Indoor Navigation for the Visually Impaired

By

Akhil Alapaty

Kushagra Tiwary

Saleh Ahmad

Final Report for ECE 445, Senior Design, Spring 2019

TA: David Hanley

1 May 2019

Project No. 73

Abstract

For our senior design project, we built wrist wearable devices that would provide indoor navigation to blind people. They function through Bluetooth communication with strategically positioned Bluetooth beacons across the characterized apartment. In order to give the best experience, we have developed user localization and path planning features in order to guide users to their final destinations. On the hardware side, the user can pick his/her destination through touch and will receive haptic feedback for navigation. The testing and characterization was done in a team member's apartment. Ultimately, we got each of our modules working with successful Bluetooth communication and proper input/output interaction. However, we did run into some memory computation problems, as the localization was done through a Particle filter, which required an immense amount of dynamic memory from the microprocessor.

Contents

1. Introduction

Blind people encounter several obstacles on a daily basis with actions we take for granted. In an interview conducted with blind people, the participants noted that the majority of the problems they faced were due to unsafe sidewalks, existence of obstacles, navigation in new spaces, people moving things around, and crossing streets [1]. To rectify some of these problems, many cities have adapted tactile paving inside public spaces and sidewalks to alert the blind of stairs, roads, and platform edges [2] [3]. However, navigating the space especially in larger indoor areas such as airports, malls, and apartment complexes still remain a challenge. Currently, this is largely done through guide dogs, assistance or the individual having to memorize the layout (usually for their homes and personal spaces) [4]. The guide dogs are extremely sensitive to the cues from the blind person and are trained to find exits, stairs and elevators, and even respond to threats [5] [6] [7]. However, wayfinding through airports and large indoor locations without assistance still remains an extremely difficult task mostly due to the complex indoor layouts, lack of directional-aid in non-visual formats and the loud constant echo-noises which can cause disorientation [1] [8]. In fact, independent grocery shopping is still one of the most functionally challenging tasks a blind and visually impaired person can do; therefore, it is essential that we design spaces and build the relevant infrastructure to be more inclusive [7]. The architecture and the lack of accessibility causes the blind person to largely avoid these spaces or only use them when absolutely necessary [1] [7] [8].

Our goal is to attempt to solve the wayfinding task for the blind and visually impaired individuals in indoor spaces such as apartment complexes, malls, etc. The wayfinding task includes locating the person and then subsequently successfully navigating them to their destination through constant vibrational feedback. To build the wayfinder application we will build "walkability" maps of the indoor spaces and use bluetooth beacons to localize the person. The blind person will be carrying a bluetooth compatible wearable-device on each hand, which will connect to the bluetooth beacons and also be used to input directions and output feedback. Using touch, the user will be able to select a destination from preset choices, and will receive navigational feedback through haptic and vibrational feedback. Our application will only handle the navigational and the associated feedback aspect of this problem. Testing of our design will be conducted in Kushagra's apartment; we will characterize the 27x14 m apartment into the aforementioned walkability map.

There are a couple of companies that currently work on indoor wayfinding devices for larger indoor spaces; however, these solutions are not necessarily geared towards the visually impaired. BlindSquare is one of the world's most popular apps the blind and visually impaired tend to use for navigation in outdoor settings; it announces points-of-interest such as street names, and saved locations and helps its users look up information regarding their surroundings with the help of GPS and third-party-apps [9]. IndoorAtlas is an application for indoor positioning system, which finds your position with the help of bluetooth beacons, WiFi signals, and barometric pressure to measure elevation for general users (is not focused on the visually challenged). This particular application can be downloaded off the Apple or Android Store; however, the application is not for the blind or visually impaired since it is simply indoor positioning technology for retail, healthcare and airports [10]. Regardless, the use of such bluetooth beacons is on the rise due to the large number of bluetooth

compatible devices people use. “Indoors” is a company that uses bluetooth beacons along with its mapping systems to help blind people navigate the complicated structure of the Los Angeles Airport. Wayfindr is a similar company focusing its product on the visually impaired and operates in some stations in the London Tube. Indoors has a similar structure in place where they first create high quality maps of the system and then use iBeacons with some more input from WiFi to get the location and then subsequently provide directions [11] [12].

We plan on our application and wearable device being as compatible, cheap and user-friendly as possible to the blind person- especially from the guidance perspective. Additionally, we will also have to keep in mind that there should be no discrepancies or position inaccuracies in regards to location performance within our mapped areas, when we position our beacons.

1.1 Block Diagram

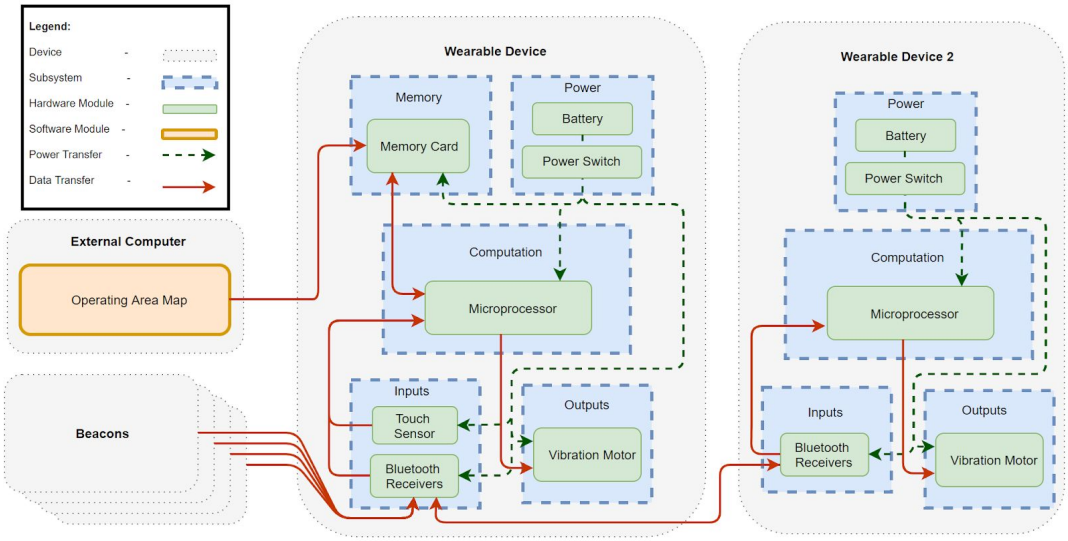

Figure 1 Top Level Block Diagram

The top level block diagram consists of many individual modules interfacing with each other. Wearable Device 1 seen above is receiving beacon RSSI (Received Signal Strength Indicator) values. Wearable Device 1 and Wearable Device 2 consist of the same components, except for the touch sensor and SD card on the former. The user will select his/her destination by utilizing the touch sensor, as the number of touches will correlate to various preset destinations. Through the Bluetooth receivers, Wearable Device 1 automatically connects to Wearable Device 2 in order to relay the beacon and user information. The information received from the touch sensor and Bluetooth receiver are passed on to the ATMega328P microprocessors, which is where the navigational decisions occur. The microprocessor on Wearable Device 1 will simultaneously communicate with the SD card, which consists of precomputed values that help determine the user location. Once the localization is completed, the microprocessor will determine the user’s location relative to the computed path from the user’s location to the destination.

Consequently, output correlating to either straight, left, or right will be passed back to the vibration motors. Straight is indicated by both motors vibrating and a turn is indicated by only the respective motor vibrating. This relay of information is illustrated in the following OV-1 Diagram.

1.2 OV-1 Diagram

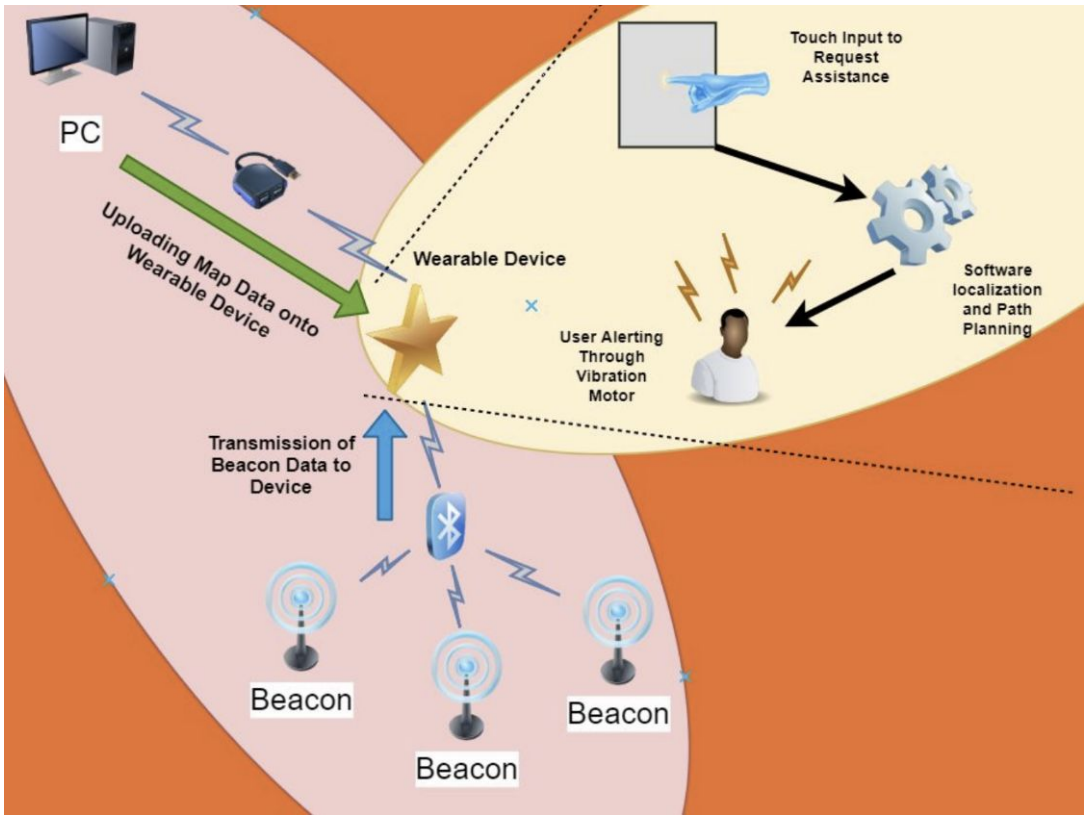

Figure 2 OV-1 Diagram

2 Design

As we worked on our project, we decided to make some changes to our original design. After testing our localization via triangulation using beacon RSSI values, we realized it wasn't very accurate or robust in its output. We decided to change to a more complex algorithm to solve our localization problem. Initially our design was comprised of only one wearable device, but as the project progressed we decided to add a secondary device on the left hand to make our direction output to the user easier to understand. Our direction output went from changing frequency patterns on a single device to determine direction, to two devices with left side vibration to go left, right side vibration to go right, and both sides vibration to go straight.

2.1 Processing

The microcontroller consists of our core software processing, which comprises of the beacon signal interpretation, determine the user's position, and the path planning to destination module. Additionally, it also contains the I/O module interfacing with the beacons, the touch sensor while simultaneously communicating the path to the user through the vibration motor. The microprocessor will be powered by a 3.7 V battery and will draw data input from our SD card. Our chosen microcontroller for this project is the ATMega328P, which will serve as the processing unit between the hardware and software modules, as well as the battery and SD card. The ATMega328P microcontroller is capable of operating on voltages between 1.8 V and 5.5 V, which is why we chose a 3.7 V Li-On battery.

2.1.1 Position Determination

As proposed initially, we had hoped to calculate the distance using the RSSI values from the beacons and use intersecting circles to calculate the position of the person the map. However, the RSSI values that the beacons outputted weren't reliable as they had a variance of at least 4-6 dB. Moreover, the values between 1.5 and 3m varied anywhere between -63 to -70dB depending on how the receiver is held, and interferences.

To create a robust localization system we evaluated existing literature and decided to compare performances of two filters: Discrete Bayes Filter and a Particle Filter. To implement both of these models we needed a motion and a measurement model. For simplicity we assumed the motion model to be the change in position between previous two states. That change in position will be propagated forward.

To create a measurement model, we created multiple fingerprint maps which are shown in the appendix. A fingerprint map of the environment records the RSSI value at that location from a particular beacon. This training data helped us create a model for our state space and beacon values. The values with -90dB represent obstacles.

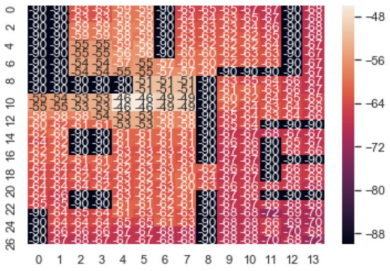

Figure 3 showing the fingerprint map for Beacon 4. All are shown in the appendix in figure 4

The training model used to get a mapping between a state space (x,y) to RSSI values is a Gaussian Process Regression model. The output of the model are 4 mean RSSI values, one corresponding to each beacon, and their corresponding variances. This model served as a measurement model for our localization task as we are able to get the likelihood estimates through this.

We noticed that the performance of the particle filter was substantially better than the discrete bayes since it did its computations in the continuous domain instead of discretizing it into bins. Moreover, the fact that our environment map was very small (27x14) contributed to the poor performance of the Bayes Filter. Please see figure 5 for a good output of our localization method.

Given a state (x,y) which represents the location of the particle, the GP model should output a mean and a covariance. Due to the already computationally expensive particle filter, we did not implement the posterior calculation done by the GP model online in the AtMega. Instead, we stored discretized the state space of the apartment such that each 1x1 block in the map was now a 10x10 block - every state was now rounded to the nearest decimal place. We pre-calculated the posterior mean and covariance on this discretized state space and saved it in the SD card. At every localization call, we then did a lookup on the SD card after discretizing the state. Figure 6 in the Appendix shows the file structure stored in the SD card that we index from the localization module.

2.1.2 Path Planning

The path planning module should be able to find the safest path to the destination given the current location. The safest path is a path that doesn't go through any obstacles or walls. The path planning is also then responsible to get the immediate next path and communicate that to the user in an understandable way therefore, map the next action to a set {STRAIGHT, LEFT, RIGHT} actions that the vibration motor can output.

To do so we leveraged the fact that our map was small and broke it into "waypoints" such that the action to get to every other reachable waypoint from a particular waypoint belongs in the action set that the vibration motor can output. Given a location, the algorithm computed the path to the closest waypoint and took there. This design was a powerful alternative from the BFS search as it took much less time to compute on the ATMEGA. Please see figure 10 for an example output of our path planning waypoint algorithm.

2.1.3 Beacon Signal Interpretation

This submodule was originally responsible for computing and choosing three strongest beacon RSSI values, and then filtering the values. However, since the localization module was changed, this module's task changed to just computing the RSSI values from all the beacons, filling in default values for the beacons from which no data was received, normalizing the data so that they can be inputted to the GP model for localization.

2.1.4 Input/Output

This block is responsible to communicate with the touch sensor and the bluetooth receivers. It received input signals from the bluetooth receivers and the touch sensor. It also outputted action states to the vibration motor along with its relevant frequencies for it to vibrate at. In the new block diagram this module also communicated with the slave receiver to output signals to its vibration motors in order to give appropriate feedback to the user regarding the direction.

2.1.5 Microprocessor Integration

We faced a lot of issues when we tried to integrate this system with the microprocessor. Since we had changed our localization system to be more complex and computationally more expensive, we had issues

with dynamic memory. We had accounted for a need of static storage by inserting a SD card, however, due to the particle memory we faced issues with the ATMega328p's dynamic memory allocation. Since the application kept interfacing with the SD card at every localization call, there was time extra overhead.

Even with all the enhancements, we could only successfully run the path planning and the localization module with 5-10 particles to get a close to real time result. This was because of loss of dynamic memory and the constant global lookups every localization call- there were 2 lookups per beacon and there were 4 beacons. In the future, we would want to use a processor that is more powerful especially now that we know that trilateration doesn't work and a more complex localization algorithm is required.

2.2 Power

The Power subsystem consists of the battery and power switch we used in our design. This module supplies power to the entire system, comprising of the microcontrollers, touch sensor, vibration motors, and Bluetooth receivers.

2.2.1 Lithium-Ion Battery

Our power source for the design is a 3.7 V 1500 mAh Lithium-Ion Battery. The main objective for this battery is to power the circuit and each of the hardware components. Additionally, we wanted a longer battery life for the design, as we do not want navigation to be disrupted in the middle. The table below shows our computations for the chosen battery:

Table 1 Expected System Power Usage

Module	Power Usage (mW)
Microcontroller	4-6
Bluetooth	150-200
SD-Card	150-200
Touch Sensor	1-3
Vibration Motor	150-200
Total	455-609

$$\text{Battery Energy: } 1500\ mAh * 3.7\ V = 5550\ mWh \quad (1)$$

$$\text{Device Power Consumption: } 455 - 609\ mW \quad (2)$$

$$\text{Battery Life Max: } (5550\ mWh/455mW) = 12\ hours \quad (3)$$

$$\text{Battery Life Min: } (5550\ mWh/609mW) = 9\ hours \quad (4)$$

Battery Life Range: 9 − 12 *hours*

2.2.2 Power Switch

The power switch is an easily accessible switch that will allow the user to easily turn the system on/off, since the power switch is directly connected to the Lithium-Ion battery. Its purpose is to preserve battery life as the system need not be turned on when not in use.

2.3 Memory

The Memory module is responsible for storing the walkability map, beacon locations, pre-computed location probability values, etc.

2.3.1 SD Card

The SD Card will store the information mentioned above in order to facilitate localization and support navigation. The SD Card will be directly interfacing with the ATMega328P microcontroller, as it sends the stored information. Since there will be multiple locations, there will be different maps stored in the SD card. When a particular beacon will connect to the system, the map belonging to that location will be used.

2.4 Inputs

These serve as the physical devices that the user communicates with (touch sensor) and the bluetooth receivers and beacons that will be used to localize the user in the operating area by the microcontroller.

2.4.1 Touch Sensor

This block serves to provide the microcontroller user input. The touch sensor feedback will determine the user's destination that the application guides to. This information is then submitted to the I/O module within the microprocessor, which is then processed in order to determine the user's starting and end point. At this point of our project, we are using a touch sensor to relay feedback through physical taps on the sensor. Beyond the scope of this semester, we expect to convert this input feedback into a voice processing system, in order to facilitate the process for our users.

2.4.2 Bluetooth Receiver

The receiver is incredibly important for our system since user location depends on the strength of the receiver. The Bluetooth receiver will be continuously detecting surrounding bluetooth beacon signals during operation. After the addition of our second device we need to add some functionality to our bluetooth module. We needed the two devices' bluetooth modules to communicate. We decided that the main (right) device's bluetooth module would be only one that scans for beacon signals, since both devices scanning wouldn't help with our localization. The secondary (left) device only waited for a connection in order to receive direction information. After determining direction, the right side's bluetooth needed to connect to the left side, send the direction information, disconnect, and start scanning for beacons again.

2.4.3 Beacons

The beacon module will consist of at least 4 bluetooth beacons located in the operating area. The positions of these beacons will be fixed in the operating area environment, and the receiver will communicate with these devices. The locations of these devices will be fixed and registered in the operating area map. The device will use this information with the strength of the signals to determine the user's location.

The placement and type of these beacons will be critical to the accuracy of the location of the user since there can be heavy interferences from WiFi, refraction due to pillar, walls, room size, other electronic devices, etc. The requirements below are made with these considerations in mind and we aim to have at least 4 beacons setup and simultaneously running.

2.5 Outputs

The path information will be relayed back to the user using vibration generated from the vibration motor. The path information that is communicated will be the next set of steps/directions that the user will need to take to get to their destination. We will encode a series of directions such as left, right, straight, backwards that the vibration motor will output to alert the user.

2.5.1 Vibration Motor

This block serves as the primary output module interfacing with the user. Once the path planning module determines the optimal path from the user's starting point to the destination point, that information is conveyed to the microprocessor's I/O module, which is then relayed to the vibration motor. We envision this module to communicate the path using vibrations to the user. Ideally, we want the user to be notified the next series of steps such as turn left, right, etc. and then alert them when the destination has arrived.

The vibration motor will be responsible for providing navigational feedback to the user. The vibration motor will vibrate constantly at a fixed pattern when the guidance is straight. When a turn (right/left) comes up, the vibration motor will change that pattern until the turn is detected by the system and the motor will go back to vibrating in the straight pattern. The user will be notified before about these different vibrations.

Our initial design began with only one wearable device. We were advised to add an additional device to increase hardware complexity and we decided to pursue that path after some testing with the vibration motor. While working on the vibration motor, we realized the frequency changes, our original design choice for direction output, was not as easy to detect as we would have liked. While testing the vibration motor we felt that we could only notice the frequency change at very low frequencies, around 100 Hz. Adding the second device for the other arm made our direction output much clearer with left side vibration to go left, right side vibration to go right, and both side vibration to go straight.

2.6 PCB/Mechanical Design

2.6.1 Schematic & PCB

Our original schematic and PCB design is shown below. When we decided to add the secondary device we were able to use the same PCB due to the similarities in the two devices. The secondary device was identical, except for the touch sensor and SD card. Please see Figure 7 and 8 in the Appendix for the full schematic and the board layout of pcb design.

2.6.2 Device Casing

To hold our wearable device we 3D printed two pieces. The piece on the left is the housing for the PCB, battery, and other electronics. It contains slots at the bottom of the sides for the elastic band to fit through for the user to wear around their arm. The bottom of the piece has an indented section for the vibration motor to fit into. There are also some corner shelves on the inside for the pcb to sit on and the battery to sit under. The piece on the right is the top of the device holder. It contains two openings for the SD card and bluetooth module to fit into. These openings are important to make the casing size and weight as small as possible, as well as to improve bluetooth performance because blocking the bluetooth module diminishes our beacon readings. There is also an indent on the top for the touch sensor to fit into. Please see the Figure 9 for the 3D printed casings.

3. Design Verification

It was necessary to have verification procedures documented when debugging, considering the numerous modules in our design. Listed below are our verifications for each of the individual modules and the relevant interfacing between them.

3.1 Processing

3.1.1 Position Determination

Our main requirement for the position determination algorithm was that the algorithm should be able to get a CE85 error of less than 4.0m by using at least 3 beacons in a 6m by 5m room. Since we weren't using trilateration anymore, we tested the system on a harder requirement. In a 7.5m x 14m room with obstacles, and using only 2 beacons, we were able to locate a person within a 4m error 100% of the time and within 85% of the time within 2m. This was done using 800 particles. This is a great benchmark that is set by the Position Determination algorithm in an enclosed space with just two beacons.

3.1.2 Path Planning

The path planning module's requirement was to be able to find the safest (a path not through obstacles and only through specified areas in the walkability map) from any point to any destination granted that such a path exists. Even with the waypoint method that we describe above to lower the computation costs, we were able to meet this requirement. We tested this through writing a unit test which iterated through every possible destination from every point in the environment and then testing if the path went through any obstacles.

3.1.3 Beacon Signal Interpretation

Due to the change in the localization algorithm, the requirement changed as explained in the previous section. This block didn't have requirement before, and still didn't need.

3.1.4 Input/Output

We didn't have any requirements or verification for this module, however, since we added an extra feature we added another requirement: one bluetooth receiver must be enabled in Slave mode and automatically connect to the other bluetooth receiver which is enabled in Master mode. To verify this module we would send a digit such as '1' through the slave mode receiver, we should see it appear on the terminal of the master mode receiver. This verification was successful.

3.1.5 Microprocessor Integration

Our microprocessor requirements revolved around its interfacing with the Bluetooth receivers, vibration motors, touch sensor, and SD card. We verified successful communication between the ATMega328P and each of these individual components in separate manners.

For the Bluetooth receiver, we first ensured that the Slave receiver was working through Arduino before uploading that code onto the microprocessor. With the Master receiver also functioning through Arduino, we went ahead and ran the code on the Master chip to see if it would successfully pair with the code running through the microcontroller. Through LED setups on the breadboard, we would have a green LED light up if the Slave receiver successfully got the command we passed from the Master receiver.

For the vibration motors and touch sensor interfacing with the microcontroller, they were a relatively simple setup. Using simple C code that we uploaded onto the microcontroller, we checked for binary inputs with the touch sensor and further verified by having an LED light up every time a touch was registered. Similarly, we used our functioning Bluetooth and touch sensor programming to verify the vibration motors. After registering a touch, the Master receiver would transmit that signal to the Slave receiver, which would then result in starting up the vibration motor at a predetermined duty cycle.

For the SD card, we put a sample text file on the card and then read from it using Arduino's Serial Monitor in order to ensure that our written program was functioning properly.

3.2 Power

3.2.1 Lithium Ion Battery

No verifications.

3.2.1 Power Switch

Our requirements for the power module were that the system could be completely turned on/off. To do this we used a push switch that either connected or disconnected the circuit. This was placed after the battery to ensure that no power went to the system when the switch was off.

3.3 Memory

3.3.1 SD Card

No Verifications

3.4 Inputs

3.4.1 Touch Sensor

The original requirement we had for the touch sensor was to ensure that the touch sensor wasn't overheating by drawing too much current. With our setup being powered by the 3.7 V battery, the touch sensor performed as expected. We added a resistor just to ensure that this requirement would be met.

3.4.2 Bluetooth Receiver

The first requirement for the Bluetooth receiver was that the range should be greater than 20 meters. We verified this requirement with our initial Bluetooth characterization of the apartment. Figure 1 from the Appendix shows that the first beacon was placed at door, as soon as the user walked in. Since the apartment's size was roughly 27 x 14 meters, we tested that when we took the HM-19 chip to the other end, we were still receiving RSSI values from that beacon by the door. Clearly, the range was met as it exceeded 20 meters, as it can be seen in that same figure. This very same verification applied for the requirement of communicating with a beacon at least 5 m away in the test environment, since it was recognizing and connecting with beacons from more than 20 meters away.

Next, we added a new requirement stating that two HM-19 modules must be able to communicate with each other. In order to verify this requirement, we confirmed the Master-Slave pairing by sending a message from one Serial Monitor and saw that it was received on a separate Serial Monitor.

3.4.3 Beacons

Our documented beacon requirements require that we use either 4.0 or 5.0 module beacons along with having the capacity to send signals up to at least 20 meters. Through our characterization within the apartment, it is evident that both of these requirements are met. We only used the HM-19 since it is a 5.0 BLE module, which receives data from the 5.0 beacons we set up across the apartment.

3.5 Outputs

3.5.1 Vibration Motor

Our initial requirement for our vibration motor was that it be sensitive enough to output vibrational frequencies between 120-240 Hz. After setting up a transistor circuit to control motor output and using a signal generator to output a square wave signal in a range of frequencies, we saw our vibration motor was most sensitive around 100 Hz. It didn't satisfy our initial requirement, but since we added an additional device and changed our direction vibration scheme we didn't need the requirement anymore. Instead we just had to be able to tell if the motor was vibrating or not, which we could.

3.6 External Computer

3.6.1 Operating Area Map

The first requirement for the operating area map was for their to be at least one pathway, given that a path exists, connecting two seperate locations, and that the bluetooth beacons be placed such that the entire area was within range. We verified this by recording beacon data from every walkable spot in the apartment and also simulating path planning from a variety of locations to destinations (Figure 10 shows an example path planning output).

Our second requirement was to have at least 2 layers to our map. We have multiple layers in our final device, one for the walkability and an additional beacon RSSI values map for each beacon in the apartment. This structure also ensures that the map meets the IEEE-RAS/MDR standards.

4. Costs & Schedule

4.1 Labor

Our fixed development costs are estimated to be $45/hr, 10 hours/week for the three people in our group. During the 16 weeks we have available this semester, we are considering costs for approximately 75% of our project. This doesn't include customizing our final wearable product and establishing partnerships with malls, airports, etc.

$$\text{Employee Wage: } 3 * \frac{\$45}{hr} * \frac{10\ hr}{week} * \frac{16\ weeks}{0.75} * 2.5 = \$72{,}000 \quad (5)$$

4.2 Parts

Table 2 Parts Costs

Part	Manufacturer	Retail Cost ($)	Bulk Purchase Cost ($)	Actual Cost ($)
Microcontroller (ATMega328P)	Atmel	2.14	1.78	2.14
iBeacon (x4)	Feasycom	21.99*4=87.96	21.99*4=87.96	21.99*4=87.96
Capacitive Touch Breakout	Sparkfun	6.95	5.91	6.95

Board				
Vibration Motor (x2)	Seeed Technology Co.	1.44*2 = 2.88	1.44*2 = 2.88	1.44*2 = 2.88
3.7 Volt Li-Ion Battery 1500mAH (x2)	Uxcell	7.02*2=14.04	7.02*2=14.04	7.02*2=14.04
4 GB MicroSD Card	SanDisk	10.63	7.66	10.63
Micro SD Card Adapter	SanDisk	3.50	3.50	3.50
HM-19 Bluetooth 5.0 (x2)	DSD Tech	9.99*2=19.98	9.99*2=19.98	9.99*2=19.98
Push Button Latching Switch (x2)	Wealth Metal	0.08*2=0.16	0.08*2=0.16	0.08*2=0.16
PCB	PCBWay	3.10	0.10	3.10
Total		151.34	143.97	151.34

Our clear majority of costs comes from the iBeacons that we plan on purchasing to implement our product. With a 500 meter range in free area, we determine that the cost is worth the performance. Currently, we would be purchasing from Amazon; however, as we plan on moving to bulk manufacturing, we plan to discuss with the seller (Feasycom) about cheaper prices in bulk.

4.3 Schedule

Week	Saleh	Kush	Akhil
2/18/19	Buy Parts & Integration of Bluetooth Receiver Module with Microprocessor	Buy Parts & Work on Map Creation Software Module	Buy Parts & Integration of Touch Sensor Module with Microprocessor
2/25/19	Integration of Vibration Motor Module with Microprocessor	Work on Map Creation Software Module	Integration of Bluetooth Receiver Module
3/4/19	Work on Input/Output Software Module	Develop Position Determination Module	Work on Input/Output Software Module
3/11/19	Develop/Test Beacon Signal Interpretation Module	Develop Position Determination Module	Work on Memory Module

3/25/19	Integration of Power Module	Develop Position Determination Module	Integration of Power Module
4/1/19	Determine Device Case and Packaging	Finish Position Determination Module	Integrate/Test Path Planning Module
4/8/19	Work on Integration of Software	Work on Integration of Software	Work on Integration of Software
4/15/19	Demo Preparation	Demo Preparation	Demo Preparation
4/22/19	Present Demo/Prepare for Presentation	Present Demo/Prepare for Presentation	Present Demo/Prepare for Presentation
4/29/19	Deliver Presentation	Deliver Presentation	Deliver Presentation

Table 3: Schedule

5. Conclusion

5.1 Accomplishments

One of the primary successes in our project was the Bluetooth communication between the two HM-19 modules. This was an essential component for the proper functioning of our overall design, as we keep track of the beacons and relay commands from the "Master" receiver to the "Slave" receiver through this Bluetooth communication. Having this module functioning as intended meant we can properly take input through the touch sensor and then provide the correct feedback to the user through the vibration motors. Understanding the HM documentation and learning more about AT commands significantly increased our proficiency in Bluetooth communication over the course of the project. Spending so much time debugging Master-Slave connection issues helped speed up debugging, as we finally had the opportunity to present the working connection. Another accomplishment was the successful path planning module. After characterizing the apartment, we were successfully able to implement a Breadth First Search to find the shortest path between any point of the apartment to any of the four given destinations. This path was linearized by creating hubs across the apartment, so that the number of turns a user had to take was minimized.

Additionally, the localization process was definitely the most challenging aspect of our project. From starting at trilateration to considering many other techniques, we faced a good amount of setbacks in this module. Eventually, after weeks of work, we were able to see the Particle Filter implementation accurately implement localization in Python. Although, memory computation issues cropped up when moving it to the processor, having it functioning was a major accomplishment since it took numerous testing and characterization attempts. Furthermore, we pride ourself on creating a wearable footprint of the entire design. With so many moving components, it was a major challenge to condense each of these

pieces to fit on a wrist. This lead to further soldering and PCB challenges, which we overcame successfully over the course of the semester.

5.2 Uncertainties

As mentioned above, we definitely faced severe memory constraints when moving the localization module to the microprocessor. Unfortunately, the localization module required immense amounts of dynamic memory, which the ATMega328P could not support. This is definitely an area of concern as navigation cannot occur successfully without proper localization. A possible fix for this solution would be to use a much stronger microprocessor, in order to avoid memory computation errors. A quantitative example can be seen with path planning in Arduino. Declaring one instance of the 27x14 binary map of the apartment as a global variable caused 71% of the dynamic memory to be used, which was enough to cause stability issues within the system. Another issue that came up dealt with hardware reliability. Late into our project, we started dealing with our HM-19 receivers randomly going into an offline mode, where they were only transmitting data, but not receiving data. This became a huge problem and took away a lot of time when trying to debug, as we eventually realized that it was an internal issue. Eventually, one of the Bluetooth receivers started receiving data again, before going back offline. Considering how important Bluetooth is to our project, we need to conduct more research into the Bluetooth chips that we use later on. Lastly, another issue we sometimes faced was external Bluetooth signal interference. When trying to pair the Master with the Slave Bluetooth module, we sometimes dealt with situations in which the Master would automatically connect with other Bluetooth devices, such as headphones, within a 5 meter radius. As we scale this to bigger public spaces, this is an area of concern as we need to ensure that the Master and Slave pair together. A possible solution would be to conduct more parsing when scanning devices, in order to ensure that only HM-19 chips are showing up.

5.3 Future work/Alternatives

An immediate alternative would be to change the microprocessor into something that is more powerful and can handle a more complex localization algorithm. Now that there are two wristbands, we can potentially separate the computations between the two processors on each of the hands. We can then use the receivers to share results with each other given that the bluetooth communication modules between the two bands is robust.

One of the main drawbacks that we faced when localizing a person is that we estimated the prior by propagating the particles by the how far they had moved multiplied by some factor that was dependent on the elapsed time between the current and last localization call. One way to get a better prior is to use an IMU. However, we would require a lot of filtering since waving arms can be a source of a lot of noise. One of the areas for future work would be to integrate the system with Alexa and have voice feedback.

5.4 Ethical considerations

During the use of our product, one potential concern is the fact that internal surroundings of viable locations could be slightly altered after uploading a blueprint to the system. Therefore, we must ensure that our blueprints are constantly being updated when major internal design changes are being made. Prioritizing this action will allow our users to have full confidence in our product. As seen in the IEEE Code of Ethics, the number one interest is to “...protect the safety and welfare of our users”.

Additionally, we want to update our blueprints when needed as a means of transparency that we're doing our work on the back end. This directly relates to #3 of the IEEE Code of Ethics because we must be honest and realistic with the information we are providing based off the data available to us. Moreover, it is vital that we market our product as a navigation tool which is meant to be utilized in supplement with regularly used walking aids. Industry-standard beacons tend to have a rough 1-2 meter in-accuracy, which would simply corrected by users having their canes/walking dogs/etc.

Additionally, the updated maps should follow the IEEE-RAS/MDR (Robotics & Automation Society/Map Data Representation) society whose goals are to define norms to commonly "represent and encode a map for robotics navigation". This will make our product versatile and our maps can be used throughout the industry and the updates explained above can happen in a systematic manner. We wish to separate our maps into global and local maps and then represent them in a hierarchical manner with clearly defined coordinate spaces, grid and topological maps. Also, users might be worried about their locations and privacy since they are being tracked in public areas such as malls, airports, etc. Our duty is to inform our users that the beacons are only communicating with their wearable sensors and that the data is not being sent elsewhere.

Looking more into the elements of our design, using a Lithium-ion battery will definitely come with its own set of challenges and necessary precautions. We must account for not placing them in extreme temperatures or overheating within the circuit. Similarly, another concern is that none of the electrical components present in the wearable part of the design should get overheated, as that would directly impact our users. Part #6 of the IEEE Code of Ethics applies most, as we have all taken extensive electrical engineering courses and plan to conduct testing during the development in order to ensure that our designs are working as intended.

Aside from the functioning of our product, we also encountered challenges in the form of activities that needed to be completed in the senior design electronics laboratory. We dealt with soldering elements to our boards and using power supplies in order provide power to our product. Each of our group members has taken the Lab Safety certification and was prepared/well-equipped to deal with these challenges.

References

[1] Riazi, Abbas & Riazi, Fatemeh & Yoosfi, Rezvan & Bahmeei, Fatemeh. (2016). Outdoor difficulties experienced by a group of visually impaired Iranian people. Journal of Current Ophthalmology. 28.10.1016/j.joco.2016.04.002.

[2] Kavanagh, A. (2018, July 21). Tactile paving: The secret code that helps me get around. Retrieved from https://www.bbc.com/news/newsbeat-44861568

[3] Scottish Mobility Unit Department of the Environment, Transport and the Regions (1998). Guidance
on the use of Tactile Paving Surfaces.

[4] VisionAware, American Foundation for the blind, Dog Guides for People with Vision Loss. (n.d.). Retrieved from http://www.visionaware.org/info/everyday-living/essential-skills/an-introduction-to-orientation-and-mobility-skills/dog-guides-for-people-with-vision-loss/1234

[5] Faa Naptf. (2017, September 25). Indoor Airport Wayfinding for Blind and Visually Impaired Travelers. Retrieved February 7, 2019, from https://www.airporttech.tc.faa.gov/Download/Airport-Safety-Papers-Publications/Airport-Safety-Detail/ArtMID/3682/ArticleID/20/Indoor-Airport-Wayfinding-for-Blind-and-Visually-Impaired-Travelers

[6] Harris, M. A. (2018, September 22). This Is What It's Like To Train Dogs To Guide Blind And Visually Impaired People. Retrieved from https://www.huffingtonpost.com/entry/training-guide-dogs-blind_us_5ba508c1e4b0181540dc743a

[7] Kulyukin, Vladimir & Kutiyanawala, Aliasgar. (2010). Accessible Shopping Systems for Blind and Visually Impaired Individuals: Design Requirements and the State of the Art. The Open Rehabilitation Journal. 3. 158-168. 10.2174/1874943701003010158.

[8] Saarela, M., Dr. (n.d.). Solving way-finding challenges of a visually impaired person in a shopping mall by strengthening landmarks recognisability with iBeacons. Retrieved from http://blindsquare.com/wp-content/uploads/2015/06/Solving-way.pdf

[9] http://www.blindsquare.com/about/ [Company Website]

[10] https://www.indooratlas.com/ [Company Website]

[11] https://indoo.rs/solution/indoor-positioning-system/ [Company Website]

[12] https://www.wayfindr.net/ [Company Website]

Appendix A: Requirement and Verification Table

Table 4 System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. Power Switch a. If the system is on, pressing the power switch should bring the circuit's current flow to ~0 mA. b. If the system is off, pressing the power switch should register a nonzero current flow across the circuit.	1. Verification a. Probe the circuit by placing an ammeter in series with any of the other elements (vibration motor, touch sensor, etc.) b. If testing for the system to be on, ammeter should detect non zero current i. Otherwise, it should detect a zero current	Y
2. The ATMega328P should be able to communicate with the SD Card, on which we upload the walkability maps. This should be done using the SPI communication protocol interface - the mode should be set to (0,0) and the clock frequency should be set in the range of 100 kHz and 400 kHz prior to initializing the card.	2. Initialize the SPI interface and conduct a transfer of data using a script that will send/receive a byte. a. Test the interface with a loopback: one can wire the MOSI output pin to the MISO input pin, and check that the two bytes match each other.	Y
3. The ATMega328P should be able to take input from the touch sensor. The transmitted signal to the microcontroller should be 1 of two output states: High when equivalent to 3.7 V and Low when equivalent to 0 V.	3. Connect touch sensor to ATMega328P using serial input a. Write basic script that displays "Output detected" every time touch sensor is touched b. Leave device stationary for 10 seconds	
4. The ATMega328P should be able to take input from the Bluetooth receiver. Using a clock frequency	1. Connect Bluetooth Receiver to ATMega328 using the PB1 and PB0 pins of the controller	Y

between 100 kHz and 400 kHz, the baud rates of both the microcontroller and Bluetooth module should be the same (industry standard is 9600bps).	a. Write a demo script that initializes the USART communication and receives data from the Bluetooth module b. For example, a '1' or '2' will be sent from a smartphone to turn an LED on ('1') or off ('2') as a means to test the connection	
5. The ATMega328P should be able to provide output to the vibration motor at different patterns (used to distinguish directions). With a threshold value for moving straight, a higher frequency of rotation will indicate to move right and a lower frequency will indicate moving left.	2. Connect the ATMega328P pins (PC3-PC0) to the inputs of the motor a. Write a demo script that will rotate the motors and provide a vibration that is of noticeable intensity and a different pattern.	Y
6. In an empty medium sized room at least of size 6m X 5m with no interferences between the sender and receiver, but possible interferences with WiFi, other bluetooth devices, pillars etc. and with at least 4 Bluetooth beacons, the position determination algorithm should have CE85 error of 4.0m.	1. Set up a unit test wrapper function around this block of code in the module and set up the appropriate *verification* environment.[1] a. Fix the bluetooth beacons in the room to meet the R&V of the operating Area Map. b. Use the receiver and the implemented position determination algorithm to find the euclidean distance between the calculated vs. actual position. c. Travel through the entire room at-least twice (reset the device before each run) to get at least 50 different data points. d. Calculate the precision.	Y (req. was met wasn't able to transfer well to AtMega as explained)

[1] Please refer to the Appendix for more information on "test" environment and "verification" environments specifications.

7. In the T&E environment the path planning module should be given the current user location, the operating area map, any specified destination from the list of all possible destinations and the model should be able to find the safest path every time.	1. Set up a unit test wrapper function around this block of code in the module and set up the appropriate T&E environment. a. From a set of 5 random locations in the T&E environment, iterate through every possible available destination, plotting the output path into a grid map over the walkability map. The output path should be safe i.e. doesn't navigate the user through static obstacles such as the walls, a table specified on the map, etc.	Y
8. The T&E environment map should have at least one pathway, assuming a path exists, connecting two separate locations and the bluetooth beacons should be placed to have this entire area in range.	1. Set up a unit test which prints the packet that the receiver receives with the corresponding device. a. Walk through the operating area and check the receiver output from the unit test: if we receive a signal from at-least one beacon at all possible walkable locations then the requirement is verified.	Y
9. The map should have at least 2 different layers: walkability map and a beacon map.	1. Display the beacon map into a grid array and check if that layer only includes beacon positions. a. Display the walkability map into a grid array and check if the physical space corresponds to the map.	Y
10. Touch sensor should only be able to draw at most 5V since, we don't want to draw too much voltage and overheat the touch sensor. Attach a voltmeter in parallel and read the voltage reading across the touch sensor	1. Attach a voltmeter in parallel and read the voltage reading across the touch sensor to ensure that the voltage draw is within the parameter	Y

to ensure that the voltage draw is within the parameter		
11. The range should be at least greater than 20 meters (radius) in an indoor free, unobstructed, space environment.	1. Place a beacon in an indoor free space environment a. Walk to 20m and check the signal strength. In unobstructed space, the bluetooth receiver should detect a signal.	Y
12. One bluetooth receiver must be enabled in Slave mode and automatically connect to the other bluetooth receiver which is enabled in Master mode	1. Send a digit such as '1' through the slave mode receiver, we should see it appear on the terminal of the master mode receiver.	Y (shown to be working at the Mock Demo independently)
13. Bluetooth receiver should be able to communicate with a beacon at least 5m away in the T&E environment	1. Place a beacon in a sample T&E environment. a. Walk to in a radius 5m from the beacon. The bluetooth receiver should detect the beacon signal throughout the walk.	Y
14. Beacons should be based on bluetooth 4.0 and 5.0 module and Advertising packets must comply with the Eddystone-UID/URL or the iBeacon Format	1. We have to buy the product with those specifications. a. Using the receiver, send and receive packets from different beacons and check if the format is the same.	Y
15. It should be a BLE (bluetooth low energy) beacon that sends out signals in a radius upto at least 20 meters in our desired location.	1. Test the strength using the receiver of each beacon by placing it in the location it will be actually placed in a. Test the strength upto to an area of 20 meters.	Y
16. Our system's coverage of the T&E area should 11 be 90%.	1. Hold a receiver and walk around the entire area to check if the receiver detects at least one beacon at every location.	Y

17. Should be sensitive enough to distinguish between frequencies	1. Connect the vibration motor to a function generator and oscilloscope. a. Sweep the generator frequency through frequencies to ensure they are noticeable to user.	Y

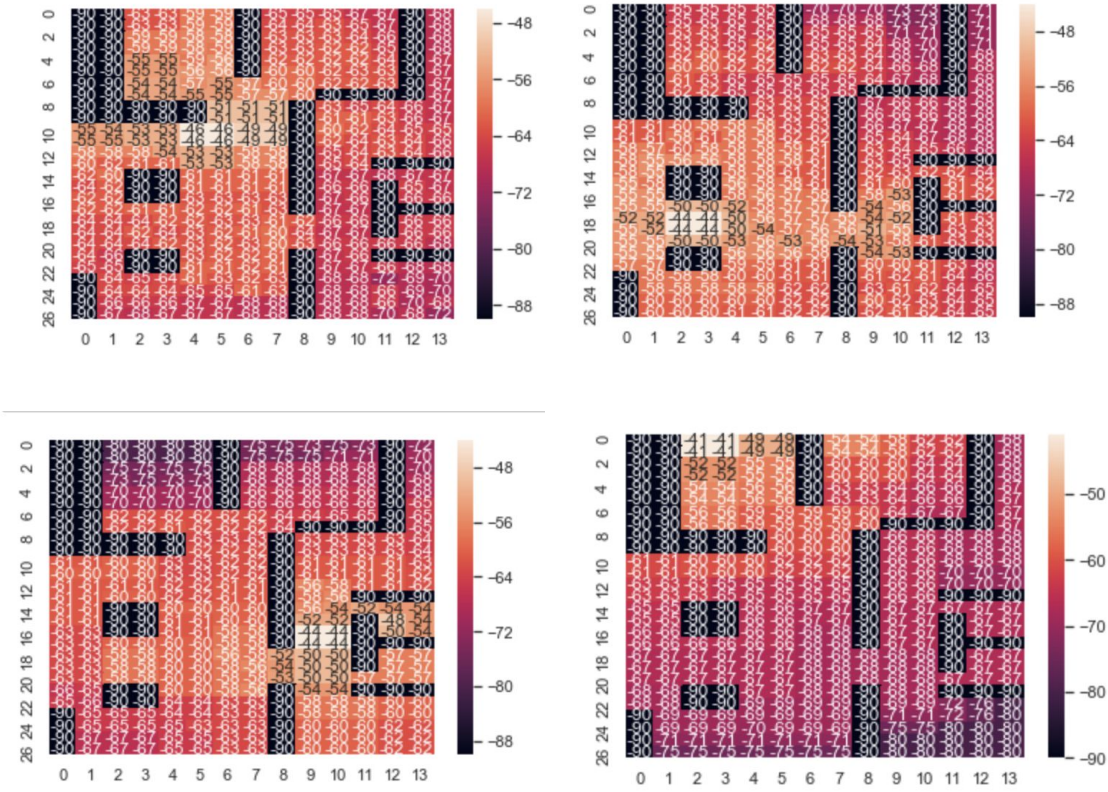

Figure 4 showing the environment characterizations

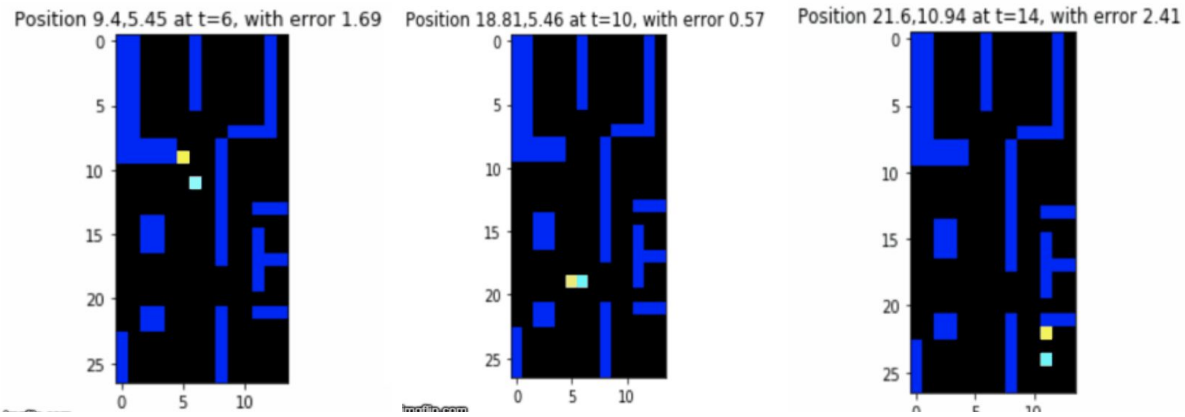

Figure 5 showing the particle filter outputs as it tracks the user (python implementation with 800 particles)

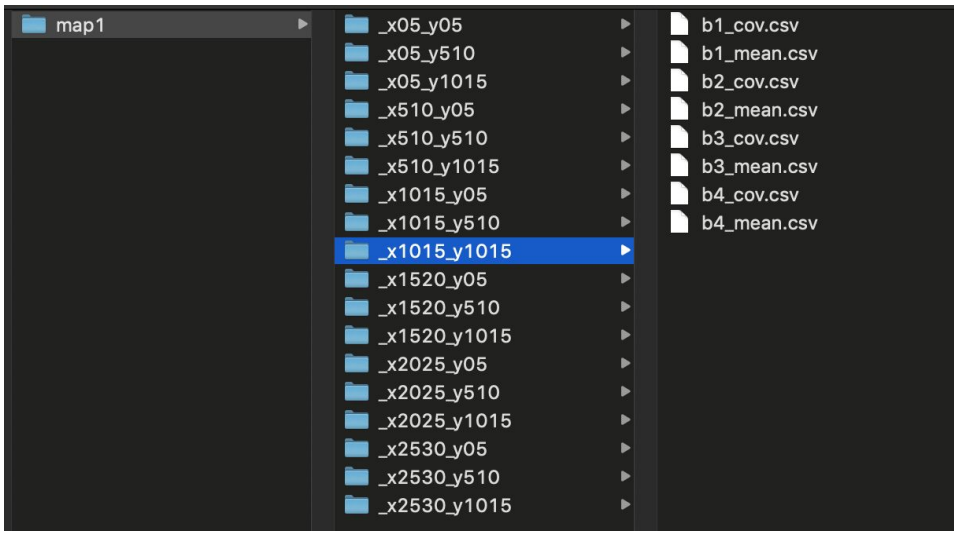

Figure 6 showing how the discretized GP Model's mean and variance outputs are stored in memory to make the localization module faster

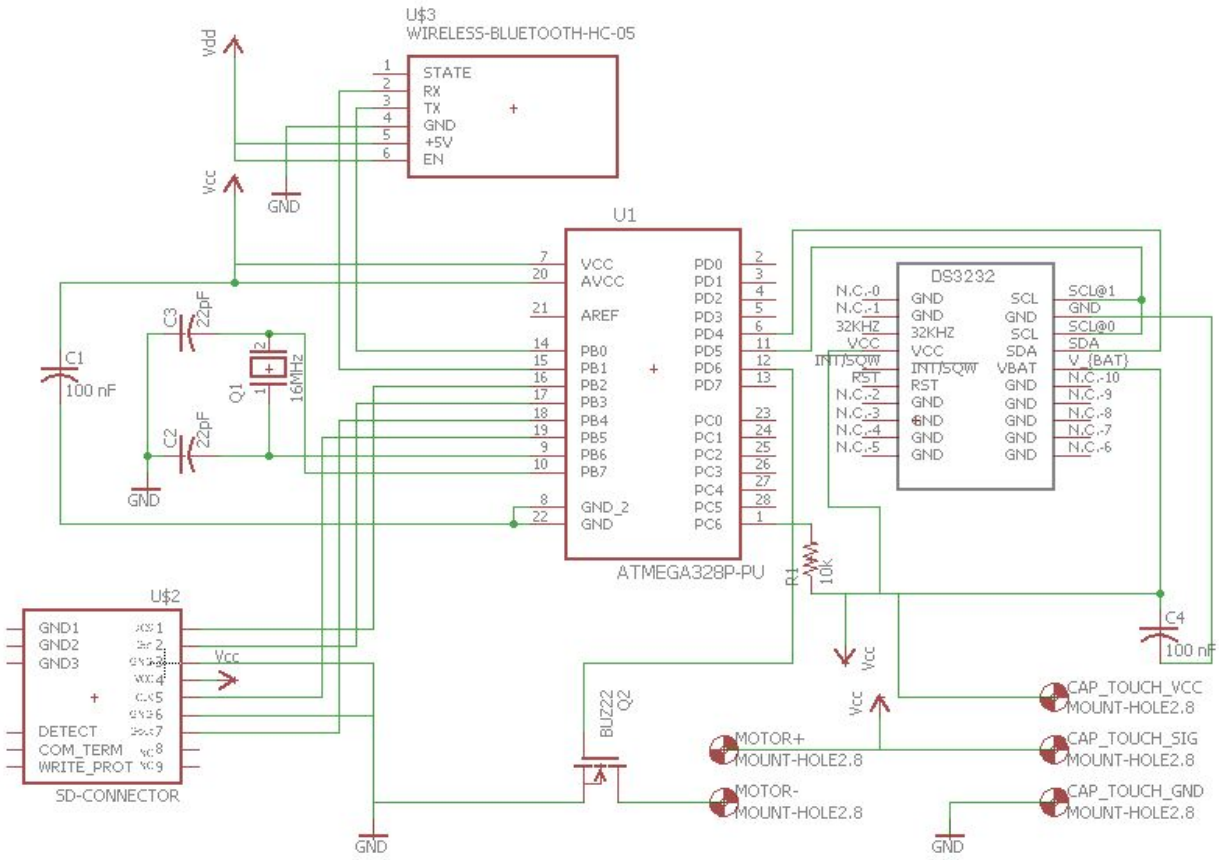

Figure 7 showing the schematic of the PCB

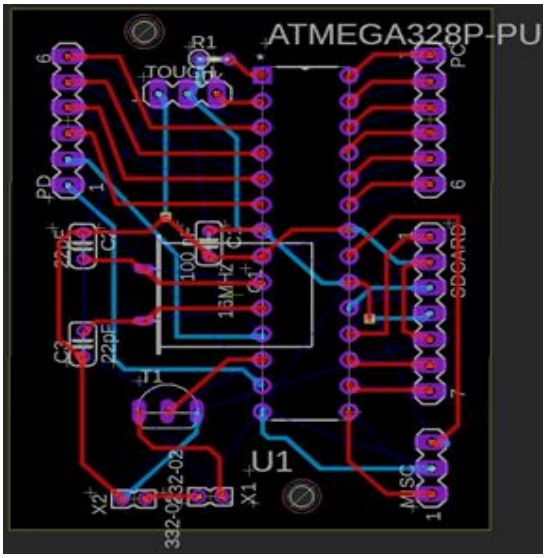

Figure 8 showing the board layout of pcb design

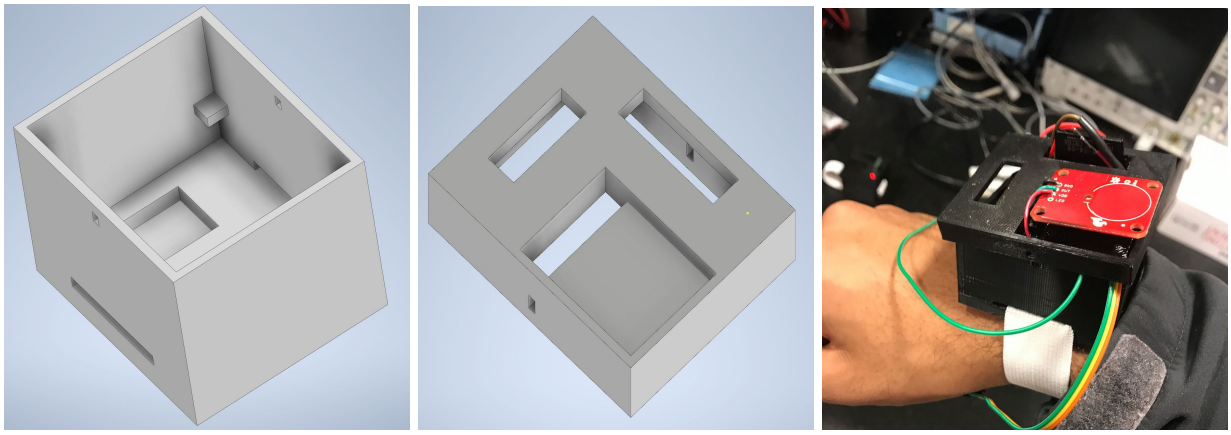

Figure 9 showing the 3D printed design of the wearable device

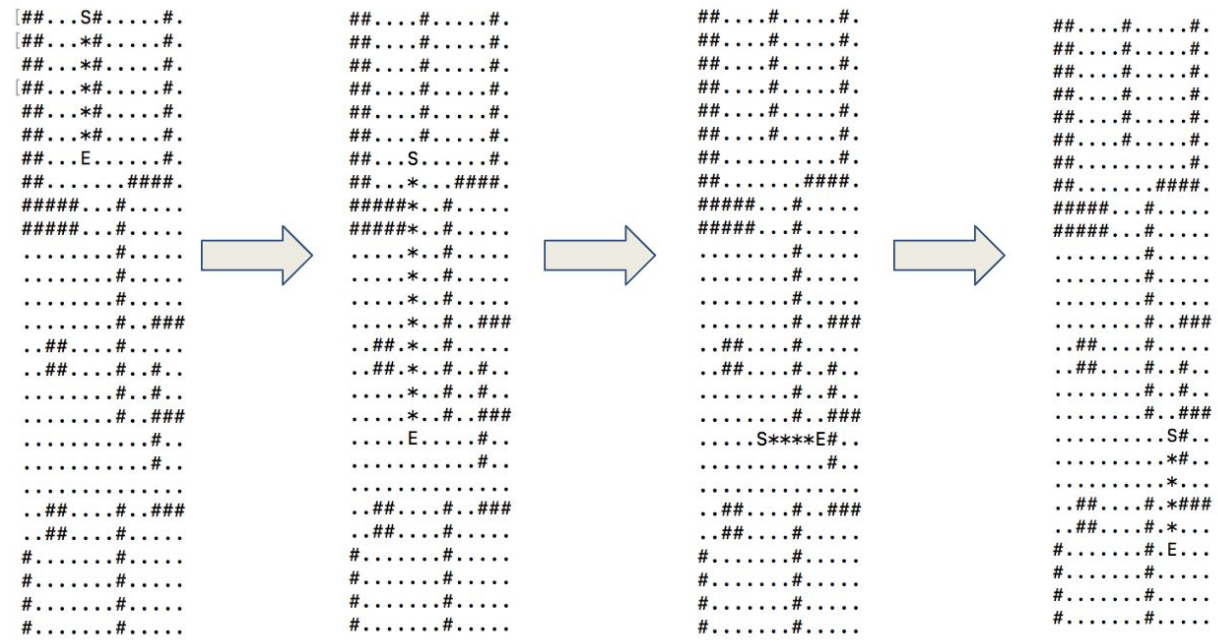

Figure 10 showing the output our path planning module. The * represents the immediate path and the output action by the vibration motor