Group 3

Frank Alexin

Nicholas Dijkhoffz

Adam Hollifield

Mark Le

August 7th, 2015

Sponsored through ECE by Boeing, Inc.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1 EXECUTIVE SUMMARY

This project is designed with the goal to make the game of football, and other contact sports, safer for the participating athletes. The motivation of this project primarily stems from the NFL Players Association (NFLPA) suing the National Football League over players who have suffered concussions and thus debilitation mental issues later in life. The BIO-Helmet is designed to track and accurately measure the amount of the g-force that is received to a player's helmet during a typical football tackle or hit. An accelerometer sensor placed within a regulation size NFL helmet is used to track the amount of g-force the player experiences as well as the direction of the applied force. The BIO-Helmet is also designed to use an EEG sensor array to measure the brain waves of the player. Both accelerometer and EEG sensor data is processed by a TI Tiva C microcontroller. This microcontroller poles the sensors for data and communicates this data over Wi-Fi to a local server. The server then applies additional processing the incoming data and enters it into a database. A medical professional can view the sensor data in a customized MATLAB environment for analysis. This data can then be used by a neurologist, either during the course of a football game or for additional diagnosis after a game, to determine if the player is showing any biological signs of a concussion. The availability of both the accelerometer and neurological data allows the heavy g-force hit to be related to the brain data and a more rapid diagnosis and treatment recommendation be provided to the player.

In order to make this project a reality we needed massive amounts of research into not only the realistic applications of this project but also the hardware possibilities. There are many parts that we may have chosen from and there are pros and cons for each and every single one. There are also design aspects that we chose over others in order to maximize reliability and safety for the players.

Research from many other papers and similar studies is incorporated into this project. Most notably, a similar study performed by the National Football League in 2012 is used as a basis for the project inspiration and implementation goals. This study focused on the accelerometer data alone and without real time tracking or an accurate historical approach. This project seeks to expand this study by also measuring brain waves and relating these waves to the accelerometer data. Various other research into the technical details and implementation of this project was also conducted. Several impacting standards, from multiple bodies, were researched and adhered to, or used components which include adherence to relevant standards, as part of this project. Various IEEE and ISO standards are referenced in the context of this project both for the electrical and power system designs as well as the communications aspect of this project.

Health and safety constraints truly pose the greatest impact on this project's design. This applies specifically to the testing phases of this project. It is obviously not in the interest of safety or ethics to attempt to give a test subject a concussion, or near concussion symptoms, to then have brain wave data to analyze. This project will therefore exist as a proof of concept module for relating these two data sets that can then be implemented or tested in a future environment. All testing was performed in an ethical manner with g-force specific design and testing conducted independently from the brain wave testing. Other constraints that have a significant impact the design of the BIO-Helmet are economic constraints. This project was sponsored through ECE by Boeing, Inc. One design constraint of the implementation of this project is for the project budget to remain within (or reasonably above) the provided amount.

2 PROJECT DESCRIPTION

2.1 PROJECT MOTIVATION AND GOALS

This project is motivated primarily from a 2013 lawsuit in which the NFL Players Association (NFLPA) sued the National Football League (NFL) over mental injures which players sustained throughout their careers as NFL players. This lawsuit ended with the NFL paying out $756 million to its retired NFL players. Additional funds were allocated for players' medical expenses as well as for additional research to improve player safety. Many of these players suffered concussions from high impact hits to the helmet, during their careers as NFL players, which then led to several mental issues later in life. The motivation for this project is to both protect football players as well as remove liability from the sanctioning league. The BIO-Helmet contains two sensor arrays that monitor both the force load of any impact to a player's helmet as well as the player's neurological response to this impact. Other project motivation stems from providing the medical world with new correlations, and history accuracy, between the force impact and neurological data sets.

A reliable device was developed that can detect concussions at time zero so that the athletes can be diagnosed and receive medical care at the earliest possible time, thus reducing the chances of permanent injury astronomically. To our knowledge, the correlation between the detection of an actual concussion with brainwaves does not yet exist in the medical field, however we can detect the signs of a concussion and that was the group's approach to this device.

The primary goal of this project is to develop a prototype hardware and software model which can protect professional sport athletes from the damaging effects of a concussion. This design can be referenced for future mass production and expansion into professional football. The technology could also be incorporated into collegiate or high school football helmets as well as other contact sports such as soccer or rugby. Another goal of this project was to improve the speed at which a concussion diagnosis is given. The current system for detecting a concussion includes asking the patient a series of questions such as "What is your name?" and "Who is the current president of the United States?". The patient's answers to these questions determine whether or not they possibly have a concussion. A goal of the BIO-Helmet was to quickly determine whether or not a player is at risk of a concussion, by providing force impact and brain wave data to a neurologist. Coupled with the traditional question based diagnosis, the BIO-Helmet can provide a medical professional with additional physical and neurological cues to be able to deliver a quicker, and more accurate diagnosis. One final goal for the BIO-Helmet was provide advancements in the medial world by allowing neurologists to study any possible correlations between force impact, brain waves, and concussions.

2.2 OBJECTIVES

The primary objective of this project was to create a complete hardware and software prototype that will measure the amount of the g-force applied to an athlete's skull and the brain wave data from the contact sport athlete in an effort to speed up concussion diagnosis. This project was completed at the end of EEL 4915 Senior Design II in August 2015. Another primary objective of the BIO-Helmet was to display the accelerometer data along with the brain waves in MATLAB for a neurologist to diagnose the athlete for a possible concussion or other mental injury. Modularity was another design objective of this project. This project was divided into a number of subsystems that are each managed by their own software and hardware design. Another objective was to keep this modular design as much as possible throughout all stages of the project to reach a final device which was easily modifiable and replaceable in the future. A final objective was to remain within the budgetary constraints of this project. Sponsored funding was received for this project and BIO-Helmet was designed and implemented in such a way that maximizes the provided funds without heavily exceeding this amount.

2.3 REQUIREMENTS SPECIFICATIONS

The following items represent both hardware and software requirements specifications for the BIO-Helmet project. Each of these items are researched, expanded, and designs implemented throughout this design document.

- The helmet must meet safety protocols prior to alterations for the project.
- The microprocessor must be able to output accelerometer and EEG data at a maximum of 20 Hz.
- The helmet must have a battery which lasts at least 2 hours.
- The accelerometers must be able to detect not only g-force but also the angle of impact.
- The EEG sensors must be able to provide valid EEG data from the surface of head without any invasive impacts to the user.
- Measure 100g impacts without losing reliability in data output.
- The helmet must have a battery led status bar to avoid any possibility of sending a player onto the field with a low charge.
- The battery must be rechargeable as to reduce waste.
- The accelerometers and EEG sensors must be able to work under the constraints of a single microprocessor.
- The charger must be able to fully charge the battery via USB or charging port without overheating or damaging the battery.

- All electronic devices must be able to withstand the impacts without losing reliability in data output.
- All electronic devices must be able to operate under the padding and exterior of the helmet with little or no air ports.
- The BIO-Helmet must communicate wirelessly over 2.4 GHz Wi-Fi frequency with a local server.
- The local server must collect and process all sensor data (accelerometer and brain wave) received from the BIO-Helmet at a rate of twenty times per second.
- The local server must store all sensor data in a historical database for historical view and retrieval.
- Reporting software must be implemented which allows a user to view all sensor data in an easy to read graphical and/or tabular format; viewed as a single dataset obtained from the database.
- The reporting software must alert a user on the side line, with a popup message and a historical table that a hard impact has occurred.

2.4 Related Standards

The following are related standards which had a design and/or implementation impact to this project. Each section includes a brief overview of the standard and how the standard specifically impacts this project.

2.4.1 BSR/IEEE 802.11ac-201x

IEEE 802.11 is the wireless communication standard for WLAN. This specific revision of the standard improves greatly on both the range and throughput of a wireless local area network. The IEEE 802.11 family of standards allows WLAN communication devices from multiple manufactures to operate together. This standard impacted this project in that embedded parts that comply with the IEEE 802.11 must be selected. This will allow for any local server hardware (which includes a Wi-Fi chip) to be used to connect and read data from the BIO-Helmet. IEEE 802.11 ensures the project design goals of both cross platform and cross hardware wireless communication ability.

2.4.2 ISO/IEC 14766:1997

ISO/IEC 14766:1997 focuses on the set of standards behind the Transmission Control Protocol. TCP is used throughout network communication to provide reliable data transfer at the session layer of the TCP/IP stack. This standard allows different network communication devices to present and communicate data at a high level independent of the physical medium. This standard was applicable to this project because the transmission control protocol standards suite will allow for communication from the wireless part in the BIO-Helmet to the local server over TCP sockets. These standards ensure absolute data integrity between communicating devices and will ensure that all sensor data sent from the BIO-Helmet is accurately received by the connected server.

2.4.3 RS232

The RS232 family of standards provides a standard for serial communication between devices. The RS232 standards contain voltage, pinout, and cabling standards for this serial communication. The Tiva C microcontroller selected for this project contains a UART (universal asynchronous receiver/transmitter) which uses RS232 standards to perform serial communication over a USB port. Most Wi-Fi boards on the market also make use of RS232 communication standards to send and receive information from the microcontroller processor and memory subsystem.

2.4.4 PEP 8

The PEP 8 contains the style guidelines and standards for writing Python code. This standard covers code structure and comment style, use of white space, and function and variable naming standards. This standard was applicable to the BIO-Helmet project because a majority of the server code was written in Python. Following the PEP 8 standard for all Python code allows for trivial future expansion of server code as well as readability for future developers.

2.4.5 PEP 249

The PEP 249 standard contains style guidelines and standard for writing Python code that interacts with SQL based databases. This standard also includes many security related coding styles to avoid a Python application being vulnerable to an SQL injection attack. These security related standards are essential for the operation of all Python database connected programs to not cause corruption or interruptions in the database service. This standard applied to the scope of this project because the integrity and security of the database subsystem must be a top priority for any production database connection application. This standard also offers style guidelines for all of the Python related code for this project which connects to the BIO-Helmet database.

2.4.6 MATLAB Style Guidelines 2.0

This set of MATLAB coding style standards covers standard coding practices with a goal of producing MATLAB code that is easy to read and has a higher likely hood of correctness. This style guideline was relevant to this project because MATLAB will be used for the reporting software to display both sets of sensor data. Following this standards guideline allowed for the BIO-Helmet reporting code to be easy to read and easily modifiable in the future. These guidelines also encourage the use of coding style that is easy to debug. This ensured that development of the BIO-Helmet reporting software progressed smoothly with easy to locate and fix bugs.

2.4.7 IEEE 1625-2008

The IEEE Standard for Rechargeable Batteries for Multi-Cell Mobile Computing Devices standard is pertinent within the project because it has rechargeable Li-ion batteries within the BIO-Helmet to reduce waste and ensure that the batteries can be at a full charge before every game. This standard affected this project because it establishes the quality and reliability of the batteries within the design specifications. The standard also includes vital information on the battery's electrical and mechanical construction, cell level charge, discharge controls, and battery status communications. The battery status communication was not an initial part of the original design but to abide by the IEEE standards it was added to include any and all considerations for end-user notifications as stated within this standard.

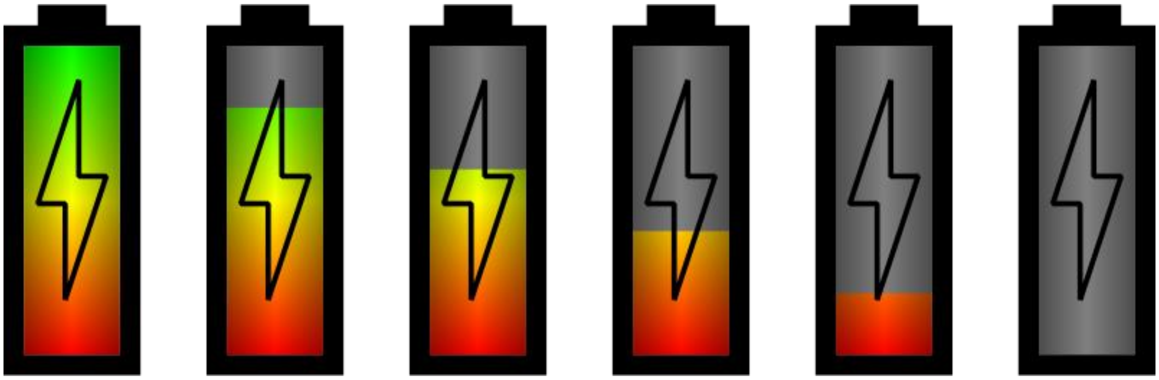


Figure 2-1: Battery Status Indication; An image showcasing potential battery status indicators for use in the reporting software; reprinted with permission from Openclipart

2.4.8 IEEE 1680.1-2009

The IEEE Standard for Environmental Assessment of Personal Computer Products, Including Notebook Personal Computers, Desktop Personal Computers, and Personal Computer Displays standard was pertinent to this project because every helmet will essentially include a personal computer due to the sensitive materials it will be sending through to either the researcher or the medical personnel. Even though this standard is meant for personal computers, mainly aimed towards desktop and notebook computers, it was believed that even though there is not a user interface on the BIO-Helmet itself, the materials used to construct it should be environmentally friendly and should be designed for energy conservation as stated within this standard. Overall, this standard pertained to this project in that the criteria to reduce or eliminate the use of environmentally sensitive materials is a design goal of any engineering project.

2.4.9 IEEE 2010-2012

The IEEE Recommended Practice for Neurofeedback Systems standard covers electroencephalography biofeedback which was a core standard to this project. It gives recommendations on the system and software to enhance the reporting quality and make information more available for device users. Even though the use of known software (MATLAB) was used, and controlled devices, this standard was used as a reference to abide by to create the BIO-Helmet with the highest possible quality.

2.4.10 IEEE 1686-2013

The IEEE Standard for Intelligent Electronic Devices Cyber Security Capabilities covers various standard for data transmission security. Security was a very important goal for this project as it is transmitting sensitive materials in real time. Outside personnel must not be able to easily crack the BIO-Helmet software and see or manipulate the readings for their advantages. For example, selling access to player brainwaves or pulling a player out of the game by sending false information. The design team believed that brainwaves are very personal information and should only be used to enhance a player's health and well-being; not to be abused in ways that will damage the player or manipulate the game.

2.4.11 Health Protection Standards

The standards available today to protect athletes have proven to be inefficient and have led to some athletes being seriously injured with lasting, permanent damage. With this project, a new standard was set that might revolutionize the health protection standards for athletes. Today's helmets were brought to a new age by making the BIO-Helmet a full EEG and accelerometer sensor to show brainwave activity before and after impacts. This data can then be used to correlate concussions and other head injuries so that the medics can better diagnose the athletes. It is believed that due to the fact that athletes, when playing in a game, will have high adrenaline levels that play negatively towards their perceived ability to continue playing. The only way to address such an issue is to correlate the change of their brain wave activities to see if any symptoms of a concussion are detected before he or she goes back into the game and causes even more damage to themselves. The symptoms that are focused on are drowsiness, rapid eye movement, or blinking, and vertigo. Even though the athlete may not feel any different after the impact, even for periods up to several hours, this project is able to correlate heavy injuries using several biological symptoms of concussions to greatly improve the medical diagnosis of a concussion. Today's standards for testing for concussions are to ask the players questions about the year, who is the president, etc. To entrust athletes, who have high adrenaline levels and careers on the line, and ask them if they feel any pain is not scientific and reliable enough to recommend any sort of medical diagnosis. It is believed that the best way to detect a concussion on the field is by detecting the symptoms of a concussion even if the players themselves cannot feel them.

2.4.12 Helmet Standards

According to the National Operating Committee on Standards for Athletic Equipment (NOCSAE), the standards for helmets must allow impacts to be substantially below 1200 SI. This is tested via twenty-nine different impacts all varying from impacts at 12.2 mph, on six different locations, 4 impacts at high temperatures, and 7 impacts of 7.7 mph that must be under 300 SI at the lower speeds. These test on impacts at all different angles to ensure reliability and safety of the helmets before use. The tests at high temperatures ensure that the product will still be able to perform when in use in extremely hot or cold environments. Finally, the tests at low speeds are expected to yield much lower SI are done to ensure that the helmet is safe for low level impacts. This is to reduce any possibility that the player may be injured due to a low impact. To ensure that this project abides by the same standards that are listed above, safety tests were performed to ensure that the BIO-Helmet worked under the strict guidelines that are listed by the NOCSAE before the BIO-Helmet can actually be used in a real life game. However, for the purposes this product prototype, many of these extreme testing conditions were removed and only the tests mentioned within Section 6 were performed.

2.5 Realistic Design Constraints

Every project falls under some form of constraints. In this project, however, the team faced many constraints due to the available technology and what could actually be afforded. The following sections outline and describe various realistic design constraints from several constraint categories. Constraints in each category are analyzed and described how these constraints applied to the BIO-Helmet project.

2.5.1 Economic and Time Constraints

This is one of the most difficult constraints faced by any project team. In this project, economic constraints also have a large impact on the outcome of the system. The cost of many of the components are on the higher side, given that the application is medical and required more precise components, such as the precision amplifier and the electrodes. The project was funded by Boeing, Inc. and also through a component funding by Texas Instruments. There were no measures taken to directly reduce cost. There were, however, components for use in the Senior Design Laboratory from previous projects. Texas Instruments provided funding, given that the project contains a number of TI components. This reduced the cost of basic amplifiers and other simple parts. Other costs included the ordering of the PCBs, the components for the wireless hardware, and the power system. Any other costs that were encountered in the project were provided by the group members in order to fulfill the requirements.

Another important constraint is time. The frame of time is one of the biggest challenges that a project can encounter. In any project, many factors can have an impact on time. For this semester, the goal of the project is to deliver a working prototype and final version of this design document. The due date for the deliverable is August 7th, 2015. The first time constraint was the research process where the project team needed to find how to accomplish the application. The second part was to understand how to accomplish the deliverable and finally document the needs, and requirements, to deliver. The design and the testing of the concept must also be included. For the second part of Senior Design, the goal was to build the design and deliver the initial goal. The most important factor was to order all the parts in time so that there was enough time to complete the project. PCB builders took a couple of weeks so that must have been taken into consideration. During the semester break, simulations and breadboard prototyping were done to accelerate the process. Planning needed to be done for adjustments and unexpected events that could have delayed the project.

2.5.2 Environmental, Social, and Political Constraints

The vast medical advantages of having a real time portable EEG machine coupled with an accelerometer can greatly increase the probability of correctly diagnosing brain abnormalities after an impact, the repercussions that can arise from the use of such a personally evasive and financially burdening medical device must be examined. The environmental burdens of the BIO-Helmet will be very minute as it was built to be as green as possible. As stated within Section 2.4, the product was built out of materials that will take the environment into consideration which greatly reduced possible damages both in research and development as well disposal of our product once it is obsolete.

Socially, the repercussions that will be faced by this product are vast. Do players really even want to know if they actually have brain concussions or are too hurt to continue? A medical examiner will definitely say yes, however, in a financial aspect, this conclusion is much less likely. If a player is pulled from a game and is diagnosed early, to prevent any possible permanent brain damage, they will undoubtedly be hit financially because the more games that they cannot play, the less money they make. This same logic would apply to the league that they are playing for; can they have a majority of their most famous players benched because their brains are hurt even though physically they can still play? With the increased amount in the player's adrenaline levels at the time of impact, the players themselves will not even feel as though they need to sit out and seek medical attention. Knowing the fact that if they do follow the BIO-Helmet's warnings, they will be losing money so the reality of a player actually heeding and listening to the device's warning is most likely going to be either very small or nonexistent.

Politically, players will have to sacrifice some privacy for the BIO-Helmet to be able to collect brain wave data. Today, there exist many privacy related issues surrounding cellular phone or online tracking to create personalized advertisements. The BIO-Helmet is able to detect somewhat of what is going on inside of a person's brain and can be misused as a lie detector. Even though the BIO-Helmet was programmed to detect only symptoms that will lead towards brain injury or abnormalities, and not be used to create a portable real time lie detector, this is a huge political constraint that was considered for the scope of this project. To do avoid this, the monitoring users will need be registered under doctor patient confidentiality, as they are looking at very personal data. The BIO-Helmet security was top notch to mitigate any possibility of data interception to manipulate or sell the data coming from the players.

2.5.3 Ethical, Health, and Safety Constraints

According to the code of ethics provided by IEEE, which served as a guideline for this project, any action that violates this code should not be considered. This project, and its team members, must abide by the ethical standards put in place by the IEEE organization. Also, given the project goals, and the involvement in the sport of football, it was important to recognize the responsibility of providing a safe and accurate system. The device needed to be able to output reliable information and reduce the risk factor that comes with the sport. If the project increases in magnitude, then considerations must be taken when working with complete teams. Their needs must match the goals of the project and also abide by the same ethical standards followed to build the design.

Another constraint is that due to the fact that helmets may not be a sterile environment our sensors must be noninvasive to the human user. We did not risk any damage to the user who might get tackled or injured and we definitely did not want anything poking any part of their head when they can be under such conditions. Of course we do not require the user to have a shaved head to improve the sensors output of data so we had to sacrifice some reliability in this aspect as the best way our device would work if it is as close to the scalp as possible but, taking into consideration of the average user, it works without.

The next constraint would be that the system is safe and not a hazard to use. The device must undergo a review process of all of the components and sections. Conclusions were drawn that no section has any repercussions in the safety and well-being of any user. Given the amount of signal processing, sensors, and components it is important to analyze the effects of these in the user. Testing and shut down procedures were included in case of a negative outcome.

This device not only affects the professional football players but, since it was implemented correctly and made with financial burdens in mind, it might be a huge market for the youth players as well. The human brain is still growing until the age of sixteen, which means that while it is great that younger players are active, the damages they may be causing to themselves is alarming. If such harmful activities can cause grown men to have such devastating after-effects, then think of the harm that children can be exposed to everyday they the same contact sports. In future models, we may be able to implement GPS tracking and cellular internet connectivity within the devices so that parents can get real time feedback and know where their child is anywhere at any time. This device may change the way that helmet safety standards are defined, not only in adults but also in children as well.

2.5.4 Manufacturability and Sustainability constraints

This project required many components and stages that needed to be integrated together to achieve the desired outcome. Initially, the prototyping stage was not manufacture friendly. Although the initial design was hard to build, after designing the PCB that integrated all the components together, the process of manufacturing became simple and efficient. The casing was completed through CAD design and 3D printing. As mentioned before, since the project was in the prototype stage, the need for a manufacturing plan is not as important as having a functioning design. Once several design goals are completed, the team can focus on making the system more efficient in many ways. The user of this product needed it to be versatile, light, and accurate so there is room for improvement.

The packaging for the system is an important part as it is the first line of protection for the device. The best approach for this was to use CAD design and 3D print a case for the prototype size device. Many considerations made in Section 6.1 were not completely achieved in the proof of concept stage of this project. Considerations must be made, given that the main goal was to have a working design and then move on to developing solutions to other problems. During the second semester of this project design, possible solutions for the system issues were discussed in detail.

3 RESEARCH RELATED TO PROJECT DEFINITION

3.1 EXISTING SIMILAR PROJECTS AND PRODUCTS

The following sections represent projects that are related to the BIO-Helmet project definition. Since the field of brainwave analysis is fairly new, much of the projects involve different areas of specialization. In some cases they fall in direct correlation to the BIO-Helmet project definition and in other cases the information acquired just serves as a base for the development of this project.

All of these projects and products serve as a base for future development in the area of sports technology. Specifically for this project, they provided prior experience and development ideas that helped create a system that is more efficient. Given the complexity of detecting a concussion, it is important to consider that any system or project has to be able to meet certain requirements. Most of these are directed towards the simplicity to include in safety gear. At the same time, they are efficient in the process of collecting data, while providing real time decision making information. In regards to the design of this project in comparison to other products, there had to be significant improvement. It is imperative that the system be able to measure, process and collect brainwave data. Secondly, the system must be able to transmit and record the data for further processing or for analyzing. These two requirements were the innovative drive of the project as whole. Furthermore, the integration of the system as a whole including the EEG, accelerometers, and power source must be done efficiently. Packaging is an important consideration in this project. As mentioned before, athletes must be able to wear the system in the helmets and still be able to perform at the highest levels. The system should not have a negative effect in the playing performance of the athlete. This means that the size and shape of the system circuit must be comfortable enough to fit or attach to a helmet. The EEG sensors should be held to the scalp by some fabric along with the conductive gel.

3.1.1 Head Impact Telemetry System (HITS)

The head impact telemetry system is a system designed by researchers at Virginia Tech and Dartmouth in 2002. It uses six sensors mounted in the helmet to detect impact force and relays the information back to the sidelines when it detects g-forces greater than ten g's. The sensors are mounted normal to the head in a ring around the top of the skull. The helmet maker Riddell is using this system in its InSite system to track the severity of impacts and help determine the likelihood of a concussion. The HITS system has been used to measure thousands of impacts already which resulted in useful data that can be used to calibrate the Bio-Helmet's sensors properly.

The system that Riddell has developed is similar to the BIO-Helmet. Both are collecting impact data and transmitting it back to the sidelines. Riddell's system has a proprietary handheld unit that relays the information to the coaching staff on the sidelines when the player receives an impact to the helmet. Riddell also implements a scoring system into their devices which they call HITsp (Head Impact

Telemetry severity profile). It takes several factors into account during impact and wraps gives a score based on this data that indicates the probability of a concussion. In regards to the BIO-Helmet this may be something that is implemented in the future to provide the end user with an easier experience rather than just displaying raw data.

3.1.2 2012 NFL Accelerometer Study

In the professional era that represents the National Football League, there have been many events that are painfully remembered. These events represent the negative side to any sports worldwide. In some cases, they do not involve on the field issues such as domestic violence, suicides, or felonies. The National Football League has experienced over that decades an increasing awareness of traumatic brain injuries. More commonly known to the public as concussions. These common injuries were part of the game of football as many other sports and were not related to any future problems to the players. Finally in 2009, the NFL indicated that there could be a correlation between concussions and long term concussion effects. According to Foster, in his article for Popular Science, “Professional football players receive as many as 1,500 hits to the head in a single season, depending on their position” (Citation needed). This number is extremely alarming as a normal career for a professional football player begins at a very early age. Foster goes on to mention that the number does not count the number of hits a player receives before his professional career.

The National Football League responded to this problem by announcing in 2009 that concussions have long lasting effects as detailed in a New York Times article by Schwarz (Citation needed). Along with the support from the league, and many other cases (players retiring early), the research for limiting concussions in athletes took off. The biggest issue as mentioned in Foster’s article were the limitations provided by the protective helmet gear. Helmets are used across a variety of contact sports such as hockey, lacrosse, and others. The first protective resource that a player has against concussions is the use of a helmet. Studies by the CDC has shown that as much as 3.8 million sports related concussion happen every year and that was in 2009 (citation needed). Most of the time players receive impact on the side of the head, which in turn leads to one of the most important factors in concussion events: rotational acceleration. Foster then goes on to mention that most research and companies are driving hard to reduce the rotational acceleration that a player experiences upon impact (citation). Furthermore many articles support the fact that there was a need to identify when and how to measure the impact a player experienced.

The next step mentioned in Foster’s article regarding the move by the league and the industry was the introduction of accelerometer sensors into the helmets. The sensors were being used for research purposes to measure and collect data that could then be analyzed and processed. The results were to provide better solutions to the event of traumatic brain injuries. In his article, Foster mentions the improvement of Riddell helmet products by increasing the padding inside (citation). This, and other physical changes to the helmets, were to make claim that the products were safer to use in play. In another study, the addition of nine accelerometers to the helmet helped measure the force of impact of linear forces; after, the data was then processed to measure the rotational acceleration.

According to Foster's article, and much of the research that was completed during the past several years, the biggest problem is being able to detect in real time when a layer has suffered a concussion. The league has moved towards increasing research funding which has been provided with ideas and innovation but has still failed to deliver a safe and accurate way to detect concussions. What has been unknown from this research is what is driving the definition of this project. The provision of a device that can monitor and process data that can support the detection of traumatic brain injuries at any stage is the goal. The previous research has improved the gear in a physical way, but has also provided with the informational resources required to move forward with real time technology.

3.1.3 Mind Controlled Car

This project was completed for a senior design by a group of engineering students within UCF EECS. The main goal was to be able to receive input from brainwaves via a headset that would later be translated into RF signals that could be processed by the radio controlled car (citation needed). According to the documentation, the purpose was to be able to think and move the radio controlled car. The research included finding the brainwave signals that could be used to move the car, and how to convert that signal into a RF signal.

This project served as an example on how powerful brainwaves can be if they are used in a correct manner they have the power to move physical objects. Again, this proves that there is much potential in the use and reading of physical objects. This application related to the BIO-Helmet project regarding the potential of using brainwaves in different ways to perform a diagnosis. If there is a known behavior or reaction in certain situations, then results can become predictable. If players can be monitored constantly, their brain wave signals become more predictable and therefore diagnosed quicker.

3.1.4 Penn State University: The Center for Sports Concussion Research and Service

This is a center for sports concussion research at Penn State University. This center for investigation of sports concussions contain several papers that discuss EEG studies of athletes that experienced a concussion and the effects that occurred after the event. In a specific study titled "Residual alterations of brain electrically activity in clinically asymptomatic concussed individuals: An EEG study" A number of athletes were submitted to EEG studies in order to research their brain activity after a concussion event. The purpose of this research is to identify a correlation between brainwave activity and concussions. Are there immediate signs, or symptoms that appear at longer period of time? This is where real time data collection systems are necessary to create a history for athletes. This history will facilitate the follow up with patients, and allow physicians provide a diagnosis based on trends or patterns in the data.

3.1.5 Drowsy Driving Study

The biggest proponent of this project is the use of brainwaves to identify certain conditions in which a person may have a concussion. This leads to the study of brainwaves and how can these be measured to identify symptoms of a TBI (traumatic brain injury). Brainwaves will be further discussed in more detail in Section 3.2.1. The brain has different types of waves that represent different events. These waves have different frequencies, and an EEG can measure and display these waves for further analysis. In an article by Orwig from Business Insider, a teenage girl who made a science project out of drowsy driving discusses her use of EEG to prevent these events. The student, who was a Broadcom MASTER's finalist, discusses her project and how she identified the different meanings of the brainwaves and frequencies. Orwig goes on to mention how Wu used a correlation between the alpha and beta waves to detect if a driver is drowsy. The other big side to this approach is that Wu also used EEG readings to see how much the driver is blinking while drowsy.

Another note, on the non-technical side of this example, is the fact that one of the project goals is to offer a solution at a low cost. There are several ways online today in which a person can practice to self-measure their brain activity. This is being done to improve brain health but also indicates the reducing cost of this technology.

This article serves as a reference for new ways to detect physical changes in subjects when exposed to a certain event. In the case of Wu's project, the event concerned drowsy drivers. This event leads to more research on what symptoms are present when an individual is drowsy. With EEG readings, drowsiness can be measured, but blinking plays a supportive role to the initial premise of the brain activity readings. This opens the opportunity that a system that can read multiple inputs to support an initial premise that upon a certain event you have a certain outcome. More so, this article supports the initial premise that if a system can measure multiple symptoms of a TBI (g-Force, Dizziness, Light Sensitivity, etc.) then the detection of the same can be more accurate and time sensitive.

One last component from Orwig's article is that the data can be measured, collected, and analyzed. The next step would be to save the data and analysis for future reference. Players could have their own personal history regarding concussions that can later on serve as a prevention tool for long term effects of brain injuries. To be able to track the data over time can also serve as research for future opportunities.

3.1.6 Shockbox: Sports Helmet Sensors

Shockbox is a product that is sold commercially for use in helmets for contact sports. The overall product has a similar purpose; to provide information to the person on the sideline or observing the wellbeing of the players. It provides information such as the amount of hits the player has received, and whether a hit was powerful enough to cause a concussion in a player. The information is processed and the transmitted to a smart device. The most important aspects of this product are its portability and wireless transmission of information into a portable smart device. The lower cost also is beneficial to the product commercialization, which is due to how the product is manufactured.

3.1.7 BrainSentry

BrainSentry is another company that has introduced to the market another sensor that goes on the helmet of a football player and helps in the detection of a concussion. The basic functionality of the sensor is to count the amount of hits the player is receiving, the impact force, and illuminate an LED indicator if a player should come off the field. The interesting sides to this product are the simplicity and the functionality. The LED indicator is also a nice addition that could help anyone on or off the field to see if a player should stop playing.

3.1.8 Mindwave

This is a product made by Neurosky that is used for personal EEG training. It's portable, light, and contains most of the circuitry for processing EEG already. The cost is low, within $100. This served as a backup option to the project in case the device did not deliver the proper needs to fill the requirements of the project.

3.1.9 OpenBCI

This is a community created device that can sample EEG, EKG, and EMG signals. Supported by an open source community and various electronic hardware communities, OpenBCI has a vast library of medical applications. It is supported by an Arduino boot loader which does not allow for use in this project. Nonetheless, the main idea of this project is to have a final design very similar to this one, where all of the necessary elements for processing of physiological signals can be processed in a small and portable device. The cost, however composes more than half of the budget for this project, which would make one of the desired requirements not possible; low cost development. Figure 3-1 contains an image of the OpenBCI board.

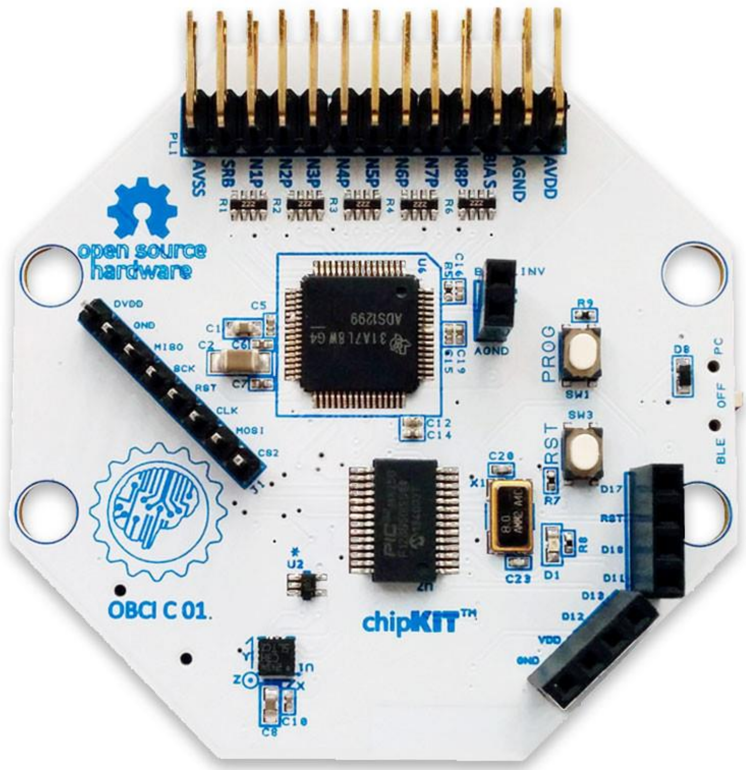


Figure 3-1: OpenBCI Board; An image of the OpenBCI board; reprinted with permission from OpenBCI

3.2 EEG Sensor Array

The following subsections discuss research related to brain waves and the implementation of an EEG sensor array for the BIO-Helmet.

3.2.1 Brain Waves

One of the innovative aspects of this project is the use of brain waves; or the electrical activity of the brain to see if a player (upon impact) is in condition to return to gameplay. Also, this collection of data is unique to each athlete. Each athlete can then have their own history of brainwave activity that can be referenced historically to observe any changes. As in many cases, multiple hits can have long lasting effects that cannot be observed immediately upon injury. Understanding the electrical activity of the brain is an important step in the process of this project. The following represents a brief description of brain waves and how they intend to be used in this project.

According to Brainworks, a leading neuro-feedback company based in the UK, brainwaves represent “the root of all of our thoughts, emotions and behavior” and “the communication of neurons within our brains” (citation needed). In other words brain waves are electrical impulses that represent the way neurons communicate with each other in the brain. There are different types of brain waves and each one means something different when analyzed. More accurately, there are 5 brain waves that are measured: Alpha, Beta, Delta, Gamma and Theta. These waves each have a different characteristic to be at different frequencies from high to low. They are measured in hertz or cycles per second. Table 3-1 below better describes the waves in terms of frequencies and meaning.

Wave Type	Location	Frequency (Hz)	States of Mind	Amplitude
Delta	Frontal Cortex	0-4 (high amplitude)	Asleep	20-200
Theta	Not Used	4-8	Drowsiness, Idling, Arrousal	10
Alpha	Posterior Regions	8-13	Relaxed, Open Eyes	20-200
Beta	Frontal Region, either side	13-30	Alert, Working	5-10
Gamma	Somatosensory Cortex	30-100	Cross Modal Sensory Processing	Not Used

Table 3-1: Brain Wave Types; A table describing the different types of brain waves, their frequencies, and states of mind

Each wave signal represents a different state of mind. For the purposes of this project, certain wave signals carry more importance than others. A simple approach to explain this would be to consider Beta waves. An athlete that is on the field during play should have his Beta waves dominate. If there is a disruption of some kind and Delta waves dominate then there might be something wrong. This is merely an example for how the system should look for discrepancies of what is normal in brain wave activity.

Difficulty comes with each individual player as brain waves can vary between people. The main characteristics may be the same, but it is important to make the data understandable so that the person in charge of reading the information can make a decision.

Brain waves can be measured using EEG or Electroencephalogram. EEG sensors will be discussed in Section 3.2.2. The next step in understanding brain waves is to how they correlate to the brain. For this project it is important to know how to measure brain waves and what they represent to the simplicity of the design. This means that considering how the data is collected, helps in the development of a system that is easy to install and does not reduce the performance of the athlete. Figure 3-2 below demonstrates where the different brain waves are measured in the brain.

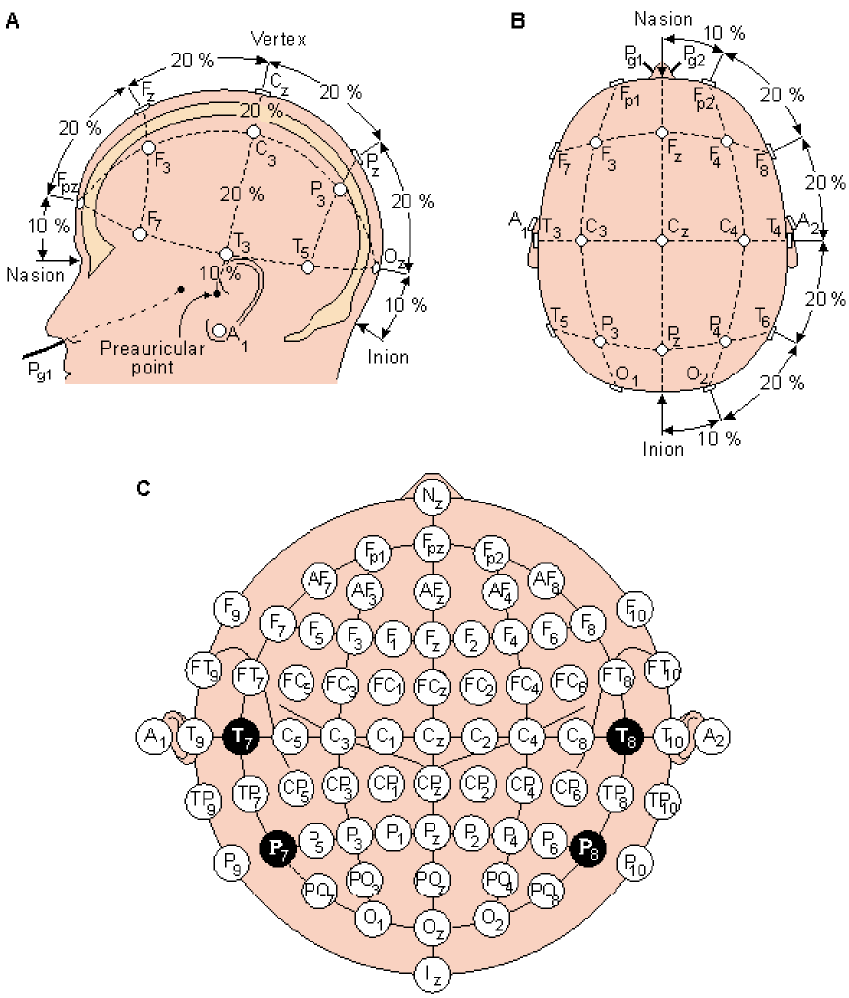


Figure 3-2: Brain Wave Location; A figure demonstrating the measurement points of various types of brain waves; reprinted with permission from BCI2000

The locations of the EEG sensors, known as electrodes, will capture the data that will be displayed as the waves. The division of the scalp into sections where each have an electrode can improve the reception of the brain wave signals. The different areas of the brain also refer to different meanings such as emotions, thoughts, etc. The location of electrodes on the scalp is also to be considered for the comfort of the athlete. The amount of electrodes that can fit into the helmet needs to be considered as well.

3.2.2 EEG Sensors

The sensors used to collect brain wave information during an EEG is called an electrode. An electrode is simply a conductor through which electricity enters or leaves an object. In the case of electrodes, the sensors are the way in which the electrical activity of the brain is used as input for the circuits. The main component to this sensor is the “metal-electrolyte interface” (citation). This electrodes use a metal and a solution (which could be a gel, or simply tissue) to measure the electrical activity that is then converted into electrical current into the circuits for further processing of the information. There are many types of electrodes in the industry. Depending on the requirements for the project each one can provide a different approach.

In order to obtain the best readings of the EEG signal, it is important to consider the type of electrode. The two most important characteristics of the electrode are the contact with the scalp and the materials. The material that makes up the electrode defines how well the electrical activity is conducted, providing more accurate readings.

3.2.2.1 Reusable Disks

Reusable disk sensors can be placed near the scalp and there can be hair in the region. Gel needs to be applied for conductivity. They are held in place by an elastic headband. Reusable disk sensors are made of tin, silver, and gold. They can be cleaned, and reused. The main benefit is the low cost of implementation. Figure 3-3 below shows an example image of reusable disk EEG sensors.

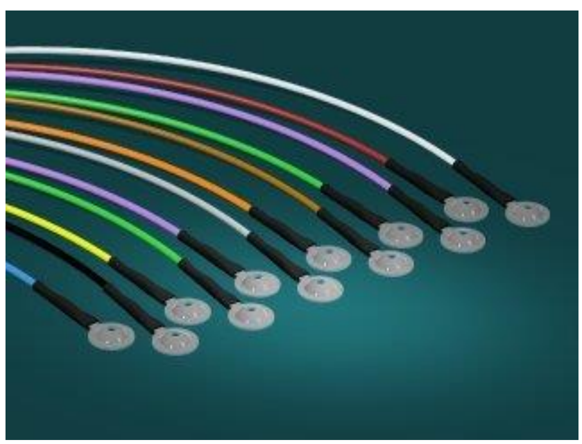


Figure 3-3: Reusable Disks; An image of reusable disk EEG sensors; reprinted with permission from Richard Kaiser

3.2.2.2 EEG Caps with Disks

EEG caps with disks come in a different variety of sensor numbers and types. They work with the reusable disks as well. The do not directly contact with the scalp, requiring more gel. These caps require much more cleaning and expense may vary depending on type and length.

3.2.2.3 Adhesive Gel Electrodes

This electrode design is very similar to the leads used in EEG and EMG. They are an inexpensive solution for recording data from regions of the scalp without hair. When purchased in bulk, the cost is very low. Figure 3-4 below contains an example of adhesive gel electrodes.

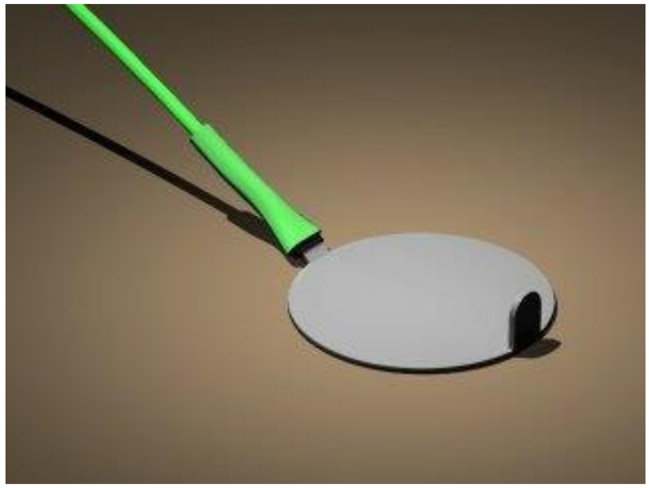


Figure 3-4: Adhesive Gel Electrodes; An image containing an example of adhesive gel electrodes; reprinted with permission from Richard Kaiser

3.2.2.4 Subdermal Needles

Subdermal needles are sterilized and used only once. They are inserted underneath the scalp tissue. They have permanently attached wire leads and are disposed completely once the procedure is done. Since they are a single use item the cost is much higher than the previous types of sensors. For these reasons, these type of EEG sensors were not considered for use in the BIO-Helmet project. An image containing an example of subdermal needles is included below in Figure 3-5.

Figure 3-5: Subdermal Needles; An image containing an example of subdermal needles; reprinted with permission from Richard Kaiser

3.2.2.5 DIY Electrodes

One of the options for EEG Electrodes is to make them from other parts to save money in the budget. The electrodes can be made and savings could be up to 90%. The design that was found for the homemade electrodes is saline design. Commercial electrodes use precious metals because they do not react (rust). Saline is better than conventional electrodes because they do not require conductive paste that could be expensive. The one problem in the design is that the screws used may rust. The part list for the building of these electrodes is shown below.

Items	Part #	Amount
Head Band		1
Elastic		30 cm
5 Core Shielded Cable	W2040	2 m
Twin Core Shielded Audio Cable	W2034	1 m
Male Plastic Coax Cable	P2021	5
Sponge Ear Plugs		Packet of 5
Heat Shrink Tubing 4.8mm	W4104	1.2 m
Heat Shrink Tubing 6.4mm	W4104	1.2 m
Table Salt		1
5 Pin DIN Plug and Socket		1
Insulation Tape		1

Table 3-2: DIY Electrodes; A table showing the parts list for a DYI electrode

The electrodes are made of coax cable and sponge earplugs. The sponges are pre-soaked in saline for an entire day. One of these electrodes can also be used as the Drivern Right Leg. These will go through the head band before assembly and then screwed to the head band. Figure 3-6 shows the building process of an electrode using earplugs and coax cable.

Figure 3-6: DYI Electrodes; An image showing the building process on a DYI electrode; reprinted with permission from OpenEEG

The wiring should be connected with a five pin din connector, which keeps it from snapping. The cable must be split progressively from a five pair to a single core for each electrode. The shielding must continue throughout the cable. The head band is a wide black ladies hair band for chemists. This headband must then be drilled for the spacing where the electrodes will lie. An image of the head band is included in Figure 3-7.

Figure 3-7: DYI Electrode Headband; An image showing the headband used for an EEG sensor mounting; reprinted with permission from OpenEEG

The final product will be some homemade electrodes that can produce good signals for this project. This would have been a nice touch of innovation for the project, but since the goal is not to create the electrodes, it was preferable not to take this approach. In the end, this adds to the research of the project and serves as knowledge on how electrodes work. The most significant factor of making the electrodes would be in the savings. Electrodes were one of the highest expenses in the project. The quality, however, of commercial electrodes versus homemade ones was too wide to even consider home making electrodes. The final result of homebuilt electrodes is included below in Figure 3-8.

Figure 3-8: Final DYI Electrode Sensor Array; An image showing the final version of the DYI sensor array; reprinted with permission from OpenEEG

3.2.3 Electrode Classifications

There are two types of electrodes: surface electrodes and the needle electrodes. Both of these electrodes have advantages to their use. The most widely used electrode is the surface mounted electrode. This is due to the relatively low impedance achieved in the surface electrodes. The most widely used electrodes are the metal-electrolyte disks.

Figure 3-9 below contains comparisons between surface electrodes and needle electrodes.

Figure 3-9: Electrode Classifications; An image demonstrating the physical differences between surface electrodes and needle electrodes; reprinted with permission from Richard Kaiser

Surface electrodes can either be flat or cup shaped. The differences in the shape or type of the electrode outputs different accuracies. For reasons of the project goal, surface electrodes were the obvious choice. Some other ideas to consider would be how comfortable and safe the system be when there are electrodes attached to the player’s head. Would their performance be effected? The goal of this project is to provide a system that is safe, does not affect performance, and capable of collecting data for further processing.

3.2.4 Electrode Impedance

For optimal readings, it is preferred to have an electrode impedance of less than 5k ohms. Anything higher than this will cause reading inaccuracies. If the impedance is much lower, then recording problems will also occur.

3.2.5 MATLAB EEGLAB Toolbox

This software contains a toolbox compatible with MATLAB. As the name, implies it is specific for the measurement and analysis of EEG, EMG, and MEG signals. This was one of the options to be considered for analyzing the data. Reviewing the needs of this project, real time analysis is vital to secure accurate information regarding concussions. EEGLAB provides the analytical power and versatility this project needs. One of the issues surrounding EEGLAB Toolbox is that there is not a real time processing feature; it is designed to work on sets of data that can be analyzed. For the purpose of testing and prototyping, this option will be used compared to other ones such as Processing. That way the best software can be used for the processing of the data.

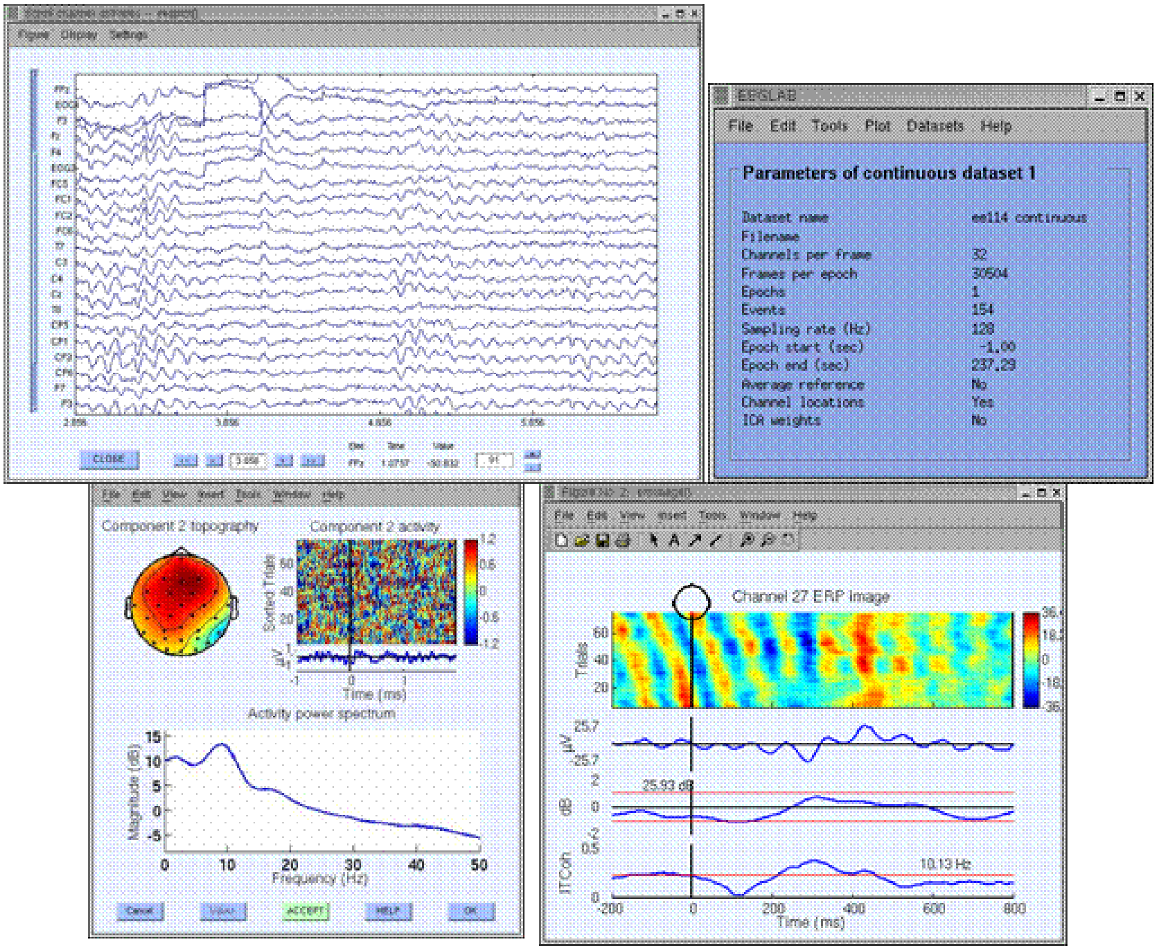


Figure 3-10: Typical EEGLab Output; A figure demonstrating the reporting of data within EEGLab; reprinted with permission from EEGLab

As can be seen in Figure 3-10, EEGLab allows the processing and display of the data in multiple windows. This serves as an idea on how to display the data for this project. The information must be collected and processed in order to be displayed. The information has to be useful and accurate for medical teams to be able to make decisions in real time about their players. A window displaying the EEG readings is necessary, but not any person is qualified to read this signals. The system must be able to read the signals and detect changes in them. Some of the approaches that this project will take will be the calculation of the ratios of the signals. Alpha and Beta waves are the brainwaves that involve being drowsy or alert. In the drowsy driving study discussed in Section 3.1.5, the researchers check the ratios of these signals to determine drowsiness. A same approach will be taking in this project. The main difference will be that the individual will not have to manually calculate the ratio, the system will do that automatically. This ratio will then be displayed in an understandable way for the user. Blinking is also an important factor that can be measured. Keeping count of much blinking the athlete is experiencing is important to help in the detection of a concussion. EEGLab helped in the development of UI for the system to be user friendly providing accurate information.

With regards to the software portion for MATLAB EEGLab, the project has the data collected into the microcontroller. That data is processed through a DSP that amplifies and filters the signal. Finally, going into the analog to digital converter pins. The signal is measured in points with respect to a reference that together will compose the response off the brainwaves. An algorithm was developed order to process the data to a readable format such as a graph. EEGLab allows the user to do this by processing the data in sets that will be processed through scripts. There are functions within the toolbox that allow the representation of the data. These functions include graphs, brain imaging, and others. The main goal of the project was to be able to display the graphs with the data sets from the signal, and perhaps include an image of the brain that would demonstrate the impact location. Figure 3-11 below includes an EEG graph output example.

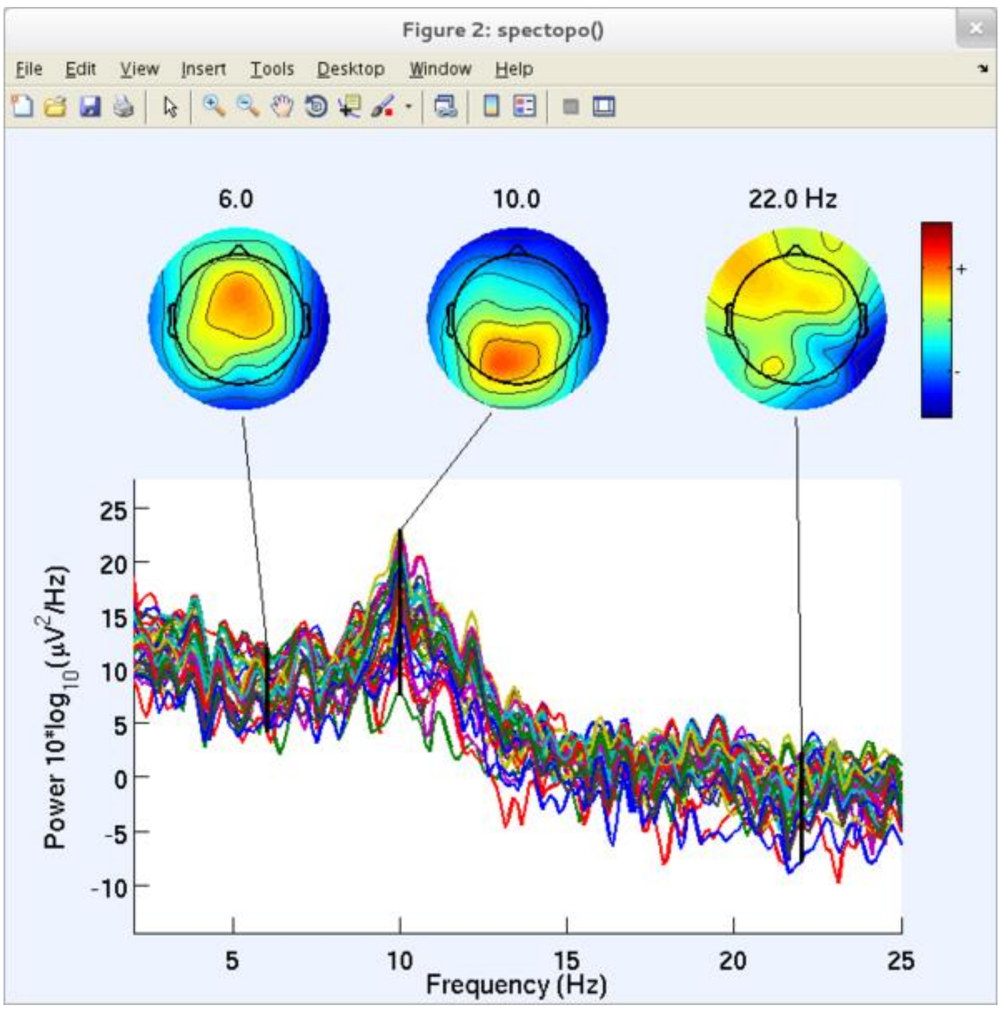


Figure 3-11: MATLAB EEGLab Output; An image displaying an example of an EEG output in graphical format; reprinted with permission from EEGLab

Figure 3-11 demonstrates the graphing capabilities of EEGLab. Also, multiple data sets can be processed together to obtain various signal graphs at the same time. The obstacle for this project would be reflect this information in a simpler manner. This software also allows for multiple subject display analysis. This means that data sets from different athletes could be processed at the same time. A more desirable look would be something as in Figure 3-12.

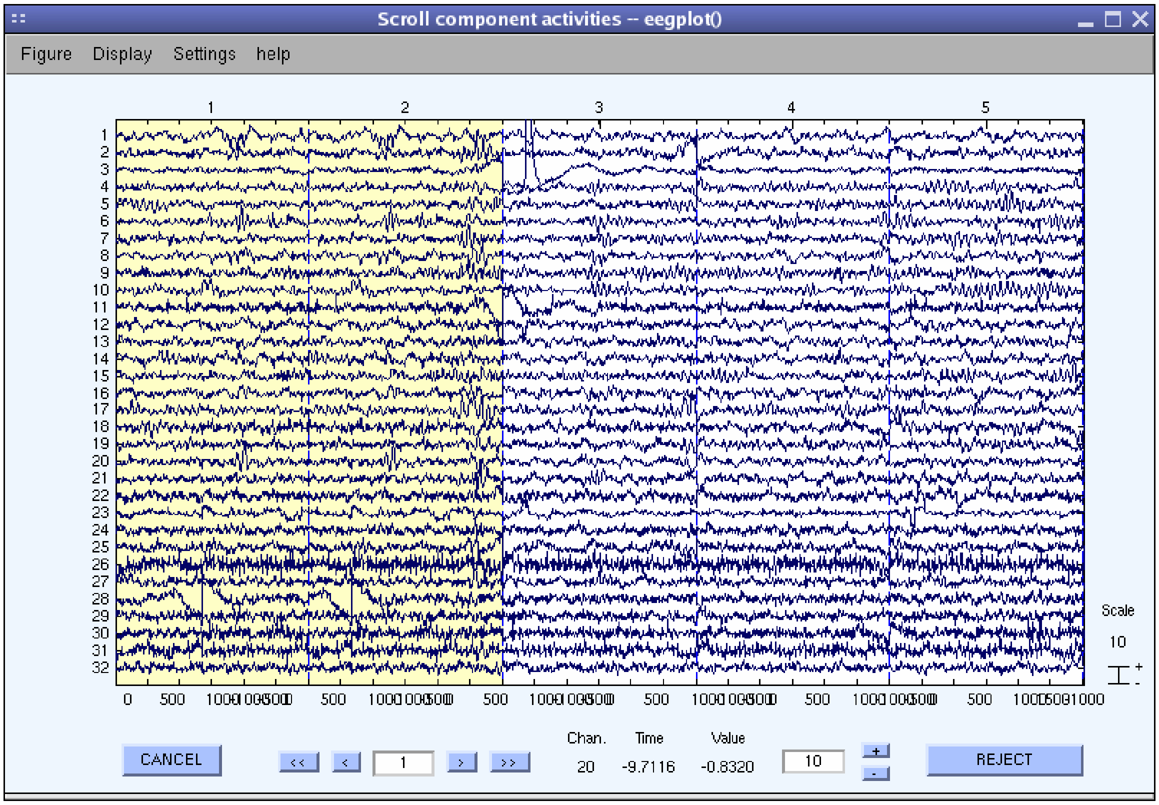


Figure 3-12: Multiple Subject EEG Output; An image containing an example EEG output of multiple subjects; reprinted with permission from EEGLab

3.2.6 Processing

Another option for EEG data processing would have been to use Processing as the software that will analyze the data. There are some benefits to using this option as Processing is open source and there is large community following with numerous example available for reference. This could lead to a better approach to the software processing of the signal data. However, the alternatives that use Processing as their data analyzer have also used a PC as their main computing driver; meaning drastic changes would have to be made to the project itself. This option will still be considered for testing purposes. Some applications of DIY EEG projects use a sound card and Processing to compute the information collected and display it on a PC. As it can be seen in Figure 3-13 below, Processing displays the EEG signal along with other graphs. Since the software is open source, the sketches needed to display the information are already available to use and could be changed to meet the needs of this project.

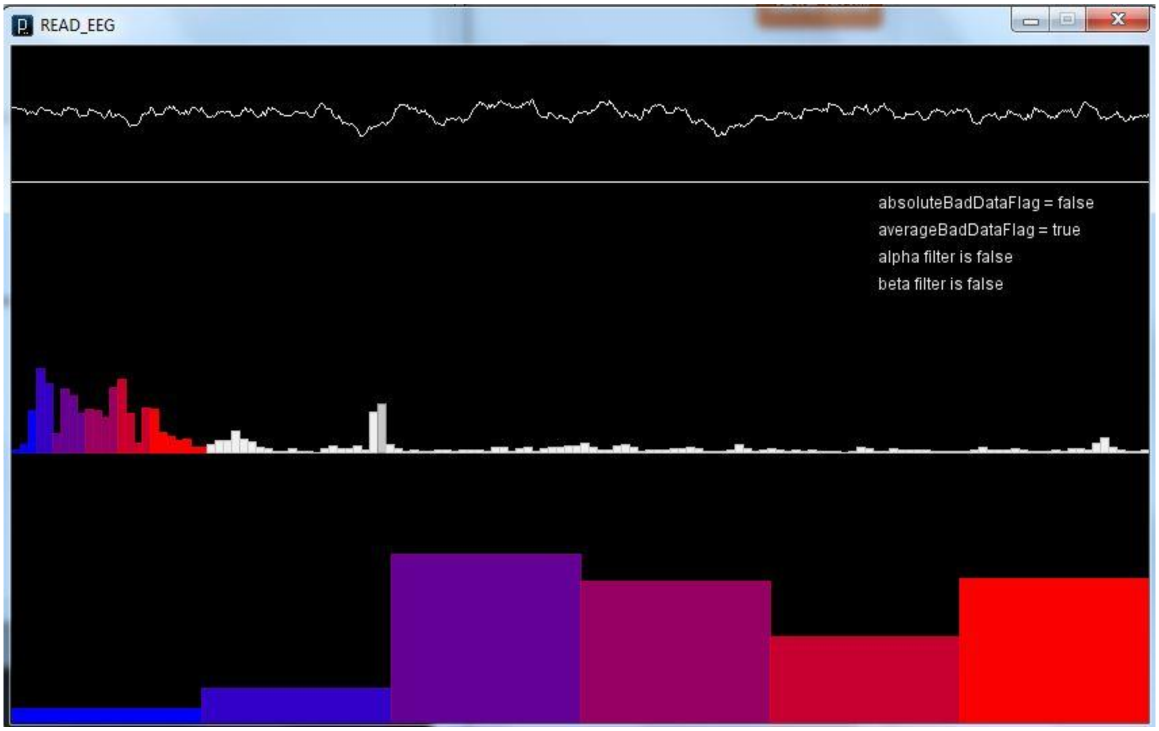


Figure 3-13: Processing EEG Output; An image demonstrating output graphs of EEG data generated by Processing; reprinted with permission from Processing Foundation

3.3 Accelerometer Sensor Array

The following subsections discuss research related to accelerometers and the implementation of an accelerometer sensor in the BIO-Helmet.

3.3.1 G-Force to Concussion Relationship

The relationship of the amount of g-force to the likelihood of a concussion occurring can actually vary quite drastically. Concussions can occur at g-forces as low as forty g's but may not occur at g-forces as great as eighty g's. This phenomenon can be attributed to several different factors that constitute a hit to a helmet. First and foremost we have the impact velocity of the hit. The higher the velocity, the greater the g-force. Secondly, the location of the hit on the helmet has an effect on the likelihood of a concussion.

Figure 3-14 below shows four main impact points that most commonly result in a concussive impact for NFL athletes. Site A is a hit directly to the face mask, site B is a hit to the side of the face mask, site C is a direct hit to the side of the head, and site D is a hit to the back of the head. Using a single six axis accelerometer the point of impact can be reliably determined and used to better determine the impact severity.

From the impact velocity there are two measurements that can be good factors in determining a concussion; peak linear acceleration and rotational acceleration.

Through research performed using HITS (Head Impact Telemetry System), the peak linear acceleration was found to be particularly important in the diagnosis of head injuries and therefor is what the bio-helmet will be using to help determine the potency of a hit. Through the data gathered by the HITS system it was found that the average peak linear acceleration for concussive impacts is ninety-four plus/minus twenty-eight g's. Using this information a threshold can be set at around ten g's that indicates the possibility of a concussion.

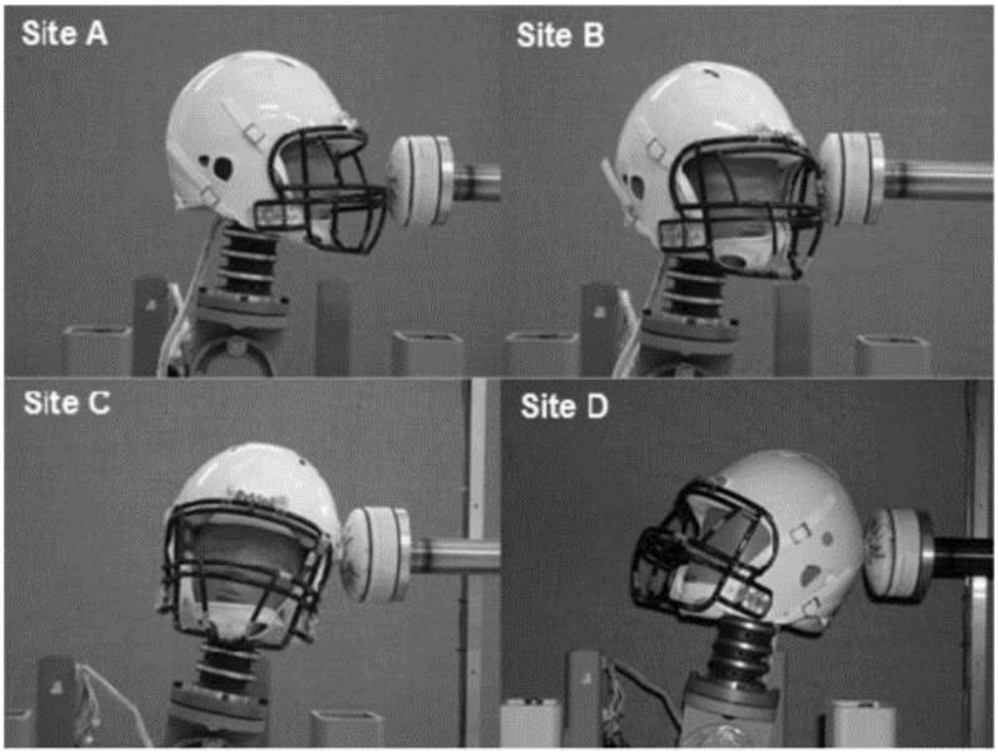


Figure 3-14: NFL Athlete Impact Points; A figure demonstrating the most common impact points that cause a concussion; reprinted with permission from National Library of Medicine

3.3.2 Accelerometer Sensors

For the BIO-Helmet, a certain criteria was needed for the project to function properly. First, an accelerometer that is able to measure high g-forces is a must. Most concussive impacts start to occur at around ten g's and can max out at over one-hundred g's for a particularly hard hit. Therefore, a sensor that can read up to two-hundred g's was required to prevent the clipping of data at higher g forces. Secondly, because this system was mounted in a helmet which dictates the need for a small battery, the less power the accelerometer uses the better. Third, it must interface with the microcontroller and provide enough resolution to accurately gauge the force of impacts in real time.

3.3.2.1 Digital Accelerometers

The digital accelerometer outputs data that is very easy to interpret being that in most cases a 8-32 bit signed integer is the output. They are also extremely power efficient using only around 0.1 micro amps in standby mode and forty micro amps when measuring g force. If the BIO-Helmet were using a fully digital microprocessor this would be the best option, but upon further research there are a few problems with the digital accelerometer as it relates to this project. The high g-force tolerance needed for our project is difficult to find and expensive in a digital accelerometer. It also creates the need for a separate interface for the microprocessor to be able to read the data correctly. Many of the digital accelerometers output their data in a serial format which may be difficult to sync up correctly in the real time application that the BIO-Helmet required. Size was also a factor and the digital accelerometers are slightly larger than a standard analog sensor due to the extra hardware required to convert the signal to a digital serial output.

3.3.2.2 Analog Accelerometers

After much research, analog accelerometers were the best choice for this project. They are purchasable in variations that can read up to a couple thousand g's as well as being tolerant to higher shocks that can occur such as dropping the helmet on to a hard surface. This is an important factor to consider being that helmets are typically thrown around on the sidelines and frequently dropped. Being that the microprocessor used in the Bio-Helmet has its own Analog to Digital conversion inputs, reading the data from the accelerometer is fairly simple. The size of the analog accelerometer is smaller than a dime in most cases and was easy to mount and implement inside of the BIO-Helmet. Low power draw is another positive factor to the analog accelerometer. Compared to the digital accelerometer the power draw is almost a magnitude greater but at about three-hundred micro amps it is sufficiently low enough for the needs of the project.

The output of the analog sensors is determined by the amount of axes that need to be measured. For the purposes of the BIO-Helmet a three axis accelerometer was used to read the data for the positive and negative x, y, and z axes. Each axis outputs its own signal which gives a sensitivity of 6.5 mV per g for the chosen accelerometer. By using this setup that has all the axes contained on a single chip, it was easier to mount in a helmet because there is only one point of contact to worry about, and the calibration is far easier because you do not have to take the positions of multiple accelerometers into account.

3.4 POWER SUPPLY

The following subsections discuss research related to the power supply system of the BIO-Helmet. Different battery types and charging approaches are discussed to determine the best battery and charging paradigm for the BIO-Helmet.

3.4.1 Battery Types

Upon further research of the different types of batteries needed in order to operate the BIO-Helmet properly, and with decent longevity, many different types of batteries could have been used to power the device; some, however, do seem to be better than others for the implementations required in this project. A power supply is needed that was either easily replaceable or rechargeable, very ergonomic, as it needed to fit in a very flat area while still leaving enough room for proper safety padding present in a regulation NFL helmet. This system must also give the proper voltage needed for multiple microcontrollers with Wi-Fi connectivity and, most importantly, have to fit within the project's budgetary constraints.

3.4.1.1 Alkaline Batteries
Alkaline batteries were looked very highly upon due to the fact that they have a long shelf life and they can provide a high yield in power compared to some of the other choices. In this project these batteries would have been disposables due to the fact that they cannot be reliably recharged and reused on a normal basis. However a big plus for alkaline is that they are very cheap and easily replaceable. It was decided against using these batteries due to the fact that they have to be replaced would not form well to the space constraints of the design. Another reason it was chosen not do use alkaline batteries was the additional weight impact due to having to use between four or six AA batteries. Because of all of these negative constraints, it has been decided to consider other battery technologies.

3.4.1.2 Zinc-Carbide Batteries
These batteries had not been met with much enthusiasm due to the fact that they have a very low output and the project design may include three boards to power. However, they are appealing due to their extremely low price. These batteries would be easily replaceable after every game, if needed, and would still leave plenty of money left over in the power supply category of the project budget. If these batteries were chosen, it would not matter if defective batteries were encountered, they would be economically easily replaced. However, the same drawbacks found in alkaline batteries would still persist here. These batteries will weigh our device down too much to be feasibly used in a situation on the football field. However, for prototype purposes there is a real possibility to test the BIO-Helmet with these batteries before possibly endangering a more expensive battery. In turn these batteries could have been used to test an early version of the prototype but were not be used in the final display of the prototype in summer 2015.

3.4.1.3 Mercury Batteries
Although mercury batteries have a very low voltage output, these batteries were considered due to the fact that their voltage does not drop almost the entire life of the battery. The battery will output 1.35V until it reaches below 5% of its battery life. Using these batteries, the BIO-Helmet would have been able to reliably perform throughout the game and not have to consider fluctuating voltages, as may be necessary with other battery technologies. However, due to their unpopularity and the fact that mercury is quite poisonous to humans this battery were not considered for use in this project.

3.4.1.4 Lithium Batteries
Lithium batteries are the front runner in the research for the perfect power source for the device. It is rechargeable, so the same batteries can be reused over and over until the life of the battery is depleted. The drawback to using this battery is that the device will have to be changed to create some charge port or to implement a way to make wireless charging extremely portable. However, due to the fact that wireless charging is extremely wasteful, the charging port was considered. The fact that it is not very feasible to make a charge port that will withstand the elements

and the sweat from the athletes wearing the BIO-Helmet does pose a problem for the project budget. The prototype was introduced with a more traditional charging port that would be much cheaper than developing a port using a much more expensive metal that will withstand the elements and human fluids typically exposed during a football game. However, these batteries can output more than acceptable voltages and they are available in a variety of shapes and sizes (special thanks to the cell phone and tablet market). It was easy implement these batteries in the device and, due to their portability and weight, they fit within the project specifications.

3.4.2 Battery Charging

The research conducted has found several ways to recharge the batteries contained in the BIO-Helmet. Wireless recharge once the helmet is placed within the charging station, recharge simply by plugging it into a port, or by having the motion of the athletes charge the batteries. Every charging option has its drawbacks, however, we must take into account that there is a manager that is responsible for the teams gear so the fact that the players themselves might not put the helmets to charge is of little concern when dealing within the design aspects.

3.4.2.1 Wireless

When recharging wirelessly, magnetic resonance can be used in order to charge the device. Although this is a very “cool” method and the fact that physical wires will be eliminated completely, the efficiency of such a method is a design worry. An electrical engineering project should try to eliminate waste and think of a better future for the world. The fact that thirty to fifty percent of the power will be lost in the charging sequence, puts a strain on the power companies to burn more fossil fuels.

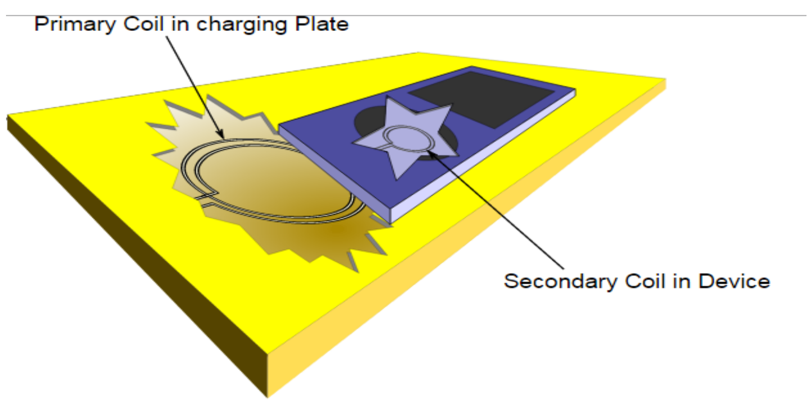


Figure 3-15: Wireless Charging; An image demonstrating wireless charging; reprinted with permission from Egmason

3.4.2.2 Motion-Based Charging

Motion-based charging is a very good and “cutting-edge” method of recharging. The fact that a huge leap in medical technology was made and at the same time made completely eco-friendly might bring huge publicity for both the team as engineers and UCF as a school. The only drawback to this is method, is that the project of creating motion-based charging is a Senior Design worthy project in its own. To avoid overreach of personnel and time resources, this method was not considered, however, the actual implementation might be put to better use on the final design and not on the prototype.

Figure 3-16: Motion Based Charging; An example implementation of motion based charging; reprinted with permission from Richard Wheeler

3.4.2.3 Plug-to-Port Charging

Plug-to-port charging is the most efficient method to charge a battery. It is also the oldest method. Plugging it into a power source to charge is the most basic and elementary method to charge the BIO-Helmet batteries. The only drawback to this charging method was that there must implement a way to make the charging port withstand the weather and bodily fluids, as this was on a helmet of an athlete that will experience sweat and blood in both rain or shine conditions.

3.5 MICROCONTROLLER

The search for the perfect microcontroller was perilous. Choices had to be made between functionality, hardware, software, and support for each device. The necessary specifications were that a microcontroller was needed that could handle a live stream of data from different sensors at once. It needed to have wireless data transfer capabilities and something that can possibly put two or three in tandem. Lastly, something with an online community so that it will make our jobs of learning and testing a lot easier because we will have a footprint from other users to build off of.

3.5.1 Atmel

The first microcontroller that was considered for this project was the Arduino. For its functionality and ease of use, this was the primary choice. The online community that can help with this is vast and it is a very well-known board to build off of. However, the fact so much data needed to be processed in real time made this choice inadequate for the design requirements. It was also agreed that the 16MHz clock speed was going to be too low and the real time data requirement made the Arduino an impossibility for this project.

3.5.2 Tiva C

Although relatively new compared to the Arduino, and nowhere near the amount of online community for it, the Tiva C was a strong choice this project. With a clock speed of 120 MHz, 1 Mbyte of flash, and 256 Kbytes of SRAM, it is a very powerful microcontroller. This was microcontroller originally recommended to the project team by Dr. Richie. Although the popularity of the Tiva C is nowhere that of the Arduino, the fact that it is manufactured by TI provides stability and reliance on the TI ecosystem, abundant documentation, and human points of contact throughout EECS.

Another advantage with regards to the Tiva C is that there is already a sensor hub booster pack that can be purchased with the Tiva C. This includes three axis gyro, three axis accelerometers, and a three axis compass. Even though the price for this pack is a quite high, this illustrates the compatibility of these types of sensors and technologies with the Tiva C. Given, that TI manufactures this boost board, there is also ample documentation and resources available for software programming for these sensors. Even though the booster pack is a nice addition, standalone sensors were connected to the Tiva C due to the fact that three axis accelerometers were not precise enough for this project's design goals. With Tiva C fully implemented, it is possible to use two or three standalone six axis accelerometers along with the booster pack.

3.5.3 PCB Research

The BIO-Helmet will contain a single main board that will contain all of the circuitry for connecting the sensors, Wi-Fi, and battery charging with the microprocessor and peripheral sensors. The industry standard is to use a printed circuit board which holds all the traces and pads used to make the electrical connections between the different components. After designing and making a PCB, the parts need to be mounted to the board through solder joints. The following sections describe a few researched methods for the creation of a PCB.

3.5.3.1 Flex Circuit

Flex circuit is a technology developed that allows electronic parts to be mounted on a flexible substrate. The BIO-Helmet might have benefited from this technology because it will be mounted inside a helmet. The flex circuit would allow for form fitting the circuit board to the back of the helmet to maintain space. Figure 3-17 below shows an example of the flex circuit.

Figure 3-17: Flex Circuit Example; An image containing an example implementation of a flex circuit; reprinted with permission from Steve Jurveston

The flex board is typically more expensive than a rigid PCB, which could have cut into the project's budget. The few manufactures that were researched would be charging \$100 to \$200 for a board of the BIO-Helmet's size. There can also be issues with tearing in flex circuit as well as changes in resistance if the circuit board is bent to an extreme. In the end, the flex circuit could benefit the project with its flexibility but the cons outweigh the pros in the case of the BIO-Helmet. Given the time constraints of the project, it would be extremely difficult, and costly, to work out all the requirements of implementing a flex circuit into the final design.

3.5.3.2 Rigid PCB

The rigid printed circuit board is the standard for many electronic devices. The rigid PCB can have multiple layers of traces allowing for very complex electrical circuits to be built in a fairly small area. This trait was especially important for the BIO-Helmet, being that the board was mounted inside the helmet, and under the padding, to avoid any damage to the electrical components.

With the curve of the helmet, the board needed to have a small footprint to fit internally. As the area of the board increases, it will sit farther and farther off of the inside of the helmet. If the board sits too high off the helmet, it could potentially cause discomfort to the user, or in a worst case scenario, render the helmet unusable. Initial estimates of the board size required to fit inside the helmet would require a width of 5cm and a length of 5cm. Figure 3-18 shows a small printed circuit board design that would be similar to what is required for the BIO-Helmet.

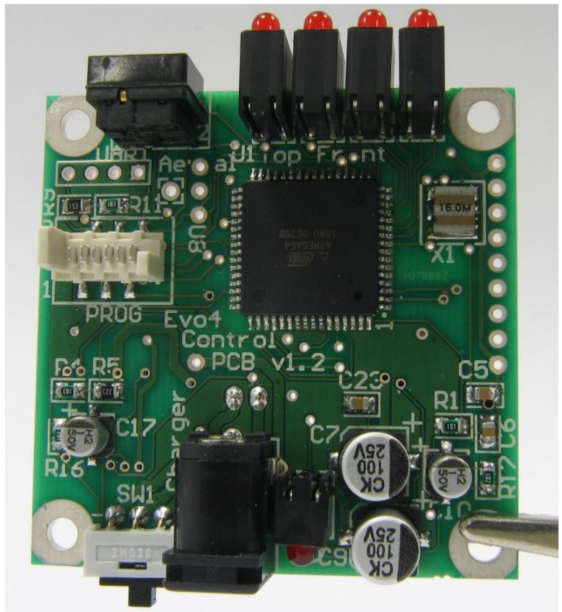


Figure 3-18: Rigid PCB Example; An image containing an example of rigid PCB; reprinted with permission from User:Mike1024

The printed circuit board is relatively cheap with a cost of around $50 to prototype a board. This would fit nicely within the budget. Compared to the flex circuit, the standard printed circuit boards are easier to design and easier to manufacture. The methods for designing for the rigid PCB are tried and true which made it a strong candidate for use within the BIO-Helmet. The surface area of the board was kept to a minimum, thus fitting it inside the helmet was not a problem.

3.5.3.3 Parts Mounting

Once the printed circuit board was produced, there was still the process of mounting the parts to the board and connecting everything together. For a standard PCB there are two ways to mount parts to the board: through-hole parts and surface mounted parts.

Through-hole mounting is the older of the two methods for attaching parts to a board. With through-hole mounting, the leads from the integrated circuit of a simple part are passed through a hole drilled in the board and connected by solder. This method ensures a good connection and has a high reliability. The cost of through-hole mounting is relatively low and can be performed with a simple solder gun by someone with little experience. The downside of through-hole mounting is that parts take up a large amount of space which requires a bigger board and less complexity in the circuit design.

Surface mounting functions exactly as it sounds. Each part is mounted on the top-most (surface) layer of the board to conductive pads and secured with a little bit of solder. The benefit to this type of mounting is that the parts can be made extremely small which allows for an overall smaller board. The downside is that it takes specialized equipment to mount the small pieces to the board which can be costly through a third party company. Figure 3-19 below shows a comparison of through-hole mounted parts versus surface mounted parts.

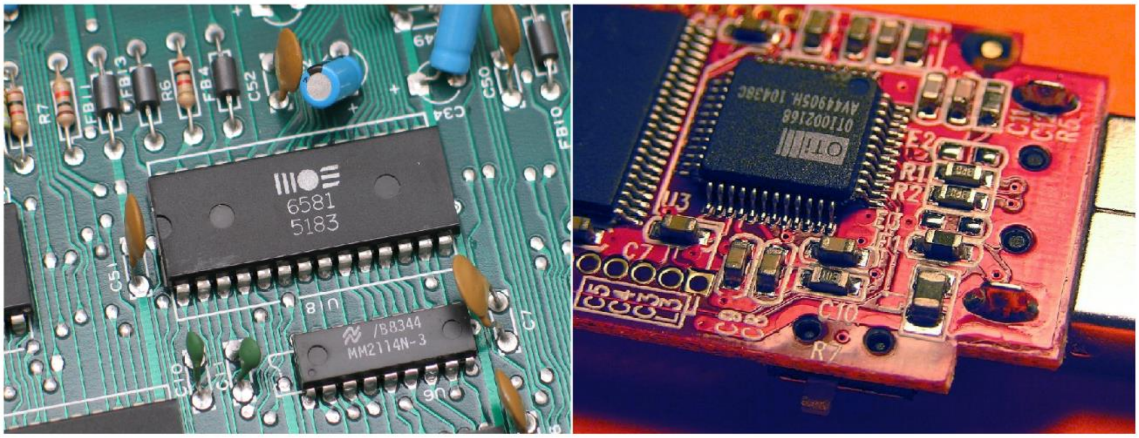


Figure 3-19: Hole and Surface Mounted Parts; An image showing a comparison of hole mounted parts and surface mounted parts; reprinted with permission from User: John Fader and Christian Taube

For the BIO-Helmet, surface mounting of the parts was preferable due to the overall smaller size. Since some through-hole mounted parts were required, those parts were be soldered to the board by the project team to reduce the cost in the budget for assembly.

3.6 Wireless Communication

Part of this project's design needs was a network interface that can connect the devices together so that data can be transmitted to an external computer where the data can then be processed and displayed. Only wireless networks were considered due to the fact that that having athletes wearing wires while in a contact sport is not realistic. The network technology must be able to satisfy several design requirements: range, bandwidth, security, reliability, and support for multiple users. The network will need to span across a football field, be able to transmit all the data necessary, and have high security. The two primary technologies being investigated are Bluetooth and Wi-Fi.

3.6.1 Bluetooth

Bluetooth would have performed quite well for this project and is very cost effective to implement. Antennas could be used to boost signal to possibly get the range required and Bluetooth uses very little energy as compared to Wi-Fi. The drawbacks are that security will be an issue, Bluetooth has more latency than Wi-Fi, and the bandwidth is much lower. Since Bluetooth did not meet the criteria in multiple factors, it was not considered for this project's design.

3.6.2 Wi-Fi

Wi-Fi meets all design specifications for the BIO-Helmet. The range for a typical Wi-Fi network is up to 200 meters; this range can be boosted further by the use of antennas. Wi-Fi more than accommodates bandwidth requirements and the security specification was fulfilled by configuring the access point and Wi-Fi chip to use WPA2 encryption standards. One drawback to Wi-Fi was the higher power requirements than Bluetooth. A higher capacity battery was needed to ensure the device stays powered for the duration of an entire game, including the possibility for overtime. The Tiva C supports an existing Wi-Fi hub interface that was referenced as a starting point for this project.

As previously mentioned, due to the realistic aspects of this project, a suitable wireless communication method was implemented to avoid athletes wearing wires while on the playing field. Due to these requirements, the more complex Wi-Fi communication system was selected to create a design that can realistically be implemented in reality, while avoiding any high level of scrutiny from the security of the BIO-Helmet.

3.6.3 Wi-Fi Hardware

There are many modules on the market that met the project criteria. Research was conducted on three modules that went above and beyond the project criteria and within the necessary cost range in order to maximize performance and minimize the chances of difficulty that could be reached by having to incorporate something that was not fully vetted by the community as of yet.

3.6.3.1 WI05

The WI05 module is an integrated 802.11 b/g/n module that offers great features at a low cost. The module itself costs around seventeen dollars, a low-powered design (100ma to 200 ma peak), easy access to physical devices, and provides a UART interface for data transfers. This module integrates the hardware MAC, radio frequency transceiver unit, and power amplifier. It supports up to three GPIOs output which will allow for control of the BIO-Helmet directly as a multi-point control unit. It is designed to support most routers and devices in the world and it is easily embedded into the BIO-Helmet system with settings that can all be completed by a web console. Lastly, it has a flexible antenna which will help maneuver it to the shape that will allow for a faster and more reliable connection. All of the features are included in a die size that fits on the top of a finger as shown in Figure 3-20. The part's dimensions are 22*13.5mm.

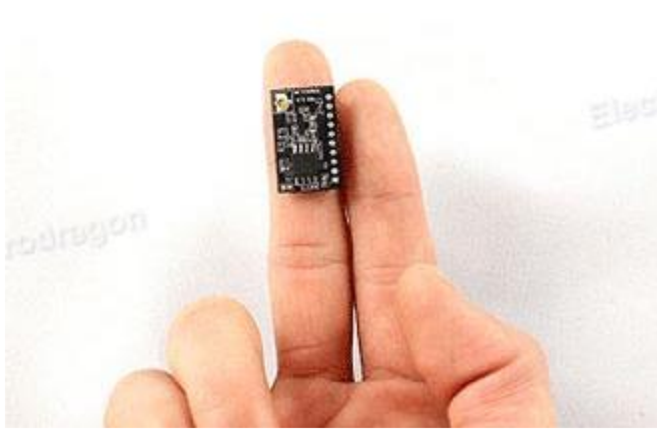


Figure 3-20: WI05 Size Reference; An image showing the relative size of the WI05 Wi-Fi module; reprinted with permission from Chow He

3.6.3.2 The Photon

This Wi-Fi module offers a bounty of features at only a twelve percent increase in costs. It is capable of 802.11 b/g/n with a soft AP Wi-Fi setup. It also utilizes the Broadcom BCM 43362 wireless module with the STM32F205 microcontroller allowing the device to control the BIO-Helmet system remotely. It offers 18 GPIO pins and UART interface for data transfers. It also has its own CPU with a clock speed of 120 MHz and 1 MB of flash memory, with 128KB of RAM. However, it comes in slightly larger than the WI05, at the size of a postage stamp as shown below in Figure 3-21. This Wi-Fi module was dropped from the possible choices for the BIO-Helmet project due to the fact it was not available.

Figure 3-21: The Photon Size Reference; An image demonstrating the relative size of The Photon Wi-Fi module; reprinted with permission from Kevin @ Spark

3.6.3.3 CC3100

The CC3100 Wi-Fi module by Texas Instruments was an extremely strong contender, offering high end features at a very low cost of only seven dollars. It features an 802.11 b/g/n radio, Medium Access Control (MAC), baseband, and Wi-Fi drivers. It offers 8 simultaneous UDP or TCP stacks to offer fast real time information transportation. The CC3100 was also the only considered Wi-Fi part that was a Wi-Fi Certified chip, which, in turn, assures its compatibility and its purpose for the BIO-Helmet project. The module also has a dedicated ARM MCU which will completely offload all the processing of the wireless protocols from the microcontroller. Another great aspect of this module was that it has low-power modes such as deep sleep and hibernate which will use only 115 Nano-amps of power and 4 Nano-amps of power, respectively. The device is contained within a small footprint of 9 mm by 9 mm and fits perfectly within the size constraints of this project. From this research, and given the fact that it is manufactured by Texas Instruments, the available documentation and customer support was invaluable for implementing Wi-Fi for the BIO-Helmet project.

3.7 Local Server

This section covers research conducted for the implementation of the BIO-Helmet local server. This local server collects and stores sensor data, over a wireless communication interface, and implements a user interface

3.7.1 Programming Languages

The research selection demonstrates the various programming languages considered for use with the local server that reads data from the BIO-Helmet. This local server houses a database, data receiving programs, and reporting software.

3.7.1.1 Python

Python is a high-level programming and scripting language implemented in C. Python was used to power the backend of the BIO-Helmet to local server communication as well as performing data processing and database insert statements. Python is a general purpose programming language which focuses on code simplicity, readability, and length. These traits, especially length and complexity, make Python a better candidate for the data receiving portion of this project over other programming languages such as C or Java. Python also incorporates OS and code level support for BSD style sockets. Socket programming with Python is discussed further in Section 3.9.5.2. These sockets will be used to read data from the BIO-Helmet output over a Wi-Fi signal. Python also provides several graphical frameworks which are considered for use in the graphical frontend of the BIO-Helmet reporting software running on the local server. Python’s standard graphic library is TkInter and is considered to be a very fast and simple GUI framework. This speed is of major importance in the scope of this project because the BIO-Helmet’s aim to provide a quicker and more substantial recognition of a possible concussion. An example of a graph drawn using TkInter is included below in Figure 3-22. Lastly, Python is cross platform with support for all major operating systems.

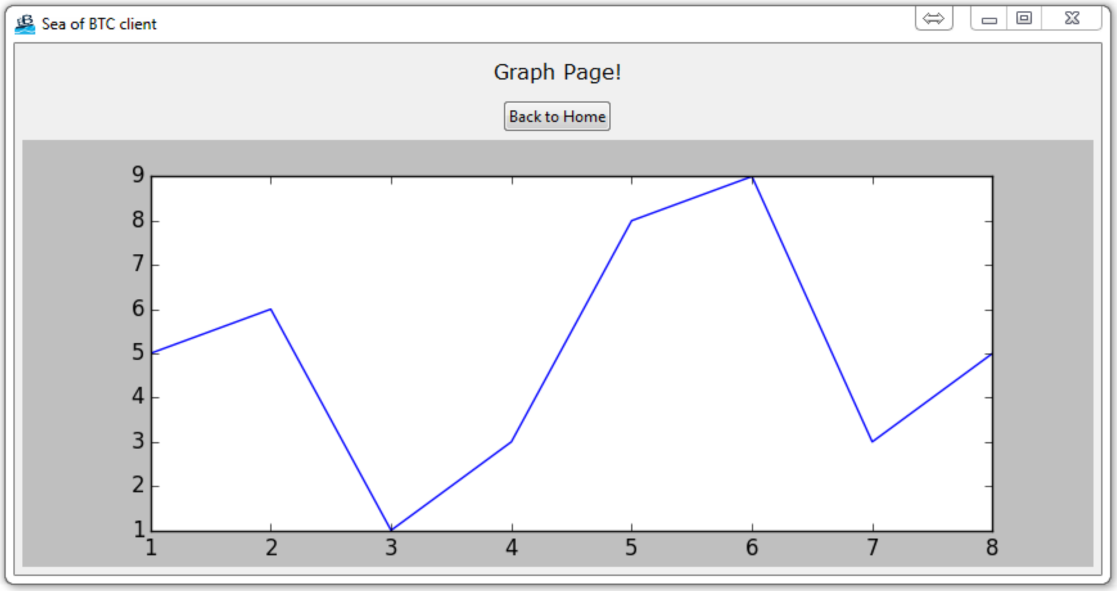


Figure 3-22: TkInter Example; An example of graphing with Python's TkInter; reprinted with permission from Harrison Kinsley

3.7.1.2 Java

Java has many well documented and traditional graphical user interface frameworks such as the java.awt.swing package. Java is considered for the implementation of the graphical user interface of the BIO-Helmet reporting software. Since Java is cross platform, support for all major operating systems can be achieved by using Java. However, the Java GUI packages are much slower and antiquated than those included with Python. This poses a problem for hardware selection in which a tablet or laptop will be used to run the reporting software. Because of the possible performance implications, Java was not selected for the implementation of the graphical interface of the BIO-Helmet reporting software.

3.7.2 Local Server Hardware Selection

Hardware selection can vary immensely for this portion of the project. All code for this project is developed with cross platform compatibility in mind. This allows the end user of the device and reporting software to choose any piece of hardware they wish. However, the hardware device operating system must be able to support the following: BSD sockets, SQLite, Python, and MATLAB. Support for ad-hoc Wi-Fi networks must also be included.

3.7.2.1 Tablets

Any convertible Windows tablet such as the Microsoft Surface Pro 3 fit the design constraints mentioned above. These style tablets run a full version of Microsoft Windows which allows for traditional desktop software packages to be installed. Other closed ecosystem tablets, such as the iPad or any Android tablet, do not support the necessary software packages as mentioned above. Also, any convertible x86 based tablet running any major Linux distribution is also available for use with the BIO-Helmet reporting software. A Microsoft Surface Pro 3 would be a typical tablet chosen for implementation of the software side of this project. This is a full x86 based twelve inch tablet which runs Windows 8.1 Pro, comes in a variety of Intel Core CPUs, and comes with four or eight gigabytes of RAM.

3.7.2.2 Laptops

Laptops (x86 based) are also a valid contender for the local server hardware selections. Operating system requirements are similar to that of the tablets with the inclusion of OS X. Laptops do have an advantage over convertible tablets when performance and system specifications are considered. The database entry processing and graphical user interface overhead may put strain on the power efficient hardware in these tablets. Laptops typically include higher powered processors than tablets and are less power sensitive, therefore sacrificing battery life for performance gain. Performance is a major consideration of server hardware choice for this project as warning reports must be shown in a rapid manner and database entries must also always complete properly as fast as possible so that the reporting module has access to this data. High performance laptops such as the Apple MacBook Pro or the Razer Blade are considered as the ideal hardware candidates for the local BIO-Helmet server. Apple's latest MacBook Pro ships with a thirteen or fifteen inch screen and an Intel Core i5 or i7 CPU with eight or sixteen gigabytes of RAM. The LCD screen within the MacBook Pro is a Retina Display with over four million pixels. A Razer Blade edge ships with similar screen sizes, however at a lower resolution, and similar processor and RAM configurations.

3.7.3 MATLAB

MathWorks MATLAB is a high level programming language and application which is designed for use in mathematical and engineering analysis. MATLAB has a large focus on graphing data sets, making it an excellent reporting software candidate for displaying both the accelerometer and brain wave data for the BIO-Helmet. MATLAB is also cross platform (running on Windows, OS X, and Linux), therefore satisfying the software cross platform compatibility requirement, preserving the option for the user to select their own preferred local server hardware. The research regarding the specific MATLAB toolbox used to graph and report the EEG data is discussed in section 3.2.5.

3.8 DATABASES

Database implementation is a critical part of the BIO-Helmet design. A database allows for all of the sensor data (both EEG and accelerometer) to be historically archived and viewed at a later time. This storage also allows both data sets to be compared from the same time interval. This is useful for a neurologist to coordinate a player's brain waves to the exact time that a hard hit occurred on the player's helmet. Research was conducted on both the relational database model as well as the non-relational database model.

3.8.1 Relational Databases

Relational databases store data points in a table format using rows and columns. Performance of relational databases are very high compared to that of non-relational databases. This performance comes from the ability to sort specific data on particular fields and use computational resources to process only the rows and columns of a particular table that is required by the query. Because of these speed improvements, and massive community support, a relational database engine has been chosen for this project.

3.8.1.1 MySQL

MySQL is published by Oracle and is the open source leader in relational database systems. It is the most popular database engine and used to power the backend of thousands of websites. MySQL is best suited for large, multiple user, and multiple client implementations. MySQL scales extremely well and can easy support large datasets on a wide variety of hardware and operating systems. MySQL is cross platform, satisfying this software requirement of this project. MySQL could be chosen for implementation in this project, however, its strongest features are something that are not necessary for storing the BIO-Helmet sensor data. The complexity of setup and maintainability are also of concern when considered in the scope of this project.

3.8.1.2 SQLite

SQLite is also a relational, SQL based database engine and is much better suited for the implementation goals and scope of this project. SQLite is a server-less engine designed for use in smaller, non-multi-user environments. This perfectly fits within the scope of the BIO-Helmet project. User setup and maintainability is also significantly decreased with the implementation of SQLite. The SQLite database is stored in a single file rather than distributed across the file system and memory, as with a MySQL implementation. The portability that SQLite offers allows the database to be moved between different local servers and provided to different reporting systems for historical analysis and athlete diagnosis. Another advantage to SQLite for this project is the ability for SQLite to be embedded with a program's environment. Separate client configuration, as is required with MySQL, is not always necessary when implementing SQLite. For these reasons, SQLite was selected as the database engine for the BIO-Helmet reporting software suite.

3.8.2 Non-relational Databases

Non-relational databases do not use a table structure; rather they are considered "flat" with all necessary data entered randomly within the database. Each data point can still be called upon, as with a relational database, however the stored data itself has no default structure when returned to the calling program. Non-relational databases are best suited for fast changing environments. Non-relational databases also have performance implications not seen in relational databases. Because there is no tabular structure, all database data must be searched when performing a query. It is not possible to restrict a query based on a particular data column or type. These performance obstacles present a major problem for implementation this project. The sensor data must be available as soon as possible to the graphing and reporting application. Database queries should not be allowed to slow down the generation of these reporting graphs. The

BIO-Helmet is also not a rapidly changing product with regards to the database structure; once the structure is designed and configured it does not need to be changed. For these reasons, a relational database structure has been chosen over a non-relational database structure.

3.9 Wireless and Network Programming

This section discusses research conducted for the wireless and network communication aspects between the Tiva C microprocessor present on the BIO-Helmet and the local server. This section covers wireless protocol research as well as specific software suites and implementation techniques used to write socket programs. Figure 3-23 below shows the TCP/IP stack. The following sections of research focus on multiple layers of this model.

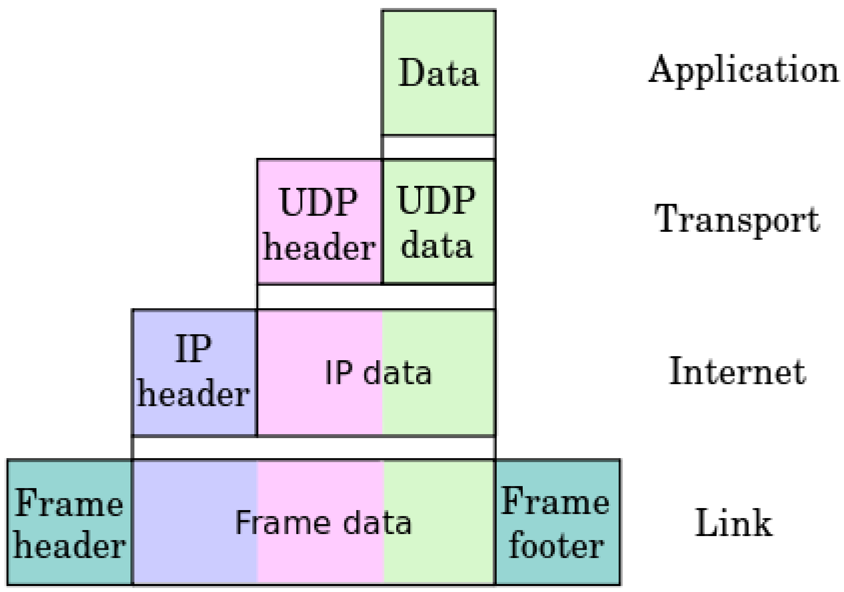


Figure 3-23: TCP/IP Stack; A figure demonstrating the TCP/IP stack model; reprinted with permission from User: Cburnett

Section 3.9.1 covers information about the link layer as it applies to this project. Sections 3.9.2, 3.9.3, 3.9.4, and 3.9.5 cover information about the internet and transport layers as they apply to the scope of this project.

3.9.1 Direct Wireless Mode

The communication channel between the BIO-Helmet and the local server was built using Wi-Fi direct or ad-hoc mode. In this mode, a point-to-point Wi-Fi network is built between two devices. A base station or Wi-Fi transmitter is not required. This is in contrast to the traditional infrastructure mode which requires a base station and traditional addressing schemes. Wi-Fi direct allows for secure uninterruptable communications between the two devices since, because of the point-to-point nature of the protocol, there are simply no other clients or devices on the network. Throughput and latency statistics can also be guaranteed because no reliance will be placed on any Wi-Fi or other network infrastructure that is beyond the control of this project or its users. Wi-Fi direct supports full standard Wi-Fi speeds and WPA2 encryption. This implementation will both secure and ensure fast bandwidth between the BIO-Helmet Wi-Fi transmitter and the local server. Wi-Fi direct has the same design constraint acceptable range values as traditional Wi-Fi, with a typical maximum of two-hundred meters. This distance is more than enough to satisfy the requirement of covering an entire football field in length.

3.9.2 Embedded Network Programming

Embedded Wi-Fi parts typically have a UART included which accepts RS232 signals from the main processor and then the Wi-Fi part handles the full TCP/IP stack, packaging, and sending of the data over the wireless signal. The main processor is unaware of the method in which the data is sent, only that it is being outputted over a UART device. This approach to Wi-Fi part design is taken by Texas Instruments and fit well with this project's design using a Tiva C processor. The Wi-Fi parts can be programmed for a multitude of wireless configuration modes such as infrastructure or direct wireless modes. These parts implement a full TCP/IP stack, offloading this process from the main CPU. This should improve CPU performance. Wireless processing, depending on the size and frequency of the sent data, can be very power intensive. These considerations must be taken into account when designing a wireless system's power delivery. Typical Wi-Fi parts also have support for encryption, oftentimes also hardware accelerated. This keeps traffic between the BIO-Helmet and the local server secure as well as prevents additional clients from connecting the local server access point. Lastly, Texas Instruments Wi-Fi parts come with built in support for multiple standard internet standard applications, such as DNS, HTTP, and XMPP. Multiple sample code and software libraries are included to help the device programmer implement these technologies into their design. These code libraries for the Tiva C microprocessor, known as TivaWare, contain support and implementations of the open source Light Weight IP stack. This TCP/IP stack is discussed further in Section 3.9.3.

3.9.3 Transmission Control Protocol

Transmission Control Protocol, or TCP, is a standard of communicating offering reliable data transfer between two hosts connection by internet protocol. Internet protocol, or IP, is a sublayer to TCP which handles IP addressing and routing. TCP exists on top of this network layer and operates a system of packet pipelining and communication. TCP operates based on a port numbering system. An arbitrary port is chosen and agreed to by both communicating hosts. A system of acknowledgments is used to check that each packet is received exactly as the transmitter sent it. TCP was the protocol of choice for the BIO-Helmet product as the implementation of TCP sockets makes communication between devices trivial and provides that the data received on the local server is as exactly was sent from the BIO-Helmet. Since wireless communication will be used in the BIO-Helmet system, accounting for packet loss and corruption, two items the TCP protocol natively handles, is extremely important. Any loss or corruption of a packet containing a high impact sensor flag will lead to incorrect results and risk that an athlete is not properly identified for having a possible concussion. A simplified TCP state diagram is included in Figure 3-24.

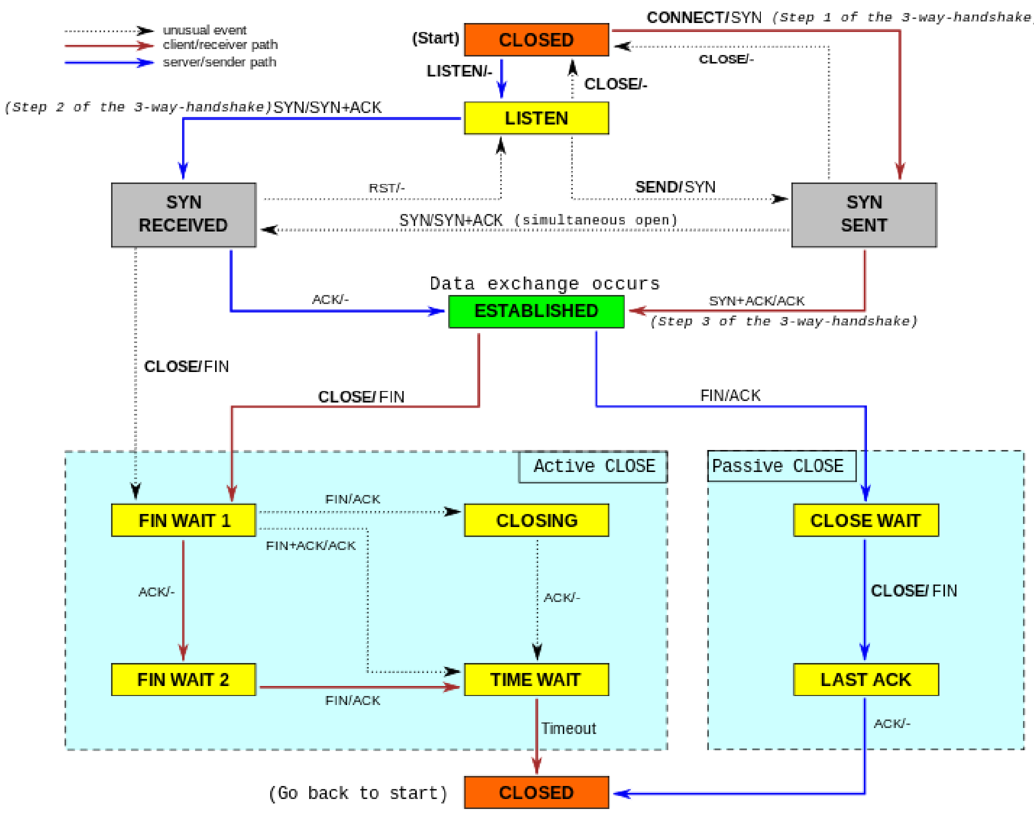


Figure 3-24: TCP State Diagram; A diagram describing the various operating states of Transission Control Protocol; reprinted with permission from Sergiodc2, Marty Pauley, and Scil100

3.9.4 User Datagram Protocol

User datagram protocol is another major transport layer protocol. UDP is a connectionless protocol (opposite that of TCP) and does not provide any data redundancy or loss packet prevention. UDP applications are typically suited for low latency applications where packet loss or corruption is not a priority. UDP sockets and programs are typically faster at processing data than TCP connections due to no need for data integrity checks or connection setup. While this speed trait did present a benefit for the BIO-Helmet project, the speed gains for sending simple text and floating point numbers (what will be sent over the socket) between the BIO-Helmet and the local server, do not outweigh the costs of having incorrect or incomplete data received. The byte size of the data sent between the BIO-Helmet and the local server is not enough to impose a speed barrier on the TCP protocol. Player safety and data accuracy must take precedence over transmission speed.

3.9.5 Socket Programming

Socket programming is the process of connecting standard running code to a network interface. Sockets traditionally refer to the TCP (or UDP) connection established between two network hosts. A socket is established by writing a client-server model socket program. The client side will initiate a TCP connection to the server over a predefined, mutually agreed, port number. In this project, a socket will be established between the BIO-Helmet and the local server. This socket will handle the transmission of the sensor data from the Tiva C microprocessor to the local server for archiving and display.

3.9.5.1 lwIP

Light Weight Internet Protocol, or lwIP, is the integrated TCP/IP stack included with the Tiva C TivaWare software suite and is fully compatible with the Tiva C microprocessor. This protocol suite also has support for BSD style sockets. This allows for a compatible data transfer connection to be opened between the BIO-Helmet and the Python data receiving script running on the local server. lwIP includes many standard TCP/IP stack features such as DHCP, DNS, optional BSD socket support, and ARP. The socket interface is especially useful for the scope of this project as it will serve as the base communication between the BIO-Helmet and the local server. The socket layer exists between the application (the data collection aspects of the embedded code) and the IP stack itself. The lwIP protocol will handle both the encapsulation and TCP sessions as well as the IP addressing and network routing provided by IP.

3.9.5.2 Local Server Socket Programming

Python has native support for operating system level BSD style sockets. Python exists above the operating system TCP/IP stack and instructs the operating system to open, accept, and send data on certain ports. The particular running Python script that opened the socket, then is able to receive any bit stream input received by the TCP/IP stack implementation present in the operating system. The specific operating system TCP/IP stack implementation varies across platforms. Python's underlying framework includes support for most major operating system TCP/IP stacks and is able to seamlessly integrate with these protocol suites. This allows for code portability between platforms, a goal of the BIO-Helmet local server software suite. A Python socket is first created by opening the necessary port on the local host and then using an API call to connect to the socket. The Python script will then listen for any TCP connections to originate on the local host device on the defined port in the Python script. To use a socket, the Python script simply reads and writes to the socket just as if it as standard in or any other input or output interface. Sockets between multiple platforms, such as that between the local server (Python based) and the BIO-Helmet (lwIP based), are capable of sending data only as byte streams. Arrays, lists, or other custom objects cannot be sent over this type of socket. Data types such as these must be unpackaged, sent over the socket as standard, bit representable data, and then reassembled into the data type on the receiving side. This should not be an issue for the BIO-Helmet project as only integers, character text, and floating point numbers will be sent over the socket.

4 PROJECT HARDWARE AND SOFTWARE DESIGN DETAILS

The following sections describe the hardware of software design details of the BIO-Helmet. The following sections are intended to be a detailed description of each hardware and software component, design parameters associated with the component, and how the components fit into the overall design of the BIO-Helmet. A high level hardware block diagram is included below in Figure 3-2. This block diagram includes the system layout for the components designed in Sections 4.1 through 4.5.

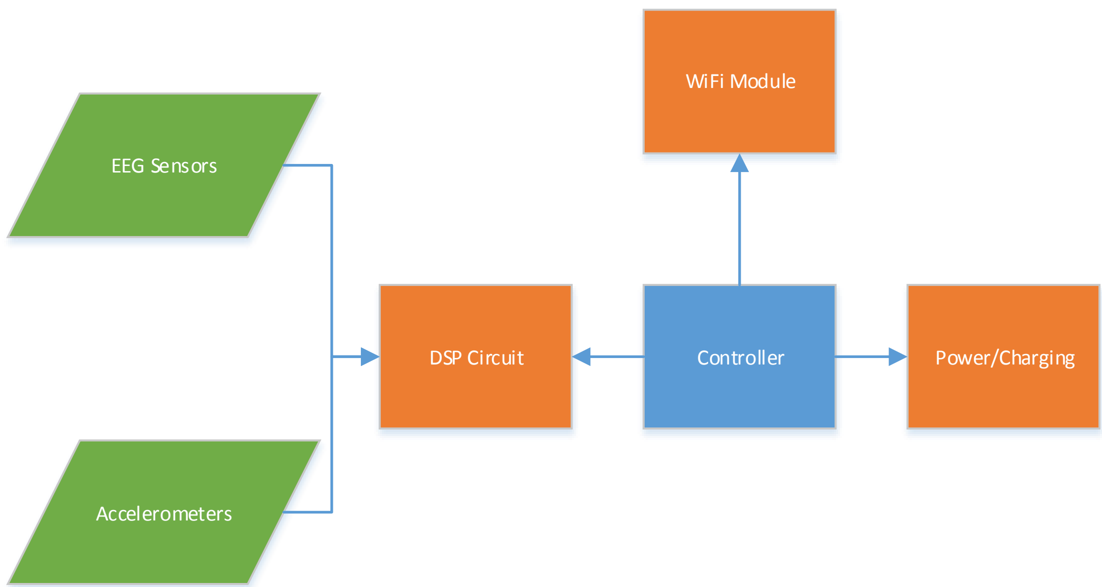


Figure 4-1: Hardware Block Diagram; A high level block diagram describing the hardware system layout of the BIO-Helmet

4.1 EEG SENSOR ARRAY

In this section the hardware design that captures the raw data from the electrodes will be discussed. The design contains several stages that will amplify the brain wave signal and also filter out unwanted noise. As previously discussed, within Section 3.2.1, brainwave signals are collected in different frequencies. Table 3-1 describes the different types of brainwaves and their respective frequencies.

Table 3-1 also contains the amplitude voltage range of the signals, in microvolts. This means that the signal needs to be amplified a couple thousand times before the processing of the data can begin. The amplification of the signal also brings another problem: the human body naturally produces a significant amount of noise and humming. It is imperative that the hardware design take this into consideration as noise can significantly reduce the quality of the data collection. In order to reduce noise, there will be various stages of filtering. Filtering of the data depends on the frequency of the noise needed to cancel. Further consideration and explanation will be given in each stage in regards to noise reduction. Other considerations for this section of the hardware design will be the selection of the components. Because of the selection of the parts for the design, other stages are required such as voltage regulators and or voltage inverters. Many amplifiers require an inverted voltage input for difference amplification. Voltage regulators may be needed depending on the power sources being used. Finally, a protection circuit was used for the protection of the circuitry and the high amplification produced. In Section 2.5.3 and Section 5, safety and manufacturability of the design will be discussed. The next discussion includes the block diagram of the EEG data collection segment.

As mentioned before, this section will focus on the part of the diagram that includes the signal processing circuit and the electrode sensors. The components include an instrumentation amplifier and several types of filters along with the possibility of integrating a voltage regulator. Breaking down this section into a more detailed diagram, the different stages needed for a processing of the EEG signal can be seen. After much research, Figure 4-2 below has been determined.

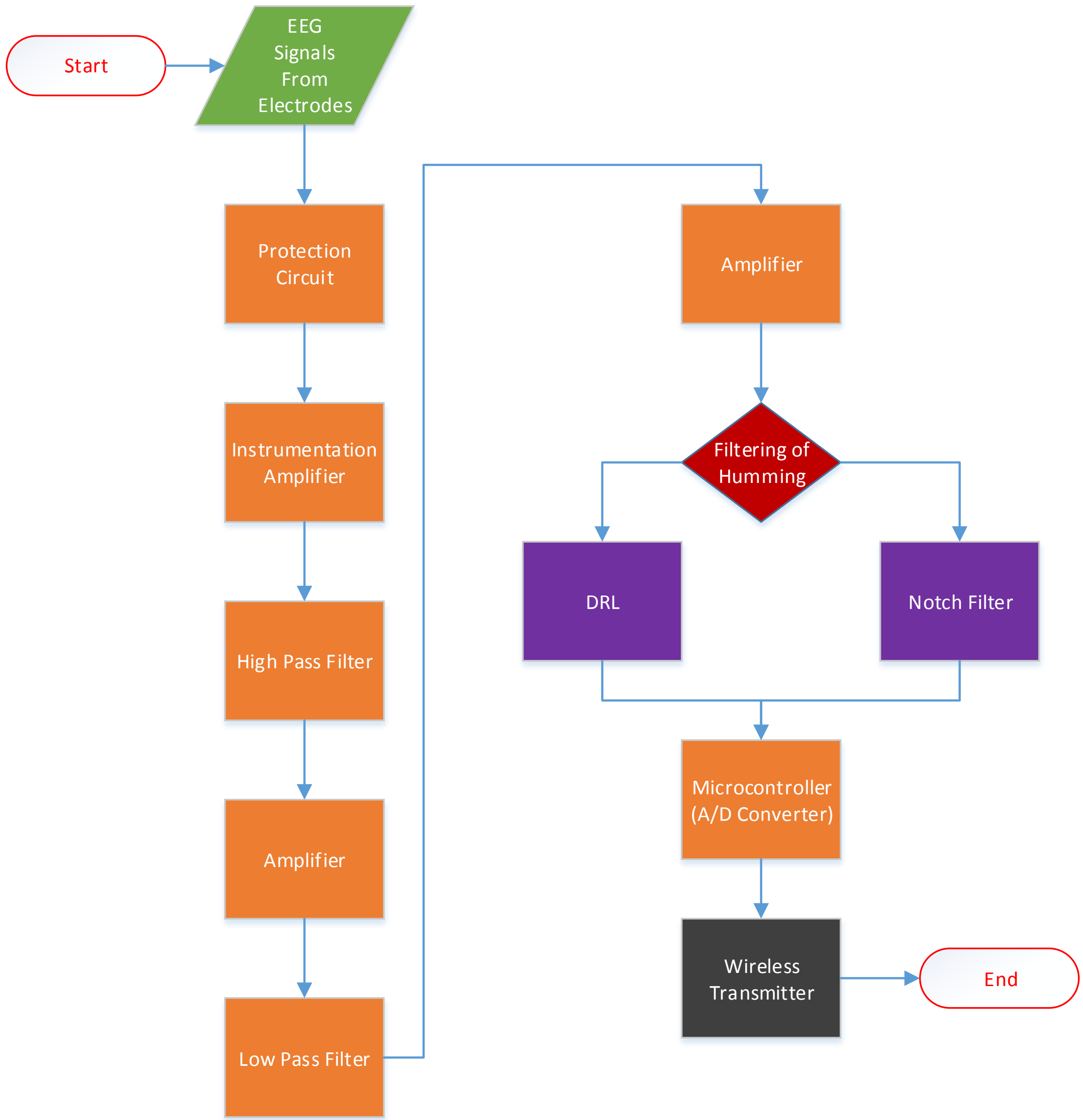


Figure 4-2: EEG Signal Processing; A diagram describing the BIO-Helmet system processing of an EEG signal

As can be seen in Figure 4-2, there are multiple stages to the signal processing circuit. The main components involve the amplification and filtering, as previously mentioned. Before the signal enters the processing section, there is a protective circuit that provides safety measures. Later on, for the filtering of the humming, there are two options. The use of a DRL or a Notch filter for the body humming noise canceling. The following is a brief but detailed explanation of the different parts that compose Figure 4-2. Each segment will include a description of why it is included in the hardware, along with a corresponding schematic.

4.1.1 Protection Circuit

The first part of the design to discuss will be the protection circuit. This circuit serves as a safety measure to protect the rest of the hardware design from electrostatic discharge and it should prevent failed circuitry. This is composed of capacitors, resistors, and transistors. The schematic of this circuit is included below within Figure 4-3.

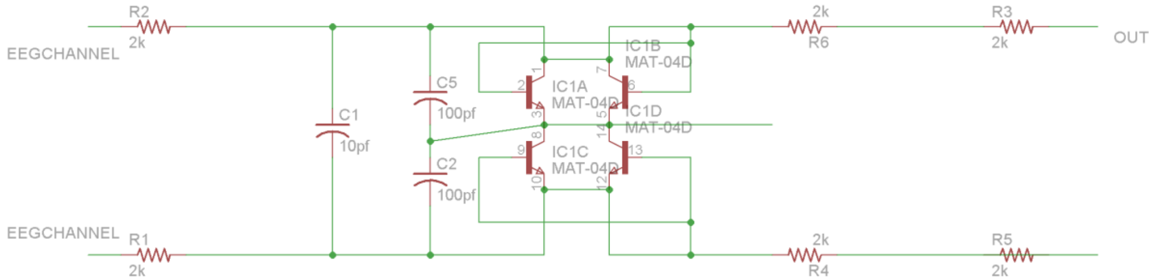


Figure 4-3: Protection Circuit; A figure containing a circuit schematic of the BIO-Helmet protection circuit

This schematic references the protection circuit found during research conducted for the Open EEG project. It contains 3 capacitors that suppress radio frequencies entering the circuit from the electrode sensors. (Griffiths, Nelo, Peters, Robinson, Spaar, Vilnai, 2003).

4.1.2 Instrumentation Amplifier

The amplification coming from the electrode signal begins by going through an instrumentation amplifier. The main reason behind using this type of amplifier is because it lowers impedance of the signal, making it less sensitive to noise (Griffiths, Nelo, Peters, Robinson, Spaar, Vilnai, 2003). This type of amplifier provides also a high CMRR (common mode rejection ratio), which allows the output value to be near perfect. Building the amplifier using components was an option, but using an IC amplifier provided better results. The options for this project were the INA114 or the INA126. Both components are precision amplifiers and can serve as good options. The project used the INA114 amplifier. This will provide a gain of about 16 times the initial signal size. The circuit schematic diagrams describing the instrumentation amplifier are included below in Figure 4-4 and Figure 4-5.

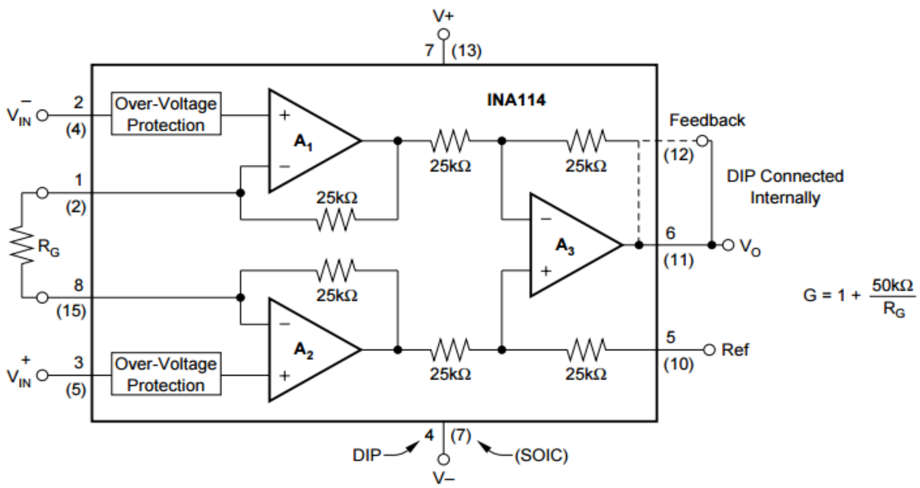


Figure 4-4: INA126 Schematic Diagram; A circuit schematic diagram describing the INA126; reprinted with permission from Texas Instruments, Inc.

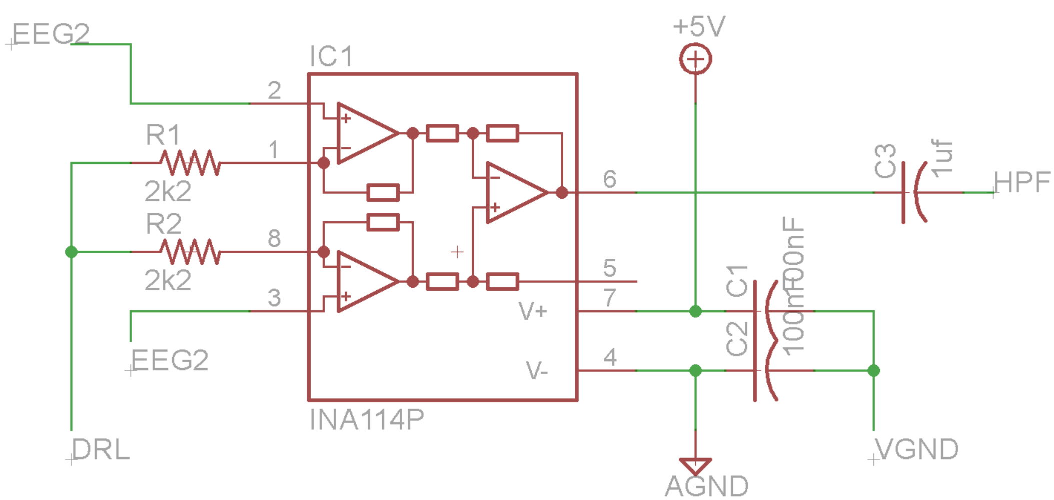


Figure 4-5: Instrumentation Amplifier; A circuit schematic describing the instrumentation amplifier

The gain for the amplifier is measured by the following equation:

$$G = 5 + \frac{50k\Omega}{R_G}$$

Where R_G will vary to produce the desired gain. The amplifier would need an approximate resistor value of 2.381Ω. The values of the resistors have to be as close as possible. Preferably with a margin of error of 1% due to the precision required in this system.

4.1.3 High Pass Filter

After the instrumentation amplifier, a high pass filter is the next stage of the circuit. This filter removes the DC offsets of the human body. The cutoff frequency is at about 0.16 Hz. The schematic for the high pass filter is included below in Figure 4-6.

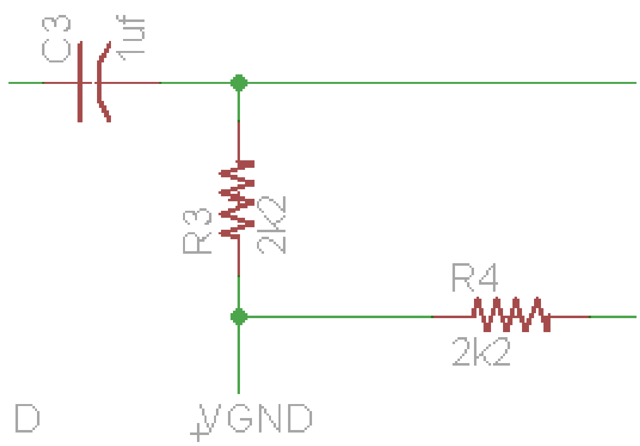


Figure 4-6: High Pass Filter; A schematic diagram representing a high pass filter used in the EEG circuit

4.1.4 Non-Inverting Amplifier

Following the high pass filter described in Section 4.1.3, there is a non-inverting amplifier that further amplifies the signal coming from the electrodes. The gain here can vary due to the resistor used, but the aim is to amplify the signal to a gain of forty. The schematics for the amplifier are included below in Figure 4-7 and Figure 4-8.

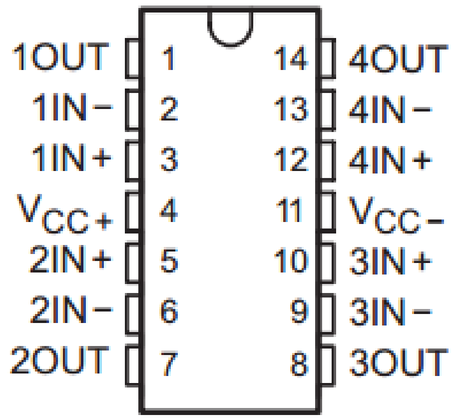


Figure 4-7: TL084; A diagram containing the pin layout for the TL084 non-inverting amplifier; reprinted with permission from Texas Instruments, Inc.

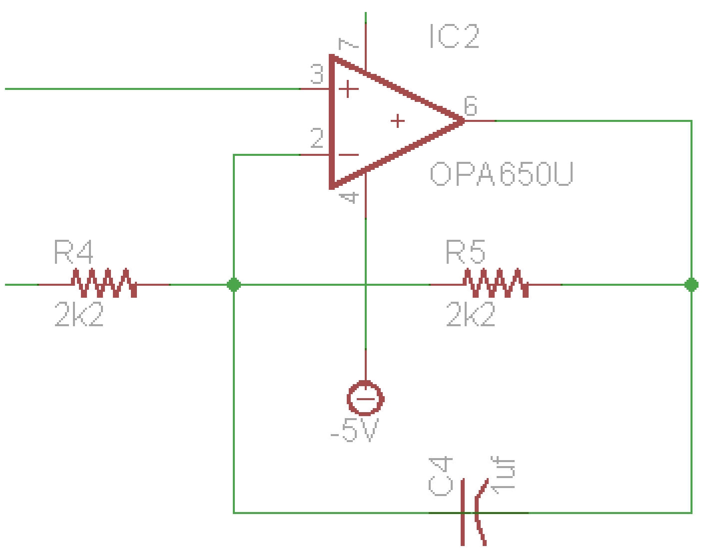


Figure 4-8: Non-Inverting Amplifier; A schematic diagram describing the non-inverting amplifier used within the EEG sensor array

4.1.5 High Pass Filter

A high pass filter is added which performs the same function as the high pass filter described in Section 4.1.3; to reduce the DC offset. The many stages of this overall circuit design are to amplify the signal but also reduce the unwanted noise present in the signal. The schematic is the same as that described within Figure 4-6.

4.1.6 Low Pass Filter

This stage of the circuit is design to filter out the higher unwanted frequencies. This is a 3rd order filter that also amplifies the signal with a gain of sixteen. This represents the final stage of amplification. The schematic for this portion of the circuit is included below in Figure 4-9.

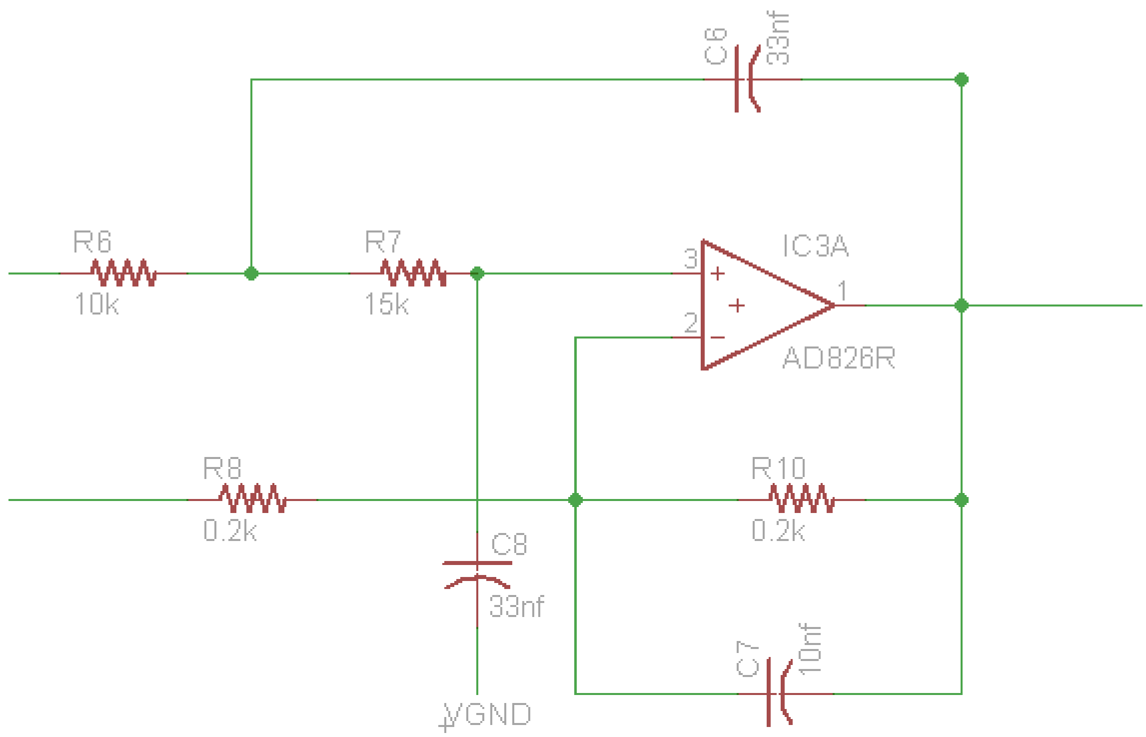


Figure 4-9: Low Pass Filter; A schematic diagram describing the design of a low pass filter used in the EEG sensor array

4.1.7 Voltage Regulator

Depending on the choice of power supply, a voltage regulator may be necessary. This voltage regular will ensure a steady and adequate power supply to the amplifiers and the microcontroller. An example of a voltage regular is included in Figure 4-10.

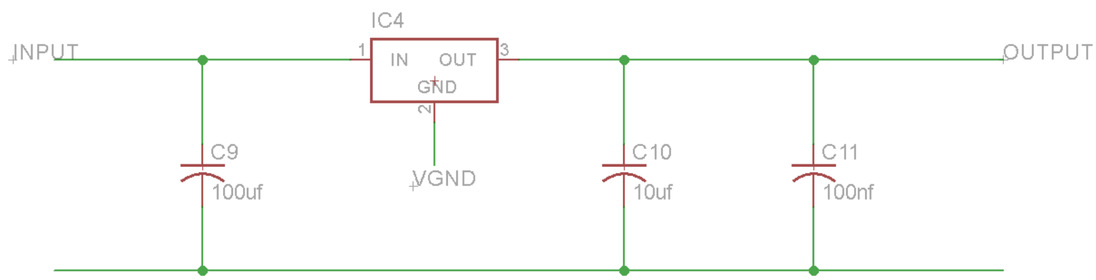


Figure 4-10: Voltage Regulator; A circuit diagram describing a voltage regulator

4.1.8 Voltage Inverter

The amplifiers need two power supply rails; one positive, the other negative. For this to occur, the power source voltage needs to be inverted and sent to the amplifiers. The schematic diagram of this voltage inverter is included below in Figure 4-11.

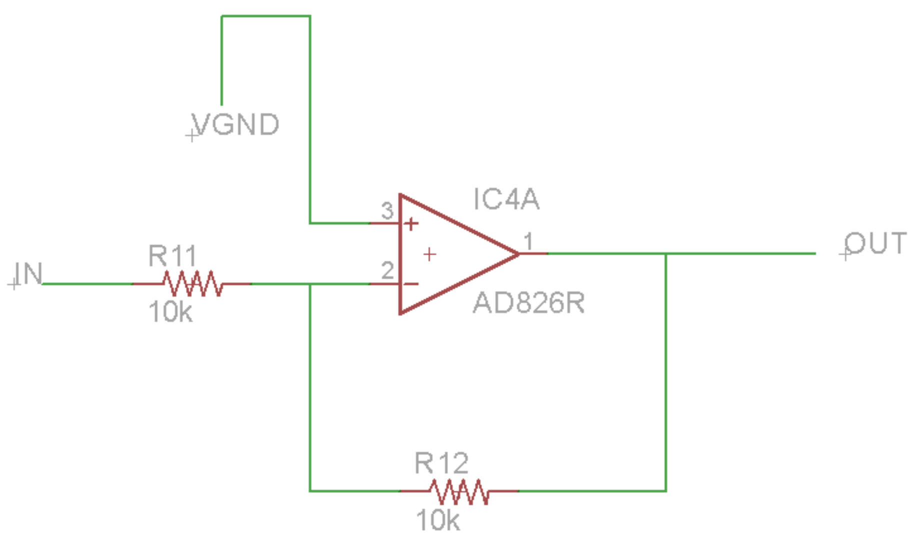


Figure 4-11: Voltage Inverter; A circuit schematic diagram representing a voltage inverter

4.1.9 Notch Filter

This filter is used to minimize the humming that comes from the brainwave signals. Since the humming is between 50-60Hz, the filter was made to block out the range desired. This can be used to substitute the low pass filter. Figure 4-12 includes the schematic diagram for a notch filter.

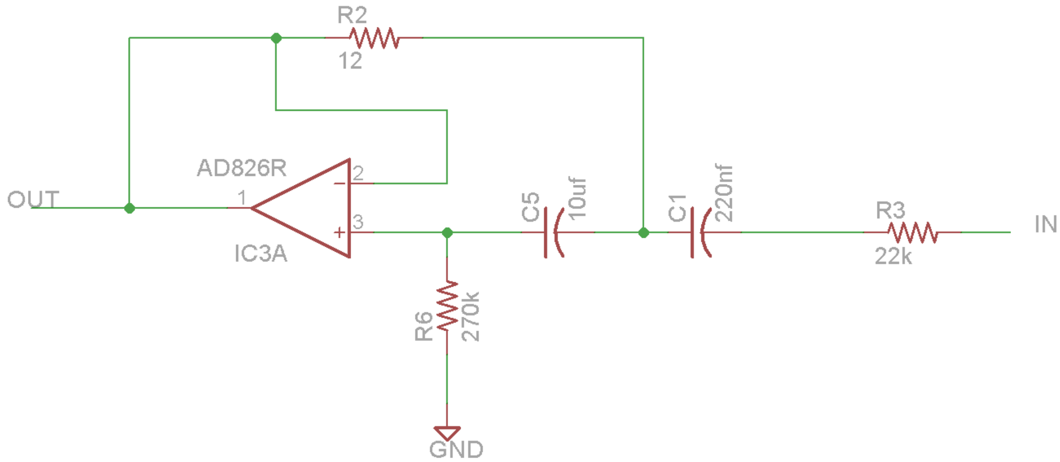


Figure 4-12: Notch Filter; A circuit schematic diagram describing the design of a notch filter

4.1.10 DRL

Short for driven right leg, this circuit is used to reduce common-mode signals such as 50/60 Hz hums and cancel them out. Previous EEG design used a ground electrode. This can furthermore attenuate main hums up to one-hundred times more than an instrumentation amplifier. This serves as another stage in the filtering process in order to obtain the best signal possible. Figure 4-13 includes the schematic for the driver right leg.

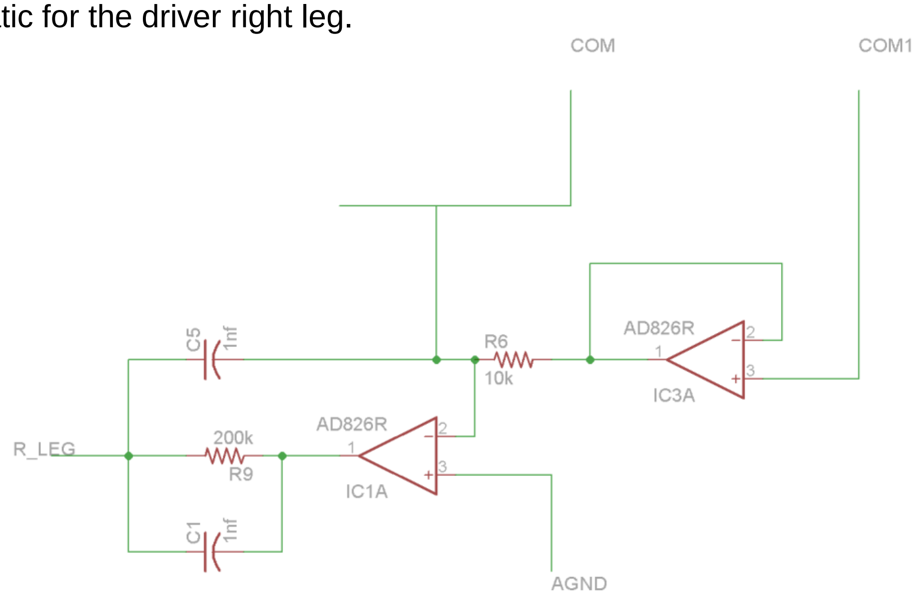


Figure 4-13: Driver Right Leg; A schematic diagram representing a DRL

4.1.11 Completed EEG DSP Hardware Design

Most of the hardware discussed above was used for the design. Some decisions were made regarding the filters. From research of other applications and projects, this design is composed of high pass and low pass filters. The amplifiers were powered by the microcontroller. There is a two channel EEG, as the project needed the alpha and beta wave signals for processing. Other signals could be included for further research capabilities and for emotional status analysis. The completed EEG DSP hardware design is included in Appendix C.1 Completed EEG DSP Hardware Design Schematic. An image of the printed and assembled EEG DSP board is included below in Figure 4-14.

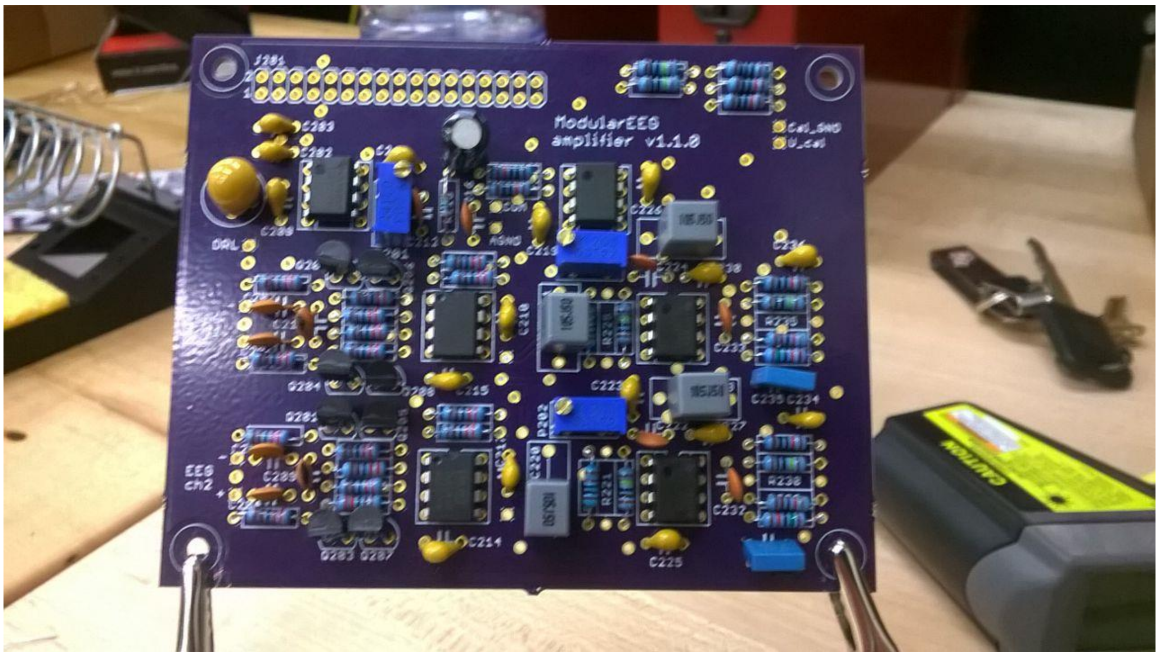


Figure 4-14: Assembled EEG DSP Board; An image containing the final assembled EEG DSP board.

4.1.12 Final Block Diagram

The design will contain, as described above, several stages of amplification and filtering to obtain the best quality and accuracy in the signals. The use of the notch filter or the DRL for filtering out the specified range of frequency was determined later on in the prototyping testing stage. This way the best choice can be taken. Included below in Figure 4-15 is the final EEG sensor array block diagram.

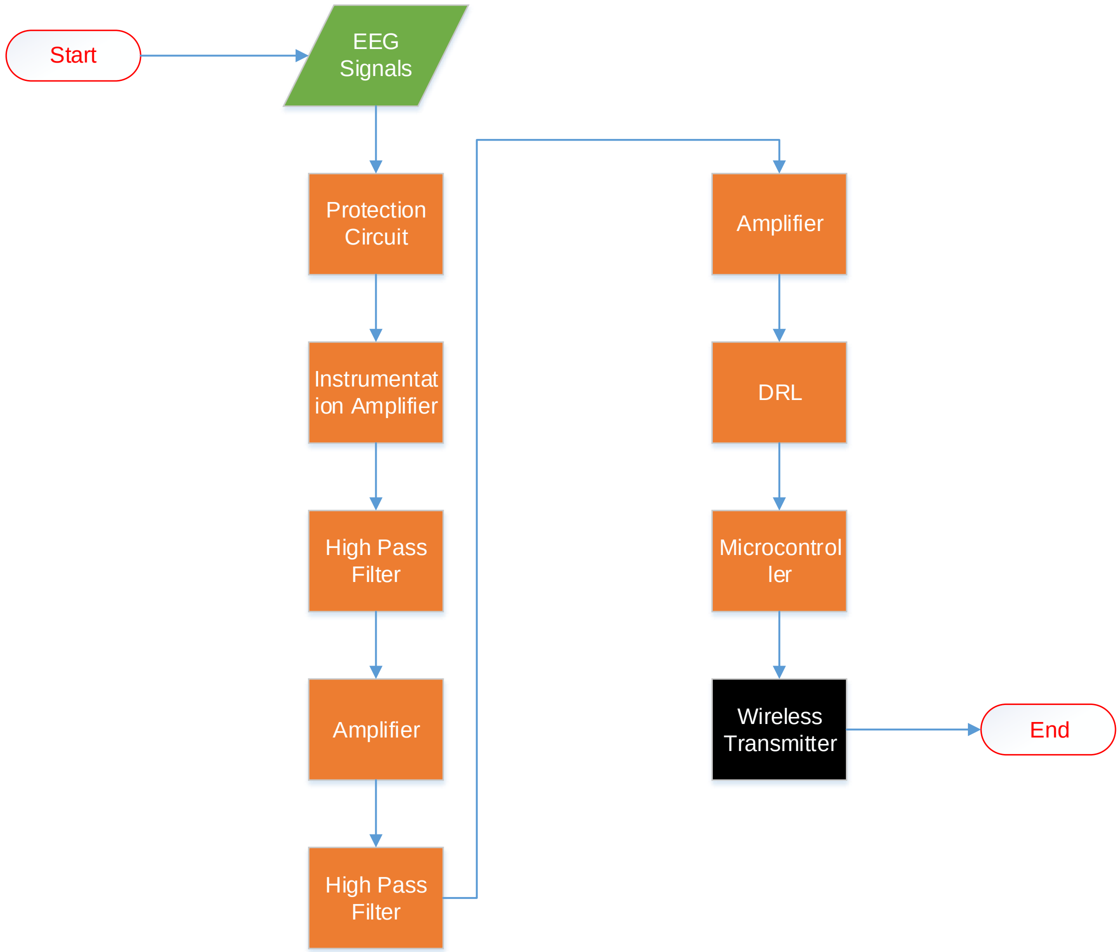


Figure 4-15: Final EEG Block Diagram; A block diagram representing the final EEG sensor array block diagram

4.2 Accelerometer Sensors

This section covers the accelerometer subsystem which was used in the measurement of impacts to the helmet. A single ADXL377 accelerometer was used. The decision to use a single accelerometer is based on a couple limiting factors.

First, the micro controller has a limit of 12 analog inputs and each accelerometer has 3 outputs, one for each axis. Not only would adding more accelerometers increase the size of the board, which is also limited by the requirement of fitting into a standard football helmet, but it would also increase the overall cost of the project.

Second, the accuracy of a single accelerometer is more than sufficient to provide meaningful data. As mentioned in section 3.3.1, the research done using the HIT system has given good data that will be used in calibrating and interpreting the data from the accelerometer. Using more than one sensor could potentially skew this accuracy if for example, the sensors are place imperfectly in the helmet or they get dislodged during an impact.

As for the data output, the microcontroller polls data directly from the sensor across three axes. The data is then used to determine the force and direction of the impact on the helmet. From this data the area of impact can be extrapolated, which is an important factor in determining the possibility of a concussion.

4.2.1 Accelerometer Mounting

The mounting of the accelerometer is an important factor in determining the location of impact on the helmet as well as calculating the linear peak acceleration. The accelerator is mounted on its own PCB and wired into the main PCB separately to achieve the mounting location shown in Figure 4-16.

Inherently, there are some errors in determining exact impact force of the brain hitting the skull due to the accelerometer being mounted on the helmet and not directly on the skull. Thanks to the research done with the HITS system, it is maintained that any impact to the skull over ten G's in force has the possibility to cause a concussion. This mounting location allows for easy determination of this threshold and report the relevant data back to the user.

This setup is beneficial to the project in several ways. It allows for use of a minimal number of inputs on the microprocessor which leaves enough open for the EEG sensor array. It keeps power consumption low because only a single low power chip is used. It makes it easy to determine the location of the hit on the helmet because each axis is relative to each other rather than having multiple sensors mounted in different parts of the helmet. Lastly, it makes installation a simple process when all that is necessary is to mount a single sensor to the top of the helmet and simply calibrate the sensor to be pointing up and down.

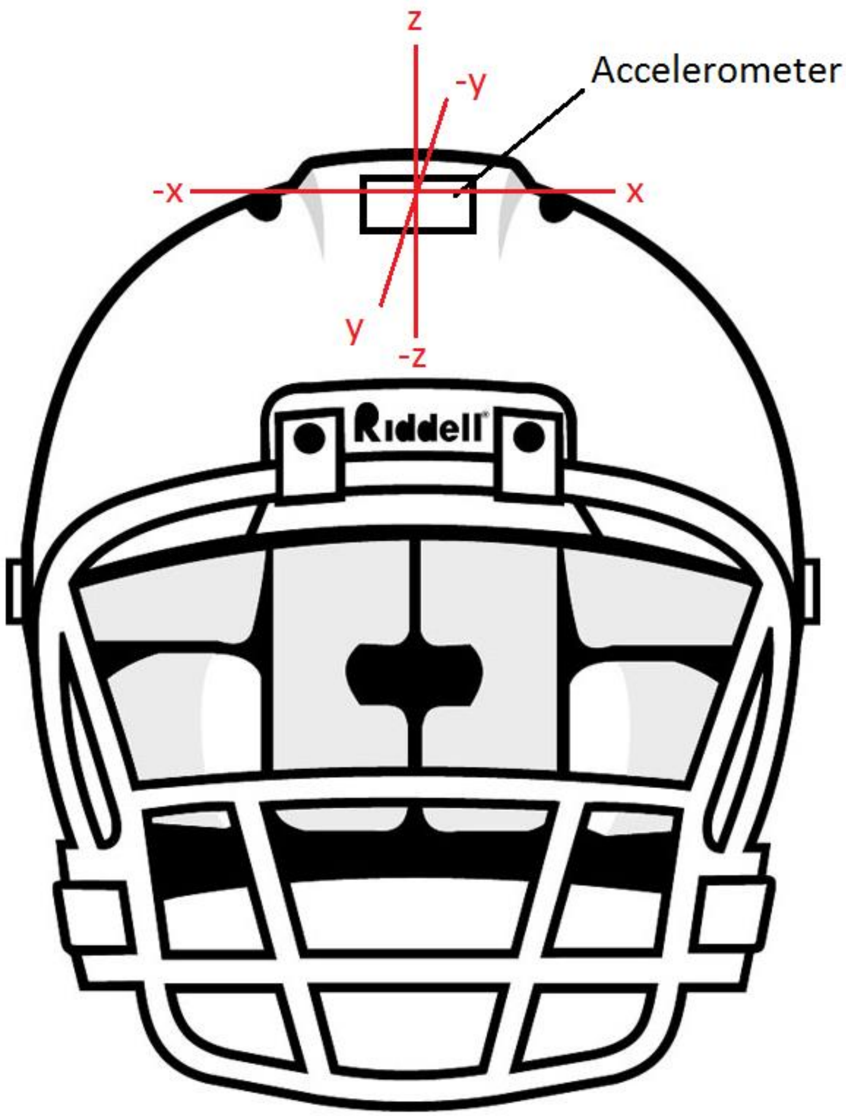


Figure 4-16: Accelerometer Helmet Mounting Location; A figure demonstrating the placement of the accelerometer sensor on the BIO-Helmet

4.2.2 Accelerometer Sensor

The ADXL377 3-Axis High G Analog MEMS Accelerometer developed by Analog Devices is the accelerometer used in the BIO-Helmet. It is capable of measurements up to ±200 g's which gives the project plenty of overhead to accurately measure impact force of helmet to helmet contact. The sensor can detect forces on all three axes, both in the positive and negative directions. This is in line with the design to mount the sensor in the top of the helmet as shown in Figure 4-16. It is also low powered and only consumes 300 micro amps at 3.3V. Shown below in Figure 4-17 is the block diagram for the ADXL377.

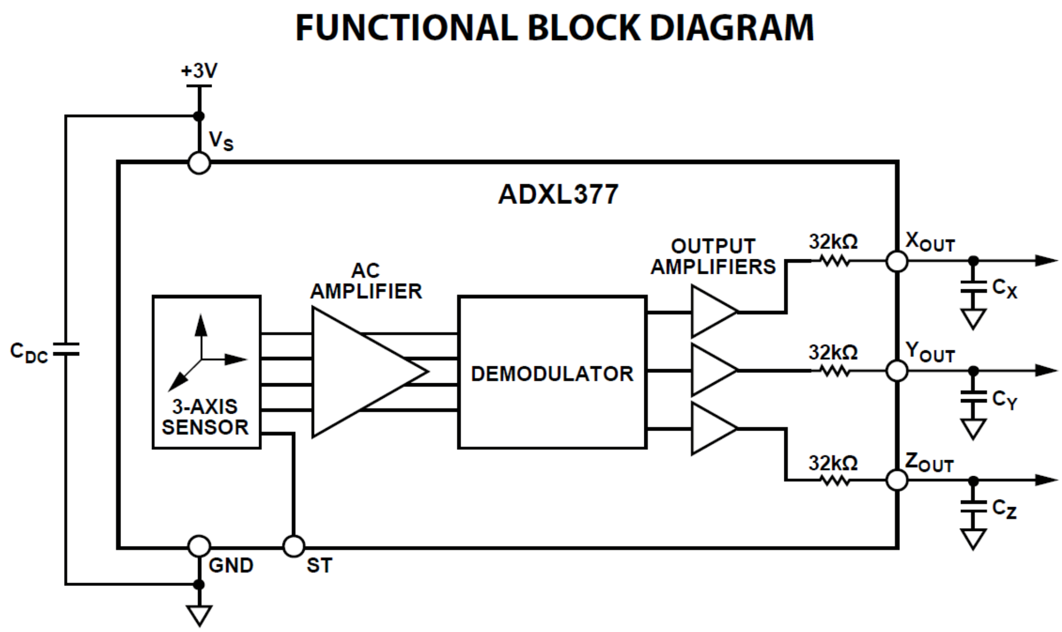


Figure 4-17: ADXL377 Block Diagram; A block diagram of the ADXL377 3-Axis High g Analog MEMS Accelerometer; reprinted with permission from Analog Devices, Inc.

Figure 4-17 shows a few important characteristics that affect the performance of the device when implemented in the BIO-Helmet. The capacitor C_{DC} is used to decouple the sensor from the power supply and was added to the PCB separately. As stated in the data sheet, a single 0.1 µF capacitor was enough to eliminate interference. The C_X, C_y, and C_z capacitors were also added to the design. A minimum of 1000 pF capacitor for each output is recommended in the data sheet to filter out noise and interference below 50 HZ. The BIO-helmet runs on a DC battery so the 1000 pF is enough for this filtering.

The 32 kΩ output resistance was also taken into consideration when attaching the outputs to the inputs of the analog to digital converter. On the Tiva C microprocessor, it is recommended to not exceed an input resistance of 10 kΩ on the analog to digital converter so some extra circuitry to reduce the output resistance of the ADLX377 is required. The easiest way to accomplish this was through the use of a low input offset rail to rail operational amplifier placed on the output of each axis. This allows the analog to digital converter in the microprocessor to properly read the data. A TLC2201 from Texas Instruments was used for this application.

4.2.3 ADXL377 Pin Configuration

The ADXL377 has a total of 16 pins and one exposed pad. There are 4 pins located on each side of the chip as shown in Figure 4-18 below. Figure 4-18 also details the orientation of the sensor relative to which axis will be activated during an impact. During construction, special attention was paid to the mounting position to minimize error in the resultant force vector.

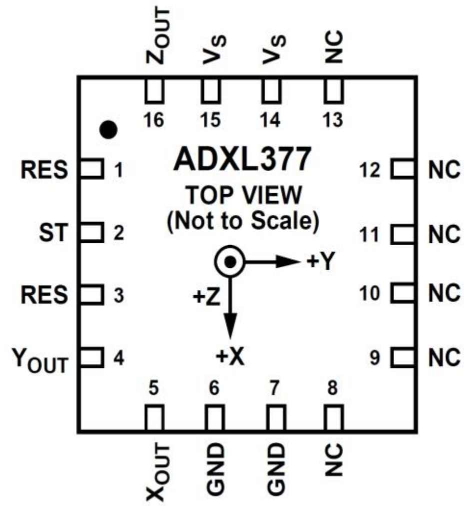


Figure 4-18: ADXL377 Pin Configuration; A figure showing a top down view of the ADXL377 pin configuration; reprinted with permission from Analog Devices, Inc.

Pin #	Pin Name	Description	Input/output
1	RES	Reserved, will be left open	NA
2	ST	Self-Test	Input
3	RES	Reserved, will be left open	NA
4	Yout	Y-axis output	Output
5	Xout	X-axis output	Output
6	GND	Must be connected to ground	Input
7	GND	Must be connected to ground	Input
8	NC	No connect	NA
9	NC	No connect	NA
10	NC	No connect	NA
11	NC	No connect	NA
12	NC	No connect	NA
13	NC	No connect	NA
14	Vs	Supply Voltage 3.3V	Input
15	Vs	Supply Voltage 3.3V	Input
16	Zout	Z-axis output	Output

Table 4-1: ADXL377 Pin Configuration; A table showing the ADXL377 pin configuration

For the BIO-Helmet, the main pins of interest are the Xout, Yout, and Zout. These pins output a certain voltage based on the force of an impact to the helmet which is then used to create an impact force vector. They each are connected to one of the analog to digital converter inputs on the MCU.

The Self-Test pin was helpful in determining if the accelerometer is working correctly during the build process. By applying Vs to the ST pin the typical voltage of Xout will change by -6.5 mV (about -1.08 g), Yout will change by +6.5 mV (about +1.08 g), and Zout will change by +11.5 mV (about +1.83 g). In the final design the

ST pin was left open as it is not needed once the accelerometer has been deemed functional.

The No Connect pins were not taken into account in the circuit schematic design but did come in to play when building the board. Due to the force of an impact on the helmet, it was beneficial to have the no connect joints soldered to the board for extra stability. There is also an exposed pad located on the bottom of the chip that was soldered to the board to aid in keeping everything connected during a forceful impact.

Although the supply voltage can operate in a range from 1.8 volts to 3.6 volts, the accelerometer circuit provides 3.3 Volts for VS. This decision is based on the operating logic level of the MCU which uses a 3.3 Volt signal. This allows for control over the power consumption of the ADXL377 should the power usage of the system ever become a problem.

4.2.4 ADXL377 Design Schematic

The design schematic for the ADXL377 is based on the evaluation board provided by Analog Devices, Inc. During the building phase of the project, adjustments were made to the capacitor values to compensate for some bandwidth and noise issues. Figure 4-19 below shows the initial schematic design for the ADXL377.

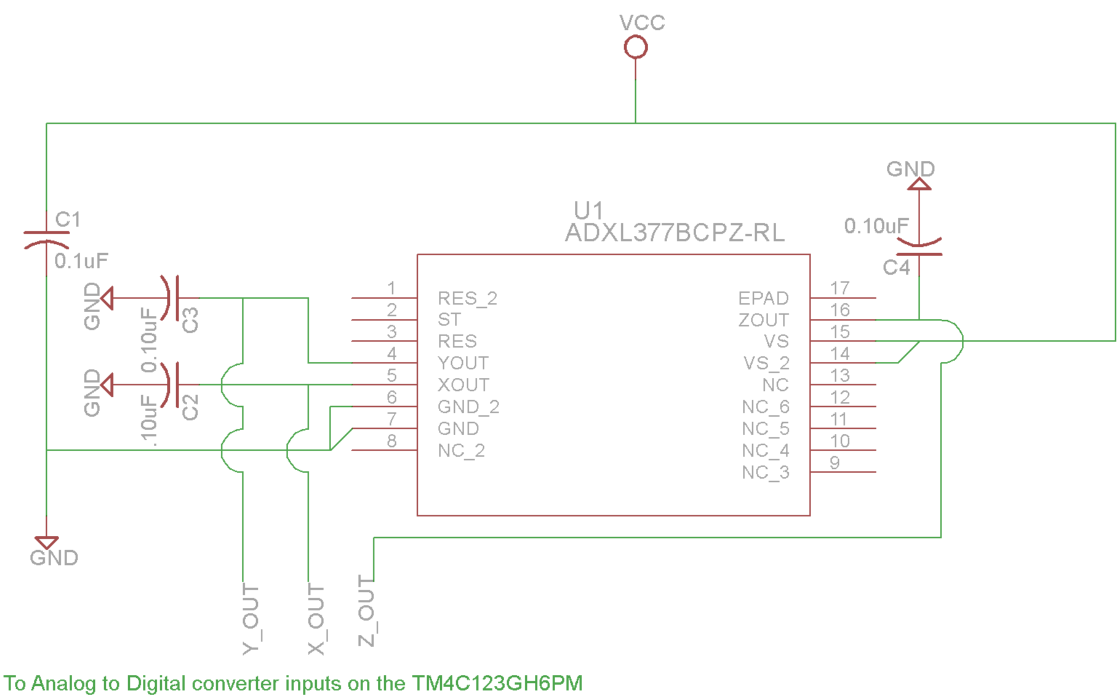


Figure 4-19: ADXL377 Schematic Design; An EagleCAD schematic design describing the ADXL377

Component	Value	Purpose
Vcc	3.3 V	Supply Voltage
C1	0.1 µF	Decoupling
C2	0.10µF	Bandwidth Adjustment
C3	0.10µF	Bandwidth Adjustment
C4	0.10µF	Bandwidth Adjustment

Table 4-2: ADXL377 Schematic Design; A table describing the components of the ADXL377 Design Schematic shown in Figure 4-19

The purpose of the C2, C3, and C4 capacitors are to adjust the bandwidth of the of the output signal for each axis. These capacitors implement a low pass filter that aids in antialiasing and noise reduction. By choosing a value of 0.10 µF a bandwidth of 50 Hz is achieved. This was a sufficient bandwidth for polling the sensor. If need be, the bandwidth can be adjusted later by using smaller capacitors at C2, C3, and C4.

The capacitor C1 is used to decouple the ADXL377 from the power source and prevent noise from the power supply from affecting the axis outputs. A 0.1 µF capacitor is recommended in the data sheet and was implemented in the design.

4.2.5 ADXL377 Output Sensitivity

For each axis, the output at 0g is equal to about ½ full-scale. With Vs set to 3.3 V this gives a 0g output voltage of 1.65 V. There can be inaccuracies based on temperature and the manufacturing process of the device, as seen in below.

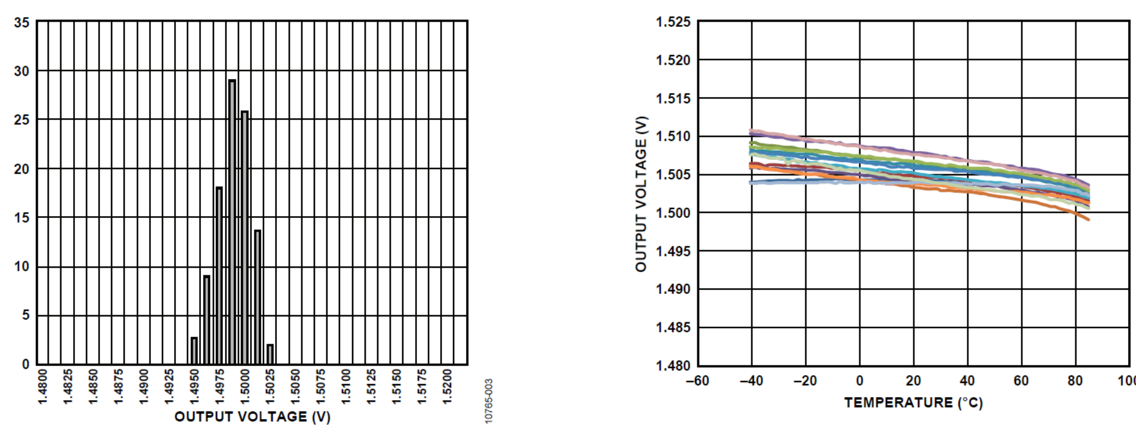


Figure 4-20: ADXL377 Output Inaccuracies; A figure demonstrating the output voltage vs tested population and temperature at 0g and Vs =3V; reprinted with permission from Analog Devices, Inc.

Although a perfectly accurate reading was not necessary for the BIO-Helmet, it is important to get as close as possible. Being that the sensor was mounted in the top of a helmet, with zero cooling or ventilation, high variations in temperature are bound to occur. If a game is taking place in cold weather, the temperature could change by as much as 40 degrees Celsius from the time it is sitting on the bench, to being put on an athlete’s warm cranium. To account for these differences, a calibration method was used in the software which kept the accuracy and readings consistent throughout use.

The ADXL377 is designed to change the output voltage at each axis by ±6.5 mV for every g sensed in the positive or negative direction. With a range of -200 g to +200 g the voltage output range goes from 0.35 V for -200 g to 2.95 V for +200 g.

4.3 TIVA C MICROCONTROLLER: TM4C123GH6PM

The Tiva C microcontroller, also known as the TM4C123GH6PM, has a main targeted clientele for industrial applications. The uses for this microcontroller are literally endless but the designed purposes are remote monitoring, fire and security, gaming equipment, transportation, HVAC and building control, motion control, point-of-sale machines, test and measurement, and lastly network appliances and switches. We however used it to send out data to an external computer as shown below in Figure 4-21. This allows us to monitor updates on player brainwaves and activities in as close to real time as possible. This also allows the medic or user to scroll back and see the brainwaves before and after the point of impact.

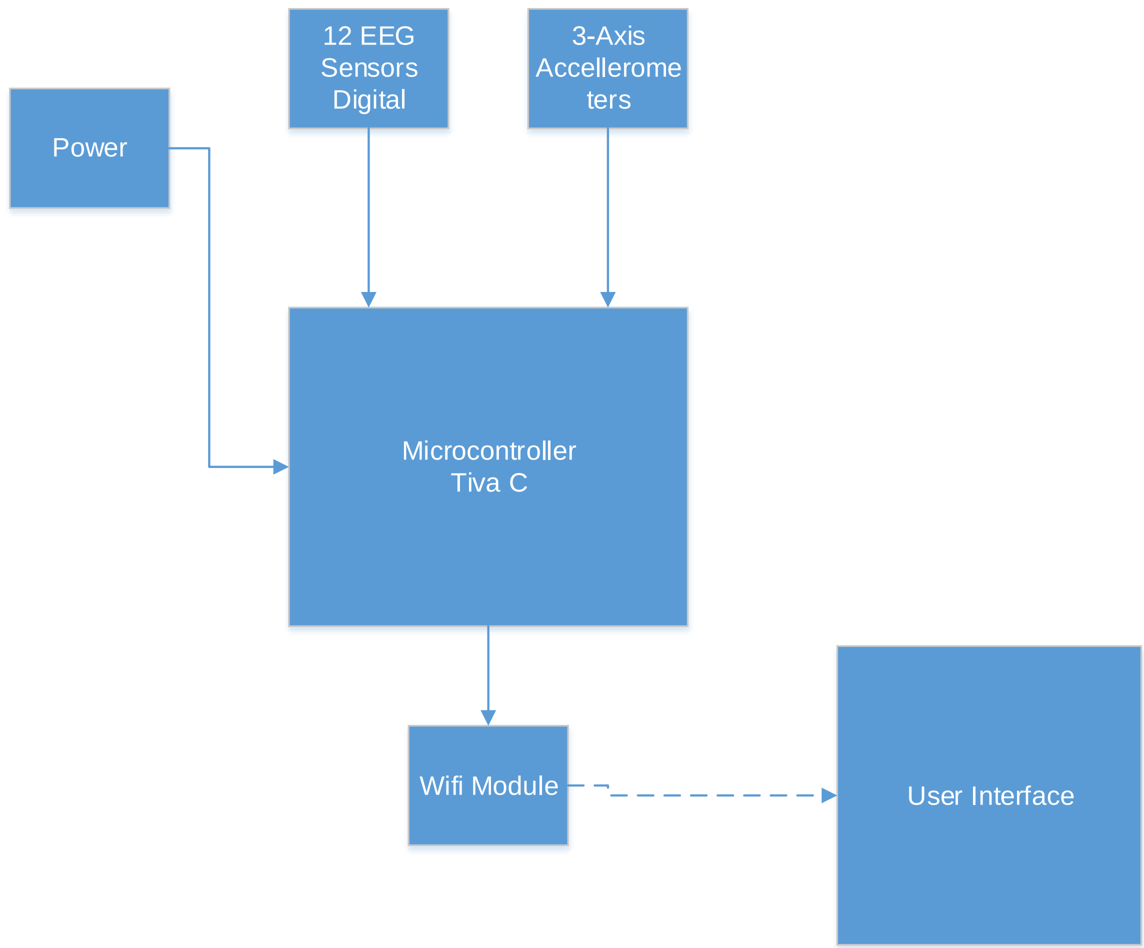


Figure 4-21: Design Layout; The block diagram of the full design layout with all sensors connected

The Tiva C has many purposes but the reason why this microcontroller was selected for this project is the fact that it was designed for remote monitoring and motion control. With these two subjects in mind, this project was designed accordingly. The basic needs of this project were to have a microcontroller that can monitor and transmit data in real time from the EEG sensors and accelerometers. The motion control design goals of the TM4C123GH6PM were especially useful in processing the accelerometer sensor data to calculate impact. As shown in Figure 4-22 below, the Tiva C has many features that appealed to this project's uses and needs.

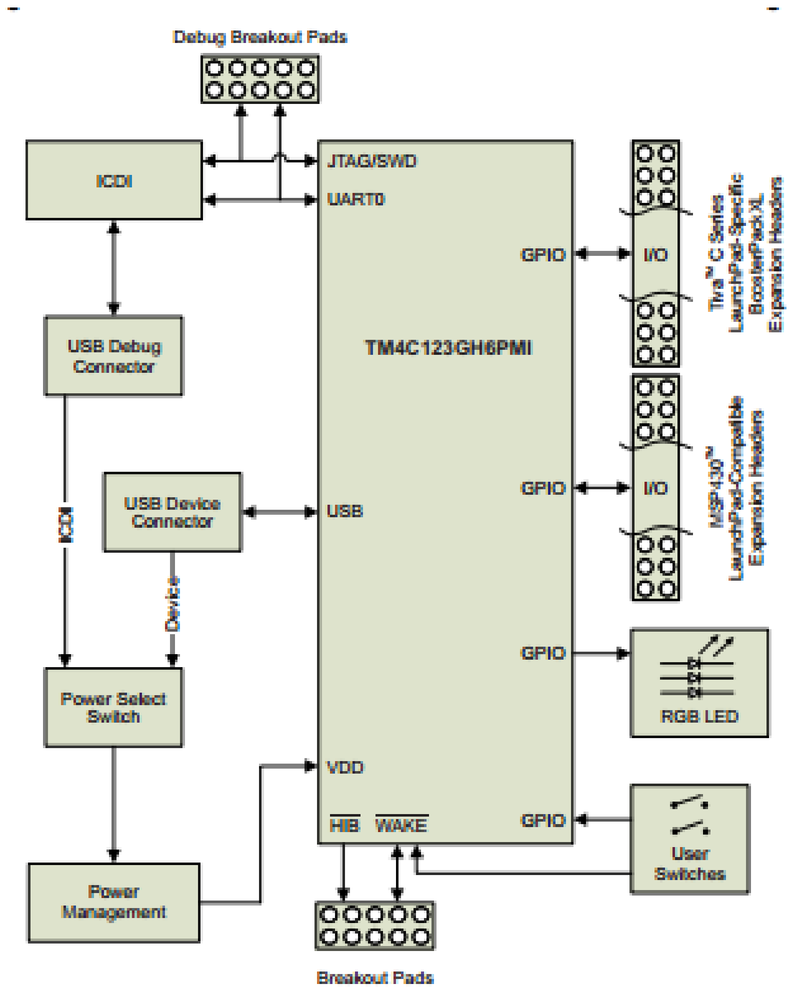


Figure 4-22: Tiva C TM4C123GH6PM; A figure demonstrating the overall layout of the TM4C123GH6PM microcontroller; reprinted with permission from Texas Instruments, Inc.

4.3.1 Important Features

A powerful chip that can collect and transmit huge amounts of data was required for this project. The Tiva C is a 32-bit ARM Cortex processor with an on chip memory featuring 256 KB of flash and up to 40 MHz performance benchmark. It has an ARM prime cell 32-channel controller that allows for very efficient use of the bus bandwidth and the processor. Also, the analog support for this microcontroller is extensive. It features twelve 12-bit analog to digital converters which will be very useful when reading the analog output from the accelerometers. The features of the Tiva C are outlined in Table 4-3 below.

Pin and Package	64LQFP
CPU	ARM Cortex-M4
Flash	256 KB
SRAM	32 KB
Max Speed	80 MHz
Motion PWM Outputs	16
QEI	2
GPIOs	43
Operating Temperature Range	-40 degrees C to 105 degrees C
OTG	Yes
SSI/SPI	4
I2C	4
UART	8
ADC Channels	12
ADC Resolution	12 Bits
CAN MAC	2
SysTick	Yes

Table 4-3: Tiva C TM4C123GH6PM Important Features; A table describing the important features of the TM4C123GH6PM

4.3.2 Packaging

The Tiva C microcontroller was released in 2014. It is a new model that is assembled in the Philippines with no antimony or boron as to keep the microcontroller's footprint on the environment as small as possible. It is RoHS and High-Temp compliant; it has been tested to have extremely low levels of cadmium and hexavalent chromium (<.01%) and very low levels of lead, polybrominated biphenyls, and polybrominated diphenyl ethers (<.1%). The packaging type is PM with 64 pins and the mass of the device is 319 mg.

4.4 CC3100 Wi-Fi Module

The CC3100 is a Wi-Fi module designed for low-power wireless transmissions with high levels of data transfers. It includes 802.11 b/g/n radio, baseband, and medium access control capabilities. The major modules within this chip are depicted below in Figure 4-23.

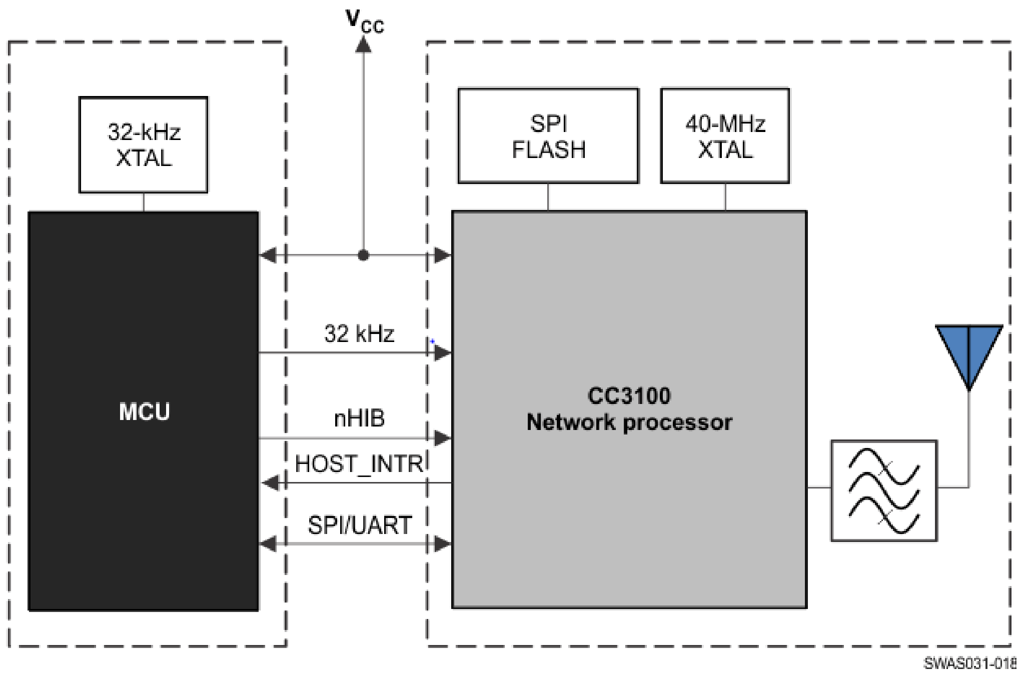


Figure 4-23: CC3100 Modules; A figure describing the major modules present within the CC3100 chip; reprinted with permission from Texas Instruments, Inc.

4.4.1 Important Features

Power consumption was a huge aspect of this project as it is not known how long the device has to be in use for before it can be recharged. The health of the wearing athlete is at stake if the BIO-Helmet were to fail and stop sending relevant data to the observing medical user, defeating the purpose of the device entirely. This module has a sleep mode that can drastically reduce the amount of power the module will use while the player is benched or currently not moving for a prolonged period of time. There is not a point in transmitting data from a person that is not active at the time unless it is for research reasons. For this case, the user can easily control the device via the Wi-Fi module so that it stays on for a prolonged period of time. However, for normal uses the ability to go to sleep mode was important for the implementation goals of this project. The various power consumption metrics of the CC3100 are included in Table 4-4 below.

Wide-Voltage Mode	2.1 to 3.6V
Pre-regulated Mode	1.85 V
Hibernate with RTC	4 nA
Low-Power Deep Sleep	115 nA
RX Traffic (MCU Active)	53mA
TX Traffic (MCU Active)	223mA
Idle Connected	690 nA

Table 4-4: CC3100 Power Consumption; A table demonstrating the various modes and power consumption of the CC3100 Wi-Fi module

4.4.2 Packaging

The CC3100 Wi-Fi module came to market in 2014. As it is a new model, Texas Instruments chose to manufacture it completely green compliant. This primarily means that power consumption is low and no lead, or other heavy metals, are used in this part. It is made of materials that will not make a negative impact on the environment when it is disposed of. The package dimensions are 9mm by 9 mm and the height is slightly above 1mm and offers 64 pins as shown in Figure 4-24 and Figure 4-25 below. It comes with contact pads on the bottom of the package and large ground/thermal plane in the middle.

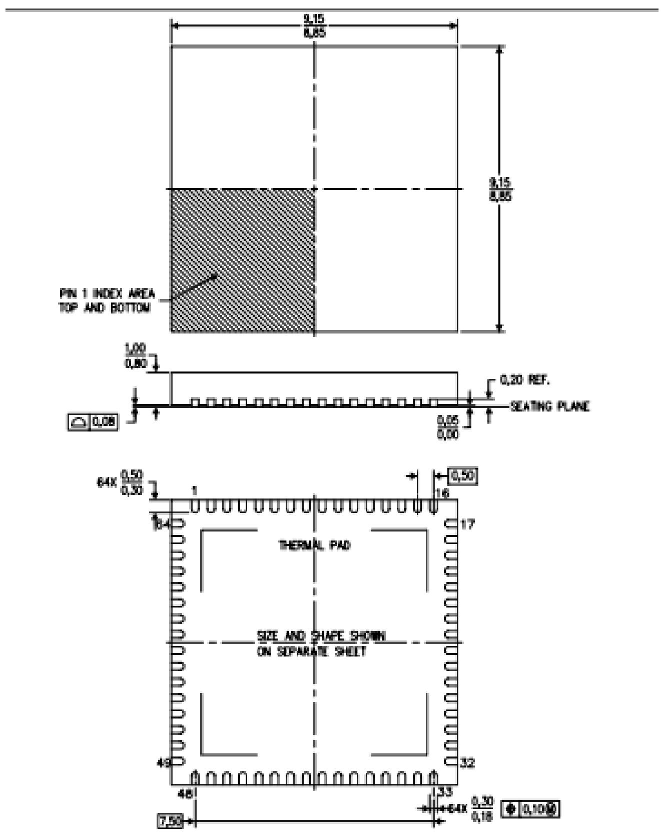


Figure 4-24: CC3100 Dimensions and Layout; A figure describing the size, dimensions, and layout of the CC3100; reprinted with permission from Texas Instruments, Inc.

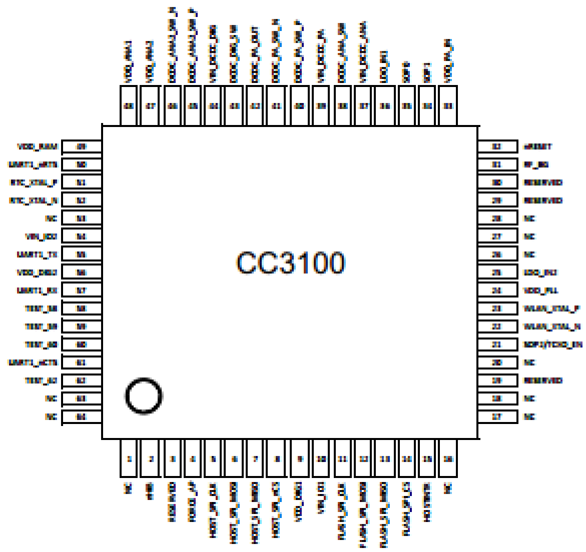


Figure 4-25: CC3100 Pin Layout; A figure describing the 64 pins present on the CC3100; reprinted with permission from Texas Instruments, Inc.

The pin layout of the CC3100 shown above in figure 4-23 allowed us to network our microprocessor to the server module quite easily. As they are both products from TI, and the CC3100 is compatible with multiple TI microprocessors, we did not experience any issues with the module. A complete reference design for the CC3100 can be seen in Figure 4-26.

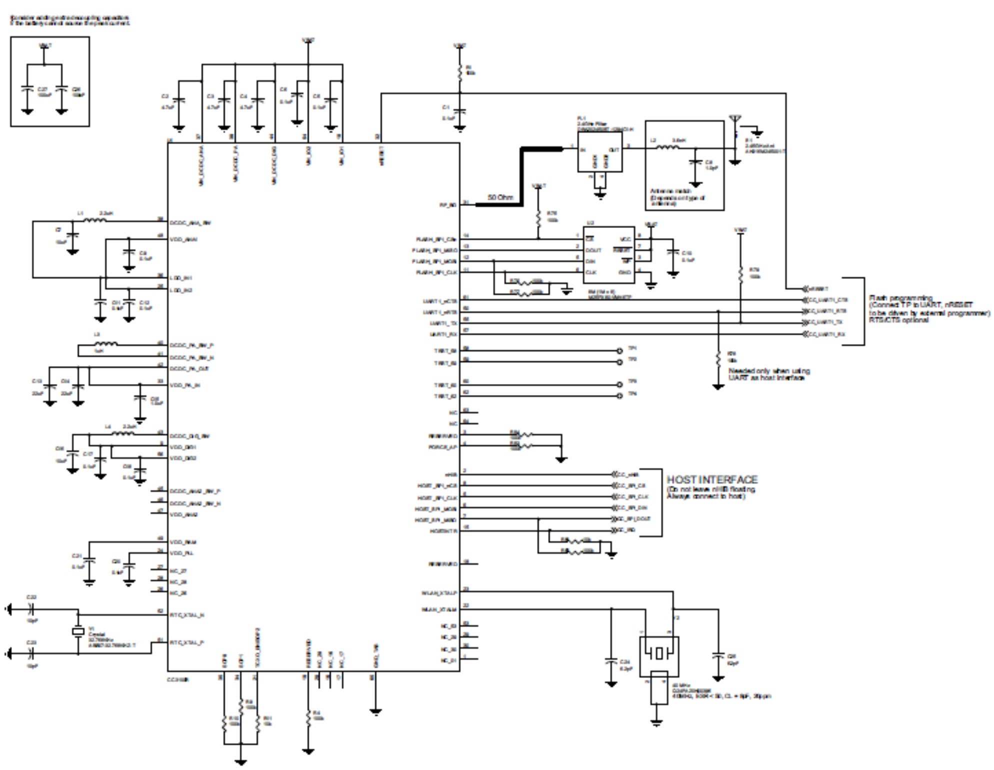


Figure 4-26: CC3100 Wide-Voltage Mode; A reference circuit schematic for the CC3100; reprinted with permission from Texas Instruments, Inc.

4.5 POWER SUPPLY

The following subsections discuss the power supply design for the BIO-Helmet. Specific parts, designs, and power schematics are included.

4.5.1 Battery Pack

Lithium ion batteries were implemented in the BIO-Helmet prototype. This decision was due to the fact that the battery can be manipulated to output enough voltage while still staying within the project's budgetary constraints. Lithium ion batteries can be flat and fit within a helmet. Lithium ion batteries are also rechargeable. This allowed these batteries to be used throughout testing without the need to continuously replace the batteries. This battery was placed near at the bottom back of the helmet to allow easier access to the charging ports. Placing the battery in this area was also critical for the least chance of direct impact, thus reducing the risk of athletes getting hurt due to the placement of the electronic components.

A PKCELL ICR18650 6600mAh 3.7V 1S3P lithium-ion battery pack was used to power the project. This pack is made up of three 2200mAh cells connected in parallel and spot welded to protection circuitry which prevents over-voltage, under-voltage, and over-current protection. This protection circuitry prevents the battery from over-heating and exploding which is of the utmost importance in keeping the helmet wearer safe.

The battery pack includes a JST two pin connector which connects to the board and charging circuit through a JST jack. This made battery replacement simple and adds the ability to charge the battery separately or have a backup in place if need be. The battery pack also has a fairly small form factor that fits well within a helmet. A size comparison can be seen in Figure 4-27 below.

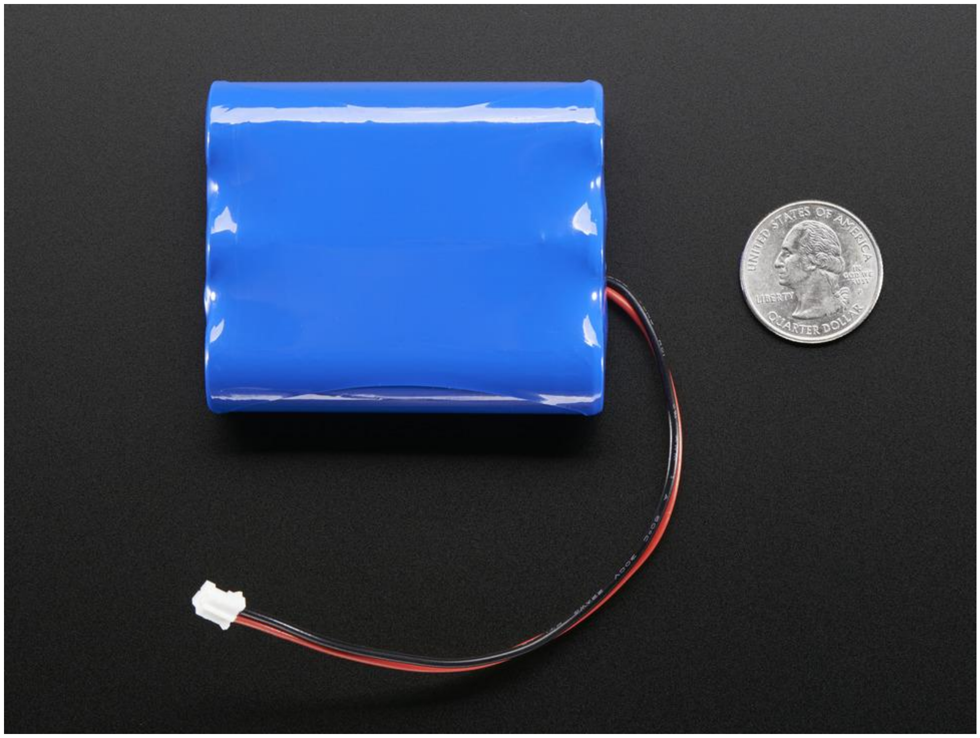


Figure 4-27: ICR18650 Size Comparison; An image demonstrating the size comparison of the ICR18650 to a quarter

More specifically, the battery measures in at 69mm x 54mm x 18mm and has a weight of 155 grams. Compared to the weight of the helmet itself this does not affect player performance or lead to any discomfort while wearing the BIO-Helmet.

4.5.2 Battery Pack Characteristics

The PKCELL ICR18650 was chosen for its high milliamps per hour rating which provides the BIO-Helmet enough power to last through a whole two and a half hour football game; one of the project requirements. A detailed table of the battery pack specifications can be seen in below.

Item	Characteristic
Nominal Capacity	6600 mAh
Nominal Voltage	3.7 Volts
Charging cut-off voltage	4.7 Volts
Discharging cut-off voltage	3.0 Volts
Max Charging Rate	3 Amps
Max Discharge Rate	6 Amps
Operating Temperature: Charge	0°C - 45°C
Operating Temperature: Discharge	-20°C - 70°C

Table 4-5: ICR18650 Characteristics; A table describing the various characteristics of the ICR18650 battery pack

The operating temperature is an important characteristic that was taken into consideration for this project. With the battery being mounted inside the helmet, the temperature could become very high. During a charging cycle, it is necessary to remove the helmet due to the possibility of the temperature of the battery reaching 45 degrees Celsius and exceeding the operating temperature. With an operating temperature of 70 degrees Celsius during the discharging period, which is the active period for the helmet, it is extremely unlikely for the battery to exceed this operating range.

Current draw was another important aspect to take into consideration. When charging, the max rate is 3 Amps, which equates to about a four charge time. The charging circuit was designed around this to ensure it does not exceed the max rating. With a standard USB device charger, the output is typically only one to two volts, which increases the charge time but is a nice universal form factor for the BIO-Helmet to use. Based on the chosen microprocessor, the max discharge rate stays well below the recommended three amps. To prevent any melted wire, it was required to keep the discharge and charge rate under two amps since the included wiring on the battery pack is used; which is only rated for two amps max.

4.5.3 USB Charging Design

Considering USB is widely accepted and used form factor, with a multitude of cheap and easily obtainable chargers, this was the specification used for the BIO-Helmet. A micro USB port was used to connect the charging circuit to a standard USB wall charger and micro USB cable. The small port size of the micro USB allowed for easy mounting of the port inside the helmet for hassle free charging. To ensure that the lithium ion battery pack is being properly charged, a specially designed IC manufactured for this purpose was used. The MCP73833/4 by Microchip fits this need, it is also very cheap at about one dollar per chip and is readily available from various manufacturers. Figure 4-28 shows the pin layout and package description for the MCP73833/4, followed by Table 4-6 which shows the pin number and description for each of the 10 pins.

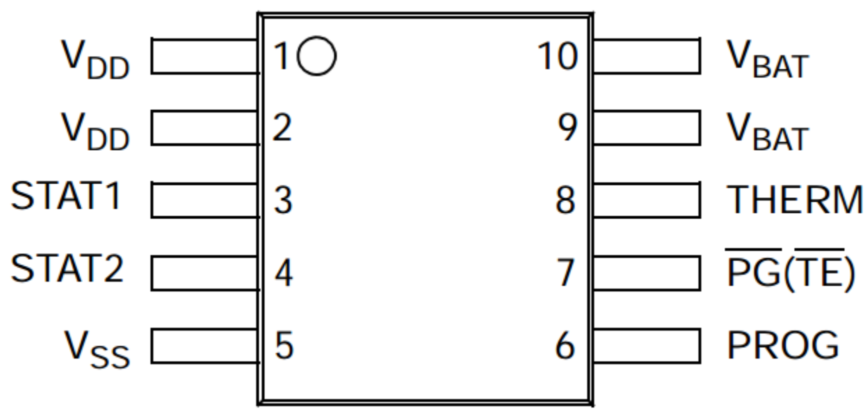


Figure 4-28: MCP73833/4 Pin Layout; A figure demonstrating the pin layout and package description for the MCP73833/4; reprinted with permission from Microchip Technology, Inc.

Pin #	Pin Name	Description	Input / Output
1	V_{DD}	Battery Management Supply	Input
2	V_{DD}	Battery Management Supply	Input
3	STAT1	Charge Status	Output
4	STAT2	Charge Status	Output
5	V_{SS}	Battery Management 0V reference	Input
6	PROG	Current Regulation Set	Input
7	PG	Power Good	Output
8	THERM	Thermistor	Input
9	V_{BAT}	Battery Charge control	Output
10	V_{BAT}	Battery charge control	Output

Table 4-6: MCP73833/4 Pin Description; A table showing the pin number and description for each of the ten pins on the MCP73833/4

The MCP73833/4 is designed specifically for use with USB charging applications and can handle a max charge current of 1 amp which allows the battery pack to recharge in a couple hours. It also has built in status updating so that connected LEDs show the status of the charge to the user. Pin 3 lights up an LED when the battery is in charging mode. Pin 4 lights an LED when the charging is complete. Pin 7 is used to light an LED to show that there is power plugged in via the USB port. Respectively, the color of the LEDs are orange, green, and red which easily indicate to the user the status of the battery and charge. Pin number 8 provides support for a thermistor to measure and adjust for temperature changes. For the purposes of this project, it was not necessary to implement the thermistor because the charge rates do not need to be very high, therefore it was simply replaced with a 10kΩ resistor in the final design.

Figure 4-29 shows the schematic for the charging circuit. It is based on the Adafruit Lithium Ion/ Lithium Polymer battery charger breakout board.

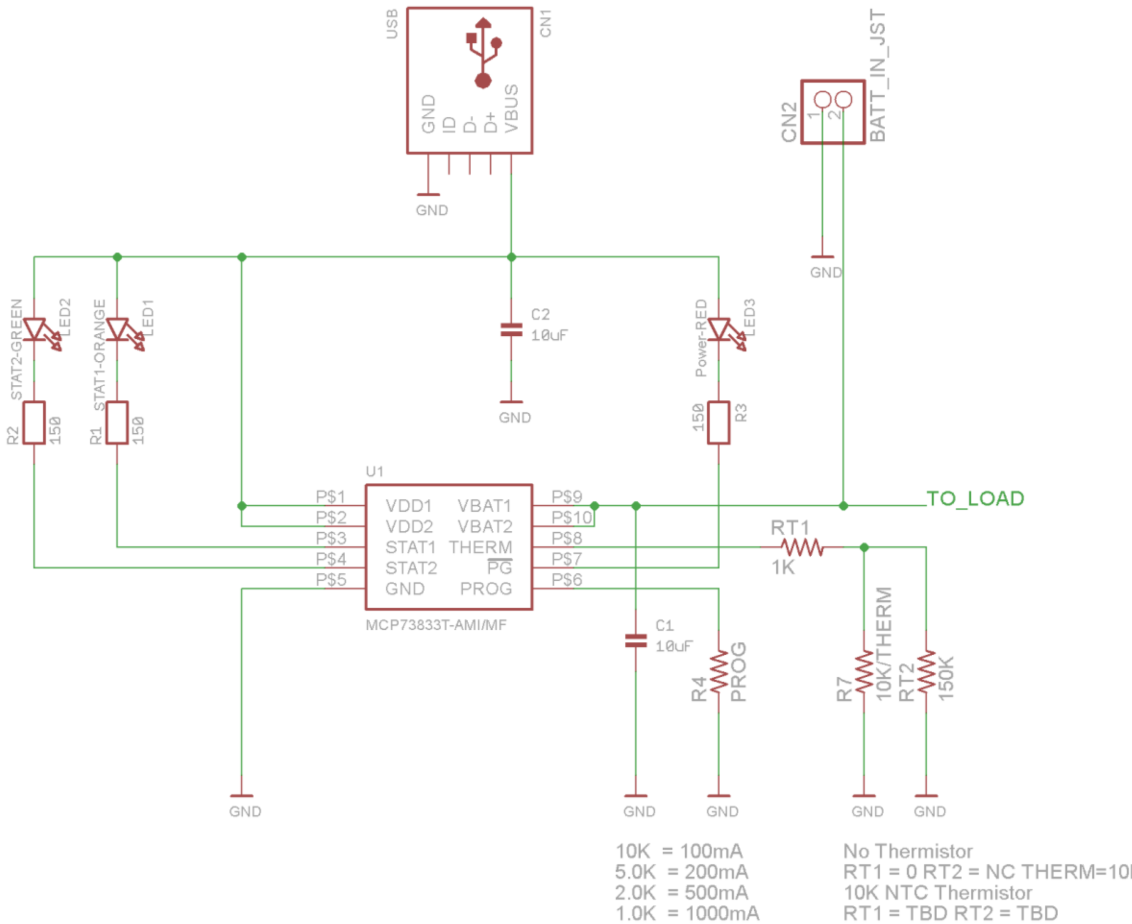


Figure 4-29: MCP73833/4 Charging Circuit; A schematic diagram detailing the battery charging using the MCP73833/4

The layout described above in Figure 4-29 was implemented on the main board. The JST connector was also mounted directly to the board so that the battery could be mounted fairly close to the rest of the electronics to prevent unneeded resistance from long wires

This board was built with the idea that the helmet would not be in use while it is charging. The location of the board and the battery will cause excess heat if the battery were to be charging while the helmet is in operation.

The TO_LOAD line shown in the schematic makes its way to the microprocessor through a voltage regulator to ensure a constant voltage as the battery drains. The EEG sensors and accelerometer also make use of the power output of the battery on the load line.

4.6 FULL BIO-HELMET HARDWARE SCHEMATIC

The schematic diagram describing the complete design of the BIO-Helmet system is included below in Figure 4-30. A larger, landscape orientation, of this schematic is included in Appendix C. Figure 4-31 shows the board layout of the final PCB design.

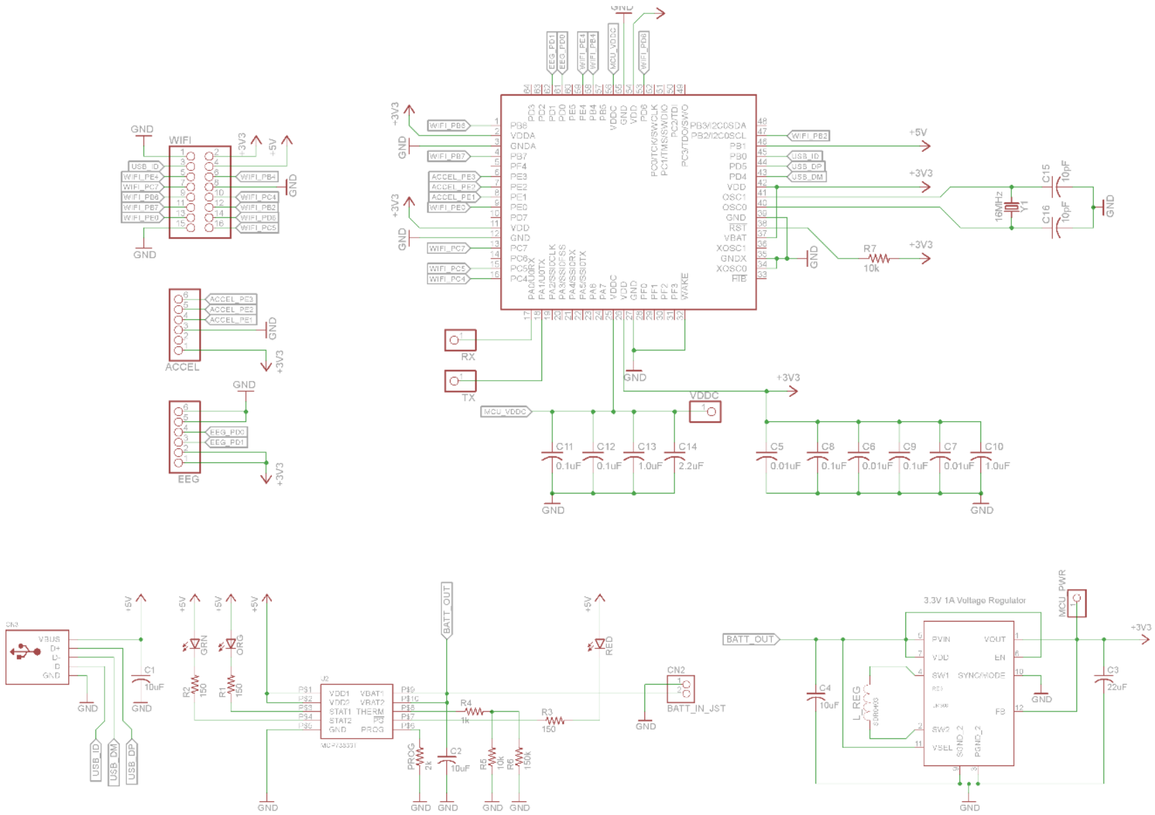


Figure 4-30: Full BIO-Helmet Hardware Schematic; A schematic diagram representing the full implementation of the BIO-Helmet hardware design

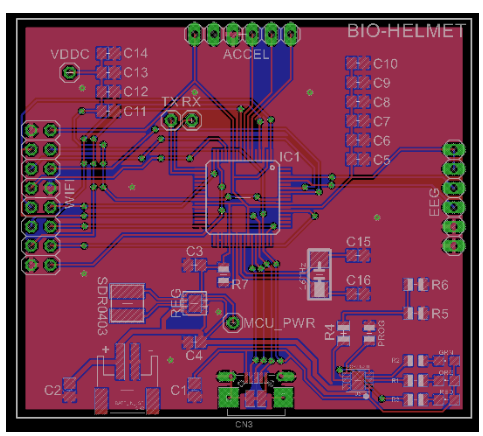


Figure 4-31: Full BIO-Helmet Hardware Board Layout; A board schematic diagram representing the full implementation of the BIO-Helmet hardware design

As shown above in Figure 4-28 we were able to completely wire out entire board with all sensors into just one microprocessor. Initially we thought that we might need multiple microprocessors in order to free up the sufficient number of pins required to have all the sensors we wanted running all at the same time while transmitting all the data in real time. However, doing this would have required two microprocessors just taking in the data and a third taking in all the data from both microprocessors and sending it out to the external Wi-Fi connection. When we thought of this plan we did not think it to be efficient as our goals for this project were to utilize as little power as possible as to save energy and stop players from having to recharge their helmets mid game. We expanded the pins by connecting similar EEG channel sensors to the same pin allowing for some pins to be utilized by multiple sensors. This enabled us to keep everything to a minimum by allowing all the processing and data transportation to be done on a single microprocessor.

4.7 Embedded Software

This section covers the detailed design aspects of the software running on the Tiva C ARM Cortex microcontroller present within the BIO-Helmet. This piece of software is responsible for polling the accelerometer and EEG sensors, processing the received raw input from the sensors, and outputting this data to the Wi-Fi part for sending to the local server. The logical data flow for the Tiva C microprocessor can be observed in Figure 4-32. This software also covers the basic operation of the Tiva C, including watchdog timer, UART, and Wi-Fi component initializations. This program consists of a single Energia sketch running on the ARM Cortex CPU.

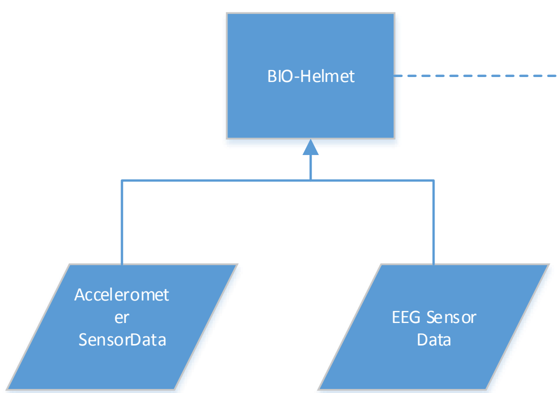


Figure 4-32: BIO- Helmet Data Flow; A diagram describing the flow of data between hardware and software modules

4.7.1 Initializations

The initialization phase of the embedded program sets up various aspects of the ARM Cortex microprocessor, input and output pins, UART, and wireless setup. This phase of the embedded program is always run first after the BIO-Helmet is powered on. The initialization order of the following elements in the actual embedded program will be performed in the order in which they are listed in this document. An overall flow and state diagram is included below in Figure 4-33 that describes the order in which these initializations are performed.

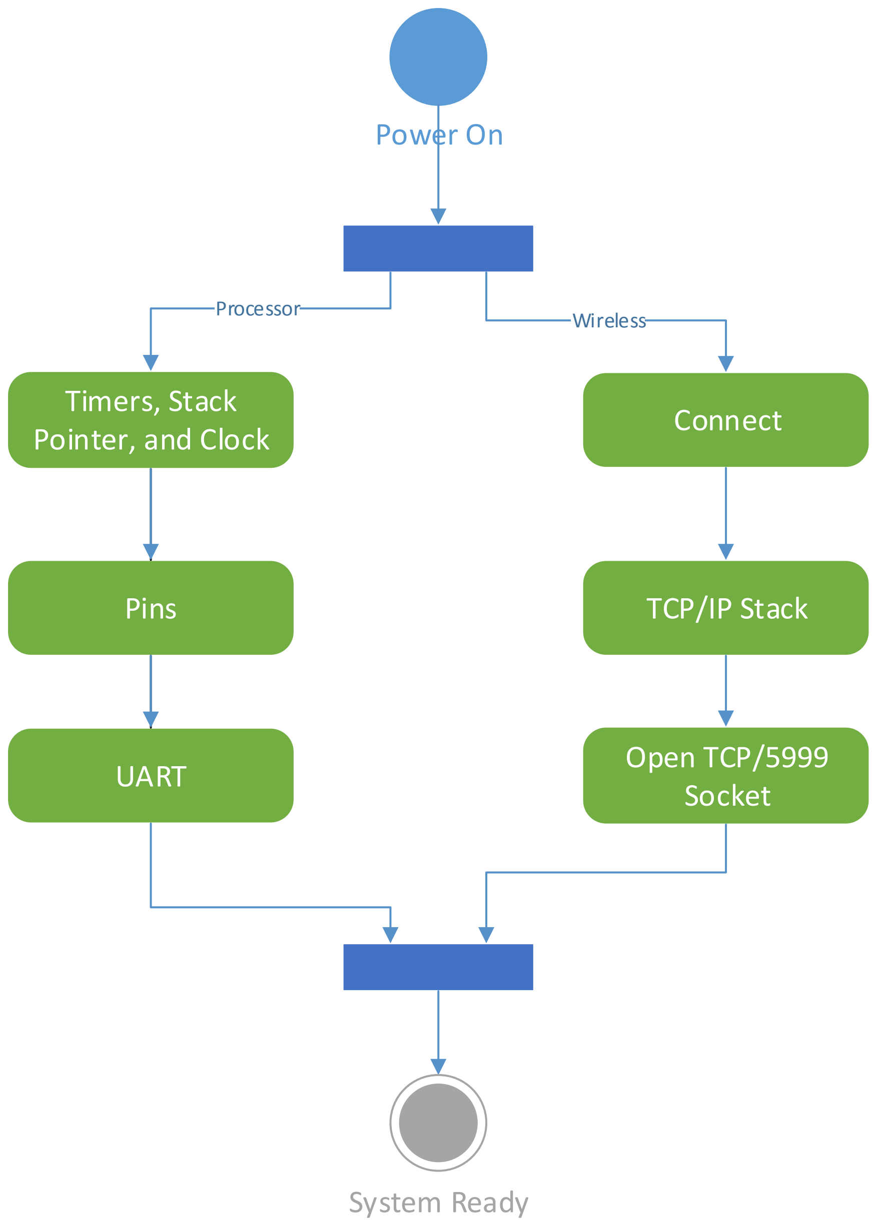


Figure 4-33: BIO-Helmet Initializations; A diagram which describes the order of the initializations when the BIO-Helmet is powered on

4.7.1.1 Processor and Memory

This portion of the initialization phase configures the timers so that the processor does not get reset due to inactivity. The stack is useful for various parts of the embedded software coding and algorithm design. The stack is especially useful for storing incremental input readings from the sensors before passing this data onto the data processing phase.

Lastly, the processor clock and timing configurations are made. The system is set to directly run of the crystal present on the printed circuit board. This configuration allows for the fastest and most accurate clock measurements. Timing is an incredibly important portion of this project due to the fact that the sensor input pins are e read from in set intervals to avoid any loss of data. If the timing and frequency settings are not correct, the BIO-Helmet processor could miss a high impact event (or any other type of data reading) causing additional risk for the athlete if they remain on the playing field.

4.7.1.2 Input

The next initialization to occur on the embedded program is the initialization of the input pins of the microprocessor. Input pins are set for each of the three axes (x, y, and z) of the accelerometer outputs. Two additional pins will be set to input on various EEG electrode sensor channels placed throughout the BIO-Helmet. All pins on a Tiva C ARM Cortex microprocessor are automatically set to an input so there is no need to statically set these pins to an input.

4.7.1.3 UART

The next initialization to occur enables two UART communications on the microprocessor. The first is for the USB debug port for code uploading and debugging purposes. This port will not be used in production and may be disabled in the future. A second UART is responsible for communications between the microprocessor and the Wi-Fi module. The CC3100 Wi-Fi part includes a UART input that is used for data transfer and communications between the wireless interface and the microprocessor.

4.7.1.4 Wireless

The last initialization to occur is the wireless connection and setup. This initialization occurs on the MCU contained on the CC3100 itself and does not involve the Tiva C ARM Cortex microprocessor. This configuration occurs by a preprogrammed set of configurations loaded onto the EEPROM of the CC3100. The chip is first initialized to build a wireless direct connection from the CC3100 to the local server over 802.11 wireless. An IP address is previously statically assigned to the BIO-Helmet and to the local server. The wireless module then opens a TCP socket between the BIO-Helmet and the local server. This TCP socket is used to transfer data between the BIO-Helmet and the local server. This TCP socket is opened over port 5999 and is closed by the local server after a timeout value (assuming that the BIO-Helmet has lost power or manually turned off) or closed when the monitoring user exits the BIO-Helmet reporting program on the local server.

4.7.2 Sensor Polling

Each of the pins connected to a sensor array, both accelerometer and EEG, will be sampled twenty times every second. The results from this sample are stored in floating point variables for later processing.

Each accelerometer sensor reading includes the velocity in each direction (x, y, and z), thus requiring the embedded program to poll three different pins which are connected to each of the three outputs for the accelerometer. Each of these three measurements are stored in a pointing float double precision variable for later processing.

The EEG sensor array has pin inputs from several different electrode pairs placed around the inner cap of the BIO-Helmet. Each of these electrode channels is connected to an input pin on the BIO-Helmet printed circuit board. The input from these electrodes is stored in a floating point number with double precision. This value is then be extrapolated and stored as the necessary brain waves needed to identify certain biological conditions of the athletes.

4.7.3 Data Processing

This phase of the embedded program will normalize and adjust each of the received inputs from the accelerometer and EEG sensors. The inputs to this phase are each of the three accelerometer axis measurements. These are all represented as floating point numbers. This portion of the accelerometer data processing also converts each of the x, y, and z sampling points to an overall g-force value observed at each second sampling point. This value is then compared against the threshold for an athlete sustaining mental injury. If this value exceeds the threshold, then a Boolean value is set in the embedded program which will be referenced in the next phase of the embedded program.

The EEG sensor data is also sampled at the same interval and normalized in a similar pattern. The input to this phase is the EEG sensor data. There are no high impact levels associated with the brain wave data thus the only output from this phase of the embedded program are the values of each of the EEG sensors. At this stage of the data processing program, there are several pieces of data ready for transfer to the Wi-Fi module for packing and sending. These include: the high impact Boolean flag (true or false) the x, y, and z values of the accelerometer data inputs, and the brain wave data.

4.7.4 Packaging and Sending

This portion of the embedded program is responsible for packaging the data points read from the input pins and sending this data to the local server. Part of this program runs on the Tiva C ARM Cortex microcontroller, other portions run on the MCU contained in the wireless module, described within Section 4.4.

4.7.4.1 UART Communication

The BIO-Helmet processor writes the sets of sensor data to the UART configured in Section 4.7.1.3. The data is written as one contained output line, starting with the helmet ID, Boolean high impact flag, next the three accelerometer output values, and lastly the observed brain wave activity. This UART passes the data to the input pins on the CC3100 Wi-Fi module. The data packaging and wireless communication is covered next, in Section 4.7.4.2.

4.7.4.2 Wireless Communication

The CC3100 wireless module makes use of the TCP socket opened in Section 4.7.1.4. This TCP socket ensures data reliability when sent between the BIO-Helmet and the local server. The CC3100 receives the input as a string over the UART connection between the processor and the MCU on the CC3100. This input string is packaged into a TCP packet and written to the TCP 5999 socket. The socket API included on the CC3100 packages the string into an IP packet and sends it over the wireless direct connection built in Section 4.7.1.4. This sent data is then received by the wireless chip present in the local server and passed to the Python data receiving script described in Section 4.8.1.1.

4.8 Local Server Software

This section covers the detailed design aspects of the backend software used for the BIO-Helmet server. This set of software collects the wireless data received from the BIO-Helmet, processes, and prepares that data; and insert this into a database for historical records of all sensor data from the BIO-Helmet. This software is also responsible for preparing a set of files that can be used in a customized MALAB environment to view the sensor data in graphical form. The logical flow of data can be observed in Figure 4-34.

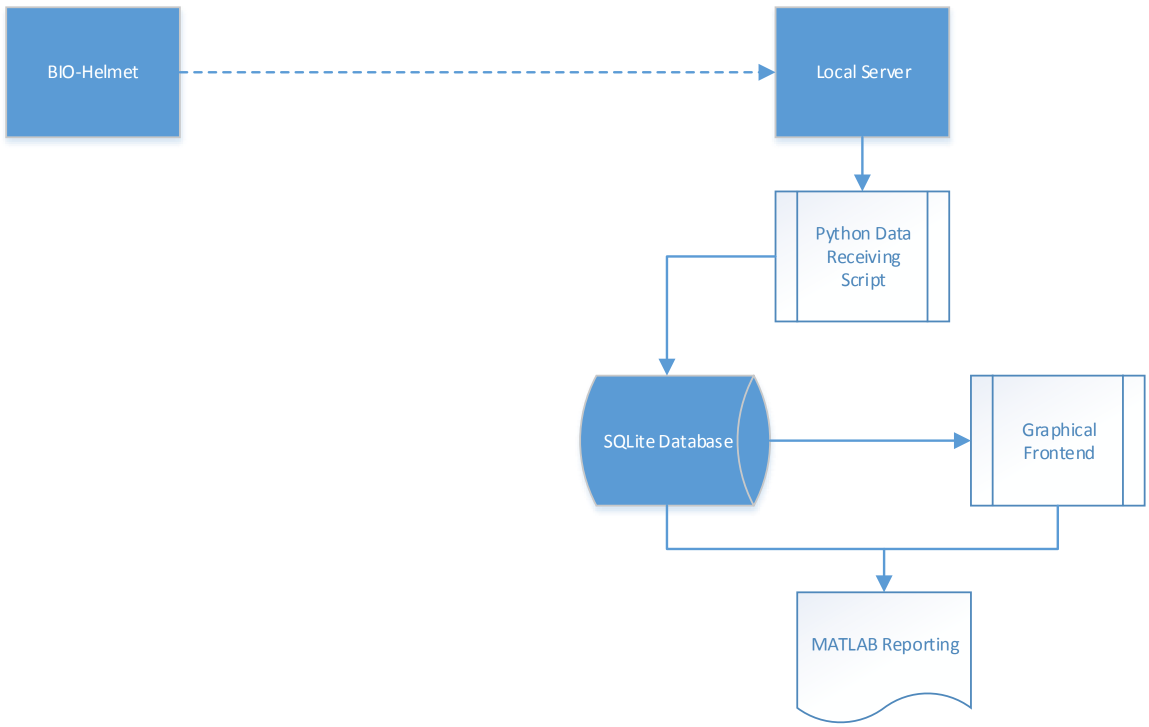


Figure 4-34: BIO-Helmet Local Server Data Flow; A diagram describing the flow of data between hardware and software modules

Data is received by the local server from the BIO-Helmet over a Wi-Fi direct connection. The Python data receiving script, accepts and prepares this data and then enters it into a SQLite database. The user is then able to interact with a graphical frontend to display the raw sensor data directly from the database or launch a customized MATLAB environment in which the sensor data can be viewed in graphical form. The graphical user frontend will also alert the user if a high impact event occurs.

4.8.1 Python Data Receiving Script

This Python based receiving script runs on the local server and is responsible for taking in the sensor data sent from the BIO-Helmet over a Wi-Fi direct connection to the local server. This data is then processed and then inserted into a SQLite database for historical record keeping. This database is then queried by the reporting software suite for display. Figure 4-35 below includes a class diagram for the Python data receiving script.

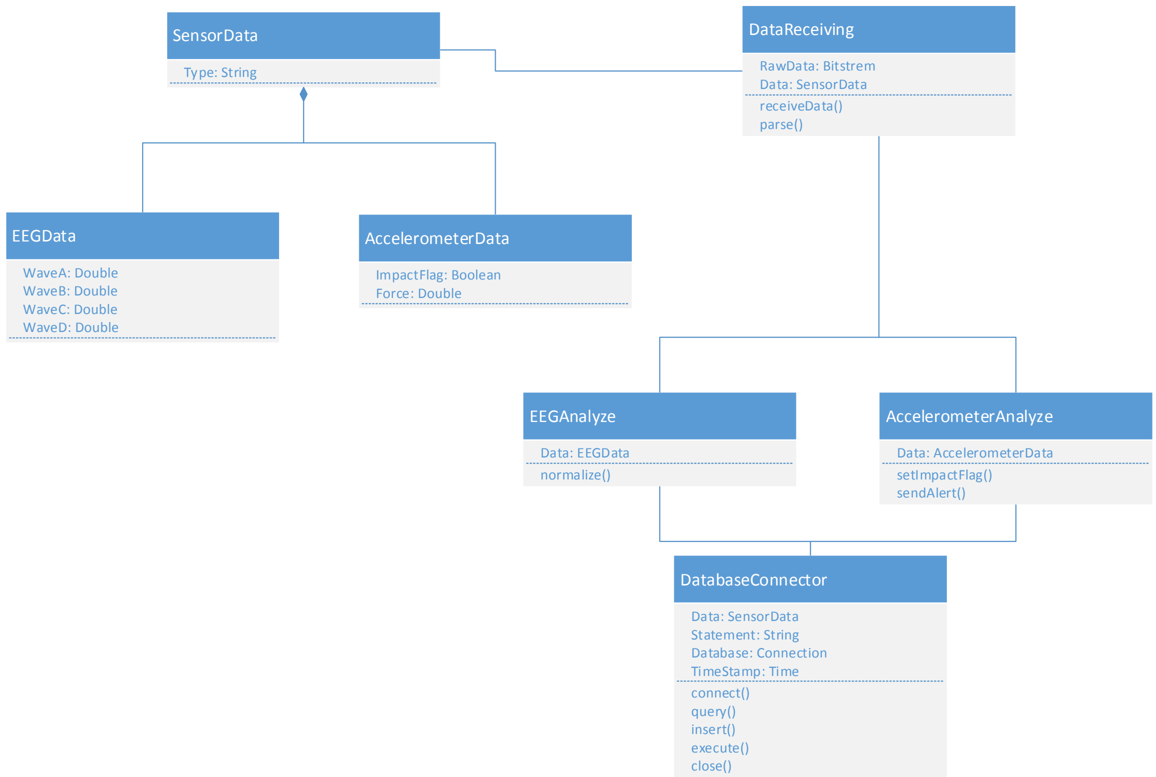


Figure 4-35: Data Receiving Class Diagram; A UML class diagram representing the BIO-Helmet data receiving Python script

4.8.1.1 Receiving Data

This Python program opens a socket on the local server over TCP port 5999. This arbitrary high port has been chosen for use in this project so as not to conflict with any other lower reserved ports for any other services which may be running on the local server. Python relies on the operating system level of TCP socket implementations, described in Section 3.9.5.2. These operating systems level implements are made available for use in Figure 4-36.

```
import socket
import os
```

Figure 4-36: Python Socket Import Statements; A code segment describing the use of OS level sockets in the Python data receiving script

The Python program first opens the necessary TCP 5999 port on the localhost (by using the ‘’ operator). The Python script then listens continually on this socket for a connection from the BIO-Helmet. Once, a connection is received from the BIO-Helmet, the Python data receiving script prints the IP address of the BIO-Helmet and then enters an infinite to loop to listen for string inputs sent from the Tiva C microprocessor on the BIO-Helmet. These inputs are processed accordingly, the accelerometer data, high impact warning message, and EEG sensor data. If the socket reaches a set timeout value without any activity from the BIO-Helmet, assuming that the BIO-Helmet has been turned off, then the Python script will close this socket. The listening socket will be kept open to listen for additional connections from the BIO-Helmet once it has been turned back on. An activity diagram for the data receiving portion of the Python data receiving script is included below in Figure 4-37.

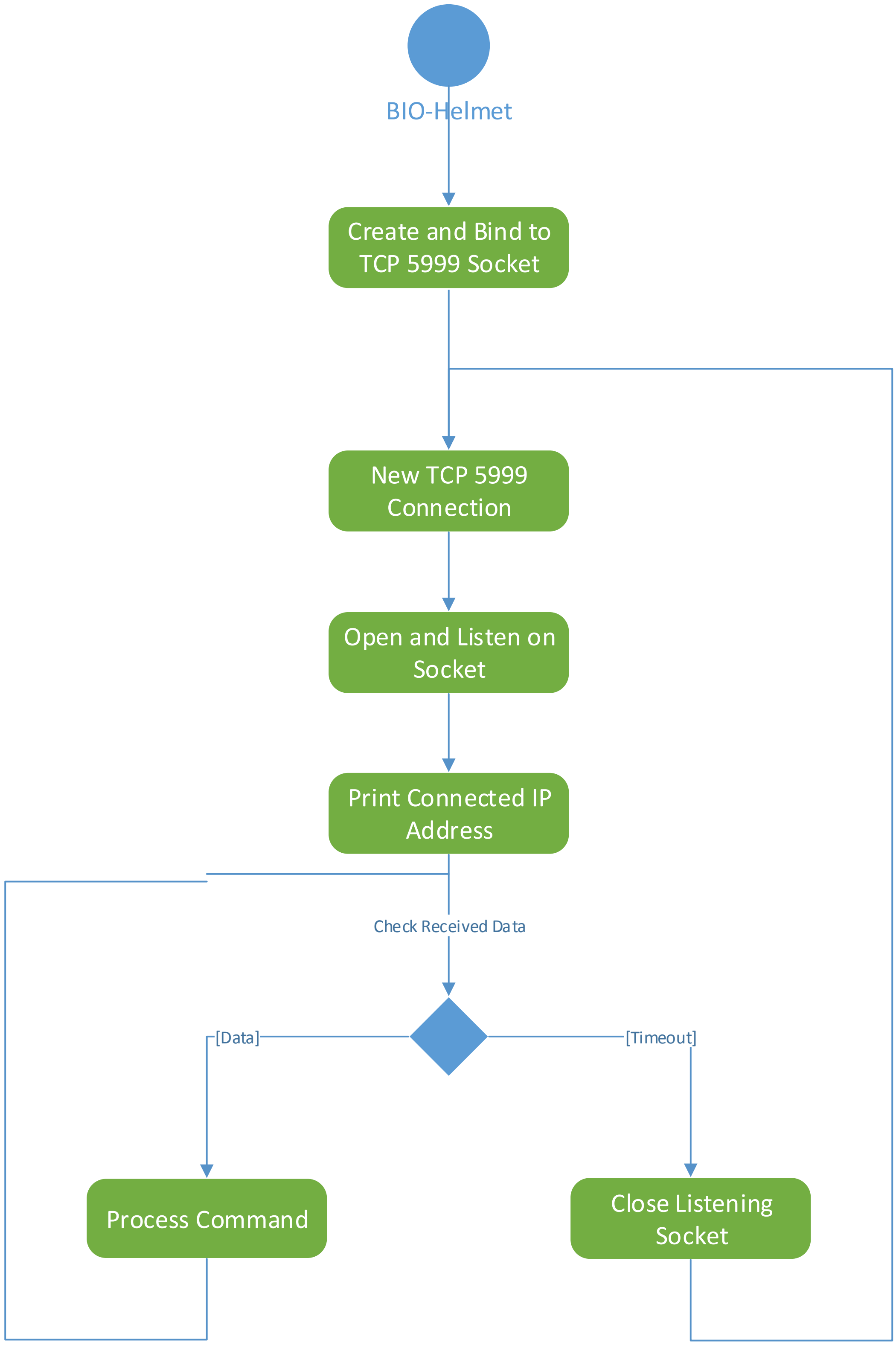


Figure 4-37: Receiving Data Activity Diagram; A UML activity diagram describing the data receiving module of the Python data receiving script

4.8.1.2 Data Preparation

Once the data is received from the data collection portion of the script, the data is processed and analyzed by this program. The brain wave EEG sensor data is processed into each of the necessary types of brain waves. The EEG sensor data is not analyzed for any threshold or anomalies as it must be analyzed by a neurologist. This data is saved until the program is ready to perform the database entries. The impact flag is analyzed to determine if a high impact occurs. If the Tiva C set the impact flag (see sections 4.7.3 and 4.7.4), a message is passed from the receiving script to the graphical user frontend to alert the monitoring user than this event has occurred. The user can then begin to monitor the brain wave activity of the athlete. A special flag is also inserted into the database so that this event can easily be analyzed later. An activity diagram covering the data preparation is included below in Figure 4-38.

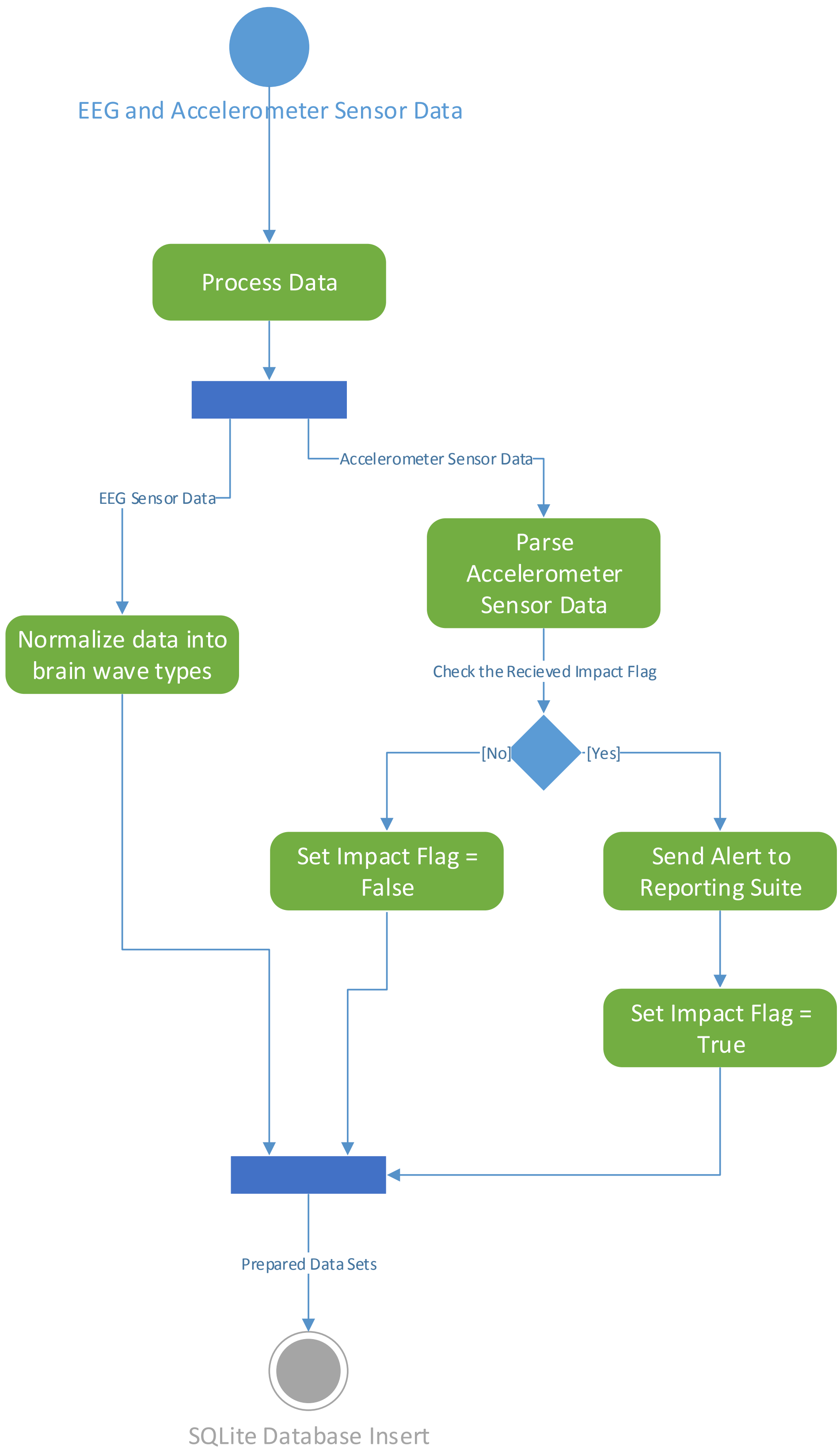


Figure 4-38: Data Preparation Activity Diagram; A UML activity diagram demonstrating the data preparation module of the data receiving Python script

4.8.1.3 SQLite Insert

For database efficiency, and to ensure single write access to the database by any portion of this program to the database, both of the SQL inserts for the accelerometer and EEG sensor data will be performed in one portion of this Python script. This design allows for all database related code to be kept in one portion of the program and prevents errors such as performing multiple, overwriting database inserts at one time. Python includes the sqlite3 DB-API 2.0 interface for SQLite database. This API is utilized by the data receiving script to make a connection to the local SQLite database engine, build the necessary SQL insert statement, and perform the actual insert of the data. This module first defines the connection between the Python data receiving script and the SQLite database engine running on the local server.

The cursor object is created to allow the Python program to read and write to the SQLite database. Various statements can then be executed against the database by calling the cursor object. The sqlite3 API then creates a connection to this database by executing the arguments and configurations performed in the previous step. This module receives as input the sensor data from Section 4.8.1.1.

This module then performs a series of string concatenation steps with the received data to form an SQL compatible insert statement. Next, the sqlite3 API is used to execute and commit the insert statement into the SQLite database running on the BIO-Helmet local server. Lastly, the connection to the database is closed upon exit of the script.

The execute statement performs a preliminary insert into the database. These changes are formally saved by executing the commit method on the conn object and finally (at the end of the running script), the connection to the SQLite database is closed. The data flow diagram for this module of the Python data receiving script is included in Figure 4-39 below.

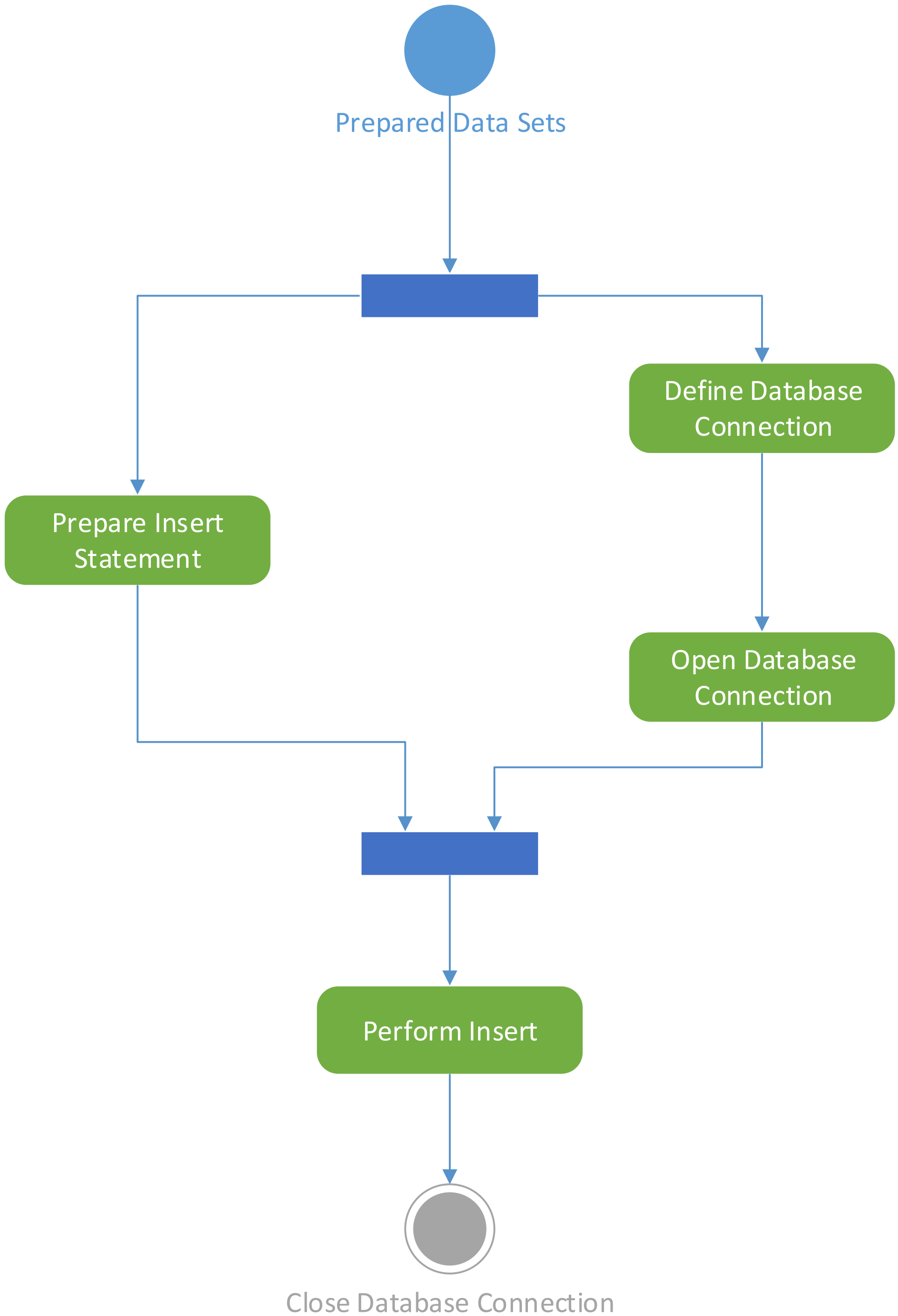


Figure 4-39: SQLite Database Insert Activity Diagram; A UML activity diagram demonstrating the SQLite database insert module of the data receiving Python script

4.8.2 Database

The database engine for the BIO-Helmet local server is SQLite. This database is a fully SQL compatible database engine contained within a single file. This makes the BIO-Helmet sensor database extremely portable and has the ability to be ported to other systems for analysis. The database structure is composed of one database which contains the sensor data for both the accelerometer and EEG sensors. Within this database are two tables; one containing the accelerometer sensor data and another containing the EEG sensor data. Both the accelerometer and EEG sensor data tables are described below in Table 4-7 and Table 4-8, respectively. Each column is labelled with the column name and the data type for that attribute.

ID: INTEGER	Time: TEXT	ForceX: REAL	ForceY: REAL	ForceZ: REAL	GForce: REAL	High Impact: BOOLEAN

Table 4-7: Accelerometer Sensor Database Table; A table demonstrating the accelerometer sensor database structure

The Time column contains the test representation of the time in which that data event occurred. The text is formatted as “YYYY-MM-DD HH:MM:SS” and is populated in the field by using the “now” string entry into the SQLite datetime() function. The “now” string returns the current date in UTC format and then stored as TEXT in the Time column. Each of the Force columns store the value of the accelerometer reading in an eight byte IEEE floating point number for each of the three x, y, and z axes. This precision is more than enough to store the output from the accelerometer sensor mounted on the BIO-Helmet. This datatype can be easily read by the Python data reporting script, detailed in section 4.8.3, and subsequently the MATLAB graphical reporting tools, detailed in section 4.9.2. These values can be converted to an overall g-force vector at the time of data reporting.

ID: INTEGER	Time: TEXT	Alpha: REAL	Beta: REAL	Delta: REAL	Gamma: REAL	Theta: REAL

Table 4-8: EEG Sensor Database Table; A table demonstrating the EEG sensor database structure

The Time column for the EEG sensor data is the same as described above for the accelerometer sensor data table. Each of the five subsequent columns represent five major brain waves. The REAL datatype properly measures the output received from the EEG sensor array on the BIO-Helmet enough precision and data size for these brain wave readings. A formal database UML describing the BIO-Helmet SQLite database is included in below in Figure 4-40.

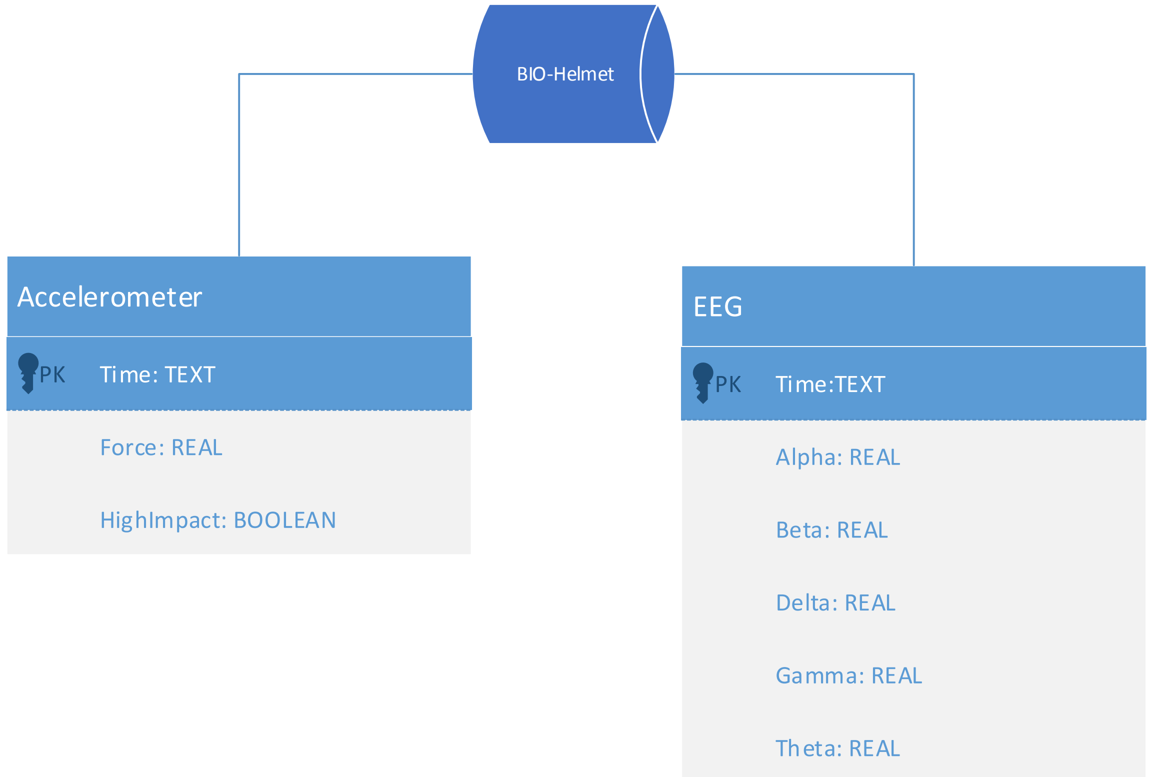


Figure 4-40: BIO-Helmet Database; A UML diagram describing the BIO-Helmet SQLite database design

4.8.3 Python Data Reporting Script

This Python script is designed to dump the BIO-Helmet database into a text file format which is readable by the MATLAB reporting software. Specific design elements regarding the implementation and design elements of the customized MATLAB environment, for both the accelerometer and EEG sensor data is discussed in Sections 4.9.2 and 4.9.3. This Python script uses the sqlite3 DB-API 2.0 interface for SQLite databases. The implementations and specifics regarding this API are discussed further in Section 4.8.1.3. The execute operation for the GUI and data reporting script will perform a SELECT SQL query rather than an INSERT query. A SELECT query will read information from the BIO-Helmet SQLite database and return this information to the program as string for processing.

The execute operator is used to execute a statement against the already configured and connected database. This particular query reads and return all data stored in the accelerometer table. The fetchone() operation will return the result of the previous execute query as a string. The BIO-Helmet Python data reporting script writes this result to a text file for MATLAB to open and read into graphical format.

A class diagram describing the Python data reporting script is included below in Figure 4-41.

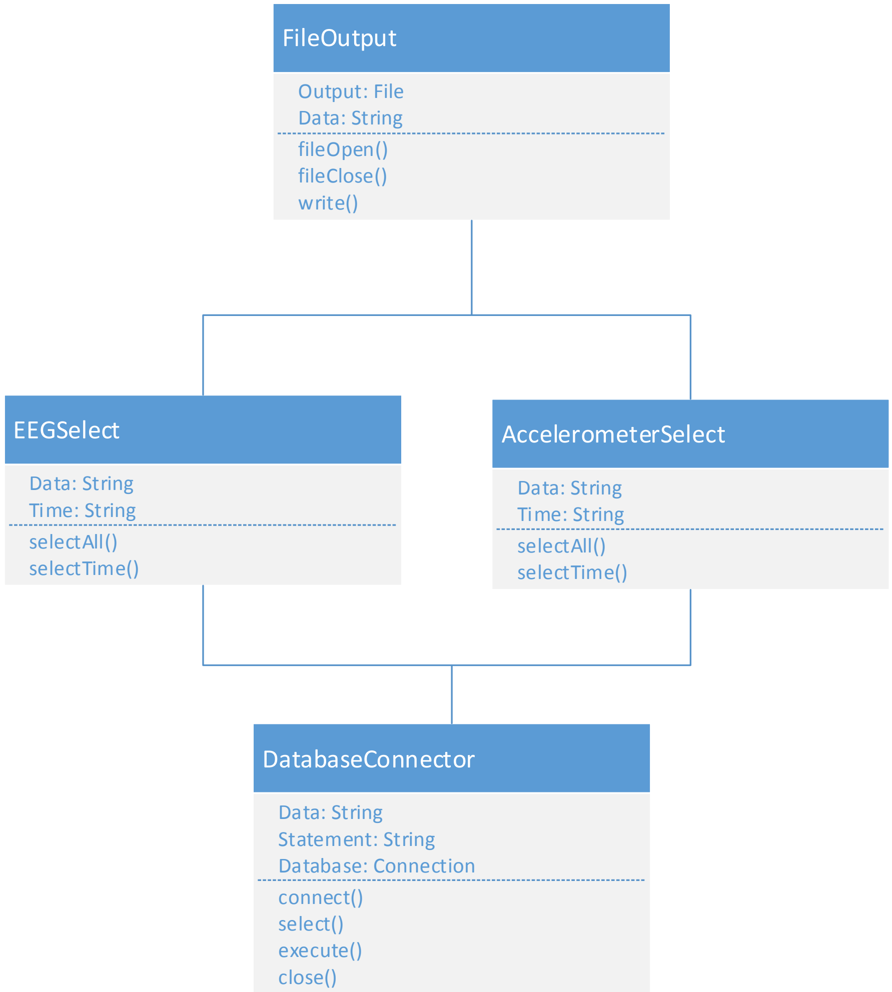


Figure 4-41: Python Data Reporting Script Class Diagram; A figure containing a class diagram for the Python data reporting script

The Python data reporting script first creates and opens a file for sensor data output. The script runs every minute reading all database information from both sensor data tables in the SQLite database and outputs this information to the opened text file. This file is overwritten every time the script runs so that the customized MATLAB environment has access to the latest information from the BIO-Helmet sensor database. A UML activity diagram describing the logical flow and program processes described above is included below in Figure 4-42.

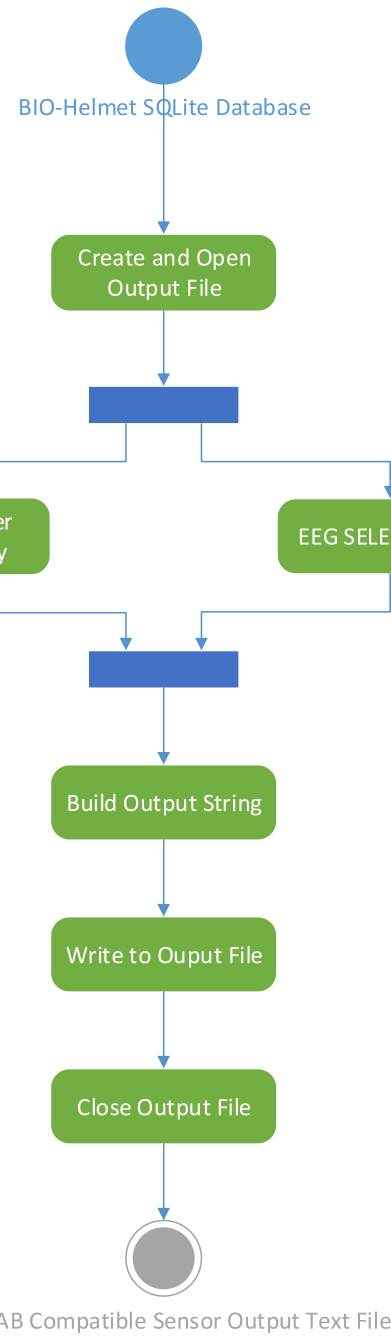


Figure 4-42: Python Data Reporting Script Activity Diagram; A figure containing a UML activity diagram describing the program flow and function

4.9 Reporting Software

This collection of software design subsections cover the user interface and data reporting functionalities for the BIO-Helmet project. Section 4.9.1 covers the design and implementation of a Python based graphical user interface which will run on the BIO-Helmet local server. Section 4.9.2 covers the design and implementation of a customized MATLAB environment which will contain graphs for the accelerometer sensor data as well as the EEG sensor output. This section also covers the operation and use of a MATLAB environment and reading from the output file created by the Python data reporting script (detailed in Section 4.8.3). Finally, Section 4.9.3 discusses the operation, design, and implementation of a customized MATLAB environment for graphical reporting.

4.9.1 Graphical Frontend

This Python program runs the graphical user frontend for the BIO-Helmet reporting software. This program runs on the local server and serves as the direct point of interaction between the user monitoring the sensor data and the BIO-Helmet sensor data. Python's Tkinter is used to implement the necessary GUI features of this design. For more information on Python's Tkinter, please see Section 3.7.1.1. This GUI implements a series of buttons which lead the user into various areas of the reporting software. The user is able to view the raw sensor data in tabular format, directly from the SQLite database. A button is also present which performs a database dump of the sensor data to a text file and will then open a customized MATLAB environment in which the user will be able to view both sets of sensor data in graph format. This graphical frontend also alerts the monitoring user if a high impact event has occurred on the athlete wearing the BIO-Helmet. Lastly, the Python graphical frontend displays the status of the connection status between the local server and the BIO-Helmet. A graphical user interface layout diagram for the main page of the Python graphical frontend is included below in

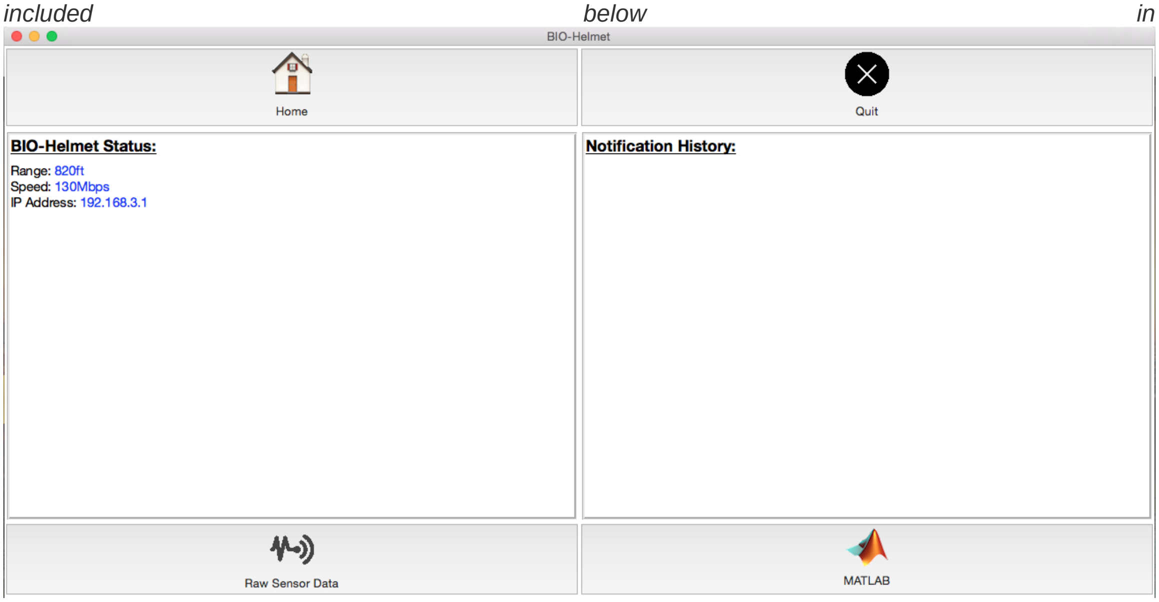


Figure 4-43.

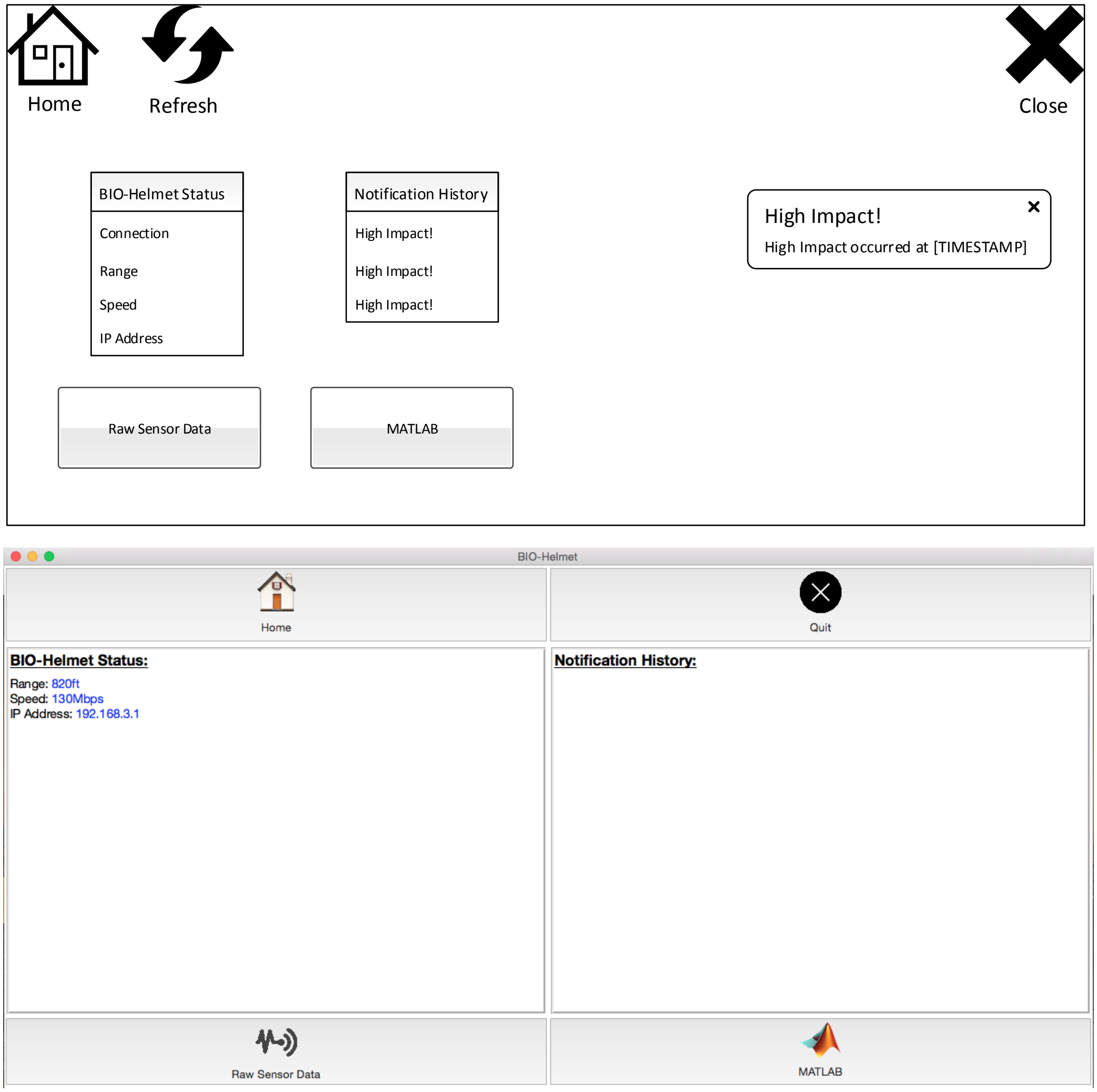


Figure 4-43: Python Graphical User Interface Layout; A figure containing the GUI layout for the main page of the BIO-Helmet graphical user interface

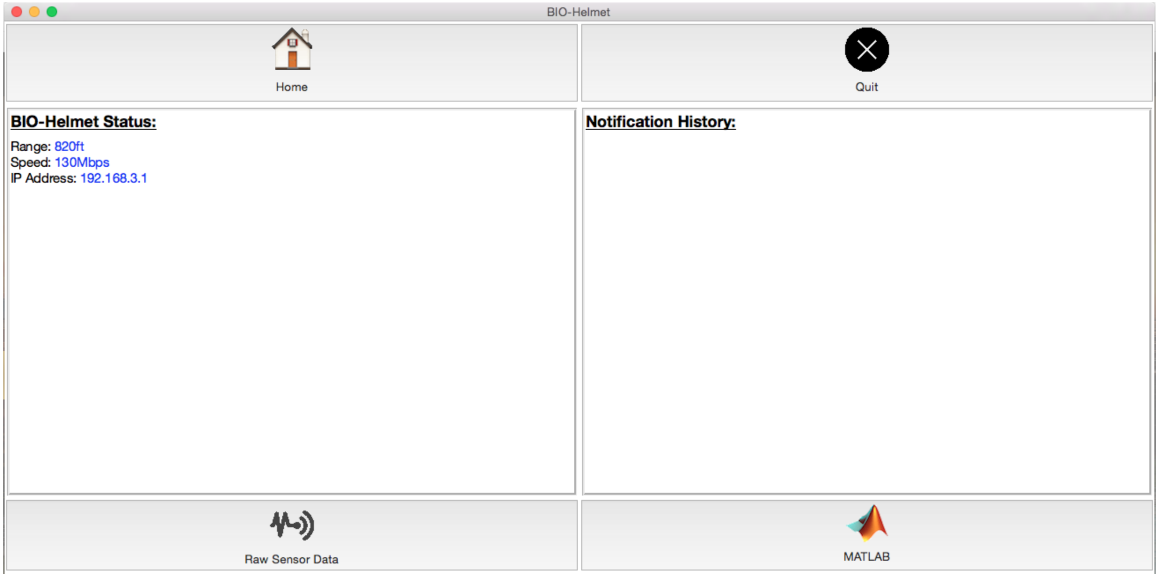


Figure 4-43 describes the layout of the homepage for the graphical user interface. The Home, Refresh, and Close icons remain present on all pages and subsections of the Python graphical user frontend. The sample notification pop-up is also available (if a high impact event occurs) on every page and subsection of the Python graphical user frontend. The Home button returns the user to this main page. The Refresh button updates all items on the current page the user is viewing. The Close button closes the database and BIO-Helmet connections and exits the local server application. This notification contains the message (high impact event) as well as the timestamp in which that event occurred. This notification window also flashes red to alert the monitoring user that this event has occurred. The main page contains a box with a list of the various connection status parameters between the BIO-Helmet and the local server. The connection status (connected or disconnected), the maximum range between the BIO-Helmet and the local server, the speed of the Wi-Fi wireless direct connection, and the IP Address of the BIO-Helmet are included. Another box contains a history of notifications received from the BIO-Helmet. Lastly, the front page contains two buttons. The first button leads the user to the raw sensor data output. Clicking this button queries the SQLite database and display the raw sensor data page; see Section 4.9.1.1. The second button dumps the current database to MATLAB compatible text file and launch MATLAB for viewing the sensor data in graph format.

A UML activity diagram describing the user interface flow of the main page of the Python graphical front end mentioned above is included below in Figure 4-44.

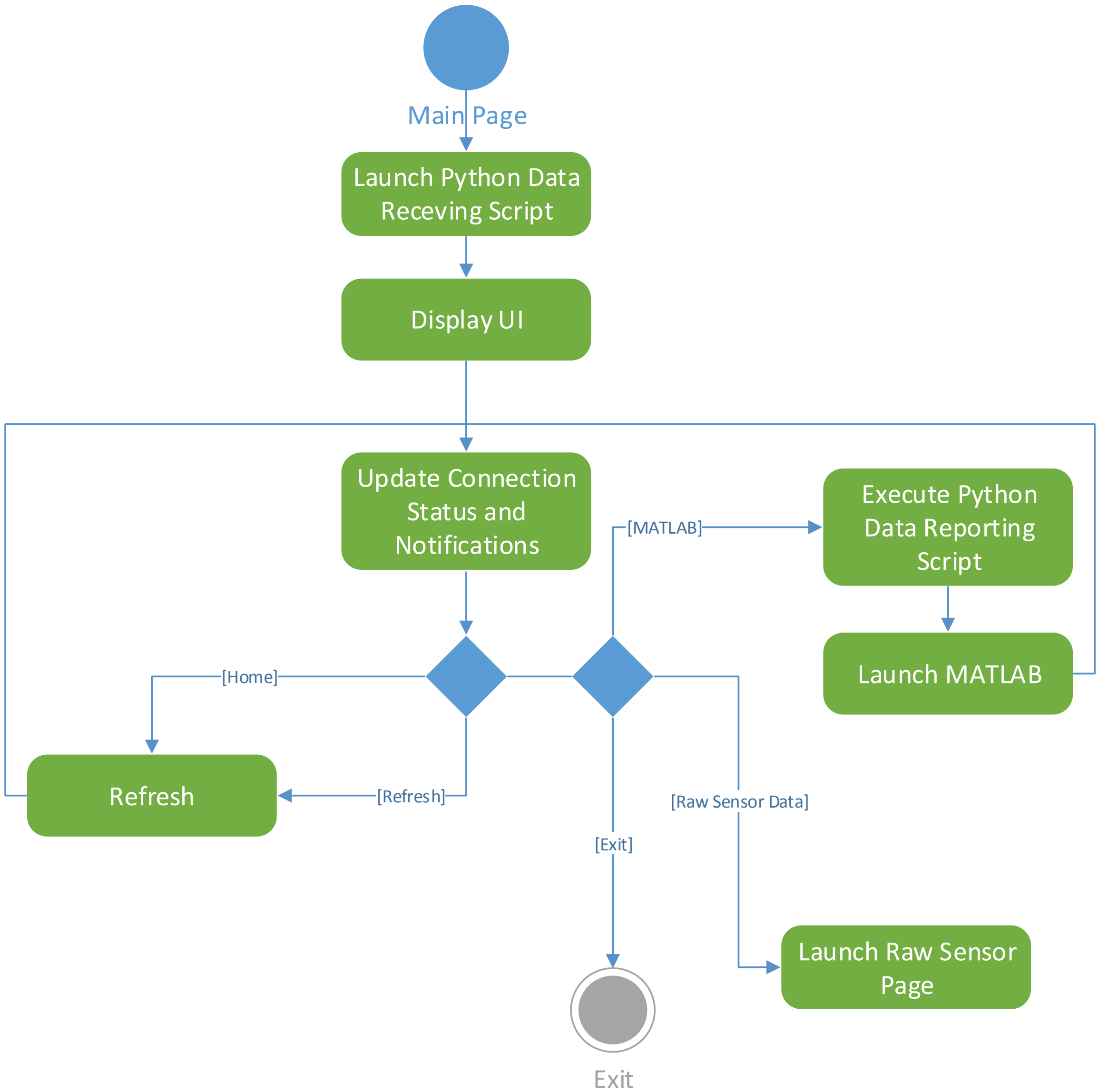


Figure 4-44: GUI Homepage; A UML activity diagram describing the user interface specific flow of the home page for the Python graphical user frontend

A class diagram representing the implementation of the Python graphical frontend is included below in Figure 4-45.

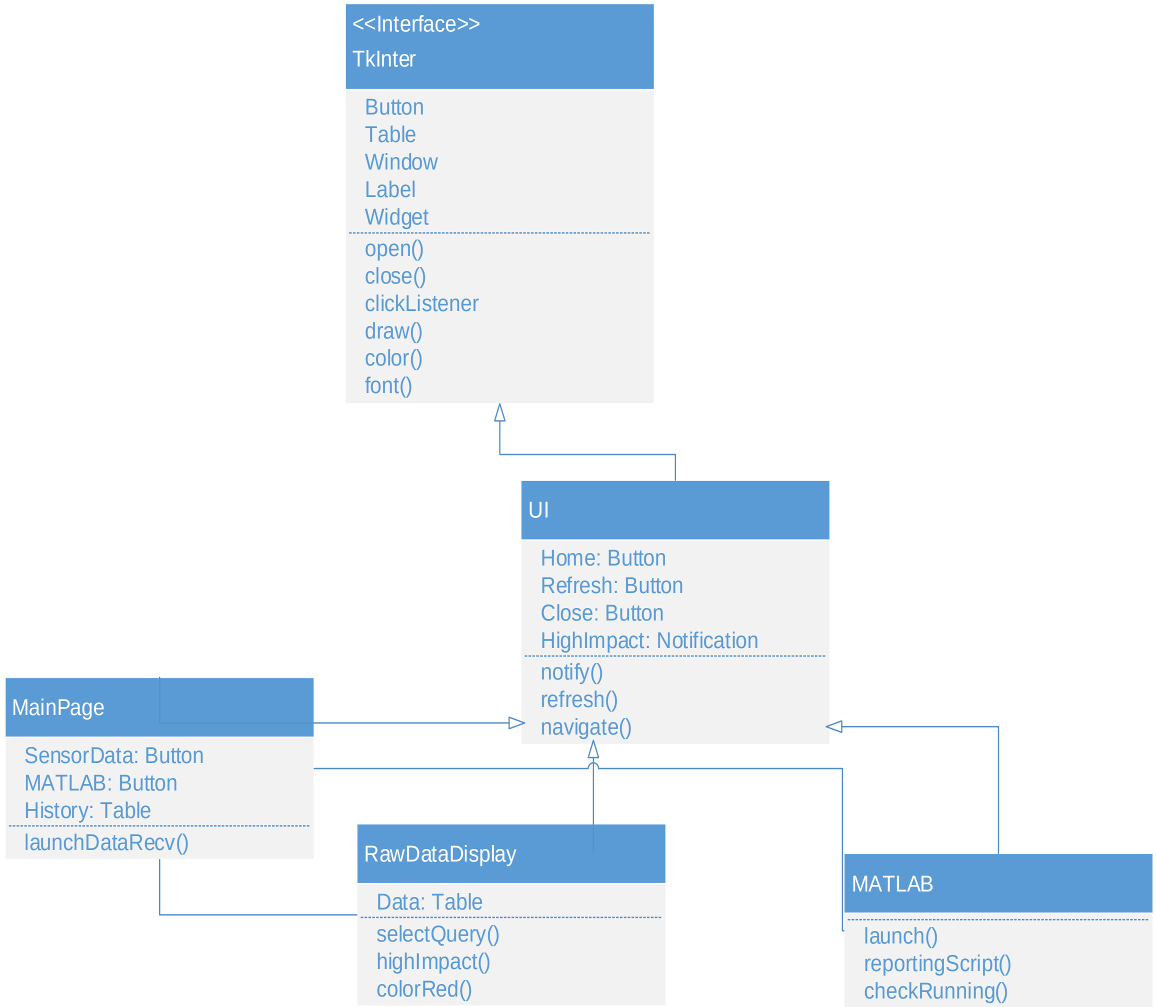


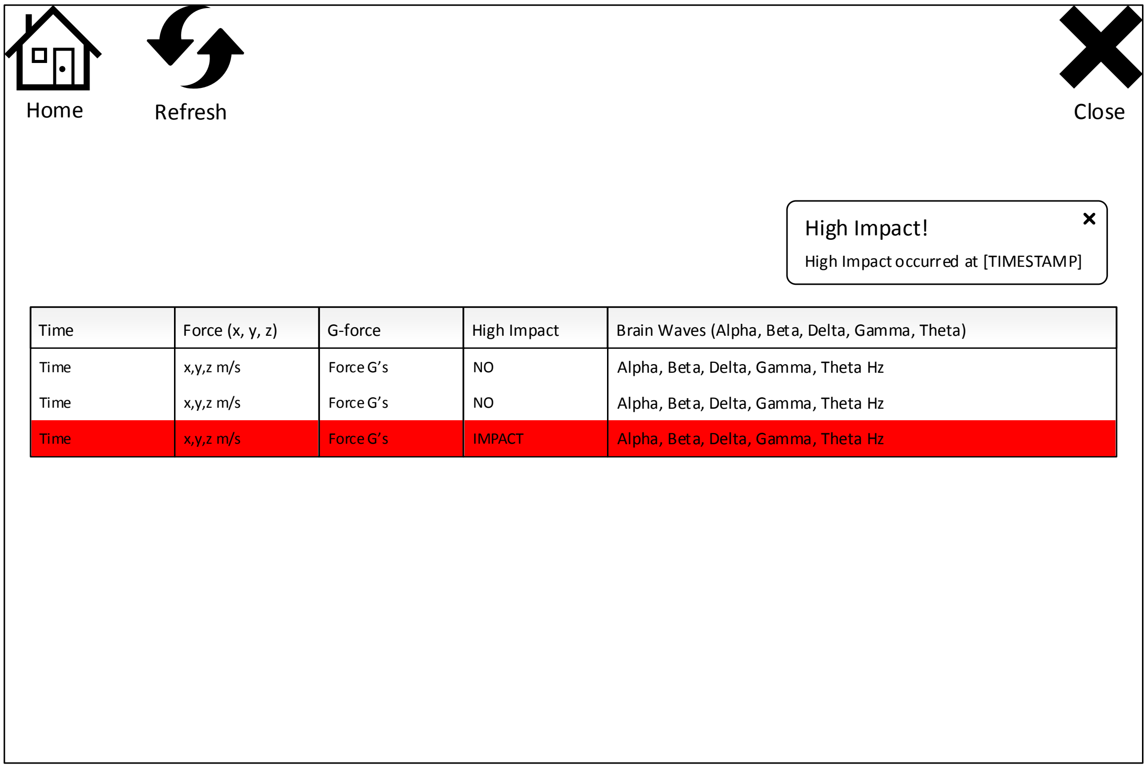
Figure 4-45: Python Graphical Frontend Class Diagram; A UML class diagram representing the implementation of the BIO-Helmet local server GUI

Tkinter operates by creating various widget classes and defining defs. This is very similar to HTML and CSS coding where an object is segmented with a <div> and the corresponding cascading style sheet is applied to that HTML element.

Tkinter includes many other widget types that will be used for the creation of the BIO-Helmet graphical user frontend. When the program is executed, the Tkinter API calls the createWidgets function to populate the generated window with the specified widgets. The message widget is especially useful for the notification pop-up that a high impact event has occurred. The text widget will also be useful for displaying all textual elements in the GUI, especially the raw sensor data output page.

4.9.1.1 Raw Sensor Data Viewing

This section covers the raw sensor data viewing page of the Python graphical user frontend. The raw accelerometer and brain wave data are both be contained in a table, sorted by time. A graphical user interface layout diagram for the main page of the Python graphical frontend is included below in Figure 4-46.



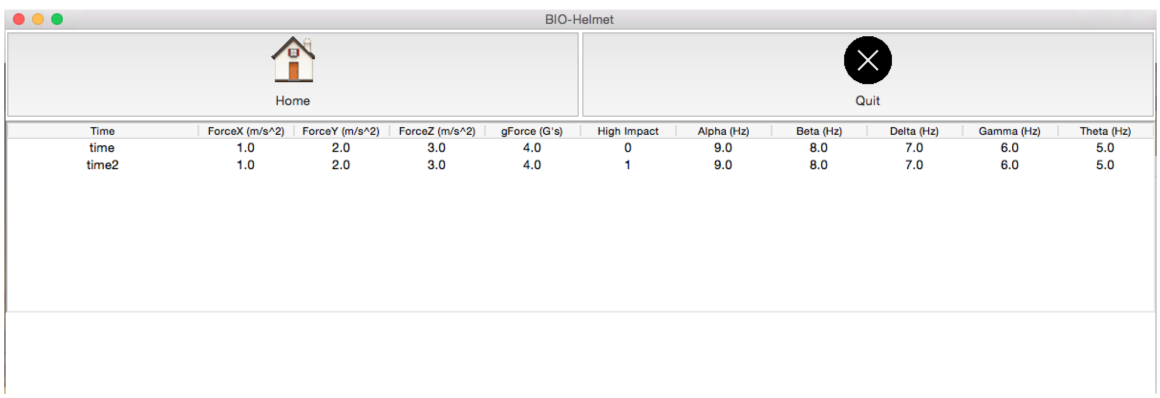


Figure 4-46: Python Graphical User Interface Raw Sensor Data; A figure describing the GUI layout of the raw sensor data display screen of the Python graphical frontend.

The Home button returns the user to the main page of the graphical user frontend and the Refresh button updates the table with the latest sensor data. The Close button and high impact notification behaves as described in Section 4.9.1. The table described in Figure 4-46 contains each time point sorted from oldest to most recent. Each entry in the table contains the time the reading was taken, the accelerometer sensor output, and the EEG sensor output. The accelerometer data contains the observed value in each of the x, y, and z directions as well as an overall g-force value. The Boolean flag for whether a high impact event occurred is also displayed. The EEG sensor array output will display each of the five brain waves measured.

A UML activity diagram describing the user interface flow of the raw sensor data viewing page is included below in Figure 4-47.

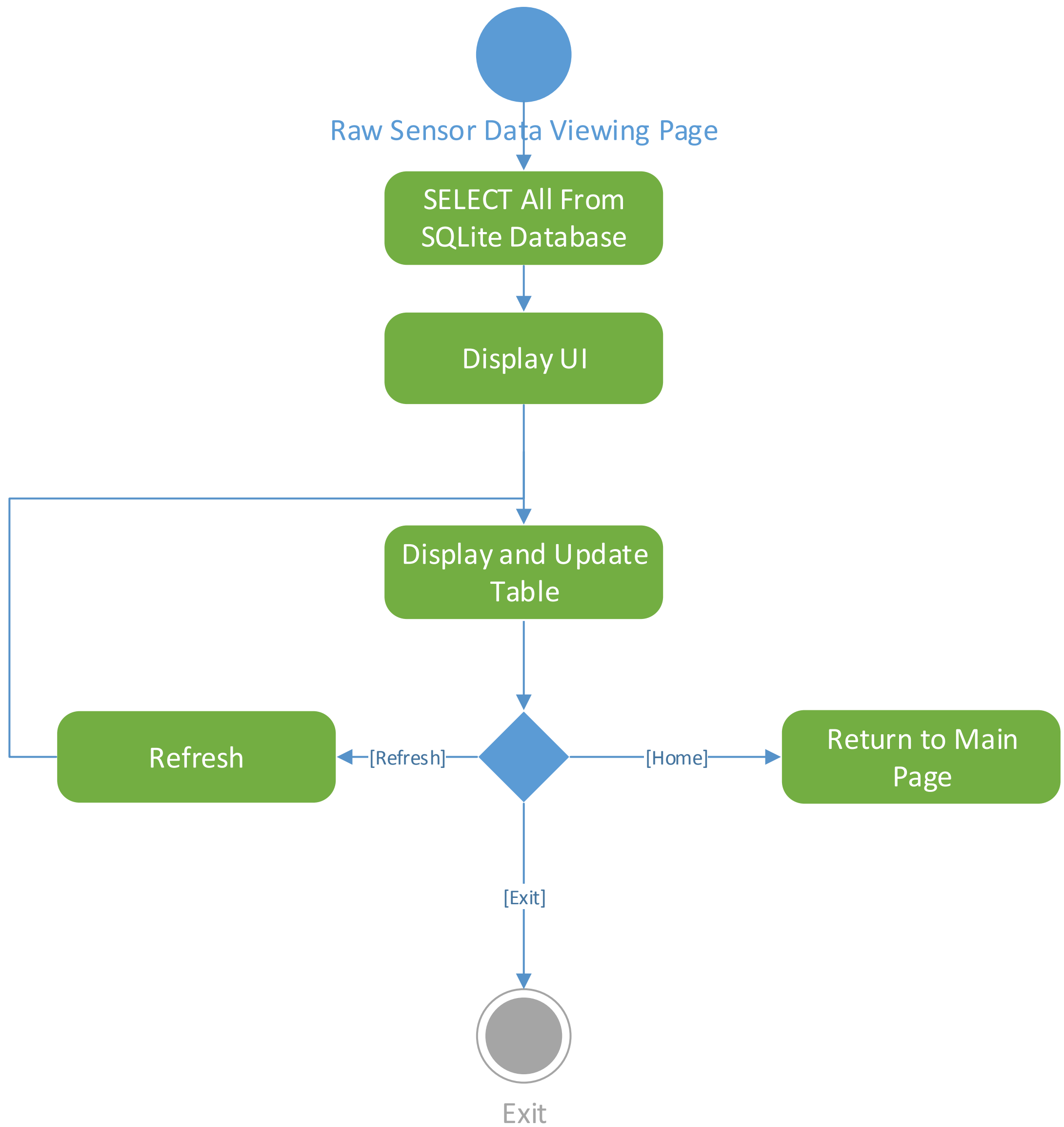


Figure 4-47: GUI Raw Sensor Data Viewing; A UML activity diagram describing the user interface specific flow of the raw sensor data viewing page for the Python graphical user frontend

4.9.1.2 MATLAB Launch

This section covers the design and implementation of the main page MATLAB button. This MATLAB launch function is considered to be a function of the main page of the Python graphical user frontend. This button calls the Python data reporting script (see Section 4.8.3) which performs a database dump of both the accelerometer and sensor data in a MATLAB compatible text file. The Python graphical frontend then checks if MATLAB is already running, if not this Python program opens MATLAB, and then loads a customized MATLAB environment that displays both sets of sensor data in graphical form. Once MATLAB is launched, the user will have to return focus back to the GUI window for the Python graphical user interface in order to view any status items on the homepage or see notifications that a high impact event has occurred.

4.9.1.3 High Impact Notification

This notification window has the ability to appear on all pages of the BIO-Helmet Python graphical user frontend. If the BIO-Helmet detects a high impact event, a Boolean flag is sent from the BIO-Helmet (see Sections 4.7.3 and 4.7.4) to the local server. The Python data receiving script will enter this value into the database and send a message to the Python graphical user frontend to display the notification pop-up. This notification shows no matter what page the monitoring user is on. The notification is not an OS level notification (such as those in Windows 8.1 and OS X 10.10) and thus only displayed within the GUI window. Upon message receipt, the notification is also inserted into the Notification History table present on the main page of the Python graphical user frontend. A UML activity diagram describing the notification flow beginning with the received message from the Python data receiving script is included below in Figure 4-48.

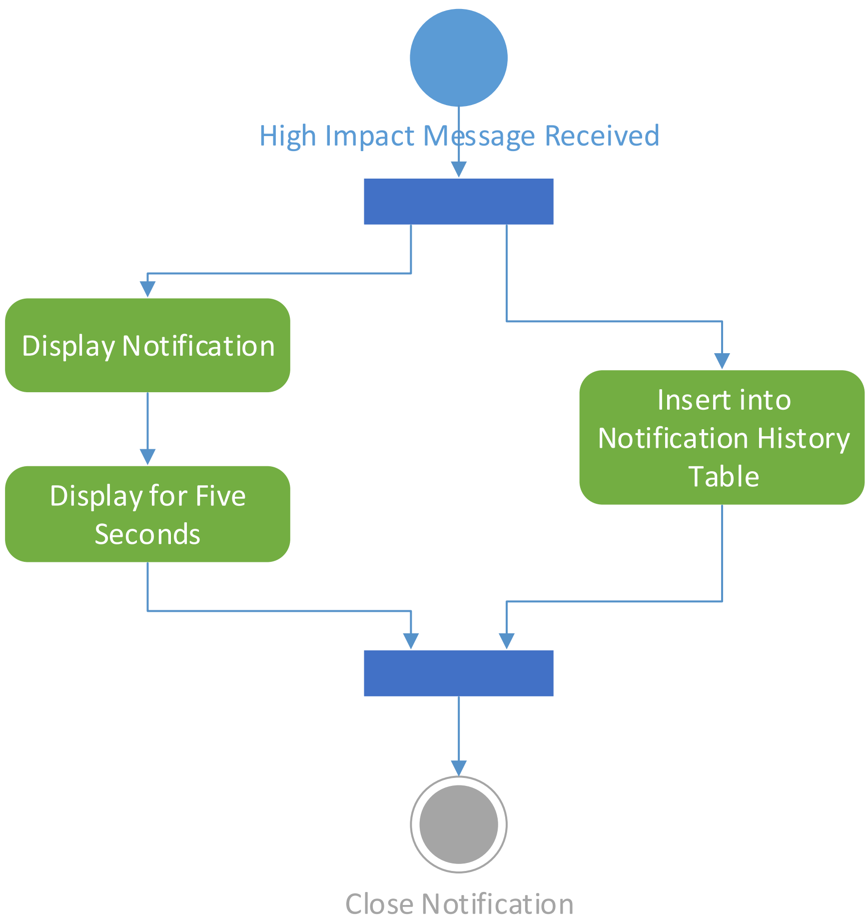


Figure 4-48: High Impact Notification UML; A UML activity diagram describing the high impact notification flow of the Python graphical user frontend

4.9.1.4 BIO-Helmet Connection Status

This section discusses the design and implementation of the BIO-Helmet connection status table present on the main page of the Python graphical user interface. This table displays whether or not the BIO-Helmet is connected to the local server, the maximum range between the local server and the BIO-Helmet, the bandwidth of the Wi-Fi wireless direct connection, and the IP address of the BIO-Helmet. All of this information is obtained from the Python data receiving script (where the socket connection over wireless is made) and referenced directly by the GUI. The refresh button on the main page will update this connection status table with any new parameters that have been read in from the local server Python data receiving script. The line indicating whether the BIO-Helmet is connected to the local server is essential for testing and operation. Although none of the other connection status monitors are expected to change, due to constructing only being one prototype, these status indicators are designed for future use of multiple BIO-Helmets.

4.9.2 MATLAB

Matlab is a high level computing language that allows for complex problems to be solved via the computer. We use this computer language to decipher the massive amounts of data we get from not only the EEG sensors but the accelerometer sensors as well. MATLAB is the perfect program to display our data correctly in forms of charts and numbers that can be interpreted to read the symptoms of a concussion. Using this computer language we allow not only EEG specialists to be able to interpret the readings that are coming from our device but also the average user that may have very little medical background on this matter. A huge advantage of using MATLAB is that it is a very established program with a very large community of users that can help. There are also plenty of tutorials and libraries for users that are starting and wish to learn about the program. MATLAB does have its disadvantages however, if the user does not understand any computer language at all transitioning to such a difficult language with so many shortcuts and keys that are unknown to a beginner be prove to be challenging. In other words, it is not very user friendly to a person that does not have any form of programming expertise in his or her background. To fix this, a script that can run off of MATLAB has been implemented.

4.9.3 MATLAB EEGLAB

MATLAB EEGLAB is an interactive toolbox for people who want to utilize MATLAB with their EEG sensors in order to read and compute the data that is coming from the sensors. It is a very powerful toolbox that allows the user to customize and incorporate what they need and allows visual aids in order for the user to easily interpret the data. The toolbox is open sourced so it is completely free to use and would have been a great asset in our project is it greatly reduces the amount of outside programming that will be required of us. It has a graphical user interface which allows the user to easily interpret the data in both independent components analysis and time/frequency analysis as shown below in Figure 4-49 below.

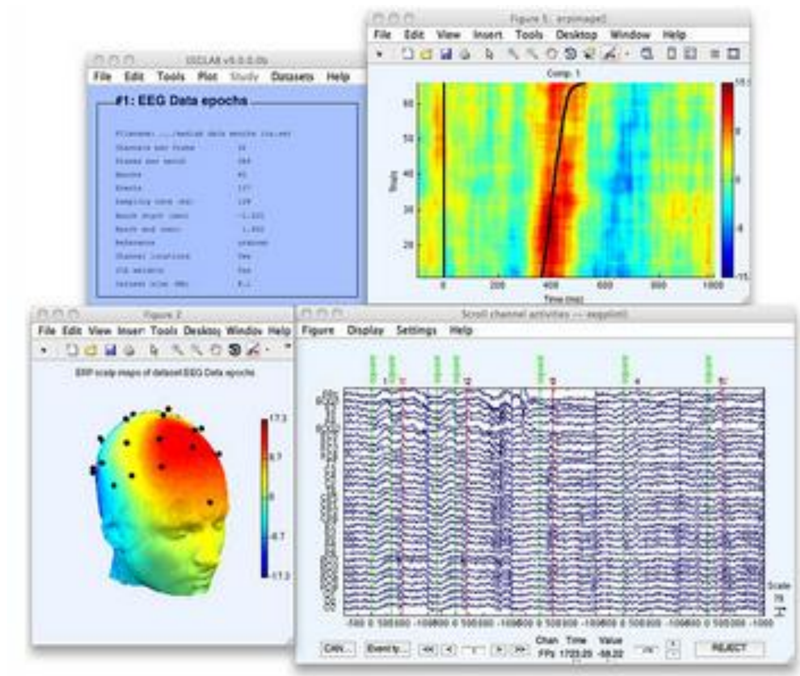


Figure 4-49: Matlab EEG GUI; The graphical user interface of matlab EEGLAB; reprinted with permission from EEGLab

Unfortunately, this toolbox is exclusively for commercial EEG products and was not possible to implement in this project. A similar customized environment was developed that can be viewed in below Figure 4-50.

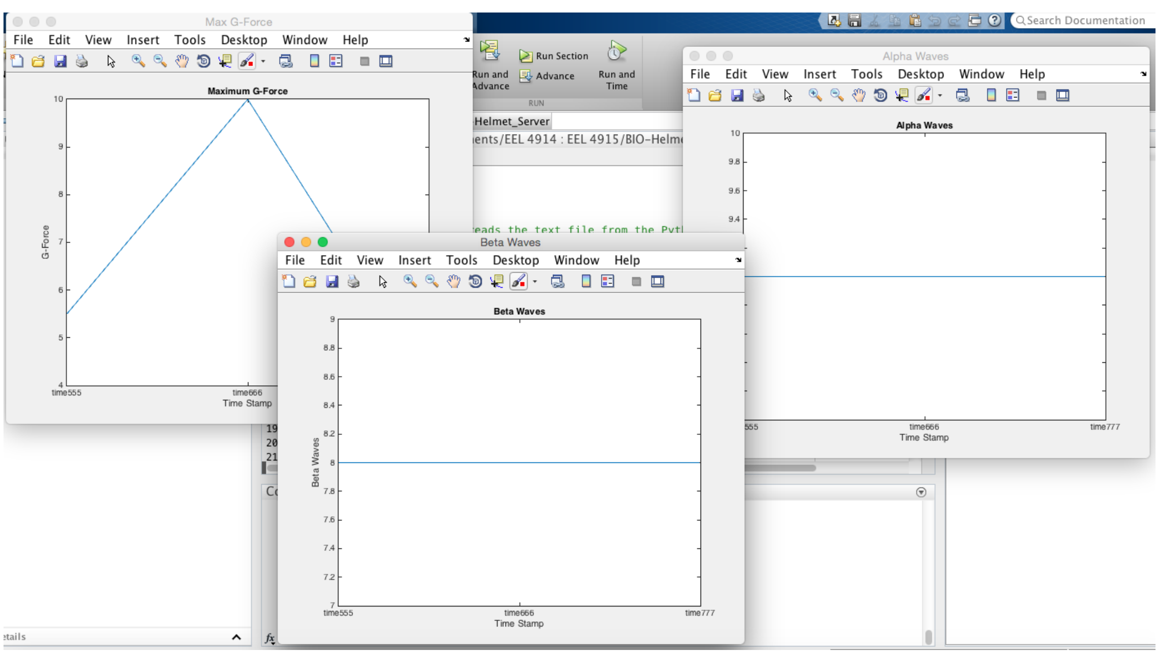


Figure 4-50: Customized MATLAB EEG Reporting; A customized MATLAB environment developed for the BIO-Helmet reporting software suite

5 PROJECT PROTOTYPE CONSTRUCTION AND CODING

5.1 PARTS ACQUISITION AND BOM

The following sections describe the acquisition of parts and bill of materials for each part of the project. It is divided into six sections: accelerometer, EEG array, charging and battery, Wi-Fi module, software, and microprocessor. The parts are attached by surface mount to the main PCB which is to be built professionally and surface mounted by the team. Table 5-1 below details the bill of materials. Any part names followed by an SM indicate surface mounted parts.

5.1.1 Parts Acquisition

Parts acquisition tok place as soon as Senior Design II starts. It was imperative to get the parts as quickly as possible to ensure that they not only work together but that there was also enough time to test the design before ordering a printed circuit board. To receive funding from the Boeing sponsorship, all parts requiring reimbursement must be sent directly to UCF. All parts chosen were in stock.

The following lists details some of the vendors that were used to purchase parts:

- Analog Devices – integrated circuits, online orders
- Digi Key – surface mount components, online orders
- Microchip – integrated circuits, online orders
- Mouser – surface mount components, online orders
- Texas instruments – microprocessor and integrated circuits, online orders
- Adafruit – DIY electronics retailer, online orders
- Sparkfun – DIY electronics retailer, online orders

For low cost items, two or more identical parts were ordered. This prevented lost time due to the inevitable damage of parts during the build and testing process. This also ensured that no difficulty was had in acquiring parts due to parts going out of stock during the build process.

5.1.2 Bill of Materials

The following table represents the final Bill of Materials for the BIO-Helmet project. This table includes the specific part ordered, the cost of that part, the quantity of that part needed to complete the project, and the total cost for the quantities of that part. The vendor who will supply the part is also included. The Bill of Materials table is divided up into each of the major subsections of the project with a final total project cost in the last row of the table.

Accelerometer Sensor				
Part	Cost	Quantity	Total	Vendor
ADXL377	$11.49	1	$11.49	Analog Devices
Capacitor SM	$0.24	4	$0.96	Digi Key
EEG Sensors				
Part	Cost	Quantity	Total	Vendor

COM-10969 ROHS	$7.95	2	$15.90	SparkFun
TL084cdr	$0.50	4	$2.00	Texas Instruments
INA114	$11.59	4	$46.36	Texas Instruments
PRT-00124 ROHS	$6.95	3	$20.85	SparkFun
709-1110-ND	$53.98	1	$53.98	Digi Key
511-L7805CV	$0.48	1	$0.48	Mouser Electronics
445-10G-48TP	$85	1	$85	Jari Supply
PCB	$53.33	1	$53.33	OSH Park
Wi-Fi Module				
Part	Cost	Quantity	Total	Vendor
CC3100	$14.07	1	$14.07	Texas Instruments
Microprocessor				
Part	Cost	Quantity	Total	Vendor
TM4C123GH6PI7	$11.42	1	$11.42	Texas Instruments
PCB	$53.33	1	$53.33	OSH Park
16 MHZ Crystal	$1.50	1	$1.50	Sparkfun
Misc Parts	$47.75	1	$47.75	Mouser
Power Supply				
Part	Cost	Quantity	Total	Vendor
MCP73833	$0.85	1	$0.85	Microchip
Battery	$29.50	1	$29.50	Adafruit
LED	$0.35	3	$1.05	Sparkfun
Resistor SM	$0.10	6	$0.60	Mouser
Micro USB SM	$1.50	1	$1.50	SparkFun
JST SM	$0.95	1	$0.95	SparkFun
Capacitor SM	$0.24	2	$0.48	Digi Key
Misc /Software				
Helmet	$169.00	1	$169.00	
MATLAB License	$49.99	1	$49.99	
Total Cost:			$672.35	

Table 5-1: Bill of Materials; A table describing the bill of materials for the BIO-Helmet project

5.2 PCB Vendor and Assembly

The assembly for the board will take place in two parts. First, a breadboard was used to assemble all the parts for testing. Additional hardware specific testing is detailed in Section 6.1. Second, two PCBs were ordered and then any parts will be surface mounted professionally.

5.2.1 Breadboard Build-Out

By building the design out on a breadboard first, it allowed for correction of any incompatibilities between parts that were not apparent during the design phase of the project. To test the parts with the microprocessor, a Tiva C series LaunchPad evaluation board was used. For some of the more complicated ICs that the BIO-

Helmet is using, such as the ADXL377 accelerometer, an evaluation board was used to help interface with the Tiva C LaunchPad. An example of this can be seen in Figure 5-1 below.

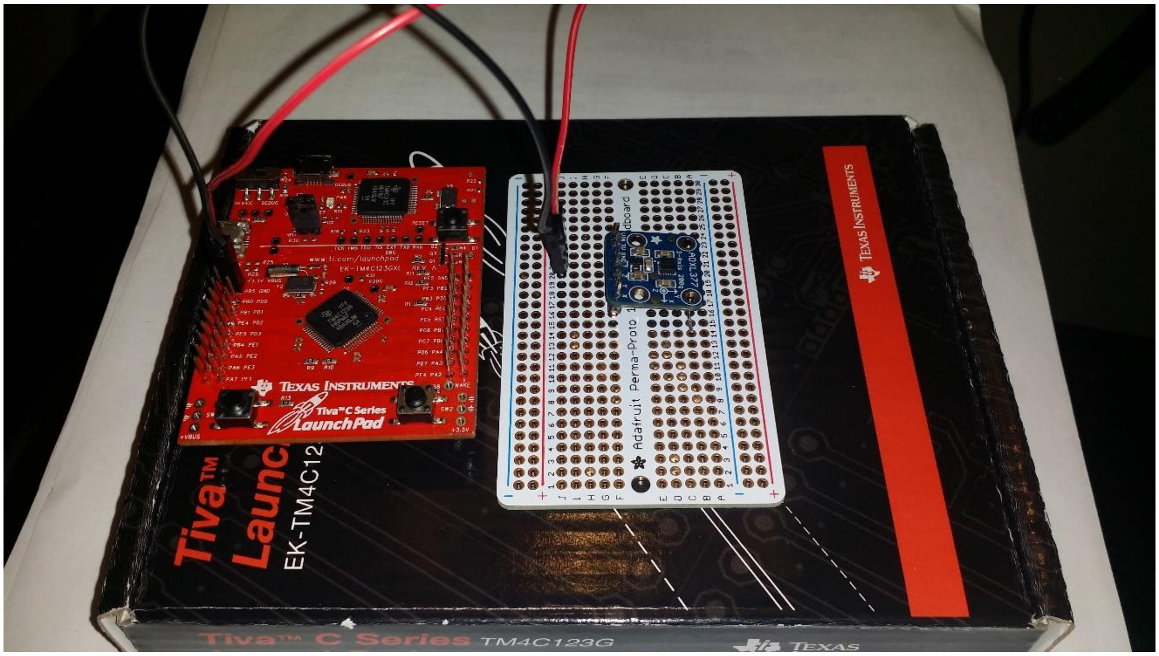


Figure 5-1: Tiva C LaunchPad Test Interface; An image of an example of using the Tiva C LaunchPad to interface with a separate IC

Other circuits that contain many small individual components were built out separately on a breadboard and connected from there to the LaunchPad. The EEG circuit needed to be done in this way to ensure that everything is working as expected. Testing equipment included a digital multi-meter to ensure correct voltages to each part, an oscilloscope for testing the EEG sensor circuit and waveforms, and a power supply for applying test voltages to different parts of the design.

5.2.2 PCB Vendor and Assembly

After ensuring all project parts function together seamlessly, and a final design is reached, the two PCBs were ordered through OSH Park for a price of $53.33 each. The PCB had no components once it was manufactured so surface mounting of parts has to be done. There are two ways to approach surface mounting parts on a PCB. First, it can be done manually or by a third party company. For budgetary constraints, parts were surface mounted using solder paste and hot plate. The EEG PCB used through hole components and these were attached by hand soldering.

Soldering is a very cheap process and can be done with little experience. The process of soldering involves the following steps. First, the surface of the PCB needs to be prepared. This can be done by cleaning the surface of the board with a scotch bright pad and using acetone to wash off any contamination. The second step is to place all the components on the board, paying special attention to orientation and polarity of each component. The typical procedure is to place the smallest components first, such as resistors and diodes, and then move on to the bigger components such as capacitors and any integrated circuits that can be soldered. The last step is to solder the component using a solder gun and lead solder. It is important to make sure that cold solder joints and bubbling solder caused by overheating are avoided. These can lead to electrical and structural instability on the PCB, these are imperative to the BIO-Helmet due to the high velocity impacts that it will have to endure.

5.2.3 Helmet Installation

Due to the constraints of the project, the goal was to fit the boards within the helmet. In the final design it was important that no electrical parts, beside the EEG electrodes, came in contact with the user's head. With this in mind, an enclosure was constructed to not only hold and protect the main boards but to also allow it to fit comfortably inside the helmet and out of harm's way.

This enclosure was made from foam and fit very close to the board. Figure 5-2 shows the locations for the installation of the battery pack and the main boards.

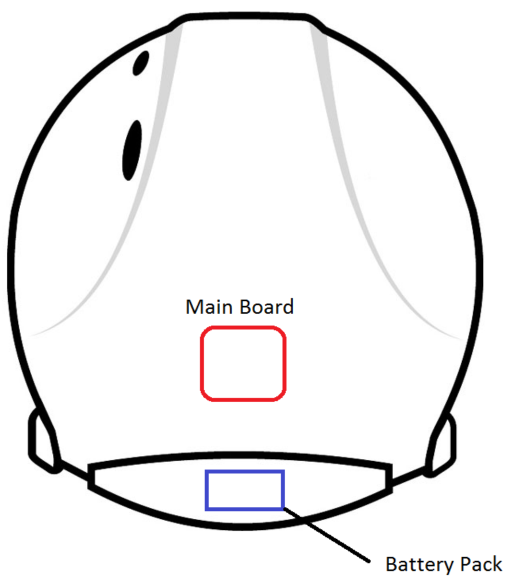


Figure 5-2: Component Layout; A figure demonstrating the location of the battery pack and the main PCB within the helmet

These locations were shown for several reasons. First, the battery pack needed to be easily accessible for replacement. By placing it at the bottom back of the helmet, under the internal helmet padding, allows for these quick battery changes and keeps it out of the most common impact zones. The location of the main PCBs were chosen for similar reasons. It kept the sensitive components out of common impact zones and allows fairly easy access. This location is also central to helmet and allowed for the running of lead wires out to the EEG sensors and up to the accelerometer which was placed as shown in Figure 4-16.

As stated previously, all components and wiring were placed underneath the helmet padding, except for the EEG electrodes which must have been in contact with users head. This ensured that the integrity of the helmet was maintained and its main purpose ensured, to protect the head. Comfort was taken into consideration as well; having electrical components accidentally poking the player in the head or getting hot and burning the player would violate the project's safety constraints. Keeping the padding intact prevented these Issues from occurring.

5.3 Final Coding Plan

This section details the final coding plan for the software aspects of the BIO-Helmet project. These sections include the programming interfaces used, the programming languages used and is divided into the software which will run on the local server and the code which will run on the Tiva C ARM Cortex microprocessor. All code for the BIO-Helmet is stored in a public GitHub repository so that all group members have access to view and make changes to all BIO-Helmet code.

5.3.1 Energia

Energia is an IDE released by Texas Instruments designed for developmental use on various Texas Instruments microcontrollers. This IDE is based on the open source Wiring IDE and is used to develop simple sketches. The embedded program for the Tiva C ARM Cortex MCU will be written in Energia. This program is described in detail in Section 4.7. The coding plan for the embedded program was to focus on the initializations and input pins configurations first. Debugging then begin with a simple sketch which configured the microprocessor as necessary for this project. The next phase of development was reading in the inputs from the accelerometer sensor and the EEG sensors. The data processing and averaging code was also implementing in this portion of development. Testing and debugging was performed at this stage using the USB UART configured on the microcontroller and simply printing the read values to the console screen to confirm that each pin is read. More detailed explanations of the planned software testing are included within Sections 6.2 and 6.3. The last phase of development was writing the output data to the wireless communication module UART for sending to the local server. This stage of development took the least amount of time and debugging and was therefore reserved for the end of development. The final embedded program was written to the Tiva C ARM Cortex ROM from Energia using the USB UART debug port.

5.3.2 Python

Sections 4.8.1, 4.8.3, and 4.9.1 discuss the implementation of several Python programs for the BIO-Helmet local server. Each of these programs was developed within Xcode. Python specific testing will be discussed in Sections 6.2 and 6.3.

The first Python program developed is the Python data receiving script. This program is pertinent to the BIO-Helmet project and continually testing and implemented was performed so that the other Python programs have valid input from the BIO-Helmet. This program was invaluable for testing that data was properly sent from the BIO-Helmet to the local server over the wireless direct TCP socket. Debugging of this program relied on the completion of the embedded program described in Section 4.7. The SQLite database subsystem also needed to be configured before the database insert statements were tested and debugged. The database configuration and design are discussed in Section 4.8.2.

The next Python program that was implemented was the Python data reporting script. This script dumps information from the SQLite database to a file format which can be read by MATLAB. This program was very easy to implement but was critical when developing the MATLAB environment for reporting sensor data.

The last Python program to be implemented was the graphical user frontend. This GUI did not take very much time to implement and GUI related issues were trivial in comparison to the basic functioning of the BIO-Helmet. GUI related bugs were saved for fixing until the very end of the coding plan. This way, it was ensured that the basic functionality and data transfer aspects of the BIO-Helmet project were in full operating condition for project prototype testing, described in Section 6.

5.3.3 SQLite

The SQLite coding plan consisted of configuring the SQLite database in the manner outlined in Section 4.8.2. This configuration took place against a single SQLite file which was configured with the proper database tables to store the sensor data. This database configuration was completed before the debugging phase of the Python data reporting script. Because of this dependence on a major piece of code for this project, the database configuration and setup was performed before all other local server code implementations.

5.3.4 MATLAB

The MATLAB coding plan consisted of writing a MATLAB script and configuring MATLAB to read in the text file created by the Python data reporting script. This MATLAB environment was modified to include graphical forms of the accelerometer and EEG sensor data. The first item that was completed in this coding plan was configuring and debugging the MATLAB text file input. This display and reporting suite was integral to the display of the brain wave data, as no other reporting interface was developed for this data. Once the MATLAB script is properly displaying all data from the EEG sensor array, custom graphs for the accelerometer data were also added to this MATLAB environment. Since, the accelerometer data was able to be viewed from within the graphical user frontend, implementing these custom graphs within this MATLAB environment was saved for the end of the MATLAB coding plan.

6 PROJECT PROTOTYPE TESTING

6.1 HARDWARE SPECIFIC TESTING

The following sections discuss testing of each individual hardware element included in the BIO-Helmet system. There are two areas of testing within the hardware section. The first one is the accelerometer section, and the second one is the EEG section. Both of them need previous testing to confirm they are working properly. In order to determine the functioning of both sections, a testing procedure was developed. The hardware design for the EEG is made up of several stages of amplification and filtering. This plan includes individual areas of testing and then a final overall test of the system. The first area of testing needed was the circuit functioning. To detect any faults in the circuit, a procedure for determining the location of the fault was developed. Using an oscilloscope and a digital multimeter, the design was tested by sections individually. It was important that all the values of the gains and filtering matched the values stipulated in the original design. Probable causes of hardware failures may be short circuits, open circuits, resistance, ESD, and others. Other testing was needed for the hardware was based on the use. Since the main use is by athletes playing a contact sport, the system must surpass the effects of the elements. Some of these elements include water, temperature, pressure, vibration, shock, and humidity. Hardware testing includes the approach to protect the system from these elements.

6.1.1 Water Testing

The system will be in contact with some water as games are usually played through rain showers. This means that the circuitry must be protected in order for the device to keep functioning. The two options available to complete this requirement, in a future version of the BIO-Helmet, would be to either create a case that would attach itself to the helmet and cover the circuit from water or use epoxy to create a protective layer. Both options would work but the plastic case would be a more esthetic option. The testing for this requirement would involve having both design options on the athlete during rain and observing the device. After removing the BIO-Helmet, check how much of the water was able to penetrate the casing. A final option would be to have the system take the shape of a helmet and be able to house the entire system inside of the helmet. This would eliminate the trouble of outside elements.

Another form of water that our system might become exposed to is bodily fluids. Football is a full contact sport and there is a very high likelihood that there might be blood, sweat, or saliva that might come into contact with our device. To mitigate this, in a future version of the BIO-Helmet, we will reduce the amount of physical interaction between the player and our internal parts. To do this we will make sure that the device can be completely controlled remotely because the user will not have access to the microprocessor or any other internal parts other than the charging port. This will ensure that it will be better protected against all forms of water.

6.1.2 Sand and Dust

The system hardware will be exposed to dust, sand, and dirt in various conditions. It was important take into consideration all the elements the system were exposed to. As a piece of football equipment, the device must be well protected at all times. Players falling down, rolling on the ground, or even during storage, the system could be infiltrated by particles. This could cause a short in the circuit or increase the temperatures that the system is able to handle, causing unexpected failures.

The housing of the system must be closed to not allow the entrance of particles. Nonetheless, the housing must also be able to open easily for repair and testing. For testing, the housing was exposed to many particles and then observed how the design keeps particles from penetrating the case.

6.1.3 Shock and Impact

Given that the system would be used in a contact sport, it is necessary that it can hold integrity against an impact. Football players sustain hits and blows to the body multiple times throughout the course of a game. The simplest way to prevent shock or impact damage is to attach the system to either the outside or the inside of the helmet. The inside back or the outside back of the helmet were the best locations, as these are not a usual places of impact. For testing, the system was exposed to high impact forces to see if it can sustain the impact and keep functioning.

6.1.4 Vibration

During impact vibrations also occur. The system was designed in a way such that vibration will not cause any of the components to move, fall, or break. To avoid vibrations from occurring, the device was properly attached to the case and the helmet. Using Velcro, padding, and a base that will attach to the case, lowered the vibration levels that the system will experience upon impact.

6.1.5 Temperature

This element is a big factor in all electronic systems. Athletes will be playing during different times of day and different seasons. Outside or indoors, temperature changes are more than expected and therefore were prepared for. For testing purposes, the system was exposed to both high temperatures and low temperatures. During the test, observations and notes were taken to determine what the limits of the circuit are in terms of when how long the system functioned under these conditions. This might depend on the quality of the components or how the PCB was made. This project was a proof of concept or prototype design, meaning that before the system becomes generally available, it must work in various conditions.

6.1.6 Humidity

Being that this project is to be done in Florida, one of the harshest elements that we must face in an everyday situation would be the constant humidity. This element can cause many problems in hardware. Among the problems that may occur are: physical and chemical deterioration, swelling of the material, loss of physical strength, changes in mechanical properties, electrical shortages, binding parts due to corrosion, oxidation, and loss of plasticity. In order to understand the limitations of the system, testing for humidity took place. Observations on how much moisture that is allowed inside the casing (in levels to cause possible damage) was recorded. In order to reduce these problems we had to reduce the exposure to humidity by using silicon to shield the connections from the moisture. This greatly reduced the amount of damage that humidity can do to our system.

6.1.7 Altitude and Low Pressure

Altitude changes may cause several issues within the system. Given that teams usually travel across the country and play in different locations, indicate that altitude changes will be experienced. In order to test the system's limitations during altitude changes, a chamber test must be completed. Since this project is academic, and at a smaller scale, this was not be a major test subject. Problems that may occur due to altitude changes are rupture or explosion, change in physical or chemical properties, erratic operation or malfunction, overheating, and failure of hermetic seals. If the project is to become generally available, these tests must take place and the limitations of the system must be known.

6.1.8 Accelerometer Testing

Hardware testing of the accelerometer was used to ensure that the ADXL377 is properly calibrated and giving accurate impact readings. Through the use of a pendulum based test rig, meaningful data on the impact force vectors was measured. The test rig was constructed as low cost as possible, with simple parts, to ensure that the project stayed within budget. The design of the rig was based on US Patent 6871525 B2 which describes a pendulum based rig used for the testing of helmet impacts. Figure 6-1 illustrates the basic design and concept of what the test environment looks like.

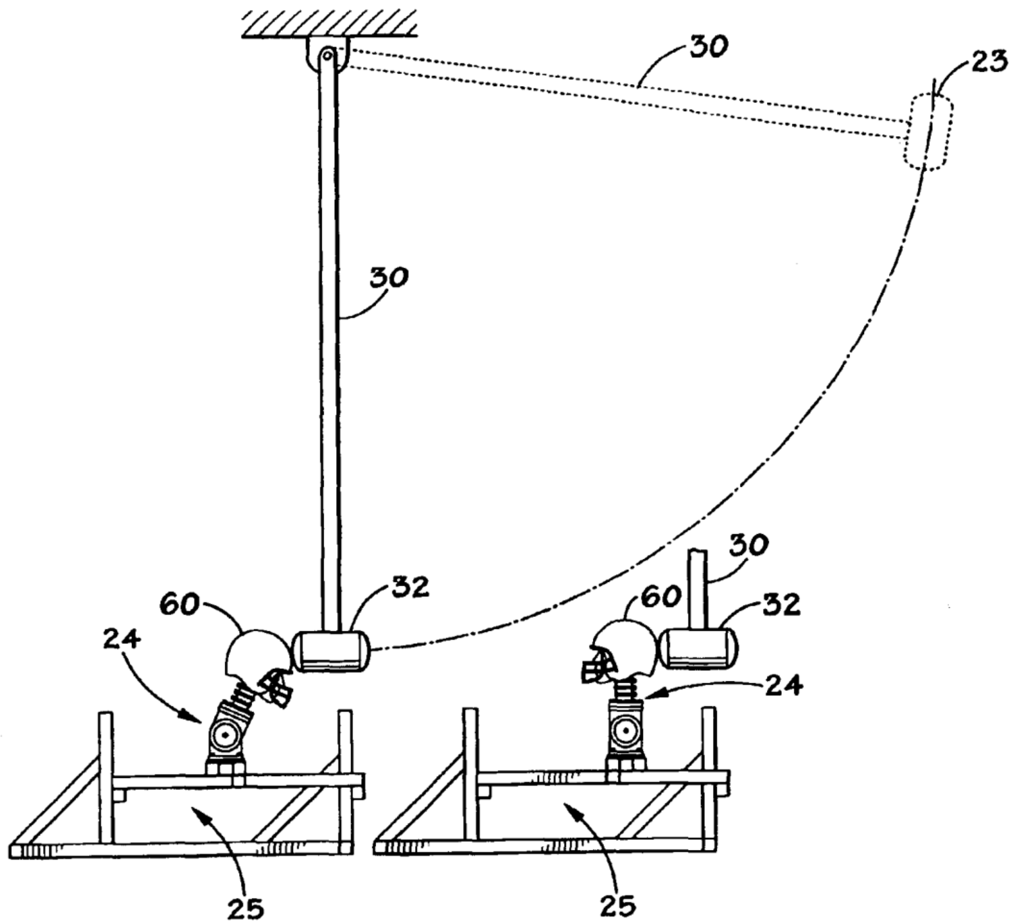


Figure 6-1: Accelerometer Test Environment; A concept diagram describing the test environment and methods for testing the accelerometer sensor; reprinted with permission from Christopher R. P. Withnall and Timothy D. Bayne

As shown in Figure 6-1, the helmet sits at the base attached to a rig to keep it steady. The pendulum swings down and strikes the helmet in a certain position that can be set using the base rig. This was easily be set up using some weights, a ladder, and a stand with some sand bags.

The procedure for the testing was based on the locations described in Figure 3-2. Five impacts to each location were performed and the data the accelerometer senses from the impact was recorded. The accelerometer was calibrated to gravity and using the following pendulum equations in Table 6-1 to determine the force of the impact on the helmet and ensure that the accelerometer was outputting reasonable data.

Description	Equation
Acceleration due to gravity	$g = 9.807\frac{m}{s^2}$ or $g = 32.174\frac{ft}{s^2}$
Length of mass center to the pivot point	$L := 2.0m$
Circumference of the circle	$C := 2 \cdot L \cdot \pi$
Mass of the compound pendulum	$m := 40lbm$
Initial displacement from vertical axis	$\theta_0 := 75deg$
Moment of inertia	$I := m \cdot L^2$
Natural Frequency of the pendulum	$\omega_n := \sqrt{\frac{g}{L}}$
Period of the cycle	$t_p := 2 \cdot \pi \cdot \sqrt{\frac{L}{g}}$
Frequency of the pendulum	$f := \frac{1}{t_p}$ and $\theta := \theta_0 \cdot \cos(\omega_n \cdot t_p)$
Max Velocity reached by pendulum	$V_{max} := \sqrt{\frac{2 \cdot g}{L} \cdot (\cos(0) - \cos(\theta))}$
Torque on the pivot	$\tau := m \cdot g \cdot L \cdot \sin(\theta)$
Angular acceleration	$\omega_d := \omega_n^2 \cdot \sin(\theta)$ and $\alpha := \frac{m \cdot g \cdot L \cdot \sin(\theta)}{I}$
Force acting horizontally	$F_x := m \cdot g \cdot \sin(\theta)$
Distance above rest position on y axis	$\Delta y := L \cdot (1 - \cos(\theta))$

Table 6-1: Pendulum Equations; A table describing the equations used in the accelerometer testing phase

Using this set of equations, the force of impact on the helmet was determined from the length of the pendulum, mass of the pendulum, and initial angle of displacement. Comparing the calculated values versus the actual values picked up by the accelerometer ensured the ADXL377 was working as intended and within a reasonable amount of error.

6.1.9 Battery Testing

One of the most important parts of our project was to ensure that we have constant power to our microcontroller and sensors. To ensure that our device has maximum reliability we needed to test the battery under the stressful conditions of a football game. To test the battery we had to first test the battery outside of the system to see if it can withstand the gravitational force of the impacts listed under the helmet safety requirement section above. We covered the batter in similar material to the helmet and dropped it multiple times while it is in use to see if it can reliably keep everything powered without any drops in current or voltages. Because the device must be able to transport data in real time, and any drops in services might be the second that the player gets injured, causing data not to be recorded during the injury, which in turn defeats the purpose of the entire project.

6.2 Software Test Environment

The following two subsections discuss the software test environment for the BIO-Helmet system. These testing environment discussions are split into the software which will run on the embedded device and the software which will run on the local server.

6.2.1 Embedded Software Test Environment

The embedded software testing environment consisted of the final PCB board with a USB UART for connecting to an instance of Energia. This UART was used for loading data onto the ROM of the microcontroller for testing. The accelerometer sensor was wired to the board as well. For preliminary testing, before the final PCB is delivered, a Tiva C LaunchPad was used with a bread board. Printed console outputs and LEDs were used to indicate proper inputs and power.

6.2.2 Local Server Software Test Environment

The local server software test environment existed largely independent from the BIO-Helmet embedded software test environment. This testing environment consisted of a standard operating system personal computer or laptop capable of running Python. The Python code (especially that of the GUI) was tested and debugged as soon as it was implemented. The SQLite database was configured and populated with test data to test the Python data reporting script as well as the customized MATLAB environment. The SQLite database contained manually entered data for testing. Once the embedded software test environment was operational, the Python data reporting script was included in the local server software test environment. This final test environment contained every aspect available for a full software test of the BIO-Helmet system. This final test environment was able to test senor data inputs, sending sensor data, receiving data at the local server, entering the data into a database, and running reporting software on the sensor data.

6.3 Software Specific Testing

Testing the software for our device was also as important as hardware because we wish our device to follow all specified guidelines that we labeled above. We ensured that our device’s software was reliable and as user friendly as possible. The following two subsections discuss software specific testing of the BIO-Helmet system. The BIO-Helmet software specific testing was divided into two subsections, the software specific testing for the embedded program and the software specific testing for the various Python based programs that run on the local server.

6.3.1 Embedded Software Specific Testing

Testing for the embedded software first commenced by testing a basic Energia sketch loaded onto the MCU. This program ensured that all pins are correctly set to an input and that the debug UART interface is properly configured. The next phase of this software testing involved the accelerometer. A debug program was written that polls the sensors periodically and prints the read input to the console screen. The accelerometer was then moved and the testing user observed the changes in the output voltage. This ensured that the configuration settings for the accelerometer input pins were correct, the DSP circuit was behaving as expected, and the software calculations for normalizing the accelerometer sensor data were working correctly. The high impact flag was also tested during this phase of testing by moving the accelerometer in such a way that triggered the high impact flag. When this high impact flag was triggered, the debug program printed a status message to the console.

The next phase of the embedded software test was the EEG sensor testing. The EGG sensors were be placed on a team member's head and the sensors were polled in the same manner mentioned in Section 4.7.2. The data read in from the input pins connected to the EE sensor array output the two brain wave EGG channels to the debug console. The amount of read and analyzed was the alpha and beta waves for the final prototype of the BIO-Helmet.

The last phase of the embedded software specific testing was the wireless communication portion of the BIO-Helmet. At this stage, the final embedded program was ready for full testing as all previous software parts had been fully tested and integrated. This wireless program made a connection to the local server and sent the data across the wireless direct connection. A simple debug receiving script was created on the local server to print the raw input received from the BIO-Helmet to the Python console window. This confirmed that the communication channel between the BIO-Helmet and the local server was working and data was able to be sent across the wireless link. This concludes the embedded software specific testing.

6.3.2 Local Server Software Specific Testing

Testing for the local server specific testing first commenced with the Python data receiving script. As described in the last portion of Section 6.3.1, a debug version of the script was developed that printed the receiving input to the Python console window. This confirmed that the established TCP 5999 socket was configured correctly and the local server was able to receive data from the BIO-Helmet. The next portion of the Python data receiving script to be tested was the SQLite database insert. The database was cleared (if populated during previous testing or setup) and the necessary database connection code was added to the Python data receiving script. The BIO-Helmet then began sending sensor data to the local server. A database dump was then be performed after collecting sensor data for a period of five minutes. This test confirmed that the Python data receiving script was able to receive data from the BIO-Helmet and enter that information into the database. The high impact flag was also tested here to ensure that the Python data receiving script properly set the flag and inserted this set Boolean into the database.

The next phase of the local server specific testing was the Python data reporting script. This script dumps information from the SQLite database to a MATLAB compatible text file that was used to provide the accelerometer and EEG sensor in graphical format. This module was tested both on manually inserted database information (for testing purposes) or actual sensor data obtained from the previous testing. This test confirmed that the database was being populated as well as the Python script was properly writing the data to a text file.

The last module to be tested in the local server software specific testing was the reporting suite. The graphical user frontend basic functionality was tested without any dependence on any other modules. The notification window was tested by triggering the high impact notification on the BIO-Helmet. This test was performed in a manner similar to the method mentioned when testing the high impact on the embedded specific software testing. The BIO-Helmet connection status table was confirmed at the point of testing the Python data receiving script. Lastly, the graphical user frontend was tested that it properly displays the accelerometer sensor data in tabular format. This functionality was tested with manually entered sensor data as well as actual data obtained through previous testing phases. The last portion of the reporting suite included testing the basic functionality of the MATLAB environment. This includes the custom graphs created for the accelerometer data as well as the EGG data within the MATLAB environment. The text file read portions and graphical figure displays were tested during this phase.

7 PROJECT OPERATION

This section focuses on the operation of the BIO-Helmet. This section includes a short user manual describing the usability of the BIO-Helmet and the correct operation of all functions. A short troubleshooting guide is also included for resolving issues with the BIO-Helmet hardware or software compatibility. Figure 7-1 below shows an external view of the assembled BIO-Helmet prototype.

Figure 7-1: Assembled BIO-Helmet Prototype; An image showing the external view of the assembled BIO-Helmet prototype

7.1 HARDWARE OPERATION

This section covers the hardware operation of the BIO-Helmet. This hardware section includes the power delivery system, the EEG sensor array, the accelerometer sensor, the Tiva C MCU, and the Wi-Fi system.

7.1.1 Power System

The BIO-Helmet prototype is powered by a USB power pack. This power pack requires recharging and can be detached from the BIO-Helmet Velcro for charging. This pack can be plugged into any standard USB ports that supplied five volts with the included low profile USB cable. The power pack has status LEDs on the side to indicate the charge level. The output port of the power pack must be connected to the black USB cable which leads to the Tiva C MCU. The MCU is then used to power all other BIO-Helmet systems.

7.1.2 EEG Sensors

There are two sets of EEG electrodes. The red and yellow EEG probes correspond to channel one and the blue and purple probes correspond to channel two. The EEG cap should be placed on the wearing user's head for probe placement and stability. For added accuracy, EGG conductive gel can be applied to the probe points of the subject's head. The red and yellow probes should be placed in the cap cutouts above the user's eyes. The blue and purple leads should be placed in the two cap cutouts above the frontal lobe. These leads connect to the EEG processing board which is then read by the MCU after amplification and filtering. Once the EEG cap and probes are secured, the helmet can be placed on the wearing users head. Figure 7-2 below shows the location of the EEG DSP printed circuit board and the electrode leads.

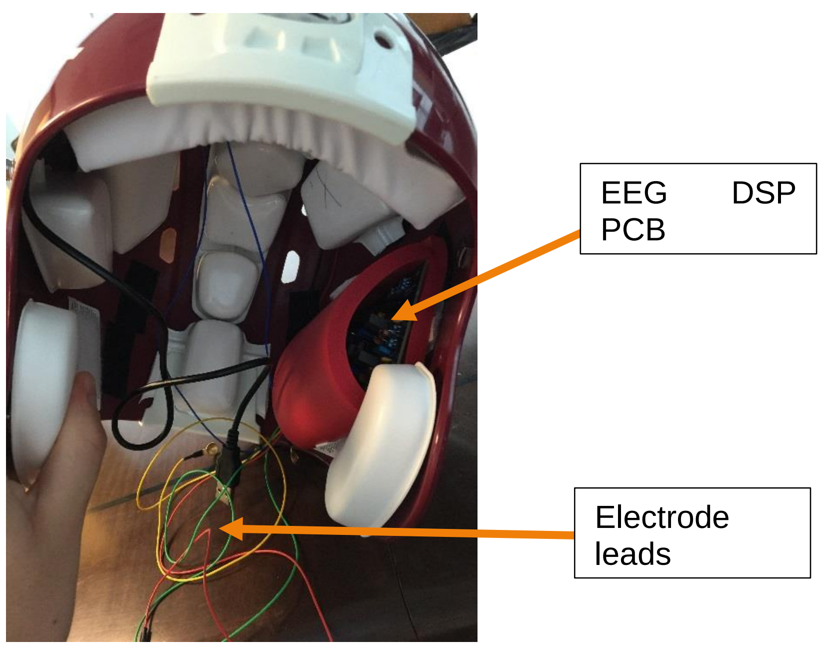


Figure 7-2: BIO-Helmet EEG; An image showing the location of the EEG DSP PCB and electrode leads

7.1.3 Accelerometer Sensor

The accelerometer sensor is mounted within the helmet's foam padding at the top of the helmet. This sensor is permanently soldered to wire leads connecting to the MCU. There is no user operation necessary for this component and no status LEDs or other configuration options are needed. Figure 7-3 below shows the location of the accelerometer sensor.

7.1.4 Tiva C MCU

The Tiva C MCU receives sensor inputs from the accelerometer and EEG sensors, packages, and sends this data via the Wi-Fi module to the local server for analysis. The Tiva C MCU contains the power input mini USB input to power all BIO-Helmet from the battery. The MCU has a green status LED to indicate that power is being delivered to the MCU. The Tiva C LaunchPad contains two button, however the BIO-Helmet embedded program makes no use of these buttons. There is no user operation necessary for this component and no configuration options are needed. Figure 7-3 below shows the location of the Tiva C MCU.

7.1.5 CC3100 Wi-Fi Module

The CC3100 Wi-Fi module is responsible for the wireless connection between the BIO-Helmet and the local server. This component is attached to the Tiva C LaunchPad breakout pins for booster packs. This component can be detached as necessary, but should be left in place for normal operation. The CC3100 includes three status LEDs and no buttons. The red LED indicates that power is connected to the Wi-Fi module. The yellow LED indicates that the device is functional and the green LED indicates that the device is not in hibernation mode. There is no user operation necessary for this components and no configuration options are needed. Figure 7-3 below shows the location of the CC3100 Wi-Fi module.

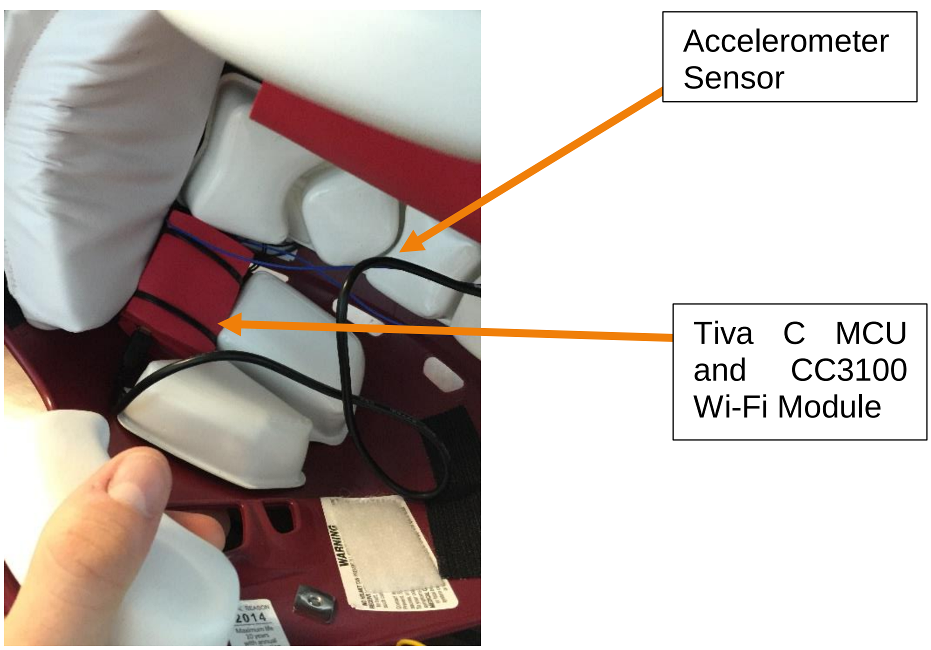


Figure 7-3: BIO-Helmet MCU, Wi-Fi, and accelerometer; An image showing the placement of the Tiva C MCU, the Wi-Fi module and the accelerometer sensor mounted inside the BIO-Helmet

7.2 SOFTWARE OPERATION

This section of the BIO-Helmet operation guide focuses on the software operation of the BIO-Helmet. This operation guide will only focus on the local server software as the embedded program is directly written to the MCU and should only needed to be modifying for troubleshooting purposes. This operation guide is divided into five sections, one for each major step of configuring the BIO-Helmet server. Note that all BIO-Helmet source code and the database file (bio-helmet.db) must all be placed within the same directory.

7.2.1 Wi-Fi Connection

Once the BIO-Helmet is powered on, it will begin creating a Wi-Fi network named "BIO-Helmet", the local server must first connect to this network using the password "12345678". This password should be changed later in the embedded code once the BIO-Helmet is configured and running. The local server will automatically receive an IP address from the BIO-Helmet and will the BIO-Helmet Wi-Fi module will begin trying to connect to a TCP socket on the local server.

7.2.2 Python Data Receiving Script

The first script that must be started is the BIO-Helmet data receiving script. This script listens to communications from the BIO-Helmet over the opened TCP 5999 socket. To start this script, execute, at a terminal window, "python BIO-Helmet_Server.py". Once the BIO-Helmet connects to the socket, the script will print that it has received a connection from the BIO-Helmet and the BIO-Helmet's IP address. This message indicates that the BIO-Helmet is connected to the local server and is sending sensor data over the wireless communication channel. To exit the script, press ctrl+c on the keyboard. This stroke closes the TCP socket and database connections and exits the script.

7.2.3 Python Graphical User Interface

This script creates and runs a GUI environment for sensor data reporting for the BIO-Helmet. This script can be opened by executing, at a terminal window, "python BIO-Helmet_GUI.py". The GUI is composed of several elements, the Home button returns the user to the home page, the Quit button exits the script, the Raw Sensor Data button takes the user to a table showing the raw sensor output, and the MATLAB button launched MATLAB. Figure 4-44 and Figure 4-46 show screenshots of the BIO-Helmet graphical user interface. The status box on the home page indicates various status metrics corresponding to the wireless connectivity of the BIO-Helmet. The High Impact Notification table will populate with each high impact event triggered by the BIO-Helmet. The GUI also has support for a high impact notification. If the BIO-Helmet sets the high impact flag for an event, the GUI will open a popup window informing the monitoring user that a high impact event has occurred. This popup window can be dismissed by pressing the "Ok" button on the popup.

7.2.4 Python Data Reporting Script

This script is used to dump the SQLite database containing all of sensor data to a MATLAB compatible text file. This text file is then referenced by the customized

MATLAB environment. Operation of this environment is covered in Section 7.2.5. This script automatically dumps the database to new text file once every second. Once started, it will continually create a new text file called MATLABData.dat. No user modification of this text file should occur as the formatting must be preserved for MATLAB to properly read in the sensor data. This script can be started by executing, from a terminal window, “python BIO-Helmet_Reporting.py”. This script can be stopped by pressing ctrl+c on the keyboard. This stroke closes all database connections and exits the script.

7.2.5 MATLAB

The MATLAB environment is contained within a MATLAB script named ReportingMATLAB.m. Depending on the operating system being used, and the other programs installed on the local server, the monitoring user may simply double click on this script to open it within MATLAB. If the script does not open by double clicking, the monitoring user may need to manually open MATLAB first and then open the .m script file. Once the script is open, click the run button and MATLAB will read the MATLABData.dat file and open a figure window which displays the accelerometer g-force values and both EEG wave outputs in graphical format. The script can be closed by closing the figure window and run again, with the most up to date data file, by simply clicking the run button again.

7.3 TROUBLESHOOTING GUIDE

This sections includes a troubleshooting guide for the BIO-Helmet. This guide is split up into hardware and software troubleshooting sections.

7.3.1 Hardware Troubleshooting

This section includes a basic troubleshooting guide for the BIO-Helmet hardware system. The table below includes the symptoms of a possible failure, the cause, and the action to fix that particular hardware related issue.

SYMPTOM	CAUSE	ACTION
No Wi-Fi Network is created	Power Unplugged	Ensure power is plugged into the BIO-Helmet
	Battery has no charge	Charge battery
	Connection between Wi-Fi and MCU has shifted	Ensure a solid connection between the Wi-Fi module and the MCU
No status LEDs displayed on MCU or Wi-Fi Module	Power Unplugged	Ensure power is plugged into the BIO-Helmet
	Battery has no charge	Charge battery
EEG data is invalid or inaccurate	EEG probes are improperly placed	Place the EEG probes in the cap as mentioned in Section 7.1.2

Table 7-1: Hardware Troubleshooting Guide; A table describing how to identify and correct several BIO-Helmet hardware issues

7.3.2 Software Troubleshooting

This section includes a basic troubleshooting guide for the BIO-Helmet software system. The table below includes the symptoms of a possible failure, the cause, and the action to fix that particular software related issue.

SYMPTOM	CAUSE	ACTION
Python Data Receiving Script does not have a connection to the BIO-Helmet	Power Unplugged	Ensure power is plugged into the BIO-Helmet
	Software Firewall	Disable any software firewalls running on the local server
	Range	Move the BIO-Helmet and the local server closer together
Python Data Receiving reports data type error	Intermittent failure due to speed of sensor input	Power cycle the BIO-Helmet; no need to restart the Data Receiving Script.
Database connection error reported by any of the Python reporting scripts	bio-helmet.db file missing or in improper directory	Move the bio-helmet.db file to the same directory as all reporting scripts.
BIO-Helmet GUI does not show notification popup	Python Data Receiving Script is not running	Ensure the Python Data Receiving Script is running and is not reporting any errors.
BIO-Helmet GUI does not update sensor data table when refreshed		
MATLAB graphs do not load	Empty Database	Ensure the database has been populated with some sensor data from the BIO-Helmet
MATLAB script performance is slow	Local server does not meet MATLAB system requirements	Ensure the local server meets or exceeded the system requirements to run MATLAB
MATLAB does not load the MATLABData.dat file	The file has not been created or deleted	Run the Python Data Reporting Script to generate a new copy of the file in the same directory.

Table 7-2: Software Troubleshooting Guide; A table describing how to identify and correct several BIO-Helmet software issues

8 ADMINISTRATIVE CONTENT

8.1 MILESTONE DISCUSSION

The following discussion diagrams mention milestone goals for the BIO-Helmet project for Senior Design I and Senior Design II. Figure 8-1 below demonstrates the planned and dated milestones for Senior Design I. Figure 8-2 below demonstrates the planned and dated milestones for Senior Design II.

8.1.1 Past Milestones

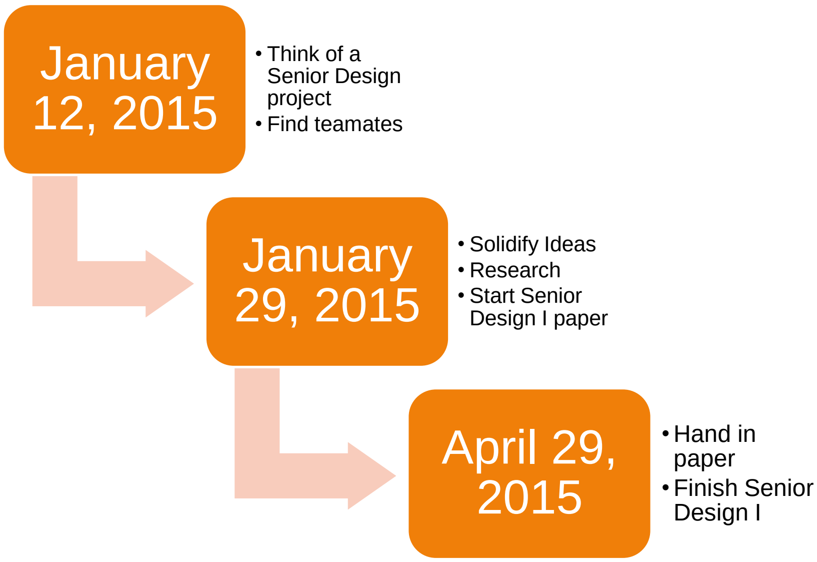


Figure 8-1: Senior Design I Milestones; A figure discussing the past milestones for the BIO-Helmet project in Senior Design I

8.1.2 Present Milestones

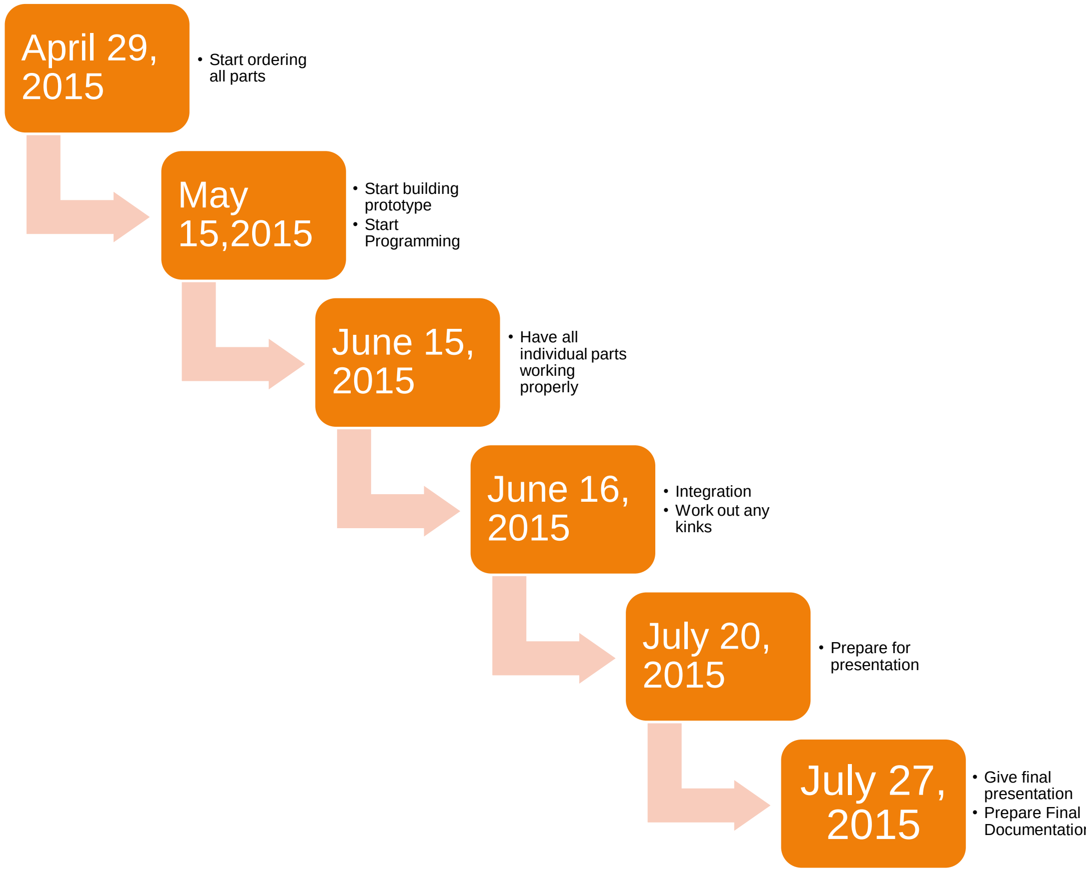


Figure 8-2: Senior Design II Milestones; A figure discussing the current milestones for the BIO-Helmet project in Senior Design II

8.2 Budget and Finance Discussion

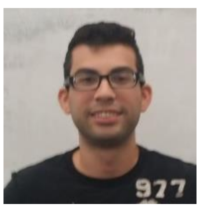
As a college student senior design project, with a fixed budget, the funding for this project primarily came from sponsorship from Boeing. $600 dollars was obtained as sponsorship from Boeing. This project's budget worked to find ways to incorporate this and not go over budget. The original budget estimations for building such an elaborate project were in the order of $1,500. To trim the costs down, a lot of the materials were built by the project team in order to stay within budget. The main costs for this project came from the EEG device that is required to obtain good reliable readings. The device previously considered was approximately $900; thus this idea was scrapped and it was decided that a custom built device would better suite this project's budget constraints. Doing so reduced the costs of the EEG sensor array considerably. The project was also enrolled in the TI project giveaway. Texas Instruments made available $200 worth of TI products towards use in a senior design project. Since TI has such a huge array of products, a number of TI products have been used in this project, in order to stay with one manufacturer and keep project costs as low as possible. Any and all expenditures over the $600 provided by Boeing and $200 from TI were split amongst the team members equally. With much research into all the parts that was required, it has been found that an evaluation Tiva C can be obtained in Engineering 2 for free (of course it is asked that once preliminary testing is complete, it be returned so that someone else can get a chance to use it) and almost all other integrated circuit hardware portions of this project were bought from Texas Instruments. This budget also accounts for the possibility that we may have broken some of our parts during experimenting with them in order to get the best performance that we can achieve from the BIO-Helmet.

8.3 PROJECT PERSONNEL

Frank Alexin – EE

Frank Alexin is a graduating Electrical Engineering student. He hopes to pursue a career in the Orlando area in micro-electronics after graduating. His strengths are in circuit design and programming in regard to embedded systems.

Nicholas Dijkhoffz– EE



Nicholas Dijkhoffz is a graduating Electrical Engineering student. Nicholas has accepted a job offer with Honda Manufacturing of Alabama and will relocate after summer. In the future he would like to work with technology that improves quality of life for people.

Adam Hollifield – CPE

Adam is a graduating Computer Engineering student. His areas of interest include enterprise networking and cyber security. He has accepted a full time position as a Network Engineer at Tech Data Corporation.

Mark Le – EE

Mark Le is a graduating Electrical Engineer. He plans to work as an engineer in the Air Force working on sensory applications on aircrafts and vehicles. He hopes that one day his findings will lead to safer conditions in civilian vehicles.

9 PROJECT SUMMARY AND CONCLUSIONS

The goal of this project was to develop a sports safety device that can be used to help lessen the dangerous effects of concussions on athletes in contact sports. This project satisfies this goal, specifically with the high impact warning and historical data reporting of force data. Another goal of this project is to conduct additional research into possible correlations between concussions symptoms and observed brain waves. This goal was satisfied by creating a prototype that can present data to a neurological research teach for later analysis and research. Each team member, throughout the completion of this project, was able to use each of their areas of expertise and learn about new interests. This project was concluded with the presentation of a functioning hardware and software BIO-Helmet prototype.

APPENDIX A – COPYRIGHT PERMISSIONS

Figure 2-1:

All Clipart are Released into the Public Domain.

Each artist at Openclipart releases all rights to the images they share at Openclipart. The reason is so that there is no friction in using and sharing images authors make available at this website so that each artist might also receive the same benefit in using other artists clipart totally for any possible reason.

Figure 3-1:

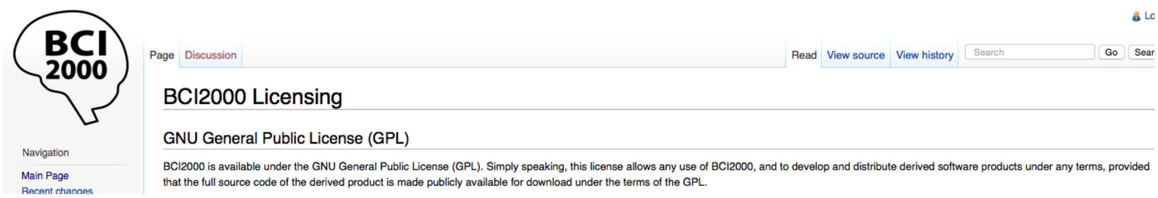

Help Shape The Future of OpenBCI Fill Out Our Survey!

Copyright Notice

Editorial contents of the OpenBCI.com website, such as texts and photos, are released as Creative Commons Attribution ShareAlike 3.0.

This means you can use them on your own derived works, in part or completely, as long as you also adopt the same license. You find the complete text of the license here. OpenBCI brand, OpenBCI logo, design of the website and design of the boards are copyright of OpenBCI and cannot be used without formal permission. For informations about the right way to use them, please write to contact@openbci.com

Figure 3-2:



BCI2000 Licensing

GNU General Public License (GPL)

BCI2000 is available under the GNU General Public License (GPL). Simply speaking, this license allows any use of BCI2000, and to develop and distribute derived software products under any terms, provided that the full source code of the derived product is made publicly available for download under the terms of the GPL.

Figure 3-3, Figure 3-4, Figure 3-5, and Figure 3-9:

Sonja Lujan <sales@jarisupply.com>
Fri 4/24/2015 12:57 PM

Nicholas,

You have our permission to use the images of our disc electrodes referred to in your email request. Good luck with your design project.

Richard Kaiser, president
Chalgren Enterprises, Inc
dba Jari Electrode Supply

REPLY REPLY ALL FORWARD ...

Nicholas Dijkhoffz
Fri 4/24/2015 11:54 AM
Sent Items

Mark as unread

To: sales@jarisupply.com;

To whom it may concern,

I would like to reuse the Non-Sterile Reusable Disc Electrodes with PVC Leadwires images (found on http://www.jarisupply.com/non-sterile-reusable-disc-electrodes-with-pvc-leadwires.html for a design report for an undergraduate Electrical/Computer Engineering Senior Design project. I would like to use these images to show some examples of reusable disc electrode sensors. This project requires that all reprints of any images or charts be documented with copyright approval from the holder unless specified as free to use in the Terms and Conditions of the particular website or document.

Thank you,

Nicholas Dijkhoffz

Figure 3-6, Figure 3-7, and Figure 3-8:

Default site-wide license

The intention of having a default site-wide license is so that general text and graphics on the web-site should be 'free' (in the GNU sense) so that they can be used and modified by others in the group and elsewhere, and so that no-one has to worry about using or modifying previous contributors' work even if they temporarily disappear or leave the project.

Figure 3-10, Figure 3-11, Figure 3-12, and Figure 4-49:

Open source policy

1. EEGLAB is distributed under the GPL GNU license, which states that the software cannot be modified for commercial purposes. Any contributed functions we add to EEGLAB will be made available for free non-commercial use under this license.
2. We will credit your authorship of the functions on the function help pages. The authors will retain all commercial rights to the functions.

Figure 3-13:

The Processing development environment is released as open source under the GPL. The export libraries (also known as 'core') are released under an LGPL license, which means they can be used as a library and included in your project without you having to open up your code (though we encourage people to share anyway). But if you make changes to core, you have to submit back to us. More information about the GNU Public License can be found at 1.

Figure 3-14:

NCBI

Copyright and Disclaimers

PubMed | Entrez | BLAST | OMIM | Taxonomy | Structure

NCBI

SITE MAP

Copyright Status

Information that is created by or for the US government on this site is within the public domain. Public domain information on the National Library of Medicine (NLM) Web pages may be freely distributed and copied. However, it is requested that in any subsequent use of this work, NLM be given appropriate acknowledgment.

Figure 3-15:

Summary [edit]

Description	**English:** A rather poor diagram of inductive charging.
Date	16 April 2010
Source	Own work
Author	Egmason

Figure 3-16:

Licensing [edit]

I, the creator of this work, hereby grant the permission to copy, distribute and/or modify this document under the terms of the **GNU Free Documentation License**, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. Subject to disclaimers.

SOME RIGHTS RESERVED

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 License.

This licensing tag was added to this file as part of the GFDL licensing update.

Figure 3-17:

Date	23 November 2004, 11:15:37
Source	Flickr
Author	Steve Jurvetson from Menlo Park, USA
Permission **(Reusing this file)**	This file is licensed under the Creative Commons Attribution 2.0 Generic license. You are free: • **to share** – to copy, distribute and transmit the work • **to remix** – to adapt the work Under the following conditions: • **attribution** – You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work). This image was originally posted to Flickr by jurvetson at http://flickr.com/photos/44124348109@N01/2265519. It was reviewed on 7 April 2007 by the FlickreviewR robot and was confirmed to be licensed under the terms of the cc-by-2.0.
Reviewer	Andre Engels

Figure 3-18:

Description	**en:** This picture shows (left) a printed circuit board (PCB) layout, created on a computer and (right) the manufactured board, populated with parts. Both through-hole and surface mounts components have been used; the board is double-sided, with through-hole plating, green solder resist, white solder paste (visible on unsoldered holes, such as the four corner mounting holes and the nine holes down the right hand side of the board) and white silkscreen printing. **it:** Progetto di un circuito stampato a sinistra e realizzazione a destra. **de:** Links das im CAD erstellte Layout einer Leiterplatte, in dem die verschiedenen Lagen mit unterschiedlichen Farben dargestellt sind. Rechts die daraus hergestellte und mit Bauelementen bestückte Leiterplatte. **es:** A la izquierda la imagen de la PCB diseñada por ordenador y a la derecha la PCB manufacturada y montada. **sv:** Datordesign av ett mönsterkort (vänster) och det färdiga kretskortet (höger) **zh:** 印刷電路板 **hu:** A baloldalon egy NYÁK CAD program segítségével tervezett rajza, a jobboldalon pedig a rajznak megelelően elkészített panel látható. **id:** Foto dari desain sirkuit dan realisasinya **ko:** 컴퓨터로 설계된 인쇄회로기판 (왼쪽) 및 기판과 부품이식된 실장 (오른쪽). 기판은 홀로 통하는 도금, 녹색 땝납 저항, 흰 땝납 페이스트, 흰 실크스크린 프린트가 있는, 양면이다. 양쪽 표면 실장과 홀로 통하는 부품은 사용되었다.
Date	2 February 2007
Source	Photographed by User:Mike1024
Author	User Mike1024
Permission **(Reusing this file)**	I, the copyright holder of this work, release this work into the **public domain**. This applies worldwide. In some countries this may not be legally possible; if so: *I grant anyone the right to use this work **for any purpose**, without any conditions, unless such conditions are required by law.*

Figure 3-19:

Description	표면 실장 기술
Date	7 December 2004
Source	en:Image:Smt closeup.jpg
Author	en:User:John Fader

Licensing [edit]

Permission is granted to copy, distribute and/or modify this document under the terms of the **GNU Free Documentation License**, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled *GNU Free Documentation License*.

This file is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license.

You are free:

- **to share** – to copy, distribute and transmit the work
- **to remix** – to adapt the work

Under the following conditions:

- **attribution** – You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **share alike** – If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

Description	A MOS 6581 sound chip from a Commodore 64 main board
Date	29 December 2006
Source	Own work
Author	Christian Taube

Licensing [edit]

I, the copyright holder of this work, hereby publish it under the following license:

This file is licensed under the Creative Commons Attribution-Share Alike 2.5 Generic license.

You are free:

- **to share** – to copy, distribute and transmit the work
- **to remix** – to adapt the work

Under the following conditions:

- **attribution** – You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **share alike** – If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

Figure 3-20:

Mark Le
To: Adam Hollifield
Fwd: Re: General Questions

Today at 4:50 PM

ML

---------- Forwarded message ----------
From: Chow He <info@electrodragon.com>
Date: Apr 26, 2015 11:32 PM
Subject: Re: General Questions
To: Mark Le <flafner@knights.ucf.edu>
Cc:

Hello Mark,

Yes, it is no problem, you can use it.

Best regards,

On Sun, Apr 26, 2015 at 5:18 PM, WordPress <www-data@www.electrodragon.com> wrote:

> From: Mark Le
> Subject: General Questions
>
> Message Body:
> Hello I am a UCF student and wish to use the Wi05 module inside of my senior design project. May I use your pictures inside of my documentation? I will cite you as the source and will not alter or change the picture.
>
> Thanks for your time.
>
> --
> This mail is sent via contact form on ElectroDragon http://www.electrodragon.com

--
Kind regards,
Chao

Electrodragon Team
E: info@electrodragon.com
Web: www.electrodragon.com
Store: Store.Electrodragon.com
Blog: Blog.Electrodragon.com
Twitter I Facebook I Google+

Figure 3-21:

Mark Le
To: Adam Hollifield
Fwd: [Spark] Re: Olark chat with flafner@knights.ucf.edu

Today at 4:50 PM

ML

---------- Forwarded message ----------
From: "The Spark Support Team (Spark)" <support@spark.zendesk.com>
Date: Apr 27, 2015 3:13 PM
Subject: [Spark] Re: Olark chat with flafner@knights.ucf.edu
To: Mark le <flafner@knights.ucf.edu>
Cc:

##- Please type your reply above this line -##

Your request (17629) has received a response.

To continue this dialogue, please reply to this email.

The Spark Support Team (Spark)
Apr 27, 09:39

Hi there,

Of course it is! Spark is an open-source company, so anything you see that's public, it's free to use. Thank you for asking though, much appreciated. What is your project on? We'd be interested to hear about it.

Regards,
Kevin @ Spark

Some Super Spark Resources
Documentation of both hardware and software: docs.spark.io
Internal and external resources: www.spark.io/resources
FAQs and troubleshooting guide: www.spark.io/support
Forum full of amazing community members: community.spark.io

Mark le
Apr 26, 02:49

Location: USA (Orlando, FL)
Referred from: http://blog.spark.io/2014/11/12/introducing-the-19-dollar-photon/
On page: https://store.spark.io/?utm_source=SparkBlog&utm_medium=blog&utm_term=PreOrder&utm_content=Button&utm_campaign=BlogButton
IP address: 108.81.234.51
Hello I am a UCF student and wish to use your Photon inside of my senior design project. May we utilize some of your pictures to show the device in our documentation? We will not alter the pictures and we will cite you as the source and give you credit for it, we just want to use the pictures to show how it will be connected and how small it is. Thanks

This email is a service from Spark. Delivered by Zendesk.

Figure 3-22 (http://pythonprogramming.net/how-to-embed-matplotlib-graph-tkinter-gui/):

Harrison Kinsley Yesterday at 9:51 PM HK
To: Adam Hollifield
Re: Image Reuse Permissions

If you document the what page you got the images from, you're free to use them.

-Harrison

On Thu, Mar 26, 2015 at 3:07 PM, ahollifield <ahollifield@knights.ucf.edu> wrote:

> To whom it may concern,
>
> I would like to reuse the Sea of BTC Client Graph Page images (found on http://pythonprogramming.net/how-to-embed-matplotlib-graph-tkinter-gui/) a design report for an undergraduate Electrical/Computer Engineering Senior Design project. I would like to use these images to show some examples of how graphs are generated with Tkinter. This project requires that all reprints of any images or charts be documented with copyright approval from the holder unless specified as free to use in the Terms and Conditions of the particular website or document.
>
> Thank you,
>
> Adam

Figure 3-23:

Description	**English:** Encapsulation of application data descending through the layered IP architecture
Date	July 14, 2008
Source	Original artwork by en:User:Cburnett
Author	en:User:Cburnett original work, colorization by en:User:Kbrose
Permission (Reusing this file)	GFDL
Other versions	[edit]

English

French

Licensing [edit]

Figure 3-24:

Licensing [edit]

Figure 4-17, Figure 4-18, and Figure 4-20:

Proprietary Rights And Use Restrictions

Figure 4-7, Figure 4-22, Figure 4-23, Figure 4-24, and Figure 4-25, Figure 4-26:

Figure 4-28:

Figure 6-1:

Patent Information

Appendix B – References

About SQLite. (n.d.). Retrieved March 27, 2015, from https://sqlite.org/about.html

Arduino - ArduinoBoardUno. (n.d.). Retrieved April 8, 2015, from http://www.arduino.cc/en/Main/arduinoBoardUno

Chao. (2015, January 12). WI05 Wifi Module. Retrieved April 27, 2015, from http://www.electrodragon.com/w/WI05_Wifi_Module

Cobb, B. (2013). Measuring Head Impact Exposure and Mild Traumatic Brain Injury in Humans. Brain Injuries and Biomechanics. Retrieved April 8, 2015, from https://vtechworks.lib.vt.edu/bitstream/handle/10919/23815/Cobb.pdf?sequence=1

Greenwald, R., Gwin, J., Chu, J., & Crisco, J. (2009, December 9). Head Impact Severity Measures for Evaluating Mild Traumatic Brain Injury Risk Exposure. Retrieved April 8, 2015, from http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2790598/

Griffiths, D., Nelo, Peters, J., Robinson, A., Spaar, J., & Vilnai, Y. (2003). The ModularEEG Design. Retrieved April 27, 2015, from http://openeeg.sourceforge.net/doc/modeeg/modeeg_design.html

Grubb, J. (2013, May 17). Ignored By Dinosaurs. Retrieved March 27, 2015, from http://www.ignoredbydinosaurs.com/2013/05/explaining-non-relational-databases-my-mom

Football Helmet Standards Overview. (2015). Retrieved April 27, 2015, from http://nocsae.org/wp-content/uploads/2011/10/NOCSAE-Football-Helmet-Standards-Overview-1-2015.pdf

IEEE Standards Association. (2015). Retrieved April 27, 2015, from http://standards.ieee.org/

Lemburg, M. (n.d.). Welcome to Python.org. Retrieved April 11, 2015, from https://www.python.org/dev/peps/pep-0249/

Microcontrollers (MCU). (2014). Retrieved April 28, 2015, from http://www.ti.com/lsds/ti/microcontrollers_16-bit_32-bit/c2000_performance/control_automation/tm4c12x/overview.page?DCMP=tivac-series&HQS=tiva-c

Mirza, U. (2011, August 29). Types of Batteries and Their Applications (K. Sleight, Ed.). Retrieved April 8, 2015, from http://www.brighthubengineering.com/power-generation-distribution/123909-types-of-batteries-and-their-applications/

MySQL Editions. (n.d.). Retrieved March 27, 2015, from https://www.mysql.com/products/

NFL, ex-players agree to $765M settlement in concussions suit. (2013, August 29). Retrieved March 27, 2015, from http://www.nfl.com/news/story/0ap1000000235494/article/nfl-explayers-agree-to-765m-settlement-in-concussions-suit

Shop Riddell. (n.d.). Retrieved April 8, 2015, from http://www.riddell.com/innovation/

Supalla, Z. (2014, November 12). Introducing the $19 Photon. Retrieved April 27, 2015, from http://blog.spark.io/2014/11/12/introducing-the-19-dollar-photon/

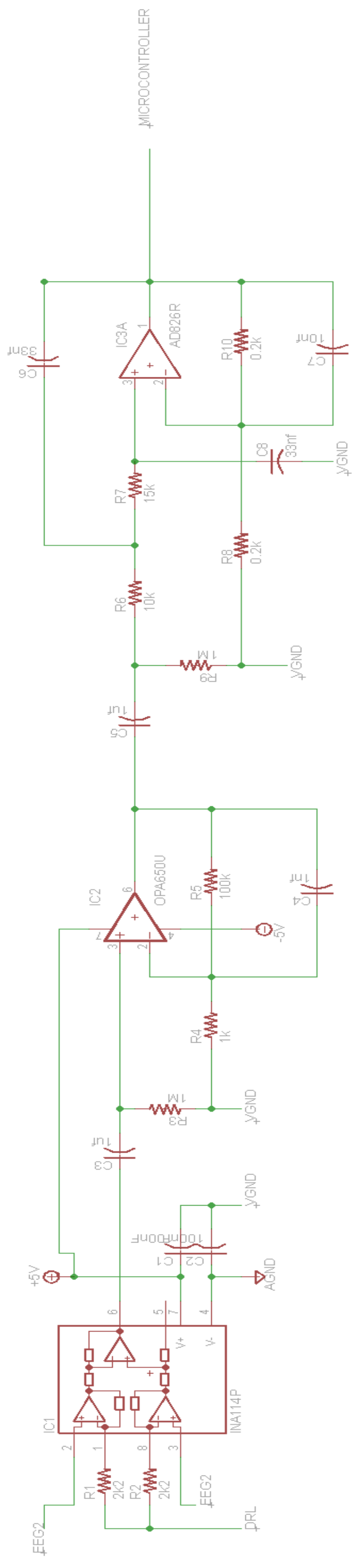
Texas Instruments. (2014, March 6). Tiva C Series Connected LaunchPad Board Tour [Video file]. Retrieved from https://www.youtube.com/watch?v=96SwKRpd1uM

What are relational databases? - HowStuffWorks. (n.d.). Retrieved March 27, 2015, from http://computer.howstuffworks.com/question599.htm

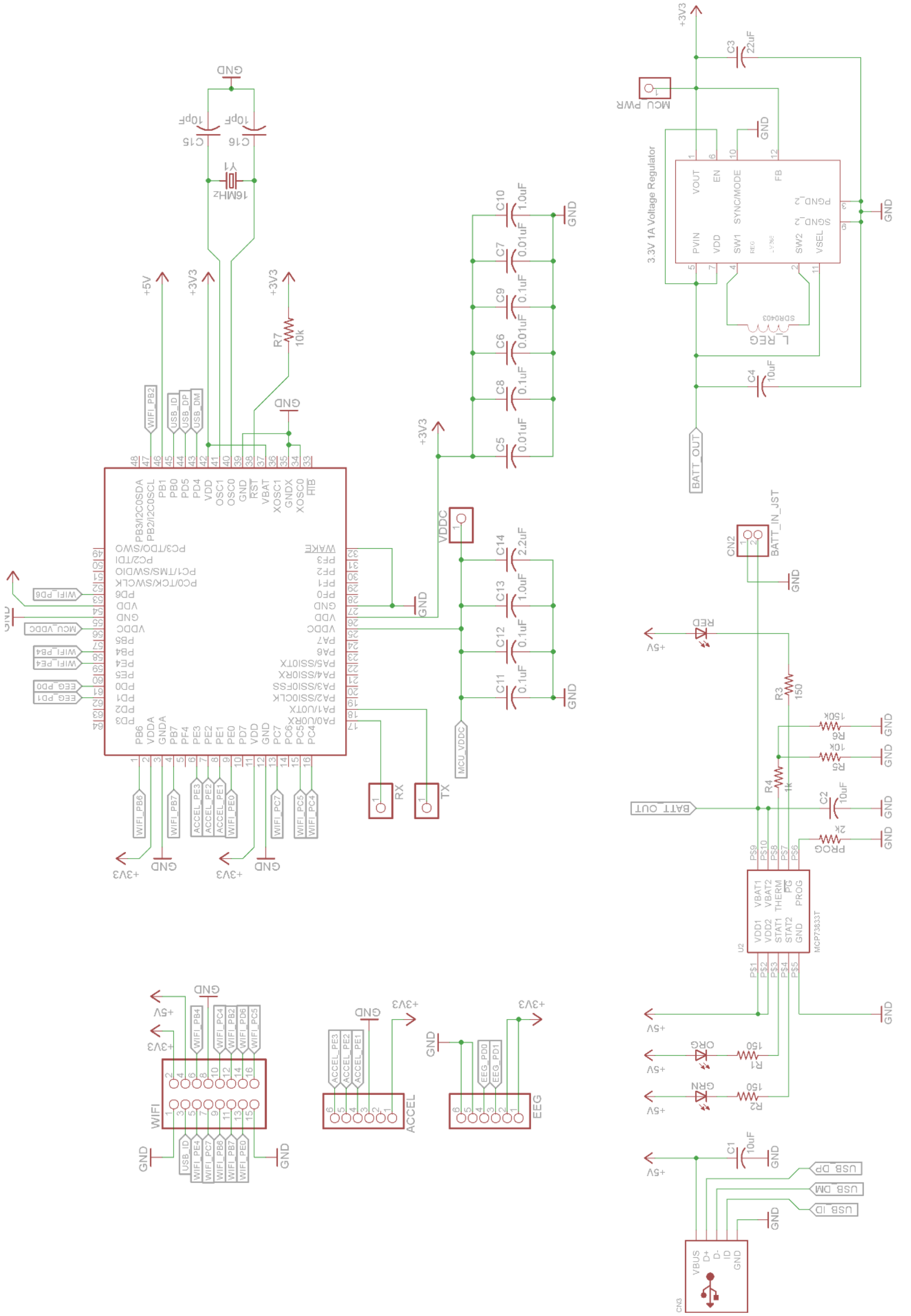
Wi-Fi Direct. (n.d.). Retrieved April 1, 2015, from http://www.wi-fi.org/discover-wi-fi/wi-fi-direct

APPENDIX C – CIRCUIT SCHEMATICS

C.1 COMPLETED EEG DSP HARDWARE DESIGN SCHEMATIC



C.2 Full BIO-Helmet Hardware Design Schematic



APPENDIX D – DATASHEETS

D.1 ANALOG DEVICES ADXL377 DATASHEET

ANALOG DEVICES

Small, Low Power, 3-Axis ±200 *g* Accelerometer

Data Sheet **ADXL377**

FEATURES

3-axis sensing
Small, low profile package
3 mm × 3 mm × 1.45 mm LFCSP
Low power: 300 µA (typical)
Single-supply operation: 1.8 V to 3.6 V
10,000 *g* shock survival
Excellent temperature stability
Bandwidth adjustment with a single capacitor per axis
RoHS/WEEE and lead-free compliant

APPLICATIONS

Concussion and head trauma detection
High force event detection

GENERAL DESCRIPTION

The ADXL377 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The ADXL377 measures acceleration resulting from motion, shock, or vibration with a typical full-scale range of ±200 *g*.

The user selects the bandwidth of the accelerometer using the C_X, C_Y, and C_Z capacitors at the X_{OUT}, Y_{OUT}, and Z_{OUT} pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1300 Hz for the x-axis and y-axis and a range of 0.5 Hz to 1000 Hz for the z-axis.

The ADXL377 is available in a small, low profile, 3 mm × 3 mm × 1.45 mm, 16-lead lead frame chip scale package (LFCSP_LQ).

FUNCTIONAL BLOCK DIAGRAM

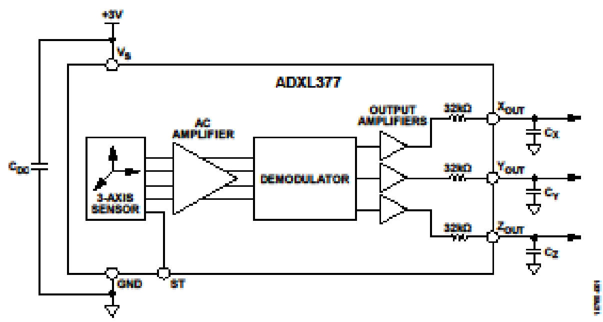


Figure 1.

Rev. 0 Document Feedback

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700
Technical Support www.analog.com

D.2 TEXAS INSTRUMENTS CC3100 DATASHEET

Product Folder | Sample & Buy | Technical Documents | Tools & Software | Support & Community

CC3100 SimpleLink™ Wi-Fi® Network Processor, Internet-of-Things Solution for MCU Applications

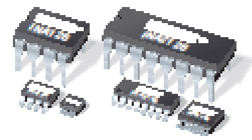
1 Device Overview

1.1 Features

- CC3100 SimpleLink Wi-Fi Consists of Wi-Fi Network Processor and Power-Management Subsystems
- Wi-Fi CERTIFIED™ Chip
- Wi-Fi Network Processor Subsystem
  - Featuring Wi-Fi Internet-On-a-Chip™
  - Dedicated ARM MCU
    Completely Offloads Wi-Fi and Internet Protocols from the External Microcontroller
  - Wi-Fi Driver and Multiple Internet Protocols in ROM
  - 802.11 b/g/n Radio, Baseband, and Medium Access Control (MAC), Wi-Fi Driver, and Supplicant
  - TCP/IP Stack
    - Industry-Standard BSD Socket Application Programming Interfaces (APIs)
    - 8 Simultaneous TCP or UDP Sockets
    - 2 Simultaneous TLS and SSL Sockets
  - Powerful Crypto Engine for Fast, Secure Wi-Fi and Internet Connections with 256-Bit AES Encryption for TLS and SSL Connections
  - Station, AP, and Wi-Fi Direct® Modes
  - WPA2 Personal and Enterprise Security
  - SimpleLink Connection Manager for Autonomous and Fast Wi-Fi Connections
  - SmartConfig™ Technology, AP Mode, and WPS2 for Easy and Flexible Wi-Fi Provisioning
  - TX Power
    - 18.0 dBm @ 1 DSSS
    - 14.5 dBm @ 54 OFDM
  - RX Sensitivity
    - –95.7 dBm @ 1 DSSS
    - –74.0 dBm @ 54 OFDM
  - Application Throughput
    - UDP: 16 Mbps
    - TCP: 13 Mbps
- Host Interface
  - Interfaces with 8-, 16-, and 32-Bit MCU or ASICs Over SPI or UART Interface
  - Low External Host Driver Footprint: Less Than 7KB of Code Memory and 700 B of RAM Memory Required for TCP Client Application
- Power-Management Subsystem
  - Integrated DC-DC Supports a Wide Range of Supply Voltage:
    - V_{BAT} Wide-Voltage Mode: 2.1 to 3.6 V
    - Preregulated 1.85-V Mode
  - Advanced Low-Power Modes
    - Hibernate with RTC: 4 µA
    - Low-Power Deep Sleep (LPDS): 115 µA
    - RX Traffic (MCU Active): 53 mA @ 54 OFDM
    - TX Traffic (MCU Active): 223 mA @ 54 OFDM, Maximum Power
    - Idle Connected: 690 µA @ DTIM = 1
- Clock Source
  - 40.0-MHz Crystal with Internal Oscillator
  - 32.768-kHz Crystal or External RTC Clock
- Package and Operating Temperature
  - 0.5-mm Pitch, 64-Pin, 9-mm × 9-mm QFN
  - Ambient Temperature Range: –40°C to 85°C

D.3 Burr-Brown Products (Texas Instruments) INA126 Datasheet

Burr-Brown Products
from Texas Instruments



INA126
INA2126

SBOS062A – JANUARY 1996 – REVISED AUGUST 2005

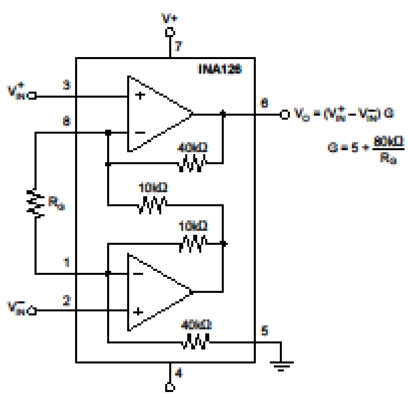
*Micro*POWER INSTRUMENTATION AMPLIFIER
Single and Dual Versions

FEATURES

- LOW QUIESCENT CURRENT: 175µA/chan.
- WIDE SUPPLY RANGE: ±1.35V to ±18V
- LOW OFFSET VOLTAGE: 250µV max
- LOW OFFSET DRIFT: 3µV/°C max
- LOW NOISE: 35nV/√Hz
- LOW INPUT BIAS CURRENT: 25nA max
- 8-PIN DIP, SO-8, MSOP-8 SURFACE-MOUNT
  DUAL: 16-Pin DIP, SO-16, SSOP-16

APPLICATIONS

- INDUSTRIAL SENSOR AMPLIFIER:
  Bridge, RTD, Thermocouple
- PHYSIOLOGICAL AMPLIFIER:
  ECG, EEG, EMG
- MULTI-CHANNEL DATA ACQUISITION
- PORTABLE, BATTERY OPERATED SYSTEMS

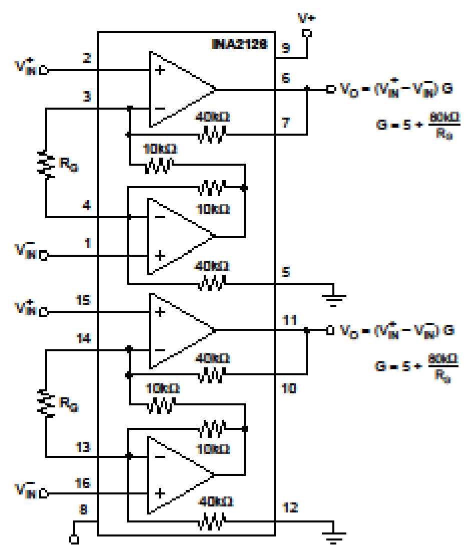


DESCRIPTION

The INA126 and INA2126 are precision instrumentation amplifiers for accurate, low noise differential signal acquisition. Their two-op-amp design provides excellent performance with very low quiescent current (175µA/channel). This, combined with a wide operating voltage range of ±1.35V to ±18V, makes them ideal for portable instrumentation and data acquisition systems.

Gain can be set from 5V/V to 10000V/V with a single external resistor. Laser trimmed input circuitry provides low offset voltage (250µV max), low offset voltage drift (3µV/°C max) and excellent common-mode rejection.

Single version package options include 8-pin plastic DIP, SO-8 surface mount, and fine-pitch MSOP-8 surface-mount. Dual version is available in the space-saving SSOP-16 fine-pitch surface mount, SO-16, and 16-pin DIP. All are specified for the –40°C to +85°C industrial temperature range.



Texas Instruments
www.ti.com

D.4 SHENZHEN PKCELL BATTERY CO., LTD ICR18650 DATASHEET

SHENZHEN PKCELL BATTERY CO., LTD

PKCELL

Polymer Li-ion Battery Technology Specification

Model：ICR18650 6600mAh 3.7V 1S3P

Customer confirmation	Corporate name	
	Checked	
	Approved	
	Corporate seal	

Signed：____________

Drafted by：____________

Signed by：____________

Document No.： **QA.S.0221** Edit：**A/0**

SHENZHEN PKCELL BATTERY CO., LTD
Company address: E2 Building,Guangming Technology Park,No.24 Zhonghua Road,Longhua New Area,Shenzhen
（Tel）：+86-755-86670672 （Fax）：+86-755-86670609
E-mail：pkcell@pkcell.net Website：http://www.pkcell.net

(If manufacturer want to modify the product technology specification, we won't inform you additionally)

MICROCHIP

MCP73833/4

Stand-Alone Linear Li-Ion / Li-Polymer Charge Management Controller

Features

- Complete Linear Charge Management Controller
  - Integrated Pass Transistor
  - Integrated Current Sense
  - Integrated Reverse Discharge Protection
- Constant Current / Constant Voltage Operation with Thermal Regulation
- High Accuracy Preset Voltage Regulation:
  - 4.2V, 4.35V, 4.4V, or 4.5V, ± 0.75%
- Programmable Charge Current: 1A Maximum
- Preconditioning of Deeply Depleted Cells
  - Selectable Current Ratio
  - Selectable Voltage Threshold
- Automatic End-of-Charge Control
  - Selectable Current Threshold
  - Selectable Safety Time Period
- Automatic Recharge
  - Selectable Voltage Threshold
- Two Charge Status Outputs
- Cell Temperature Monitor
- Low-Dropout Linear Regulator Mode
- Automatic Power-Down when Input Power Removed
- Under Voltage Lockout
- Numerous Selectable Options Available for a Variety of Applications:
  - Refer to **Section 1.0 "Electrical Characteristics"** for Selectable Options
  - Refer to the Product Identification System for Standard Options
- Available Packages:
  - DFN-10 (3 mm x 3 mm)
  - MSOP-10

Applications

- Lithium-Ion / Lithium-Polymer Battery Chargers
- Personal Data Assistants
- Cellular Telephones
- Digital Cameras
- MP3 Players
- Bluetooth Headsets
- USB Chargers

Description

The MCP73833/4 is a highly advanced linear charge management controller for use in space-limited, cost sensitive applications. The MCP73833/4 is available in a 10-Lead, 3 mm x 3 mm DFN package or a 10-Lead, MSOP package. Along with its small physical size, the low number of external components required makes the MCP73833/4 ideally suited for portable applications. For applications charging from a USB port, the MCP73833/4 can adhere to all the specifications governing the USB power bus.

The MCP73833/4 employs a constant current/constant voltage charge algorithm with selectable preconditioning and charge termination. The constant voltage regulation is fixed with four available options: 4.20V, 4.35V, 4.40V, or 4.50V, to accomodate new, emerging battery charging requirements. The constant current value is set with one external resistor. The MCP73833/4 limits the charge current based on die temperature during high power or high ambient conditions. This thermal regulation optimizes the charge cycle time while maintaining device reliability.

Several options are available for the preconditioning threshold, preconditioning current value, charge termination value, and automatic recharge threshold. The preconditioning value and charge termination value are set as a ratio, or percentage, of the programmed constant current value. Preconditioning can be set to 100%. Refer to **Section 1.0 "Electrical Characteristics"** for available options and the **"Product Indentification System"** for standard options.

The MCP73833/4 is fully specified over the ambient temperature range of -40°C to +85°C.

Package Types