Vibrational Directional Gloves (VDG)

Project Report

Daivya Shah (dts7992@nyu.edu)

Kaylee Weber (kew6119@nyu.edu)

Jiangnan Wang (jw7572@nyu.edu)

Professor Rui Li (rui.li@nyu.edu)

Engineering Design 1004

New York University Tandon School of Engineering

December 6th, 2022

Introduction

The primary objective of this project was to create gloves that vibrate to help drivers by providing tactile responses to GPS directions, either left or right. Tactile signals were utilized for several reasons, such as the fact that skin consists of at least three different types of receptors that interpret different touch sensations (1). Additionally, the human response time to tactile stimuli is "28% and 34% shorter than that of auditory and visual stimuli" (2), where the increased response time could be valuable in a driving environment. A focus was placed on driving since there were 228 million drivers in the U.S. in 2020 (3), and a survey in 2022 found that 93% of drivers depend on their GPS every day (4). GPS is a tool that focuses solely on auditory and visual communication, forcing drivers to take their eyes off the road or listen to the rarely repeated directions that Siri states before approaching an intersection. Therefore, the Vibrating Directional Gloves (VDG) aim to add a tactile element to GPS and driving to make the roads safer and drivers more confident.

The VDG focuses almost entirely on a speech recognition system designed to understand when the GPS states "left" or "right" before a turn. Once that phrase is understood, the Arduino components send the proper signals to a vibration motor on the corresponding glove. This system will allow drivers to feel more confident about where they are going and help those prone to zoning out on long car rides know which way they will be turning. Further applications of the VDG include the potential to prevent car crashes due to last-minute lane switching or the need to look at a GPS rather than the road. Currently, the VDG could help the hard of hearing and the directionally challenged feel more confident while driving. Additionally, the VDG can help the everyday person who may zone out or not pay attention to their GPS while driving.

Requirements

1. Physical Components

The design for the VDG is centered around an Arduino Nano 33 BLE Sense and vibration motors. Additional circuit-related components include a breadboard, wires, resistors, a diode rectifier, an NPN transistor, and a 3V power source. The circuit is designed to allow the Arduino Nano to control the vibration motors when responding to the auditory input and vibrate the correct motor based on that input. The entire Arduino circuit is contained in a small, 3D-printed frame designed to be worn on the forearm. 3D printing the frame has allowed the design to remain lightweight while still being sturdy enough to secure the Arduino components. The 3D-printed model is placed in a sleeve and is secured to the glove using Velcro. The glove and sleeve are both constructed using a Nike sock, allowing the design to remain comfortable and low-cost.

2. Software Requirements

The software focused heavily on a website called Edge Impulse. Edge Impulse is a free online resource that helps developers collect data and structure it in various formats. Edge Impulse was utilized to train the speech samples of "turn left" and "turn right" so that a program could recognize the phrases. Other elements of the software included the Arduino IDE, which uses the programming language C++. Multiple external libraries were downloaded to the Arduino IDE for the program to recognize the Arduino Nano 33 BLE Sense and the commands in the code generated from Edge Impulse. These libraries included Adafruit_ZeroPDM-Master, Arduino BLE-1.3.2, and the file from Edge Impulse named Project2_inferencing.

3. Project Schedule

The projected schedule starts on October 9 and ends on November 10, lasting for 33 days. The time after November 10 will be focused on advanced functions for extra credit.

ID		Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1			research and DPI	7 days	Sun 22/10/	Sun 22/10/		
2			fusion	10 days	Tue 22/10/	Sun 22/10/		
3			coding(total)	15 days	Sun 22/10/	Thu 22/10/		
4			overall checking	7 days	Sun 22/10/	Sun 22/10/		
5			purchasing of external resources after research	7 days	Wed 22/10/5	Thu 22/10/13		
6			requesting school provided materials	5 days	Sun 22/10/9	Thu 22/10/13		
7			coding test(Inside arduino)	7 days	Sun 22/10/9	Sun 22/10/16		
8			coding research and preparation	12 days	Sun 22/10/9	Sun 22/10/23		
9			final coding and check after hardware	8 days	Thu 22/10/20	Sun 22/10/30		
10			circuit design	6 days	Thu 22/10/	Thu 22/10/		
11			circuit completion	10 days	Sun 22/10/	Thu 22/10/		
12			gross assembling	4 days	Mon 22/10/	Thu 22/10/		
13			Vrecog software	6 days	Thu 22/10/	Thu 22/10/		
14			Vrecog sotware coding(possibly)	6 days	Thu 22/10/13	Thu 22/10/20		
15			complete assembli	6 days	Thu 22/10/	Thu 22/10/		
16			compiled technical problem solving	3 days	Thu 22/10/27	Sun 22/10/30		
17			extensive coding	6 days	Mon 22/10/	Sun 22/11/		
18			developed functions(possibly)	9 days	Thu 22/10/27	Tue 22/11/8		
19			final and overall check	6 days	Mon 22/10/31	Sun 22/11/6		
20			product release preparation	9 days	Mon 22/10/31	Thu 22/11/10		

4. Cost Schedule

The initial cost estimate was expected to be within $7,640. As the team progressed, the final cost estimate was determined to be $2,540.59.

RESOURCE	COST(PER UNIT)	MS1		MS2		MS3	
		QUANTITY	COST	QUANTITY	COST	QUANTITY	COST

3D Printing material	undetermined	undetermined	<10	undetermined	<10	N/A	N/A
Vibrators	1.3	2	<20	2	<20	2	2.6
Microphone and Arduino	External source	1	<100	1	<100	1	37.99
straps	undetermined	undetermined	undetermined	N/A	undetermined	N/A	N/A
Speech detection	undetermined	undetermined	undetermined	Arduino	Arduino	Edgeimpulse	N/A

Projected labor	50	150	7500	150	7500	50	2500
TOTAL		<$7640		<$7640		2540.59	

Procedures

1. Physical Construction

The physical construction of the VDG required two main elements: the CAD model to hold the Arduino components and the glove itself. The CAD model was developed in two phases. The first phase was eight inches long and built with two levels. The bottom level rested against the forearm of the user, and the top level held the Arduino components. The upper level of the CAD model was originally designed to hold an Arduino Uno and a breadboard. This was amended when it was decided that the Arduino Nano 33 BLE Sense would be used for the project, which would rest directly on the breadboard. The space for the Arduino Uno was converted into a battery holder, and the overall length was decreased to seven and a half inches so that the design could be printed with the Ultimakers. The final CAD model was printed using PLA printing material.

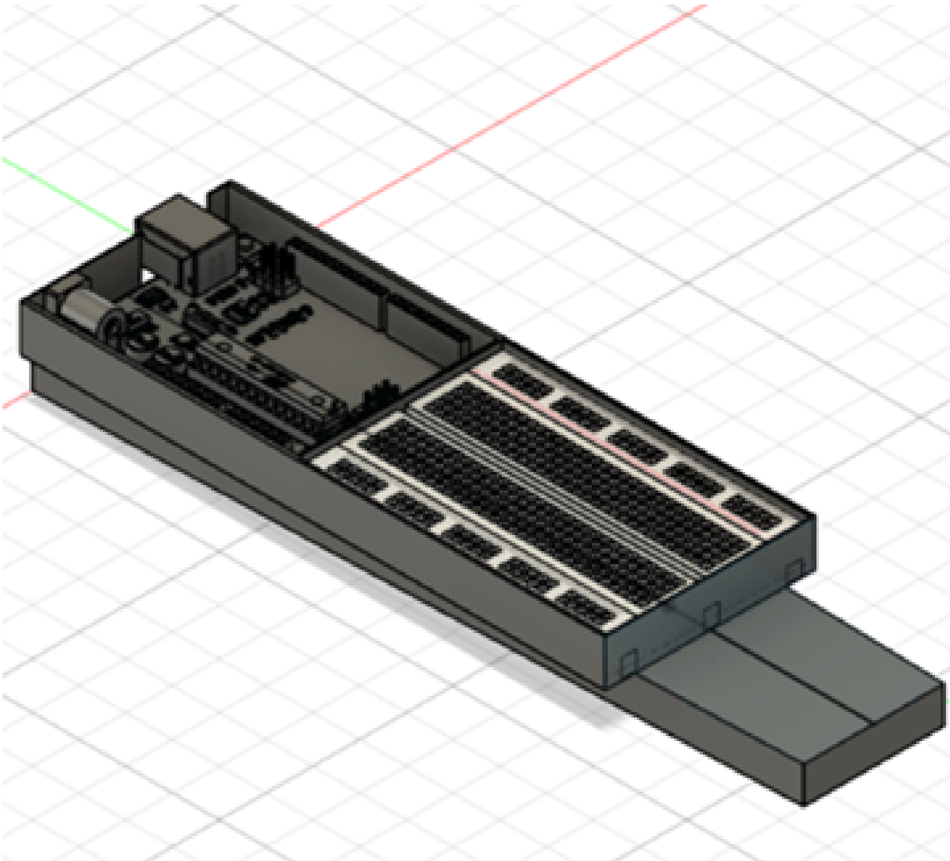


Figure 1: Original CAD Model, Isometric View

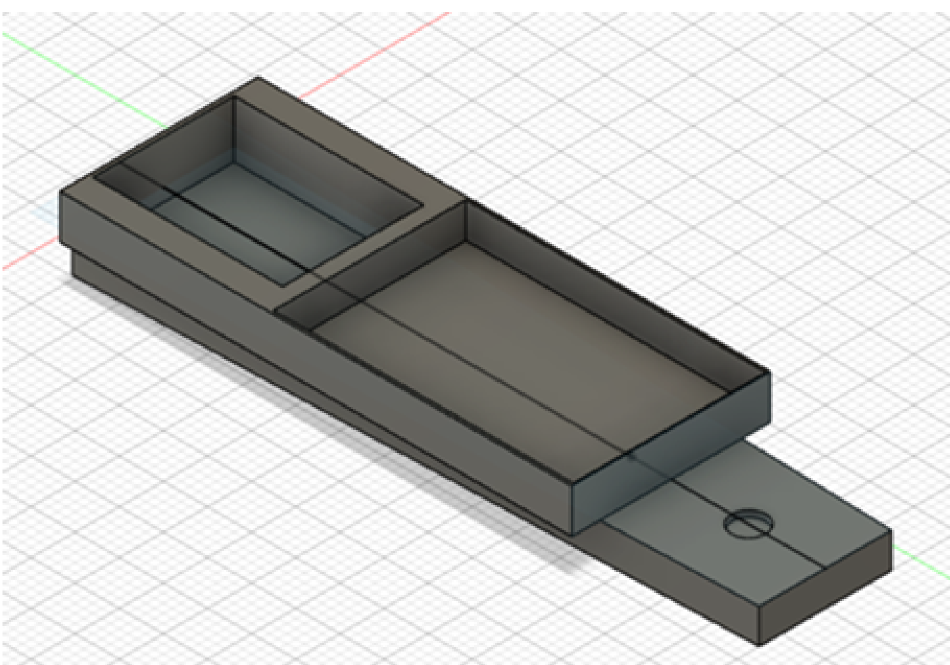


Figure 2: Final CAD Model, Isometric View

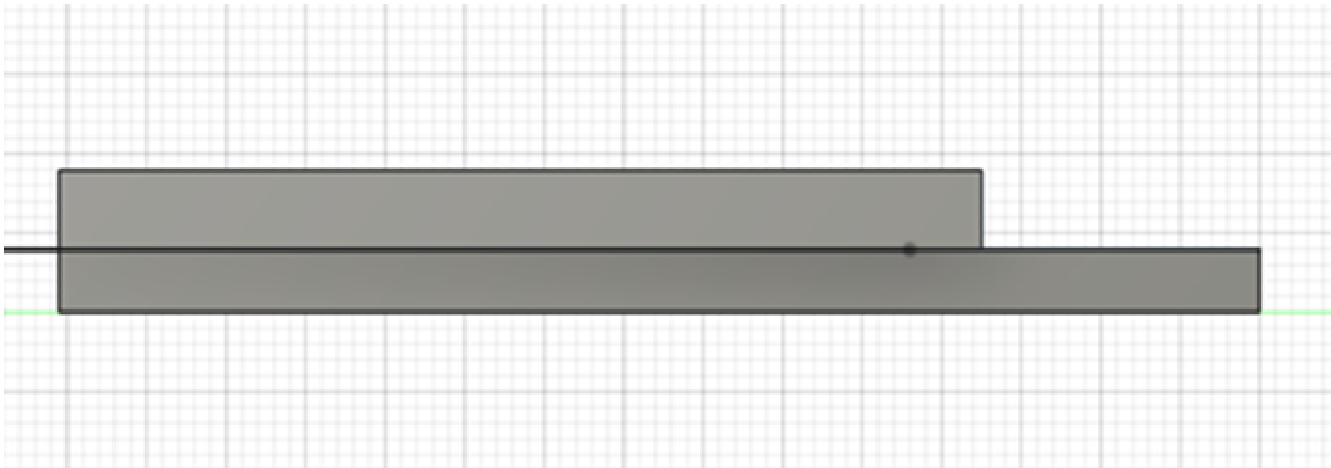


Figure 3: Final CAD Model, Side View

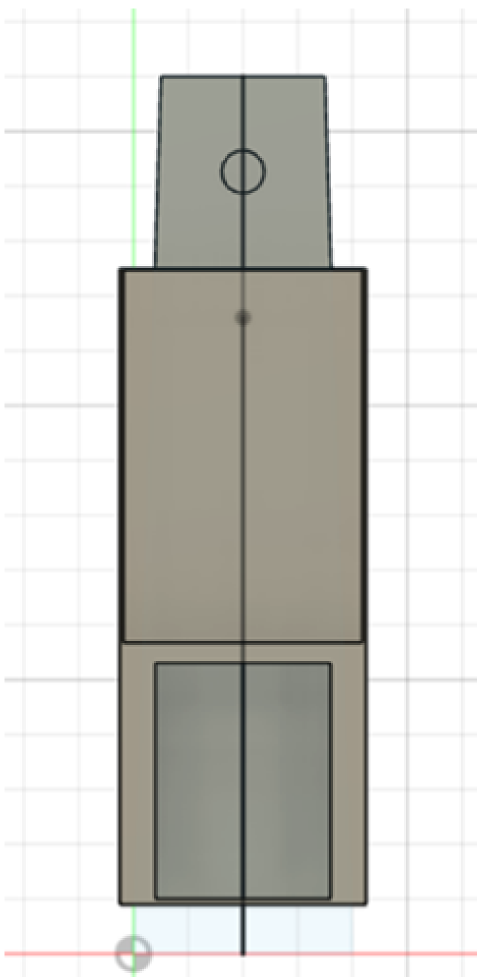


Figure 4: Final CAD Model, Top View

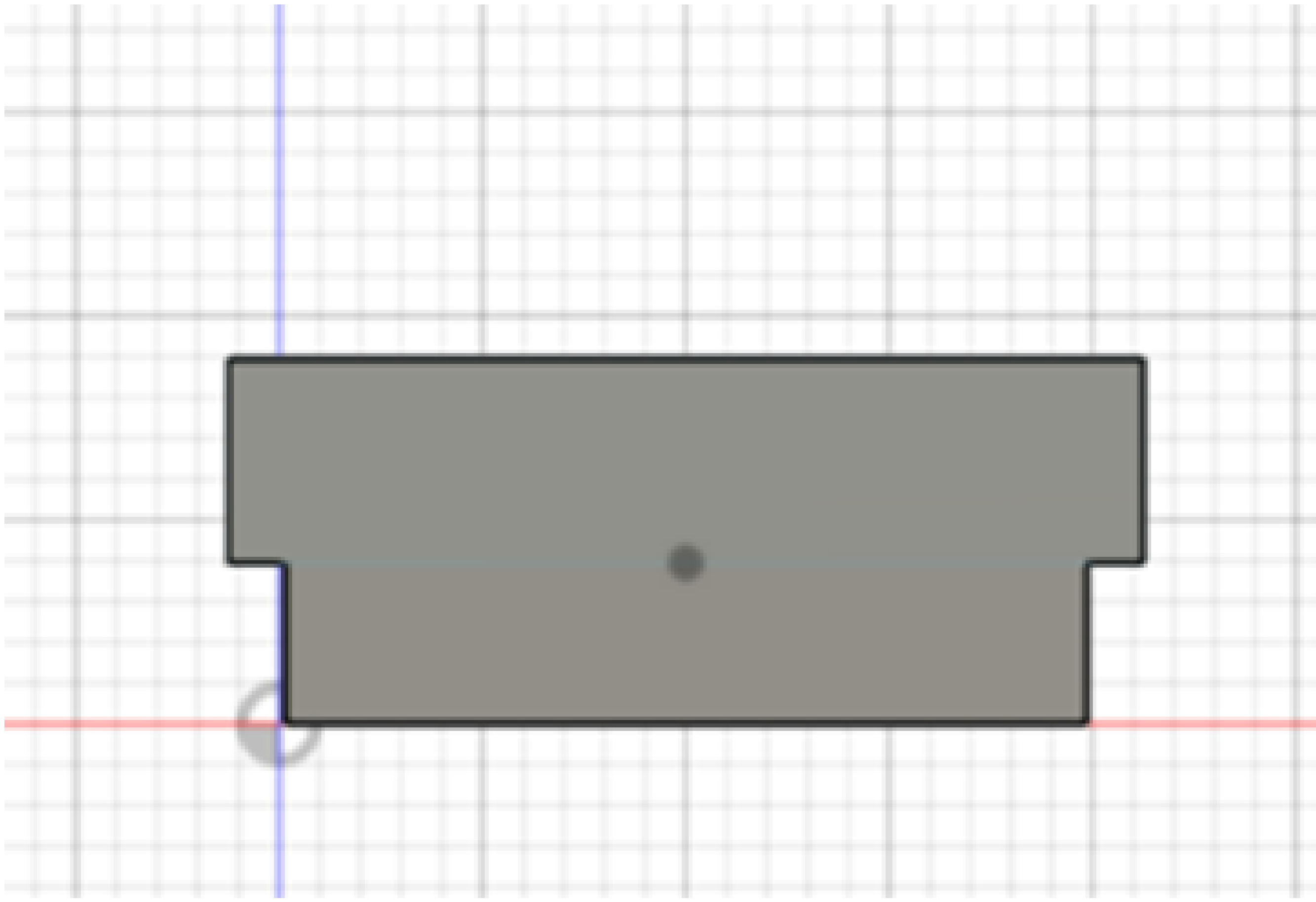


Figure 5: Final CAD Model, Front View

The glove was originally meant to be a true glove that would go over the hand of the user. When it became apparent that the design would have to be large to accommodate the Arduino components, the focus shifted to the forearm. The glove was made from a recycled Nike sock for its comfort, length, and low-cost option. Construction began by cutting the sock right above the heel. A hole was cut into the side of the sock to allow the user's thumb to fit and hold the sock in place. The remainder of the foot of the sock was found to be roughly the same size as the printed CAD model. The foot of the sock was placed over the CAD model to contain the Arduino components. The CAD model and the rest of the glove were secured together by wrapping Velcro around both components, securing the device to the user's forearm.

Figure 6: Glove Base

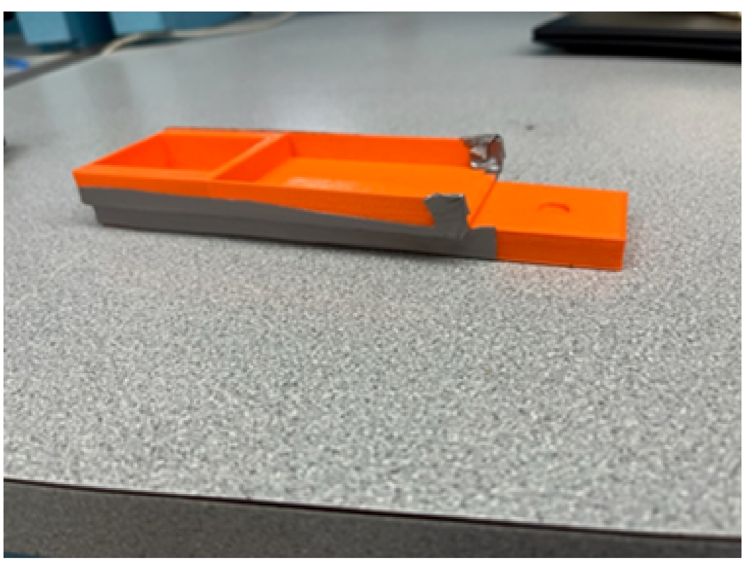


Figure 7: 3D Printed CAD Model

Figure 8: Wearable Assemble

The other main element of the physical construction was the Arduino circuit for the vibration motors. The circuit included the Arduino Nano 33 BLE Sense, a breadboard, wires, resistors, a diode rectifier, an NPN transistor, and a 3V power source. Problems arose when it was realized that the wiring done with the Arduino Uno for the vibration motors was not applicable, and new circuit diagrams had to be found. The circuit was wired in accordance with the specifics listed on Circuito.io (cite source, 5). Unfortunately, the circuit that was found did not appear to be functional, and no other easily accessible wiring diagrams existed. Research into forums around this issue yielded inconclusive results. If time permitted, the next step would have been to add a 100 nF capacitor, which was seen in many posts, and test the new circuit with the Arduino Nano 33 BLE Sense.

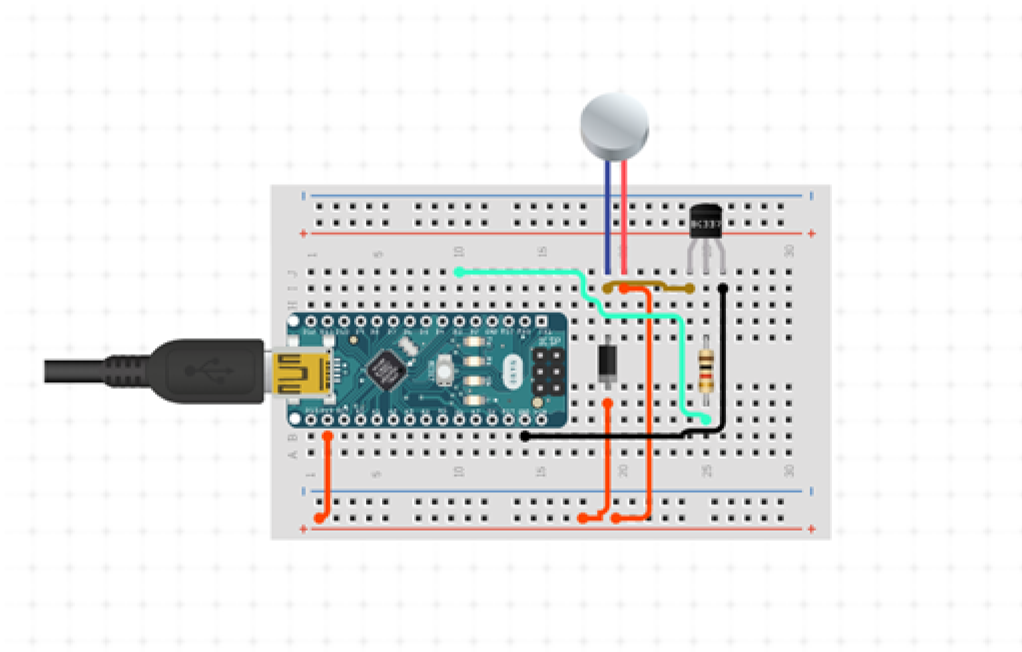


Figure 9: Circuit Diagram Courtesy of Circuito.io

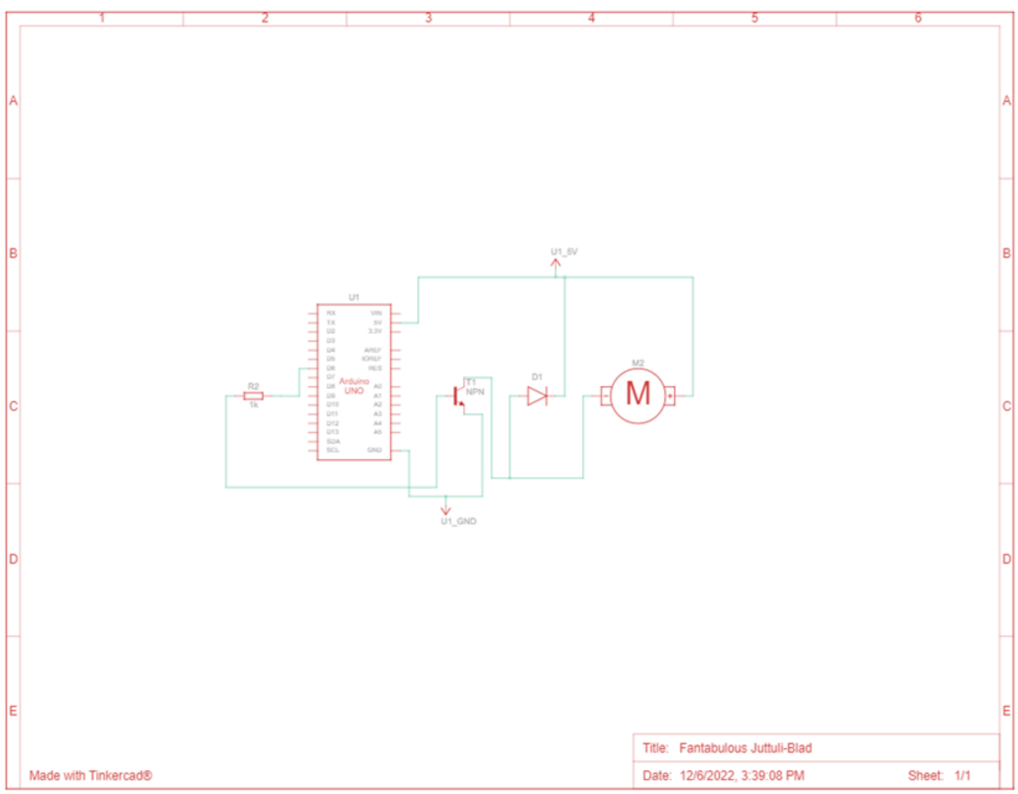


Figure 10: Wiring Schematic (There is no Arduino Nano option in Tinkercad, so an Arduino Uno was used for clarity)

2. Software Setup

The software setup was done almost entirely using Edge Impulse. To train a program to understand an automated voice saying "turn left" and "turn right," Edge Impulse was used to eliminate user error. Projects were built using the free "developer" option on Edge Impulse, and a category for the data to be used was selected—in this case, audio. The next step was to collect various audio files of "turn left," "turn right," and "noise." Thirty samples of "turn left," thirty samples of "turn right," and twenty-six "noise" samples were used in the final project. Edge Impulse automatically split the data into training and testing datasets, with 83% for training and 17% for testing. An "impulse" had to be created to train the data. The impulse includes the "window size" (how long the clips were expected to be), the "window increase" (how many training steps the data would run through), the "processing block" (MFCC audio was used to recognize voice commands), and a "learning block" where the "Classification (Keras)" option was selected.

Once the impulse was created, the next step was to run the data through the specified parameters and check the accuracy of the data recognition. Steps for this process were followed from CircuitDigest (cite source, 6). The neural network of the classification learning block was edited using the code provided by CircuitDigest to allow the program greater flexibility for voice recognition. The final overall accuracy for the project was 76.5%. When the training was completed on Edge Impulse, the program was exported to an Arduino library file named Project2_inferencing. The zipped folder was downloaded into the Arduino IDE along with the "Adafruit_ZeroPDM - Master" and "Arduino BLE-1.3.2" libraries to allow the code to compile. The code successfully compiled; however, due to minimal prior knowledge of coding, the program was difficult to understand. Therefore, there was difficulty in transferring the steps obtained from CircuitDigest into a new format. Unfortunately, this learning process was slow and could not be completed in the allowed time frame.

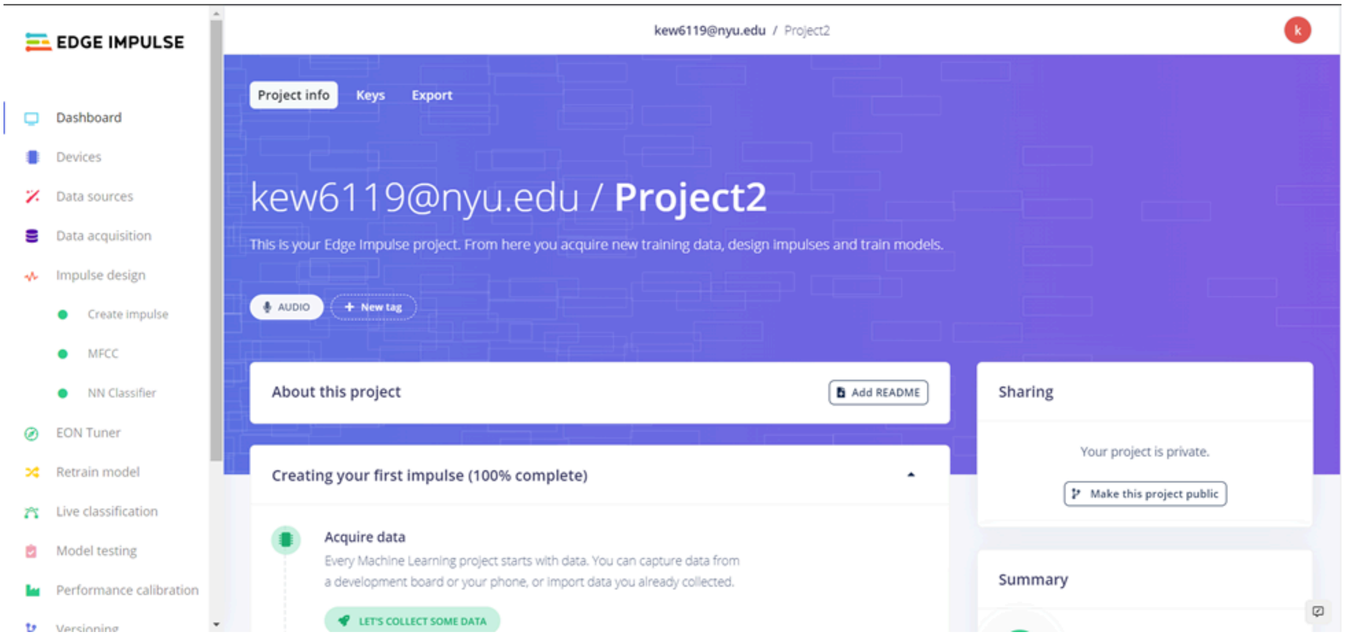


Figure 11: Edge Impulse Homepage

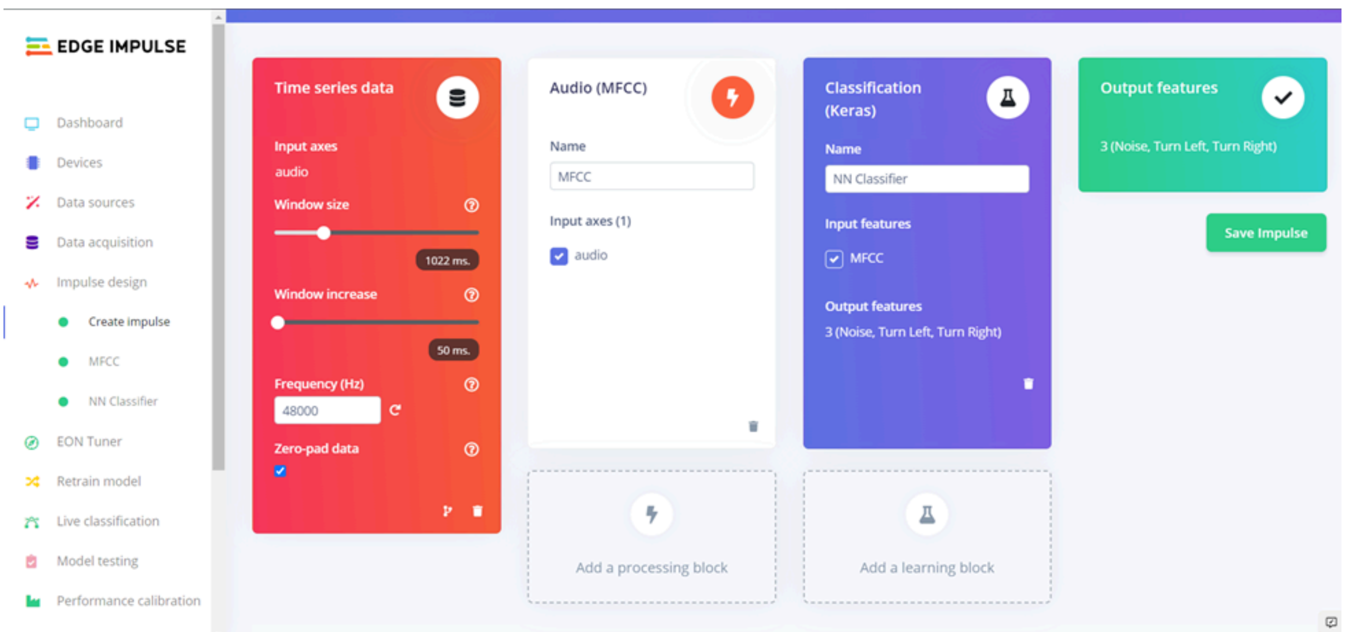
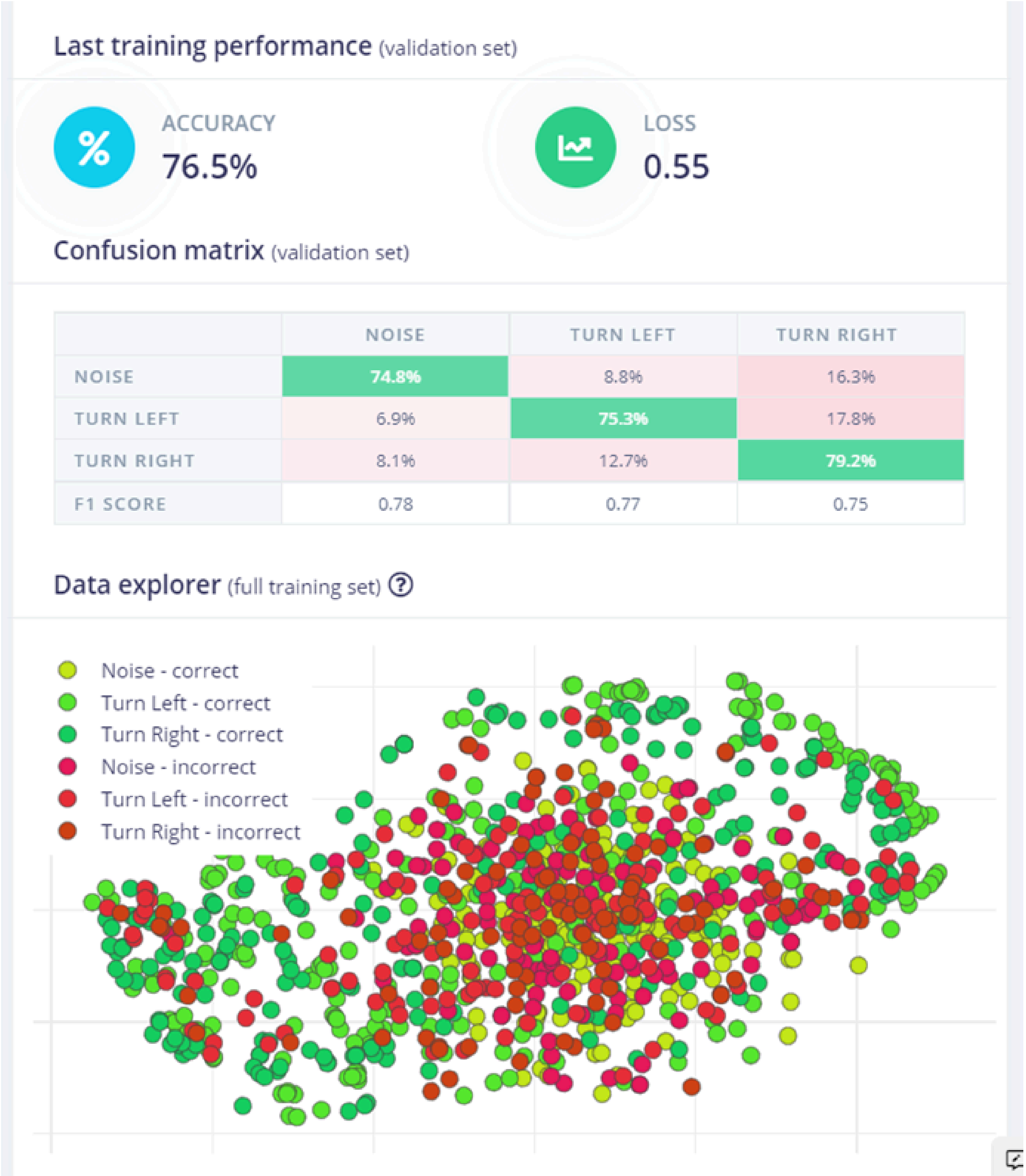


Figure 12: Learning Blocks Used for Voice Recognition



	NOISE	TURN LEFT	TURN RIGHT
NOISE	74.8%	8.8%	16.3%
TURN LEFT	6.9%	75.3%	17.8%
TURN RIGHT	8.1%	12.7%	79.2%
F1 SCORE	0.78	0.77	0.75

Figure 13: Final Accuracy of the Project

3. Software Troubleshooting

Problems inevitably arose due to limited knowledge on the topic of speech recognition and Arduino. Regarding vibration motor circuits, extensive research was done. A somewhat helpful tool was forums on the Arduino website itself, where other posts were made by designers to ask for help from the community. While the problems posted did not directly match the circuit and issues encountered in the construction of the VDG, many similar problems were found. Unfortunately, not all the postings had an "answer" to the problem, and many different postings were researched to find common denominators for wiring the vibration motor circuit. Through this research, it was discovered that the vibration motor circuit for an Arduino Nano was different from the circuit required for the Arduino Uno. Conflicting answers were found regarding the Arduino Nano circuit; however, commonalities suggested that the Arduino Nano circuit required an additional diode and possibly a 100 nF capacitor. Unfortunately, due to the time it took to research these topics, this solution was not able to be implemented.

Regarding the speech recognition side of the project, many roadblocks were encountered. There was no prior experience with Edge Impulse or speech recognition, and a lot of experimentation was done with the website to understand its functionality. Multiple different projects were created through the website to test different aspects of the training program and the factors that controlled the speech recognition accuracy. One project was created with a minimal dataset—ten samples for each category—to test how adjusting the "window increase" changed the accuracy. It was discovered that decreasing the "window increase" generated more training steps and was able to increase the accuracy of a project. Meanwhile, a separate project was examined to understand the difference between using an automated versus human voice. The dataset for the project was minimal; however, the initial results suggested that human voices might be easier to decode, prompting the addition of more automated voice recognition samples in the main project to balance the discrepancy. Unfortunately, raising the accuracy of the voice recognition was not an exact science, as there were many competing variables at play. Contrary to prior belief, the addition of more voice samples did not always raise the accuracy, adding to the complexity of the project.

Milestone and Final Product Requirements

Benchmark A was completed by the end of Model Shop Session II. The requirements for Benchmark A included the initial CAD model, preliminary design investigation, team logo, updated engineering notebook, and specific requirements assigned by the RAD mentor. The requirements had to be checked by an Open Lab TA to complete the benchmark.

For the VDG, the Benchmark A requirements were:

- Initial CAD model
- Completed preliminary design investigation
- Updated engineering notebook
- Logo submission in .STL or .gcode file
- Arduino code flowchart to take in two separate inputs and output to the two vibrators. One input should turn on vibrator 1, and the other input should turn on vibrator 2.
- Completed circuit with the two vibrators that will take in two different inputs to turn on the corresponding vibrator. The circuit should be built in Tinkercad.
- Research on speech recognition

Benchmark A early submission was due on Tuesday, October 4th. Regular submission was due at the end of Open Lab on Sunday, October 16th. Similar to Benchmark A, Benchmark B included an updated CAD model, a team logo approved through the 3D Printing Submission portal on the EG website, an updated engineering notebook, and specific requirements outlined by a RAD mentor.

The Benchmark B requirements for the VDG were:

- Updated CAD model
- Updated engineering notebook

- Logo approval in Proto Lab by a Proto TA
- Make the device functional. The Arduino code should be working, meaning it will take in an input (left or right) and turn on two different vibrators based on the input. Turn on the left vibrator if the input is "left," etc.
- Create a wearable glove and assemble the Arduino onto the wearable to model the device. The wearable must hold the Arduino with enough support. Assembly should be complete and removable.
- Printed CAD model to be assembled (this includes casing, support, anything required to assemble the wearable)

Benchmark B early submission was due on Tuesday, October 25th. Regular submission was due at the end of lab on Tuesday, November 1st. Early final submission was due at the end of the lab on Tuesday, November 29th. Regular final submission was due at the end of the lab on Tuesday, December 6th.

The following items were expected from all RAD groups:

- Final presentation
- Circuit Diagram
- Schematic Diagram
- Code Flowchart
- Commented code
- All CAD drawings of your design
- Video advertisement of the prototype in use
- Final Microsoft Project schedule
- Final cost estimate
- Resume

- Final Engineering Notebook
- Final Design Report

The distribution of work assigned Daivya Shah as the head of programming, Kaylee Weber as the head of design, and Jiangnan Wang as the head of production. Other requirements stipulated that at least one group member had to receive MakerSpace training for 3D printing on Ultimakers to print CAD designs. Kaylee Weber and Jiangnan Wang both received online training and in-person check-ins to use the Ultimaker printers. No further training on advanced 3D printers was necessary for the project.

Results

The Benchmark A tasks assigned were all completed by the end of Open Lab on Sunday, October 16th. These tasks included an initial CAD model, the preliminary design investigation, an updated engineering notebook, a submitted logo design, a code flowchart that takes two separate inputs and vibrates two separate vibrators accordingly, and a completed circuit with two different inputs and their corresponding vibrators on Tinkercad. The Benchmark A tasks were approved by Aden Lin.

The Benchmark B tasks assigned were all completed by the end of the lab on November 8th. The tasks required included an updated CAD model, an updated engineering notebook, a logo approved by the Proto Lab TAs, functional Arduino code that could take input of left or right and turn on the correct vibrator, and a wearable glove assembly that could support the Arduino components. The Benchmark B tasks were approved by Sydney Blum.

Various obstacles were encountered, and these were apparent by Benchmark B. During Benchmark A, there was hesitation and confusion regarding how to complete the speech recognition. The problem was later addressed with advanced research into speech recognition, and Edge Impulse was found to be the solution. The largest problem occurred during Benchmark B due to the lateness of the Arduino Nano 33 BLE. The problem was addressed through communication and negotiation with Professor Li, who allowed the team to complete the benchmark the week after receiving the Arduino Nano 33 BLE Sense. Benchmark B was completed with an approved time extension.

Conclusion

The team succeeded in software development using Edge Impulse; however, limited understanding of code prohibited further development with the generated Arduino code. Additionally, unresolved circuit challenges and the delayed arrival of the necessary Arduino caused unexpected delays, leading to partial commissioning. Despite this, the project is not a failure. The team is the first to use the Arduino Nano 33 BLE, a specific type of Arduino aimed at speech recognition. Online test results revealed its high accuracy and potential. All efforts of team members were appreciated. Given more time, the functional product could be fully developed. At this stage, due to our high level of completion, several future improvements can be anticipated.

The device is expected to read numbers, not just words, allowing it to send not only qualitative signals (left or right) but also quantitative signals, such as changing vibration frequency in accordance with the distance from the crossing or direction points as spoken by the GPS. This advanced function only requires additional construction blocks in the software development, which can be updated and improved simultaneously. This methodology of transforming different types of signals can also be applied to other fields. With further development and safety enhancements, similar products may allow visually or auditory-challenged individuals to receive tactile sensations, applicable in various aspects of their lives beyond driving. Future developments of the product, aimed at promoting human well-being and functionality, hold significant potential and promise.

Works Cited

1. https://askabiologist.asu.edu/explore/how-do-we-feel-touch
2. https://www.iaeng.org/publication/IMECS2012/IMECS2012_pp1449-1454.pdf
3. https://usafacts.org/data/topics/people-society/transportation/safety/licensed-drivers/.
4. https://www.carpro.com/blog/where-drivers-are-most-dependent-on-gps-systems#:~:text=93%25%20Of%20Drivers%20Depend%20On%20GPS&text=In%20fact%2C%20researchers%20say%2093,would%20get%20lost%20without%20it
5. https://www.circuito.io/app?components=514,8449,11022
6. https://circuitdigest.com/comment/33907#comment-33907