Navigating and Energy Generating Insole: Vibrating Walking Directions

Prof. Bala Maheswaran, Northeastern University

Bala Maheswaran received his M.S. and Ph.D. in experimental solid state Physics, and MSEE in Electrical Engineering from Northeastern University. He is currently a senior faculty at Northeastern University. He has contributed and authored about seventy publications consisting of original research and education related papers, and conference proceedings. He has over twenty-five years of experience in teaching at Northeastern University. He is the Chair of the Engineering Physics Division, ASEE, Chair-elect and executive board member, ASEE NE Section; the co-chair of TASME Conference (Technological Advances in Science, Medicine and Engineering, Toronto, Canada), Academic Member and the Unit Head, Electrical Engineering, ATINER (Athens Institute for Education and Research, Athens, Greece).

Liza Brooke Russell, Northeastern University

I am a mechanical engineering student at Northeastern University looking forward to my first co-op in mechanical design at Fikst Product Development.

Cailey Moy Denoncourt, Northeastern University

Navigating and Energy Generating Insole: Vibrating Walking Directions

Cailey Denoncourt, Liza Russell, and Bala Maheswaran
College of Engineering
Northeastern University

Abstract
Across the world, navigating city streets can be a daunting task. Between avoiding cars, bikers, or other people, having to look down at a phone for directions not only creates a potential hazard, but also distracts from enjoying the surrounding city. As students on a college campus in the middle of a city, Google Maps walking directions are a common method of finding one's way around. However, using Google Maps still poses the same, potentially hazardous problems. Thus, we explored the idea of creating insoles that could fit into one's shoes and vibrate corresponding to the direction one should turn. Furthermore, with the mindset of trying to utilize unharnessed energy from our everyday lives, we aimed to use piezoelectric elements to harness energy from the steps taken by the user. This paper examines the solution by designing a prototype comprised of an Arduino Nano and an HC-05 Bluetooth receiver to connect to Google Maps, making the corresponding insole vibrate in the direction one should turn and further innovating this design by incorporating piezoelectric elements. The unharnessed energy from human movement has a huge potential to save a significant amount of energy, even in the simplest of ways. Thus, by harnessing the energy from steps using piezoelectric elements, the vibrating insole has the ability to generate its own power. This navigating insole could also potentially be of great use for individuals with visual impairments.

Introduction
We have developed an innovative teaching approach for the newly designed eight credit hour cornerstone course for first year engineering students. Rather than studying for exams, through the skill- and knowledge-integrated approaches, highly motivated students are able to interact with other students and faculties from various institutions and take further strides towards real world situations. This method is effective and well-suited to educate our students through hands-on problem solving and team building.

This paper shares a sample project illustrating a new teaching approach through innovation. One of the objectives of the Experiential Engineering Education and this paper is to reform engineering education by moving away from the boundaries of traditional classroom-based approaches to project-, concept-, and team-based approaches using real world situations. This new teaching approach can improve the effectiveness of engineering education by allowing students to think about real problems by developing and executing individual ideas on how to solve the problem faced.

As technology continues to dominate our everyday lives, Google Maps walking directions are a common method of finding one's way around. However, walking across streets filled with cars and people can be dangerous when staring at a phone screen rather than in the direction one is going. Thus, we explored the idea of creating insoles that could go into someone's shoes and vibrate corresponding to the direction they should turn. In addition, with the mindset of trying to utilize the unharnessed energy from our everyday lives, we aimed to use piezoelectric elements to harness energy from the steps taken by the user.

Our project consists of two main components that work in tandem to create a useful and innovative product. First, the smart insole interprets directions from Google Maps and vibrates in the corresponding shoe. This mechanism allows the user to get directions from one place to another without having to look at his or her phone. Second, this device is energy generating due to the utilization of piezoelectric elements within the insole. Both of these components are prototypes but are designed to clearly illustrate a proof of concept.

Google Maps Direction

For the connection between the insole and the walking directions, Bluetooth technology was utilized in our project. Bluetooth utilizes short wavelength UHF waves in the ISM band in order to exchange information.[1] By incorporating Bluetooth technology, the product can be mobile and practical.

The Google Maps portion of our insole is accomplished using an Android application called Tasker, which allows users to build profiles to perform tasks to personalize and automate their phones. Tasker also has multiple plugins that expand the capabilities that the user can perform.[2] We used multiple Tasker plugins called Auto Notification, Bluetooth Serial, and Auto Tools.

Piezoelectric Charging

Piezoelectric energy was experimentally discovered by Pierre and Jacques Curie in the 1880s. They discovered that when mechanical stress is applied to certain materials, charges are displaced, and materials produce voltage under mechanical stress.[3] We harness the energy of human walking movement by placing piezoelectric elements under the foot so that energy is produced with each step.

When mechanical strain is applied to the piezoelectric elements, the crystals produce alternating current (AC). For a battery to charge, it requires direct current (DC). We used rectifying circuit so that the current can only flow away from the piezoelectrics.[4]

Furthermore, because the mechanical strain of walking does not provide constant pressure, the piezoelectric elements produce electricity in bipolar bursts. Since batteries require a more uniform, direct current to charge, a capacitor is necessary to store the electrical energy. The stored electricity is then directed to a rechargeable battery that is better able to retain charge.[5]

Method, Approach, & Design Details

Through the engineering design process[6], the design of the insole went through multiple constructions and trials as the biggest obstacle was obtaining the connection between the insole and Google Maps. After finding ways to connect the insole and the phone, to differentiate separate walking directions, and to successfully piezoelectrically charge the insole and ensure that energy was being collected, a functioning prototype was created. This prototype allows individuals on Android phones to get walking directions through an insole on college campuses.

Google Maps Directions

Tasker Connection

We first focused on how we would establish the Bluetooth connection, for this is the foundational element of this portion of our project. After researching ways to possibly create our own app and connect it to the insole using a Bluetooth LE, we discovered the HC-05, which is one of the most practical Arduino compatible Bluetooth receivers. It allows the user to pair with the phone and receive a signal via Bluetooth connection.[7] We were able to send this signal using the Bluetooth Serial Tasker plugin and interpret this signal using Arduino code.

With the established Bluetooth connection, we then focused on how to obtain directions from Google Maps. Through recommendation from James Packard, a CE/CS major at Northeastern University, we used an Android application called Tasker. The Tasker plugins, namely Auto Notification and Auto Tools, allowed us to intercept notifications from any application on the device, and use that signal to run code for a desired action. Thus, Tasker was programmed to intercept Google Maps notifications, which contain both the distance and direction. At this point, we had solidly established how we were connecting Google Maps data to the phone, the phone to the HC-05, and the HC-05 to the Arduino Nano. With the generalities determined, the specifics for direction differentiation was our last challenge for a successful prototype.

When using Google Maps, the words “Turn right” and “Turn left” do not appear on the screen or the notification pull down. Rather, Google Maps displays a right or left arrow. After systematically testing all the variables gathered from the notification, we concluded that Tasker’s Auto Notification feature is unable to differentiate between the left and right turn arrows. (See Appendix 4). We considered using other map applications, but either ran into similar problems or had problems running the apps from the 2013 Android we were using.

However, we discovered while trying to intercept those notifications when there are no street names, as on college campuses like Northeastern University, Google Maps walking directions fully spells out the direction to either “Turn left” or “Turn right,” along with displaying the turn image without specifying the street name. By intercepting this notification through Tasker, we were able to find and isolate the variables that give the direction and the distance [Figure 1]. The grey pop-up at the bottom of the screen is called “Flash”, which indicates and displays the variables being intercepted by Auto Notification and Tasker. Thus, our insole only works for directions on college campuses or other situations in which there are no street names.

With the variables intercepted, we created two different tasks: one for the right turn and the other for the left turn. Each of these tasks were given a different profile, which is a program that is constantly running. Thus, when Google Maps is on, it consistently intercepts the notification looking for the variables we indicated (See Appendix II for more Tasker Code). Within the two tasks for both the left and right turns, we first had Auto Notification “Query,” or search, for our two variables, which are in this case distance and direction. To make sure that the interception is working correctly, we had Tasker “Flash” the values on the screen for the variables being intercepted. We created an if-statement, so the signal is only sent when the turn is needed, thus the statement will continue only if the direction matches “Turn left” or “Turn Right” and the distance is “0 ft.”

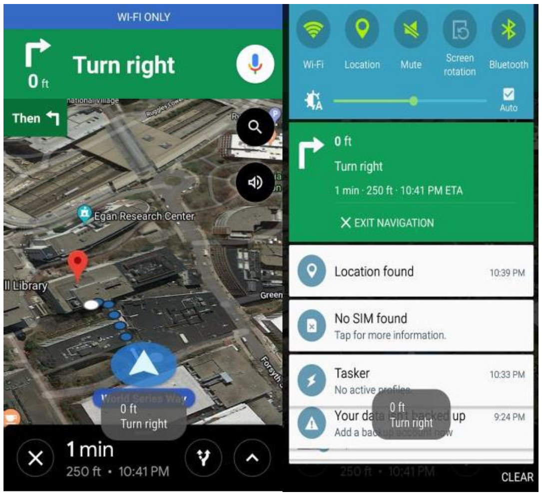


Figure 1: Screenshots of the Turn Right Command and Corresponding Flash in Google Maps (left) and in Notification Panel (right)

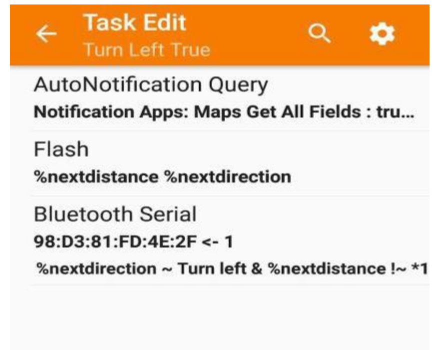


Figure 2: Tasker interface showing the task for a left turn.

When specifying the distance, Tasker did not run correctly when we had the distance match "0 ft." To work around this issue, we created a series of "does not" statements for all of the distance increments that Google Maps uses for walking directions. If all statements within the if-statements return true, the message "1" is sent to the HC-05 [Figure 2]. Since we made a prototype of only one insole, for proof of concept purposes, we wrote the code so that both the left and right turns would send a signal to the one insole.

With more time and resources, we could make a second insole. The accommodation for this would be simple. The second insole would be built exactly like our current prototype but housed inside a right insole rather than a left. We would have to employ a master-slave relationship, because a device can only pair with one Bluetooth device at one time. The left insole would be the master, and correspondence with the right insole would have to occur through the left insole, with the left insole being the master and the right insole being the slave. Master-slave relationships are used in Bluetooth networks, commonly called piconets, to control how data is transmitted between devices. A master device can be connected to as many as seven slave devices, while a slave can only connect to a single master device. The master and slaves send and receive data back and forth, and thus the insole would be able to communicate with both insoles using a master-slave relationship.[8]

Physical Design

Our final model involves a 3D printed insole designed to accommodate all elements of our design in the most practical and efficient manner [Figure 3]. On SolidWorks, we downloaded and traced an image of a real shoe insole in order to design it as realistically shaped as possible. This insole fits the average male shoe size. We then extruded the sole and made all necessary adjustments. On the top of the insole, there are 3 holes that fit the vibration motors. This design allows for the vibration to reach the bottom of the foot and make it detectable to the user as he or she walks.

On the bottom of the insole, there is a large box-like cutout in which we place all electronics of our design, including the Arduino Nano, HC-05 Bluetooth Element, mini circuit board, battery, switch and wires. This box is connected to indented "trenches" so that the wires from the vibration motors and piezoelectric elements can run to the electronic elements without putting pressure on them. There are three small holes that go through the entire sole to connect the compartments for the vibration motors to the trenches that run to the electronics compartment [Figure 4].

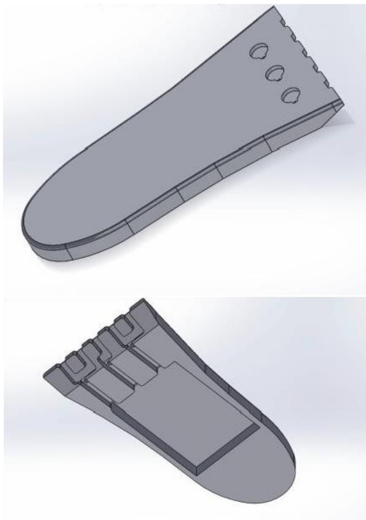


Figure 3: SolidWorks Insole Design

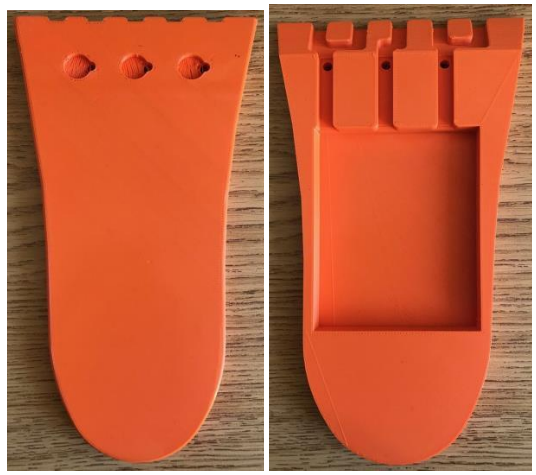


Figure 4: Top View (left) and Bottom View (right) of 3D Printed Insole

Late in the design process, we experienced technical difficulties with the battery and piezoelectric elements, so due to time constraints, the piezoelectric elements and vibration elements remain in two separate soles. These parts would easily be integrated together with a correct battery, however the connection is still being investigated, so that power could be saved for later use. The sole is cut off at the point at which the foot flexes, allowing for natural foot movement. Furthermore, there is an incline on the bottom of the insole so that foot movement will not be interrupted while moving. The insole is connected to a foam cutout of the insole and a cutout of the top of the insole so that it is not uncomfortable but also functional. All electronic elements relating to the Google Maps interaction in our project are housed in these compartments [Figure 5]

We needed the piezoelectric generators to be able to produce enough electricity to charge a battery. We started by using two 3.7V batteries in the hope of generating a total of 7.4V. However, the batteries could not be put in series, and in a parallel circuit would only produce 3.7V together. We had to buy a larger battery that would produce 7.4V in order to have enough voltage to power both the Arduino Nano and HC-05. However, this battery did not fit in the cutout on the bottom of the insole, for it protrudes about 2mm out of the cutout, creating a potential hazard that would need to be fixed for any following prototypes.

Figure 5: Physical Electronic Setup of Insole without Foam (first and second) and With Foam (third and fourth)

Since the amount of energy generated by these piezoelectric generating disks are based on the amount of pressure, they would be most effective if they were placed on areas of the foot that experienced the highest pressure while walking. A study done at Beth Israel Deaconess Medical Center Harvard Medical School in 2005 comparing the pressure during walking between elderly and young adults showed that the two highest pressure points in both age groups was on the medial and lateral calcaneus (heel) and the metatarsals (ball of the foot).[9] When designing the 3D printed insole, we only arranged for trenches to direct wires toward the front, thus decided to place five piezoelectric disks under the ball of the foot. However, for future improvements, the piezoelectric elements could also be placed at the heel in order to harness the most energy generation.

Once the circuit design was devised, we placed the piezoelectric elements in between two pieces of foam, serving both as protection and to increase the mechanical stress in each footstep. This in turn increased the energy they are able to generate.

Piezoelectric Charging

Circuit Set-Up

For the physical setup of our circuit, we connected a switch to the battery so that the Arduino Nano would only receive power when the switch was flipped on. This was done to conserve battery life. We made all the connections practically to save space but used a mini circuit board in order to solder all ground connections together [Figure 6].

The piezoelectric generators we used consisted of small quartz crystals sandwiched between two metal plates, with a positively charged top plate and a negatively charged bottom plate. The piezoelectric elements were connected in parallel, with all the positive and all the negative plates connected as shown in Figure 7. Before running the positives from the vibration elements through the capacitor, we needed to convert from alternating to direct current, so we added four 1N4007 diodes to the circuit. After the diodes, the wires branch to connect to the battery to power the vibration/Bluetooth circuit, which allowed us to take readings of the voltage with the Arduino [Figure 7].

As we lacked access to a capacitor to hold and store the charge, we used the data testing as a method to ensure that the piezoelectric elements were indeed producing some voltage that with the right supplies could be stored in order to power the vibration aspect of the project.

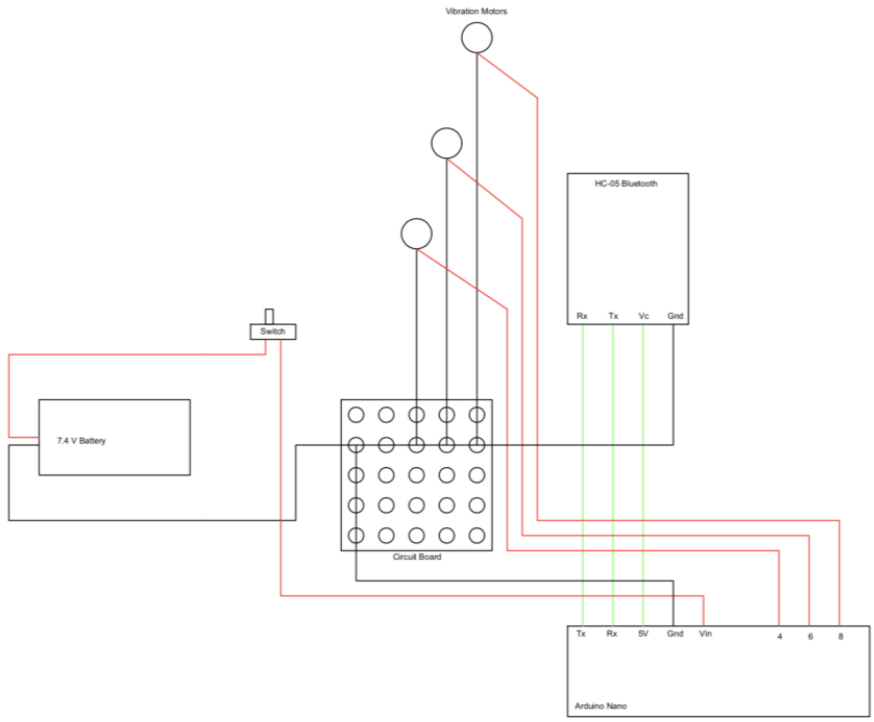


Figure 6: Vibration Circuit with HC-05 and Arduino Nano

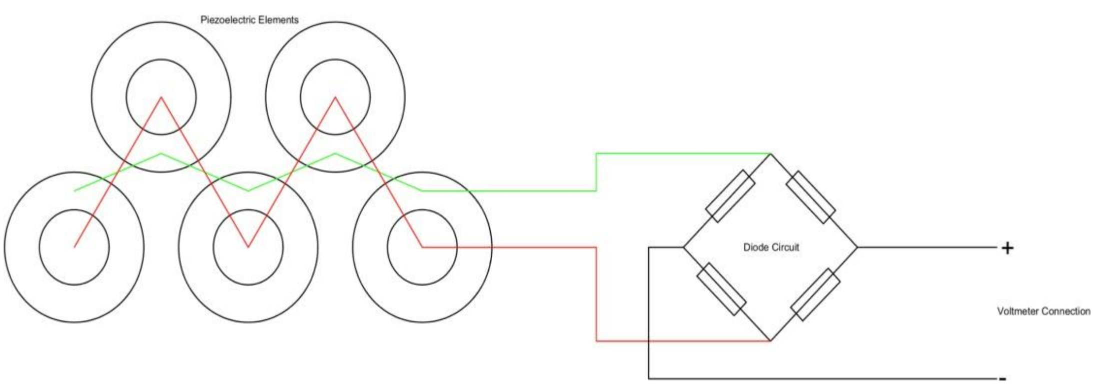


Figure 7: Piezoelectric Circuit

Data Analysis

In order to collect accurate and real-time data, we set-up an Arduino circuit to measure the voltage and gather data that we could plot to analyze the consistency and magnitude of voltage generated [Figure 8]. This circuit was created in order to measure the voltage at every step. Due to a lack of time and resources we were unable to integrate the capacitor into this design. We carried out two tests: CPR method and walking.

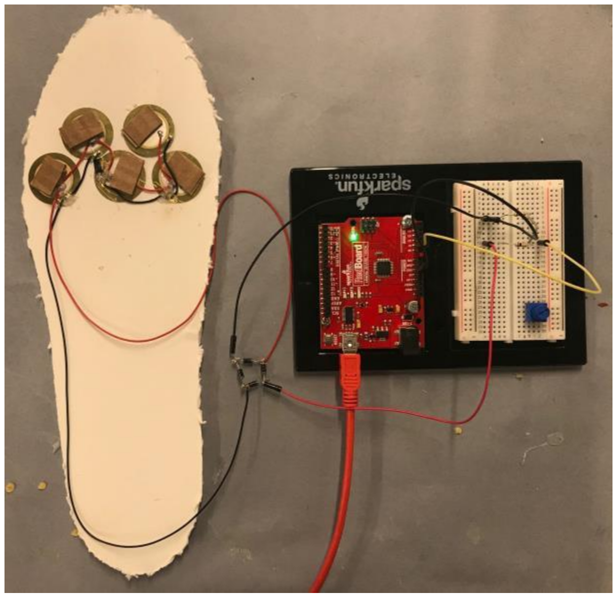


Figure 8: Piezoelectric placement and diode circuit with voltmeter.

The first test was a measure of the generation of voltage when force was applied directly to them with hands. Using a CPR-like method, we pressed firmly on the piezoelectric elements all at once at approximately a frequency of every second and took voltage measurements every 100 milliseconds. Our Arduino code recorded all the numbers, and we used Excel to graph and analyze the data [Figure 9].

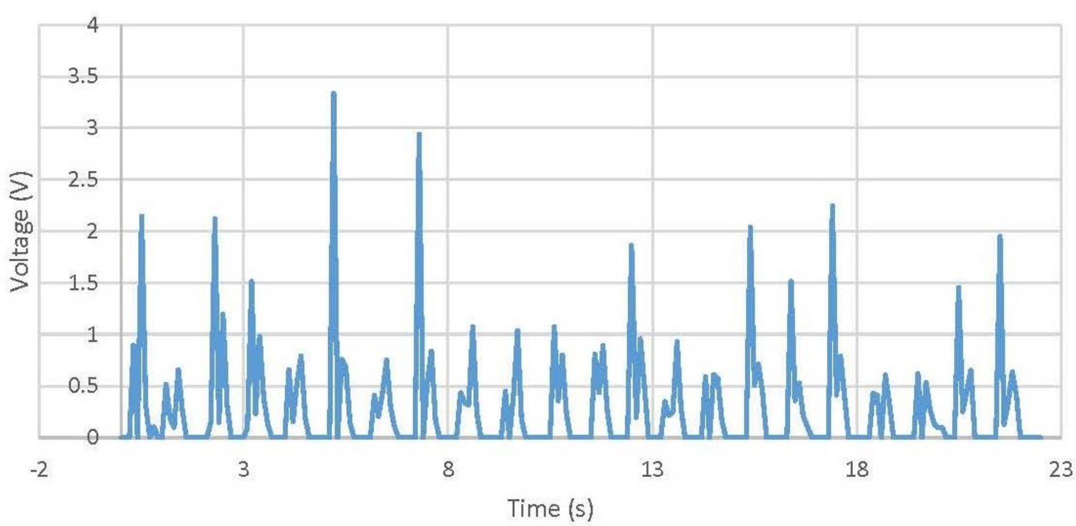


Figure 9: Graph displaying voltage generated by piezoelectric elements when compressed by palm using the CPR method.

The second test gathered data on how much voltage the piezoelectric elements would generate while the user is walking. Since the sole must remain connected to the RedBoard during the data collection, walking was simulated by rolling the foot of the sole. The heel strikes first and then the foot rolls onto the ball of the foot, which pressed on the piezoelectric elements, and then roll onto the toes before starting this process again.[10] Data was recorded every 100 milliseconds, and a full step was completed about every 3 seconds. This voltage data was used to create a graph which displayed a more accurate representation of the electricity produced by the piezoelectric elements while experiencing the mechanical stress of typical walking gait and fast-paced frequency of steps [Figure 10].[11]

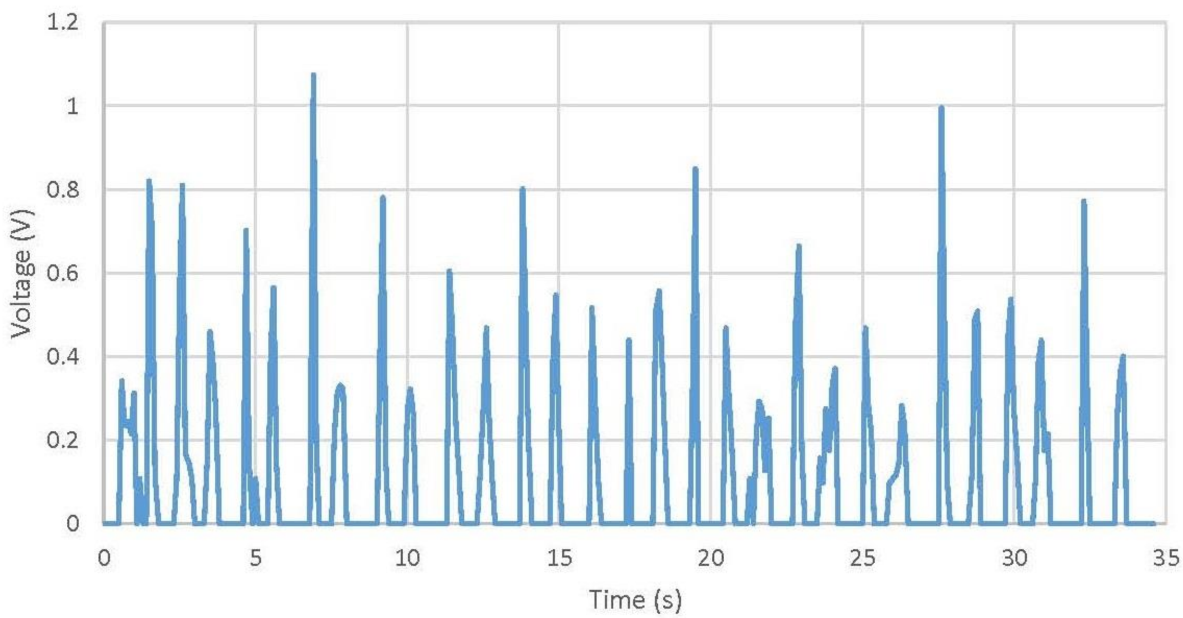


Figure 10: Graph displaying voltage generated by piezoelectric elements when compressed during walking

Results and Discussion

We were able to attain our goal for both the piezoelectric elements and the Google Maps walking directions to a reasonable degree of function and effectiveness. To test the connection and program within Tasker to Google Maps and also with the HC-05 and Arduino Nano, we went on a walk from one building on campus to another using directions from our insole. Following the directions, once the distance read "0 ft," after about a delay of one second, Tasker would intercept the notification as shown with the flash display, and the vibration elements would immediately buzz. Until the turn is made, Tasker continued to send the signal to the HC-05, causing the vibration elements to also continue to buzz until the turn is made. However, with only one insole, we programmed the right and left turn notifications to signal the same HC-05 in the same insole, meaning that the one insole would buzz for both the right and left turn. A conceptual diagram below denotes how different parts of the prototype were connected, with green representing the generation and movement of electricity and gray representing the transmission of information [Figure 11].

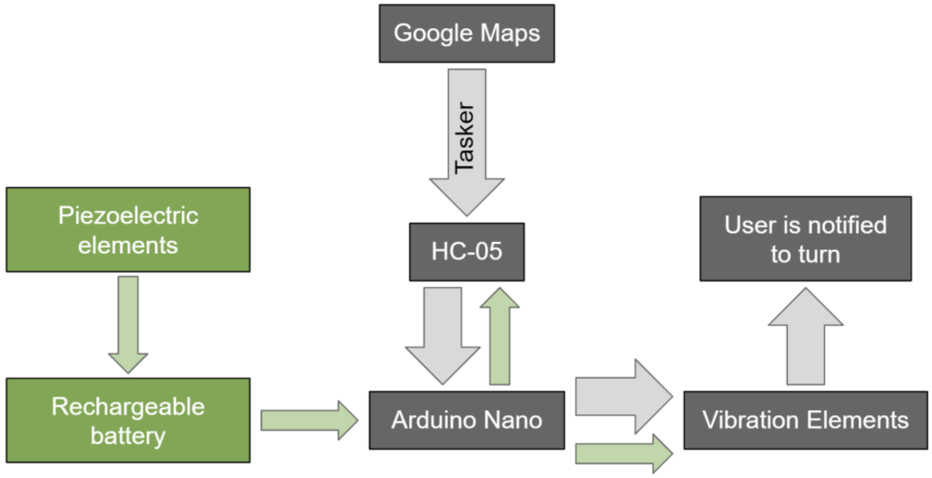


Figure 11: Conceptual diagram illustrating how different parts of the prototype are connected

The purpose of the insoles was to eliminate the need to constantly reference one's phone while using walking directions. However, whenever innovating a new product there are certain ethical concerns of which we must be aware. With limited time and resources, the insole has no seal or cover for the wires. Thus, if the user were to step in a puddle or walk through the rain, the wires and other electrical elements would get wet and likely damage the product with the potential to electrically shock the user. For future designs, this would be modified to create a seal around the circuit to ensure that no water will damage the product or harm the user.

Conclusion

Considering that our design is a proof of concept and we faced constraints for time, money, and available resources, we successfully met our goal to create an energy generating shoe that provides walking directions through vibration elements. Although the insole does give left and right turn directions corresponding to the left and right turn directions received from Google Maps, it does have to be in a setting in which directions do not involve street names. Furthermore, it is not charged via the piezoelectric elements, for these mechanisms are installed on two separate shoes. However, future improvements could involve integrating both functions into one insole. Though we only have one insole finished due to available funds, it can act as a left or a right insole, and the process of making the second insole would be exactly the same, for all we would have to change would be the MAC address of the second HC-05.

If we had more time and resources, we would also be able to optimize our design. In terms of adapting our current design, we would first like to put the Bluetooth and Google Maps interaction circuit in the same insole as the piezoelectric elements in order to have a compact, completely functioning prototype. We would also put three additional piezoelectric elements in the heel in order to maximize the energy output of our design, as following the study suggesting a high amount of pressure in the calcaneal area. Furthermore, we would like to find another battery that would output the necessary voltage for the Arduino Nano (5-8 volts) while still being under 10mm in height, so that the design would be comfortable and not bulky for the user.

Since our tests indicated that Tasker is incapable of accessing the information we need to give universal turn by turn directions, in order to have a more finalized product we would need to reevaluate our approach and find another way to create interaction between Google Maps and the HC-05 in our Arduino circuit. By finding an alternative program, this would also allow for use with iOS.

Our design has many exciting implications and applications in the real world. This product would be appealing to anyone who uses Google Maps walking directions, for it makes walking navigation safer and easier. The user will be able to look at his or her surroundings rather than consulting his or her phone. Furthermore, this design could be extremely beneficial to the blind. Walking directions can be daunting to those who are blind or visually impaired, and our design could ease these struggles. While Google Maps can speak to the user, this can be dangerous in a busy setting, especially since blind or visually impaired individuals rely on their ears as a main method of interaction with the world. The tactile aspect of our project allows their ears to be completely engaged with their surroundings. Even simply expanding our code on Tasker, we could communicate with the user through the number of vibrations sent, i.e., three buzzes for when there is a crosswalk or a long buzz for continue straight. This expansion could virtually eliminate the guesswork for blind or visually impaired individuals following directions and allow them to be more responsive to their surroundings, creating a safer and more comfortable walking experience.

References

[1]. "Bluetooth - An Overview." Science Direct. Accessed April 18, 2019. https://www.sciencedirect.com/topics/computer-science/bluetooth.

[2]. Wagoner, Ara. "What Is Tasker and How Does It Work?" Android Central. August 07, 2017. Accessed April 18, 2019. https://www.androidcentral.com/tasker.

[3]. Woodford, Chris. "Piezoelectricity." Explain That Stuff. August 21, 2018. Accessed April 18, 2019. https://www.explainthatstuff.com/piezoelectricity.html.

[4]. Engineering, Learn. "How Does a Diode Work ?" YouTube. September 23, 2016. Accessed April 19, 2019. https://www.youtube.com/watch?v=JNi6WY7WKAI.

[5]. Mechatronics, How To. "How a Capacitor Works - Capacitor Physics and Applications." YouTube. June 17, 2017. Accessed April 19, 2019. https://www.youtube.com/watch?v=L6cgSxpGmDo.

[6]. "The Engineering Design Process." Science Buddies. March 09, 2019. Accessed April 18, 2019. https://www.sciencebuddies.org/science-fair-projects/engineering-design-process/enginee ring- design-process-steps.

[7]. "Arduino and HC-05 Bluetooth Module Tutorial." How To Mechatronics. February 08, 2019. Accessed April 18, 2019. https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-05- bluetooth-module-tu torial/.

[8]. "Bluetooth Basics." SparkFun.com. 2019. Accessed March 08, 2020.

[9]. Hessert, Mary Josephine, Mitul Vyas, Jason Leach, Kun Hu, Lewis A. Lipsitz, and Vera Novak. "Foot Pressure Distribution during Walking in Young and Old Adults." BMC Geriatrics. May 19, 2005. Accessed April 18, 2019.

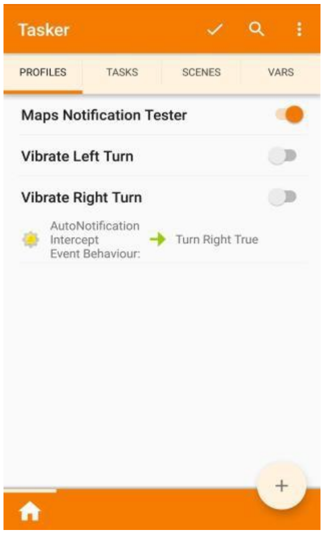
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1173105/.
[10]. Bassett, David R., Holly R. Wyatt, Helen Thompson, John C. Peters, and James O. Hill. "Pedometer- Measured Physical Activity and Health Behaviors in U.S. Adults." *Medicine & Science in Sports & Exercise* 42, no. 10 (2010): 1819-825. doi:10.1249/mss.0b013e3181dc2e54.
[11]. Fedgety Blogspot. “Arduino to Excel Communication.” *Arduino to Excel Communication*, Blogger, 31 Mar. 2019. http://www.fedgety.blogspot.com/2019/03/arduino-to-excel-communication.html.

Appendices

Appendix 1: Arduino Nano Code for Receiving Signal from HC-05

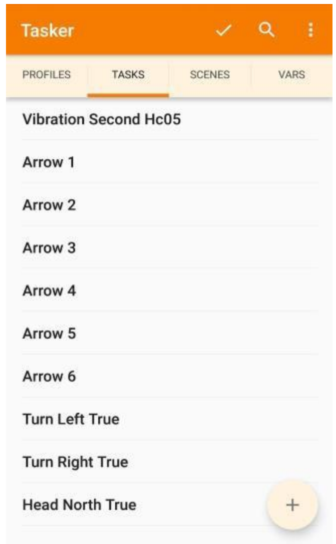
```
int buzzers[] = {4, 6, 8}; void setup()
{
                Serial.begin(9600);
                for (int i = 0; i < 3; i++)
{

pinMode(buzzers[i], OUTPUT);
}
    pinMode(13, OUTPUT);
}

void loop()
{
    if (Serial.available()) // if a message has been received from the HC-05
    {
        buzz();
        digitalWrite(13, HIGH);
    }
}

void buzz()
{
    for (int i = 0; i < 3; i++)
    {
        digitalWrite(buzzers[i], HIGH); // turn each vibration motor on
    }

    delay(500); // keep each vibration motor on for half a second

    for (int i = 0; i < 3; i++)
    {
        digitalWrite(buzzers[i], LOW); // turn each vibration motor off
    }
}
```

Appendix 2: Tasker Screenshots



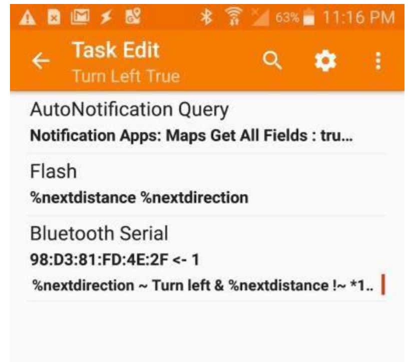
A: List of Profiles

"Vibrate Left Turn" and "Vibrate Right Turn" are the profiles for the left and right insole. When turned on, they vibrate the insole corresponding to a left or right turn. "Maps Notification Tester" is the profile for our in-class demo, which vibrates at the first command of a given set of directions.



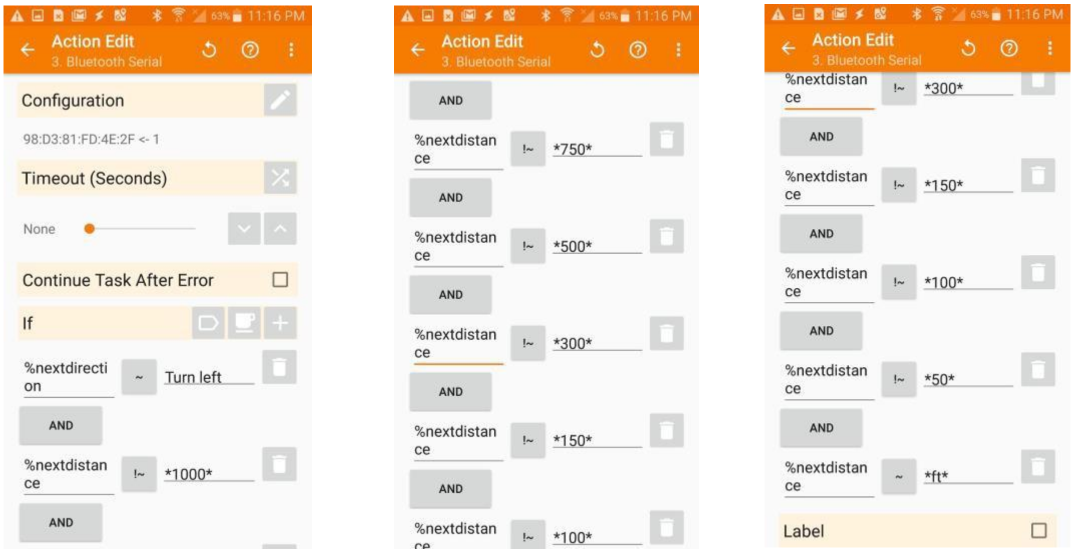
B: List of Tasks

The first task, "Vibration Second HC-05," sends a signal to the HC-05 to make the vibration motors vibrate. We used this to test that our circuit and connection worked, for it does not interact with Google Maps. The tasks named "Arrow 1-6" were used to flash all variables at a given turn, in an effort to differentiate between the left turn and right turn images. These trails suggested that Tasker cannot interpret the left and right turn images. The tasks "Turn Left True" and Turn Right True" evaluate if a signal should be sent to the HC-05.



C: Task Edit

In the "Turn Left True" task, the first action is to query Auto Notification to get variables. We then flash, or display on the bottom of the screen, the "nextdistance" and "nextdirection" variables, so we can evaluate what Tasker is receiving. Finally, we made an action to send a message to the HC-05 if our conditions are met



D: Action Edit

This third action for the left turn sends the message “1” to the HC-05 when the below statements all return true. First, Tasker parses the notification for the variable we created called “nextdirection.” This variable must match “Turn left.” Next, Tasker parses the notification for the variable we created called “nextdistance.” We ran into problems stating that nextdistance much match “0 ft,” so we did the inverse. This variable must not include the numbers 1000, 750, 500, 300, 150, 100, or 50 and must include the phrase “ft.” The asterisks around these statements mean that everything before and after the phrases is ignored to ensure extra spaces would not interfere with whether or not the statement is true.

Appendix 3: Process of Establishing Google Maps Connection

Focusing on the vibration elements connecting to Google Maps first, we explored a couple of ideas that we believed could accomplish this connection. It would be impractical and dangerous to have a cord running from the shoe to the phone, so we determined that some sort of Bluetooth connection would have to be established.

In order to establish this connection between vibration elements and the phone, our first design was using a RedBoard and Breadboard to form the connection. We wrote Arduino code which signals the vibrations when we sent the correct signal. Using an Adafruit Bluetooth Low Energy, we successfully sent a signal from the Adafruit Bluetooth LE Connect App on a phone to the RedBoard, in turn causing the vibration elements to buzz[1]. However, although we were able to establish a connection between the phone and RedBoard through Bluetooth, we were still unable to establish a connection between Google Maps and the RedBoard. After more research we found that Tasker is not able to communicate with the Adafruit Bluetooth LE, and thus we had to search for another way to establish a Bluetooth link between the Arduino and the phone.

As we continued to search for more solutions, we bought a Google Maps API key, which would give us access to Google Maps information that would be necessary to establish the connection[2]. Although this seemed like a plausible way to gain access to Maps walking directions, we needed to find the address and password for our WIFI network. Being on a college campus with campus wide WIFI, we were unable to gain permission to obtain this information.

Appendix 4: No Tasker Variables for Right and Left Arrows

Since we were intercepting the notification, we determined that this information must lie within a notification variable. In order to test this, we methodologically flashed (displayed) all the variables intercepted by Auto Notification at different left and right turns. We recorded and compared all the variables at each direction with each other, looking for any type of similarities between left and right turns. After comparing all the variables, we discovered that each variable either had no difference between left and right turns, or there were differences in every single notification that could not be linked to one direction over the other.

References for Appendices

[1]. "Bluefruit LE - Bluetooth Low Energy." Adafruit Industries. Accessed April 18, 2019. https://www.adafruit.com/product/1697.

[2]. "Maps API." Google Maps Platform. March 11, 2019. Accessed April 18, 2019. https://developers.google.com/maps/documentation/javascript/get-api-key