EarEEG Final Report

Evan Goble
Jeremy Herwig
Dillon Hunneke
Tyler Stuessi
Arthur Vidineyev

Customer: Dr. Qi

Team 18
ECE 402
April 25th, 2017

Table of Contents

Table of Contents **2**

Executive Summary **3**

Introduction **4**

Requirements **4**

Changelog **5**

Design Process **5**

- Research and Investigation 5
- Decomposition 7
- Selected Solution 7
  - Hardware 9
  - Software 10
- Prototyping of Solution 10

Results **12**

- Hardware 12
- Software 12

Lessons Learned **13**

- Hardware 13
- Software 13
- Future Work 14
- Relevant Coursework 14

Team Contributions **14**

Signatures **16**

Executive Summary

In this project, we attempted to create a proof-of-concept for an in-ear EEG device, which we called the EarEEG. This idea was proposed by Dr. Qi of the University of Tennessee EECS department, and we have worked with her throughout the semester to accomplish our goals.

As far as goals were concerned, we aimed to create a device that was built on top of the NeuroSky EEG headset platform, as it provided a ready-made EEG device that we could modify to our own means. We needed our device to perform similarly to the original headset as far as strength of readings were concerned, and we needed it to fit comfortably in the ear.

On the software side, we needed a platform that could cohesively stream readings from the EarEEG device and a second, unmodified headset at the same time, so that we could accurately compare readings from both EEG devices. We also needed a visualizer that could display the data coming from both headsets at the same time, so that we could visually compare the data.

To accomplish these goals, we proceeded incrementally throughout the semester by writing small portions of code and incrementally building the device. On the hardware side, we began by connecting a wire to the forehead electrode on the headset and extending the electrode. We then nested the electrode in a silicone swimmer's earplug, but this was eventually changed to a foam earplug. On the software side, we began by writing code in Python to read from the device using the NeuroPy library to interface with the device, and proceeded to add a websocket server that echoed readings to the visualizer, which was written in JavaScript.

We faced many challenges over the course of this project. On the software side, the main challenge that we faced was the question of how to compare the data received from the two headsets. We were unfamiliar with brainwaves and the format of the data that the headset used to give the results, so we were unsure what methods we could use to compare the unmodified headset with the EarEEG. Ultimately, we solved this problem by using a visualizer that shows both readings on the same graph, thus allowing the user to visually inspect the quality of the recorded data.

On the hardware side, the headset was prone to randomly losing the connection between the extended electrode and the headset, so the readings were not consistent across trials. Ultimately, this ended up being a problem with interference on the wire, and this problem was solved by making the wire much shorter, with the original headset being worn around the neck instead of held in the hand.

Ultimately, we were able to overcome these challenges and build a working prototype of the EarEEG, with the readings being similar enough to the original that we classify it as a success.

Introduction

This report has been prepared by our team to present the results of our in-ear EEG project. With the help of Dr. Hairong Qi and Dr. Jindong Tan our team was able to successfully develop a functional proof of concept in-ear EEG device. Our team consisted of two Computer Scientists, two Electrical Engineers, and a Computer Engineer. In the following pages we will present the project requirements, the design process, and the lessons our team learned during this past semester. The idea of the creation of an in-ear EEG was proposed by Dr. Qi and Dr. Tan, with the idea and technique behind it being a recently developed idea with little research to back the concept. Since conventional EEG devices require placing electrodes on the scalp, the use of an EEG to monitor brain activity has, for the most part, been relegated to laboratories and more recently home use with more compact consumer headsets being brought to market. The in-ear EEG is an idea to take that idea one step further, allowing a less accurate general purpose EEG be worn for long periods of time to collect information about large scale brain activity. This is useful for not only curious individuals, but also for medical uses where long term EEG monitoring can help to characterize disruptions in the normal brain activity of epileptics as well as helping to more comfortably allow sleep analysis to be done.

Requirements

The following list of requirements specifies the design parameters for the product and were approved by the customer:

1. Platform
    - 1.1. The application must be built using the MindWave Mobile EEG Headset
        - 1.1.1. The application must connect to a host computer using bluetooth
    - 1.2. The headset must be able to connect to a computer running Linux
    - 1.3. The application must gather data based on the readings given by the MindWave Mobile EEG Headset
2. Functionality
    - 2.1. The device must be designed to wear in ear
        - 2.1.1. The electrodes must all be located in the ear
    - 2.2. The device must be able to be used for long periods of time
        - 2.2.1. The device must be battery powered
            - 2.2.1.1. The battery must be no larger than two AAA batteries
            - 2.2.1.2. The battery must allow the headset to log data for 24 hours
        - 2.2.2. The headset must be comfortable enough to wear for extended periods
            - 2.2.2.1. The device must fit unassisted in the ear
            - 2.2.2.2. The device must feel no less comfortable than a standard ear plug
    - 2.3. The data logged must be easily parseable

2.3.1. The data must be consistent between the modified device and unmodified device.
2.3.2. The data must be in the format as specified on the NeuroSky Developer's website

3. Views
   3.1. The data will be visualized in a web application
      3.1.1. The frontend of the web application must be built using HTML, CSS, and Javascript.
      3.1.2. The backend of the web application must be built using Python
         3.1.2.1. The server must be written using the Django web framework
         3.1.2.2. The data must be collected and parsed using Python
      3.1.3. The visualization must plot the real time data obtained from the device
         3.1.3.1. The application must support the visualization and comparison of up to two devices simultaneously.

These requirements were used in the initial construction of the device, but quite a few requirements were changed as we developed the project.

Changelog

[March 15th]	Switched from multiple electrodes to single electrode design
[April 5th]	Switched from silicone putty ear pieces to foam
[April 11th]	Switched from a single cable to a twisted pair to reduce noise
[April 17th]	Switched from Django web framework to Bottle web framework

Design Process

Designing the EarEEG was no simple process. We began by performing extensive research and investigation into the problem set before us, which led us to choose a reasonable solution for both hardware and software. We then decomposed the project into manageable pieces and built it, evaluating our progress along the way. A more detailed look at those steps is as follows:

Research and Investigation

While looking for solutions to the problem presented before us, we did extensive research on EEG machines and an in-ear EEG solution. We examined several EEG headsets that are currently on the market, including various solutions from NeuroSky and EMOTIV. Many of these headsets are extremely high quality, with some even reaching research-grade standards. These headsets typically have anywhere from five to fourteen nodes and can vary in price from $200 to $800. However, we decided that a more cost-effective solution would be better for our project,

which led us to investigate the NeuroSky MindWave Mobile. This headset was significantly lower in cost which allowed us to purchase one as a control and as well as one to use for the EarEEG setup. Ultimately, we also decided that we did not need the extra electrodes that the lab quality headsets provided.

Once we purchased the headsets, we decided to research the capabilities of the NeuroSky MindWave Mobile. The MindWave Mobile headset can monitor delta, theta, alpha, beta, and gamma brain waves with the alpha, beta, and gamma waves split up into smaller subunits of their full ranges. In addition, the headset also processes the brainwaves in order to give generalized attention and meditation levels. These metrics are reported approximately once a second by the device through a bluetooth serial interface which allows the data to be read in and parsed easily. Alongside all of these processed readings, the device also sends along the raw wave data points.

When we were familiar with what readings the headset would deliver, we researched how exactly we would acquire this data. We had several options in this regard, as NeuroSky has an extensive SDK (software development kit) that would have allowed us to easily parse the data coming out of the headset. However, these libraries placed many restrictions on our development, as they were limited to MacOS, iOS, Android, or Windows development, and none of these platforms appealed to the kind of research that we wanted to perform. Since many people in our group were familiar with the Python language, and it is extremely easy to process data in Python, we wanted our solution to be implemented in Python. This led us on a search for an external library that would connect to and read data from the MindWave Mobile. Eventually, we were able to find the NeuroPy library which did most of the parsing for us, though we had to modify the library for it to fully work. Once we had this connection, we were able to collect data from a single person over a period of time.

After we solved the problem of collecting data from the device, we decided to investigate two possible solutions to the problem of actually constructing our headset. The first solution involved disassembling the MindWave Mobile headset to create our in-ear version. Since the headset already had electrodes that connected to the bluetooth device and was able to transmit data, we wanted to use the system already in place to accomplish our goals. We would create an in-ear EEG by using the electrode from the headset and simply placing it on an earplug. This was the initial solution we came up with, but it was complicated due to the construction of the MindWave Mobile headset and would result in one of the headsets being torn apart which our team felt uncomfortable about doing when we could not find any schematics of the device.

The second solution we thought of was to create our own electrode and connect it to the MindWave's electrode, thus extending the electrode without actually disassembling the headset. This would allow us to embed the electrode in an ear plug without losing the data transferring capabilities of the MindWave. Since we would not be using the large electrode connected to the MindWave headset, it would also be easier to fit this electrode in an ear. Ultimately, this was the solution we decided to pursue. We had issues beginning this process with electromagnetic

interference being a bigger issue than we had expected, but some research on communication transmission techniques showed that the best solution would be to twist the wire to cancel out noise. This technique was first used by Alexander Graham Bell, and the technique works due to the nature of the noise all being in the same direction along the wire. The twisted pair causes the induced fields to negate each other, resulting in a much stronger and more reliable signal, as well as a physically more robust connection between the earpiece and the data module.

Decomposition

In order to get a handle on what our objective for this project would be as well getting a better understanding of the objectives specified by our customer, we first needed to undertake the task of breaking down the project into manageable components. We decided to simplify the design into software and hardware aspects. For the software aspects of the design, we decided to have a two part interface, with one part serving as the backend that connects to the headset and serially sends and receives data packets in order to capture the brainwave data read by the headset. The second part of the software was the frontend interface that would take data from the aforementioned backend and display real time graphs of the measurements being taken by our headsets. For the hardware side of the design we decided to break it down into electrode design and the task of connecting said electrode to the device. After receiving the devices, we first discussed how we would connect to the headset and decided, for a variety of reasons, to simply attach the in-ear electrode to the electrode that was on the headset. With the electrode design, we first assumed the use of silicone earplugs would work well, however after a number of tests, they proved to be much too malleable and would not retain their shape well in the ear canal. On top of that, the pressure between the electrode and the ear was not great enough for a strong signal. Switching to a foam based ear plug gave us more desirable results and cleared up both of those problems because the foam would actually expand to fit your ear, thus providing greater pressure and in addition better surface contact.

Selected Solution

The final solution that we decided on is as follows:

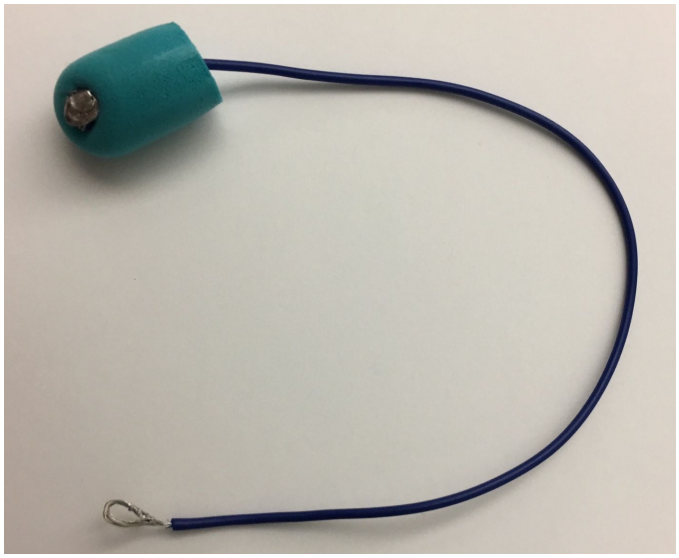

Figure 1: Earplug

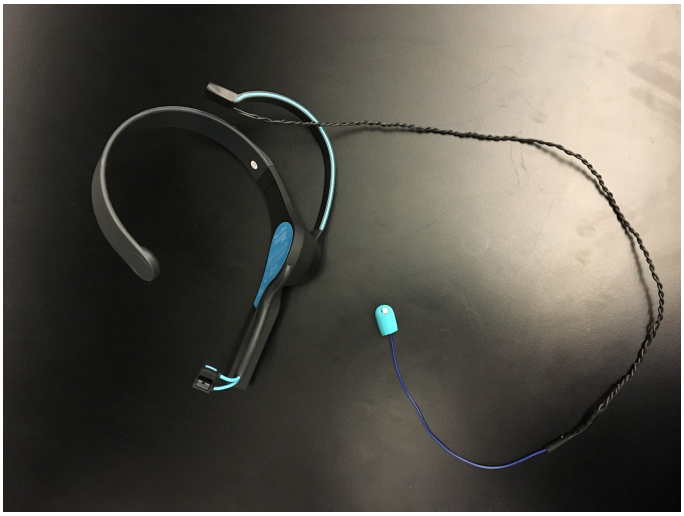

Figure 2: Entire Headset

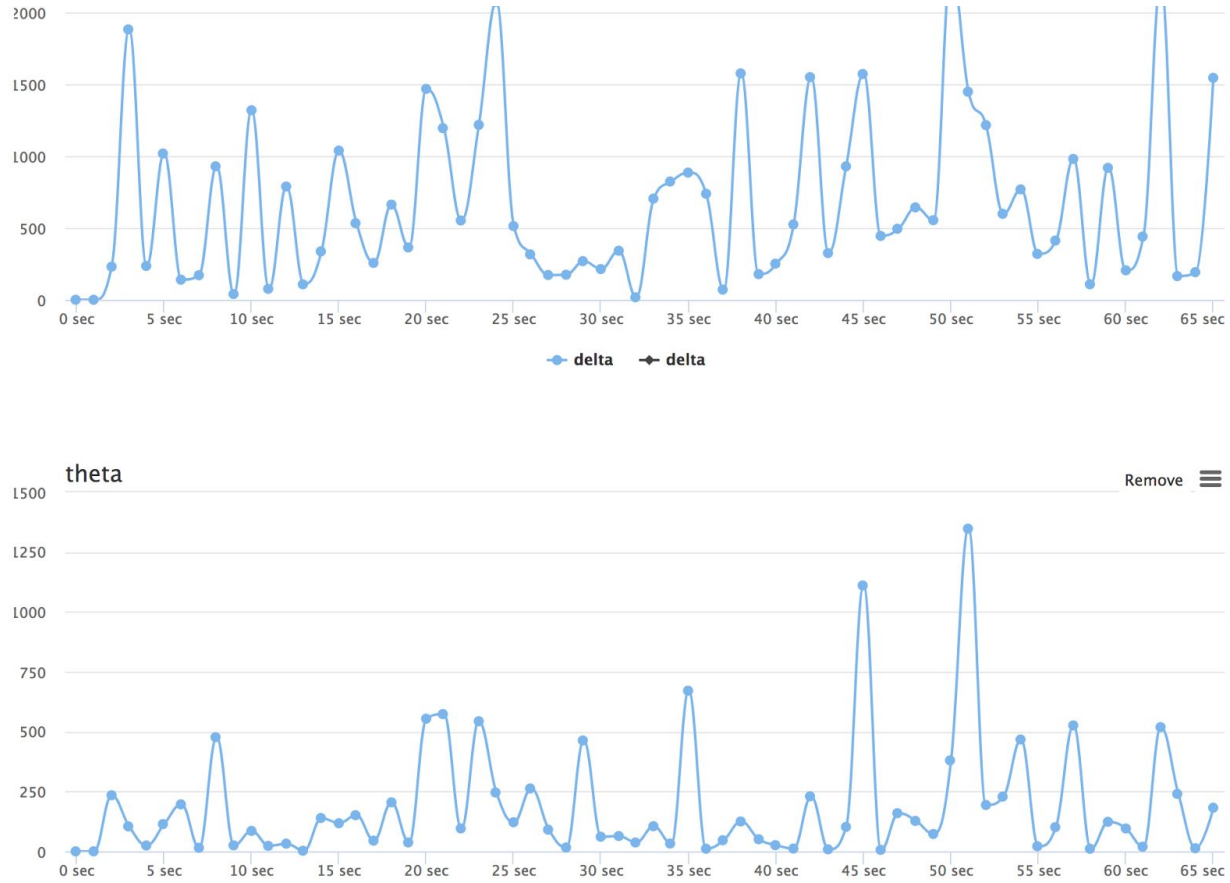

Figure 3: Web Visualizer

Hardware

In the end, for the hardware we decided that the design and implementation of the electrode was the most crucial aspect that we wanted to focus on for our project. Rather than using a large amount of time trying to design a consumer product we wanted to create an easy way that an in-ear EEG could be implemented without heavily modifying or destroying the mindwave mobile headset. This was decided since our work was more about creating a proof of concept and testing the idea to make sure that it would be feasible. In addition, making it so that only simple modifications were necessary would allow other people and institutions to create a similar implementation to quickly test the results that an in-ear eeg would show before deciding whether undertaking a more in depth and larger project would be worthwhile for the application that they are considering. For the electrode material we originally thought that the use of moldable silicone earplugs would be the best option as it would resemble the use of a custom fabricated earplug molded to a person's ear. This idea, however, did not work out as the silicone did not provide enough pressure to hold the electrode in place for proper readings and when enough pressure was applied, the electrode irritated the ear canal and caused some discomfort for the user. After this idea was scrapped, we decided to go with using a foam earplug as it

would fit well enough to the shape of the ear canal while providing adequate pressure for the electrode to get EEG readings.

Software

We decided to break the software solution into three parts: the headset client, the websocket server, and the visualizer. The headset client used the NeuroPy library, written in Python, to read data directly from the bluetooth serial port. The library then parsed the data into a usable format and returned it to the user. The library also allowed for callback functions, which is what we used to keep track of the data that we read in. Whenever the library would update a parameter, it would call a function that we specified which either sent the data along a websocket, wrote it to a file, or both. This client was written in Python and has multiple useful features for the user, including allowing the user to specify a length of time to record data and allowing the user to specify filenames and websocket servers.

As for the websocket server, we used the Bottle Python framework to accept data from the client and echo it to the frontend. This allows multiple clients to connect to the same websocket server, which in turn allows the frontend to visualize multiple headsets at the same time. The data includes a header specifying which device it came from, the value of the data, and the timestamp when it was created.

The frontend of the application consisted of a single page web application that visualized the data from the devices. The page was built using minimal HTML/CSS, with most of the page content dynamically added using JavaScript. The data was plotted on time-series charts using the JavaScript library HighCharts. The HighCharts library allows the user to dynamically add data to an existing plot, enabling real time plotting of the data. The web application received data from the backend over a WebSocket connection.

Prototyping of Solution

To see if the solution we chose to pursue was viable, we first tested it with something similar to the solution we had in mind. In order to do this, we touched the male end of a pair of earbuds to the forehead EEG electrode on the headset which created something similar to the extension that we wanted. Once this quick test was found to give us readings from the headset, we decided to stick with a similar solution for the hardware design of the headset. Once our hypothesis was confirmed we considered some possible materials to work with for our in-ear electrode. The initial material for the body of the electrode was initially chosen to be silicone since this would allow us to mold the earplug to the users ear, making for conditions similar to a custom molded earplug that would most likely be used in the scenarios that were considered for this concept. In addition, silver was chosen as an electrode material due to its high level of conductivity as well as the ability to find it at an affordable price.

Once our materials were received, we set to work, creating and testing silicone electrodes to see if they would fit our use case. Unfortunately, the silicone earplug base turned out to be a much poorer choice than anticipated, with the designed and tested earplug not providing nearly enough pressure on the electrode to get consistent readings. The only way that good readings could be received was through putting uncomfortable amounts of pressure on the electrode, making it useless in the scenario for which it was chosen, long term constant usage. After this realization, we considered some other in-ear electrode design properties, settling on the use of foam earplugs for the electrode base. This material was a good choice as it provided enough pressure on the electrode to get consistent readings while remaining comfortable and having some of the properties of a custom molded earpiece.

When the user was wearing the headphones we received readings to the device, so this has made us hopeful that our proposed solution will work. We also connected our control headset which was unaltered from the start, and this headset gave us readings that were somewhat similar to the headset readings when compared by a visualizer. However, more tests are necessary before we can confirm the data from the standard headset is similar to the one captured by the headphones, and we would also like to test with our actual device. While we worked on constructing the EarEEG, however, we wanted to get ahead by processing the data from the unaltered headset. So, we focused on building the software that would interact with the headset.

In order to make sense of the measurements taken by the EEG headset, we needed to be able to read and collect the data sent out by the EEG headset. This was a more complicated process than it seemed, as the libraries developed by NeuroSky were not geared towards the collection of data. These libraries were geared towards fun applications such as apps for children or apps for fun visualizations, not research. Also, these libraries were written for platform-specific languages, whereas we wanted to make our application cross platform. This led our team to look into ways to gather the data with Python. Initially, we looked into a few different libraries that were written in Python, but none of these libraries worked with the Bluetooth drivers on our Mac laptops. We briefly considered writing our own library, as the headset connected through a serial port that has a set specification. However, we decided to keep looking for libraries and eventually found NeuroPy. NeuroPy took care of the heavy lifting in parsing the device's reading, which allowed us to focus on developing data analytics rather than parsing data.

Once we had a method of getting data, we were able to make progress on analyzing the data that we had. In the NeuroPy library, whenever a variable is updated a callback function is called, so we used that callback function to note the logged data in our own global variable. We were able to use this global variable to log our data into a text file, which will allow us to examine the collected data when we have methods of analyzing it. In the future, we will be able to use these callback functions to perform real-time transformations on the data and use the data class to send the collected data to the frontend.

In order to visualize the data that was read from the devices we created a web-based visualization tool. The purpose of the visualization was to compare the readings from our modified, in-ea EEG device with the unmodified EEG headset. Our first solution and the one we ended up choosing was to plot the brain waves in time-series plots. At first the application only supported plotting recorded data but due to the modular design of the application functionality was added to support visualizing live readings. The visualizer supported plotting data from both devices on the same plot as well as on separate plots. Once the brain waves were being visualized correctly we started working on adding more functionality to the application. This included adding the ability to dynamically add and remove brainwave plots as well as future support for initiating a recording from the application. This was accomplished using HTML/CSS, JavaScript, and the WebSocket protocol. When the application is opened, a WebSocket connection is created using the JavaScript WebSocket API. This WebSocket connects to a server and sends a message containing metadata about what data the user will request. This flexibility allows the application to visualize both recorded and live data. The WebSocket server then pushes data as it becomes available to the browser over the connection. The web application uses the HighCharts JavaScript library to synchronously plot multiple brain waves on time-series plots.

Results

Overall, our project was a success. Here are the details of what we accomplished with our project:

Hardware

While we may not have been able to flesh out a commercial grade product that is completely worn around the ear, we did accomplish the goals that were in the scope of our project; we proved that it is possible to take accurate EEG measurements from within the ear. Our electrode design ended up being fairly robust for what it was, and extremely cost efficient. The individual electrode retrofit itself can be achieved for less than $5.00 in parts and 5 minutes in time, provided you have the base EEG device to work with.

Software

Over the course of the semester, we were able to write software that fulfilled the original requirements given to us by the customer. Our websocket-based setup as described in the “Selected Solution” section meets all the requirements we set out for both the platform and the visualizer, as we are able to compare the data feeding in from both headsets in both a live and recorded manner. The visualizer displayed the data in an intuitive and easy-to-view fashion, giving us a clear indication of how the two headsets compared to each other. This fulfilled the requirements of the project and will be a good foundation for future projects in this area.

Lessons Learned

While most of our project was successful, we had our fair share of learning experiences. Many problems arose over the semester, most of which none of us had experience with. These problems typically arose from us being inexperienced with the technology being used, though we quickly became familiar with them which reduced the amounts of mishaps later on. These problems were on both the software and hardware side, and they gave us great experience with adapting to problems and working around them.

Hardware

For hardware, we had a terrible problem with noise for most of the duration of our project. Initially we believed that it was a problem with the device itself not being strong enough to measure the data properly, followed by the belief that our construction methods were faulty, and eventually narrowed down to electromagnetic interference. The length of the wire was longer than the length specified in the data sheet, and since we couldn't shorten the wire to meet our needs we found an alternate approach of twisting the cables.

Software

On the software side, the biggest problem that we faced involved comparing the data received from each headset. Ultimately, the goal of this project was to create a proof-of-concept for an in-ear EEG device, so the device had to perform as well as a forehead EEG. This meant that we somehow had to compare the readings from both headsets to show that they were similar within some threshold of error, which posed many problems.

The first problem was that our team was inexperienced in the area of neuroscience, as none of us had any basis in that field. That meant that we had only a rudimentary idea of what the brain waves were typically supposed to look like, and thus had no clue how to compare them to, say, a control source.

Secondly, once we finally understood more about brain waves from our research and speaking with Dr. Qi., we realized that even from second to second brain waves are changing extremely rapidly and vary widely, and the patterns depend entirely on the exact state and situation that the person being observed is in. This makes it difficult outside of a lab environment to take sample runs and then compare other runs to those sample runs.

Finally, the headsets that we bought were not high quality, and therefore we could not expect high amounts of accuracy or even consistency from the different unmodified headsets themselves, let alone with our modified headset.

Ultimately, we overcame these problems by collecting data from both the modified and unmodified headset at the same time from the same person, such that the person being monitored was wearing both headsets. We then used a visualizer to visually inspect correlations between the data. This solution ended up working well and showed a similarity between the two devices.

Future Work

We believe that this project has the potential to grow into an extremely interesting project. First, we would like to see more accurate headsets used, as the quality of the NeuroSky headset impaired the tests were attempting to perform. Eventually, we would like to see the transmission chip embedded in the earpiece, so that the user would not have to wear a necklace with the embedded chip and power supply. We would also like to see more tests performed to get a more accurate picture of the similarity between the unmodified headset and the modified one. Finally, we would like to see the earpiece 3D printed with a custom mold for each ear, as this would allow for maximum comfort and contact between the electrode and the skin. Ultimately, we see this being an excellent candidate for an interdisciplinary project, and we would like to see it taken in that direction.

Relevant Coursework

CS340 - Software Engineering
ECE435 - Microelectronics
ECE342 - Communications
ECE315 - Signals and Systems

Team Contributions

Tyler Stuessi performed the role of Team Lead and was also involved in writing the backend software. On the managerial side, he organized meetings and managed the workload assigned for the team. This involved assigning tasks to each member of the team, finding times where every member could be present at the meeting, and keeping up with the schedule for the assignments for the class. On the software side, he fixed the NeuroPy library and integrated it into the software that connected to the headsets. He also wrote the software that sends headset messages to the websocket server. He was also involved in designing the architecture of the software side of the project, and he participated in creating the general design for the hardware side of the project as well.

Jeremy Herwig performed the role of Time Librarian and was involved in writing the frontend software and architecting the full stack software application. On the software side, he prototyped

and built the web application to visualize the brain waves and contributed to the development of the backend data server. In collaboration with Tyler Stuessi, he architectured the software side of the project.

Dillon Hunneke performed the role of Solutions Architect and was involved in designing and building the hardware for the project. For the electrode, he researched the materials needed and worked through several prototypes before selecting a final earpiece design. This involved lots of testing on various ear types and with various materials. He also tackled the problem of the noise issue, using knowledge obtained from a communications textbook and the datasheet for the NeuroPy. He was also involved with overseeing the assembly line of earpiece production.

Evan Goble performed the role of Lead Presenter, and assisted in the development of the electrode for the headset. He made sure that the team was informed on their duties and prepared for all of the presentations that our team was required to give. He helped develop earbuds for the demo for Dr. Qi and Dr. Tan. Evan also took part in researching ways to allow the earbuds to be modular and interchangeable for our poster session.

Arthur Vidineyev performed the role of Lead Researcher and helped to communicate between the hardware and software side of the project, bridging the gap in the understanding that the computer scientists and electrical engineers had with regards to the other's areas of academic pursuit. He helped to research the avenues of data collection and connection to the headset; in addition he assisted with assembly of the electrodes as well as the practical considerations of the headset's performance.

Signatures

Dr. Qi: _____ Date: 4/25/17

Tyler Stuessi: _____ Date: 4/25/17

Evan Goble: _____ Date: 4/25/17

Dillon Hunneke: _____ Date: 4/25/17

Jeremy Herwig: _____ Date: 4/25/17

Arthur Vidineyev: _____ Date: 4/25/17