to get our work done. When we were not in person, we used Discord to stay in contact and utilized application specific channels to organize our work and send documents and images amongst each other.

One example of when we as a team helped each other out is instances of when a group member had concerns outside of SDP and each of the other group members stepped up to pick up any work that needed to be done.

As a team, we were fortunate to not have any communication issues and were able to always work through any issues simply that we faced on the project.

E. Beyond the Classroom

Callum L:

The main skill I have needed to learn and develop outside of the classroom for this project is PCB design. With the help of Chris Caron and his instructional videos as well as his in-person help, I have learned a great deal about PCB design. Using the Altium software and other instructional videos, I was able to successfully design the system schematic, the PCB and order a completed PCB as well as export/order the BOM. Along with designing the PCB, I have also learned about assembling it. Throughout this project I have greatly increased my soldering skills in both SMT and through hole soldering methods. These skills will definitely have a connection to my future, professional career as I plan to work closely with PCB design and verification after I graduate.

Pradeep M:

The skill I developed from this project was PCB design and testing sensors. Watching Altium tutorials online from Chris Caron as well as from others have been helpful. Additionally, Chris' and the M5 staff's assistance has been helpful going through the process of gathering components to design the actual PCB itself. Learning soldering techniques is something that will be useful for me in the future. Sensor testing was another skill I learned, from choosing the correct sensor for each application, to writing test scripts and verifying if they were accurate and precise. These skills will have a connection to professional life since designing and testing PCBs will be a part of the research I hope to pursue in graduate school.

Jon M:

This project has, to a large extent, helped improve my understanding of how to code properly and the importance of structuring code in software development (Objects and Classes). Proper file structuring allowed us to code each sensor simultaneously as we just needed to include the C and header file to the main branch when it was ready to be added to the system. Understanding how to read the Zephyr documentation gave me so much insight into activating the peripherals from scratch in addition to the various peripherals and protocols each one must follow. One thing I was not aware of before starting this project was how complicated powering a system was when using a low power board such as the nrf52840. We had all sorts of voltage problems because our sensors required 5V, but our board could only output 3.3V so we had to use the power booster. It was of the utmost importance to confirm what we were inputting into the pin otherwise there could be an electrical failure.

One of my favorite parts about this project is verifying that our system works and creating our own specific tests from scratch. For a specification, we needed to measure voltage off a pin which required us to verify it beforehand with a multimeter. I thought this was quite interesting because the feature is such a small part of our design but is detrimental enough to ruin our system since if the charge drops too low, the battery can become irreparably damaged. I also learned how 3D printing works which might evolve into a hobby later in my life!

Matt C:

The skills I learned for this project have been centered around software development for our system, but also skills from applying digital design principles. Whilst I have a good foundation for programming microcontrollers, the nRF MCU and Zephyr OS were foreign to me, but I learnt to read through the datasheets more efficiently as time went on and now, I would say I have a familiar understanding of the Zephyr OS and programming MCUs that utilize it. Programming the chip on a custom PCB also required intimate knowledge of the circuitry the MCU was interfacing with. This helped me come up with some hardware designs, particularly with our communication circuitry on the LiDARs, and to have a good idea of the system's physical workings in order to create the code. I hope these skills will translate over to working and designing systems, as well as software programming in an embedded systems context.