Construction of a Smart Shirt for Medical Testing Purposes

by

Veda Booth, Leila Chow, and Courtney Lewis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science

in

Electrical and Computer Engineering

_____

March 2018

APPROVED:

_____

Professor John McNeill, MQP Project Advisor

Abstract

When studying the human body, researchers may face difficulties developing test methods broad enough to study large groups yet narrow enough for specific topics. Cost is a factor, as it may halt research. One must also consider comfort during tests. The goal of this project was to create a shirt for medical testing that allows doctors to conduct research for developmental disorders and stroke rehabilitation.

Research shows benefits of low frequency vibration for rehabilitation, yet testing devices are large. There are claims of calming benefits of compression vest for autistic people, yet few studies confirm this. Current products don't allow for compression to be controlled. This project allows the above to be tested conveniently in the form of a modifiable low-cost smart shirt.

Acknowledgements

Foremost, the team would like to thank Professor John A. McNeill for advising our Major Qualifying Project. His patience, knowledge, and wit were all crucial in helping this project become what it is today.

Besides our advisor, the team would like to thank Analog Devices and Allegro Microsystems for helping sponsor our project. Additionally, we would like to thank Bill Appleyard for helping us choose and order parts for our project.

Executive Summary

To culminate one's education at WPI students are required to complete the Major Qualifying Project (MQP), a capstone, in their respective majors. Though some teams begin their project with a solidified idea, our project was introduced to us with the general description of "an MQP with Analog Applications". With no concrete direction on where to begin, the team began an intense process of brainstorming possible ideas for our MQP. Nine possible project ideas were created, they were: wearable heart monitor, automated compression/weighted vest, wearable fetal monitor, window implosion pressure sensor, smart home module, water landing/take off drone, piezoelectric generator, salt water powered emergency light, and endangered animal monitor.

After creating nine possible project ideas and conducting preliminary research on each, the team needed to critically examine each idea and decide which one was not only most feasible to complete in three terms, but also had a significant concentration of analog material. To help guide the decision process the team developed 13 decision criteria that we felt a comprehensive project should include and address. The 13 criteria were the following: Broad applicability, cost, time, learning curve/reference availability, winnability, wow factor, innovation, amount of resources, liability/risk, testability, benefit to society, electrical and computer engineering (ECE) content, and personal marketability.

With these criteria being used as guidelines the team then used three decision methods to narrow down the potential MQP choices. The first decision method was process of elimination; the team used process of elimination as an initial step of removing ideas that distinctly stood out as being unfeasible to work on either due to not meeting many of the decision criteria or being a mutually disliked idea by the team. This first round of decision-making removed the salt water, animal monitor, wearable fetal monitor, and drone ideas. Left with five ideas, the team progressed to using a decision matrix to generate a numbered score for each idea. Using the decision criteria, the team created weights, scored each project idea in the criteria categories, and found the final scores for each project idea.

There were distinct number separations for all ideas except the wearable heart rate monitor, automated compression/weighted vest, and piezoelectric generator, which scored 68, 67, and 69

respectively. To help reach a conclusion, the team consulted their last decision method, the advisor suggestion. As the name suggests, the advisor's suggestion was a decision method that allowed the advisor's recommendations to guide the team's project decision. Our advisor had a high preference for the compression vest, not only due to its application in the health field, but also for the potential elaborations that could be added to the vest system. Due to this, the team chose the compression vest as the project.

Using the automated compression vest as the base for the MQP the team proceeded to develop other functions the shirt system could include, from this second brainstorming period the team developed 3 niche applications the vest could target. The first application, inspired by a teammate's brother is geared towards special needs individuals. This system, composed of a heart rate monitor, pressure sensor, and compression system, will alert a caretaker if the individual wearing the system is self-harming, provide comforting compression when needed, and will monitor the heart rate of the user.

The second function is targeted towards stroke victims; recent research has shown targeted and/or full body vibration may help stroke victims regain muscle function. From this concept the team created a system function composed of an inertial measurement unit (IMU), and four vibration motors. The IMU will track the body posture of the wearer and can notify the user whether they are not sitting or walking properly. Additionally, the vibration motors will allow the user to conduct targeted muscle therapy not only in the hospital, but also in the comfort of their own home. The final application targets athletes; with the use of a heart rate monitor and an IMU the user will be able to monitor their heart rate and physical performance during athletic activity.

Once the team solidified what the functions of the shirt would be, the next step was choosing components that would satisfy each of the shirt functions. To control the entire system the team used a microprocessor. The microprocessor needed would have to be compact, inexpensive, lightweight, easy to use, and have enough memory to control all the system requirements.

Initially the team chose the TI CC2650, a microprocessor that met most of the specifications. However, as the team progressed through the system integration, difficulties programming the microprocessor became prominent. There were little resources online and on

campus to help with the troubleshooting process. As a result, the team switched to the Arduino Uno. The Arduino Uno initially met all the specifications, however the team noticed that the system code was using most of the storage capacity and was interfering with the performance of various shirt functions. To solve this storage problem, the team once again switched microprocessors, this time to the Arduino Mega, a microprocessor very similar to the Arduino Uno but with 8 times the storage.

For the heart monitoring function, the team needed an electrocardiogram (ECG) system that would be low cost, compact, and can send information wirelessly. After the team examined various types of ECG systems, the team chose to use the Polar T34 Heart Rate Transmitter. The system is composed of a strap that sits on the user's skin, below the bust, and a receiver that obtains the heart rate signal from the strap.

The main limiting factors when choosing vibration motors were the operating voltage, current draw, and frequency. Research on the benefit of targeted vibration for muscle therapy specified low frequency vibrations (80-100 Hz) which were not as readily available as vibration motors at high frequencies. The team decided to use 10MM coin vibration motors from Precision Motor drives, as they met every specification.

For motion tracking the team wanted a device that had the capability of measuring the position of a human body in space and in real time. Initially the team explored accelerometers and gyroscopes separately, but after researching IMU units which combine both functions often with a third, decided to focus research there. The team compared three IMU systems and decided to choose the LSM9DS0 an IMU system sold through Adafruit. For the comfort and self-harm detection function the team needed to research two things 1) how to create a system that would create a feeling of compression 2) how to detect whether the wearer of the shirt was self-harming themselves.

To create a system that mimicked the feeling of being squeezed the team decided to model their function after a blood pressure compression sleeve constructed by students at. To construct the system the team needed to use automated air pumps that were small, lightweight, had a low current draw, and were relatively quiet. Traditional air pumps used to inflate tires or air mattresses

are loud and obstructive therefore the team decided to explore small air pumps used in home aquariums. The air pumps that the team explored were all similar in terms of price, size, and voltage specifications, however some drew more current than others. Therefore, the deciding factor of the air pumps chosen was current. The team chose to use four air pumps with an operating voltage of 5V and a max current of 130mA.

For self-harm detection the team explored force sensitive resistors (FSR). The FSRs researched fell into two categories, premade and self-constructed. Premade FSRs had the benefit of being a complete system already constructed with all the specifications already documented. Unfortunately, all premade FSRs covered a very small area with the largest one the team documented measuring at a 10mm by 100mm (0.39in by 3.9in). Since the team wanted an FSR that would be able to detect self-harming in large areas of the body, the decision was made to go the self-construction route. To construct a basic FSR, the team used velostat (a pressure sensitive plastic material), copper, and clear tape.

With numerous system functions it was crucial that the team find a power source that not only had the ability to support all the features, but also was lightweight enough to be carried on a human body. The team found portable power banks used to charge personal electronics to be the best choice of battery to power the system, and thus compared specifications of various battery packs. The team explored four power banks sold on Amazon and decided on the Anker Portable Charger PowerCore 20100, as it was relatively small, lightweight, and had a large capacity of 20Ah.

To control the system, the team decided to use Bluetooth Low Energy (BLE) and a compatible phone app. The team reviewed two Bluetooth modules, the first was the Bluefruit LE. Sold through Adafruit, this module was in the form of a breakout board and had an existing phone application that could be used to send and receive data. The second module was the RedBear BLE Shield which also had a compatible app, however came in the form of a shield for the Arduino. Due to the very similar specifications the team chose the shield as it would save space on the future PCB with the system components.

After choosing the necessary components the team proceeded to the construction, testing, and debugging stage. Here the team constructed the physical systems for each of the shirt functions

and wrote the necessary code to control each of the functions. The ECG records heart rate in the form of beats per minute and accurately measures the heart rate of individuals. Additionally, the data can be visualized through the RedBear BLE app or a computer, allowing for future data manipulation.

In the self-harm system, the FSR determines whether the wearer is engaging in self-harming behavior by examining the frequency at which hits are being registered. If the threshold for maximum hits in a period is met, the phone application will display the message "Self-Harm Detected" prompting the individual with the app to check on the wearer.

The compression system can be controlled through the phone app. When desired, the individual can turn on the air pumps through the app and the compression system will inflate. Vibration motors are controlled through the app as well and the user can control whether the motors are on or off and how many motors run at one time.

For posture tracking, the IMU continuously saves the values of the x, y, and z coordinates relative to its position in space. These values can then be displayed on a graph and the numbers recorded can be further analyzed and manipulated. The overall system works as expected, and the team believes with more refining the project has the ability to become an on-market product.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

When initially given the guidelines for what our Major Qualifying Project (MQP) was to incorporate, we were given two words, "analog applications". With a group of three individuals, each with interests in various concentrations of Electrical and Computer Engineering, the task to develop a project that not only satisfied each of our curiosities (and that of our advisor) but also possibly had an impact on individuals' lives was nothing short of monumental. Throughout the first weeks of our project we began to propose various ideas to one another and soon, one specifically stood out.

The brother of one of our team members has autism. Throughout her experience with him and his classmates at his school, she became aware that weighted vests were sometimes used to help calm the students. As we researched the existing vests on market and the information behind them, it became increasingly apparent that there were no vests available that allowed the user to control the location and intensity of compression automatically, nor was there any evidence, besides anecdotal accounts, of the success of these vests [1], [2]. It was evident that this was a topic that should be explored due to the potential benefits, and thus the idea of creating an inexpensive automated weighted vest system with targeted compression functions was developed.

Concurrently, per our advisor's suggestion, the team began to explore the area of stroke muscle and balance rehabilitation. Present research has alluded to the benefit of targeted low frequency muscle vibration in regaining muscle function in stroke victims; however, current apparatuses are large, intrusive, and only allow for testing to be conducted in a lab. Each year 795,000[3] Americans experience either new or recurrent strokes and, these events often lead to decreased motor function in areas of the body and can contribute to decreased self-autonomy and feelings of depression [2], [4]-[8]. To aid in the research with a goal of moving testing from the confines of the lab to the home, the team decided to incorporate a low frequency targeted vibration module in the shirt.

Using the elements needed to create an automated weighted vest and targeted muscle vibration the team was able to expand our applications and add posture correction and athletic performance tracking as two sub features on the shirt, creating a comprehensive and customizable smart shirt system.

2. Brainstorm

Before our MQP project became the smart shirt, the team went through a period:

- Developing an array of possible projects with analog applications
- Analyzing and evaluating the feasibility of conducting each project
- Selecting the smart shirt as the best option
- Defining and refining the functions of the shirt
- Researching and planning how we were physically going to achieve the desired operations

Through extensive research and brainstorming, what began as a project with the generic description of analog applications, became the development of a "smart-shirt" with a variety of applications. This investigation and discovery stage involved not only the proposition of various ideas related to analog applications, but also a process of deciding which of the ideas proposed was best to pursue for the duration of the MQP process. It additionally allowed the team to develop a strong fundamental idea of what the end project was to look like, which reduced the risk of future problems from occurring due to an underdeveloped proposal. Detailed below is the process by which the smart vest project was decided and refined upon.

2.1 Proposed Projects

The process by which possible project ideas were proposed was wholly organic and a product of the creativity of each respective member. The team developed nine initial project ideas which can be categorized into the following:

- Health Applications
- Natural Disaster Mediation/Rescue
- Other

2.1.2 Health Applications

2.1.2.1 Wearable Real-Time Heart Monitor

An Electrocardiogram (ECG) is a test that measures the electrical activity of the heart [9], [10]. Doctors use the data gathered from this test to assess the overall function of the heart and

identify heart complications such as, irregular heart rate, fluid or swelling in the sac around the heart, heart attacks, or blocked arteries [9], [11], and [12].

This project would create a wearable real time heart rate monitor in the form of a watch to provide accurate ECG measurements and information for individuals concerned about developing heart issues, or persons with preexisting cardiac problems. This monitor would use one of Analog Devices' wearable ECG's [13] and a microprocessor to interpret the data. The data would then be sent to an app or database for further analysis from a doctor.

Table 1 displays wearable products currently on the market that use ECGs to monitor or record data about the heart [14]-[16].

Table 1 Wearable Heart Rate Monitors on the Market

Product Name	Cost	Description	Real Time
Qardiocore[14]	$449	Strap placed under chest to provide a heart health analysis.	Yes
Hexoskin[15]	$400	Wearable shirt to track cardiac, respiratory, sleep and activity biometric data.	Yes
AliveCor Kardia[16]	$99	Pad to measure heart rhythm with finger tips.	No

Although performing an ECG test from the wrist appears to be most optimal when creating a wearable device for a large population, wrist measurements do not provide the same accuracy as chest and multi-lead tests, which, especially in health applications, is of the utmost importance. There is the additional question of whether continued heart measurements provide any use to the individual; the American Heart Association suggests against routine screening for the prediction of coronary heart disease for low risk adults and children [17] and the United States Preventive Services Task Force has additionally concluded that there is insufficient evidence to support routine ECG screening [18].

2.1.2.2 Wearable Fetal Monitor

There are many factors that can lead to birth defects and complications during the gestational period [19]-[21]. Women who have high risk pregnancies are especially susceptible to difficulties and must be closely monitored by a doctor for any abnormalities [22], [23]. Wearable fetal monitors are devices that allow the mother to record and monitor the health status of the fetus while at home. This fetal monitoring helps mothers and doctors understand the physical state of both the mother and the child. Prototypes for fetal monitoring devices are shown in Table 2 however, current at-home fetal monitors on the market today are uncommon [24]-[26].

Table 2. Fetal Monitoring Prototypes

Product Name	Purpose
Modoo [24]	To monitor the baby's heart rate and movement
PregSense [25]	To allay the mother's fears by transmitting data about the health of the mother and fetus.
TinyKicks [26]	To help mothers count the amount of fetal kicks

The aim of this project would be to provide pregnant women with a device that monitors the health of their fetus at home at an affordable cost. A device with an ECG testing system and a microprocessor would be attached to the mother's stomach with a skin-safe adhesive, the data would then be sent over Bluetooth to a mobile application as well as to a medical database for analysis by a medical professional.

The difficulties of this project stem from the inaccuracy of current wearable ECG monitors, as well as the cost of creating a small, affordable, and accurate product. For increased accuracy, the price of the wearable ECG rises, thus making it difficult to create a reasonably priced project. Additionally, this project would be difficult for the team to test on campus, due to the lack of readily available pregnant test subjects.

2.1.2.3 Automated Compression/Weighted Vest

A compression or weighted vest is an article of clothing that provides proprioceptive stability to the wearer. The constant pressure may help increase body awareness, improve balance, and provides sensory feedback for those who have sensory integration disorder, autism, or other neurological disorders [27], [28].

Shown below, Figure 1 is an example of compression/weighted vests currently on the market. Figure 1 depicts the Squeeze Pressure Vest, a compression vest that allows the wearer to pump air to create the desired pressure. Figure 2, weighted vests are also used to provide sensory stimulation.
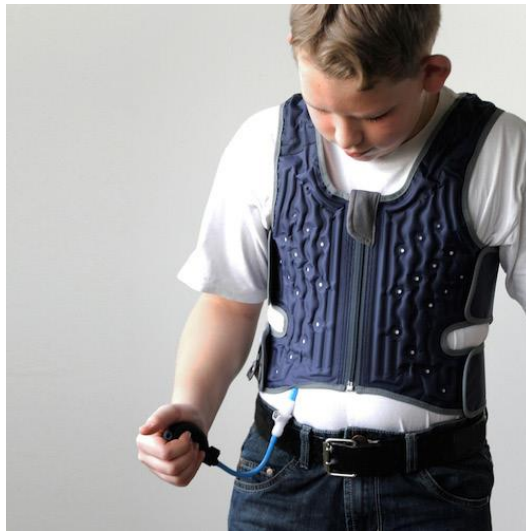
Figure 1. Squeeze Pressure Vest [10]

Figure 2 Weighted Vests

Presently, little research has been conducted to verify how useful these products are in helping patients, and what research that has been conducted concluded that compression and weighted vests were ineffective for treating autistic patients [27]-[31].

One difficult feature of this project would be to have the vest accurately respond to the wearer's physical responses. Ultimately, the vest would utilize a feedback system to learn how much physical sensation is useful for the user.

2.1.3 Natural Disaster Mediation/Rescue

2.1.3.1 Window Implosion Pressure Sensor

Hurricanes, tornadoes, monsoons, and other high wind weather activities affect numerous people across the globe [32]-[34]. In many of these circumstances, individuals living in developing or underdeveloped countries do not have the resources to board their windows, evacuate their homes, or whether the storm in a basement. In such situations it would be beneficial if individuals had small, inexpensive, flexible modules which could be fixed on a window and would alert the user whether the window was at risk for fracture or shattering. Applications would be especially beneficial in locations where individuals cannot evacuate such as a hospital or nursing home. An additional possible application would be fixing the module to windows of cars, to use the data gathered in crash forensics.

One difficult feature of this project would be figuring out how to devise a method of warning the individual of the chance of the window breaking with enough time for them to evacuate. Challenges also arise when considering the method of testing this project. To understand how windows break during tornadoes and other high wind/pressure situations, the team would need access to a wind tunnel powerful enough to break a window. Furthermore, because applications would be in lower income areas, it would be important that the module be inexpensive, which becomes a challenge when developing a sophisticated system with high accuracy.

2.1.3.2 Water Landing/Take-off Drone

For both leisure and emergency situations, having a drone that could be used for surveying or dropping off supplies, while having the ability to take-off and land on water would be beneficial. The drone would have to be lightweight, and the battery needed to provide power for the extra speed and lift from a water take-off would have to be small enough so as not to interfere with the drone's normal flying functions.

One main problem associated with this project stems from its aerospace and not ECE concentration. Since there are many drones available for purchase, the team would likely be working on modifying a drone to allow it to take-off and land on water; these topics are rooted more in physics, aerodynamic, and aerospace areas, thus it would not make for a pertinent ECE capstone.

2.1.3.3 Saltwater Powered Emergency Light

According to the United Nations, between 1995 and 2015 2.3 billion people have been affected by flooding, 157,000 of these individuals have died as a result [35]. As this number continues to rise both due to population and changing weather patterns brought by climate change, the need to quickly identify where individuals are trapped by rising waters is apparent. In various locations where flooding occurs, the waters are saline concentrated. The purpose of this project would be to create a small, yet powerful emergency light powered by saltwater; this would allow victims to drop these lights in the water around them providing rescuers with an identifying beacon.

For this project to succeed the team would have to conduct extensive feasibility research into the mechanisms of using saltwater to generate electricity, and whether there are conductors efficient enough to utilize the amount of salt in, for example, ocean water to power a small yet bright light bulb. The project would additionally have heavy concentrations in chemistry which is beyond the scope of knowledge of this team.

2.1.4 Other

2.1.4.1 Smart Home Bedside Module

Figure 3 Amazon Echo

The modern homeowner has many smart devices that allow them to control various aspects of their life, however, only recently is attention being focused towards the place they spend a majority of their time: the home. The proposed module would be small, lightweight, and allow the homeowner to control various aspects of their house from any location in their home. Functions would include, locking or unlocking doors, arming/disarming one's security system, or controlling the function of lights in one or multiple rooms.

The device interface would have to be able to control the various functions of the house from all locations of the house, so many wireless interference patterns would have to be taken into account. Additionally, the goal of the module would be to produce an object that was lightweight and compact, thus difficulties may arise when attempting to find components that are both inexpensive, small, and have the capacity to perform the functions needed.

If this project were to be developed, the team would have to consider the many products that are currently on the market such as the Amazon Echo, Google Home, Wink Hub, and Logitech Harmony Elite [36]-[39] (Table 3).

Table 3. Similar Items on the Market

Product Name	Function	Price ($)
Amazon Echo[36]	Play music, make calls, set alarms and timers, answer questions, control smart home devices	99.99
Google Home[37]	Play music, answer questions, set alarms and timers, control smart home devices	129.00
Wink Hub[38]	Lock and unlock doors, control lights, control thermostat, control home security	99.00
Logitech Harmony Elite[39]	Control lights, activate home devices, television and app control, cross functional with Echo and Google Home	349.99

2.1.4.2 Piezoelectric Generator

To show individuals of the WPI community an interesting property of materials, the team would develop a piezoelectric generator which, through some method of percussion, would generate enough electrical energy to light a novelty module. The generator would be placed on the WPI campus in an area that received enough movement/traffic to generate the electricity needed.

Despite being rooted in electrical engineering, this project may be too simple to be completed in three terms, and unable to provide the team with the academic rigor wanted from an MQP.

2.1.4.3 Endangered Animal Monitor

Currently there are thousands of animals living in zoos and sanctuaries worldwide [40], [41]. Many of these animals are endangered species and targets of poachers. To help prevent animals in high-risk locations of being poached or maimed the team would create a "smart" collar. This collar would not only provide the GPS locations of the animals but would additionally monitor heart rates and other bio-information of the animals. Using the animal's heart rate, the team would create a system that would correlate and monitor the heart rate of the heard. With this, if there was an external threat, multiple animals would show similar symptoms of fear such as increasing heart rate.

This would allow rangers to quickly respond to the stimuli and help prevent external animal and poaching attacks.

This project relies on easy access to animals to test our device on which would be difficult as it would mean the team would need to find an external organization to allow us to test our project on their animals.

3 Decision Methods

Before deciding what project idea to select for the MQP, the team created a table of the benefits and risks of various decision-making methods to understand which method would allow the team to choose a project that we not only found interesting but was also the most relevant. Table 4 displays the various decision methods.

Table 4. Decision Methods

Decision Method	Benefit	Risk
Decision Matrix	• Rational/Scientific Method	• Can tie equally good ideas • Sensitive to weights
Process of Elimination	• Avoiding something you think you don't like	• Overlooking qualities
Advisor's Preference	• Quicker • If successfully completed increases chance of getting an A • Makes your advisor happy	• Being unhappy • Not feeling a personal connection to the project • If not completed to advisor's vision increases chance of receiving bad grade

Based on the benefits and drawbacks of each method, the team chose to use all three methods presented in Table 4 to decide upon a project. The decision methods and how they were applied are described in further detail below.

3.1 Process of elimination

The team used process of elimination as an iterative decision-making process that included both an objective view of the facts and took into account each member's personal preference

towards the project options. Though beneficial this method posed the risk of the team overlooking qualities of an idea they are not inherently interested in.

Initially, the team began with the following nine project ideas:

1. Wearable Heart Monitor
2. Automated Compression/Weighted Vest
3. Wearable Fetal Monitor
4. Window Implosion Pressure Sensor
5. Smart Home Module
6. Walter Landing/Take Off Drone
7. Piezoelectric Generator
8. Salt Water Powered Emergency Light
9. Endangered Animal Monitor

The team then briefly researched each project idea, and through process of elimination chose to narrow the list to seven projects, eliminating projects 8 and 9, to expedite the research brainstorming phase. The team removed project idea 8 because it was found least desirable for the team. The removal of project idea 9 stemmed from the lack of available access to endangered animals for testing.

After completing research for the seven remaining projects, the team used process of elimination again to refine the options to five choices for the decision matrix. In this stage the wearable fetal monitor, project idea 3, was eliminated because the team believed it would take too long to fully understand the concepts around pregnancy and fetal monitoring. The team also removed the water landing/take-off drone, project idea 6, because its application lacked strong ECE content.

3.2 Decision Criteria

To help guide the next elimination process, the team created a set of 13 decision criteria. The criteria compiled are the most important qualities the team felt wanted the project to include; ranked in no specific order, Table 5 contains the specifications.

Table 5. Decision Criteria

#	Criteria
1	Broad applicability
2	Cost
3	Time
4	Learning curve/Available references
5	Winnability (MQP Award Competition)
6	"WOW!" Factor
7	Innovation
8	Amount of Resources
9	Liability/Risk
10	Testability
11	Benefit to society
12	ECE Content
13	Personal marketability

3.2.1. Broad applicability

Broad applicability refers to the number of markets the project can be applied to. A project that can be utilized by a variety of fields can increase the number of individuals who find benefit from it. Additionally, if we were to sell the product we developed, having multiple applications increases the chance of making a profit.

3.2.2 Cost

Cost takes into consideration both the amount of money it would take to prototype the project and how much it would cost for the target audience to purchase it. An over budget project is infeasible, and a product with an exorbitant market price would defeat the team's goal of creating a project that is affordable to consumers. A project proposal would be deemed feasible considering that the estimated costs would be near or below the budget of $1000; an idea would be considered less if the estimated cost would be significantly over the budget.

3.2.3 Time

The project best suited for the team must be completed within three academic terms. Therefore, the team considered how much time would be needed to fully realize the project. This category includes the time needed to learn and understand new material to complete the project, the prototyping period, the testing period, the writing process, and finalizing the project.

3.2.4 Learning curve/References

It was not expected that every team member would be well versed in all the topics to be covered by the MQP, therefore it was important to take into consideration the time it would take the team to learn, understand, and apply new concepts. Equally as important was ensuring the information needed to understand these new concepts, such as reports, books, or professors, was easily available. The learning curve had to be manageable to allow the team time to both learn the new material and apply it in the specified time frame.

3.2.5 Winnability

The Provost's MQP award gives recognition to students who demonstrate achievement within their discipline [1]. The team wanted to develop and chose an idea capable of achieving recognition for this award.

3.2.6 "WOW!" Factor

In line with the previous criterion, WOW Factor refers to developing a project that leaves a strong impression on both the judges of the Provost MQP award, but also all others who review our project. The team additionally had a special requirement from their advisor, who asked that the project developed be something that would "look good" on the department social media pages.

3.2.7 Innovation

Several patents, companies, new products, and research methods have risen out of MQP projects completed in various fields. Thus, although not a top priority, the team wanted to consider pursuing a project that had the potential to be developed into a commercially feasible product.

3.2.8 Amount of Resources

This criterion refers to the availability of the information needed to successfully complete the project. This information can be in the form of research materials, such as academic journals, or persons who are able to provide us with knowledge or help.

3.2.9 Liability

Projects involving the use of human or animal test subjects require detailed guidelines and contracts, and additionally are at risk of lawsuits if performed incorrectly. Since the team was including wearable devices in their brainstorming, it was important to keep liability issues in mind, and thus develop a project without high potential for injury to those testing the device.

3.2.10 Testability

In addition to conducting background research, it is important to build and prototype one's project to see the errors that may arise once the idea moves from concept to reality. It was therefore important to consider whether or not the project idea would be something in which testing, and prototyping would be feasible.

3.2.11 Benefit to society

Another criteria the team considered was the net benefit the potential project had to society. The team hoped to develop an idea that was not only interesting but aided various individuals as well.

3.2.12 ECE Content

To fulfill the WPI requirements for an Electrical and Computer Engineering (ECE) MQP, the project must involve significant ECE content. A project idea was therefore rejected if it contained no relevant application of ECE concepts.

3.2.13 Personal Marketability

The project should be applicable to the individual team member's interests in such that they each can use it to show each of their strengths. This criterion was not used in the decision matrix, but rather to make the final project decision.

3.3 Decision Matrix

To judge the remaining five ideas, the team used a decision matrix. The full decision matrix can be found in Appendix H: Decision Matrix. A decision matrix is a logical approach to narrowing down project ideas. It benefits the decision process by weighting the choices purely based on numbers, excluding personal opinions. However, this method raises the risk of unsuccessfully providing a clear choice as it is possible for the options to be close in value based on the weighted criteria.

The decision matrix used the decision criteria, now weighted, to rank the project options. The team ranked each criterion as either a three, two, or a one; where a three was a must quality, and a one referred to a quality that was considered but not a necessity. Table 6 shows the decision criteria weights.

Table 6. Decision Matrix Criteria Weights

Weight	3	2	1
Criteria	Time	Broad Applicability	"Wow" Factor
	Learning Curve/References	Cost	
	Winnability	Innovativeness	
	Amount of Resources	Liability	
	Testability	Benefit to Society	
	ECE Content		

To calculate the score, the weight of the criteria and the ranking given to each project idea was multiplied and the total was summed. The higher score signifies the project fits the criteria the team described.

Table 7 shows the final scores for the project ideas. The wearable real-time heart monitor, the smart vest and the piezoelectric generator were all close in total scores. Since those three ideas also ranked significantly higher than the other two ideas, the team eliminated the smart-home module and window implosion sensor ideas. Due to the close scores of the Wearable Real-Time HR Monitor, Automated Compression/Weighted Vest, and Piezoelectric Generator, the team proceeded to the final means of decision making.

Table 7. Decision Matrix Scores

Project Idea	Score
Wearable Real-Time HR Monitor	68
Automated Compression/Weighted Vest	67
Smart Home Module	56
Piezoelectric Generator	69
Window Implosion Sensor	54

3.4 Advisor's Suggestion

The advisor's suggestion is a method that allows the advisor's recommendation to guide the team's project decision. This method helps break ties when the team cannot agree on a project idea together. However, the team risks not being emotionally invested in or satisfied with the chosen project.

After narrowing down to three project ideas, the team utilized the advisor to help make the final decision. Our advisor had a high preference for the compression vest, not only due to its application in the health field, but also for the potential elaborations that could be added to the vest system. Due to this, the final choice for the project idea was the compression vest.

4 Background Research & Specifications

4.1 System Requirements

Using the automated compression vest as the base for the MQP project, the team proceeded to develop other functions the shirt system could include. From this second brainstorming period, the team developed 3 niche markets that the vest could be applied.

4.1.1 Special Needs

The first application targets individuals with special needs. Here, the vest would have a customizable compression system that would allow users to control the strength, location, and duration compression is applied to the body. An ECG chest strap would allow heart rate to be measured before, during, and after compression; data from this can potentially be sent to the user's physician or used by researchers to help discover correlations between compression and stress levels.

Some special needs individuals tend to self-harm and while parents, teachers, or caretakers attempt to prevent/stop this from occurring, in situations where the individual is on their own, it may take a long time before actions are taken to stop the event. To help mitigate the damage from self-harming, the team decided to include a pressure sensor to detect constant repeated pressure, such as in the case of one hitting themselves. If self-harm is detected a signal is sent to an application on a caretaker's phone, to notify them. To control the above functions of the vest the system will need a microcontroller that is able to handle the sensors needed and send and receive information via Bluetooth.

4.1.2 Rehabilitation

The team also intends to include features for rehabilitative health purposes. 795,000[3] individuals suffer from strokes each year, resulting in complete or partial loss in muscle function/control; recently research [42]-[46] has shown that targeted vibration may help individuals regain the use of affected limbs. To help in muscle rehabilitation the team will include a vibration system that allows for targeted vibration therapy. Additionally, applications in the rehabilitative sector can be for posture correction and gait monitoring. Using an inertial measurement unit (IMU) to detect posture as the user sits or walks, data can be logged to allow doctors to provide more

accurate physical therapy. Additional vibration motors located at key spots can help users themselves, self-correct posture while at home.

4.1.3 Fitness

The final application the team will be exploring is improving the physical performance in athletes. The device will utilize an ECG and IMU to measure the user's heart rate and, detect body movement, and have a controlled compression application. This data would be stored in the microprocessor used to control the system, and data will be transferred to a phone or computer using Bluetooth.

4.2 Component Research and Specifications

To have a fully functioning shirt it was necessary that the team thoroughly research each of the required components and compare multiple possible options to help select the best choice. Table 8 outlines the components required to complete each function of the smart shirt.

Table 8. Component Requirements

Shirt Function	Component Needed
Heart Monitoring	Heart rate monitoring electrode system
Muscle Vibration	Vibration Motors
Motion Tracking	Body positioning sensors
Comfort and Self-Harm Detection	Compression System Self-harm detection system
System Control	Microprocessor

4.2.1 Heart Monitoring

The smart shirt incorporates a heart rate monitoring system which allows the user to log and track data for use in assessing athletic performance, and preliminary stress and health analysis.

4.2.1.1 Background

An electrocardiogram (ECG) is a test which measures the electrical activity of the heart through electrodes that are placed on the skin. For accurate heart rate readings, each electrode must be placed directly onto the skin.

The accuracy of the heart reading is directly correlated to the number and placement of electrode leads and the type of electrode used. Certain electrode leads require the user to apply water or gel to the surface of the skin before application to increase conductive properties, while others can provide accurate readings through dry skin. Electrode construction material can range from adhesive backed options [47], [48], to textile [49]-[51] and metal electrodes as shown in Table 9.When used for wearable applications, premade straps [52]-[54] which enclose the electrodes in a small conveniently shaped package appear to be prominent.

Table 9. Heart Rate Monitoring Options

	Conductive Fabric[49-51]	Patches[47],[48]	Premade straps[52]-[54]
Description	Non-adhesive material embroidered with conductive yarn	Adhesive material	Wearable material specifically for the chest
Advantages	Customizable	Inexpensive[47][48] Direct contact with skin	Completed tool
Disadvantages	Expensive(82.50E)[49]	Must be replaced	Hard to manipulate sensor placement
Dry or Wet	Either	Wet	

The team chose to use a premade strap for the smart shirt application as the finished packaging provided us with easier means of integrating it both in our shirt and circuit design. The decision additionally shortens the amount of time needed to construct an accurate device and instead allows for more time to analyzing the data received.

4.2.1.2 Specifications

The team is using the Polar T34 Heart Rate Transmitter to read the heart rate of the user in our system. The ECG strap touches the user's chest directly and must be moistened to obtain accurate readings. A heart rate receiver obtains the heart rate signal from the strap and transmits that data to the Arduino Uno microcontroller.

4.2.2 Vibration Technology

The smart shirt incorporates a system of four vibration motors to provide users who have been affected by strokes with targeted vibration muscle therapy.

4.2.2.1 Background

To choose the vibration motor that would provide the user with the most beneficial muscle vibration, the team researched the current methods used by researchers when applying vibrations to the body. It was found that vibration techniques used for the body could be divided into two main categories, whole body vibration, and targeted vibration [43], [45], [46]. As this application was for a wearable shirt, the team chose to narrow our research to localized vibration techniques. From this, the team found that vibration frequencies used to target rejuvenating muscle function/activity were on the low end of the frequency spectrum ranging from 80-100 Hz.

When looking for a motor the team additionally included the IP 67 code [55], which classifies the degree of protection the device has against dust and water; the team took this to be an important feature as the motors were to be placed directly on the body. With the specifications of finding a vibration motor that both operated at a low frequency, was compact, and inexpensive the team was able to compile a list of possible motors to choose from. Table 10 compares these characteristics against four types of vibration motors. Based on this data, the team selected the coin vibration motors from Precision Motor Drives as they met current, frequency, and cost specifications.

Table 10. Comparison of Vibration Motors

	PCB Vibration Motor	17MM (length) Vibration Motor (Cylinder) 307-001	7MM DC Motor	10MM Vibration Motor (Coin) 310-004
Operating Voltage (V)	.8-3DC	1.5		
Cost per unit ($)	1.74	5.5		
IP 67 Rated [55]	No			
RPM	8500 +- 15%	6,300	4320	7,300
Current	40mA	85mA		35mA

4.2.2.2 Specifications

The shirts will have four vibration motors, their positions will be customizable, however for testing purposes the upper right sleeve was chosen as an arbitrary location to fix the motor. The user can control the vibration motors by using the phone app. Each vibration motor can be toggled individually.

4.2.3 Motion Tracking

The smart shirt incorporates a motion sensing device to track the positioning and motion of the user in space for applications both in balance and posture correction as well as for performance enhancing purposes in athletes.

4.2.3.1 Background

To provide the user with relevant information on their posture and body position in space the team researched the accuracy and applications of an accelerometer, gyroscope, and inertial measurement unit (IMU). While an accelerometer specifically measures the acceleration of a body, and a gyroscope measures orientation and angular velocity of a body in space, an IMU measures

both of those functions while also incorporating a magnetometer. After comparing the price and accuracy of the three units separately, the team determined an IMU would not only incorporate all of the functions needed, but also provide equally as accurate readings as the units separately, and at a reasonable price. The team explored a variety of IMU systems as shown below in Table 11. After comparing the cost, range, and additional features of each sensor, the team ultimately selected the LSM9DS0.

Table 11. Characteristics of the IMU

	MPU-6050	**MPU-9250 and SAMD21**	**LSM9DS0**
Cost ($)	39.95		14.95
Output	Digital, I2C		Digital, I2C, SPI
# of Axes	3		
Gyro Range	±250, ±500, ±1000, and ±2000 dps		±245, ±500, and ±2000
Accelerometer Range	±2g, ±4g,, ±8g and ±16g		
Additional Features	Capable of interfacing with other sensors	Compass	Magnetometer

4.2.3.2 Specifications

This device uses the LSM9DS0 Inertial Measure Unit (IMU) to track the wearer's movement. The IMU combines an accelerometer, a gyroscope, and magnetometer and transmits data to the microcontroller through a digital I2C output. Although capable of handling a variety of sensitivities, Table 12 shows the ranges the team chose for each function of the IMU, these were selected based on the range of motion humans are capable, as well as the specifications medical researches chose when incorporating IMU and similar systems in body posture and positioning research [56]-[64].

Table 12. IMU Characteristics

	Accelerometer	**Gyroscope**	**Magnetometer**
Range	+/-2g	+/-500 dps	+/-2g

4.2.4 Comfort and Self-Harm Detection

The smart shirt incorporates a variable compression system to provide soothing pressure to the user. Using a force sensitive resistor (FSR), it also can detect if the user is self-harming, and then relay that information to a relevant party.

4.2.4.1 Background

Compression System

Currently there are two types of compression vests styles on the market. One style consists of a system of weights, as low as two pounds that provide a hugging sensation with the purpose of calming and relaxing the user. The other is an inflatable vest which requires the user to manually pump air into the vest.

There are no options for an automatic inflatable compression vests available for purchase currently, therefore the team wanted to create a system that allowed for the compression to be automatic and tailorable to the individual. To create the feeling of a weighted/compression system the team decided to mimic the system of a blood pressure cuff. In order to construct such a system, the team researched various air pumping and release mechanisms. From the research, the team chose an air pump motor used in home aquarium applications due to its compact size, ability to run at a low voltage and low noise output.

Force Sensitive Resistor

The largest FSRs on the market are the Interlink Electronics 30-73258 and the Interlink Electronics 34-00069. The 30-73258 is a square shape with an active area size of 38.1mm (1.5in) and an overall size of 43.69mm (1.72in); the 34-0069 is a 100mm x 10.2mm (3.94in x 0.4in) strip. Both FSRs can detect forces between 0.04lbs ~ 4.5lbs. These available products were not large enough to cover a wide enough body area to detect potential self-harm from the wearer, thus, instead of purchasing premade FSRs, the team decided to construct their own FSR using "Velostat" (a pressure sensitive plastic material), copper, and clear tape.

4.2.4.2 Specifications

To form the compression system, two thigh blood pressure cuffs are attached together to create a unit that wraps around the wearer's waist. Air is pumped into the cuffs using four air pump motors. The FSR is placed on the chest area to potentially detect forces, such as hitting, from the wearer. The resistor produces a change in voltage that is interpreted by a microcontroller. If the voltage change is large enough, a notification will be sent via Bluetooth to a connected device, letting a caretaker know if they should activate the compression system. When the wearer no longer needs the cuffs inflated, they can deflate them by releasing a valve located at the end of the cuff.

4.2.5 Data Interpretation and Analysis

4.2.5.1 Background

To process the data received from the sensors, the team investigated a variety of microcontrollers. The microcontroller needed to have the ability to control every peripheral used by the smart shirt. Table 13 shows the amount of microcontroller pins needed for the smart shirt system. In total, the microcontroller must have at least eight digital input/output pins, one analog pin, and two pins for I2C.

Table 13. Necessary Microcontroller Pins

Smart Shirt Function	ECG	Vibration	Compression	FSR	IMU	Total
Digital IO Pins	1	5	2			8
Analog IO Pins				1		1
I2C Pins					2	2

The data processed by the microcontroller will be sent to the user wirelessly through Bluetooth. Some microcontrollers have built in Bluetooth modules while other can be interfaced with external Bluetooth modules through SPI or I2C protocol. Additionally, the team considered the microcontrollers ease of use, the amount of instructional resources available, and the cost. Table 14 compares these characteristics against three different microcontrollers

Table 14. Microcontroller Comparison

Model	ADI: ADuCM3029	TI CC2650	Arduino Uno R3
Bluetooth Low Energy	Yes		No
Pin Requirement	Yes		
Applications	Wearables: Fitness & Clinical	Consumer electronics and medical	Consumer Electronics
Price ($) per unit	9.04	30.00	25.00
Available Resources	Low amount or resources online	Large community of WPI students	High amount of resources online

Initially, the team selected the CC2650 Launchpad for the device due to the large community of users at WPI. However, after encountering multiple problems when trying to control and display shirt functions through Bluetooth, the team chose to switch to the Arduino Uno R3. The Arduino is paired with the Bluefruit LE nrF8001 to transmit data over Bluetooth low energy.

4.2.5.2 Specifications

The final product uses the Arduino Uno R3 (Atmega328 - assembled) to process the data and the Bluefruit LE nrF8001 to transmit the data wirelessly over Bluetooth low energy. The data can be visualized from an IOS and Android App.

4.2.6 Power System

4.2.6.1 Background

An integral part of our smart shirt design was the power source. The team needed a power supply which would not only be able to power the whole system, but also keep it running for a reasonable amount of time, feel lightweight enough for an individual to carry on their body for a long period of time, as well as being cost effective. Table 15 displays the current requirements of each module of the smart shirt system as well as the maximum current draw, 809.5mA.

Table 15. Current Draw of the System

	Current Draw
IMU	6.5mA
FSR	0.8mA
ECG	0.2mA
Vibration System	140mA
Air Pump System	650mA
Microcontroller	12mA
Total	809.5mA

The team also required that the power supply have USB outputs to power the microcontroller. Table 16 shows a comparison of four battery packs that meet these criteria.

Table 16. Battery Pack Comparisons

Product	DULLA M50000 Portable Power Bank	RAVPower External Battery Pack 16750	Anker Portable Charger PowerCore 20100	Aibocn Power Bank
Current Capacity	12Ah	16.75Ah	20Ah	
Cost ($)	20	30	40	20
Weight	12oz	-	12.56oz	17.6oz
Dimension	3.1 x 6.1 x 0.5 in	5 x 0.79 x 3.15 in	6.5 x 2.3 x 0.9 in	7.6 x 4.4 x 1.3 in
# USB Ports	2			
USB output current values	2A	2.25A	2.24A	2.1A, 1A
USB output voltage	5V			

Based on the above specifications the team selected the Anker Portable Charger PowerCore 20100 to power the system.

4.2.6.2 Specifications

The power supplied by the power bank is separated by the two USB ports of the module. One USB port supplies power to the microcontroller. The microcontroller supplies the IMU, ECG, FSR and the Bluetooth Module. The other USB port is used to create a 5V power rail by stripping a micro-USB cable. This power rail supports the air pumps and the vibration motors.

4.2.7 Bluetooth Low Energy

The Arduino itself does not have Bluetooth capabilities, however to control and visualize the various functions of the smart shirt system, the team decided to use Bluetooth Low Energy (BLE) to send data to a phone app. The team reviewed two Bluetooth modules, the RedBear BLE Shield and the Bluefruit LE module sold through Adafruit. Both were very similar, however the decision to choose the shield over the module was due to the additional space the shield made available on the PCB. Table 17 compares the specifications of the BLE modules compared.

Table 17. Comparison of BLE Modules

Bluetooth Module	RedBear BLE Shield	Bluefruit LE
Price ($)	19.90	19.95
Design	Shield	Breakout board
Chip	nRF51822	nRF8001
Dimensions		29mm x 28mm
Power	3.3V or 5V	5V

5. Hardware Design

This chapter details the hardware side of the system design of the Smart Shirt and elaborates on each system block.

5.1 Overall Design

Figure 4 depicts the overall block diagram of the smart shirt system. It is separated by modules that input data to the microprocessor (pressure sensor, IMU, ECG), modules that output a response (vibration motors, air pressure system), the Bluetooth receiver, and how power is supplied throughout the system. Each module and its specifications are elaborated upon below.
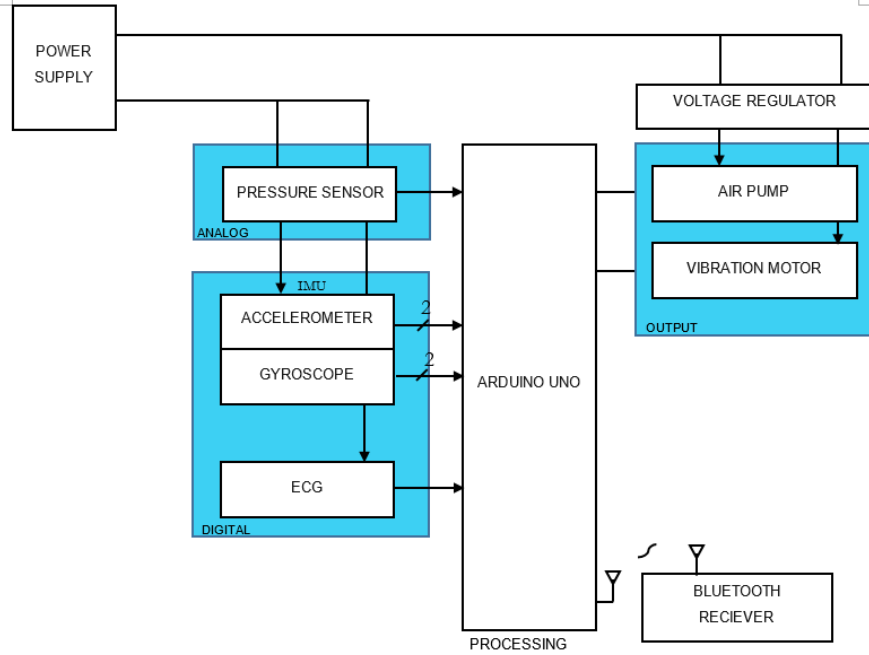
Figure 4. System Block Diagram

5.2 System Blocks

5.2.1 Power Supply: Anker PowerCore 20100

The Anker PowerCore 20100 is used to power the whole shirt system. It supplies a max of 5V out through two USB ports which both carry 2.4A max. In total the system can provide power for 20100mAh.

In our design we utilize both USB ports; port 1 powers the Arduino Uno microcontroller via USB B, port 2 uses a micro-USB cable to both supply 5V to the air pump system and to a voltage regulator which in turn provides 1.5V to the vibration system. Figure 5 below shows how the power is being distributed throughout the system.
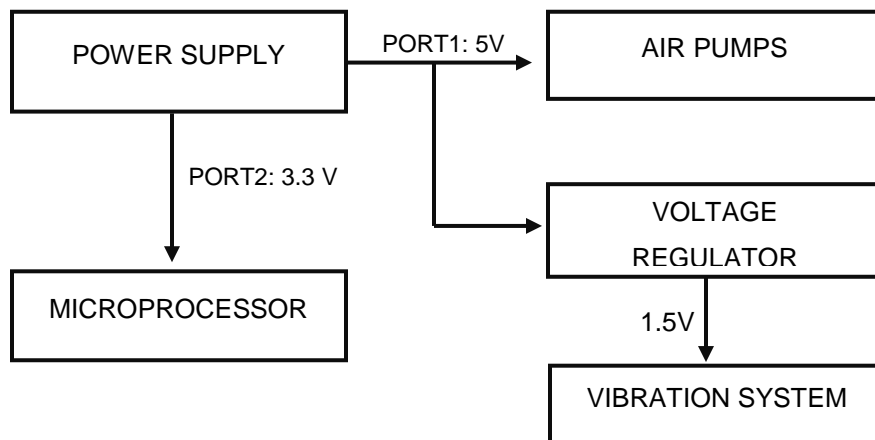
Figure 5. Power Block Diagram

5.2.2 Arduino UNO

To control the system, we used an Arduino UNO microcontroller. The first USB port of the Anker power bank powers the Arduino, which supplies power to the FSR, IMU, BLE shield, air pump motors, and vibration motors. Our system uses three analog inputs and 12 digital inputs of the Arduino. Analog pin A3 is used to read the change in voltage from the FSR, and pins A4 and A5 are used for the IMU's SDA and SCL ports respectively. Digital pins PB0-PB5 are used to connect the Bluetooth capabilities on the BLE shield to the Arduino. Digital pins PD6, PD7, and pins 3-5 are used to control the air pump and vibration motors via Bluetooth, and pin 2 is used as the output for the ECG transceiver.

5.2.3 Inertial Measurement Unit (IMU)

In the smart shirt system, the LSM9DS0 IMU, which consists of a 3-axis accelerometer, gyroscope, and magnetometer, monitors the user's body orientation. The IMU will be placed near the waist and calibrated using that point as the zero. The chest orientation is measured by the IMU and if deviations of more than a specified amount is detected in the x, y, or z direction, a vibration motor may be triggered on whichever side posture correction is needed. The data from the IMU can additionally be utilized in sports performance enhancing situations. An athlete may track their body position and acceleration during a specific action and use that information to improve their performance.

The microcontroller supplies the IMU the necessary 3.3V to function. Data is sent to the Arduino through Inter-Integrated Circuit (I2C) protocol. The data line connects to pin SDA and the clock like connects to pin SCL.
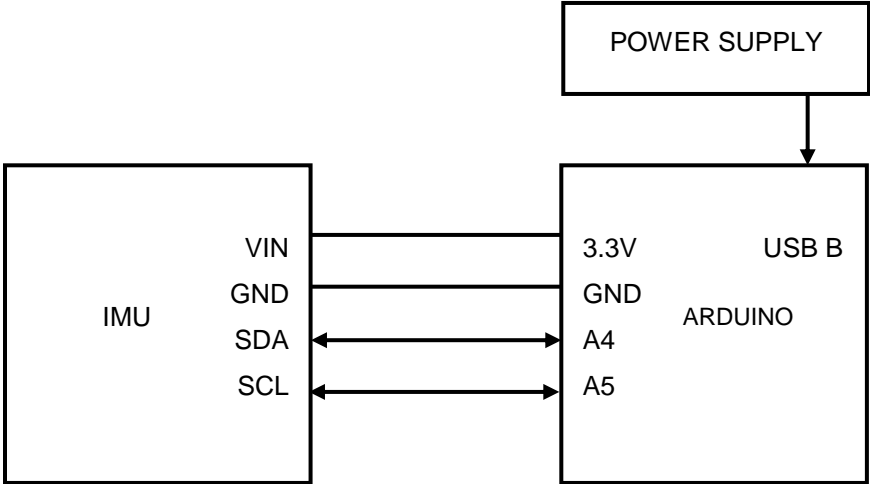
Figure 6. IMU Block Diagram

5.2.4 Heart Rate Sensor

The heart monitoring component of the design uses the Polar T34 heart rate strap, which sends logic level signals to the Polar Heart rate receiver, shown below in Figure 7.The microcontroller supplies the receiver 3.3V and the receiver outputs a logic high at 3.3V or a logic low at ground. The output connects to the digital input pin of the Arduino.
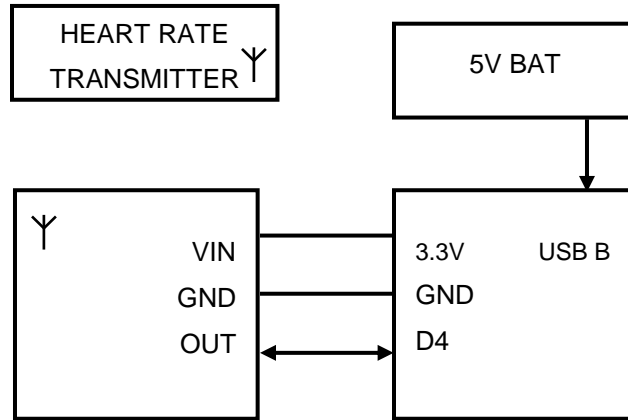
Figure 7. ECG Block Diagram

5.2.5 Pressure Sensor (FSR)

To determine whether the user is self-harming, the team used a force sensitive resistor, a specialized type of pressure sensor, connected in a voltage divider circuit, shown below in Figure 8. The FSR acts as R_1 (the resistor closest to the input voltage) and a reference resistor acts as R_{REF} (the resistor closest to ground). R_{REF} is used to measure the voltage drop. The Arduino provides 3.3V to the FSR, and the analog input of the Arduino is connected to the reference resistance to measure the change in voltage caused by a change in resistance from the FSR. To find the optimal Rref value that allows us to measure the greatest change in voltage when pressure is applied to the FSR, the following equation was used

$$\sqrt{\left(R_{FSR(\text{min})}\right)\left(R_{FSR(\text{max})}\right)} = R_{REF} \tag{1}$$

Figure 8 shows block diagram of the FSR. See Appendix G for the full derivation of the equation.
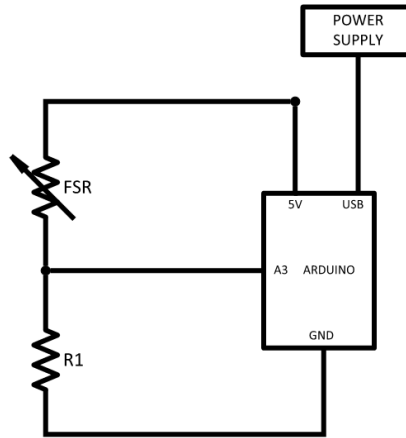
Figure 8. FSR Voltage divider Circuit Diagram

The FSR is composed of velostat, copper sheets, copper tape, and clear packaging tape. Velostat is a piezoresistive material with a volume resistance of <500Ω-cm. Its resistance decreases when pressure is applied. The copper tape acts as a conductive layer for the FSR. To test the FSR designs, two pieces of copper sheets were used to make tabs to attach alligator clips. To construct the FSR, the team used three layers of velostat to create a sandwich configuration, two pieces of copper tape on each side with the copper tabs, and put it together using the clear packaging tape. Appendix A shows and describes each FSR design along with each of their test measurements using the CC2650's ADC. Using Equation 1, Figure 9 shows the graph of the best reference resistor value to use in the voltage divider circuit with FSR #3.
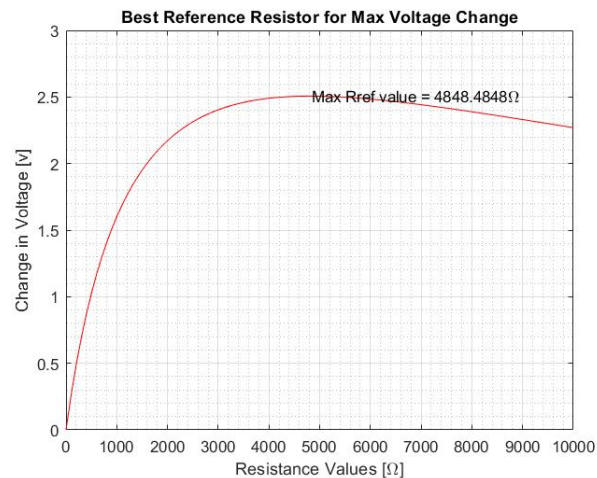
Figure 9. Best Reference Resistor

5.2.6 Compression System

The compression system consists of two thigh blood pressure cuffs that wrap around the user's waist. The team used two cuffs as one cuff was not long enough to wrap around the waist of an average individual. There are two tubes connected to each cuff; one tube pumps air into the device by using 4 mini air pumps, and the other connects to a release valve that allows the user to manually deflate the cuffs.

The air pumps are supplied 5V from the USB power rail. One pin of the air pump is connected to the 5V and the other pin is connected to the drain of a MOSFET. The source of the MOSFET is connected to GND and the gate of the MOSFET is connected to a digital output pin. The MOSFET acts as a switch to turn the air pump motors on and off.

5.2.7 Vibration System

The vibration system is composed of four coin-vibration motors controlled by the Arduino via Bluetooth. Each vibration motor is supplied 1.5V from the LM317 voltage regulator. A 0.1uF capacitor is connected from the input to ground. A 250Ω resistor is connected to the adjust pin and output pin of the regulator and a 40Ω resistor is connected from the adjust pin to ground.

The 1.5V from the voltage regulator connects to the positive lead of the motor to and the negative lead of the motor is connected to the drain of a MOSFET. A diode is connected in parallel to the microcontroller to prevent sudden current spikes the source of the MOSFET is connected to GND and the gate of the MOSFET is connected to a digital output pin. This allows the microcontroller to control the ON or OFF state of the motors.

5.2.8 BLE Shield

The BLE Shield (nRF51822), produced by RedBear, looks like the Arduino Uno board and can be placed directly on top of the Uno. Various headers connect the shield to the Uno board, and the shield itself contains the same headers of the Uno, allowing the user to access the same ports as the board. Using the Nordic nRF51822, the shield can send data using Bluetooth Low Energy to Bluetooth enabled devices. For our purposes, the BLE Controller app will be used to control the various functions of the board. Data can also be visualized on this or other apps, however for our purposes, graphs and tables will be displayed on a computer with that information being sent through USB-B.

5.2.9 PCB Design

To decrease the number of floating wires in our system and create a prototype that is more reflective of what an on-market product would look like, the team decided to create a PCB that houses all of the components used in our system. The PCB will sit on top of the BLE shield, and system outputs will be connected through a ribbon cable to increase connection stability between external outputs and board fastening.

5.2.10 Shirt/Component Housing

For the heart rate monitor of our system to record measurements accurately, the device needs to sit flush against the chest of the wearer, as close to the skin as possible. To achieve this the team decided to use a compression shirt; they provide a tight fit to the user, while remaining comfortable enough for intense athletic activity. External pockets will be sewn on to the shirt to provide a secure means of containing the power source and microprocessor of the system.

6 Software Design

This chapter details the software side of the system design of the Smart Shirt and elaborates on each system block.

6.1 Overall Design

The overall design of our code architecture follows the requirements of the Arduino. The Arduino requires two functions to be present in each program. The first function is called "setup", it is only called once and is used to conduct initializations. The second function is called loop and is repeatedly executed until the Arduino is powered off. Our loop function reads the IMU sensor data and the FSR first, then checks for any incoming messages from the app. The function then parses the message and determines what to send back to app.

To process time sensitive data, our system also incorporates interrupt service routines (ISR). The interrupt service routines interrupt the loop function at a fixed time interval and executes the code in the ISR function. Our code must be able to keep track of time to process the air pumps, FSR, and control the Bluetooth display. To do this, we created an ISR using Timer1 to occur every half of a second.

Additionally, our system must be able to process the time between the pulses from the heart rate monitor. Initially, we added a second ISR, which interrupted the loop each time the digital input pin for the heart rate monitor changed from low to high. However, this ISR was not accurately displaying the timer. Therefore, we added a second timer interrupt using Timer2 to occur approximately every 1ms. In this ISR, we poll the value of the heart rate monitor pin to calculate the beats per minute. Appendix C: Code Flowchart displays the full block diagram of the system.

6.2 Software System Blocks

6.2.1 Self Harm Detection

The input of the force sensitive resistor connects to the analog input of the Arduino microcontroller. The analog input of the Arduino converts the voltage read from the FSR voltage divider and provides values between 0 and 1023. Table 18 shows the ADC count values matched with its corresponding voltage. The system's purpose is to read the analog values and detect whether self-harm was detected.

Table 18. ADC Count to Voltage

ADC Count	Voltage [V]
0	0
1	0.004887
2	0.009775
...	...
1022	4.995112
1023	5

To accomplish this, we recorded various types of hits on the FSR to visualize the waveform in the case of repetitive hits to the device. We graphed the waveforms using the serial plotter on the Arduino IDE. This allowed us to see the ADC count values from the FSR in real time. While using the serial plotter, we saved screenshots of the waveforms in order determine the system's waveform for different states of the FSR. This included the waveform of the system at rest, light movement, and different taps. Appendix B: FSR Waveforms contains the full gallery of these waveforms.
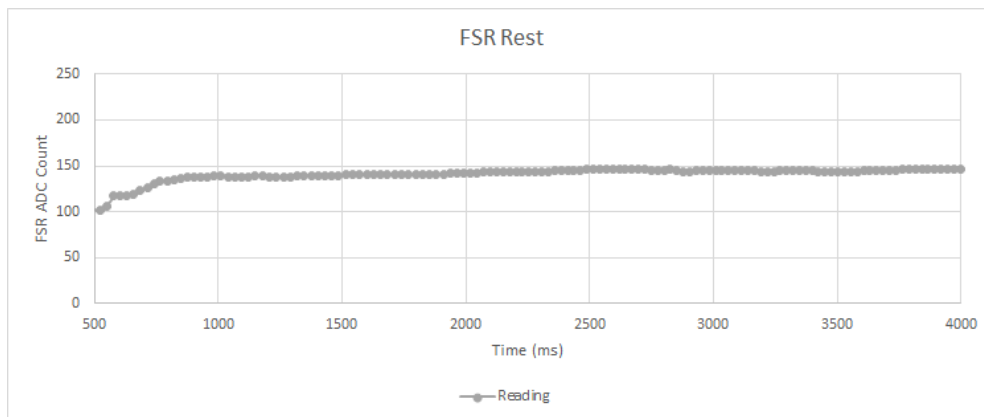
Figure 10. FSR waveform Rest

Figure 10 shows an example of the system at rest and when hit repetitively with force; the waveform displays the ADC counts versus time. The FSR without any force shows a constant waveform. We observed that if a constant force is applied to the FSR, the waveform will consistently increase, as shown in Figure 11.
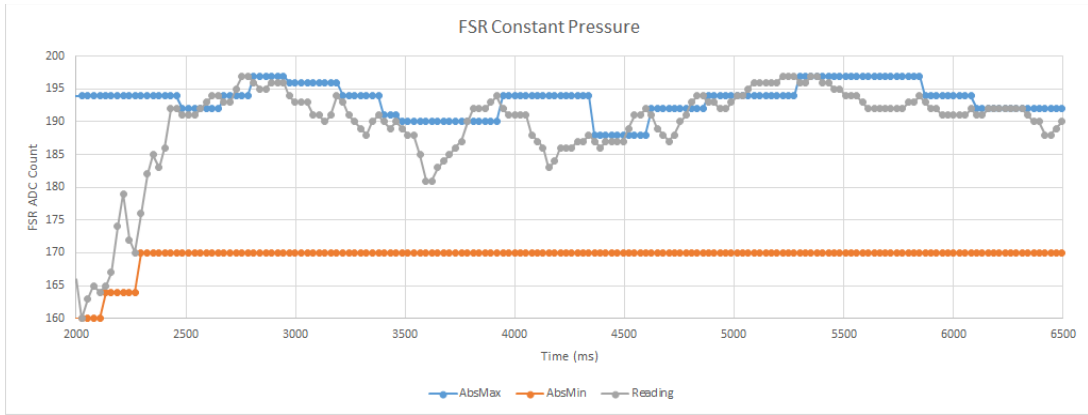
Figure 11. Constant Pressure

The graphs of the FSR while being repetitively hit displayed a waveform with multiple peaks and lows. Figure 12 shows the difference between slow taps and faster taps. The waveform with the slower hits displayed a larger difference between peak and low values, while the waveform with the faster hits displayed smaller differences. Because we had no way of measuring the force of the hit, we only used the speed of the hit to develop a simple algorithm. From the waveforms in Figure [] and in Appendix B: FSR Waveforms, we determined that the average difference between the peaks is 40 ADC count values.
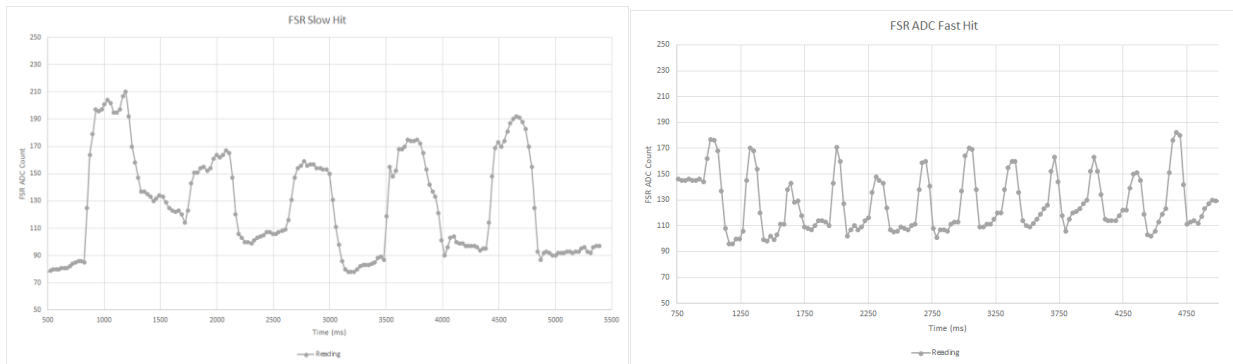
Figure 12. FSR waveform Slow vs. Fast Hit

We developed a simple algorithm to detect repetitive force on the FSR based on the minimum and maximum points on the waveforms. Appendix C: Code Flowchart contains a flowchart of the design.

The first step is to detect the absolute maximum and the absolute minimum points of the waveform. We define absolute minimum and maximum as the point where the graph changes

40

direction from increasing-to-decreasing for an absolute maximum or decreasing-to-increasing for an absolute minimum. To find the absolute maximum value of the waveform, we check if the current FSR reading is greater than the past FSR reading. If it is, then we check if the difference between the previous minimum value, and the current reading is greater than 15 ADC count values.
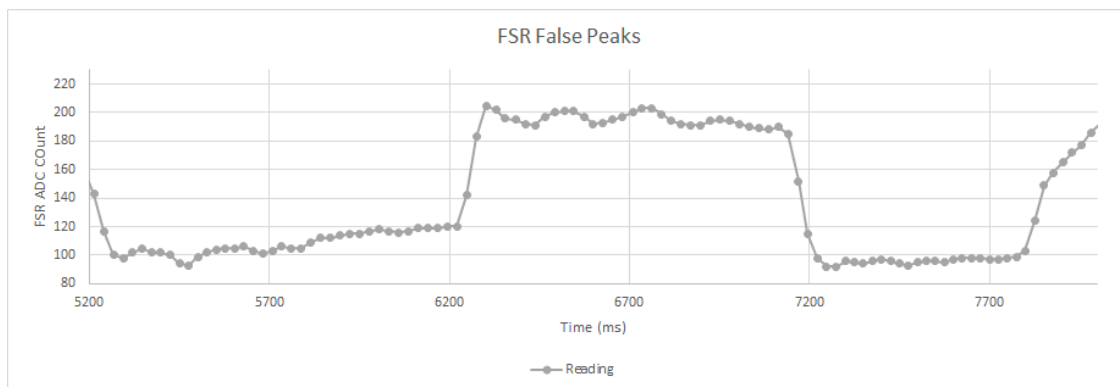
Figure 13. FSR Noise

When observing the FSR waveform, we noticed that the FSR values do not increase or decrease smoothly due to noise in the signal, as shown in Figure 13. In order to eliminate false peak or trough values, not due to a hit, we check to see if the difference between the readings is at least a15 ADC count values. By observing the waveforms in Appendix B: FSR Waveforms, we were able to discern that at least 15 ADC counts corresponds to a self-harming scenario. If the difference between the previous minimum value, and the current reading is greater than 15 ADC count values, then we set the local maximum value of the FSR to the current reading. We set the absolute maximum value equal to the local maximum value when the FSR values begin to decrease.

To detect the absolute minimum value of the FSR, we check if the current FSR is less than the previous FSR value. If it is, then we check if the difference between the FSR maximum value and the current value is greater than 15 ADC count values. If the difference between the previous maximum value, and the current reading is greater than 15 ADC count values, then we set the local minimum value of the FSR to the current reading. We set the absolute minimum value equal to the local minimum value when the FSR values begin to increase.

The next step is to check whether the maximum value or the minimum value was the same as the previous maximum or minimum value. If the maximum or the minimum value is constant, this means that the FSR value is either steadily increasing, steadily decreasing, or staying the same value, which does not happen when repetitive hits occur. If this is true, a counter is started to check

41

if the system maintains this state for 2 seconds. If the system is at rest for two seconds, then the systems flag to check for check for hits gets turned off. However, if the previous maximum value or the previous minimum value changes, then the system turns the flag to check for hits back on.

When the system flag for checking for hits is on, the system turns on a flag to check how long the system is checking for. We take the difference of the maximum and minimum points in the waveform over a 7 second window and save a running average of the values. At the end of the period, we check if the average difference between these points was greater than 40 analog values. If true, then this is detected as self-harm.

6.2.3 Heart Rate Calculation

The heart rate values are inputted from the heart rate receiver into a digital input pin on the Arduino. A logic level high determines a pulse while a logic level low signifies rest. To read the signal, we check for the value of the Arduino pin every millisecond. If the current state of the ECG is a logic high, and the current state is not equal to the previous ECG state, we save the number of milliseconds between the first pulse and the last pulse.
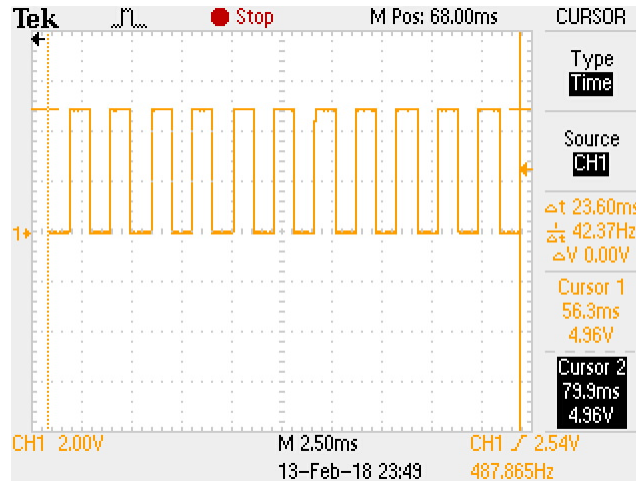
Figure 14. 1ms Interrupt

Figure 14 shows a signal of pulses of the 1ms interrupt. The oscilloscope image captured 23 clock cycles and each clock cycle occurred every 1.026 seconds. This shows that the interrupt does not occur at exactly 1ms. To compensate for this this delay in our design, we multiply the number of milliseconds passed by an offset. Then, we calculated the pulse by converting the time from milliseconds into seconds. Lastly, we divided the time in seconds from 60. This calculates the user's

heart rate in beats per minute. The offset for our final design is 1.023, elaborated upon in 8.1 Heart Rate Calculation. Appendix C: Code Flowchart contains a flowchart describing this design.

6.2.4 Compression System

The gate of the MOSFET that controls the air pumps in the compression system connects to a digital output pin on the Arduino. The initial state of the digital output pin is 0V. When the compression system flag is turned on, the digital output pin is set to 5V. This turns on the air pumps. Once the air pumps are turned on, the system starts a timer that counts to 30 seconds. After 30 seconds, the digital output pin is set to 0V, turning the motors off.

6.2.5 Vibration System

The Arduino controls the vibration motors individually by connecting the MOSFET gate of each motor to a digital output pin. The digital outputs pins are initially set to 0V. The vibration motors are turned on by setting the pin to 5V when the user signals to turn on a vibration configuration from the App. The vibration motors are turned off when the user signals to turn the vibration off from the App. This sets the digital output pins to 0V.

6.2.6 Posture Tracking

The posture tracking system uses the IMU to measure the movement of the user. We use the Adafruit LSM9DS0 library to configure and use the device. In the Arduino setup function, we initialize the accelerometer, gyroscope and magnetometer by using predefined functions.
In the Arduino loop function, we read the data from the IMU by using the "getEvent" function from the Adafruit libraries. This function returns a float variable for the x, y, and z coordinates of the accelerometer, gyroscope, and magnetometer. We save these values into global variables to be used for future use.

6.3 Bluetooth Integration

To demonstrate wireless capabilities, we added the Red Bear BLE Shield. The shield uses the nRF8001 BLE chip to enable Bluetooth capabilities on the Arduino. The shield connects to an android and IOS app called "BLE controller". We used the "Simple Chat" functionality on the BLE controller APP to implement communication between a smartphone and the device. The simple

chat functionality allows the device to send text messages to a smartphone and allows the user to send messages to the device to control functionalities.

The Arduino code implements this by polling in the continuous loop function for incoming messages from the app. If a message is received, we use a state machine to check if the message requires a system change. If the system does not recognize the message, the systems prompts the user to enter a known command.

6.3.1 Self Harm Detection

The main loop checks for the flag that the system has detected self-harm. If this flag set off, we send the message displayed in Table 19 to the app. The FSR readings can be accessed using the command "FSR".

Table 19. Self-Harm Messages

Self-Harm Detection Alert
"Self-Harm Detected"

6.3.2 ECG

For the user to read their heart rate, they must send the control in Table 20. If this message is received, the system sends the value of the heart rate in beats per minute.

Table 20. ECG Controls

ECG Controls
"HR"

6.3.3 Vibration System

The Arduino controls the vibration motors through the Bluetooth commands in Table 21.

Table 21. Vibration Controls

Vibration Controls	
V1	Toggle vibration motor 1
V2	Toggle vibration motor 2
V3	Toggle vibration motor 3
V4	Toggle vibration motor 4
VO	Turns off the vibration motors

6.3.4 Compression System

The Arduino acts as switch that turns the compression system on and off by using the smartphone app. Table 22 shows the controls for the system.

Table 22. Compression System Controls

Compression System Controls	
CON	Turn the compression system on

7 Construction

After an extensive period of component research and design development the team proceeded to begin constructing the various parts of the smart shirt system. For some modules, such as the IMU, heart rate monitor, and BLE shield, component construction was not necessary; these components were supplied in the form of pre-constructed breakout boards and needed code to be initialized to start using the system. For the FSR, vibration motors, second power port, voltage regulator, and air pumps, it was necessary to both construct the circuits for the respective components and create the code which would allow the system to perform as expected.

7.1 Pressure Sensor (FSR)

The team utilized multiple online videos and resources to provide us with a foundation on how to construct a working and accurate FSR [56]-[58]. Because there were numerous ways to construct an FSR, the team tried six different designs to produce a comprehensive scope of possible designs, which are described in Appendix A

Out of the six designs, only two performed well enough for testing purposes. FSR #2 and #5 had poor readings due to being bonded using duct tape instead of clear packaging tape. FSR #4 had too much of its area covered with the copper tape, causing the measured voltages to be low with little variance. Out of the six designs, FSR #3 had the best response and resistance range, which can be seen in Appendix F.

7.2 Vibration Motors

Once purchased, connecting the vibration motors to the system was easy. With one wire for Vin and the other for GND, the team simply connected the wires to their respective sources. Because the wires of the vibration motors were relatively small the team reinforced the wires by enclosing the connection between the vibration motors and ribbon cable with heat-shrink tubing Figure 16.
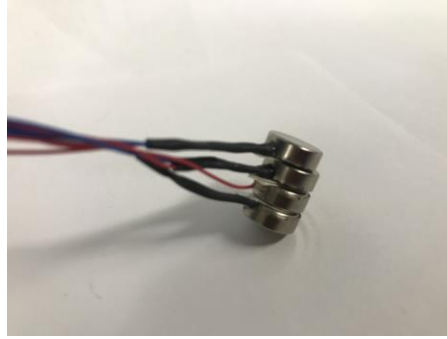
Figure 15: Vibration Motors

7.3 Power (Voltage Regulator, USB2)

Powering the Arduino board itself simply required the use of a USB-B cable, which connected the board to the Anker Power Supply. To ensure we had a comprehensive understanding of how the power source worked, we measured its V-I characteristics. Figure 16 shows the measured voltage of the battery at full charge, half charge, <25% charge, and in an open circuit. As shown, the actual voltage is slightly greater than specifications, measuring at 5.15V instead of 5V. The internal resistance of the battery was measured to be .65 Ohms.
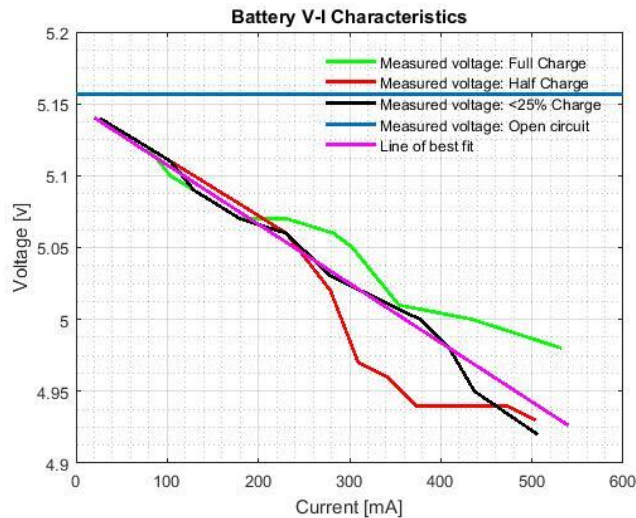
Figure 16. Battery V-I

The vibration motors and air pumps are supplied separately from the board supply. The power bank came with two USB outputs; initially the team used the second one to create a 5V source by stripping a USB cable, as shown in Figure 18, and powering a breadboard. However, per our advisor's recommendation, our final PCB design incorporates a micro-USB port to create a more secure 5V power supply connection.

Figure 17. Power Cable-Modified for Breadboard

The vibration motors used in the system do not support 5V therefore a voltage regulating circuit was necessary.

7.4 Compression Design Air Pumps

To construct the compression system of our vest, the team decided to use a blood pressure cuff as the model for our design. To wrap around an average adult's waist, the team used two thigh cuffs, cut and sewn together as shown in Figure 18. The team used a blood pressure cuff designed by students at Cornell University [59] to help guide the design.

Figure 18. Combined Blood Pressure Cuffs

The team was initially going to use a pressure transducer to regulate the airflow of our circuit, but after weeks of our component not arriving, the team began to explore other options of regulating inflation and deflation. From this, the team realized that the pressure transducer was not

necessary, instead a physical valve would be used to prevent air from escaping when the cuffs where inflating, and the same valve would be opened to deflate the cuff when necessary.
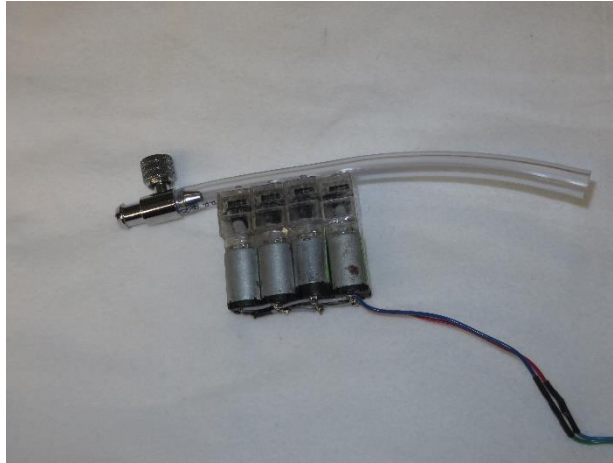
Figure 19. Air Pump Tubing

To inflate the vest the team originally used one air pump. The single pump took over 4 minutes to inflate the vest, and the compression produced by the single pump was not enough to create any noticeable compression. To increase compression and decrease inflation time, the team decided to use all five of the air pumps purchased in the system. To attach each air pump in a secure manner that prevented air from escaping, the team designed the system shown in Figure 19.

7.5 Overall Smart Shirt Design

Once all the functions were completed, we started putting the entire system together. Figure 20 shows the sketch of the shirt and Figure 21 shows the final shirt design.
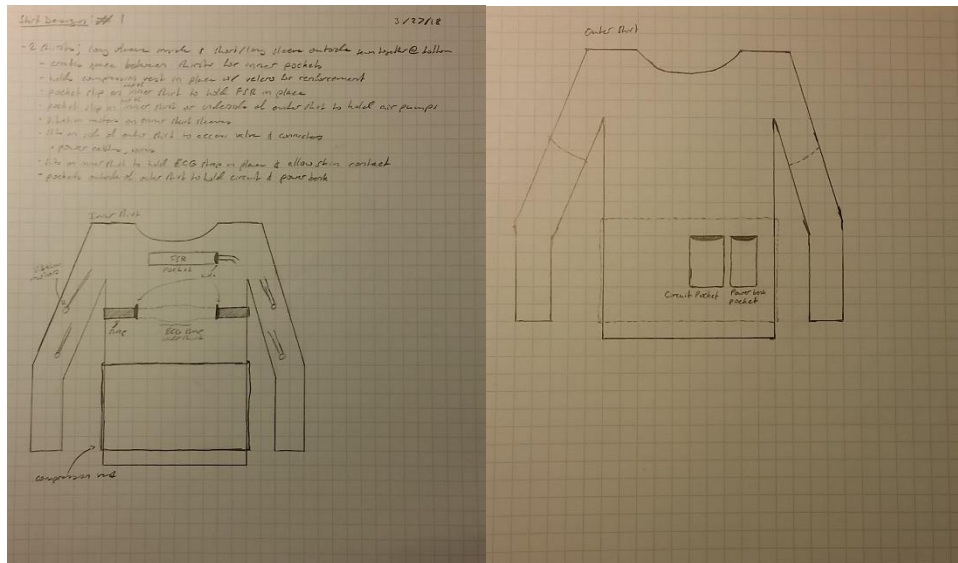
Figure 20. Smart Shirt Sketch Design
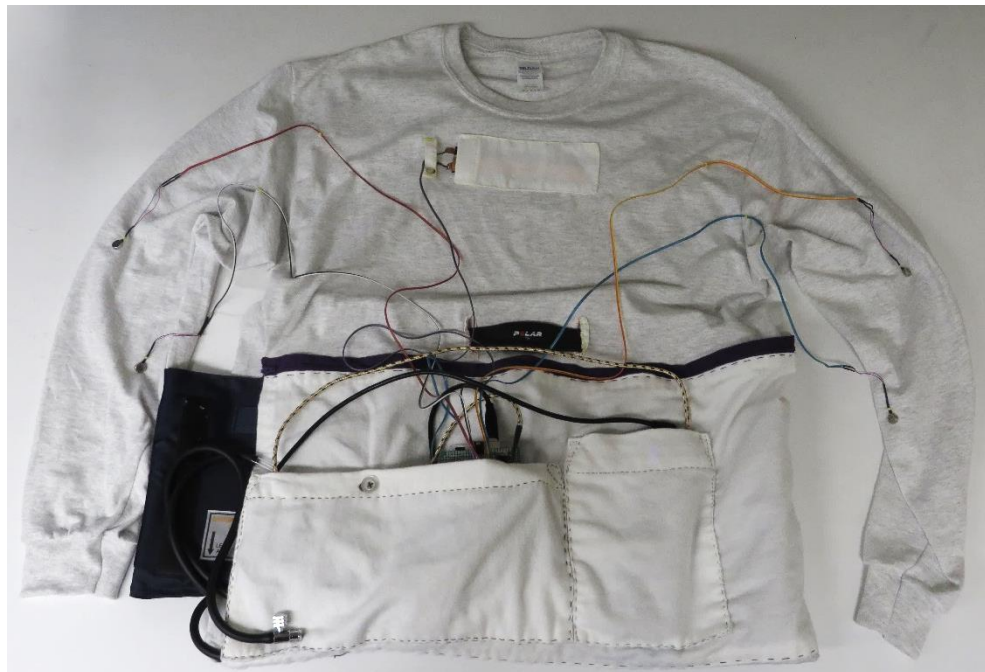
Figure 21. Final Smart Shirt Design

We purchased a long-sleeve T-shirt as the base for the system. The long sleeves would allow the vibration motors to be stationary while someone is wearing the shirt. We sewed the vibration motor wires to the shirt to keep them from moving excessively, but loose enough to adjust the lengths from the arms to the Arduino. To keep the thigh cuffs in place without making them too tight on the wearer, we extended the bottom of the shirt with fabric and brought it up to the body midway. The extended portion was sewn in place at the top with zippers to allow the wearer to

remove the thigh cuffs before taking off the shirt. On the front of the extended portion of the shirt, we sewn in two pockets, the smaller pocket to hold the power bank, and the larger pocket to hold the Arduino and circuit. We sew in a button for the larger pocket to hold in the circuit more securely, as the pocket was too large and hung out when someone was wearing it. The larger pocket also holds the four air pumps. Right above the zipper is where the ECG strap is placed. Two holes were cut into the shirt to allow the wearer to place the ECG strap on their skin without having to put it on prior to the shirt. The FSR is kept in place with its own little pocket on the chest area of the shirt, along with its own little button and strap to keep it stationary.

8 System Testing and Results
8.1 Heart Rate Calculation

To test this system, we inputted a constant pulse from a function generator into the Arduino to imitate a heart rate signal. We tested heart rate measurements from 30 bpm to 200 bpm which corresponds to a frequency range from 0.5Hz to 3.33Hz. We calculated the expected heart rate by using the following equation:

$$HeartRate = Frequency * 60 \qquad (2)$$

Then, we measured the heart rate but reading the output from the BLE application. Figure 22 shows the measured bpm minus the expected bpm versus the frequency by using the offset of 1.026. We chose to test the offset value of 1.026 first because this is the measured interrupt speed shown in Figure 14. This shows that as the frequency increases, the accuracy of the heart rate value in bpm decreases linearly.
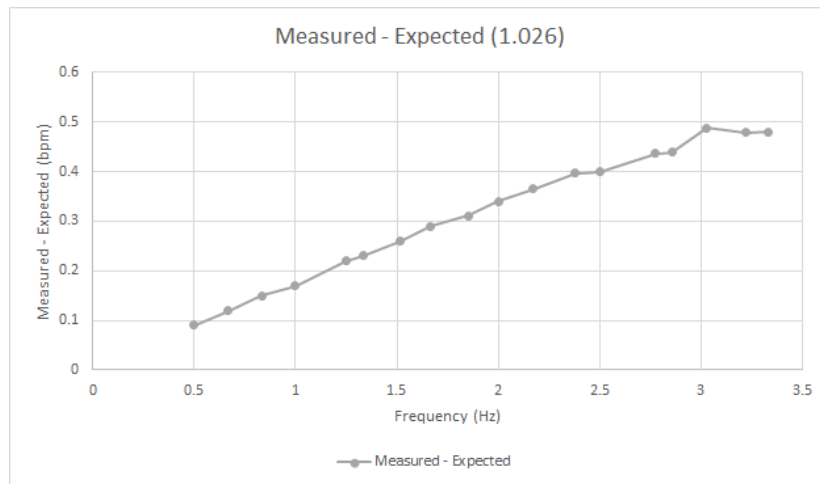
Figure 22. Measured – Expected (1.026)

To find the best offset for the range, we used **Error! Reference source not found.** to find the millisecond count from the Arduino. Then, we used **Error! Reference source not found.** to find the correct offset that would generate the expected heart rate in bpm. The full calculation is in Appendix D: ECG Tests

$$Arduino\ Millisecond\ Count = \frac{60000}{Measured\ HeartRate * Offset} \qquad (3)$$

$$Offset = \frac{60000}{Arduino\ Millisecond\ Count * Expected\ Heart\ Rate} \qquad (4)$$

After calculating the new offset values for each data point, we found that the average offset is 1.023. Figure 23 shows the Measured minus Expected bpm versus the frequency of the signal using the 1.023 offset. This shows that the error is no longer linear or predictable.
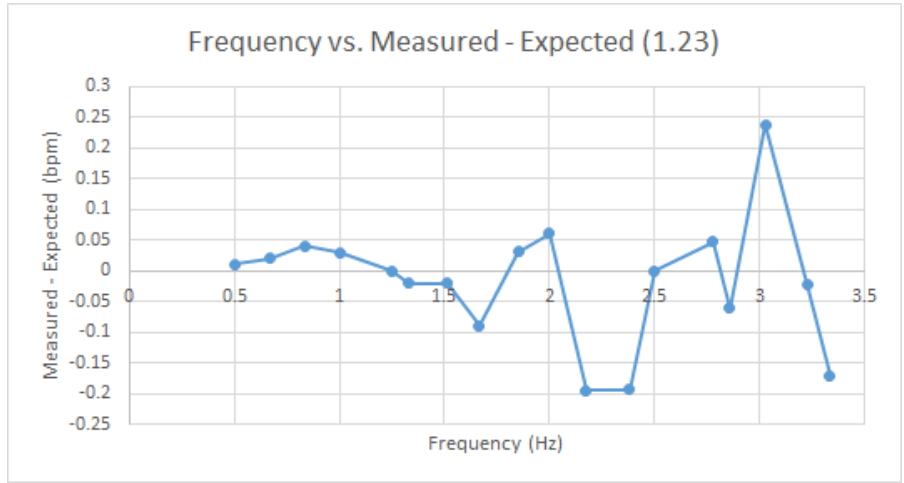
Figure 23. Measured – Expected (1.023)

8.2 Movement Tracking

We tested the movement tracking function of the device by attaching the completed proto-board to the waist. With the device attached to the body, we connected the Arduino to the serial monitor program called "Coolterm", which allowed us to save samples of the IMU data outputs into an excel sheet in real time. Appendix E: IMU Test contains the full gallery of graphs displaying the IMU data.

We configured the accelerometer to output data from +/-2g which is equal to +/- 19.62m/s^2. However, as shown in Figure 24 which shows the x, y, and z coordinates of the accelerometer, the data measured in meters per seconds squared exceeds the maximum and minimum values of the device. This shows that the data cannot be accurately used.
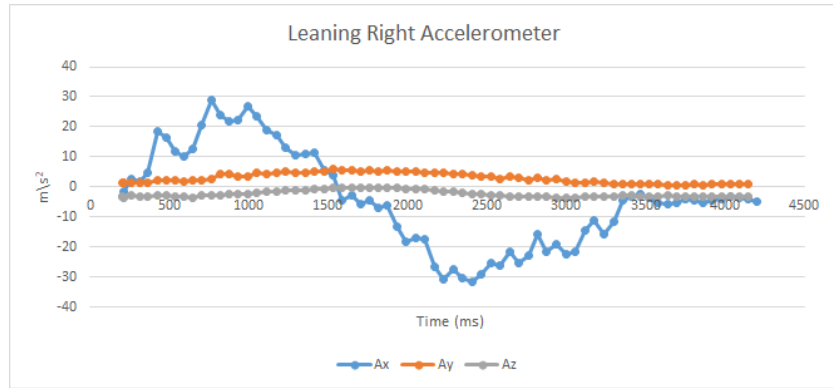
Figure 24. Accelerometer Leaning Right

8.3 Compression Vest

Using the compression command "CON" with the RedBear BLE app, the air pumps successfully activate and pump out air for one minute. The issue with the air pumps is that even with four, the system doesn't inflate the thigh cuffs enough to feel a noticeable difference within the one-minute operating period. This might be due to either the pumps not being powerful enough or how airtight the connection between the air tube and pumps are.

9 System Improvements

There were many things the team wished to accomplish during the scope of our project, however due to time constraints not all were possible. Below are the future improvements we believe will help to improve this project and assist in the development of a product ready for commercial production.

9.1 Microprocessor

At 68.6mm x 53.4mm the Arduino Uno is a comprehensive yet large microprocessor for use in wearable applications. To decrease the size of the microcontroller used while maintaining the same features and functionality, the team suggests the use of the ESP32; with a price point as low as $4.00 the ESP is not only significantly lower in cost and size (6mm x 6mm) than the Arduino and can be programmed using the same IDE, but its features also rival that of the Arduino as shown by Table 23.

Table 23. Arduino vs ESP32 comparison

Microprocessor	Arduino Uno	ESP32
WiFi		X
BLE		X
GPIO	14	32
Battery		Micro-B/LiPo
ADC	X	X
DAC		X
I2C	X	X
Clock Speed	16MHz	240MHz max
Additional Sensors		Touch, Temp, Hall Effect
Power	5V	3.3V
SRAM	2KB	520KB

While the ESP32 contains substantial advantages over the Arduino Uno, one drawback of this microprocessor is lack of fully documented peripherals with example code [60]. The ESP32 is target mostly towards developers and thus there are still some bugs that are being found and fixed; thus, if a future team were to use this microprocessor it would be key that one or more of the members were an expert in code development.

9.2 Power Supply

While the current battery used to power the system has a high capacity, it is very heavy and obtrusive. To reduce the weight the individual must wear, the team suggests either switching to a 6600mAh lithium ion battery, or, depending on how long the user wants to power each function of the vest, using a combination of a Li-Ion and rechargeable Ni-MH battery. Both suggested battery replacements are lighter in weight and smaller in size than the current battery pack used, and despite the NiMH battery having a low voltage, a series connection of 5 batteries would remedy the voltage obstacle.

Table 24. New Batteries

Full System Current Draw 842.7mA	Current Battery 20100mAh	Li-Ion 6600mAh Adafruit	Ni-MH 2850mAh 4-pack ANSMANN
Hours Powering Whole System	~24hrs	8hrs	~3hrs
Hours Powering Special Needs 696.2mA	~28hrs	~9hrs	~4hrs
Hours Powering Stroke Rehabilitation 161.7mA	~124hrs	~41hrs	~18hrs
Hours Powering Athletic Application 21.7mA	~926hrs	~304hrs	~131hrs
Weight	356g	155g	28g
Dimensions	6.5"x2.3"x0.9"	2.7"x2.1"x0.71"	3.5"x0.5"x5"
Price ($)	40	30	14
V_{out}	5V	3.7V	1.2V

9.3 Surface Mount Components & Breakout Boards

Currently, through hole components are used in our PCB design; these components are beneficial to use when soldering the connections on your own, however take up space and increase the size of the board. To help make the system more compact and conducive for wearable applications, the team recommends not only using surface mount components for the PCB design, but also eliminating the breakout boards in our system to additionally decrease system size.

Table 25. New Parts

Component	Dimensions	
	Through Hole	Surface Mount
Resistor	2.30mm x 6.00mm	1.00mm x 0.50mm
Capacitor	3.81mm x 3.05mm	0.60mm x 0.30mm
BLE Shield	95mm x 70mm x 30mm	3.50 x 3.83 mm

9.4 Compression System

For the product design described in this report, the compression function of the smart shirt is achieved by connecting four mini air pumps, designed for aquariums, into a tube which is fed into two thigh blood pressure cuffs. These pumps can inflate the vest for demonstration purposes, however they are noisy, slow, and do not achieve the range of compression that is wanted for comfort purposes.

To improve the overall compression, experience the team recommends constructing a new air pump/flow system. Instead of using multiple thigh blood pressure cuffs connected, it would be optimal to have a shirt with inflatable strips already integrated into them; this would easily allow for targeted compression. To improve rate and weight of compression, the team recommends the FlexTail air pump; the device can act as both a vacuum and a pump, is smaller than a coke can and is slightly lighter than a traditional smartphone. Despite these impressive features, the air pump remains noisy; thus, the team suggests research and testing on methods to insulate the pump to decrease sound output.

9.5 Expanding Applications

An additional niche this smart shirt could be applied towards is the military. Soldiers wear bulletproof vests, which can prevent external damage caused by a bullet, however it may be difficult to identify whether the impact was strong enough to cause internal bleeding or organ damage. It may be possible for the shirt to incorporate more advanced pressure sensors, as well as other health monitoring devices, that record information on the soldier's vitals, and is able to store this information for later examination by medics. This would help doctors determine which locations may be experiencing internal bleeding, thus helping prioritize medical treatment.

9.6 Appearance

Currently, the design of the smart shirt system is very rudimentary. Components are visible on the outside of the shirt, and there is no enclosure for the Arduino-BLE-Protoboard shield composite. For the future, the team suggests the wiring of the system outputs be incorporated with the shirt in a neater manner that decreases or eliminates the wiring that is visible. Surrounding the microprocessor and accompanying shields in an enclosure of some kind would both make the system more aesthetically pleasing and durable, along with printing a complete PCB of the circuit to ensure that the components don't disconnect.

9.7 Design Improvement

The present design of the smart shirt system was based on extensive research and personal experience. Despite this, because the purpose of the smart shirt is to act as a therapeutic device it is important to consult the individuals who will be utilizing the smart shirt to understand whether our proposed design and functions are relevant.

Bibliography

[1] N. Watkins and E. Sparling, "The Effectiveness of the Snug Vest on Stereotypic Behaviors in Children Diagnosed With an Autism Spectrum Disorder," *Behavior Modification,* vol. 38, *(3),* pp. 412-427, 2014. Available: http://journals.sagepub.com/doi/full/10.1177/0145445514532128. DOI: 10.1177/0145445514532128.

[2] A. Alajbegovic *et al,* "Post Stroke Depression " *Med Arch,* vol. 68, *(1),* pp. 47-50, 2014.

[3] WRITING GROUP MEMBERS *et al,* "Heart Disease and Stroke Statistics--2009 Update: A Report From the American Heart Association Statistics Committee and Stroke Statistics Subcommittee," *Circulation,* vol. 119, *(3),* pp. 480-486, 2009. Available: http://circ.ahajournals.org/cgi/content/extract/119/3/480. DOI: 10.1161/CIRCULATIONAHA.108.191259.

[4] I. Zavoreo *et al,* "Post-stroke depression," *Acta Clinica Croatica,* vol. 48, *(3),* pp. 329, 2009. Available: http://www.ncbi.nlm.nih.gov/pubmed/20055258.

[5] D. Sinyor *et al,* "Post-stroke depression: relationships to functional impairment, coping strategies, and rehabilitation outcome," *Stroke,* vol. 17, *(6),* pp. 1102-1107, 1986. Available: http://stroke.ahajournals.org/cgi/content/abstract/17/6/1102. DOI: 10.1161/01.STR.17.6.1102.

[6] R. G. Robinson and T. R. Price, "Post-stroke depressive disorders: a follow-up study of 103 patients," *Stroke,* vol. 13, *(5),* pp. 635-641, 1982. Available: http://stroke.ahajournals.org/cgi/content/abstract/13/5/635. DOI: 10.1161/01.STR.13.5.635.

[7] S. Paradiso and R. G. Robinson, "Gender Differences in Poststroke Depression," *The Journal of Neuropsychiatry and Clinical Neurosciences,* vol. 10, *(1),* pp. 41-47, 1998. Available: http://dx.doi.org/10.1176/jnp.10.1.41. DOI: 10.1176/jnp.10.1.41.

[8] S. Hama *et al,* "Depression or apathy and functional recovery after stroke," *International Journal of Geriatric Psychiatry,* vol. 22, *(10),* pp. 1046-1051, 2007. Available: http://www.ncbi.nlm.nih.gov/pubmed/17702056. DOI: 10.1002/gps.1866.

[9] (). *Electrocardiogram (ECG or EKG) .* Available: http://www.heart.org/HEARTORG/Conditions/HeartAttack/DiagnosingaHeartAttack/Electrocardiogram-ECG-or-EKG_UCM_309050_Article.jsp#.Wozb9KinG70.

[10] (). *Electrocardiogram (ECG, EKG).* Available: https://www.emedicinehealth.com/electrocardiogram_ecg/article_em.htm#ecg_interpretation.

[11] (). *Electrocardiogram.* Available: https://medlineplus.gov/ency/article/003868.htm.

[12] (Feb 22,). *ECG/EKG (electrocardiogram)*. Available: https://www.myvmc.com/banners-heart-health-centre/ecg-ekg-electrocardiogram/.

[13] (). *ECG.* Available: http://www.analog.com/en/products/application-specific/medical/ecg.html.

[14] Anonymous "Smart Wearable ECG EKG Monitor - QardioCore," .

[15] (). *HEXOSKIN FULL KIT - WOMEN'S .* Available: https://www.hexoskin.com/collections/all/products/hexoskin-performance-optimization-toolkit-women-s.

[16] (). *AliveCor.* Available: https://www.alivecor.com/.

[17] P. Greenland *et al*, "2010 ACCF/AHA Guideline for Assessment of Cardiovascular Risk in Asymptomatic Adults: Executive Summary," *Journal of the American College of Cardiology,* vol. 56, *(25),* pp. 2748-2764, 2010. Available: https://www.sciencedirect.com/science/article/pii/S0735109710037198. DOI: 10.1016/j.jacc.2010.09.002.

[18] (). *Coronary Heart Disease: Screening with Electrocardiography.* Available: https://www.uspreventiveservicestaskforce.org/Page/Document/ClinicalSummaryFinal/coronary-heart-disease-screening-with-electrocardiography.

[19] (). *What are the factors that put a pregnancy at risk?.* Available: https://www.nichd.nih.gov/health/topics/high-risk/conditioninfo/factors.

[20] (). *Pregnancy Complications.* Available: https://www.cdc.gov/reproductivehealth/maternalinfanthealth/pregcomplications.htm.

[21] (). *Risk Factors for Complications During Pregnancy.* Available: http://www.merckmanuals.com/professional/gynecology-and-obstetrics/high-risk-pregnancy/risk-factors-for-complications-during-pregnancy.

[22] (). *High-Risk Pregnancy - Overview.* Available: https://www.webmd.com/baby/tc/high-risk-pregnancy-overview#1.

[23] (Feb 20,). *High-risk pregnancy: Know what to expect.* Available: https://www.mayoclinic.org/healthy-lifestyle/pregnancy-week-by-week/in-depth/high-risk-pregnancy/art-20047012.

[24] (). Modoo, the world's smallest fetus monitoring patch. Available: http://www.modoomed.com.

[25] (June 1,). *Wearable device provides continuous fetal monitoring.* Available: https://www.reuters.com/article/us-israel-mobile-foetus-monitor/wearable-device-provides-continuous-fetal-monitoring-idUSKBN0OH2NF20150601.

[26] (). *TinyKicks.* Available: http://www.tinykicks.com/.

[27] T. Grandin, "Calming Effects of Deep Touch Pressure in Patients with Autistic Disorder, College Students, and Animals," vol. 2, *(1),* pp. 63-72, 2009. Available: http://online.liebertpub.com/doi/abs/10.1089/cap.1992.2.63.

[28] J. Stephenson and M. Carter, "The Use of Weighted Vests with Children with Autism Spectrum Disorders and Other Disabilities," vol. 39, *(105),* pp. 105-114, 2009. Available: https://link.springer.com/article/10.1007/s10803-008-0605-3#citeas.

[29] S. Hodgetts, J. Magill-Evans and J. Misiaszek, "Weighted Vests, Stereotyped Behaviors and Arousal in Children with Autism," *J Autism Dev Disord,* vol. 41, *(6),* pp. 805-814, 2011. Available: http://www.ncbi.nlm.nih.gov/pubmed/20839040. DOI: 10.1007/s10803-010-1104-x.

[30] J. Case-Smith, L. Weaver and M. Fristad, "A systematic review of sensory processing interventions for children with autism spectrum disorders," *Sage,* vol. 19, *(2),* pp. 133-148, 2015. Available: http://journals.sagepub.com/doi/abs/10.1177/1362361313517762.

[31] E. Morrison, "A review of research on the use of weighted vests with children on the autism spectrum," *Education,* vol. 127, *(3),* pp. 323+, 2007.

[32] E. Blake, C. Landsea and E. Gibney. Deadliest, costliest, and most intense U.S. tropical cyclones from 1851-2010 (and other frequently requested hurricane facts);2011 ASI 2188-15;NOAA-TM-NWS-NHC-6. National Weather Service. 2011Available: https://www.nhc.noaa.gov/pdf/nws-nhc-6.pdf.

[33] Anonymous "The global monsoon systems," .

[34] Chloe Farand, "At least 41 million people affected in floods in India, Bangladesh and Nepal, UN says," *The Independent (Online),* 2017. Available: https://search.proquest.com/docview/1933780244.

[35] Centre for Research on the Epidemiology of Disasters and United Nations of Office for Disaster Risk Reduction. The human cost of weather related disasters: 1995-2015. 2015Available: https://www.unisdr.org/files/46796_cop21weatherdisastersreport2015.pdf.

[36] (). *Amazon Echo.* Available: https://www.staples.com/Amazon-Echo/product_1749482.

[37] (). *Google Home.* Available: https://store.google.com/us/product/google_home.

[38] (). *Wink Hub 2.* Available: https://www.wink.com/products/wink-hub-2/.

[39] (). *Logitech Harmony Elite*. Available: https://www.logitech.com/en-us/product/harmony-elite.

[40] Anonymous "Millions Of Animals Living In World's Zoos And Aquariums To Be Afforded New Protections And Humane Standards With Launch Of Unique New Global Program," *American Humane,* 2016. Available: https://www.americanhumane.org/press-release/millions-of-animals-living-in-worlds-zoos-and-aquariums-to-be-afforded-new-protections-and-humane-standards-with-launch-of-unique-new-global-program/.

[41] (). *Zoo and Aquarium Statistics*. Available: https://www.aza.org/zoo-and-aquarium-statistics.

[42] A. Albasini, M. Krause and I. V. Rembitzki, Using Whole Body Vibration in Physical Therapy and Sport E-Book: Clinical Practice and Treatment Exercises. Elsevier Health Sciences, 2010.

[43] P. Cordo *et al*, "Treatment of Severe Hand Impairment Following Stroke by Combining Assisted Movement, Muscle Vibration, and Biofeedback," *Journal of Neurologic Physical Therapy,* vol. 37, *(4),* pp. 194-203, 2013. Available: http://ovidsp.ovid.com/ovidweb.cgi?T=JS&NEWS=n&CSC=Y&PAGE=fulltext&D=ovft&AN=01253086-201312000-00007. DOI: 10.1097/NPT.0000000000000023.

[44] B. Marconi *et al*, "Long-Term Effects on Cortical Excitability and Motor Recovery Induced by Repeated Muscle Vibration in Chronic Stroke Patients," *Neurorehabilitation and Neural Repair,* vol. 25, *(1),* pp. 48-60, 2010. Available: http://journals.sagepub.com/doi/full/10.1177/1545968310376757. DOI: 10.1177/1545968310376757.

[45] Nes, I J W van *et al*, "Short-term effects of whole-body vibration on postural control in unilateral chronic stroke patients: preliminary evidence," *American Journal of Physical Medicine & Rehabilitation,* vol. 83, *(11),* pp. 867-873, 2004. Available: http://www.narcis.nl/publication/RecordID/oai:repository.ubn.ru.nl:2066%2F57782. DOI: 10.1097/01.PHM.0000140801.23135.09.

[46] M. D. Rinderknecht *et al*, "Combined tendon vibration and virtual reality for post-stroke hand rehabilitation," in 2013, pp. 277-282.

[47] (). *Universal ECG Electrodes - Pack of 50 - EKG Heard Monitor Electrodes*. Available: https://www.amazon.com/Universal-ECG-Electrodes-Heart-Monitor/dp/B019MOJPKK.

[48] (). *ECG & EKG Electrodes*. Available: https://www.buyemp.com/category/ecgekg-electrodes1.

[49] (). *Stretch conductive fabric*. Available: http://www.inntex.com/default.asp?mod=product&product_ID=135&cat_id=89,106.

[50] L. Vojtech *et al*, "Wearable Textile Electrodes for ECG Measurement," *Advances in Electrical and Electronic Engineering,* vol. 11, *(5),* pp. 410-414, 2013. Available: https://search.proquest.com/docview/1499822401. DOI: 10.15598/aeee.v11i5.889.

[51] B. Taji *et al*, "An ECG monitoring system using conductive fabric," in 2013, pp. 309-314.

[52] (). *KardiaBand.* Available: https://store.alivecor.com/products/kardiaband.

[53] (). *ECG Chest Strap.* Available: https://shop.omegawave.com/shop/omegawave-personal/hardware/ecg-chest-strap.

[54] (). *Sync Heart Rate Soft Strap.* Available: https://www.amazon.com/SYNC-Strap-Captures-Continuous-Accurate-MapMyFitness/dp/B00BNQCS82.

[55] H. P. Bloch, "Ingress Protection code explained," *World Pumps,* vol. 2009, *(11),* pp. 26, 2009. Available: https://www.sciencedirect.com/science/article/pii/S0262176209704183. DOI: 10.1016/S0262-1762(09)70418-3.

[56] S. Mulcahy, "Creating Pressure Sensors with Velostat and Conductive Thread," vol. YouTube, Nov 20, 2014.

[57] Plusea, "Stickytape Bend Sensor in less than 4 minutes," vol. YouTube, Apr 24, 2009.

[58] studioTTTguTTT, "Force Sensing Resistor Tutorial," vol. YouTube, Oct 19, 2015.

[59] (). *Portable Digital Blood Pressure Monitor.* Available: https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2005/ww56_ws62/Final%20Project%20Web/index.html.

[60] (). *Espressif ESP32 Development Board - Developer Edition.* Available: https://www.adafruit.com/product/3269.

[61] (). *34-00069 FSR.* Available: https://www.digikey.com/product-detail/en/interlink-electronics/34-00069/1027-1029-ND/7672214.

[62] (). *30-73258 FSR.* Available: https://www.digikey.com/product-detail/en/interlink-electronics/30-73258/1027-1002-ND/2476470.

Appendix A: FSR Construction

Through initial research of FSRs available on the market, we found that the two largest sizes, 100mm x 10.2mm strip [61] and 43.69mm square [62], were not large enough to use across the chest to detect self-harm. We instead decided to construct our own FSRs using velostat, copper tape and sheets, and tape. The figures below show the designs for six FSRs each labeled and in order from 1-6. FSRs 1-3 are in the first figure and FSRs 4-6 are in the second figure. The primary FSR design was a sandwich configuration, which was made by using three layers of velostat, copper tape on the outside velostat layers, two small cuts of a copper sheet to act as a lead, and tape to keep it all intact. The table below shows the dimensions for each FSR. Although some of the designs were overall smaller than the two FSRs that we could have purchased, these were merely made to test the resistance range and sensitivity based on copper-to-velostat ratio, size, and the type of tape used.

FSRs 1 and 2 were designed using roughly the same dimensions, only differing that FSR 1 was kept together with clear packaging tape while FSR 2 used duct tape. FSR 3 is a larger version of FSR 1.
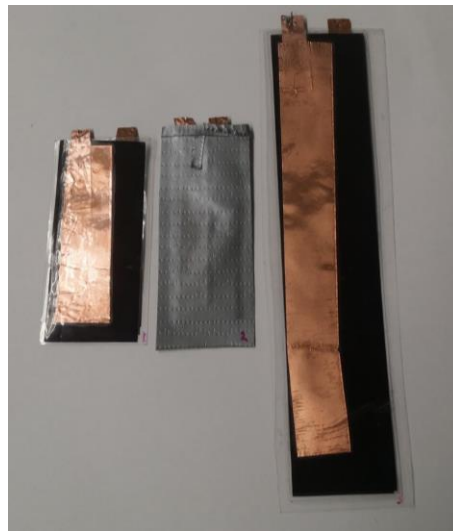
Figure A1. FSR 1-3

FSR 4 was similar to FSR 1, with the only difference being how much of the copper tape covered the surface of the velostat. While FSR 1 had about 7.94mm in width of copper overlapping, FSR 4 had complete overlap of the copper tape. FSR 5 was the largest design, kept together with duct tape and both surfaces with complete copper tape overlap. While FSRs 1-5 used three layers of

velostat, FSR 6 is only comprised of one layer with two pieces of copper tape at the edges. This design was made to test out edge-to-edge contact versus the sandwich configuration.
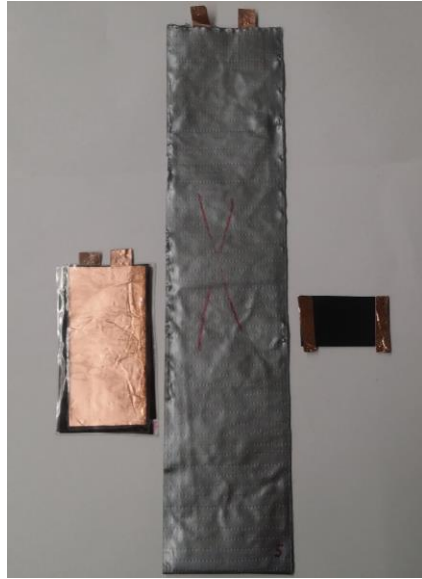
Figure A2. FSR 4-6

Table A1. FSR Dimensions

FSR #	Total Dimensions		Copper Tape Dimensions		Copper Overlap Width [mm]
	Length [mm]	Width [mm]	Length [mm]	Width [mm]	
1	58.74	28.58	51.59	15.88	7.94
2	≈58.74	≈28.58	≈51.59	≈15.88	≈7.94
3	133.35	30.16	120.65	17.46	6.35
4	63.5	38.1	58.74	30.16	full
5	204.79	46.04	201.61	36.51	full
6	19.05	34.93	20.64	4.76	-

Appendix B: FSR Waveforms

No Movement

Slow Hits

FSR Slow Hard Hit

Long Soft Hit

Hard Hits

FSR ADC Fast Hard Hit

Fast Soft Hit
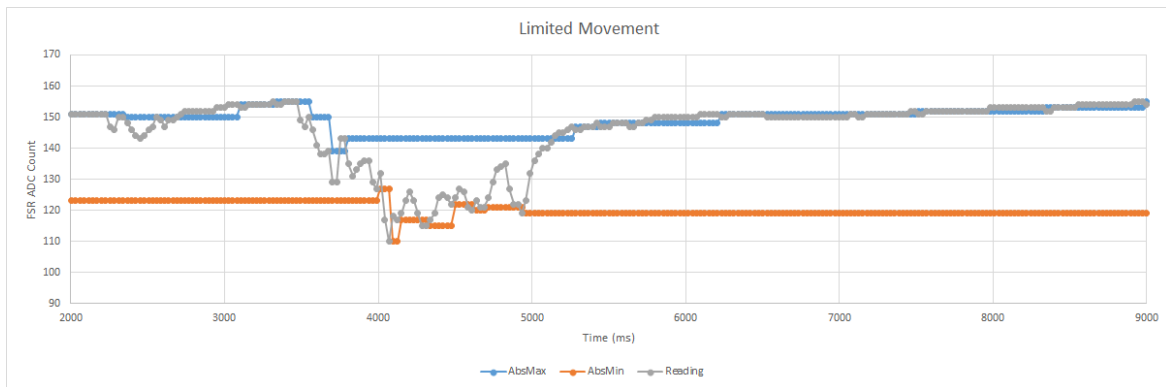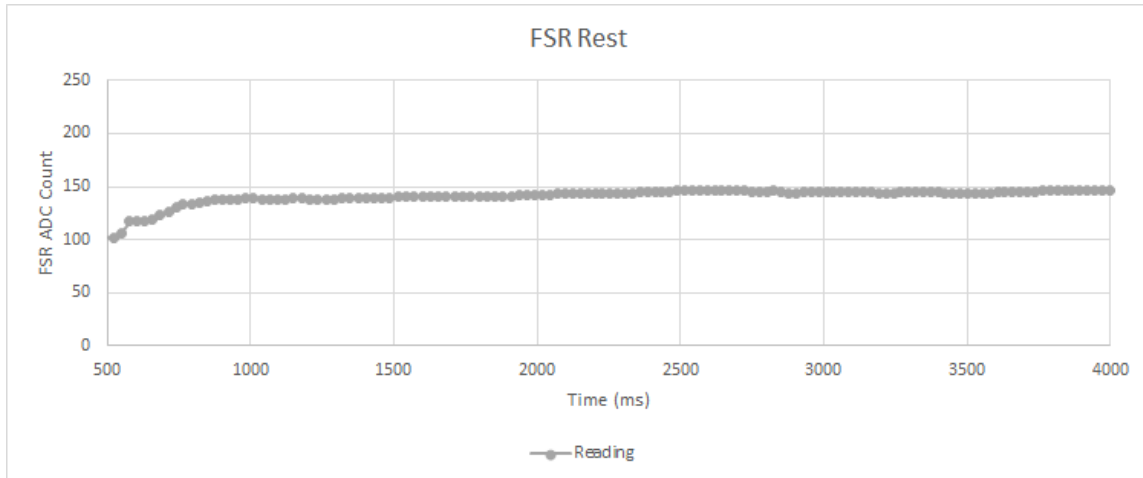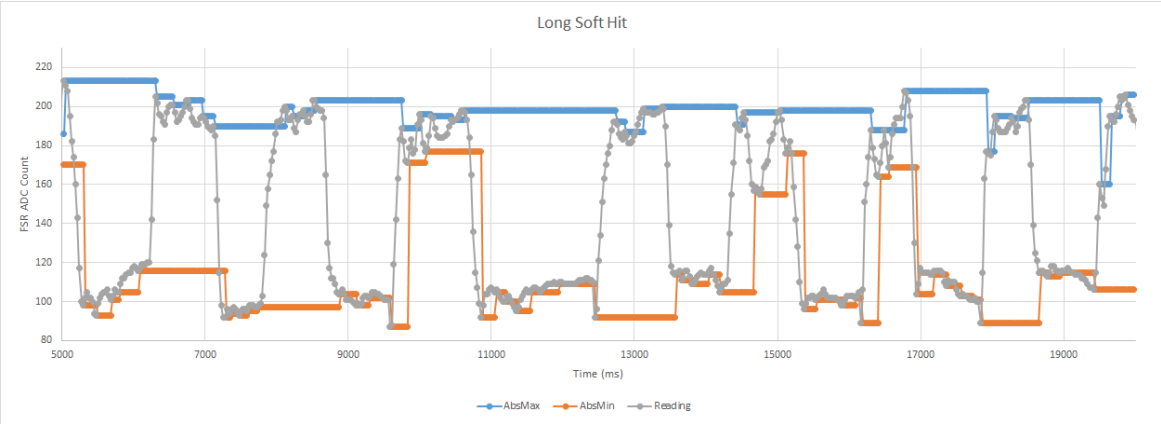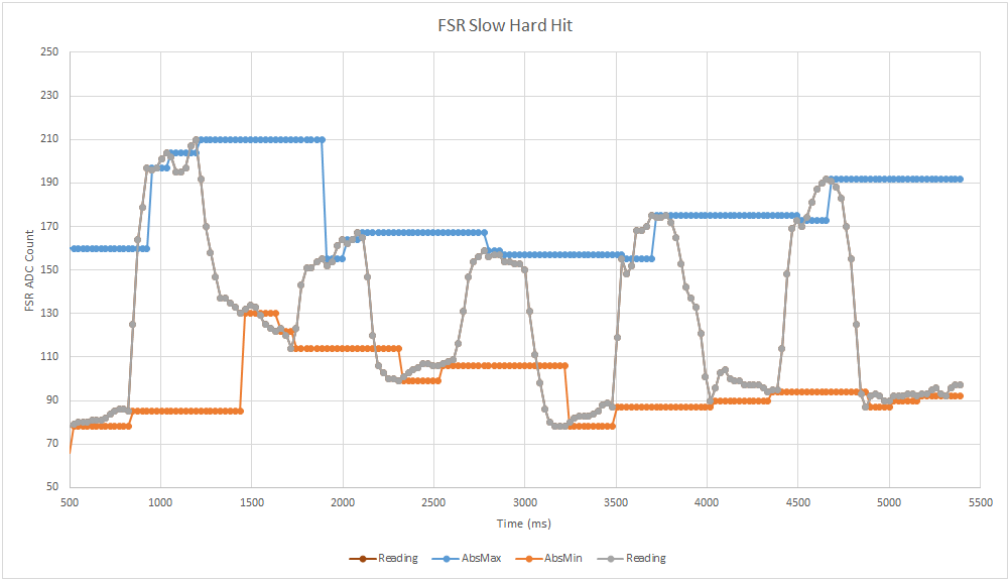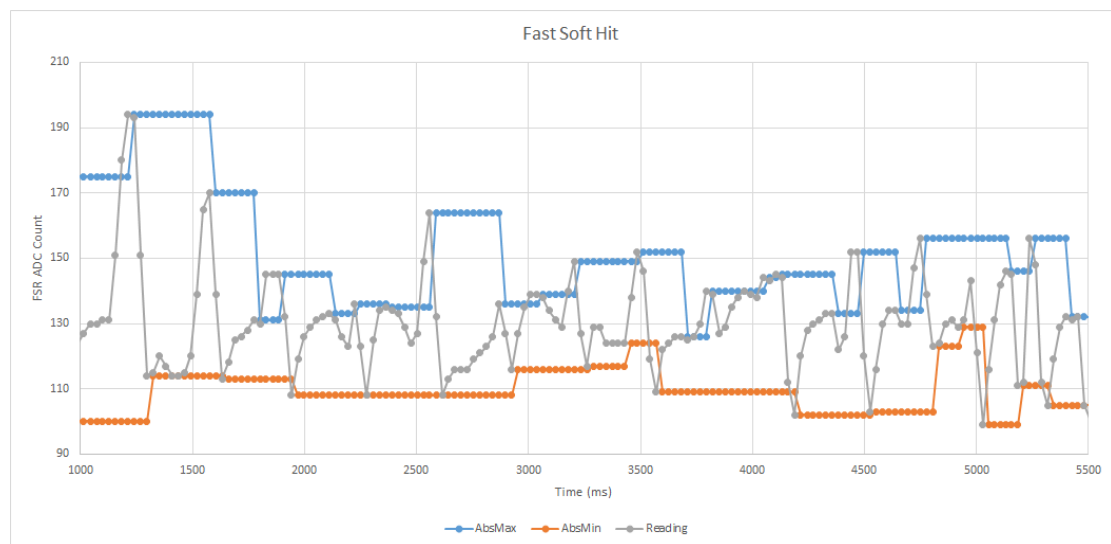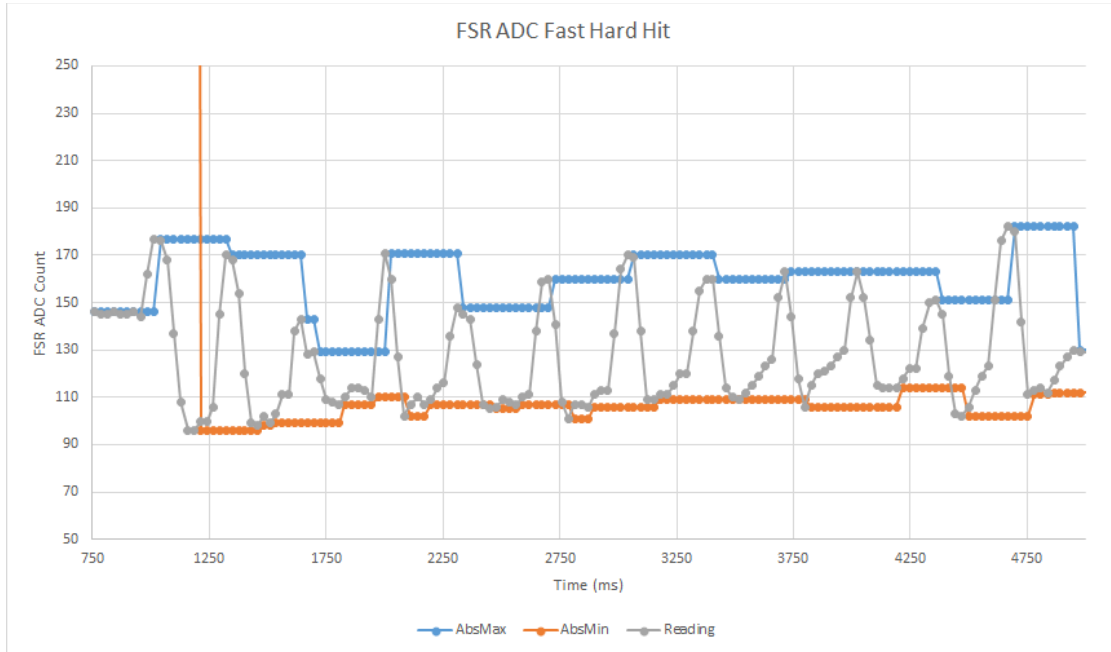
Scratching
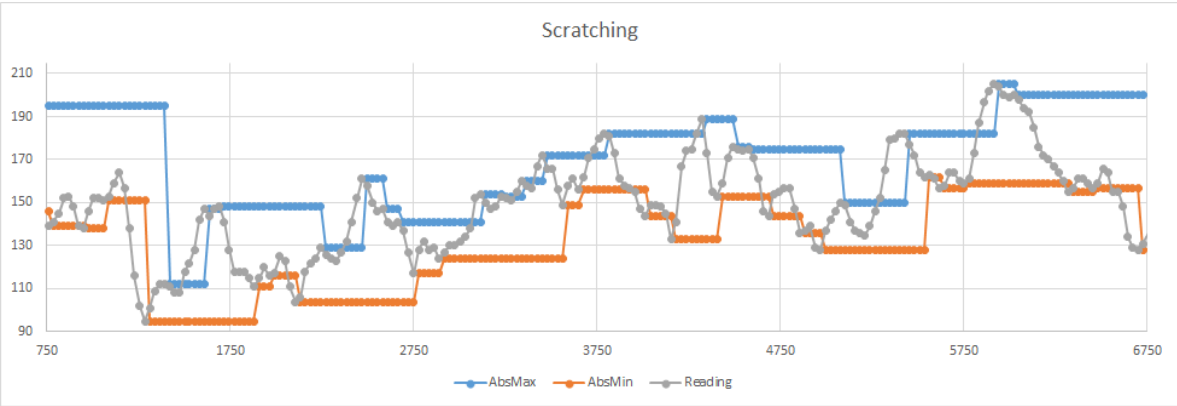
Appendix C: Code Flowchart

Overall Code Diagram

Function	Setup
Description	Set Initial Parameters
Frequency	Once

IMU	Initialize the buffer for the IMU to 0
	Configure the sensor
ECG	Initialize the buffer for bpm to 0
Vibration	Initialize the pins to outputs and set to a digital low
Compression	
FSR	Initialize the pin as an input
Timer	Initialize the timer
BLE	Initialize Bluetooth

Function	Loop
Description	controls the main function of the program executes until termination of power
Frequency	Once every 10ms

Read new sensor data	
IMU	Calls *sensorEvent()*, which saves the orientation of the IMU into the respective global variables
FSR	Calls *readFSR()*, which saves and processes the *FSR values. This function also handles the self harm detection*

Check for new App messages	
Checks if new BLE message available	
Available	Saves the text message to a variable called "printState"
Not Available	

Process message and send data to App	
Checks if print state matches keywords	
KEYWORD	DESCRIPTION
CON	Sets compression output pin to high. Sets the global variable "compressionSytem" high to signal a timer to count to 30 seconds. Then saves the value of printState to default
V1,V2,V3,V4	Checks whether "V1", "V2", "V3", "V4", or "VO" was sent. Turns on the motors by setting the output pin high corresponding to the vibration. Turns them all off if VO is received. Then, the value of printState is set to default
FSR	Continuously prints the value from the ADC
HR	Continuously prints the heart rate value
DEFAULT	Prints "Type a command" once
If the text is anything else it prints that the text is unknown and sets the value of printState to default.	
Check if the variable "fsrAlert" has been set high by another function. If so prints an alert.	
Send messages to App	

Helper Functions
These functions are called by the main loop to execute

Function	Description
printString	Sends a string of characters to the app
configureSensor	Sets the IMU accelerometer ranges to 2G, 2GAUSS, and 500DPS
turnVibrationOff	Turns vibration (1-4) off based on input
turnVibrationOn	Turns vibration (1-4) on based on input
readFSR	Reads the FSR value from the analog pin. Updates the global variable fsrReading.
sensorEvent	Saves the values of the IMU, values to global variables

Timer1 Interrupt

ISR(TIMER1_OVF_vect)		
Frequency: 500ms		
1	Check if the compression system flag is enabled	
	True	Go to 2
	False	Go to 3
Step 2: Checks for a local minimum and assigns the absolute minimum value as the previous local maximum		
2	Check if the compression system has been on for less than 30 seconds by checking if the variable "compressHalfSeconds" is less than 60	
	True	Increment "compressHalfSeconds" by 1
	False	"compressHalfSeconds = 0"
		Disable the compression system flag
		Set the digital output pin for the compression system to 0
The local check is used to check whether or not		
3	Check if the Local Check flag is enabled	
	True	Go to 4
	False	Go to 7
4	Increments variable "check_s" to count ½ seconds	

5	Check if "check_s" > 2 or 1.5 seconds have passed	
	True	If true go to 3
	False	pass
6	Check if the Minimum or Maximum values have stayed the same as the minimum or maximum value when the timer started	
	True	Difference check is disabled
	False	Difference check is enabled
7	Check if the Difference Check flag is enabled	
	True	Go to 8
	False	End
8	Check if less than less than ten seconds have occurred by reading a variable called "fsrHalfSeconds"	
	True	increment the variable value by 1
	False	Do nothing

Timer2 Interrupt

ISR(TIMER2_OVF_vect)	
Frequency: ~1ms	
1	Read the state of the ecg pin and save into variable "current_ecg_state"
2	Increment the variable "ecg_count_ms" to indicate how many interrupts have occurred

Step 2: Checks for a local minimum and assigns the absolute minimum value as the previous local maximum

3	Check if the current ecg state is HIGH and the current ecg state is not equal to the previous ecg state	
	True	Go to 4
	False	Go to 5
4	Check if the absolute value of the local minimum – FSR value is greater than 15	
	True	Local minimum = FSR value
	False	Go to 4

Step 4: Checks for a local maximum and assigns the absolute minimum value as the previous local maximum

4	Save the ecg_count_ms to beatRecordedMs
	Reset the ecg_count_ms to 0
	Calculate the pulseRateBeatsPerMinute by using the equation "60000/beatRecordedMs*(1.023)"

	Add the the new pulseRateBeatsPerMinutes to the beatRecordedatMs array and discard the oldest entry
	Take the average of the values in the array
	Round the average value to the nearest whole number and save the value to "averagePulse"
5	Set the last_ecg_state equal to the current_ecg_state
Step 6: This checks if the variable overflows	
6	Check if ecg_count_ms is equal to 65535

	True	Go to 7
	False	End

7	Set the ecg_count_ms equal to 0

FSR Algorithm

readFSR()		
1	Read the analog input pin to get the FSR value in ADC counts save into variable called average	
Step 2: Checks for a local minimum and assigns the absolute minimum value as the previous local maximum		
2	Check if the FSR value (average) is less than the previous FSR value	
	True	Absolute Maximum = Previous local maximum
		Go to 3
	False	Go to 4

3	Check if the absolute value of the local minimum – FSR value is greater than 15	
	True	Local minimum = FSR value
	False	Go to 4
Step 4: Checks for a local maximum and assigns the absolute minimum value as the previous local maximum		
4	Check if the FSR value (average) greater than the previous FSR value	
	True	Local maximum = FSR value
	False	Go to 6
5	Check if the absolute value of the local minimum – FSR value is greater than 15	
	True	Local maximum = FSR value
	False	Go to 6
Step 6 checks whether the absolute minimum or absolute maximum stays the same. Through testing, we have found that the at rest, one of the absolute maximums will remain constant, while the other will follow the value of the current FSR adc count value		
6	Check if either the previous absolute minimum is equal to the current absolute minimum or if the previous absolute maximum is equal to the current absolute maximum	
	True	Enable the Local Check flag
		Go to 7
	False	Enable the Diff Check flag
		Disable the Local Check flag
		Set the previous absolute maximum equal to the current absolute maximum

		Set the previous absolute minimum equal to the current absolute minimum
7		Check if the flag once is true (0) or false (1). This checks if the Local check counter is already counting
	True	Set the minimum and maximum check values equal to the local maximum and minimum value
	False	Go to 8
8		Check if the difference check flag been enabled
	True	The flag to count time in the ISR called "fsrCheck" is enabled
		The absolute value of the difference of the absolute minimum and maximum is saved in "MaxMinDiff".
		The variable diff_count is incremented by 1
		A variable called "runninfdiff" saves the running total of the "MaxMinDiff" variable
		A variable called "running_average" saves the (runningdiff/diff_count) to get the average difference of the absolute minimum and the absolute maximum since the difference check flag has been enabled
	False	Go to 9
9		The previous FSR value is set to the current FSR value
10		Go back to 1

Appendix D: ECG Tests

1.026						
Period	Frequency	Heart Rate	Measured HR	Measured - Expected	Raw ECG count MS	OFFSET
2	0.5	30	29.91	0.09	1955.183289	1.022922
1.5	0.666666667	40	39.88	0.12	1466.387466	1.022922
1.2	0.833333333	50	49.85	0.15	1173.109973	1.022922
1	1	60	59.83	0.17	977.4282494	1.023093
0.8	1.25	75	74.78	0.22	782.0210239	1.0229904
0.75	1.333333333	80	79.77	0.23	733.1018198	1.02305025
0.66	1.515151515	90.90909091	90.65	0.259090909	645.1134271	1.0230759
0.6	1.666666667	100	99.71	0.29	586.4961605	1.0230246
0.54	1.851851852	111.1111111	110.8	0.311111111	527.7936116	1.0231272
0.5	2	120	119.66	0.34	488.7141247	1.023093
0.46	2.173913043	130.4347826	130.07	0.364782609	449.6004625	1.02313062
0.42	2.380952381	142.8571429	142.46	0.397142857	410.4979093	1.02314772
0.4	2.5	150	149.6	0.4	390.9059637	1.023264
0.36	2.777777778	166.6666667	166.23	0.436666667	351.7989061	1.02331188
0.35	2.857142857	171.4285714	170.99	0.438571429	342.0055685	1.02337515
0.33	3.03030303	181.8181818	181.33	0.488181818	322.5033484	1.02324519
0.31	3.225806452	193.5483871	193.07	0.478387097	302.8928998	1.02346407
0.3	3.333333333	200	199.52	0.48	293.1011035	1.0235376
1.023						
2	0.5	30	29.99	0.01	1955.686108	1.022659
1.5	0.666666667	40	39.98	0.02	1467.009164	1.0224885
1.2	0.833333333	50	49.96	0.04	1173.959696	1.0221816
1	1	60	59.97	0.03	978.0061096	1.0224885
0.8	1.25	75	75	0	782.0136852	1.023
0.75	1.333333333	80	80.02	-0.02	732.9545913	1.02325575
0.66	1.515151515	90.90909091	90.93	-0.020909091	645.0129373	1.02323529
0.6	1.666666667	100	100.09	-0.09	585.9828793	1.0239207
0.54	1.851851852	111.1111111	111.08	0.031111111	528.0070795	1.02271356
0.5	2	120	119.94	0.06	489.0030548	1.0224885
0.46	2.173913043	130.4347826	130.63	-0.195217391	448.9858868	1.02453109
0.42	2.380952381	142.8571429	143.05	-0.192857143	410.0036798	1.02438105
0.4	2.5	150	150	0	391.0068426	1.023
0.36	2.777777778	166.6666667	166.62	0.046666667	352.0047197	1.02271356
0.35	2.857142857	171.4285714	171.49	-0.061428571	342.0084343	1.023366575
0.33	3.03030303	181.8181818	181.58	0.238181818	323.0037801	1.02165987
0.31	3.225806452	193.5483871	193.57	-0.021612903	302.9964684	1.023114235
0.3	3.333333333	200	200.17	-0.17	293.0060768	1.02386955

Appendix E: IMU Test
Leaning Left

Leaning Right

Leaning Right Accelerometer

Leaning Right Gyroscope

Leaning Right Magnometer

Sitting Still

Sitting Still Accelerometer

Sitting Still Gyroscope

Sitting Still Magnometer
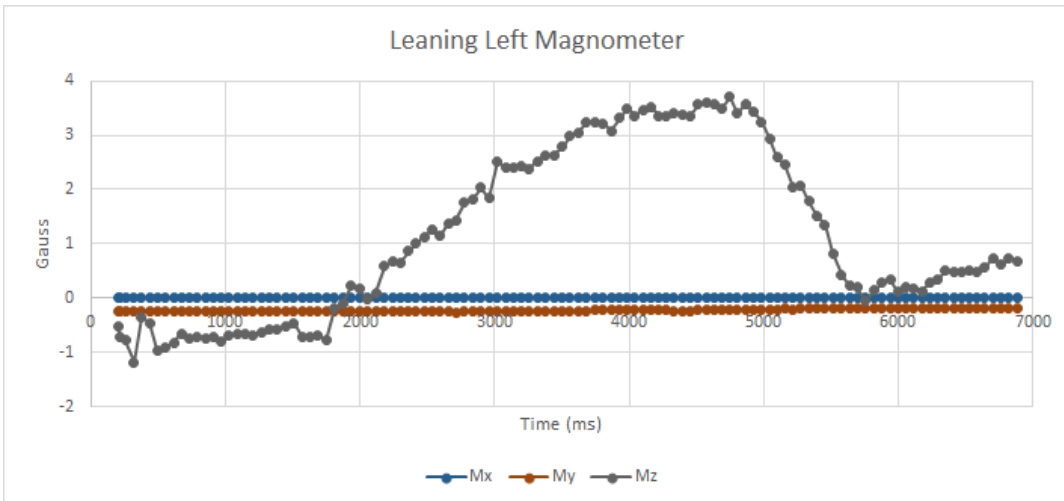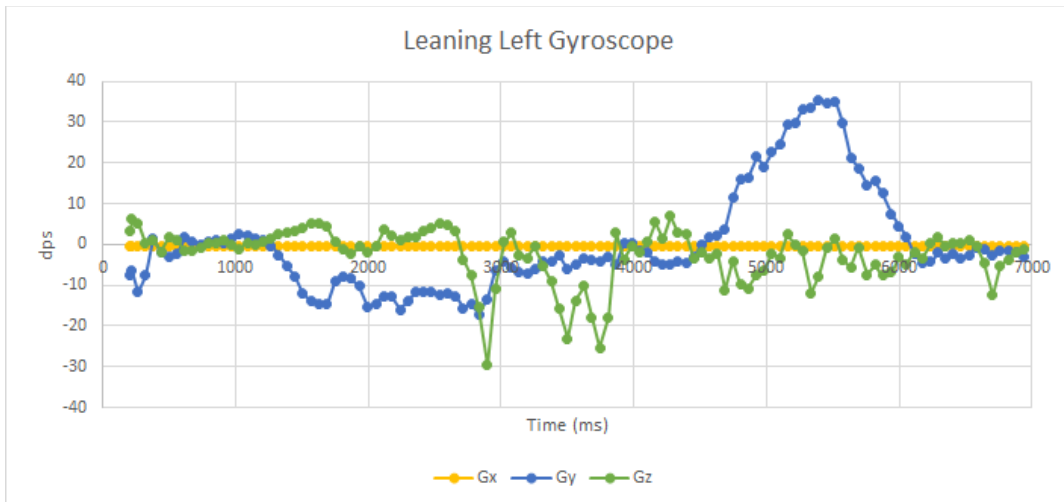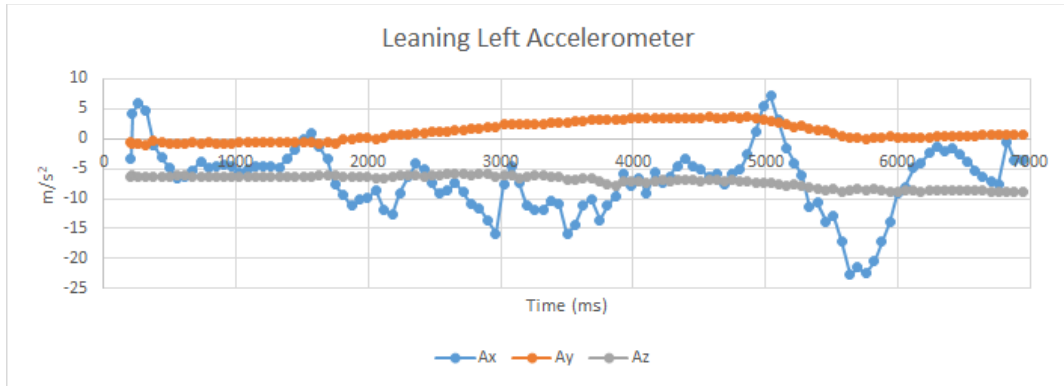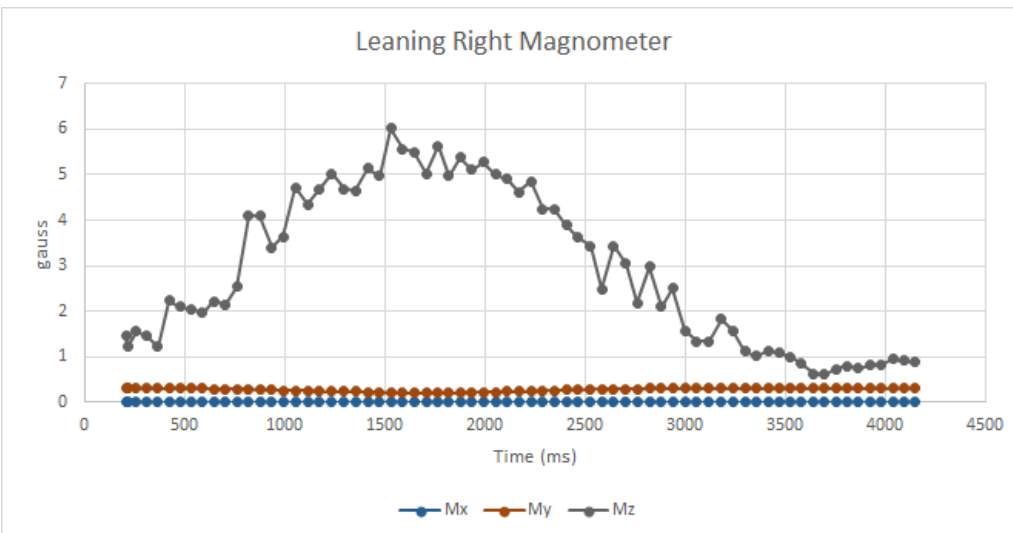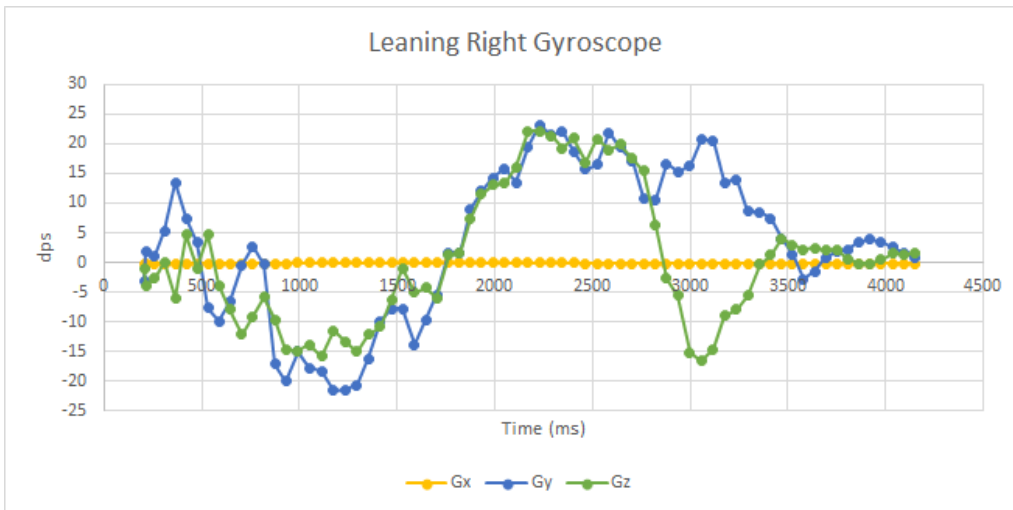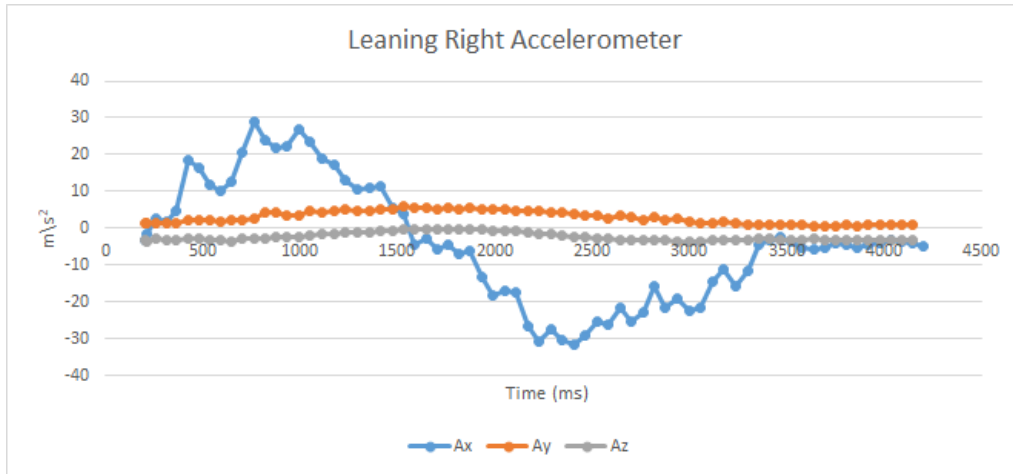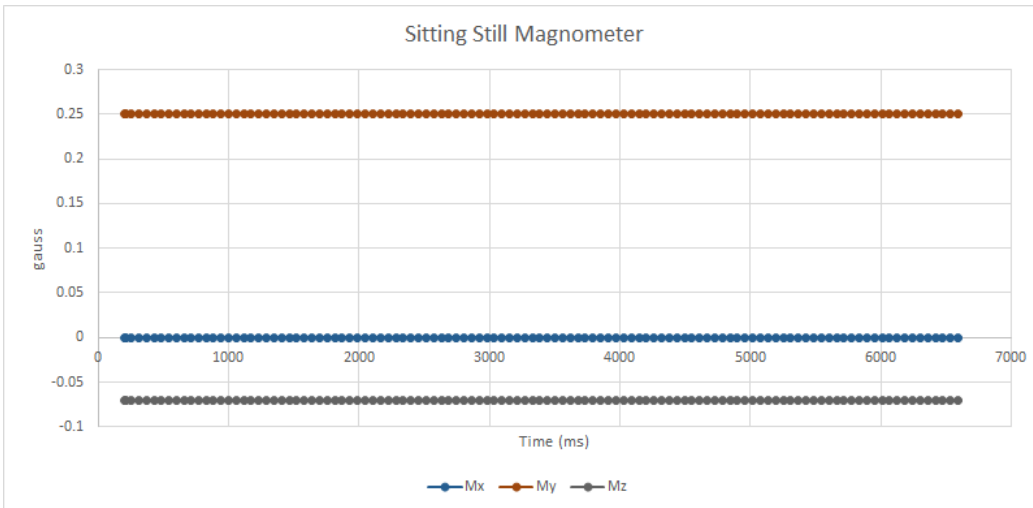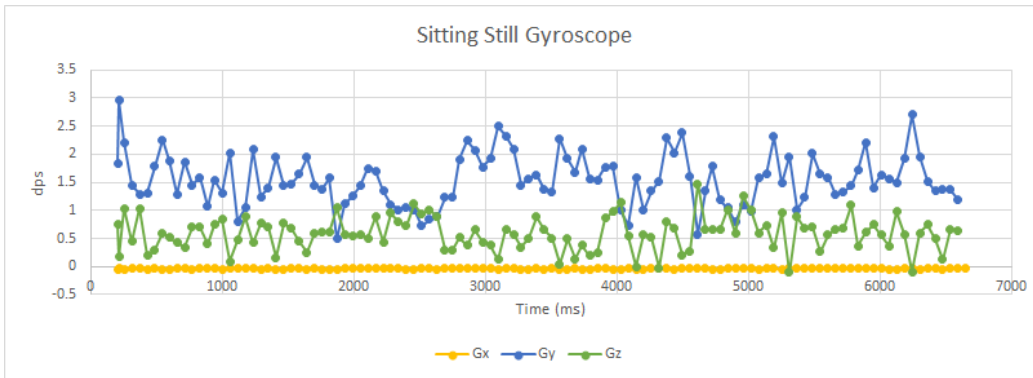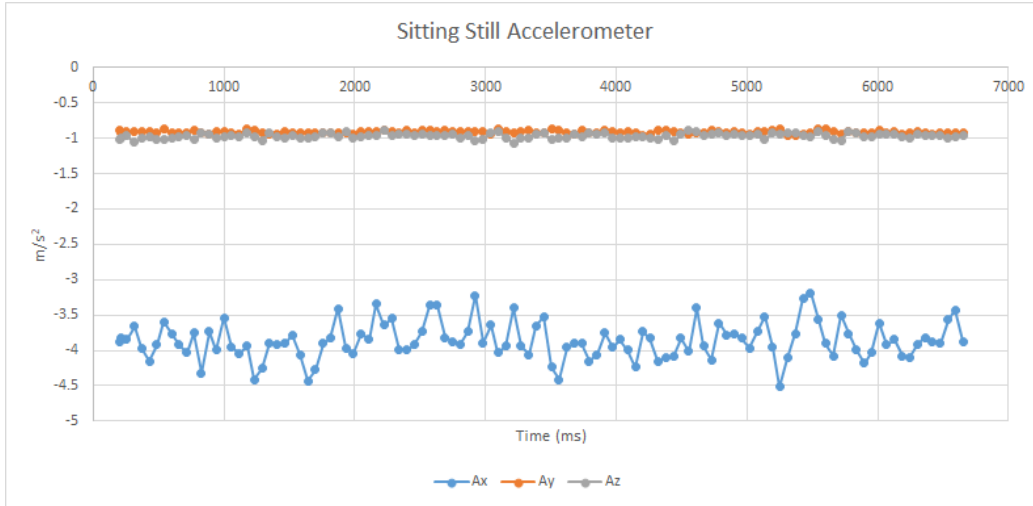
Walking
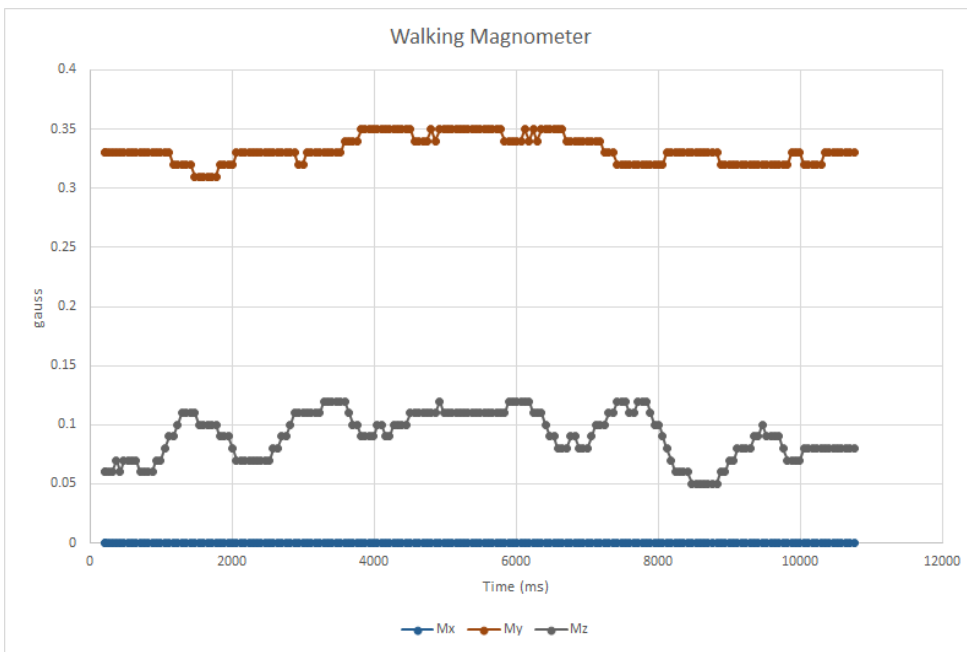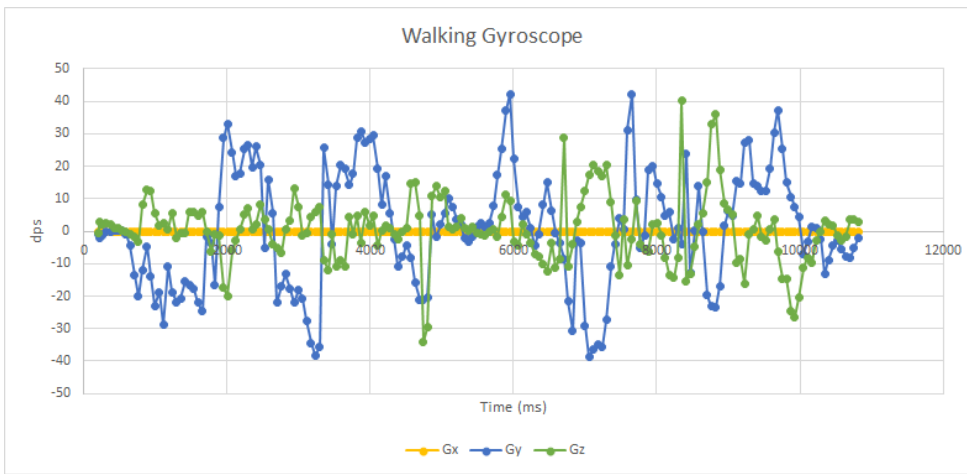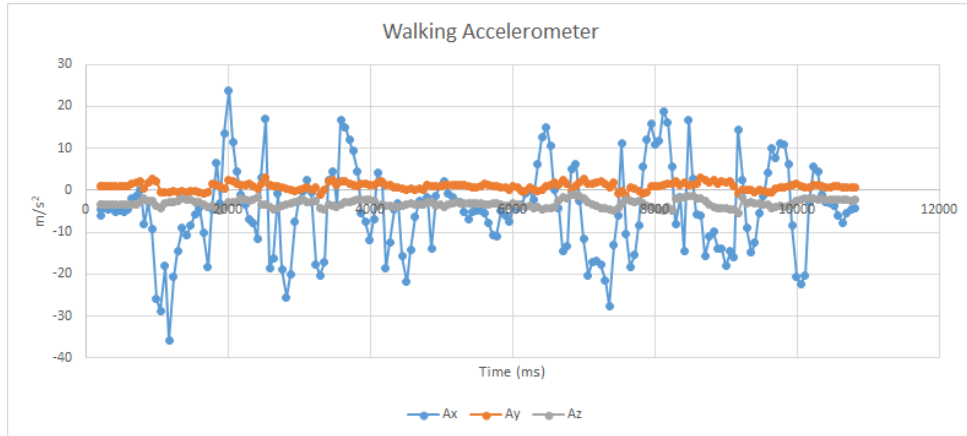
Appendix F: CC2650 Voltage Divider FSR Readings

Using $R_{REF} = 20$ kΩ

FSR #1 (sandwich config)			
ADC Reading (Pressed)	Measured V_{REF}	ADC Reading (Not Pressed)	Measured V_{REF}
2791	2.93	1766	1.854
2796	2.936	1765	1.853
2795	2.935	1766	1854
2806	2.947	1763	1.851
2811	2.952	1767	1.856
2812	2.953	1769	1.858
2814	2.955	0	0
2818	2.959	0	0
0	0	0	0

FSR #2 (sandwich config)			
ADC Reading (Pressed)	Measured V_{REF}	ADC Reading (Not Pressed)	Measured V_{REF}
2709	2.845	896	0.941
2269	2.908	900	0.945
2067	2.171	904	0.949
2225	2.336	915	0.961
2165	2.273	912	0.958
2230	2.342	897	0.0.942
2222	2.333	903	0.948
2	0.0021	889	0.934
23	0.02415	874	0.918

FSR #3 (sandwich config)			
ADC Reading (Pressed)	**Measured V_{REF} [v]**	**ADC Reading (Not Pressed)**	**Measured V_{REF} [v]**
2810	2.951	1857	1.94996
2808	2.949	1859	1.952
2816	2.957	1856	1.9489
2825	2.966	1857	1.94996
2830	2.972	1864	1.957
2833	2.975	1864	1.957
2840	2.982	1858	1.951
2826	2.968	1862	1.955
0	0	0	0

FSR #4 (sandwich config)			
ADC Reading (Pressed)	**Measured V_{REF} [mV]**	**ADC Reading (Not Pressed)**	**Measured V_{REF} [mV]**
7	7.35	14	14.7
16	16.8	5	5.25
10	10.5	12	12.6
6	6.3	4	4.2
14	14.7	7	7.35
0	0	7	7.35
0	0	9	9.45
0	0	19	19.95
0	0	0	0

FSR #5 (sandwich config)			
ADC Reading (Pressed)	Measured V_{REF} [mV]	ADC Reading (Not Pressed)	Measured V_{REF} [mV]
14	14.7	3	3.15
5	5.25	9	9.45
10	10.5	0	0
24	25.2	12	12.6
10	10.5	11	11.55
7	7.35	12	12.6
0	0	14	14.7
0	0	11	11.55
0	0	0	0

FSR #6 (edge-to-edge config)			
ADC Reading (Pressed)	Measured V_{REF} [mV]	ADC Reading (Not Pressed)	Measured V_{REF} [mV]
885	929.3	264	277.23
858	900.95	297	311.87
864	907.25	250	262.52
809	849.5	232	243.61
824	865.25	262	275.12
841	883.1	250	262.52
822	863.15	175	183.76
829	870.5	229	240.46
834	875.75	237	248.87

The following voltage readings were measured using a multimeter:

FSR #	V$_{\text{PRESSED}}$ [v]	V$_{\text{NOT PRESSED}}$ [v]
1	3.018	1.736
2	1.6 - 1.9	0.5 - 1.148
3	3.038	1.995 - 2.027
4	0.1m	0.1m
5	0.1m	0.1m
6	330m	0.5

Appendix G: FSR Equation Derivation

$$\frac{d\Delta V}{dR_{ref}} = V_{DD}\left(\frac{R_{min}}{(R_{ref} + R_{min})^2} - \frac{R_{max}}{(R_{ref} + R_{max})^2}\right)$$

$$0 = V_{DD}\left(\frac{R_{min}}{(R_{ref} + R_{min})^2} - \frac{R_{max}}{(R_{ref} + R_{max})^2}\right)$$

$$0 = \left(\frac{R_{min}}{(R_{ref} + R_{min})^2} - \frac{R_{max}}{(R_{ref} + R_{max})^2}\right)$$

$$0 = \frac{(R_{min} - R_{max})(R_{min}R_{max} - R_{ref}^2)}{(R_{ref} + R_{min})^2(R_{ref} + R_{max})^2}$$

$$R_{ref}^2 = R_{min}R_{max}$$

Appendix H: Decision Matrix

	Broad Applicability	Cost	Time	Learning Curve/References	Winnability	Wow	Innovativeness	Amount of resources	Liability	Testability	Benefit to Society	ECE Content	Personal Marketability	Final Weight
							Project Decision							
Weight	2.00	2.00	3.00	3.00	3.00	1.00	2.00	3.00	2.00	3.00	2.00	3.00	3.00	
Wearable HR monitor	2	3	3	1	2	1	1	3	3	3	2	3		68
Smart Vest	3	2	2	2	2	2	1	3	3	3	1	3		67
Smart Home Module	1	2	2	2	1	1	1	3	3	2	1	3		56
Piezoelectric Generator	1	3	2	3	2	3	1	3	3	3	1	3		69
Window Implosion Sensor	1	2	3	1	2	1	2	1	3	1	2	3		54

1. In the top row write what is important to you, for example if you are buying a computer it will be portability, speed, memory etc.

2. In the next row (weight) write a number between 1 and 2 depending on the priority factor you want to give (2 being highest)

3 Now write your choices in the first column (for example Dell, HP, DIY etc.)

4. In each cell write a number between 1 to 9 depending on how much each choice agrees with the priority in the same column

5. The final weight will be calculated as a sum of all your weighted priorities

*Cost rankings where 0 is an unobtainable amount of money, and 3 is under our budget

Resource rankings 0 the technology isn't available yet, 3 there are numerous versions of the technology we are looking to utilize

Liability: high danger to participants 0, low danger to participants 3

Time: 0 an extraordinary amount of time over x hours per week, 3 under 20hrs per week

Amount of resources: access to things

Appendix I: Code

```
/*
 *   Chat
 *
 *   Simple chat sketch, work with the Chat
iOS/Android App.
 *   Type something from the Arduino serial monitor to
send
 *    to the Chat App or vice verse.
 *
 */


//"RBL_nRF8001.h/spi.h/boards.h" is needed in every
new project
//#define SERIALPRINT
#define IMU
#define BLE

#include <SPI.h>
#include <EEPROM.h>
#include <boards.h>

#ifdef BLE
#include <RBL_nRF8001.h>
#endif //ble

#ifdef IMU
#include <Adafruit_Sensor.h>
#include <Adafruit_LSM9DS0.h>
#endif // imu
M


/***********************************************
***********************/
/*
   Defines and Variables for applications
*/
/***********************************************
***********************/
/*************Testing Changing
Variables********************/
uint8_t checkInt = A1;
uint8_t ledPin = A0;

/*************General Changing
Variables********************/
char printState[20];
uint8_t repeat = 0;
uint8_t printtype = 0;
uint8_t read_index = 0;
int timer1_counter;

uint8_t check = 1;
/*************IMU Defines********************/
/* Assign a unique base ID for this sensor */
#ifdef IMU
Adafruit_LSM9DS0 lsm = Adafruit_LSM9DS0(1000);
// Use I2C, ID #1000
char *Op[] ={"X: ","Y: ","Z: "};
/*************IMU Changing
variables********************/
volatile float print_a[3];
volatile float print_m[3];
volatile float print_g[3];
volatile float print_temp = 0;


#endif // imu

/************ECG Defines******************/
const uint8_t ecgPin = 2; //the number of the ECG pin
input
/************ECG Changing
Variables***************/
volatile uint8_t ecgStatus = 0; //variable for reading the
ecg ]
volatile uint8_t heartRate = 0;
volatile uint8_t lastHR = 0;
float pulseRateBeatsPerMinute = 0;
float beatRecordedatMs[10];
float sum;
int sum_i;
float averagePulse = 0;
int beatRecordedMs = 0;
uint8_t current_ecg_state = 0;
uint8_t last_ecg_state = 0;
unsigned int ecg_count_ms = 0;


/************Vibration Defines***************/

const uint8_t vibrationPin1 = 6; //the number of
vibration pin 1 output
const uint8_t vibrationPin2 = 5; //the number of
vibration pin 2 output
const uint8_t vibrationPin3 = 4; //the number of
vibration pin 3 output
const uint8_t vibrationPin4 = 3; //the number of
vibration pin 4 output
#define OFF 0
#define ON 1

/**********Vibration Changing Variables***********/


/************Compression Defines*************/
uint8_t compressPin = 7; // the number of
compression output
/************Compression Changing
Variables**************/
uint8_t compressionSystem = OFF;
uint8_t compressHalfSeconds = 0;

/******************FSR
Defines************************/
uint8_t fsrPin = A3; //the number of the fsrInput

/************FSR changing variables*******/
int fsrReading = 0; //variable for reading FSR
uint8_t fsrAlert = 0;
uint8_t fsrHalfSeconds = 0;
uint8_t fsrCheck = OFF;
int fsrMax = 0;
int fsrMin = 1000;
int fsrTopMax = 0;
int fsrTopMin = 0;
int prevReading = 0;
int prevFsrMax = 0;
int prevFsrMin = 0;
uint8_t LocalCheck = 0;
uint8_t LocalSame = 0;
int check_ms = 0;
uint8_t diffCheck = 0;

int fsrMaxCheck;
int fsrMinCheck;
```

```
long diff_count = 0;
long runningdiff = 0;
float running_average;
long MaxMindiff = 0;

uint8_t once = 1;
const int numReadings = 2;

int readings[numReadings];     // the readings from the
analog input
int readIndex = 0;             // the index of the current
reading
int total = 0;             // the running total
int average = 0;
/**************************************************
*************************/
/*!
   Configure the Arduino and start advertising with the
radio
*/
/**************************************************
*************************/
void setup()
{

  pinMode(ledPin, OUTPUT);
  pinMode(checkInt, OUTPUT);
  digitalWrite(ledPin, LOW);
  digitalWrite(checkInt, LOW);

  /**********IMU Initializaion**************/
#ifdef IMU
  if(!lsm.begin())
  {
    /* There was a problem detecting the LSM9DS0 ...
check your connections */
    while(1){
      digitalWrite(ledPin, digitalRead(ledPin)^1);
    }
  }
  for (int i=0; i<3;i++){
    print_a[i] = 0;
    print_g[i] = 0;
    print_m[i] = 0;
  }

  /* Setup the sensor gain and integration time */
  configureSensor();
#endif
  /**********ECG Initialization*************/
  pinMode(ecgPin, INPUT); // ecg initialized as input

  for (int i=0; i<10;i++){
    beatRecordedatMs[i] = 0;
  }

  /**********Vibration Initialization*********/
  pinMode(vibrationPin1, OUTPUT); // vibration pin1
initialized as output
  digitalWrite(vibrationPin1, LOW);
  pinMode(vibrationPin2, OUTPUT); // vibration pin2
initialized as output
  digitalWrite(vibrationPin2, LOW);
```

```
  pinMode(vibrationPin3, OUTPUT); // vibration pin3
initialized as output
  digitalWrite(vibrationPin3, LOW);
  pinMode(vibrationPin4, OUTPUT); // vibration pin4
initialized as output
  digitalWrite(vibrationPin4, LOW);


  /******Compression Initializations*******/
  pinMode(compressPin, OUTPUT); // compression
pin initialized as output
  digitalWrite(compressPin, LOW);

  /******FSR Initializations*******/
  pinMode(fsrPin, INPUT);

  /******Timer Initializations*******/
  noInterrupts();         // disable all interrupts
  TCCR1A = 0;
  TCCR1B = 0;

  // TIMER 1 for interrupt frequency 2Hz:
  timer1_counter = 34286;  // preload timer 65536-
16MHz/256/2Hz

  TCNT1 = timer1_counter;  // preload timer
  TCCR1B |= (1 << CS12);   // 256 prescaler
  TIMSK1 |= (1 << TOIE1);  // enable timer
overflow interrupt

  // TIMER 2 for interrupt frequency 1000 Hz:
  TCCR2B = 0x00;        //Disbale Timer2 while we set
it up
  TCNT2  = 130;        //Reset Timer Count to 130 out
of 255
  TIFR2  = 0x00;       //Timer2 INT Flag Reg: Clear
Timer Overflow Flag
  TIMSK2 = 0x01;       //Timer2 INT Reg: Timer2
Overflow Interrupt Enable
  TCCR2A = 0x00;       //Timer2 Control Reg A: Wave
Gen Mode normal
  TCCR2B = 0x05;       //Timer2 Control Reg B: Timer
Prescaler set to 128

  interrupts();         // enable all interrupts

  /**********BLE Initializaion**************/
#ifdef BLE
  strcpy(printState, "DEFAULT");
  // Default pins set to 9 and 8 for REQN and RDYN
  // Set your REQN and RDYN here before
ble_begin() if you need
  //ble_set_pins(3, 2);

  // Set your BLE Shield name here, max. length 10
  ble_set_name("SmartShirt");

  // Init. and start BLE library.
  ble_begin();
#endif
  // Enable serial debug
#ifdef SERIALPRINT
  Serial.begin(9600);
#endif
```

```
}
/**************************************************
*************************/
/*!
   Constantly checks for new events on the nRF8001
*/
/**************************************************
*************************/
void loop()
{
/**************Print data for
debugging********************/

#ifdef SERIALPRINT
#endif

/**************Read IMU data********************/
#ifdef IMU
  sensorEvent();
#endif // IMU
  readFSR();
/**************Check for incoming data from
APP********************/
#ifdef BLE
  /*Check if BLE Available*/
  checkInput();
  if (check == ON){
  /**************Choose variables to be
Printed********************/
    parseInput();
  /**************Send data to
app********************/
    if (fsrAlert == ON){
      String fsrA = "\nSelf Harm Detected";
      printString(fsrA);
      fsrAlert = OFF;
      strcpy(printState, "DEFAULT");
    }
    check = OFF;
    ble_do_events();
  }


#endif // BLE
}

/**************************************************
****************************************************
                Helper Functions
****************************************************
****************************************************
/
/**************************************************
*************************/
/*
   Print the string to text
*/
/**************************************************
*************************/
#ifdef BLE
void printString(String s){
  uint8_t SendBuffer[20];
  s.getBytes(SendBuffer, 20);
  char SendBufferSize = min(20, s.length());
```

```
ble_write_bytes(SendBuffer, SendBufferSize);
}
void checkInput(){
  if (ble_available()){
    char readBuff[20]; //define read buffer
    char c; //define read chacater variable
    read_index = 0; //define index variable
    /*Read input from app*/
    while (ble_available()){
      c = ble_read();
      readBuff[read_index] = c;
      read_index = read_index + 1;
      repeat = 0;
      printtype = 0;
      strcpy(printState, readBuff);
    }
  }
}
void parseInput(){
  if (strcmp(printState, "IMU") == 0){ /*Check for
IMU input*/
  }
  else if (strcmp(printState, "V1") == 0){/*Check for
Vibration input*/
    toggleVibration(1);
    strcpy(printState, "DEFAULT");
  }
  else if (strcmp(printState, "V2") == 0){
    toggleVibration(2);
    strcpy(printState, "DEFAULT");
  }
  else if (strcmp(printState, "V3") == 0){
    toggleVibration(3);
    strcpy(printState, "DEFAULT");
  }
  else if (strcmp(printState, "V4") == 0){
    toggleVibration(4);
    strcpy(printState, "DEFAULT");
  }
  else if (strcmp(printState, "VO") == 0){
    turnVibrationOff(5);
    strcpy(printState, "DEFAULT");
  }
  else if (strcmp(printState, "CON") == 0){
    digitalWrite(compressPin, HIGH);
    compressionSystem = ON;
    strcpy(printState, "DEFAULT");
  }
  else if (strcmp(printState, "FSR") == 0){
    String fsr = String(fsrReading);
    printString(fsr);
  }
  else if (strcmp(printState, "DEFAULT") ==
0){ /*Default Input*/
    if (repeat == 0){
      String def = "Type a Command \n";
      printString(def);
      repeat = 1;
    }
  }
  else if (strcmp(printState, "HR") == 0){
    if (lastHR != pulseRateBeatsPerMinute){
      String hr = "ECG: " + String(averagePulse) + "
bpm\n";
```

```
      printString(hr);
    }
    lastHR = pulseRateBeatsPerMinute;
  }
  else{ /*Unknown Input*/
    String def = "Unknown Command ";
    printString(def);
    printString(printState);
    strcpy(printState, "DEFAULT");
  }
}
#endif //ble
/*****************************************************
*****************************************************
              IMU Functions
*****************************************************
*****************************************************
/
/*****************************************************
************************/
/*
  Reads the data from the IMU
*/
/*****************************************************
************************/
#ifdef IMU
void sensorEvent(){
  /* Get a new sensor event */
  sensors_event_t accel, mag, gyro, temp;
  lsm.getEvent(&accel, &mag, &gyro, &temp);

  print_a[0] = accel.acceleration.x;
  print_a[1] = accel.acceleration.y;
  print_a[2] = accel.acceleration.z;

  print_m[1] = mag.magnetic.x;
  print_m[2] = mag.magnetic.y;
  print_m[3]= mag.magnetic.z;

  print_g[1] = gyro.gyro.x;
  print_g[2] = gyro.gyro.y;
  print_g[3] = gyro.gyro.z;

  print_temp = temp.temperature;

}

/*****************************************************
************************/
/*
  Configures the gain and integration time for the
TSL2561
*/
/*****************************************************
************************/
void configureSensor(void)
{
  /*Initialize the IMU*/
  // 1.) Set the accelerometer range
  lsm.setupAccel(lsm.LSM9DS0_ACCELRANGE_2G);
  // 2.) Set the magnetometer sensitivity
  lsm.setupMag(lsm.LSM9DS0_MAGGAIN_2GAUSS);
  // 3.) Setup the gyroscope
```

```
  lsm.setupGyro(lsm.LSM9DS0_GYROSCALE_500DPS)
;
}
#endif // IMU
/*****************************************************
*****************************************************
              Vibration Functions
/*****************************************************
*****************************************************
/
/*****************************************************
************************/
/*
  Toggle Vibration motor
*/
/*****************************************************
************************/
void toggleVibration(uint8_t pins){
  if (pins == 1){
    digitalWrite(vibrationPin1,
digitalRead(vibrationPin1)^1);
  }
  else if (pins == 2){
    digitalWrite(vibrationPin2,
digitalRead(vibrationPin2)^1);
  }
  else if (pins == 3){
    digitalWrite(vibrationPin3,
digitalRead(vibrationPin3)^1);
  }
  else if (pins == 4){
    digitalWrite(vibrationPin4,
digitalRead(vibrationPin4)^1);
  }
}
/*****************************************************
************************/
/*
  Turn Vibration Motors OFF
*/
/*****************************************************
************************/
void turnVibrationOff(uint8_t pins){
  if (pins == 1){
    digitalWrite(vibrationPin1, LOW);
  }
  else if (pins == 2){
    digitalWrite(vibrationPin2, LOW);
  }
  else if (pins == 3){
    digitalWrite(vibrationPin3, LOW);
  }
  else if (pins == 4){
    digitalWrite(vibrationPin4, LOW);
  }
  else if (pins ==5){
    digitalWrite(vibrationPin1, LOW);
    digitalWrite(vibrationPin2, LOW);
    digitalWrite(vibrationPin3, LOW);
    digitalWrite(vibrationPin4, LOW);
  }
}
```

```
/*************************************************
*************************/
/*
   Turn Vibration Motors on
*/
/*************************************************
*************************/
void turnVibrationOn(uint8_t pins){
  if (pins == 1){
    digitalWrite(vibrationPin1, HIGH);
  }
  else if (pins == 2){
    digitalWrite(vibrationPin2, HIGH);
  }
  else if (pins == 3){
    digitalWrite(vibrationPin3, HIGH);
  }
  else if (pins == 4){
    digitalWrite(vibrationPin4, HIGH);
  }
  else {
    digitalWrite(vibrationPin1, HIGH);
    digitalWrite(vibrationPin2, HIGH);
    digitalWrite(vibrationPin3, HIGH);
    digitalWrite(vibrationPin4, HIGH);
  }
}
/*************************************************
*************************************************
                FSR Functions
/*************************************************
*************************************************
/
/*************************************************
*************************/
/*
   Read value FSR
*/
/*************************************************
*************************/
void readFSR(){
  fsrReading = analogRead(fsrPin);
  if (fsrReading > 10){
    // subtract the last reading:
    total = total - readings[readIndex];
    // read from the sensor:
    readings[readIndex] = fsrReading;
    // add the reading to the total:
    total = total + readings[readIndex];
    // advance to the next position in the array:
    readIndex = readIndex + 1;

    // if we're at the end of the array...
    if (readIndex >= numReadings) {
      // ...wrap around to the beginning:
      readIndex = 0;
    }

    // calculate the average:
    average = total / numReadings;


    if (average < prevReading){ //Check for a minimum
point
```

```
      fsrTopMax = fsrMax;
      if (abs(fsrMax - average) > 15){ // Check the
difference between the Max and current val
        fsrMin = average;
      }
    }
    if (average > prevReading){ //Check for a maximum
point
      fsrTopMin = fsrMin;
      if (abs(fsrMin - average) > 15){ //Check the
difference between the min and the point
        fsrMax = average;
      }
    }


    if ( (fsrTopMax == prevFsrMax) | (fsrTopMin ==
prevFsrMin) ){// Check if the minimum or maximum
has changed values
      LocalCheck = 1; // Set flag to start counter for stable
reading ON
      if (once == 1){
        fsrMinCheck = fsrMin;
        fsrMaxCheck = fsrMax;
        once = 0;
      }
    }
    else{
      diffCheck = 1; //Check for hit
      LocalCheck = 0;  // Set flag to start counter for
stable reading OFF
      check_ms = 0; // Reset stable reading counter to 0
      once = 1;
      prevFsrMax = fsrMax;
      prevFsrMin = fsrMin;
    }
    if (diffCheck == 1){
      fsrCheck = 1; //Set counter for checking alert
      MaxMindiff = (abs(fsrTopMax - fsrTopMin));
      if (MaxMindiff > 5){
        diff_count = diff_count+1;
        runningdiff = ((runningdiff + MaxMindiff));
        running_average = runningdiff/diff_count;
      }
    }


    prevReading = average;
  }
}


/*************************************************
*************************************************
                Timer ISR Functions
/*************************************************
*************************************************
/
ISR(TIMER1_OVF_vect)        // interrupt service
routine
{
  TCNT1 = timer1_counter;  // preload timer
  check = ON;
  if (compressionSystem == ON){
    if (compressHalfSeconds < 240){
      compressHalfSeconds = compressHalfSeconds +
1;
```

```
    }
    else{
      compressHalfSeconds = 0;
      compressionSystem = OFF;
      digitalWrite(compressPin, LOW);
    }
  }
  if (LocalCheck == 1){
    check_ms = check_ms + 1;
    if (check_ms > 2){
      if ((fsrMin == fsrMinCheck) | (fsrMax ==
fsrMaxCheck)){
        diffCheck = 0;
        check_ms = 0;
        fsrHalfSeconds = 0;
        fsrCheck = 0;
        runningdiff = 0;
        MaxMindiff = 0;
        diff_count = 0;
        running_average = 0;
        once = 1;
      }
      else{
        diffCheck = 1;
      }
    }


  }
  if (fsrCheck == 1){
    if (fsrHalfSeconds < 9){
      fsrHalfSeconds = fsrHalfSeconds + 1;
    }
    else{
      if ((running_average) > 40){
        fsrAlert = ON;
      }
      fsrHalfSeconds = 0;
      runningdiff = 0;
      running_average = 0;
      MaxMindiff = 0;
      diff_count = 0;
    }
  }


}
ISR(TIMER2_OVF_vect){// Every 1ms
  /**************Read new sensor
data*******************/
  digitalWrite(checkInt, digitalRead(checkInt)^1);

  /*************Poll ECG********************/
  current_ecg_state = digitalRead(ecgPin);
  ecg_count_ms = ecg_count_ms + 1;
  if ((current_ecg_state == 1) & (last_ecg_state !=
current_ecg_state)){
    beatRecordedMs = ecg_count_ms;
    ecg_count_ms = 0;
    pulseRateBeatsPerMinute = (1.0 * 60.0) /
( (beatRecordedMs*(1.023)) / 1000.0);

    beatRecordedatMs[9] = beatRecordedatMs[8] ;
    beatRecordedatMs[8] = beatRecordedatMs[7] ;
```

```
    beatRecordedatMs[7] = beatRecordedatMs[6] ;
    beatRecordedatMs[6] = beatRecordedatMs[5] ;
    beatRecordedatMs[5] = beatRecordedatMs[4] ;
    beatRecordedatMs[4] = beatRecordedatMs[3] ;
    beatRecordedatMs[3] = beatRecordedatMs[2] ;
    beatRecordedatMs[2] = beatRecordedatMs[1] ;
    beatRecordedatMs[1] = beatRecordedatMs[0] ;
    beatRecordedatMs[0] = pulseRateBeatsPerMinute;

    for (sum_i = 0; sum_i < 10; sum_i++){
        sum = sum + beatRecordedatMs[sum_i];
    }
    averagePulse = round(sum/10);
    sum = 0;

  }
  last_ecg_state = current_ecg_state;

  if (ecg_count_ms == 65535){ // variable overflow
    ecg_count_ms = 0;
  }

  /**************Handle timer
overflow********************/
  TCNT2 = 130;        //Reset Timer to 130 out of 255
  TIFR2 = 0x00;        //Timer2 INT Flag Reg: Clear
Timer Overflow Flag
  }
```