Tongue-Driven Wheelchair for Quadriplegics: Exploring Assistive Technology's Impact on Those with Disabilities

A Technical Report submitted to the Department of Electrical and Computer Engineering

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Nicholas Talton

Spring 2024

Technical Project Collaborators:

Michael Kinsel

Dhruv Batra

Yeabsira Mekonnen

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Nicholas Talton

Advisor: Adam Barnes, Department of Electrical and Computer Engineering

Contents

List of Figures

1 Abstract

The tongue drive system is a wearable assistive device that will help individuals with spinal cord injuries or any neuromuscular diseases, also known as quadriplegics, better navigate the world by allowing them to use their tongue to control an electric wheelchair. This head-worn device is equipped with magnetic field sensors to track the fluctuation of the magnetic field due to a magnet attached to the user's tongue. Using the data from the sensors, an MSP432 microcontroller performs some data processing to determine the tongue's position and transform that into control data that can be outputted to a wheelchair. Due to limited time and resources, this project does not use a physical electric wheelchair; instead, it sends the control data to a virtual wheelchair through a computer simulation. This device is designed to be easy and intuitive to operate with as little outside assistance as necessary.

2 Background

Spinal injuries affect more than 17,800 Americans each year - with the current amount living in the United States estimated to be approximately 294,000 [1]. Of these, 60 percent suffer from Quadriplegia, paralysis below the neck [2]. Quadriplegics are unable to use traditional motorized wheelchair controls such as joysticks and, as such, require assistive devices. The most popular method is called "Sip and Puff" and is a system where a user can inhale/exhale into a tube. The strength of the sip/puff, along with the direction of airflow, determines whether a user goes left, right, forward, or backward in their wheelchair. This system has limited control precision since one sip/puff is mapped to one corresponding movement. Additionally, this system can be tiring, lead to health concerns if the tube isn't regularly kept sanitary, and be non-intuitive [3]. Thus, we have decided to explore control methods based on tongue utilization. Using tongues to control wheelchairs for quadriplegics has been explored by various researchers, but there are no current devices for commercial use. The range and ease of motion for a tongue can be compared to a finger, so it is easy to see how these devices can be groundbreaking with regard to the new level of independence they have the potential to bring to quadriplegics.

2.0.1 Prior Research

One such group of researchers from Aalborg University developed a system that requires a ferromagnetic tongue piercing, which is localized using inductive sensor coils embedded in a PCB. This PCB is then installed in the oral cavity of a patient's mouth. This device aims to simulate an analog joystick or trackpad by having a user interact with the roof of their mouth using the tip of their tongue. [4] [5]. Another by Galgotias College of Engineering & Technology uses a series of hall effect sensors embedded in a mouth guard to localize a permanent magnet attached to the tongue [6]. What differentiates these devices from ours is they are installed internally. While utilizing tongue movement may be more intuitive and achieve finer control for users, these approaches still require considerable amounts of maintenance as they must be sanitized regularly and removed while eating. The project closest to production is the Tongue Drive System (TDS) developed by Georgia Tech, a system that uses sensors embedded in two casings that protrude on either side of the face like a microphone boom on a headset to interact with fields generated by a magnet on the tongue [7]. This design allows for greater control provided by utilizing tongue control while being easy to maintain since the electronics are all external. The TDS is the most similar device to the one proposed in this paper. What differentiates the proposed work is that all the processing is done on an MSP432 Microcontroller instead of a larger computer. This results in increased accessibility, ease of use, and significantly reduced cost. However, while the TDS only uses 4 equally spaced sensors to do localization, we use more sensors. This allows for higher resolution in terms of where exactly the tongue is. It also allows for dynamic calibration of different regions to different controls based on a user's comfortable tongue range of motion.

2.0.2 Required Background for Project Execution

To complete this project, we apply material across multiple courses. We use circuit/PCB design skills and a background in electronics from the ECE Fundamentals I, II, and III (ECE 2630, 2660, 3750) to design the sensor and power distribution circuits. Introduction to Embedded Computer Systems (ECE3430), Embedded Computing and Robotics I and II (ECE 3501, 3502), and Advanced Embedded Computing Systems (ECE 4501) are employed to program the microcontroller, process the sensor data into commands, and interface with a computer to relay those commands. Software Development Methods (CS 2110), Program and Data Representation (CS 2150), and Advanced Software Development Techniques (CS 3240) are used across all collaborative software engineering in this project, from firmware tasks for the microcontroller to programming a simulation to demonstrate the commands generated by the device. Electromagnetic Fields (ECE3209) provides foundational knowledge of fields required to complete this project.

3 Project Description

3.1 System Overview

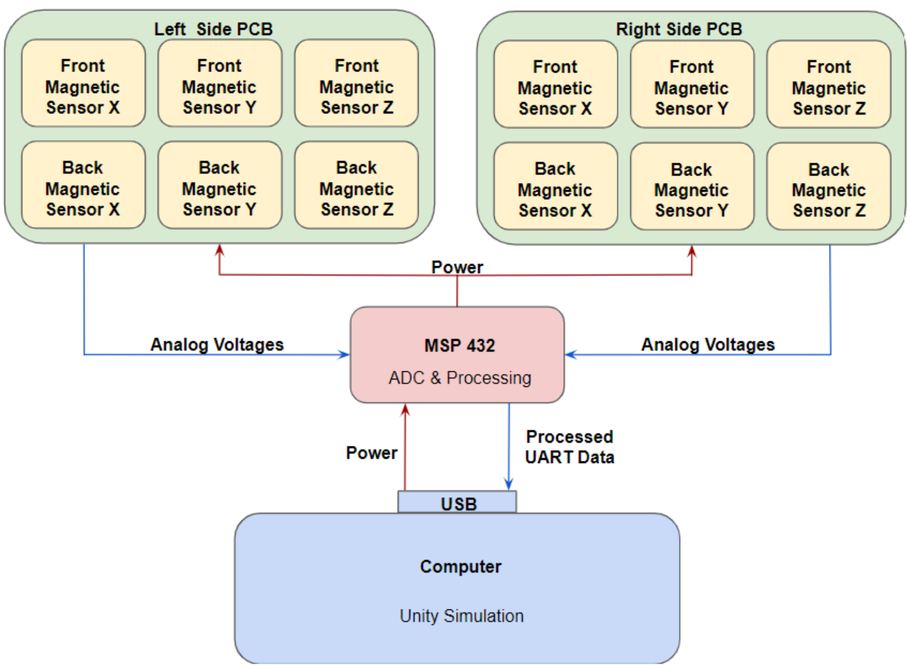

Figure 1: High-Level System Overview

The project comprises three key components: the magnetic sensor drive system, a headset designed to secure the sensor PCBs close to the user's mouth, and 3D simulation software to model wheelchair movement. For the magnetic sensor drive system, a custom PCB is strategically positioned near the mouth area on both sides of the subject's face. These boards house magnetic sensors generating analog signals proportional to the detected magnetic field. The MSP432 microcontroller reads this analog signal, converts it into wheelchair commands using our classification algorithm, and transmits it to a computer. The computer then runs a simulator. The sensor PCBs are affixed to a user-friendly headset for convenient wearability.

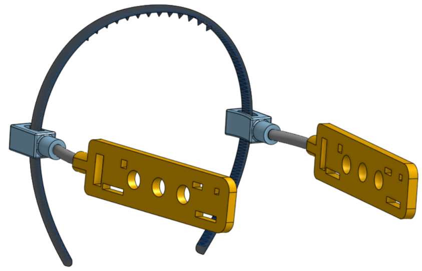

Figure 2: CAD of magnetic sensor system mount: Sensor Array PCB Mounting (Yellow), Flexible Material for Adjustment of Sensor Position (Grey), Headband for Mounting (Blue)

The 3D model to display wheelchair movement takes serial data from the MSP432 and translates that data into an interactive 3D rendering in a Unity Virtual environment, which simulates and shows the movement of the wheelchair in response to user action. The goal of the simulation is to serve as a stand-in for a motorized wheelchair. Since we are working with a reduced budget, the motorized wheelchair is substituted for the simulation. The goal of the project is to have a deliverable that can be plugged into a wheelchair and immediately serve as a source of movement for the user; thus, all processing with data and algorithms is conducted on the MSP432, and then the data is sent as serialized data for the laptop or computer to respond as if it were the motorized wheelchair.

3.2 The Role of Sensors in This Project

We employ linear analog hall effect and magnetoresistive sensors to gauge the magnetic field's strength, enabling us to discern tongue positioning and subsequently control wheelchair movement. The culmination of this project is presented through a 3D simulation depicting the functionality of the wheelchair.

3.2.1 Sensor Input

Hall effect sensors are a type of sensor that detects the presence and magnitude of a magnetic field using the Hall effect. The output voltage of a Hall sensor is directly proportional to the strength of the detected field [8]. Further, magnetoresistive sensors detect the presence and magnitude of an incident magnetic field and change the electrical resistance applied to the output signal proportional to the strength of the field. These changes in voltage or resistance can be sensed and characterized as a way to gain information from the user. [11]

3.2.2 Theoretical Framework

From our initial research, we found conflicting evidence about the rate of magnetic field decay with respect to the radius from the source in meters, where some sources claim that dipole magnets follow the inverse squared law, while others claim that the fields follow the inverse cube law in regards to field strength [10][9]. In order to solidify our theoretical framework, we acquired an axial Gaussmeter and determined that the magnets we are using obey the inverse square law. We also found that for our design to work practically, we would need a magnet with a strength of at least 12000 Gauss. Another consideration that had to be made was the geometry of the agent. The geometry of agents directly affects the concentration of the magnetic fields. We determined that we needed a magnet with high directionality for our applications. This high directionality is necessary for the classification stage of the KNN algorithm.

3.2.3 Sensor Selection

We chose to use an MSP432 Microcontroller for this project. Because of the nature of this board, we initially chose to use digital hall effect switch sensors, as we presumed that analog sensors were inconsistent, making magnet location initialization difficult in the software. However, to increase this device's robustness and implement the KNN algorithm, we decided to go with analog sensors. The sensors we used are the DRV5053VAQDBZR[17] and the RR112-1G42-531[16]. We chose the DRV5053VAQDBZR Hall Effect Sensor based on the specifications of the RR112-1G42-531 Hall Effect Sensor. This is because the RR112-1G42-531 are the only sensors on the market that are axially sensitive in the X and Y directions. These sensors have a sensing range of ± 8 mT. We had to design around this constraint to maintain similar sensitivity scales for later characterization and minimize directional lopsidedness or error. Based on this constraint, we selected the DRV5053VAQDBZR, which is axially sensitive in the Z direction, because it has a similar sensing range at ± 9 mT. This gave us 3-dimensional sensing capabilities with nearly congruent scales, making tongue localization feasible.

3.2.4 Addressing Field Disruption and Exterior Magnetic Noise

In this design, we considered the signal integrity of our I/O traveling down transmission lines, along with the integrity of the magnetic field source that is being used to interface with the sensors. The information that comes from the hall effect or magneto-resistive sensors will travel down a ribbon cable that then serves as an input for the MSP432 Microcontroller. There will be inherent transmission line coupling down the cable. We initially thought that we may have to do signal loss correction for the I/O from the sensors. However, this was found not to be necessary because we were using a classification algorithm to determine instruction based on I/O from the sensors, not hard-coded values.

3.2.5 PCB Design

For the final PCB design, we used Analog Hall effect sensors that would feed data into the ADC of the MSP. We used analog sensors oriented in the X, Y, and Z directions for full-dimensional characterization. This was needed to increase the robustness of the KNN classification that dictates user movement. We used DRV5053VAQDBZR Hall Sensors for the Z-axis sensing, then used RR112-1G42-531 Hall sensors, as the RR112-1G42-531 sensors gave X and Y dimension capabilities. The sensors are placed in 'clusters' that are used to collect data and send it to the MSP. These clusters are placed 4.5 cm apart. We came to this measurement after the first iteration of the boards was received. After testing, it was found to be more ergonomic and algorithm-friendly to reduce the space between sensor clusters from 6 cm to 4.5 cm. Each sensor has a very low current draw (3 mA for the DRV5053VAQDBZR Hall Sensors, 1.5 uA for the RR112-1G42-531 Hall sensors), so the MSP432 could supply enough current to these devices for them to operate. They also have a voltage range from 0 volts to about 2.5 Volts, which gives us enough resolution for the ADC to enumerate the values for classification. According to the datasheets, the sensors need a voltage supply minimum of 2.5 Volts. Because of this, the boards are powered from the MSP432, as the microcontroller meets the Current and Voltage requirements for the sensors to operate. When using the first iteration of the sensor boards, distortion and interference between wires were problems that did not arise, though they were anticipated. The final board size is 71.5 mm x 21 mm and features auxiliary testing points for troubleshooting purposes. A 3D render and the floorplan for the board are shown in Figures 3 and 2, respectively.

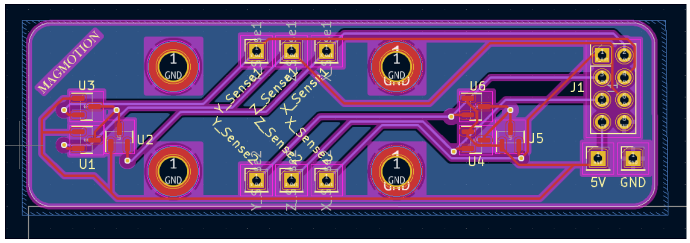

Figure 3: PCB FloorPlan

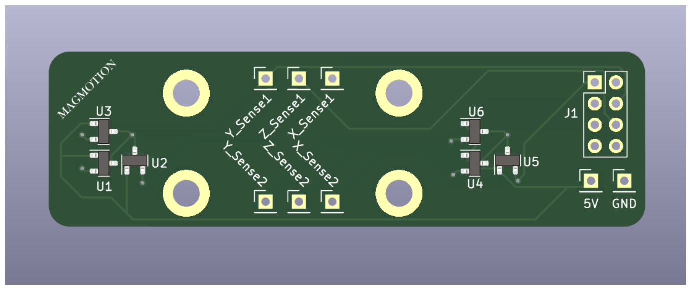

Figure 4: Hall Effect PCB 3D Render

3.2.6 Magnet Choice

We chose the Alliance ALC6416 cylindrical magnet for tongue direction detection. It is rated for 14800 Gauss. In order to increase its directivity, we change the total geometry of the magnet by stacking 3 of them on top of each other. The effect this made was a magnet with greater axial magnetic strength, improving the ability for field detection and, hence, classification ability. We experimented with rectangular magnets to produce the same effect. However, we found that the stacked cylindrical magnet geometry worked best for our applications.

3.2.7 Test Equipment and Validation

We used multiple pieces of equipment to test/observe the sensors' signal output and characterize the agents. We used the National Instruments (NI) Virtual Test Bench Software along with the NI Virtual Bench Oscilloscope and signal generator to observe the analog output of the hall effect sensors. To characterize the magnets, we used a PASCO 3-Axis Magnetic Field Sensor to measure the strength and field distribution of the magnets for testing. We used PASCO SparkVue software for data collection from the magnetic sensor. These instruments were provided to us for free by either the University of Virginia NI Lab or the ECE department faculty.

3.3 3D Modeling

3D modeling plays an important role in the development process for approaching a solution for our project. It serves as a bridge between the virtual and physical worlds, allowing us to visualize, design, and create a prototype of a helmet that allows for a fastened state of sensors to read input from the magnet on the individual's tongue.

3.3.1 Conceptual Design Requirements

The end goal is a device that can mount the PCBs along the user's jaw. This piece has to be durable and retain shape between uses, so calibration data remains true. It also has not to be bulky and should minimize the amount of space that is covered to better facilitate human connections. The design essentially fits someone's head and then has two whiskers sticking out as prongs with some mounting adapter to secure the PCB to hold all of the sensors for gathering and recording data from the user. Lastly, the general design of this headset is supposed to fit most head shapes.

The end result is shown in Figure 2. It is a flexible modified hairband that allows the headset to fit multiple shapes. Mounted on this hairband is an adapter to which a 20-gauge steel wire fits, and on the other side of that wire is a PCB adapter to which a PCB is screwed and attached. The steel wire's purpose is to adjust the PCB position on the face to accommodate multiple face shapes.

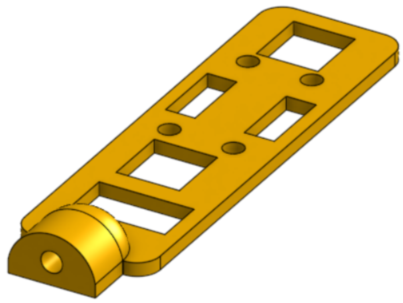

Figure 5: Final Design for Hairband to Wire Adapter

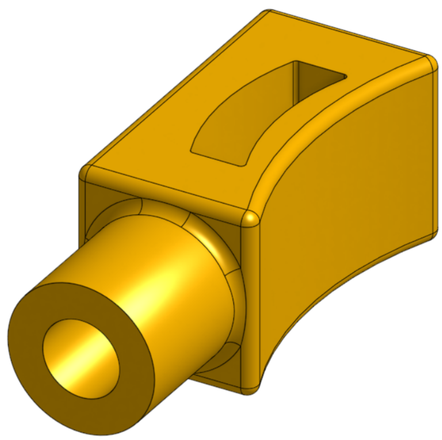

Figure 6: Final Design for Wire to PCB Holder Adapter

3.3.2 Software Tools

The software used on this prototype for the 3D model is a Computer-aided design service, Onshape. This service allows us to produce and refine our product for the headpiece. The parametric modeling, rendering, and simulation tools offered can conceptualize all our concerns before we proceed with building the headpiece via a 3D printer.

3.3.3 Prototyping and Iterative Design

Prototyping involves a series of discussions between group members, taking into consideration all aspects of development, such as how expensive it is, how large, how the customer feels about its design, and how the design makes the product better. Going through multiple variations of design, such as attaching a PCB board in front of a person's face for sensing, is considered, but it eventually is designed to maintain two identical but separate boards on either side of the face to maximize resolution within the mouth. After deciding on this approach, multiple iterations of design are made on the PCB adapter alone to maximize durability and aesthetics while minimizing size. Another important consideration is how the design fares in the manufacturing process. Since the PCB adapter is 3D-printed, designs that are flat on one side are able to be printed more cleanly and with higher quality.

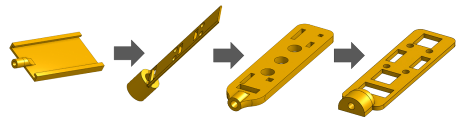

Figure 7: Iterations of PCB Mount designs

3.3.4 Material Selection

The type of material we pick for our helmet must have all of the following requirements: It must be durable, light, and cost-effective. It must be durable because it must be able to withstand impacts and provide long-lasting use for the user. The material should also be light because wearing headgear that may be heavy can be exhausting if you are wearing it all day. The goal of being lightweight is to counteract that issue by not straining patients with additional weight in the process. For the prototyping process, we choose PLA for accessibility purposes, but a more consumer-facing version of our product may use ABS, which better meets the above requirements. We chose 20 gauge steel wire for the adjustable material because of its malleability and ability to be easily manipulated yet retain shape.

3.3.5 Accessibility and Ergonomics

The accessibility and ergonomics of the headset were briefly addressed in previous sections, but the main purpose of the way it should be designed is to be able to fasten the sensors for reading input from the user, the comfort of the helmet so that it does not feel as if the user is constrained, and it should also avoid blocking as much of the user's face a possible so that others are still able to see the user and interact face to face without a metal sheet blocking that interaction [12].

3.4 Software and Firmware

The software and firmware of the project, in essence, is the component that is doing the sensor reading, processing, converting to meaningful movements, and simulation. Its purpose is to interpret sensor data and translate it into a sequence of meaningful commands that then move a simulated wheelchair.

3.4.1 Microcontroller State Overview and GUI

The firmware has nine states: Calibration Stand-by, a calibration state for each of the six commands(6 states total), a Reset state, and a Normal Operation state. This finite state machine described below works in tandem with a GUI used to calibrate the six different commands and to switch in and out of Calibration and Operation Mode. The GUI has a "Calibration Mode" switch, calibration buttons for each of the six commands, a command LED for each command, a command display, and a "Done" button. The calibration mode switch allows the user to enter calibration mode from operation mode. The calibration buttons allow the user to calibrate the corresponding command. The command LEDs indicate if the corresponding command is calibrated. The command display is used to display a command that corresponds to the current tongue position; this only occurs when the system is in calibration mode, and all the commands have been calibrated to help with recalibration. Finally, when the "Done" button is pressed, if all the commands have been calibrated, then the system changes from calibration to operation mode.

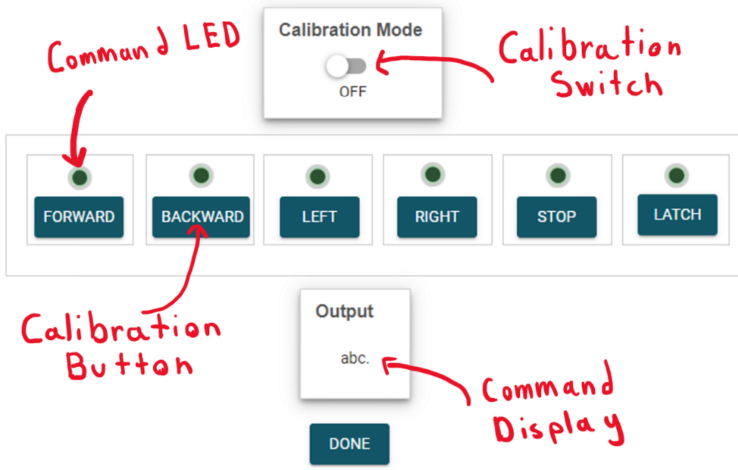

Figure 8: Calibration GUI

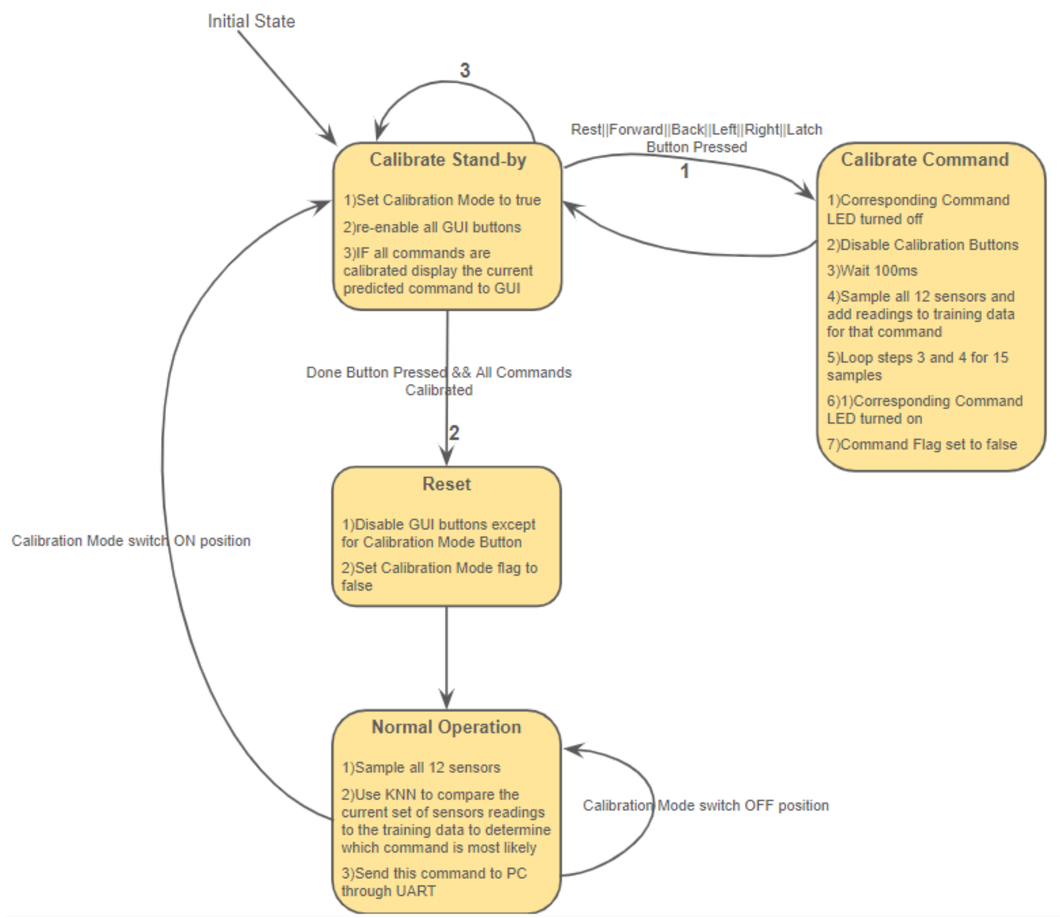

Figure 9: System FSM

Note: In the code, the Calibrate Command state is separated into six states, one for each command. The diagram differs for ease of understanding.

3.4.2 FSM States

The FSM starts in Calibration Stand-by Mode. In this state, the system sets the calibration switch to the ON position, enables all the calibration buttons to be pressable, and if all the commands are calibrated, indicated by the command LEDs, then it displays the current predicted command based on the current tongue position, in the command display. The system stays in this state until one of the calibration buttons is pressed or when all the commands are calibrated(indicated by all the command LEDs being on) and the done button is pressed.

When one of the calibration buttons is pressed, it switches to the Calibrate Command state. The first thing that happens in this state is to turn off the corresponding command's LED in case this command has already been calibrated. Next, the system disables all the buttons so that the user doesn't try to calibrate another command or leave the calibration state while a calibration going on. Afterward, the device enters a loop of waiting 100ms, then sampling all the sensors, doing this an overall of 15 times. All of these 15 samples(12 ADC readings each) are stored a specific part of training data list that corresponds to the command that's being calibrated. This training data list is used in our classification algorithm to map a set of sensor readings to a command during normal operation.

Once all the commands are calibrated and the "Done" button is pressed, the system moves to the reset state, which disables all the buttons except the calibration switch from being pressed and flips the flag that indicates calibration mode to false.

Finally, it reaches the Normal Operation state. During this state, the system samples all 12 sensors and sends unclassified points to the KNN classification algorithm. Using the training data list, the algorithm is able to determine what command the current reading corresponds best to and send this command to the computer simulation through the UART protocol. Once that processed data is sent, the system then samples all 12 sensors again and repeats the process. If the user wishes to get back to the calibration state, the calibration GUI switch can be flipped to on.

3.4.3 Sensor Sampling

The MSP432 is responsible for sampling analog readings from the sensors. This is done by setting up the Analog-to-digital(ADC) module for 12 pins one for each sensor.

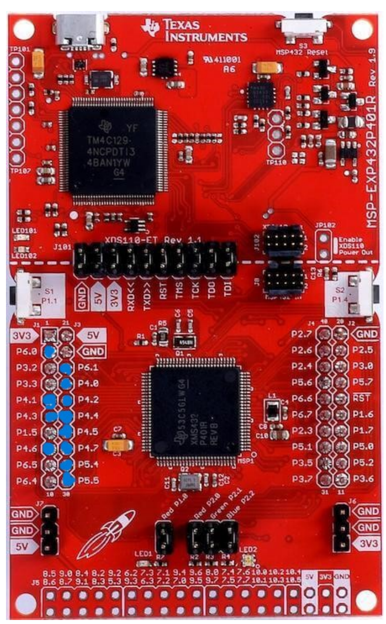

Figure 10: ADC Pins Used

The output from the sensors was not clean enough for our use, so after sampling, we had to do some signal filtering. We decided to perform digital filtering after sampling the sensors instead of using the ADC readings directly. An averaging filter was implemented to act as a low-pass filter and smooth out the sensor readings. The number of samples needed to make the sensor readings reliable enough was determined experimentally to be around 128 samples. This means each time the system asks for a sample of the current sensor readings, it uses the ADC to sample each of the 12 sensors 128 times and returns the average for each of the 12 sensors. We looked into using an analog low-pass filter instead of a digital one, but the digital filter worked well enough for our purpose and didn't require extra hardware components, so we committed to this method.

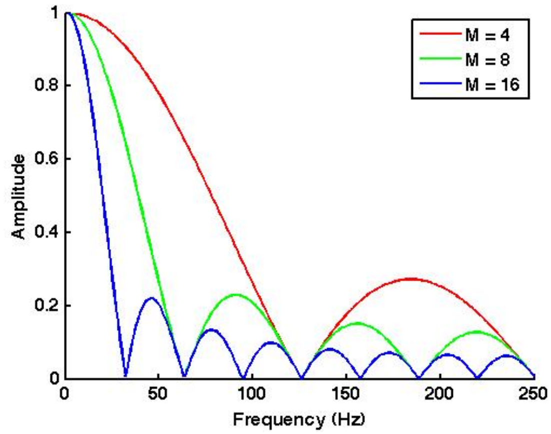

Figure 11: Frequency Response of Averaging Filter

3.4.4 Data processing and Algorithm Implementation

During the Normal Operation state, the microcontroller uses an algorithm that takes the current sensor reading and maps it to one of the commands by comparing to the training data. The algorithm that we used is the K-Nearest-Neighbors (KNN) algorithm.

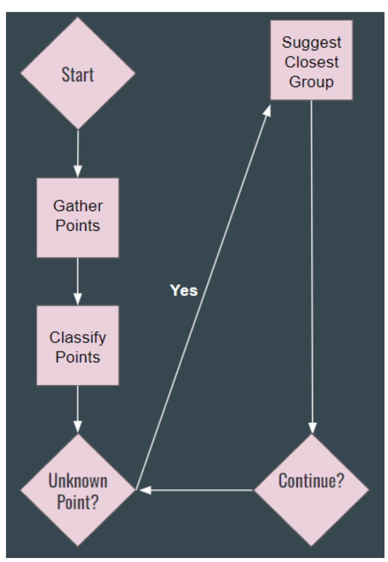

Figure 12: KNN Workflow

This algorithm is effective because it is a supervised learning algorithm that can allow the user to define where they wish to have particular commands mapped. This allows the user to customize their inputted data for comfort and command preferences. The KNN algorithm involves hyperparameters that can influence its performance so that it can make better, more concrete decisions. This parameter, known as k, allows users to balance between precision and robustness while determining the user's tongue position in a feature space, in this case, the mouth.

Every sensor sample comprises 12 ADC readings, one for each sensor, forming a vector with 12 elements. Each entry in the training set is likewise a 12-element vector, labeled with the corresponding command assigned during calibration. During the operational mode, when the system samples, it collects all the points from the training data and arranges them in ascending order based on their Euclidean distance from the currently sampled, unclassified sensor reading. Following this sorting process, the algorithm examines the first k samples from this ordered list of training data and identifies the command that occurs most frequently within these k samples. The unclassified sensor reading is then attributed to this most frequently occurring command.

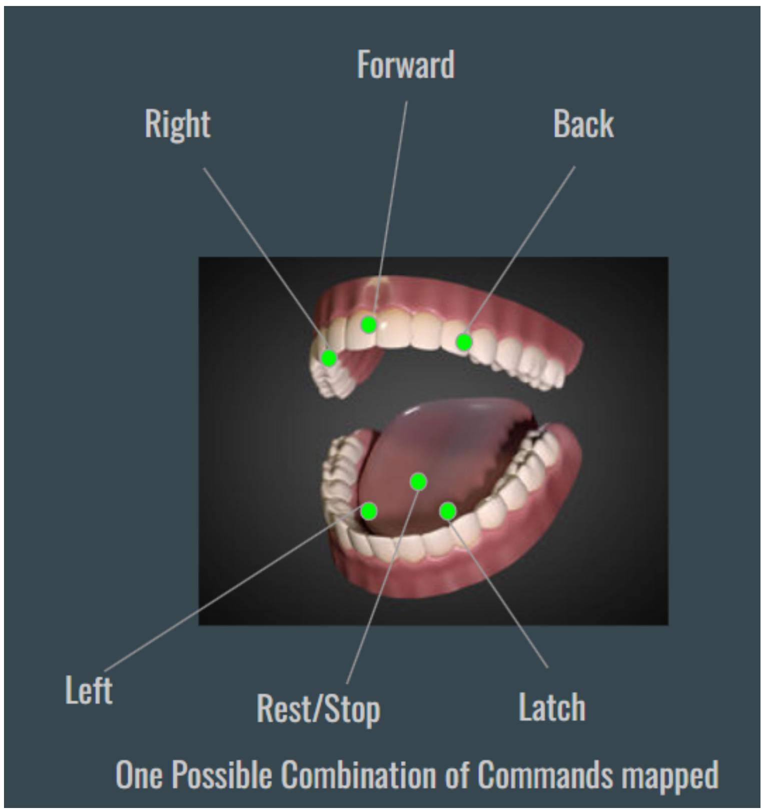

Figure 13: Command Centroids Mapped

Once the microcontroller decides which direction the user had requested using the KNN Algorithm, it sends a byte corresponding to the predicted command through the UART protocol, which a computer receives through the same protocol. UART is an asynchronous serial communication protocol. Asynchronous means there is no clock signal to synchronize the output bits from the transmitting device going to the receiving device[14]; this allowed devices running at different speeds to communicate. We used a serial communication port on the computer to set up UART communication by connecting the MSP432 and computer with a USB to Micro-USB cable.

3.4.5 Computer Simulation

Once the Computer receives the serial data from the microcontroller, it decodes it and processes it as a movement within a simulated environment. The simulation environment we elected to use is with Unity. Unity offers many benefits for simulation purposes, such as multiple free models for use as game objects as well as having a vast amount of documentation and video tutorials. In addition to this, it also has a built-in physics engine for any project using Unity, which makes integration and system testing easier for the final deliverable. In terms of the project, it allowed for the system to be tested by a simulated electric wheelchair that serves as a stand-in for the actual electric wheelchair. The final deliverable of the system can communicate with the simulation running on a computer via UART, where the computer can take data from the microcontroller and perform the specified operation on a virtual wheelchair in the simulated world as a substitute for a physical electric wheelchair. The resulting simulation is a track that is able to determine a series of defined inputs from the microcontroller where the simulation runs the simulated electric wheelchair through a track that tests the abilities to move forward, backward, and rotating clockwise and counterclockwise as well as a stop and latch command to be able to speak.

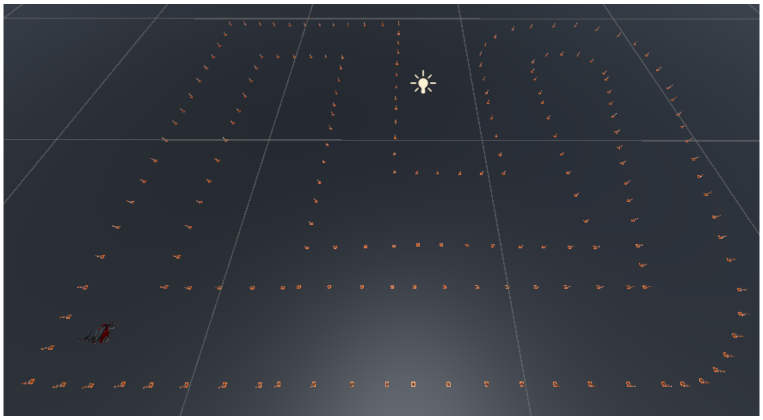

Figure 14: Track of the Simulation

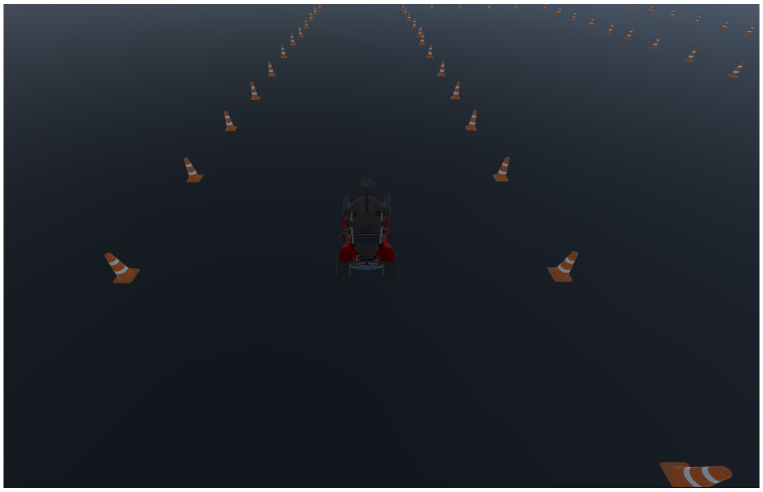

Figure 15: Point of View of Simulation Running

3.4.6 Error Handling and Testing

There are many errors and validations that we must consider handling to ensure that our microcontroller can work and maintain itself without the use of a computer monitoring its inputs and outputs. Items such as sensor data validity and error code logging are important for the development of the microcontroller, resulting in a more defined and robust intellectual motor for the project.

3.5 Project Testing

A significant part of the design for the project is testing, testing is a pivotal step in the project's lifetime to ensure that our subsystems are working as intended. Our primary subsystems will be the PCB and Sensor, Microcontroller, and Simulator.

3.5.1 PCB and Sensor Testing

The PCB and sensor testing were conducted to assess the functionality, reliability, and safety. In order to achieve this, a series of tests was carried out. The first test involved verifying whether the sensors were able to properly respond to changes in a magnetic field from a magnet. This was done by testing variable locations of the magnet and the Hall effect sensor in a manner that activated the Hall effect sensors in different patterns. Additionally, tests were performed to assess the reliability and longevity of the sensors in the presence of the magnet, with simple sustainable tests conducted to determine if the sensors remained active over an extended period. Finally, efforts were made to ensure that the sensors complied with safety standards for medical devices. Concerns such as heat generation, electromagnetic interference, and shock were investigated to ensure compliance.

3.5.2 Microcontroller Testing

The testing plan for the microcontroller was conducted as a series of four tests: Algorithm validation, communication testing, power testing, and error handling. Testing the algorithm for translating sensor data into control commands to send to the computer for emulation was crucial for communicating with the other subsystems. By implementing and adjusting the algorithm, it was verified that the microcontroller could properly differentiate against variable tongue locations. Calibration was also a part of the algorithm to generate tongue localization for algorithm testing. A microcontroller was used by itself with a hall effect sensor and a magnet to generate data to send to the microcontroller for processing. Communication testing was also a crucial part of the microcontroller, ensuring its ability to interact with the sensor and computer, which was important for communication between subsystems. Error handling was necessary testing to ensure that malformed data input or unexpected input data did not break the microcontroller and that it continued to work as intended afterward.

3.5.3 Simulator Testing

The simulation will serve as a stand-in for the actual wheelchair. As a result, it is important that the simulation is functional, compensates for edge cases, and integrates with the microcontroller. To test the simulation, we will use Unity software to generate a real-world scenario, such as grounds on UVA where a wheelchair can move based on the inputs from the microcontroller. Ensuring the simulation is able to read the serialized commands sent to the simulation is imperative for the project. Edge cases should also be tested such as rapid movements or tongue movements that are not as natural. Offering a tick rate would be able to account for the rapid movements and complex tongue instances. Finally, the simulator can integrate with the microcontroller to ensure that the inputs from the microcontroller can properly represent the user's movements with their tongue inside of the Unity simulation.

4 External Considerations

4.1 Navigating Challenges

4.1.1 Resource Constraints

Resource constraints were a tangible consideration in this project's research and development phase. This problem was encountered when trying to obtain certain components and agents for this project. Concerning magnet procurement, some constraints showed up when trying to buy magnets of varying strengths and geometries during the testing phase of this project. This is because trying to obtain a wide range of desired magnets to do ultra-rigorous testing is neither cost nor time-effective and was not even possible due to the limited selections of magnets that are available in the marketplace. To combat this, we had to set up experiments with the magnets we did have and then extrapolate expressions for a general case. This allowed us to figure out what minimum strength magnet we needed for this project. Further, finding sensors that could meet our 3-dimensional characterization requirements was difficult. The only sensor that had Hall effect sensitivity in the X/Y dimensions was the RR112-1G42-531 Hall Effect sensor. Due to this constraint, we had to build the rest of the system around the limitations of this device and the resource limitations of the online marketplace.

4.1.2 User Adaptability

User acceptability and adaptability were other restrictions that we had to be mindful of. A recent anecdote from a Georgia Tech report brought up the idea that quadriplegics still wanted the front of their faces shown to the world while operating their wheelchairs. We wanted to adhere to these considerations to make the product as user-friendly as possible. In accordance with that, our physical design went through multiple iterations.

4.1.3 Modeling Accuracy

Another limiting factor to consider was actually trying to model the interior of someone's mouth. It was impossible to account for all cases because everyone's skull is different. Because of this, we had to make approximations. We also had to account for the latent magnetic field that the earth creates. This was and could be a detectable field that might affect sensor performance.

4.2 Tailoring the Solution

4.2.1 User Variability

Everyone's face is different, so we must find ways to account for that in our software initialization and sensor placement. This is important because we need to ensure that the tongue drive system will work and be applicable to all face structures and variability. Exploring possible customization options can be used to be inclusive for all individuals using this system.

4.2.2 Structural Integrity

We need to make sure that our design has good structural integrity so users don't destroy it. It is possible that users may apply a variable amount of force to the device as a way to learn how the system works so it is important that the design is made to sustain general wear and tear and still be functionally complete.

4.2.3 Modularity and Ease of Replacement

If a user broke the item, we made it so pieces could be easily replaced or exchanged. By having a modular design process, we hoped to only need to replace certain parts that were approaching unacceptable levels of operation compared to replacing the whole system when a part became defective, thus making it more economically feasible for the user and product producer.

4.2.4 Prototyping in Real-Life Scenarios

We also had to find a way to prototype the magnet safely in a real person's mouth. Testing a prototype with real users was necessary to ensure that the magnet could be safely and comfortably placed in a user's mouth, be it from a piercing, wrapping around the tongue, or a medical adhesive on the tongue. Once figured out, we could simulate real-life scenarios and explore new factors such as comfort, ease of use, speaking while using, and a magnet's impact on maintaining oral hygiene.

4.3 Health, Safety, and Affordability

4.3.1 Health and Safety

We initially found magnets over 4000 G dangerous to the human body [15]. However, the magnet geometry is such that the magnetism seen by the human body is negligible, as the field drops to a harmless level about 1 cm from the magnet. Further, most of the materials that are in a human mouth are either slightly diamagnetic or paramagnetic. Materials that would fall under these categories are silver (such as fillings), iron in blood, water, etc. However, the effect is not enough to cause any interactions with these materials.

4.3.2 Material Toxicity

We also had to ensure that the materials the magnet was made out of were nontoxic. A nontoxic material for the magnet was necessary for the welfare of the user to ensure that the product they were using also did not inadvertently cause health risks to the user. Neodymium iron boron magnets in their solid form were found to possibly have low to moderate toxicity, though their long-term and short-term effects had yet to be fully researched [18]. The elements of this compound were not inherently toxic; however, corrosion, inhalation, and breakage were health hazards to be aware of when handling these magnets.

4.3.3 Affordability

Lastly, we wanted this to be affordable, so ensuring the price kept this option viable was important [11]. It was important that the system's design choices were consciously being considered for cost-effectiveness and optimal resource efficiency. These choices were intentional such that it was possible for any user who needed this system to afford one within an acceptable and viable price range.

4.4 Standards and Regulations

4.4.1 IPC Standards

We abided by IPC Standards for Board Fabrication. It was important to consider IPC standards to ensure the best quality board design and electronic components were used for the system. Compliance with this standard instilled a sense that the device would operate effectively and safely for users.

4.4.2 Medical Device Regulations

Depending on how we define the usage of the tongue-driven wheelchair mount, it is possible that we may run into a concern with medical device regulations. Even if this is just assistive technology, we may need to comply with regulations such as those from the Food and Drug Administration (FDA) or other proper authorities so that our device is able to be approved for safety and commercial use by any user who wishes to use it.

5 Expectations

The project aimed to emulate the tongue drive system to justify it as a viable solution for quadriplegics and those suffering from motor impairment. This project depended on four different divisions of components: Sensor Functionality, Microcontroller Programming, User Adaptability, Safety, and Health.

5.1 Sensor Functionality

The sensor functionality is expected to accurately respond to the magnetic fields created due to the user's tongue inputs. The sensors should then be able to send this data to the microcontroller.

5.2 3D Modeling/Mechanical functionality

This project's mechanical and modeling aspects had to be completed for the physical establishment and installation of the sensors to be placed and used for data input from the tongue's magnet. It was essential that the sensors be securely fastened to the user's head so the calibration data remains true even after multiple iterations of removing/putting the sensor mount from the user's head.

5.3 Microcontroller Programming

The microcontroller is imperative for success since it is the logic unit responsible for completing the data processing from the sensors and serializing this data after running it through an algorithm for the computer to produce a movement input instruction from the microcontroller.

5.4 User Adaptability

It was important that the implementation of the algorithm for the microcontroller and the design of the modeled headpiece to be held with consideration from all views and able to be intuitive and adjustable as needed for any users.

5.5 Safety and Health

It was crucial that the safety and health regulations be considered throughout the lifetime of the project. By meeting all the criteria of industry standards and regulations, we ensured that our product will be able to be used safely and without concern.

6 Final Results and Reflection

6.1 Results

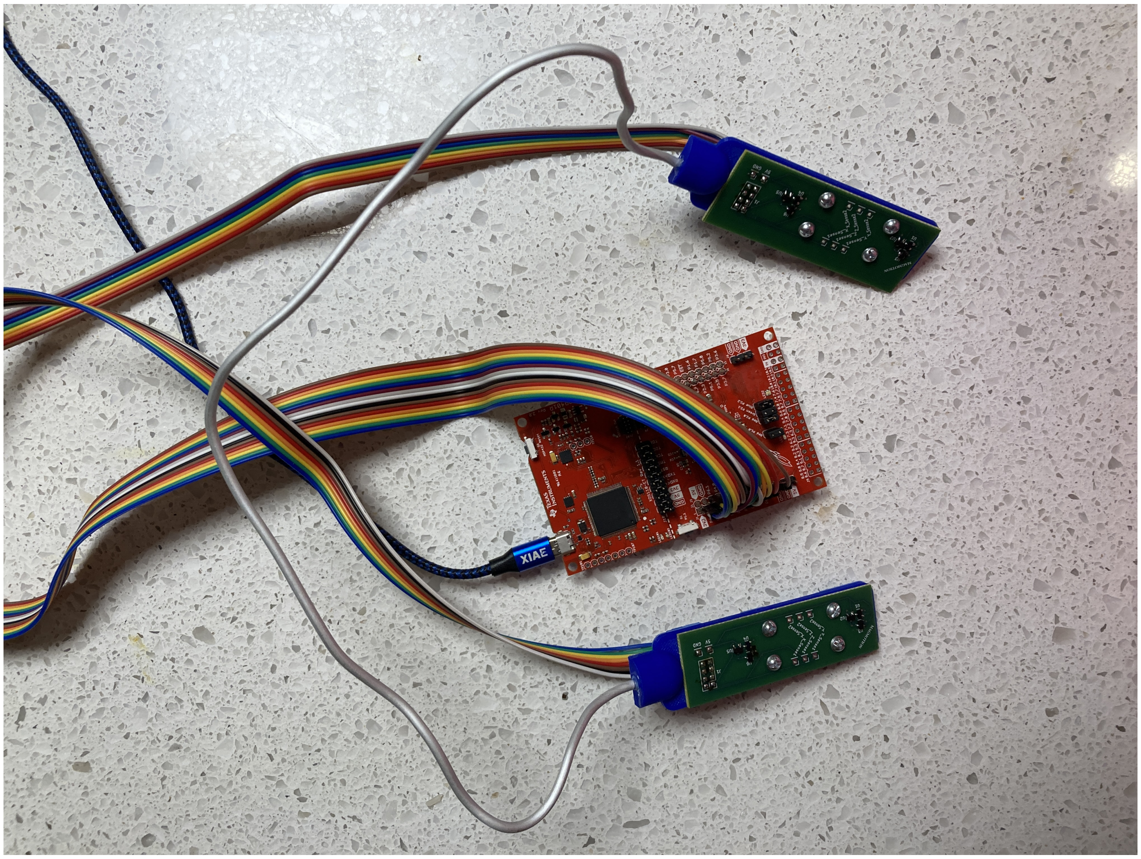

Figure 16: Full Picture of System

All components have been successfully interconnected, including Sensor design, 3D modeling, Software and firmware, and simulation. Following the integration of these elements, the system demonstrates its capability to accurately classify sensor data collected from tongue movement monitoring. The microcontroller, housing the KNN algorithm, processes this data to determine the most suitable command corresponding to the tongue's location. Subsequently, the information is transmitted via UART to the computer simulation, where the command manifests as a visual response within the simulation, showcasing effective and efficient operation.

6.2 Reflection

Overall, the system can work, but it has some limitations that we did not expect to happen until testing. One of the limitations that the system has right now is that it reads data too fast. Hence, when a user wishes to move forward and then rotate clockwise, the user may unexpectedly move backward for a brief moment because the movement the tongue used to reach the rotate clockwise (depending on the user's customized controls) passed through the backward command. Although brief, it is noticeable. We attempted multiple fixes, such as sampling slower, but sometimes the sample would catch the command that was not desired, which made the simulation fail to work as intended. Another approach was to have a list of commands over a span of time where the commands sent are the majority inside of the list. This approach did, in fact, work, but its caveat was that its computing power was demanding and failed to run at a competitive speed, which in turn served as an ineffective solution to the limitation.

While the limitation is minimal, it is still noticeable, and with consideration that this system is intended for real electric wheelchair integration, it serves as an important issue to address for a later iteration of the system.

7 Cost

The total cost to experiment, test, develop, integrate, and build the system came to approximately \$428. The breakdown of this cost is shown below in Table 1.

Component	**Cost**
Magnets	\$81.85
Sensors	\$90.78
Other/Building Materials	\$137.94
MSP432 Microcontroller	\$17
PCBs	\$100
Total Cost	**\$427.57**

Table 1: Cost Breakdown of Components for the System

8 Timeline

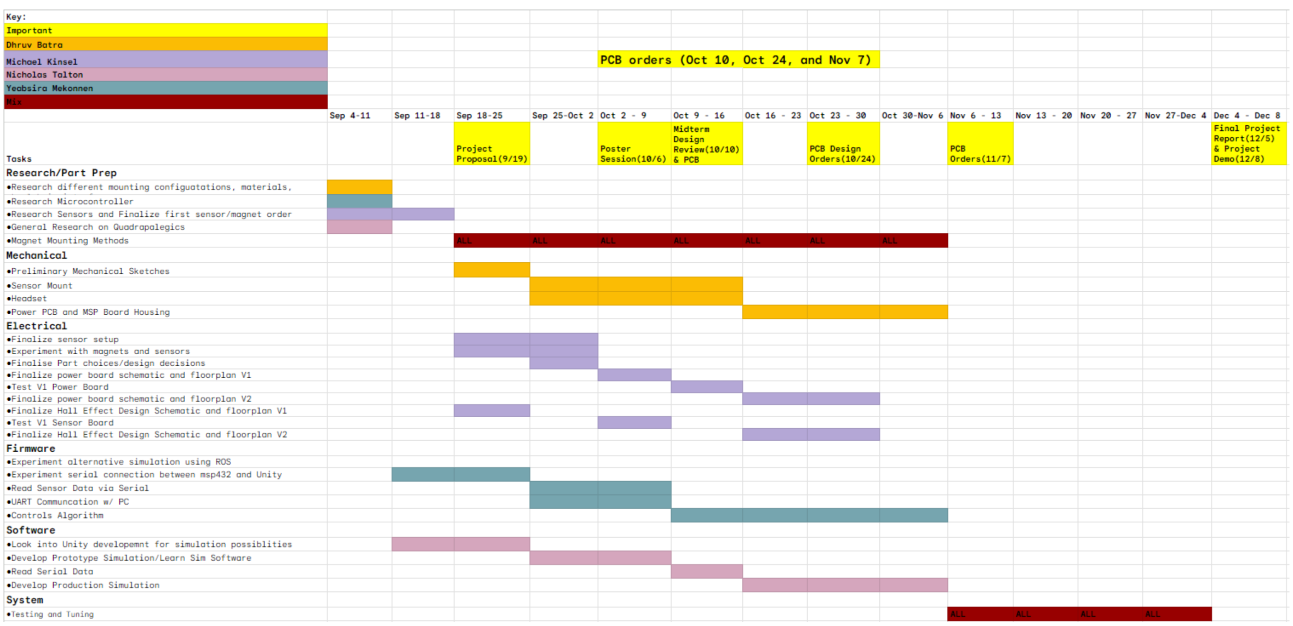

Figure 17: Gantt Chart

The above Gantt Chart shows the timeline for the project, including important deadlines. The Tasks are divided into six major sections: Research, Mechanical, Electrical, Firmware, Software, and System. Each individual in the group is assigned a color within the chart to make identifying priorities easy.

8.1 Division of Labor

Dhruv Batra was the primary lead responsible for the Mechanical tasks, with a secondary focus on the Firmware task. This took advantage of his background using CAD software from ENGR 1624 as well as embedded classes

ECE 4501 and ECE 3430. Yeabsira Mekonnen was primarily responsible for the firmware tasks with a secondary focus on the software tasks. He did this by applying embedded knowledge from ECE 4501 and ECE 3430 and previous uses of the Unity Engine. Nicholas Talton was primarily focused on the Software tasks but helped with Mechanical and Firmware tasks as needed. He completed these tasks by learning more about the Unity Engine and taking advantage of embedded knowledge from ECE 3430 and CAD experience from ENGR 1624. Finally, Michael Kinsel was primarily focused on the Electrical tasks, which were the PCB Design and the EMF knowledge needed to work with the magnetic field sensors. Michael pulled on his knowledge from ECE 3209 as well as the Fundamental course for his tasks.

8.2 Parallel Tasks and Serial Tasks

Up until the development of the controls algorithm, tasks in the separate sections were done mostly independently, with some connections between the Mechanical and Electrical sections as well as the Firmware and Software Sections. The exception to this was the system test and tuning, which was done after each section's tasks were completed. All the Research tasks were in parallel with all the other tasks within and outside of the Research section. For the mechanical section, the preliminary sketches for all the parts were done in parallel with all the other tasks. A set of serial tasks were the development of the sensor mounts and the design of the sensor PCB. The designing and production of the sensor mounts had to follow the design of the sensor PCBs because specific measurements of this board were needed to design the mounts. Once the sensor PCB design v1 was done, then the sensor mount and headset were developed independently of the other task. Lastly, the PCB and MSP432 Board Housing development had to follow after the power PCB design was finalized.

For the Electrical Section, the sensor PCB design had to precede the mounting of the sensor mounts. Also, the power PCB design had to be finalized before the housing for them was designed. Other than those two, the other tasks could proceed in parallel with each other. For the firmware section, once the development of the controls algorithm had started, it needed a usable prototype of the headset and sensor mount, prototype simulation code, and prototype PCBs as early in the development as possible. This was because it allowed for an easy way to test the classification algorithm if all the components for the final project were being used as the classification algorithm was being made. Before this point in the Firmware section, tasks were done in parallel with all other tasks.

For the Software section, the production of the final simulation software needed serial input from the MSP432 board to test, so this task needed to follow the achievement of Reading Sensor Data via Serial and UART Communication with the PC. Afterward, the final version of the simulation was worked on in parallel with all the other tasks being done. Finally, System testing needed to occur after all the tasks on the other established sections were done since it tested the whole system.

References

[1] J. E. Lasfargues, D. Custis, F. Morrone, J. Cars well, and T. Nguyen, "A model for estimating spinal cord injury prevalence in the United States," Spinal Cord, vol. 33, no. 2, pp. 62–68, 1995. doi:10.1038/sc.1995.16

[2] C. C. medical professional, "Quadriplegia (Tetraplegia): Definition, Causes & Types," Cleveland Clinic, https://my.clevelandclinic.org/health/symptoms/23974-quadriplegia-tetraplegia (accessed Sep. 20, 2023).

[3] S. Menon, A. Nambiar, A. Pariyarathu, and K. M, "Tongue Driven System," International Journal of Engineering and Technical Research, vol. 3, no. 4, Apr. 2015.

[4] M. E. Lund et al., "Inductive tongue control of powered wheelchairs," 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, 2010. doi:10.1109/iembs.2010.5627923

[5] E. R. Lontis, B. Bentsen, M. Gaihede, F. Biering-Sorensen, and L. N. Struijk, "Wheelchair Control With Inductive Intra-Oral Tongue Interface for Individuals With Tetraplegia," IEEE Sensors Journal, vol. 21, no. 20, pp. 22878–22890, 2021. doi:10.1109/jsen.2021.3111549

[6] M. Jain and H. Joshi, "Tongue Operated Wheelchair for Physically Disabled People," International Journal of Latest Trends in Engineering and Technology, vol. 4.

[7] J. Kim et al., "The Tongue Enables Computer and Wheelchair Control for People with Spinal Cord Injury," Science Translational Medicine, vol. 5, no. 213, 2013. doi:10.1126/scitranslmed.3006296

[8] "Hall Effect." http://hyperphysics.phy-astr.gsu.edu/hbase/magnetic/Hall.html (accessed Sep. 19, 2023).

[9] "How does distance affect magnetic force? — Socratic," Socratic.org. https://socratic.org/questions/how-does-distance-affect-magnetic-force (accessed Sep. 19, 2023).

[10] "Experiment of The Month — Millersville University." https://www.millersville.edu/physics/experiments/023/ (accessed Sep. 19, 2023).

[11] Magnetoresistance." Chhatrapati Shahu Ji Maharaj University. [Online]. Available: https://gyansanchay.csjmu.ac.in/wp-content/uploads/2021/11/303-L17.pdf

[12] J. Kim et al., "The Tongue Enables Computer and Wheelchair Control for People with Spinal Cord Injury," Sci. Transl. Med., vol. 5, no. 213, Nov. 2013, doi: 10.1126/scitranslmed.3006296.

[13] Texas Instruments, "MAGNETIC SWITCH UNIPOLAR TO92-3," DRV5023AJELPGMQ1 datasheet

[14] "UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter — Analog Devices." https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html (accessed Sep. 19, 2023).

[15] "Electromagnetic Fields Factsheet." Iowa State University of Science and Technology, 2022. [Online]. Available: https://www.ehs.iastate.edu/sites/default/files/uploads/publications/manuals/emf.pdf

[16] "REDROCK® RR112-1G43-531/532 TMR ANALOG MAGNETIC SENSOR." [Online]. Available: https://cotorelay.com/wp-content/uploads/2020/10/RedRock-RR112-1G42-1G43-531-532-Datasheet.pdf

[17] "DRV5053 Analog-Bipolar Hall Effect Sensor." Texas Instruments. [Online]. Available:

[18] K. T. Rim, K. H. Koo, and J. S. Park, "Toxicological Evaluations of Rare Earths and Their Health Impacts to Workers: A Literature Review," Safety and Health at Work, vol. 4, no. 1, pp. 12–26, Mar. 2013, doi: 10.5491/SHAW.2013.4.1.12.