Redefining Personal Transport with Wearable Haptic Navigation

By

Kevin Xia - kevinx3

Soo Min Kimm - skimm2

Stephen Battin - sbattin2

Final Report for ECE 445, Senior Design, Spring 2021

TA: Andrew Chen

05 May 2021

Project No. 41

Abstract

The goal of our project was to create a better way for cyclists and skateboarders to navigate while keeping them distraction free. We went about this by creating a vibration-based navigation system. This system allows for users to enter a destination into a companion app which will be paired with an arm-mounted device. This device will give haptic feedback so that the user knows where and when to turn.

Contents

1 Introduction

Although biking and skateboarding around cities is not a brand new concept, new technological advancements in recent years along with the dangers that COVID-19 poses to those who take public transportation have led to the rise in this form of transportation [1]. This fact is extremely noticeable in cities and college campuses where the majority of the community would choose to bike, skateboard, or walk instead of driving or taking the bus with the inclusion of new bike and scooter ride-share programs. The convenience of not having to find a parking spot or needing to worry about getting sick is extremely enticing and the result is an environment with high levels of walking and biking in conjunction with high levels of vehicle traffic. However, this new increase in traffic comes with major drawbacks as well. With smartphones being heavily relied upon for children and young adults, more than $20\,\%$ of cycling accidents in 2015 were caused due to the phone use [2].

To eliminate the smartphone distraction when using our wearable, we are developing technology to replicate the same functionality of the turn-by-turn navigation from a smartphone on our wearable. This system will allow a user to "feel" which direction the user will need to take by giving vibrational feedback on the left and right sides of the user's arm. By varying the intensity, duration, and position of the vibration as the user approaches a turn, we can easily communicate to the user about upcoming turns, whether they be left or right, without the need for visual or audible feedback. This solution will allow the user to safely get to the desired destination and completely remove the phone-to-user interaction that previously would have needed to exist.

To accommodate outdoor athletes, we are developing a battery system that will ensure plenty of battery life while still maintaining the comfort and convenience that people have grown to expect in modern society. With the inclusion of solar panels on our wearable, the battery life can far exceed a single charge and will be able to extend the battery life far past the hour and a half it is designed for, given adequate lighting conditions. This will increase the flexibility of where our product could be used and be able to adapt to all potential use cases. For this device to be successful, we will need a power supply unit, input devices, control unit, and output devices as shown in Figure 1.

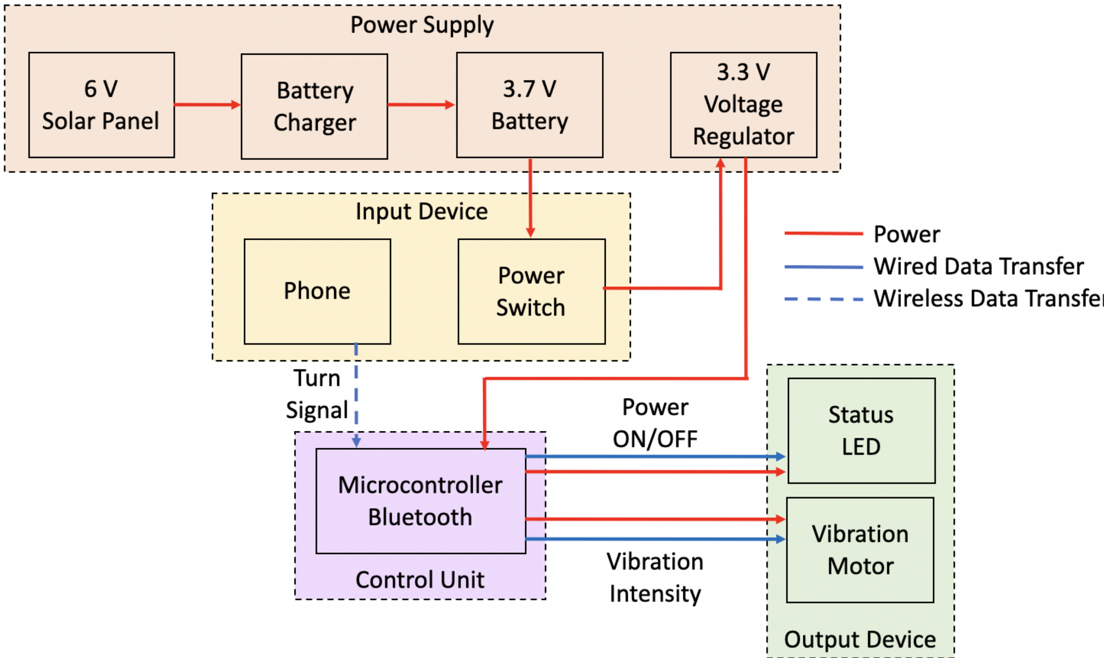


Figure 1: Block Diagram

2 Design Verification

2.1 Power Supply

2.1.1 Solar Panel

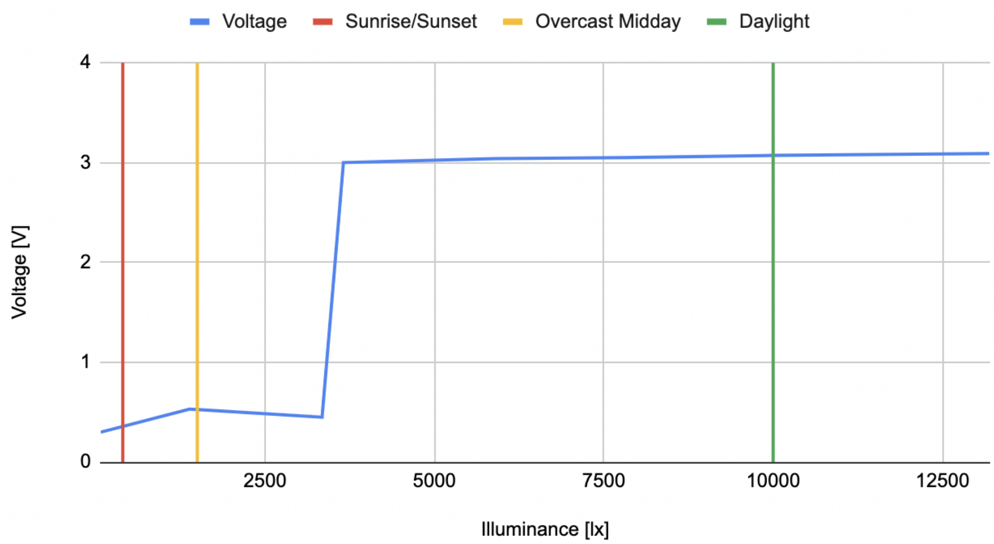


Figure 2: Voltage of Solar Panel under Different Illuminance

Figure 2 shows the output voltage of the solar panel when the intensity of light changes. The measurement is made at the battery charger, not directly from the solar panel to check the minimum voltage required to charge the battery from the solar panel. The battery will start charging if the illuminance is higher than $3500\,\mathrm{lx}$, which is the sunlight from a regular day with a bit of cloud. If the device is used outside while the solar panel is connected, the battery will be charged from the solar panel, and discharged to power the device at the same time. This will enhance the battery life of the device such that the user can use it for a longer time.

The solar panel, when measured separately from the battery charger, can generate a reasonable amount of electricity even if the light intensity is quite small. The voltage quickly goes above $0.2\,\mathrm{V}$. The regulation of the voltage and current can be achieved through the battery charger. Under the full sunlight, the voltage of the solar panel goes above $5.0\,\mathrm{V}$, but after passing through the voltage regulator, it is fixed to $3.0\,\mathrm{V}$, which indicates that the solar panel will be properly regulated. We choose to use a solar panel in this project to increase the run time that the device would have. The solar panel allows the user to charge while they are en route and or once they have arrived.

2.1.2 Battery Charger

The battery charger allows the user to charge the battery either from the wall or the solar panel. Minimum output voltage from the battery charger is set to $3.0\,\mathrm{V}$ and there will be no charging or discharging of the device if the voltage is lower than that. If charged from the wall, the Li-Ion battery with $2500\,\mathrm{mAh}$ will be fully-charged in 2 hours. Considering the solar panel is generating stable electricity, the charging may take up to 4 hours.

The battery charger we are using already has a feature to prevent overcharging of the battery. When using a chip, it is possible to connect the thermistor with the battery and the chip to check if it stops.

2.1.3 Battery

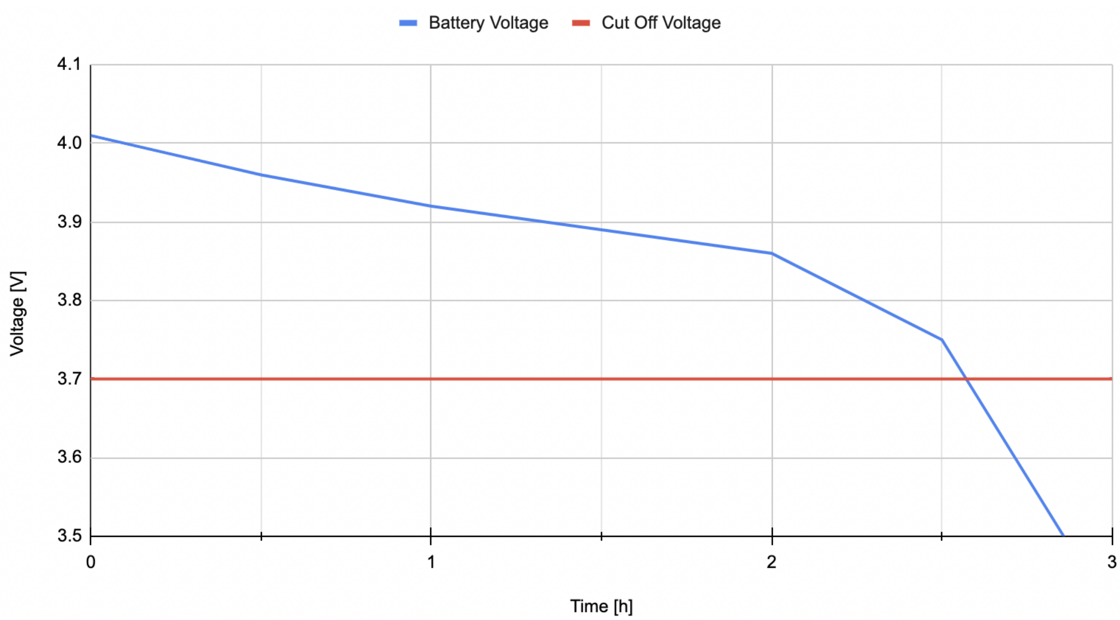


Figure 3: Discharging of the Battery when the Device is Running

Figure 3 indicates the changes in voltage from the battery over time. While the fully-charged battery can produce $4.1\,\mathrm{V}$, the voltage decreases as the device is utilized. The Bluetooth is connected with the phone application that is navigating the direction and sending the signals to the microcontroller to activate vibration motors. Under this condition, the device can operate for 2.5 hours before needing a charge. Since the use of vibration motors highly effects the power consumption, the battery life can vary depending on usage.

2.1.4 Voltage Regulator

V_{in}[V]	V_{out}[V]
3.0	1.8
3.3	1.8
4.1	1.8

(a) TPS72518 (Buck Converter)

V_{in}[V]	V_{out}[V]
1	3.3
1.8	3.3
2.5	3.3

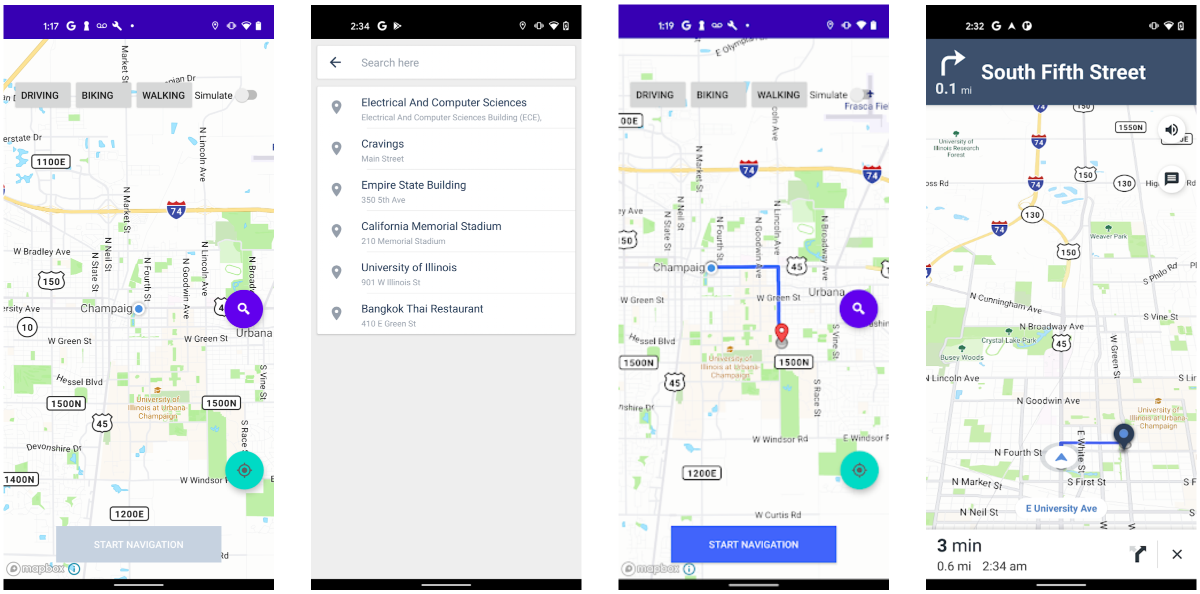
(b) TPS61322 (Boost Converter)

Table 1: Response of the Voltage Regulators

The ESP32 needs $3.0\,\mathrm{V}$ - $3.5\,\mathrm{V}$ as the supply voltage. The voltage of the battery varies from $3.7\,\mathrm{V}$ to $4.1\,\mathrm{V}$. Hence, our primary goal is to set the voltage to $3.3\,\mathrm{V}$. Table 1 gives the test result of the voltage regulators we use. We use two voltage regulators, which can drop the voltage to $1.8\,\mathrm{V}$ then boost back to $3.3\,\mathrm{V}$. Voltage regulators have dropout voltage, which is the voltage fed to assure certain output voltage. Therefore, we could not use the buck converter that drops the voltage directly to $3.3\,\mathrm{V}$ since $V_{in} \geq V_{out} + V_{dropout}$ and the input voltage of our device is lower. It is possible to use one voltage regulator which has both buck and boost converter features; however, the component with the specific requirements is very small and hard to test as well as solder. That being said, switching to a single voltage regulator would reduce costs and complexity for future development.

2.2 Input Device

2.2.1 Phone Application



(a) Main Menu (b) Search Menu (c) Route Generation (d) Navigation

Figure 4: Application Design

Figure 4a shows the main screen when the app is first launched. This is where the user is able to select from several on screen controls as well as move the map around. The user is also able to click on the map to add a marker for navigation. Figure 4b shows the search field. This allows the user to enter any address as well as search for any address. Once the address is found, the user is able to select the address and a marker will be added for that location.

Figure 4c shows once a destination has been selected. The app will then go and generate a route to the desired location. Once this route is generated, the user is able to click the ”START NAVIGATION” button on the bottom of the display to begin the navigation. Finally, Figure 4d shows once the navigation has begun. This will display the actual map as well as the route highlighted for the user. The user will also be able to see the estimated arrival time, distance to destination, as well as the next turn instruction. Once the navigation has begun, the phone will begin to transmit via Bluetooth to the ESP32 to provide directions.

2.2.2 Switch

The switch allows the user to turn the device on and off. It was connected in between the battery charger and the voltage regulator to avoid draining the battery when not in use.

2.3 Control

2.3.1 Microcontroller

There are several microcontrollers that we could have used for our project. Since our project requires a Bluetooth module to operate, we choose to use the ESP32 due to its native support for Bluetooth. The ESP32 also has development boards that were used to facilitate our debugging process. Using an ATmega328, the microcontroller used in Arduino Uno, was also an option. Although, using this could would have required us to get an external Bluetooth module. This in turn would have added complexity to our project in terms of both the circuit required as well as the software written.

Each pin of the ESP32 can provide a maximum of $40\,\mathrm{mA}$, which is enough to power the vibration motors. Testing was first done with LEDs to give a visual representation of what was going on. After verifying a working product, we switched the LEDs out for the vibration motors. For the demonstration, we used pin 25 and pin 26. The microcontroller successfully interpreted the signal obtained via Bluetooth to turn on the vibration motors.

2.4 Output Device

2.4.1 LED

LEDs indicate if the device is turned on. Our device contained 3 total LEDs. Two LEDs on the battery charger told us whether the battery was charging and discharging properly. The thrid LED would have been implemented on the PCB. Since we did not have the PCB, we utilized the imbedded LED on the ESP32 development board. This LED was used to indicate whether the system was turned on or off.

2.4.2 Vibration Motor

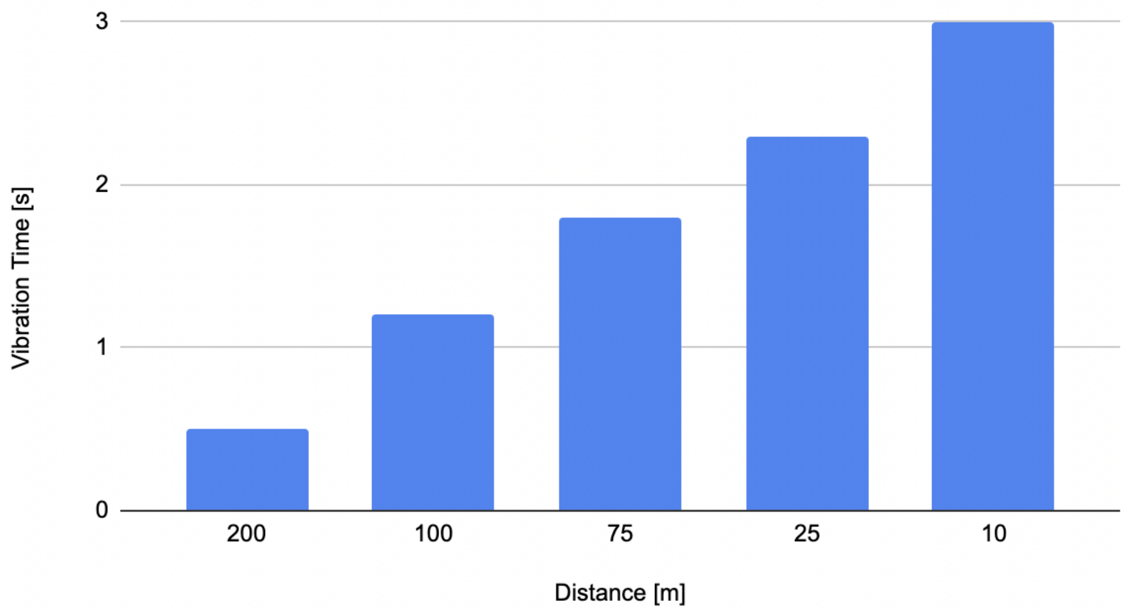


Figure 5: Response of the Vibration Motor at Different Distances

Figure 5 shows how long the vibration motors vibrate based on their distance from a turn. Each pin of the ESP32 can provide a maximum of $40\,\mathrm{mA}$. If the current through the vibration motors was lower than $30\,\mathrm{mA}$, the user would struggle to detect the vibration. Due to this, we choose to vary the time that the motors were active rather then changin the current provided to the motors. This gave us a much more reliable vibration.

The location of the vibration motors was moved as well. Originally, both vibration motors were located on either side of the user's wrist. This was changed to one motor on the wrist and the other on the upper arm. This allows the user to more easily distinguish between the two motors.

Vibration motors vibrate continuously when the user arrives at the destination. The motors continue to vibrate until the user turns off the device. When the user turns the device off, the vibration motor stops immediately. This feature was implemented to serve as a reminder to the user to turn the device off to avoid wasting battery.

3 Costs

3.1 Labor Cost

According to UIUC, the average starting salaries for graduates in Computer Engineering and Electrical Engineering are $\$96,992$ and $\$79,714$ respectively [3]. Hence, the average salary of the ECE graduate would be $\frac{\$96{,}992+\$79{,}714}{2} = \$88,353$. Assuming that the average engineer works 40 hours per week for 50 weeks per year, one would be making $\frac{\$88{,}353}{40 \cdot 50} = \$44.18$ per hour. One person will spend approximately 100 hours on the project; hence, the total cost of labor would be $\$44.18 \cdot 100$ hours $\cdot\ 3$ people $\cdot\ 2.5 = \$33,135.00$.

3.2 Component Cost

Part	Seller	Link	Quantity	Unit Price ($)	Total ($)
ESP32-WROOM Development Board	Amazon	Part	2	7.5	15
ESP32-WROOM Microcontroller	Adafruit	Part	1	8.95	8.95
CP2102 USB-UART TTL Module	Amazon	Part	2	3.25	6.5
SWD JTAG	Adafruit	Part	1	4.9	4.9
Solar Panel	Adafruit	Part	1	45	45
BQ24074 USB/DC-Battery Charger	Adafruit	Part	1	9.95	9.95
Li-Ion Battery	Adafruit	Part	1	14.95	14.95
IC REG LINEAR 1.8V 1A SOT223-6	Digi-Key	Part	1	3.65	3.65
IC REG BOOST 3.3V 1.6A SOT23-5	Digi-Key	Part	1	0.64	0.64
KAE01SGGT Switch	Digi-Key	Part	1	1	1
APT1608ZGC LED	Digi-Key	Part	2	0.75	1.5
1020-15-003-011 Vibration Motor	Digi-Key	Part	4	1.2	4.8
Arm Sleeve	Amazon	Part	1	7.64	7.64

Table 2: Component Cost

3.3 Total Cost

Section	Cost
Labor Cost	\$33,135.00
Part Cost	\$124.48
Total Cost	\$33,259.48

Table 3: Total Cost

4 Conclusion

4.1 Summary

In the end, our project was able to produce a fully functional device along with its accompanying app. Despite the final solution not containing a PCB due to the difficulties in shipping, we were able to fully unit test all aspects of the project to ensure functionality. Our app was able to find and calculate a route to a destination as well as transmit instructions via Bluetooth to our device. The device was able to receive, interpret, and act upon these signals to direct the user to their desired destination. This was able to be done consistently and reliably enough to where the user can navigate solely using the device. This ultimately is a very functional solution for those looking to ride their bike or skateboard to a new destination distraction free.

4.2 Ethics

In our final product, we will have circuits in our wearable, so it is necessary for us to address some of the safety concerns that might arise to protect our customers as well as our product itself. During use, we will have a battery and solar panel attached to our product. One safety concern could be the possibility of these components overheating and without proper cooling, we could run into trouble of burning the customer. Therefore, to mitigate this we will have an automatic shut off sequence in place where if the unit gets too hot, the product will automatically shut off to ensure no one is injured while using this product. This specifically will address IEEE ethic code II.9 [4].

Aside from the power units, we also will have vibration motors attached to the product to give haptic feedback to the users. However, this could also potentially be a safety concern as vibrations that are too intense could easily cause injuries or massive discomfort. Therefore, we will ensure to test our product intensively in order to ensure the maximum vibrations will not be too intense and also give users an option in our mobile app to let users manually adjust intensities as well.

We will test our final design intensively and ensure to prevent any potential safety hazards and use of dangerous components. We will also abide by IEEE ethic code I.5, by taking into consideration the feedback from others and using them to improve our design safety, we are certain we can produce a product both safe and functional [4]. Lastly, we will treat all of our members and testers with respect and never engage in any harassment or discrimination of others.

4.3 Future Work

Our group has several plans for future work on this project. This future work can be broken into two main categories. Those being system optimization as well as a system redesign. For the system optimizations, we would first like to implement Bluetooth Low Energy to allow for a reduction in battery consumption from the device. This would increase the battery by allowing the device to go into a sleep mode and eliminate the need to be constantly running. Another optimization would be in the software itself. We can optimize this to allow for background navigation as well as optimized signal transmissions. Meaning this would further increase our battery life.

The other area that our design could improve from optimizations would be the physical design itself. Here, we would start by trying to reduce the size of the PCB. With this reduction in PCB size, we would be able to move our design from a whole arm type device to more of a bracelet type device as shown in Figure 6. To further facilitate this transition, a flexible solar panel would be ideal. This would give us the flexibility to shape the solar panel to the user's arm to increase comfort.

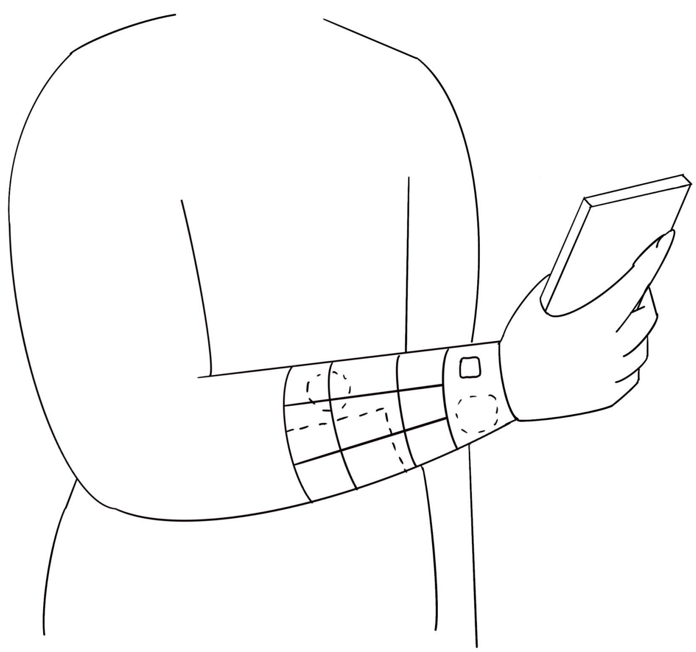


Figure 6: Potential Future Design

References

[1] Winnie Hu. (2020). "A Surge in Biking to Avoid Crowded Trains in N.Y.C.," [Online]. Available: https://www.nytimes.com/2020/03/14/nyregion/coronavirus-nyc-bike-commute.html. [Accessed: 4-Mar-2021].

[2] H. Staples. (2013). "Bike Accidents and Phone use - Is There a Link?" [Online]. Available: https://www.holland-cycling.com/blog/291-bike-accidents-and-phone-use-is-there-a-link#:~:text=The\%20group\%20of\%20cyclists\%20most,into\%20a\%20tree\%20or\%20lamppost. [Accessed: 02-Mar-2021].

[3] UIUC. (2019). "Salary Averages," [Online]. Available: https://ece.illinois.edu/admissions/why-ece/salary-averages. [Accessed: 28-Feb-2021].

[4] Banggood. (2020). "The Secret Marketing Trick behind Powerbank Capacity," [Online]. Available: https://blog.banggood.com/the-secret-marketing-trick-behind-powerbank-capacity-29982.html. [Accessed: 4-Mar-2021].

Appendix A

- **Solar Panel**

Requirement	Verification	Status
1. Maintain the output over $0.1\,\mathrm{A}$ and $0.2\,\mathrm{V}$ which are the minimum requirement for the battery charger.	1. Set up the testing environment, assuming that the solar panel is under the full sunlight, by checking the luminosity using the photoresistor. 2. Place the solar panel under the testing environment and measure the open circuit voltage using the voltmeter. 3. Connect the output with the reference load to measure current using the ammeter. 4. Change the luminosity of the environment to find the least luminosity that satisfies minimum voltage and current requirements.	✓
2. Regulate the output current and voltage under $1.5\,\mathrm{A}$ and over $4.2\,\mathrm{V}$ respectively under the power protection setup.	1. Place the solar panel under the full sunlight condition to maximize the output voltage and current. 2. Place diodes and capacitors suggested from the datasheet at the output of the solar panel. 3. Measure the voltage and current and change the value of resistors and capacitors if required. 4. Connect with the battery charger when all conditions are met.	✓

- **Battery Charger**

Requirement	Verification	Status
1. Force stop charging the battery if the temperature of the battery is too high to avoid damage to the skin.	1. Place a $10\,\mathrm{k\Omega}$ NTC thermistor between TS (External NTC Thermistor Input) and the battery. 2. Place the solar panel under the full sunlight condition and place the battery and battery charger in the container with a heat source which sets the temperature of the container to $35\,^{\circ}\mathrm{C}$, assuming hot summer days, then measure the temperature of the battery. 3. Cover the battery with thicker fabrics or put refrigerant around the battery if required.	✓

- **Battery**

Requirement	Verification	Status
1. Maintain $3.7 - 4.1\,\mathrm{V}$ of output when charged properly.	1. Fully discharge the battery. 2. Charge the battery fully under full sunlight conditions and measure the time taken. 3. Measure the voltage across the battery using the voltmeter.	✓

- **Voltage Regulator**

Requirement	Verification	Status
1. Supply the output of $3.3\,\mathrm{V}$ and $800\,\mathrm{mA}$ from the input voltage in the range of $3.7 - 4.1\,\mathrm{V}$.	1. Set the voltage regulator circuit according to the typical application circuit from the datasheet. 2. Supply the power from the battery and measure the output voltage and current. 3. Change the value of resistors to satisfy the condition if required.	✓

- **Phone Application**

Requirement	Verification	Status
1. Send Bluetooth signals to the microcontroller every two seconds or less.	1. Turn the Bluetooth module on the microcontroller on and connect the phone to the Bluetooth module by searching for it in the Bluetooth menu. 2. Verify a successful Bluetooth connection by waiting for the Android device to say “Connected” next to the microcontroller.	✓

2. Get turn by turn directions from the user's current location to the user's selected location.	1. Launch the app 2. Verify that the blue bubble icon is located on the map in the user's actual current location. 3. Search for a destination and show the route to this destination from user's current location. 4. Click a button named "Start Navigation" and have a 3D GUI display that shows the user turn by turn directions to user's selected destination.	✓

- **Switch**

Requirement	Verification	Status
1. Pass the power from the battery to the voltage regulator to turn on the device.	1. Build a simple circuit with an LED, a resistor, a power button with the power supply. 2. Flip the button and check if the LED turns on.	✓

- **Microcontroller**

Requirement	Verification	Status
1. Send two distinct power signals out at a time with each being no more than $80\,\mathrm{mA}$.	1. Build a simple circuit with two LEDs on a breadboard. 2. Attach the microcontroller as the power source for each LED with each LED being hocked up to a different pin on the microcontroller. 3. Write a program to power each pin and then to lower the power. 4. Use the ammeter to check if the output current satisfy the requirement.	✓
2. Send a power signal to the vibration motors turning them on.	1. Attach the vibration motors to the pins tested in the requirement 1 of the microcontroller. 2. Rerun the program that was written for requirement 1 and verify that the vibration motors begin to vibrate.	✓

3. Interpret the Bluetooth signal that the microcontroller is receiving.	1. Hook the microcontroller up to the circuit from requirement 1. 2. Write a tester mobile app which sends a signal of 1 for the first LED and a signal of 2 for the second 3. Write a program for the microcontroller that will turn on the pin with the first LED when a 1 is received and will turn on the pin with the second LED when a 2 is received. 4. Using an ammeter, ensure that a current is flowing and the pins are in fact turned on.	✓

- **LED**

Requirement	Verification	Status
1. Turn on the LED when the device is turned on using less than $20\,\mathrm{mA}$.	1. Connect the LED with the microcontroller. 2. Write a program for the microcontroller to power the LED on when the microcontroller gets the supply voltage through 3V3. 3. Measure the output current flowing through the LED.	✓

- **Vibration Motor**

Requirement	Verification	Status
1. Regulate the intensity of the vibration to avoid hurting skins.	1. Hook the vibration motor up to the maximum power the motor can handle. 2. Place the vibration motor on one's arm with several layers of padding in between. 3. Remove layers one at a time ensuring no harm from the vibration. 4. If the vibration is too strong for the bare skin, lower the power to the vibration motor.	✓
2. Vibrate to a level that the user can detect.	1. Hook the vibration motor up to the power and the ground. 2. Slowly increase the power to the vibration motor until one can feel the vibration.	✓
3. Vibrate for at least 10 seconds.	1. Hook the vibration motor up to the power and the ground. 2. Time 10 seconds and verify that the vibration motor is still vibrating.	✓
4. Turn off within 1 second of the power being removed.	1. Hook the vibration motor up to power and ground. 2. Turn off the power supply and measure how long it takes to turn off.	✓