Assignment 8: Laboratory Prototype Documentation

for

The Prosthetic Arm Monitor

Team #2

Erik Gonzalez, *CpE student*, Reynald Garcia, *CpE student*,
Ann Theriot-Thirakoune, *CpE student*, Jimmy Tran, *CpE student*

Abstract --As prosthetic arms improve throughout the years, the cost also rises. Shriners Hospital helps children amputees at no cost. However, they want to confirm that the purchase of expensives prostheses is justified and beneficial for their patients. In order to track whether a patient is using their prosthetic arm, a monitor will be used to record all instances of utilization. Our monitor will help clinicians determine whether the prosthetic arm is useful for their patient or not. If the prosthetic arm is not being properly utilized, they can determine if another patient will benefit more from the prosthetic arm. This paper presents the process of planning, designing, risk assessment, implementing, testing, funding, and the deployable prototype status of our prosthetic arm monitor. In the work breakdown structure, we decided to focus on four main features: sensor systems, data storage, encryption/decryption, and user interface. Our goal was to build a small-form device that would be placed on the prosthetic arm to record the usage and then encrypt the datalog onto a microSD card. After a certain amount of time, the patient would send the microSD card back to the hospital and they would be able to decrypt the data that was stored using an application that we also built. While using the application, the clinician would be able to import the new data from the microSD card, view all of the data, and use the graphing function.

Key Term Index - EMG, GPIO, GUI, Infrared Sensor, Middleware, SD Card

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Children upper-limb amputees are often prescribed a myoelectric prosthetic arm. Myoelectric prosthesis are expensive and medical teams are worried that the prosthesis is being underused. Poor aid of daily activities and a steep-learning curve are the main factors of the underuse of the prosthesis among children. Many medical teams, especially at Shriners Hospital for Children will benefit from a device that can monitor and log the usage of the prosthetic arm. From the data, the medical team can either decide if the patient should be given an alternative or lessons on how to operate the prosthesis. This paper will cover the design idea of the device including its features, product management, risk assessment, deployable prototype status and its marketable forecast.

The ultimate outcome for the project is accomplishing a compact and lightweight device that will operate for a two week timespan on a single coin-cell battery. The main features that are listed in the contract are the sensor system, data storage with encryption, simple user interface and a local back-end. The sensor system will perform data acquisition and will determine two events: arm attachment and hand usage. All data is encrypted before being transferred into a memory card providing data security. A simple graphic user interface will be provided through a custom application. The user can import the data from the memory card through the application to perform data analytics. The data can either populate a table or be represented by a line, bar or pie graph. Our local back-end will ensure data integrity and provide additional data security.

A work breakdown structure (WBS) and program evaluation and review technique (PERT) will be implemented for product management. The WBS will perform a divide and conquer approach to simplify large tasks into subtasks and then into small individual activities. The PERT was implemented to provide a well structured plan that defines an appropriate deadline for all activities. A risk assessment was performed on this device to reduce any risking elements of the design by devising a mitigation plan. It was successful and the project did not come across any major difficulties regarding risky elements.

For the deployable prototype status, all basic features were successfully implemented. However, the device is not ready for public or private use as it will only last three days on a coin-cell battery. The core functionality of the sensor system needs to be reworked to reach the original goal of logging time span of two weeks. The software to aid the medical team with data analytics is ready for the market. After resolving the poor power consumption issue, the device will be ready to enter an untapped multi-million dollar market.

I: INTRODUCTION

The price of prosthetic arms, especially those utilizing myoelectric technology, presents an issue for hospitals: how often are the patients using their arms and how can they best be utilized. To combat this issue, Shriners Hospital has suggested to create a device that monitors how the prosthesis is being used and during what times the prosthesis is active. With this data, clinicians are able to determine if the patient is utilizing the benefits of the arm. If not, appropriate actions can be taken to either encourage more use or switch to a simpler prosthesis so that another can use the arm.

This paper will document the design process taken by Senior Design Team Two, starting with the problem statement and design idea and ending with the final product. During the discussion, topics being presented will include the design process, funding, work breakdown structure, risk assessment, and details about the final deployable prototype

The monitor-system is designed as a wearable, watch-like device. Using a force sensitive resistor and a flex sensor, the device is able to detect when he prosthetic arm is being worn and when the fingers are being bent, respectively. The handling of information is accomplished by the ATmega328p microcontroller, which stores the date and timestamp, and then encrypts the data to be saved on to a micro SD card. The data is then decrypted on the software end, where the data is then organized and presented visually in the form of graphs.

The device was able to meet all basic functions as outlines in our design idea contract. The only issue was battery consumption, as our initial hardware design was not able to be fully implemented and the microcontroller being used was changed to a simpler, but less power-efficient controller. The deployable prototype meets desired functions as stated in our design contract, but does not meet the two-week battery life which was a personal goal for the team, as the battery will last less than a week at its current state.

II: PROBLEM STATEMENT

People with their limbs intact do not realize how fortunate they are until lose one. Around the world, there are individuals who are missing limbs, either from childbirth or from amputation. While there is little to no option of having an organic replacement, prostheses have proven to b e the next best viable option. With the large amount of materials being used to create the prosthetic, the cost increases as well. For this reason, doctors want to make sure their patients are properly utilizing their artificial limbs, to justify the expenses that are involved in purchasing a prosthesis. We will briefly discuss background of limb loss, how prosthetic arms are being used, reasons for limited usage, the expenses associated with the creation and procurement of a prosthetic arm, and how we plan to tackle the societal problem of patients needing to properly utilize prosthetic limbs.

A. Statistics

The percentage of limb loss has increased significantly throughout the world. It has without exception, great economic, psychological, and social impacts. According to the Amputee Coalition, about two-million people are amputated in the United States [1]. The amount of of amputations that occur each year is nearly 185,000 and counting.

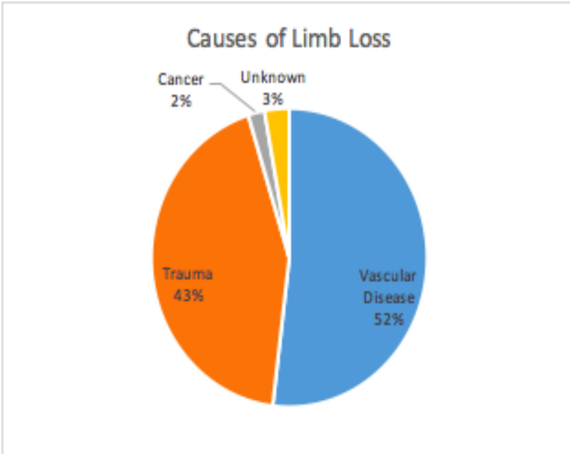

Figure 1: Main Causes of Limb Loss in America [1]

Limb loss can be attributed to several factors. Vascular disease is believed to be the most common amongst all. About 54% of Americans are amputated from diseases such as diabetes or peripheral arterial. Trauma is the second leading cause with impacting 45% of the population. The remaining 2% is cancer related and the rest are from unknown reasons.

In developed countries, about 80-90% of patients that are older than 60 associated to amputations that were caused by vascular disease. Although, in non developing countries (Tanzania, Ghana), the majority of amputees are much younger due to trauma. [2]

B. Usage

1) Learning Curve Among Children

Our societal problem can involve child amputees, in this case child amputees without an arm, and we are tackling the task of monitoring the use of the prosthetic arm. We are evaluating how well and often all of the features provided by the arm are being used by the child. One problem that will arise with children using the prosthetic arm effectively is the learning curve among children. It can be very frustrating for adults to teach children how to use a prosthetic arm because how complicated the artificial arm is. The prosthetic arm will be put to poor use if the user does not properly know how to use the arm.

A solution for this problem is given by Peter A. Smith and Matt Dombrowski, where they designed games for children to learn how to use the prosthetic arm. According to Smith and Dombrowski, child amputee suffers from little exercise of the muscle that is being used to control the prosthetic arm[3]. They solve this problem by in upcoming " future iterations there will be high and low fences that can be jumped with stronger and shorter flexes"[3]. Giving children control how high or short the character in the game can jump by controlling their own muscle contractions is a fun way to train their muscles. Below is an screenshot of one of their games taken from their article.

Figure 2: Screenshot of Smith and Dombrowski's Game [3]

2) The necessity of a upper-limb Prosthetics

Humans are very capable and adaptive beings, lower-limb prosthetics provides amputees the ability to stand and walk and upper-limb prosthetics help with the loss with one or both arms. Smith states that most patients with only one arm can do most tasks without a prosthetic arm because they can use their other arm for various tasks[4].

A prosthetic arm becomes less useful for everyday tasks if the amputee is also an lower-elbow amputee. An amputee is considered a lower-elbow amputee if only their lower arm is amputated and the patient still have their upper arm. A study was done by Helen Y. N. Lindner, on evaluating the functionality of upper-limb prosthetics among children and the results were that only 45% of children use their prosthetics at home[5]. The patient does not use the prosthetic arm very often because lower-elbow amputees can often grab things by wrapping the object with their elbow.

Although the study only consisted of 40 children, it did show that the number of children that did not use the prosthetics arms outside of home did drop to zero and the daily prosthesis use was split 50/50 for using it for over and under 8 hours per day [3]. Because of the limited functionality of the prosthetics, patients only use the arm if they must in order to complete a task.

3) Lifestyle Preference

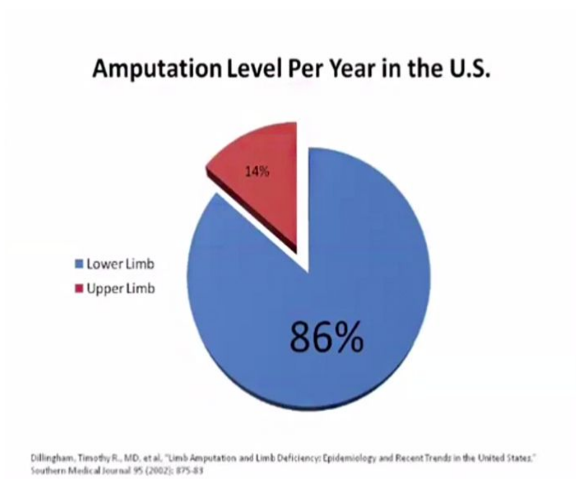

Figure 3: Amputation Level Per Year in the U.S [5]

Dr. Douglas G. Smith, MD stated that most of his lower-limb amputee patients use their prosthesis more than the upper-limb patients (specifically 80% of them). Not to mention, that there are more patients who have lost limbs below their elbows. Lower-limb amputees tend to be more comfortable with their prosthesis [5].

In contrast, upper-limb amputees do not feel the same strong vibrations that lower-limb amputees feel when they walk or run. Since their sensory input to the residual arm is blocked, their vibrations are not as strong when their prosthesis come in contact with objects and feels like a certain disconnection.

Not only is there a disconnection between the person and their prosthesis, their prosthetic arm is loose more often and weighs more. This causes them to be lopsided. For lower-limb amputees, it is more advantages to use the weight from their prosthesis. It allows them to hold them in place and evens out their body position.

Some upper-limb amputees feel much better without a prosthesis and can adjust to life using other parts of their body instead. The others that do own prosthesis, only use it when they need to do a certain task like picking/lifting objects [5].

Unlike the lower-limb amputees, most of them decide not to wear their prosthesis full-time. Even though they do not wear their prosthetic arm as much, everyday for them is still a working progress. It takes some time and dedication to get use to an object that they were not born with. Theses patients are the ones who decide whenever they want to use it and for how long. Usage of prosthesis, is solely a lifestyle preference as it varies for individuals based on their comfort level.

C. Cost Effectiveness

1) Alternatives

Adding to the societal problem is the cost of producing a prosthetic arm. The production of a prosthetic arm can be very steep as it requires certain materials and resources in order to manufacture it. The price of a prosthetic arm can vary from 5000 to 60,000 dollars. Because the price for the arm is very expensive, the resources spent on developing, testing, and manufacturing are relatively in demand for higher priority problems as well as the American healthcare system being flawed.

In addition to development costs, increased customization in the prosthetic actuators, sensors, and controllers are a hefty toll that adds on to the cost of commercial prosthetics. Even the cost of maintenance and replacement can add up as well. These costs make can make prosthetics inaccessible for families and individuals. All these obstacles individuals and families have to face can be difficult but can be solved.

LIMBS, a non profit organization, were able to use a low cost prosthetic for the low income population. They were able to bring an average commercial prosthetic knee cost of 1000 to 10,000 dollars to a knee that only costs 100 dollars. They effectively reduced the price by challenging themselves by developing a prosthetic knee that “could be made in third-world countries with simple tools and locally obtainable materials.” Gonzalez, the founder and president of LIMBS, has also tackled the problem of reusable and maintenance that is a very detrimental component in prosthetics. He says “The big thing was, and is, to design and develop prosthetic components that they can maintain and repair. What happens a lot of times is people give prosthetics components away to these parts of the world and they wear out or break and can't be repaired or maintained locally. That's not a long-term or sustainable solution."[6] LIMBS prosthetic knee meets the international standard as well as be maintained and repaired locally in any developing countries.

2) 3D-Printed Arms

Although the prices are high, it can be brought down to a more reasonable price with better development using 3D printing. With 3D printing, the cost will not be as steep and can go as low as 100 dollars. This is because of the material used in 3D printing compared to commercial prosthetics are inexpensive. For O’neil, they developed a prosthetic arm that has the mechanical structure for the prosthetic entirely 3D printed, while using other forms of alternatives for the EMG sensor and electronics. The total cost for development for a complete prosthetic arm was approximately 640 dollars.[7]The cost reduction allows for more people who can benefit from this. Another example of a cost effective prosthetic arm, according to Arjun, they engineered a low cost functional 3D printed prosthetic hand. For their product, they used a electro-thermal actuation using nylon 6-6 based artificial muscle instead of DC motors which led the material cost to be approximately 170 dollars. He explains the difference between DC driven motors and the electro-thermal actuation to be a cost-effective alternative. “These artificial polymer muscles present a cost-effective alternative to heavy and expensive DC motors typically used for the actuation of prostheses.” The nylon fiber used for the prosthetics is cheaper than other actuators. “The fishing line costs only $5/kg. These fishing line muscles also exhibit little to no hysteresis during actuation. All of these characteristics (strength, low cost, and hysteresis-free) allow the TCP muscles to easily be used in inexpensive prostheses.”[8] In addition to its low cost, it allows for easier and cheaper customization because of its inexpensive material used to print the prosthetic.

D. Implementation

The societal problem at hand is that people are not properly utilizing prosthetic limbs. The solution for our societal problem is a fairly straightforward one which is to monitor the use of prosthetic arms. This will allow doctors to know how frequently the prosthetic is being used as it is expensive and can be given to another patient that can use the prostheses more efficiently.

III: DESIGN IDEA CONTRACT

The monitoring system is required to detect when the prosthesis is affixed to the user and detect when and how often the motorized fingers are utilized. The system is also required to have a data log that is easily retrievable by the hospital.

The overall monitor system will be restricted in size and power consumption. The solution must be able to fit within the prosthetic arm easily, and not add any unnecessary extra weight or impede the function of the arm by blocking or preventing movement due to the monitor's size. The monitor system must also be conservative with power consumption, running on the lowest amount of power as possible. This restriction allows the prosthetic arm to operate for longer periods between charge cycles. Finally, the monitoring system must be restricted to work with current prosthetic appendages, without the need to heavily modify the prosthesis. By creating the monitor in such a fashion that it can be easily added to an existing prosthetic appendage rather than needing a new type of prosthesis, the cost of providing such devices to children will be minimized.

A. Feature - Arm Attachment Sensing

The prosthetic arm monitoring system will detect when the prosthesis is mounted by the user. When the prosthetic arm is attached to the user, a microcontroller with devoted sensors will identify the arm as "in-use". Once the prosthesis is removed, the period which the user had it on will be logged as a single event. Every time the arm is removed and reattached, it will be considered a new event. To fulfill this feature required of the monitor system, multiple methods and components can be utilized. These methods vary in levels of efficiency and will be explored to determine the best solution.

1) Hardware - Microcontroller: The microcontroller will be the brain of the monitoring system, constantly checking to see if the prosthesis is used and logging events. There are numerous microcontrollers that can accomplish this goal.

The Atmel series of microcontrollers are some of the most commonly used ICs and have a large community of support behind them. Depending on the model, GPIO(general purpose input/output) pins range from 5 to 86, allowing the monitor to take advantage of more sensors at the cost of higher power consumption for the more complex chips. From Atmel, the ATtiny85 and ATmega328 are possible components to be utilized.

Texas Instruments has a series of low-power microcontrollers that would also be ideal for the device. The MSP-430XX series operates at extrememly low current, in the microamps range. Some models in this line also allow different forms of memory storage, such as FRAM, which consumes less power than flash memory. However, Texas Instruments controllers are fairly complex in nature, and do not have the same support as other controllers.

After weighing the options, the ATmega controller offers a good balance of simplicity and power consumption.

2) Hardware - Detection Sensors: The monitor must be able to sense when the user has attached the arm. To accomplish this, a few different sensors can be used.

An infrared sensor can be attached to the inside of the open cavity of the arm, where the limb is inserted onto user. The sensor can continuously check if there is an object within a certain distance (the user's inserted appendage) and identify that the user is wearing the prosthetic arm.

An ultrasonic sensor can also be used in a similar fashion: attached to the inner cavity and constantly checking to see if there is an obstruction. Both options are very simple to implement but can obtain false positives rather easily, as any object that falls inside the prosthesis will be logged as the user wearing the device. Both sensors are active components and will have to be constantly powered and read to determine if an object has been inserted to the prosthesis, which raises the total power consumption of the monitor system.

A force sensing resistor is one other option that can be used to achieve comparable results. The sensors can be attached in the same cavity, and will detect when pressure is present, such as when the user inserts their appendage into the prosthesis. The force sensitive resistor, unlike the infrared and ultrasonic sensor, is a passive sensor and will not actively consume power to function. This allows detection of the user while also remaining power efficient. Another advantage held by force sensitive resistors over the previous components is that it is less likely to trigger a false positive, as constant force is required for the arm to be registered as "in-use".

The force sensing resistor is the most suitable component for detecting when the prosthesis is attached, due to its low power consumption and simplicity. The monitor system will feature 2 force sensing resistors, which will be attached to the inner cavity of the prosthetic arm at the base. Multiple components are used to provide more consistent results, only logging an event if both resistors are experiencing force applied to the surface. This method will have to be tested in multiple situations to determine the proper voltage threshold to be identified as active.

3) Software: Arduino

The Atmel microcontroller will be programmed with Arduino IDE. Arduino IDE can be used with any Arduino board and has a large amount of support and libraries available that make communication between devices under the protocols SPI/I2C trivial. Once the code is written and verified on the development board, the code will be uploaded to the Atmega chip on the embedded PCB.

The data will be logged as a text file on an SD card (feature to be described in greater detail under "memory" feature). The microcontroller will be programmed to log the day and time the prosthesis is attached, through an SPI connection to the memory device, and a real time clock module to obtain time.

4) Members Working on Feature:

The arm attachment sensing feature will be completed by team members Erik Gonzalez and Jimmy Tran. Both members will split the work evenly, switching between the microcontroller programming and the physical circuitry creation.

Once the required parts are ordered and received, some time will be needed to test our chosen methods and make any alterations. Completing force sensing detection should be relatively easy, with the main bulk of the work being associated with the incorporation of the microcontroller. An estimated 25 hours of work will be dedicated to this feature, which will include testing the sensors in order to find proper threshold voltages; discovering the most eloquent way to mount the monitor system, research into programming the unfamiliar microcontroller; the actual programming of the microcontroller to properly log the necessary data onto an external memory device, and finally testing of the complete circuitry to assure every component of the feature works together. Another 10-15 hours will be required to discuss the specifications of the prosthetic arm with representatives of Shriners Hospital, to understand how the arm works precisely and to allow time for us to test the controls of the arm. This is needed so that we can operate the arm and control it in a lab situation, testing our monitor, without damaging the device. With all aspects considered, 40 hours of work will be dedicated to this feature.

5) Outcome: The feature will be considered working once we are able to sense when the prosthesis is being worn and when the microcontroller is able to register these events in a log. The microcontroller must be able to distinguish between moments of inactivity, while still wearing the arm, versus when the arm is fully removed and unused. Once these standards are met, this feature will be considered functional.

B. Feature : Finger Motor Detection and Real Time Clock

In order to accommodate as many different styles of prosthetic arms as possible, finger use detection will be accomplished through the use of a simple flex sensor. This allows the device to be used on many different styles, as the only requirement needed of the arm is that its finger are capable of bending. This also allows for the device to be completely external, meaning the device does not need to connect to any circuitry inside the arm and can be worn on the outside.

1) Hardware: Flex Sensor

The flex sensor will be placed along the fingers, and as the fingers contract, the flex sensor bends and causes an activation reading. This is then logged into memory, along with the time stamp.

Figure 4: Flex Sensor [9]

2) Hardware: Real-Time Clock: DS3231

The DS3231 is a low-cost, extremely accurate I^2C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal [9]. This real-time clock will be used to obtain the current time and it will be saved to the our data storage device under the following format, MM:DD:YY. The RTC maintains seconds, minutes, hours, day, date, month, and year information. The DS3231 also automatically adjusts the months with fewer than 31 days, including the corrections for leap year [9]. These two features is why this real-time clock is being used in this design. Below is a table of DS3231's pinout. This RTC clock is chosen because of its backup power-supply input. This allows our system to have a backup battery for our RTC if anything happens to the main power supply.

TABLE I
DS3231 Real-Time Clock PINOUT [10]

Pin #	Pin Name	**SPI Signal Function** SPI Mode
1	32kHz	32kHz Output.
2	VCC	DC Power Pin for Primary Power Supply.
3	~INT/SQW	Active-Low Interrupt or Square-Wave Output. This open-drain pin requires an external pullup resistor connected to a supply at 5.5V or less.
4	~RST	Active- Low Reset
5-12	N.C	No Connection. Must be connected to ground.
13	GND	Ground
14	VBAT	Backup Power-Supply Input.
15	SDA	Serial Data Input/Output.
16	SCL	Serial Clock Input.

3) Software: Arduino IDE

As stated above, under the "Hardware - Detection Sensors", Arduino IDE will be used to program the microcontroller.

4) Estimated Hours for Feature and Members Working on Feature:

The two members that will be working on this feature are Erik Gonzalez and Jimmy Tran. The allocated time for this feature to be implemented is 15-20 hours. The main bulk of the work will establish the communication between the real time clock and our microcontroller.

5) Outcome: Our real-time clock will be considered to be working when our microcontroller will be able to give the time in the following format, MM:DD:YY and HH:MM:SS down to the second. The entire feature will be considered to be working once the microcontroller is able to detect when the motors is being used to control the fingers.

C. Feature: Memory

This project will use a SD (Secure Digital) card to log all data taken from the arm. We chose SD because of it's non-volatile feature, extremely large support, reliability and large capacity (Up to 2GB). In order to use a SD card with a microcontroller, there must be some program space to implement a FAT file system. The File Allocation Table (FAT) file system is a simple file system designed for small disks and simple folder structures [11]. The card must implemented using the FAT for it to be accessed by the microcontroller. Without implementing the FAT system, the SD card cannot be read by a PC, MAC or a Linux Machine through a MMC(Multimedia Card) reader. This section will briefly go over two methods in implementing our SD card in our system.

1) Hardware: SD CARD

Secure Digital is a non-volatile memory card format developed by the SD Card Association for portable device. Non-volatile memory is a type of computer memory that does not require power to hold its data. Today, SDA (SD Association) is responsible for setting the industry standards for SD, miniSD, microSD, and other varieties such as 'XC' and 'HC' that offers larger storage and faster data transfers [12]. SD offers two protocols to transfer data, SD Card protocol and SPI (Serial Peripheral Interface) Protocol. The SD card protocol is more complex than SPI, so in order to conserve time, we will use the SPI protocol.

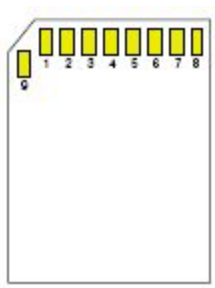

Figure 5: SD Pin Out [13]

TABLE II
SECURE DIGITAL PINOUT [11]

Pin #	Pin Name	**SPI Signal Function** SPI Mode
1	DAT3/CS	Chip Select/Slave Select[SS]
2	CMD/DI	Master Out/Slave In[MOSI]
3	VSS1	Ground
4	Vdd	Voltage Supply [2.7v or 3.6v]
5	Clock	Clock [SCK]
6	Vss2	Ground
7	DAT0/D0	Master In Slave Out [MISO]
8	DAT1/IRQ	Unused or IRQ
9	DAT2/NC	Unused

2) Software: mikroC PRO/ FatFs/ Arduino

In order for our SD card to be read easily through any PC, MAC, or Linux a FAT table must be implemented onto the SD card. Creating a library that is capable of implementing a FAT file system requires special knowledge of the subject and is large task in itself. Fortunately, there are three free fully working libraries that are being maintained, Arduino SD library, mikroC PRO and FatFs. Arduino's SD library works for all audino's development board. mikroC Pro for PIC offers a library for accessing data on Multi Media Card via SPI communication [14]. mikroC is compatible with both SD and SDHC (Secure Digital High Capacity) memory cards, but it the library is only compatible with PIC microcontrollers. PIC is a MCU that is offered by the company, Microchip, below is a schematic of one their MCU and a SD card.

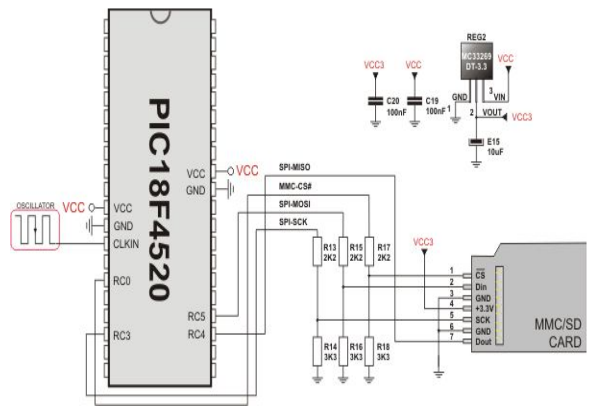

Figure 6: PIC Schematic with a MMC/SD Card [8]

Although mikroC offers a strong and easy-to-use library, we may not use its library due to the fact that we are restricted to the PIC MCU. FatFs is another free library that offers the same solution but it isn't tied to one microcontrollers. FatFs can be incorporated into any small microcontrollers such as 8051, PIC, AVR, ARM, Z80, RX, etc[15].

All programming will be done in the language C and through a supported IDE (Integrated development environment) software such as Arduino or Code Composer Studio.

3) Members Working on Feature:

The members that may be working on this feature are Jimmy Tran and Erik Gonzales. Since there is a large support for SD and multiple libraries to implement the FAT system, this feature doesn't require two members.

4) Estimated Hours for Feature:

The estimated hours for this feature will be 20- 30 hours. A more accurate estimate cannot be given because there are multiple variables that can go wrong during the setup and the debugging time can vary. Researching on the implementation and understanding how the SD card, SPI protocol, and the FAT system operate will take 5 hours. Designing a circuit that can read and write to a SD Card from a microcontroller will take 10 hours. The main bulk of the time will be dedicated to incorporate our circuit onto our design.

5) Outcome: After implementing this feature, our system will be able to store any data taken from the arm by our chosen MCU (Microcontroller Unit). The data then can be analyzed through our custom software by any PC, MAC, or Linux machine. This feature will considered to be fully functional, if it successfully records all actions (will be tested by any members) and anyone is able to open and read the contents in the SD card by normal means.

D. Feature - User Interface

The user interface will be composed of an application that displays the data imported from the SD card, which is located in the prosthetic arm. The data will be taken and then formatted neatly. Using the data given, it will be able to calculate how long the patient has worn and used the prosthetic arm within certain time frames.

1) Hardware

The user interface will be implemented on a desktop platform. Some of the code for the user interface will be written on Apple Macbooks and PCs that are running newer versions of Windows.

2) Software

The software needed to program the user interface will be Microsoft Visual Studio. The main programming language for the front end will be Javascript (specially VueJS), HTML, and CSS. The backend will remain the same with Sqlite for the database. However since we switched over to a offline web application, there has to be a middleware between the front and backend. In this case, we will use a the Laravel API which is a PHP framework that will connect VueJS frontend components to the database. The Laravel API makes data entry more efficient using their Eloquent ORM to make models in order to insert data into the database.

3) Members Working on Feature

The user interface will be completed by all of the team members, but primarily Ann Theriot-Thirakoune and Reynald Garcia. Theriot-Thirakoune and Garcia will work together switching between the front and backend of the Monitor Log.

4) Estimated Hours for Feature:

The amount of time estimated to work on the UI is about 8-12 hours per week between the two members. That includes designing visuals, refreshing our knowledge of the programming language, writing code, and testing it. Altogether, 16-24 hours of work will be dedicated to this feature.

5) Outcome: The user interface will be implemented by a web page application that has a controlled backend to store all of the data. The user will be able to import data from the textfile (via microSD card), select a time frame to see the data of their choice, and graph the selected data into a pie, bar, or line graph.

E. Resources

Our budget for this project will be approximately $500. But the fundamental resource needed for the project will be provided by Shriners Hospital such as a prosthetic arm to work on. As for the other various resources like the SD card, microcontroller, and sensors needed to complete the project will be deducted from our budget of $500. Consulting from Dr. Smith and Archana Bhalerao, former Sac State Masters Student who worked on this project in previous years, will be added as part of our resources because of effective information they can give us for designing our implementation. For lab space, we will be using the University provided lab rooms in Riverside Hall to hold meetings and extensive testing. A meeting at Shriners Hospital might be needed in the near future to discuss and consult about our implementation which will also be deducted from our budget as traveling expenses.

F. Team Members

1) Reynald Garcia

I have moderate experience working with hardware electronics and Java through course projects. But I have a good understanding of concepts taught in previous courses to help improve my experiences. I can also contribute in both hardware and software side of this project, but will be working on the software side.

2) Erik Gonzalez

I have experience working with physical electronics through multiple personal projects. My skills include 3d modeling, 3d printing, PCB design, and soldering. Because of these skills, I will be most useful working on the hardware side of the project, while providing software assistance when needed.

3) Ann Theriot-Thirakoune

I have extensive knowledge in Java, C, Objective C, Python, and HTML. My past experiences include building/designing websites, mobile applications, and databases. Even though I am working more on the software side, I am capable to use my hardware skills into this project as well.

4) Jimmy Tran

I have good understanding of OOP programming concepts and strong foundation on basic electrical circuits. Most of my programming knowledge was self-taught through various outside school projects. Because of my growing interest in electrical circuits, I will focus more on the hardware side on this project and will give some advice regarding the software if needed.

IV: FUNDING

TABLE III
COST ESTIMATION

Item	Price
Launch Pad	$20
Microcontroller	$20
Adapter	$5
SD Cards (2)	$10
Watch Battery	$3
Real Time Clock	$10
Pressure Sensors (2)	$20
Digital Multimeter	$20
Programmer	$120
Jumper Wires (60 pack)	$4
PCB Fabrication	$40
Electrical Board Components	$10
Total	$282

Allocated Budget: $500

V: PROJECT MILESTONES

A. *Sensor System*

1) Order Parts: Parts that were ordered for the sensor system include: microcontroller, real time clock, batteries, and pressure sensors. They were ordered online and used express shipping, allowing the items to be ordered and delivered within a week.

2) *Code Pressure Sensing Logic:* The logic for sensing when the arm is attached was coded in a week. This includes the logic for sensing adequate pressure (enough pressure to simulate an arm being pressed against it), writing the timestamp of the event to an SD card, and only outputting information when the action is first performed (writing the timestamp when it is first inserted, rather than constantly outputting during the whole duration of being worn).

3) *Code Finger Movement Sensing:* The logic for sensing finger movement was coded in a week. This included detecting when a voltage above 1V is detected from the flex sensor voltage divider, writing timestamp to SD card, and outputting timestamp only when the action is first performed.

B. Data Storage For Functions

1) Code MCU Interface to SD Card: The logic for interfacing to the SD card took 1 week to code. This includes establishing a connection between the controller and SD card, as well as file creation and file editing.

C. User Interface

1) *Microsoft Visual Studio and Various Package Installations:* The installation of these were to compile and write the necessary code for the Graphical User Interface. The installation took about a day. Certain things would not compile if certain packages were not installed correctly or placed into the folder where the project is saved.

2) *Code Web Application:* The code for the web application when the user wants to access information about the prosthetic arm use. The functions in this application are displaying old/new data, importing new data coming from the microSD card, inserting new data manually, calculating the usage when the patient puts on/off the arm, and graphing when the arm is being used. This was coded in a span of six weeks.

3) *Implement Database to Application:* Applying the database to the Application to establish a connection to allow data being flowed in and out of the application to be used. This involves code for the web application when importing a text file and exporting for graphical and login purposes. The code for decrypting the incoming cipher text from the SD card is within the import logic. The logic for this took about two weeks to complete.

VI: WORK BREAKDOWN STRUCTURE

The purpose of this project is to provide a method to easily monitor how often patients are utilizing their prosthetic arms, by implementing a hardware system to monitor movements and software to visualize relevant information into easily read charts.

This document will provide a breakdown of the different components of the monitor system, describing each task in a hierarchical fashion such as:

1. Sensor Systems
    - 1.1. Attachment
        - 1.1.1 Sensor Placement
    - 1.2 Movement
        - 1.2.1 Finger Movement

2. Data Storage for Functions
   2.1 Hardware
      2.1.1 Secure Digital Card
3. User Interface
   3.1 Front End
      3.1.1 Graphical User Interface
   3.2 Back End
      3.2.1 Database
4. Project Management
   4.1 Course Assignments
      4.1.1 Team Activity Reports
      4.1.2 Problem Statement and Elevator Pitch
      4.1.3 Design Idea Contract
      4.1.4 Work Breakdown Structure
      4.1.5 Project Timeline:
      4.1.6 Risk Assessment
      4.1.7 Project Technical Review
      4.1.8 Laboratory Prototype Documentation
      4.1.9 Laboratory Prototype Presentation
      4.1.10 Updates and Revisions
      4.1.11 Device Test Plan
      4.1.12 Market Review
      4.1.13 Mid-Term Progress Review and Testing Results
      4.1.14 Feature Presentations and Reports
      4.1.15 Deployable Prototype Review
      4.1.16 End of Project Documentation
      4.1.17 Deployable Prototype Presentation

A. Task - Sensor Systems

Team members will work on establishing a system of sensors that will detect how the arm is being used. Including when the arm is attached to the user, and when the different parts of the arm are moved.

1) Subtask - Attachment

Two members will develop the system for detecting when the prosthetic arm is attached to the patient, using force sensitive resistors to detect pressure.
a) Activity - Sensor Placement : The associated members will find the best location to place the force sensitive resistors on the prosthesis, that will result in most consistent results. They will be placed in such a manner that the patient will apply force a sensor when attaching the arm, minimizing false-positives results.

2). Subtask - Movement

Two group members will also focus on a system of sensors to track how the arm is utilized.
a) Activity - Finger Movement: The associated members will locate the best position in the circuit to detect when the motor controlling the fingers is active. The location for the sensor will have to be at a point where activating the finger causes the most notable crease to the flex sensor, to ensure a reading can be made.

B. Task - Data Storage of Functions

1) Subtask - Hardware

Two members will develop the system for storing data. The data that is taken by the sensor systems will be stored for future analysis by Shriners Hospital for Children.
a) Activity - Secure Digital Card: The members will design a system that will allow the microcontroller to save it's flash memory data into a SD card. The data will be saved into the microcontroller's flash memory after every intentional event. Every 15-mins the data that is stored in the microcontroller's flash memory will then be transferred and saved into a Secure Digital Card. This two step process is done to block any interference with the prosthetic arm EMG sensor when the data is being uploaded. In order for the SD card to be read in Windows, Linux, or MAC, the data will be saved in a FAT file system.

C. Task - User Interface

Team members will work on an application that will display a child's usage of the prosthetic arm. The data stored in the microSD card will be read and imported into the application. The user will be able to see what day, time, and movement (Arm(Inserted), Arm(Removed), Fingers) when the prosthetic arm has been used. They will also be able to select the specific time frames to be displayed and the results can be shown on a graph (bar, line, pie).

1) Subtask - Front End

Two team members will create an application that will be visually appealing and user-friendly.

a) *Activity - Graphical User Interface:* A team member will design how the application will look like. When the GUI design is approved by all members, then each necessary item will be implemented using VueJS, HTML, and CSS. Once the coding process is completed, the next step will be to test it. If there are any problems with the application, then changes will be done within the code or design. The application will be tested continuously until all of the functions work correctly.

2) Subtask - Back End

a) *Activity - Database:* One team member will be in charge of designing an offline database as well as handle the middleware to organize the information being imported from the microSD card. The database will store data to ensure efficient communication with the application. The developer will be using a Sqlite database to store data in. The middleware being used is the Laravel API to connect front end components to the Eloquent ORM model which is then connected to the backend database. When coding is done, debugging will need to be done by the same developer to check for any possible errors when importing, exporting, and decrypting. At the end of this process, both team members working on the software will collaborate to effectively clean up both aspects of the software side.

D. Project Management

Team members will complete various activities throughout the year, as required to finish the project.

1) Course Assignments

a) *Team Activity Reports*: Each week, team members will write a report that explains what they have done during the week, the total hours spent on the project for that period, and the hours they plan to dedicate for next week. The team leader will combine each team member's report and summarize it into one whole report. [16]
b) *Problem Statement & Elevator Pitch:* The team will choose a societal problem, come up with a potential engineering solution, and present it to the class. [16]
c) *Design Idea Contract:* The team will decide on a solution on how to solve their societal problem, how they plan to implement it, and assign each member jobs for each task. [16]
d) *Work Breakdown Structure:* The team will create a chart and write a report that highlights the main tasks and subtasks of the project. [16]
e) *Project Timeline:* The team will create a project timeline that will illustrate the project's progress over both semesters of Senior Design. [16]
f) Risk Assessment: The team will evaluate the various risks involved in the project and create a document that assesses these risk. The document will be a clear evaluation of

the impact risks have on a project of a given size. [16]

g) Project Technical Review: Team members will review the technical aspects of the project for accuracy and completeness. The document will review the various technicalities involved in the project to determine if anything needs to be corrected. [17]

h) Laboratory Prototype Documentation : The team will create a prototype to demonstrate the functionalities of the project. When doing this, the team will create a document that will delve into the process of creating the prototype as well as describe in detail the functions and specifics in the prototype. [17]

i) Laboratory Prototype Presentation
The team will demonstrate a working laboratory prototype that performs the functions that are listed in the feature set. At the end of the presentation, the audience will understand how the design address the societal problem and what will be completed at the end of the second semester. [17]

j) Updates and Revisions
As the team make progress building the design, they will have a better understanding of the societal problem compared to the first week of the first semester. The team will then review the initial design idea contract to ensure that the design fully implements all the features stated in the design idea. [17]

k) Device Test Plan
The team will create a plan that tests their design under a range of critical factors. Some factors that can be used to test the design is temperature, accidental movement, and voltages. [17]

l) Mid-Term Progress Review and Testing Results:
The team will attempt to explain how they will implement the changes to the design after the device test plan. The team will demonstrate in an oral presentation. [17]

m) Market Review: The team will research the market associated with our product, and write a report based on our findings. The report will identify our consumer base and our competition. [17]

n) Feature Presentations and Reports: The team will write a report describing in detail the different feature sets associated with our project. [17]

o) Deployable Prototype Review: The team will present the important features of our deployable product. A punch list will be created and be tested against our product, as we present to the other senior design groups. [17]

p) End of Project Documentation: The team will document the development of the deployable prototype, over the course of the full two semesters. The members will include the work needed to design the idea, assemble a lab prototype, perform research, and to create the deployable prototype. [17]

q) Deployable Prototype Presentation: All members will present the the final product of the project. The deployable prototype will be showcased described to viewers who walk by the booth.[17]

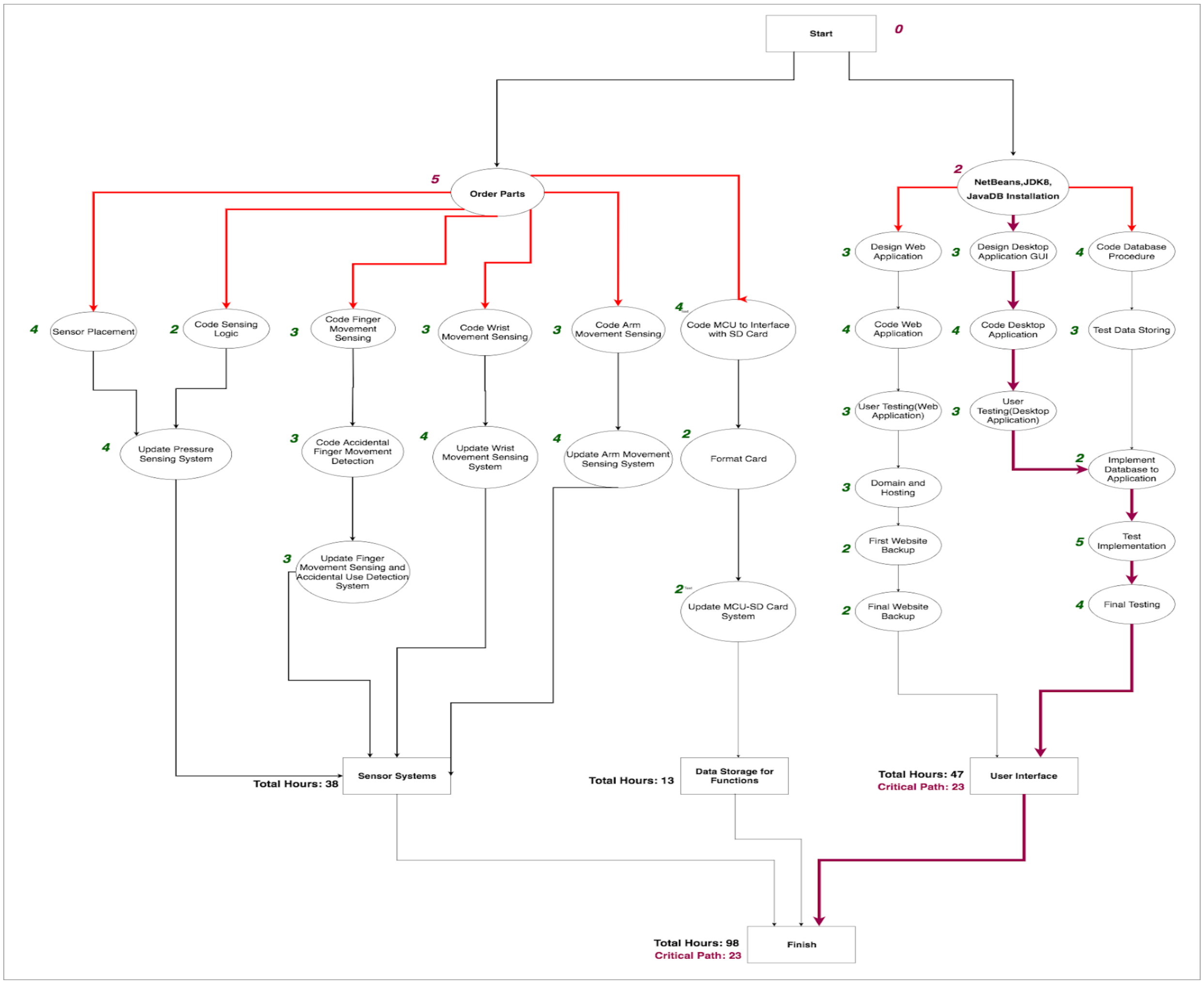

Figure 7 - Pert Diagram [20]

VII: RISK ASSESSMENT & MITIGATION

All projects will inevitably face obstacles. These obstacles must be anticipated and corrected in the most efficient matter, as to not delay the project or increase costs. This paper will describe what potential problems can arise during our project, and how we plan to mitigate those issues.

A. Critical Path

For every project, there will be some risk factors. It is important to identify the critical path to reduce these risks. Looking at Figure 7, the software (right side) has the longest overall duration. The critical path is linked to seven tasks from the User Interface section: NetBeans/JDK8 Installation, Design Desktop GUI, Code Desktop Application, User Testing (Desktop Application), Implement Database to Application, Test Implementation, and Final Testing. This link is critical since half of the project is based on the user interface. Each subtask is dependent on the subtask before it. Without completing a subtask, it will cause a hold on the rest of the subtasks after it.

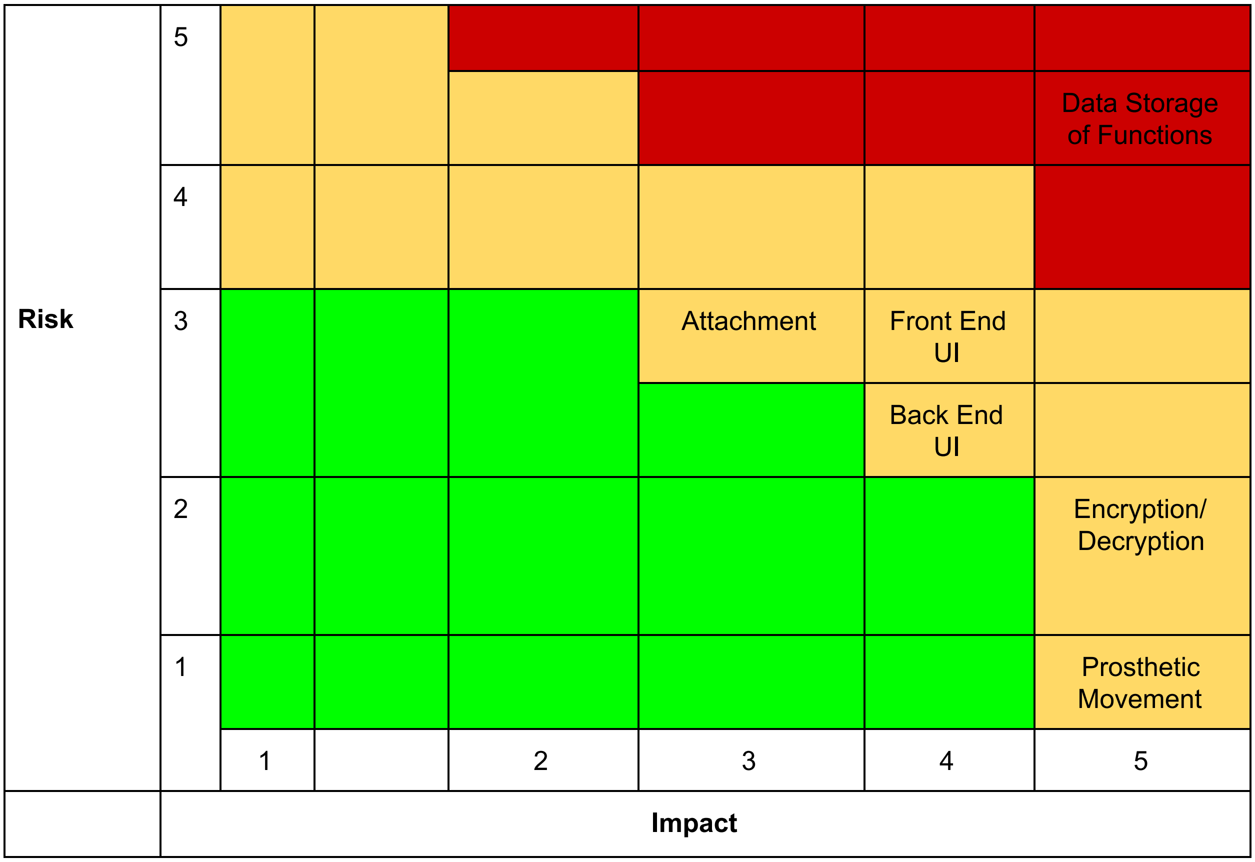

Figure 8 - Risk Matrix [21]

B. Risk Assessment and Mitigation

1) *Attachment (Medium Risk, Medium Impact)*

a) *Risk - FSR Out of Range:* It is possible that the FSR sensors can be either too sensitive or not sensitive to meet our needs. The sensors must be sensitive enough to easily detect when an arm is inserted, but not so sensitive that it creates false positives.

b) *Mitigation:* The sensor can be positioned differently so that more or less pressure is applied when inserting the arm. If the sensors range is extremely out of the required range, new sensors can be purchased that lay within the proper range.

2) *Prosthesis Movement (Low Risk, High Impact)*

a) *Risk - Power Consumption:* The device is mobile so it is very important that it be energy efficient. The devices used to detect finger movement, such as flex sensor and RTC, cause current draw and deplete the battery

b) *Mitigation:* Finding solutions that have low current draw. Flex sensor can be used in a voltage divider, with a 10k resistor which decreases the amount of current being drawn.

3. Data Storage of Functions (High Risk, Very High Impact)

a) *Risk - SD Write Operation: Performing a write* operation to the SD card can consume a current upwards of ten miliampers. This feature will risk depleting the 230 mA coin cell battery that powers the monitoring system.

b) *Mitigation:* A microcontroller with a large (1024 Kbytes) onboard flash memory will be chosen so that the system performs the write operation less frequently. The data is first saved to the microcontroller's onboard flash memory, then it is transferred over to the SD card via a write operation.

c) *Risk - File System:* The secure digital card must be formatted to the FAT32 file system. It is possible that the microcontroller we use does not have pre-built library that can format the SD card.

d) *Mitigation:* Arduino offers a function that handles the FAT32 file system.

4. Encryption/Decryption (Low Risk, Very High Impact)

a) *Risk - Timing:* Data encryption and decryption will happen separately at different times. The same encryption method must be implemented while encryption and decryption.

b) *Mitigation:* Jimmy and Reynald will communicate and decide on one encryption method.

5. *User Interface-Front End (Medium Risk, High Impact)*

a) *Risk - Reading Operation:* The application must be able to automatically read the SD card once it is inserted into the computer. It should take the data that is saved on the microSD card and organize it according to the application. The user should be able to do two main things: look at data during a selected time frame and place it into a graph. It is possible that the application will not read the data correctly and leave blanks or null values.

b) *Mitigation*: The application has to match the format that it is being imported from the SD card. All timestamps will be read in this order: "Day.Month.Year, Hour:Minute:Second, Activity,". The comma will be a separator when it gets read into the application.

6. User Interface-Back End (Medium Risk, High Impact)

a) *Risk - Database Issues:* Since the database is a local relational database, the .db file is saved locally on the computer. The data from the application is being stored into the ,db file. For any reason if the file gets deleted or compromised, no data will be saved.

b) *Mitigation:* The data will still be saved on an external hard drive. The SD card can be used to save the data that is being used for the application. But for storage limitations to record data onto the SD card, it is better to use another source of an external hard drive.

VIII: DESIGN DOCUMENTATION

A. Hardware

An arduino was used as the microcontroller for the prototype design. This allowed us to build and test rapidly, as the controller is simplistic in nature.

The final design utilizes an ATmega328p microcontroller, embedded on a custom PCB. From there, we used force sensitive resistors to detect when the arm is being worn; a flex sensor to detect when the fingers are being used; a real time clock module to keep track of the date and time; and an SD card module that interface the microcontroller to the SD card.

B. Data Storage

This is one of the most important features of this design. A secure digital memory card was deemed to be the best fit because of it large support and price. A SD card was also chosen because it removed the possibility of any interference due to accessing the monitor through a computer. The client can access the data by removing the SD card from the monitor and reading it from the computer.

C. User Interface

1) Front-End

The web application for the monitor log was written in VueJS, HTML, and CSS. When the application is opened through a localhost, the user will see the login screen. When the login details match with the credentials saved in the backend, it allows the user to access the whole application. The user will be able to see all of the patient's information and perform actions that are clicked such as importing data, graphing, and the overall calculation of the arm usage.

2) Back-End

For the backend database, we used Sqlite provided by Firefox's free add-on, Sqlite Manager. Using the add-on, we were able to see the contents of the stored data in the database. The main purpose was to manage and store our data and call the data back when need be. The database interacts with the frontend components (VueJS) when it is being called to import or export data. Decryption is done in conjunction with importing data to ensure the correct data going into the database. There are two tables for the login and main table that holds all of the patient's activities. Not only are we able to import data from the textfile, but we are able to manually add individual entries using the Sqlite Manager.

D. Mechanical (3D-Printing)

1) Prosthetic Arm

A simple representation of a prosthetic arm was created in order to showcase the features of the monitor device. The arm was 3d-modeled in Autodesk Fusion 360, and then printed to usable scale. The arm features a base, with open cavity on the underside to attach the force sensitive resistor for detecting when the arm is worn. It also features a hinge mechanism to simulate the opening and closing of fingers.

2) Device Case

A watch-style case was created to house the device. Case and watch-band were both created in Autodesk Fusion 360 and printed to usable scale.

IX: DEPLOYABLE PROTOTYPE STATUS

The deployable prototype performs the following functions: detects when force is applied (simulating the user inserting the prosthesis), detects when the fingers are used, keeps track of time, and writes relevant information to a text file on an SD card.

A force sensitive resistor is used to detect when the arm is being worn. The detection works by passing 5V through the FSR, which is in series with a 10k Ohm resistor. The 10k Ohm resistor is inserted in series to create a voltage divider, so that the varying voltage can be read across the reference resistor.

The voltage divider allows us to read the voltage at the node located between the FSR and 10k Ohm resistor. This would bring our observed values into a usable range, as the voltage difference at this node is a ratio of the resistances used in the voltage divider, multiplied by the input voltage.

The range that the microcontroller can read for analog signals is 0-1023. With this setup, values between 0- 1000 were observed when applying different levels or pressure. A value of 500 was used to determine if enough pressure was applied to register as the arm being worn. This portion meets the desired design idea as it is able to detect when the arm is being worn and has not shown any cases of false positives in the final iteration of the code.

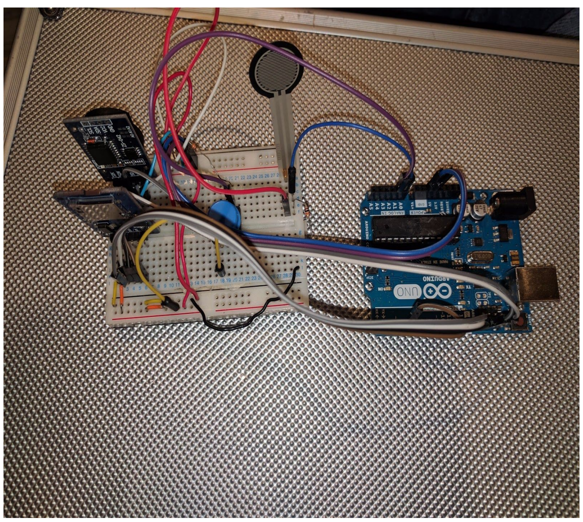

Figure 9 - Prototype Layout on Breadboard [22]

To detect finger movement, a flex sensor is placed along the fingers in such a manner that bending the fingers causes the sensor to bend as well. The sensor is setup in a similar fashion to the FSR, in that a voltage divider is used to read when the sensor is being used. The sensor is able to identify when the fingers are closed, and logs the information as expected. New cases of finger movement are not logged until the fingers are straightened and fall below the original threshold, to ensure multiple logs are not created by a single use. This portion of the device operates as according to the design idea, as it is capable of detecting finger use and no cases of false-positives have been recorded.

To keep track of time, a ds3231 RTC module with auxiliary battery was used. This device is capable of maintaining time after the main VCC power source has been disconnected, through the use of a LIR2032 coin cell battery. The date and time were set upon initial use of the device, via premade libraries, and maintained by the module with the constant power from the coin cell battery. The module is read from using i2c protocol every time an activity takes place, such as the attachment and removal of the prosthetic arm as well as any detection of finger movement.

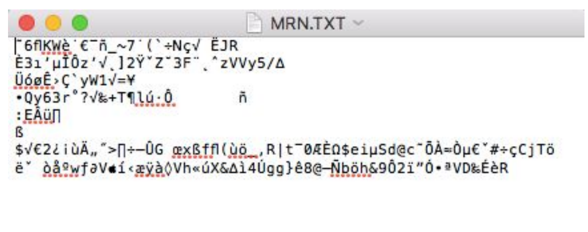

Figure 10 - Encrypted Text File [23]

Once the activity is detected, the timestamp is stored with other information such as what activity was performed, and saved to a text file on an SD card through SPI protocol. This feature meets the desired design idea as the relevant information, such as activity performed and timestamp, are successfully being written to an sd card. Information is stored within the microcontroller memory until memory is filled, at which case, the data is encrypted and saved to the SD card. In Figure 10, the text file is what will be decoded by the software end and used to create the visuals for displaying arm utilization on the application.

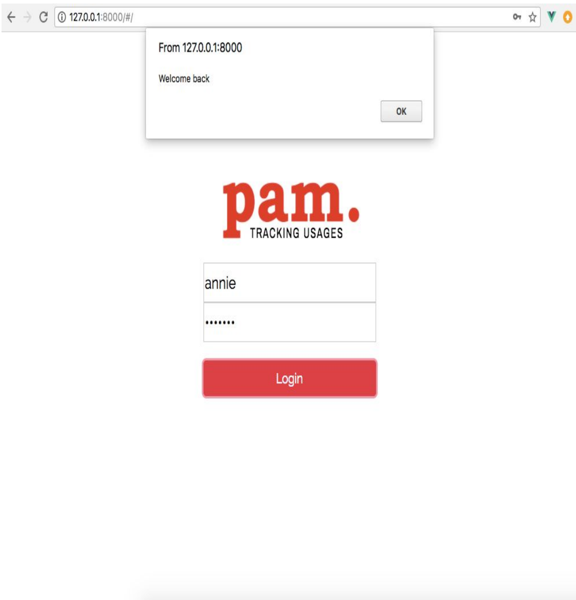

Figure 11 - Login Page [24]

When the web application is opened through the localhost, it prompts the user to enter their username and password. Then, a welcoming message will appear and the user is directed to the dashboard of the application, where they are able to perform functions that are on there. The table will display all of the activities that were stored beforehand from the backend database in ascending order.

Figure 12 - Dashboard Page [25]

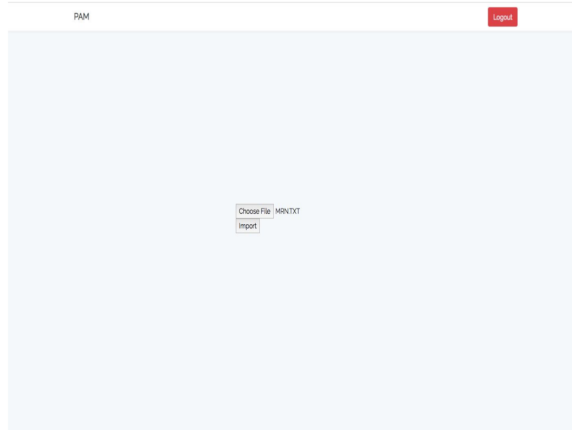

Figure 13 - Import Page [26]

The user is also able to add data by importing the encrypted textfile by clicking on the import button. It will redirect the user to the import page where the user must choose a file and then import it. After selecting a file, it will read the encrypted file and decrypt the text using the same format it was encrypted in. The format has to be AES 128 CBC. It also goes through other verification checks to ensure that the ciphertext can be decrypted.

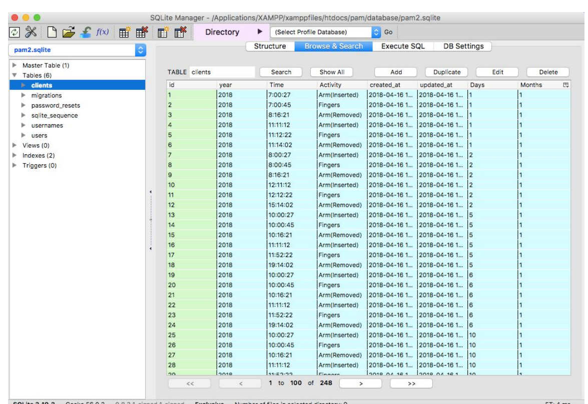

Figure 14 - Sqlite Manager Database [27]

After this process, the data will be parsed to fill the table in the database for graphing purposes. The application will read the new data that is being inserted and check if it is following the format. For the textfiles, it will search for “Month.Day.Year, Hour:Minute:Second, Activity,”. The colons and commas are important because they are used as separators. Once they are separated, they are turned into strings and placed into the correct column. The user can select a time frame they want and the table will display all data during that time.

There are three types of graphs that the user is able to use: bar, line, and pie. The graphs focuses on the usage of this year. It displays all 12 months and the number how many days the arm has been used during those months. Points in the graph will be at zero only if the arm was not used during that month or they are upcoming months. For the pie graph, when the user hovers over a specific color, it shows the amount of days that .

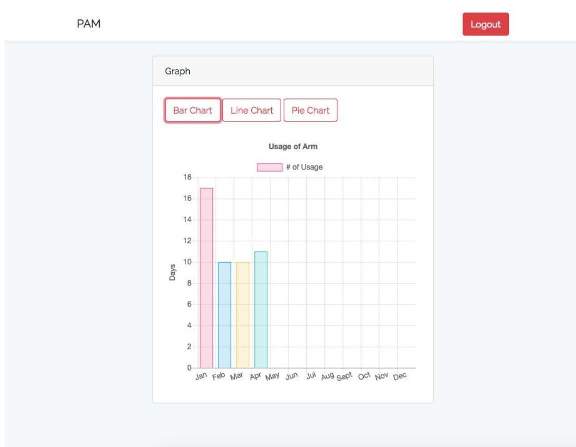

Figure 15 - Bar Graph [28]

X: DEPLOYABLE PROTOTYPE MARKETABILITY FORECAST

A. Product Innovation

After doing some research on our competitors, we realized there are a few things that can be added to make our product more marketable: longer battery life, waterproof, and bluetooth. By adding these features, it will put our product on top of the market. At the moment, the battery for our prosthetic arm monitor lasts less than a week. There are activity monitors that lasts about 18-25 days. In order to compete with the other existing products, the battery life has to be greater or somewhat close to that range. Another good feature would be waterproof. As a result of making it waterproof, it protects it from any damages to the prosthetic arm monitor that comes from a spill, hand washing, bathing, or swimming. Lastly, would be bluetooth. Instead of having the patient send back the microSD card after every two weeks to the hospital, data can be transferred over quicker using bluetooth. It will decrease the waiting time for the clinicians view new data. Not only that, but they would not have to worry about losing data when the package is lost in the mail or stolen.

B. Client Feedback

During our Deployable Prototype Review, we were able to get feedback from our sponsor from Shriners Hospital, Anita Bagley. Even though she enjoyed our presentation, she did mention a couple of things that could be added. The first suggestion was to add gripping to our movements for the prosthetic arm. At the moment, we are have two sensors. It detects when the arm is inserted or removed and any finger movement. For the finger movement, it records when the sensor has been touched. However, the clinicians do not know whether the finger movements were actually doing something or an accident. Her other suggestions was to implement more graphing functions. She explained that the more detailed the graphs are, the better. Not only do the doctors want to see how many days of the month the arm has been used, but she wanted to know how many hours during the day or week.

XI: CONCLUSION

Children that are in need of prosthetic arms are often given myoelectric prostheses. The cost of these myoelectric prostheses can be very expensive and doctors want to ensure that the prosthetic arm is being properly utilized. They want to check how frequently a child is using a prosthesis so they can decide whether to give the prosthetic arm to a patient who will better utilize the arm. Exploring this societal problem helped expand our understanding of the issue and in turn, allowed us to create a solution. This solution aligned with what Shriners Hospital for Children were pursuing and we were able to collaborate on ideas to develop a working prototype.

Since the discussion with Shiners, the team was able to formulate a design idea to solve the societal problem. The design idea involved a compact lightweight device that will operate for a two week period on a single coin cell battery. The features listed in the design idea contract are the sensor system, data storage with encryption, a simple user interface, and a local back end. The sensor system will perform data acquisition of two distinct events: arm attachment and hand usage. To ensure quality data security, encryption is done before the data gets saved into memory card. This will provide a form of HIPAA compliance that is needed when handling patient data. The graphical user interface is a local host offline web application that will provide the user data analysis. The data in the UI will have the option to be presented in a table or as a graph, such as a pie, bar, or line graph. The local back-end provides another layer of data integrity where data will be stored and saved.

Project Management is another integral piece when working with a team to design a system. The work breakdown structure (WBS) and the program evaluation and review technique (PERT) provided a layout of the work delegated by each team member to work on key tasks on the design. The WBS paved an efficient path to completing the project as a whole. In order to do this, larger tasks were broken down into smaller sub tasks and then into individual activities. While the PERT gave us clear deadlines to finish these activities.

A risk assessment was also performed and analyzed in this project to reduce the risks involved when designing the system. A mitigation plan was needed to provide the necessary steps for avoiding any risks involving the system. The project did not come across major difficulties involving the risks. Luckily, it was successful during its assessment.

As for the status of the deployable prototype, all of the basic features were successfully implemented and functional with the exception of a few mishaps. The device will not be ready for any public or private use because of this. The coin cell battery can only last for three days. The core functionality of the sensor system needs work in order to provide data for a two week period. Fortunately, the software for data analysis is ready for market. When the power consumption issue gets resolved, the entire system will be ready for the market.

This document provides a good understanding of a societal problem that needs attention. Expensive prosthetic arms for children need to be monitored effectively to help doctors and clinicians decide whether the prosthetic arm is being properly utilized. The solution formulated in the design idea contract is our take on how to develop a monitoring system. Without project management and risk assessment, designing this system would have taken longer or have not even seen the light of day. But the system we have designed is almost ready for the market with a slight mishap. Other than that, the design has great potential in an untapped market.

REFERENCES

[1] Ziegler-Graham K, MacKenzie EJ, Ephraim PL, Travison TG, Brookmeyer R. Estimating the Prevalence of Limb Loss in the United States: 2005 to 2050. Archives of Physical Medicine and Rehabilitation 2008;89(3):422-9.

[2] Chalya PL, Mabula JB, Dass RM, et al. Major limb amputations: A tertiary hospital experience in northwestern Tanzania. *Journal of Orthopaedic Surgery and Research*. 2012;7:18. doi:10.1186/1749-799X-7-18.

[3]P. A. Smith and M. Dombrowski, "Designing games to help train children to use Prosthetic Arms," 2017 IEEE 5th International Conference on Serious Games and Applications for Health (SeGAH), Perth, WA, 2017, pp. 1-4.
doi: 10.1109/SeGAH.2017.7939265

[4] Helen Y. N. Lindner, M., Ann-Christin Eliasson, & Liselotte M. N. Hermansson. (2014). Influence of standardized activities on validity of Assessment of Capacity for Myoelectric Control. Journal of Rehabilitation Research and Development, 50(10), 1391-1400.

[5] Smith, D. G. (2007, March 1). Introduction to Upper-Limb Prosthetics: Part 1. [Online] http://www.amputee-coalition.org/resources/upper-limb-prosthetics-part-1/

[6] N. Macias. (2012, Sep. 26) Making LIMBS for the poor [Online] http://engineering.utep.edu/announcement092612b.htm

[7] C. O'neil, " An advanced, low cost prosthetic arm" in SENSORS,2014 IEEE, Valencia, Spain. 2014. Doi:10.1109/ICSENS.2014.6985043

[8] A. Arjun et al. "Design of a 3D printed hand prosthesis actuated by nylon 6-6 polymer based artificial muscles" in Automation Science and Engineering (CASE), 2016 IEEE International Conference. Fort Worth, TX, USA. 2016. Doi:10.1109/COASE.2016.7743499

[9] "Flex Sensor (2.2')." Solarbotics Ltd., solarbotics.com/product/50808/.

[10] "DS3231 Extremely Accurate I²C-Integrated RTC/TCXO/Crystal - Maxim." Extremely Accurate I²C-Integrated RTC/TCXO/Crystal - Maxim, www.maximintegrated.com/en/products/digital/real-time-clocks/DS3231.html.

[11]"Fact Sheet." Fact Sheet - SD Association, www.sdcard.org/about_sda/fact_sheet/index.html.

[12] "NTFS: FAT File System." NTFS.com. The FAT File System, www.ntfs.com/fat-systems.htm.

[13] Davis, Larry. "Secure Digital Card Pinout, SD Flash Card Memory Module Pin Out." Engineering Portal, www.interfacebus.com/Secure_Digital_Card_Pinout.html.

[14] Marko Zivanovic. Multi Media Card Library, download.mikroe.com/documents/compilers/mikroc/pic/help/mmc_library.htm.

[15] ChaN. "FatFs - Generic FAT Filesystem Module." Http://Elm-Chan.org, Http://Elm-Chan.org, elm-chan.org/fsw/ff/00index_e.html.

[16] F. Belkhouche, (2017) Fall 2017 Senior Design [Online], Available: http://athena.ecs.csus.edu/~belkhouf/

[17] R. Tatro, (2017) ECS Hive, Assignments [Online], Available: https://ecshive.ecs.csus.edu

[18] Internet Center for Management and Business Administration, Work Breakdown Structure[Online], Available: http://www.netmba.com/operations/project/wbs/

[19] D.L. Cook, 1966, Program Evaluation and Review Technique – Applications in Education, U.S. Department of Health, Education and Welfare, Office of Education [Online], Available: http://eric.ed.gov/

[20] R. Garcia & J. Tran (2017) PERT Diagram Design. Sacramento State University.

[21] R. Garcia (2017) Risk Matrix. Sacramento State University.

[22] E. Gonzalez (2018) Prototype Layout on Breadboard. Sacramento State University.

[23] J. Tran (2018) Encrypted Text File. Sacramento State University.

[24] A. Theriot-Thirakoune (2018) Login Page. Sacramento State University.

[25] A. Theriot-Thirakoune (2018) Dashboard Page. Sacramento State University.

[26] R. Garcia (2018) Import Page. Sacramento State University.

[27] R. Garcia (2018) Sqlite Manager Database. Sacramento State University.

[28] A. Theriot-Thirakoune (2018) Bar Graph. Sacramento State University.

[29] E. Gonzalez (2017) Block Diagram of Sensor System. Sacramento State University.

[30] E. Gonzalez (2018) Schematic of Sensor System. Sacramento State University.

[31] E. Gonzalez (2018) Snippet of the Encrypted File. Sacramento State University.

[32] R. Garcia (2018) Block Diagram of the Software. Sacramento State University.

[33] A. Theriot-Thirakoune & R. Garcia (2018) Flowchart of the user's perspective. Sacramento State University.

[34] A. Theriot-Thirakoune (2018) Line Graph. Sacramento State University

[35] E. Gonzalez (2018) Model of a Prosthetic Arm. Sacramento State University.

[36] E. Gonzalez (2018) Prosthetic Arm Monitor Case. Sacramento State University.

GLOSSARY

Electromyogram (EMG)- A display formed by electromyography

General Purpose Input Output (GPIO) - Pins that can be controlled by a user as either an input or input on an integrated circuit controller

Graphic User Interface (GUI) - A method of interaction with a computer and human that uses graphical images and widgets

Infrared Sensor- A device that detects infrared radiation that senses things of its surroundings

Middleware - A bridge that connects an application and database

Secure Digital Card- A type of memory card

APPENDICES

APPENDIX A - USER MANUAL

A. Sensor Attachment

Placement of the sensors will vary depending on the individual prosthetic arm. Placement will be determined by the prosthesis technician when the arm is being fitted to the individual. The technician will determine the best location for the force sensitive resistor, so that the patient creates contact with the sensor. This is to be done in similar fashion to how the technician determines location for the myoelectric sensors.

B. Device Attachment

The PCB is to be inserted to the case, with battery side facing upward. The band is then connected around the wrist, and device worn similar to a watch. Insert battery and sd card. The cover to the case is then attached by sliding the cover through the slots located at the top of the case.

C. Data Transfer

Insert the micro sd card into computer either through USB adapter or micro sd-to-sd adapter. Open the file explorer and locate the sd card folder. Drag the text files associated with the datalog onto computer.

D. Log into the Application

In order to use the application, enter the username and password that is given. Once the correct information has been entered, the user will gain access to the application.

E. Importing Data

When the Import button is clicked, you will be redirected to a page where you can import a textfile from the microSD card into the application. The application will not accept data unless the textfile is encrypted. Once the textfile has been imported, the page will be redirected to the dashboard page. There, the new data will appear along with the data from before.

F. Graphing Data

There are three types of graphs to choose from such as the bar, line, and pie. The graph can change by clicking on the different buttons for the graphs.

G. Exiting the Application

To exit the application, you can either close your browser or click on the logout button located on the top right of the screen. Once you log out of the application, you will not have access to it until you log back in.

APPENDIX B - HARDWARE

A. Block Diagram and Documentation

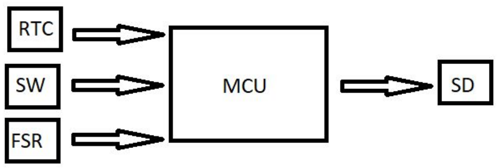

Figure 16 - Block Diagram of Sensor System [29]

The real time clock, switch to simulate motor, and force sensitive resistor are input components to the microcontroller, while the SD card is an output component. The above figure shows these components in a simplistic block diagram form. When an action is detected from the switch or FSR, the real time clock is used to acquire the date and time. This information, as well as other relevant information such as the action performed, is saved as a text string and then output to a.txt file stored on the attached SD card.

The microcontroller being used in this example is an Arduino Uno, the real time clock, switch, FSR, and SD card reader are all generic non-branded modules. The real time clock uses the DS3231 precision RTC IC.

B. Schematics and Documentation

The figure below shows the schematic layout of the sensor system hardware. The real time clock is connected to the MCU via a i2c connection, while the SD card module is connected via a SPI connection. They are connected using 4 wires and 6 wires, respectively. When the voltage values for the FSR and finger detection reach their threshold values on pins A2 and A3, respectively, the MCU will log the event. The date and time is read over i2c connection on pins A4 and A5, where it is then saved in a string, along with the specified event.The string is then saved to the SD card over SPI, using pins D11,D12,and D13 for data transfer (mosi,miso,sck).

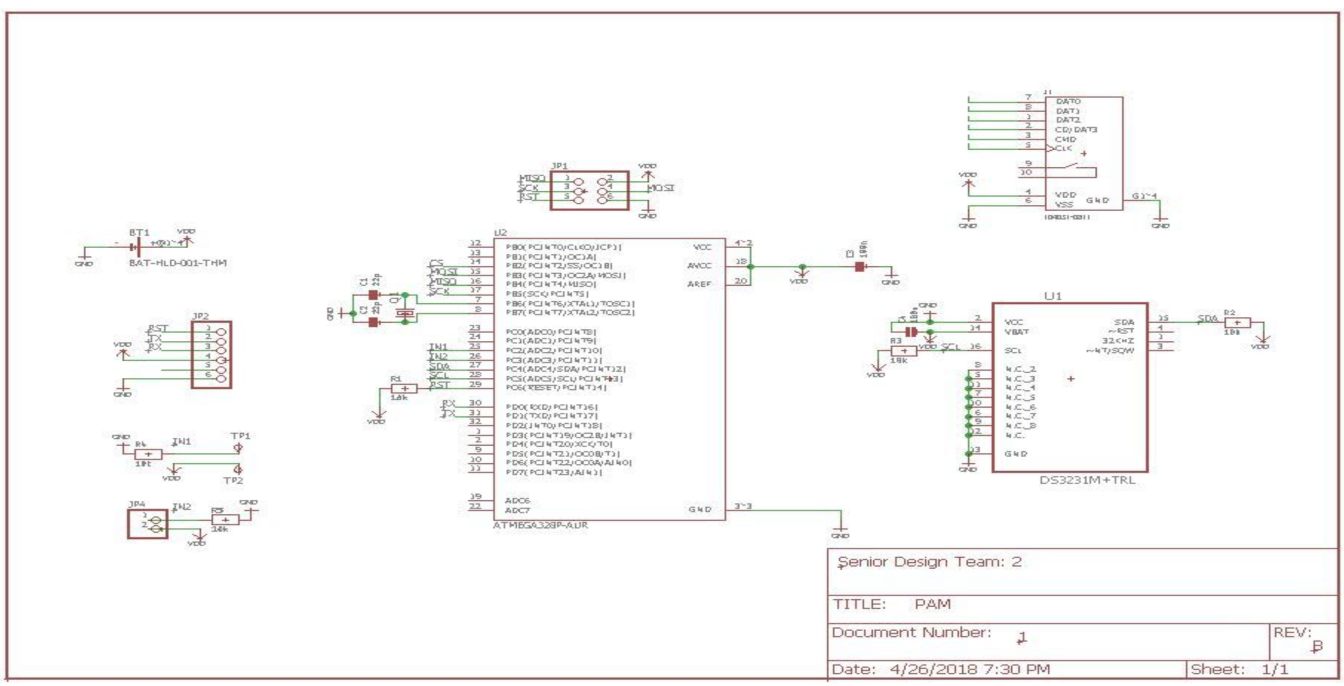

Figure 17 - Schematic of Sensor System [30]

C. Testing

1) Battery: New battery will be tested to have a voltage between 3.7-4.2 Volts, indicating the battery is fully charged and operating within the expected voltage range. A multimeter will be used to read the voltage. This test will be done by Jimmy Tran and should take roughly 5 minutes.

2) Sensors: The flex sensor will be tested to have varying increasing resistances while being bent by hand. A multimeter will be used to read resistance. The force sensitive resistor will be tested in a similar manner, with resistance decreasing with pressure if working correctly. This test will be done by Erik Gonzalez and should take roughly an hour.

3) Microcontroller Unit (MCU): The MCU will be placed on a test board, along with the minimal circuitry in order for it to function (resistors, capacitors, clock crystal, etc). Once placed, a voltage between 3.7v-4.2v will be supplied to the controller. Once powered, the controller will be running a test program that will flip the GPIO pins high and low, so the pins can be measured working through a multimeter.

The program will also set the GPIO pins that will be used to read the sensors as inputs, and test voltages between the minimum and maximum allowed GPIO voltages will be supplied and read through a monitor. The voltages used will be within the common DC characteristics range of the particular MCU, and will more specifically focus around the upper and lower hysteresis limits. This range for the Atmega MCU is specified as roughly .3*VCC for V_{IL}(max voltage for logic 0) and .6*VCC for V_{IH}(min voltage for logic 1).

Cipher Block Chaining (CBC) and Advanced Encryption Standard (AES) will be tested by encrypting and decrypting a test text file using a pre-made library. In order to pass the test, the contents of the test file should be exactly the same before encryption and after decryption. This test will be done by Erik Gonzalez and Jimmy Tran. It should take roughly four hours.

4) Real Time Clock: The real time clock IC will be tested to confirm time is advancing and is doing so in accordance to the National Institute of Standards and Technology (NIST) US clock. Any time within 1 minute of the NIST clock will be considered acceptable for our purposes. This test will be done by Erik Gonzalez and should take roughly two hours.

5) Printed Circuit Board (PCB): All hardware will be placed onto a PCB and should be tested for any broken traces. The test will include using a digital multimeter (DMM) and a magnify glass to check for continuity and any broken traces. This test will be done by Erik Gonzalez and should take roughly three hours.

Finally, the controller will have all sensors attached, and a monitor will be read while applying pressure to the aforementioned sensors to confirm that the MCU and sensors work in conjunction, after already being tested independently.

D. Results

1) Battery: The battery was tested to verify that the actual voltage corresponded with the rated voltage using a multimeter. The 3.7V battery was measured to have 4.0V. The result aligned with our expected voltage value and we were able to confirm that the batteries were operating within the voltage range of 3.7V - 4.2V.

2) Sensors: The two sensors that were tested were the force sensitive resistor and the flex sensor. The sensors were tested to ensure that the impedance value changed as pressure was applied.

A multimeter was connected to the end of each sensor to determine if the items were functioning properly. The force sensitive resistor was pressed, causing the resistance to go from infinity (open circuit) to 400ohm. The datasheet provided with the sensor indicated that the lowest value possible is 200ohm at 20 lbs of pressure, so this value operates in the expected range.

The flex sensor was then bent while the resistance was being read by the multimeter. Bending the sensor caused the resistance to increase to a value of 100kohm, which is within range of the maximum resistance of 125kohm as specified by its datasheet.

Both sensors reacted according to expectation when pressure was applied.

3) Microcontroller Unit (MCU): The AES library passed the test, and it encrypts 128 bits of data successfully. The encrypted data was also compared to a separate online AES tool to confirm the encryption accuracy and the two results were identical. The ciphertext was also tested by decrypting the text using the AES tool, and again it successfully output the original plain 128-bit data.

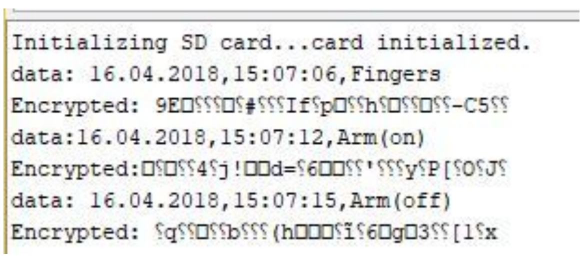

Figure 18 - Snippet of the Encrypted File [31]

4) Real-Time Clock (RTC): For our purposes, the real-time clock was tested for accuracy by the minute and time advancing according to the National Institute of Standards and Technology (NIST) US clock. The real-time clock IC was set to a time of 5:16, and then allowed to advance to a time of 7:00. This was compared to the real world time as given by NIST, and was confirmed to be accurate within 60 seconds, confirming that time was advancing correctly.

5) Printed Circuit Board (PCB): The printed circuit board was tested for continuity. A multimeter was used to test that all points were connected appropriately and aligned with the board schematic. While the traces were not damaged and were aligned with our schematics, two issues were found in the board layout. A few traces were not connected in the design, a failure caused by moving previously made traces and neglecting to reconnect them in the design software. The encountered errors were:

- Reset pin of MCU not connected to reset pin of programming pins.
- VCC of MCU no connected to battery pin.

For testing purposes, these areas were corrected by soldering jumper cables to the intended locations. A secondary PCB was fabricated to correct the issues.

APPENDIX C - SOFTWARE

A. Block Diagram and Documentation

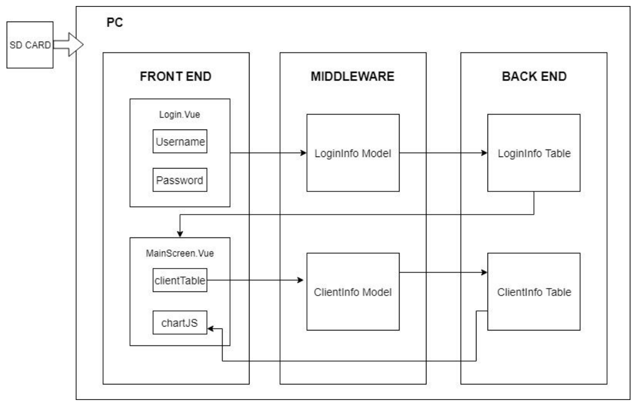

Figure 19 - Block Diagram of the Software [32]

The user will be prompt with the login screen (Login.Vue) where they will be asked to type in their username and password. When the user clicks the login button, it will check with the backend to see if the username and passwords exists in the LoginInfo table. Before checking the database if the username and password exists, the front end interacts with the middleware where the LoginInfo Model is connected to the LoginInfo Table. If the login information does not match, then the user will be directed back to the login screen. Once the login information has been verified, then the user gets access to the application's main screen. The dashboard screen(MainScreen.Vue) has all of the functions and displays the data inside of a table called clientTable. The table in the front end displays the data that is stored from the backend table, ClientInfo Table. Like before, when calling data from the backend, the clientTable communicates with the ClientInfo Model in the middleware before getting data back from the ClientInfo Table. Whatever happens to either the frontend or backend table will affect each other, since they are in sync. In order to graph anything (ChartJS), the data from the backend table will be grabbed and will display a pie,bar, or line graph.

B. Flowcharts, Pseudocode, and Documentation

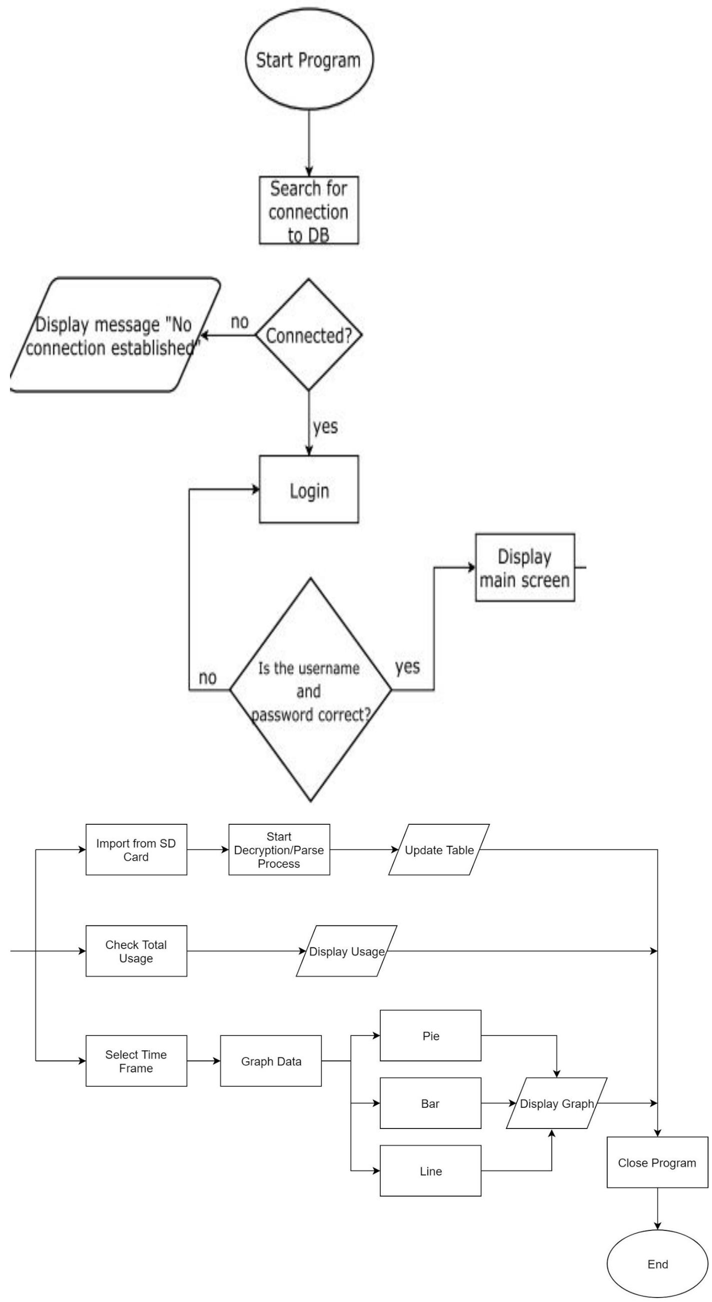

Figure 20 - Flowchart of the User's Perspective [33]

The application checks to see if the backend database is connected. When the connection has been established, the user will be asked to login. After putting in the correct login information, the user will see the main screen where they are able to do multiple things. The user can either import data from the encrypted textfile (from the SD card), check the total usage of the arm, or selecting a time frame to graph the data.

TABLE IV
LoginInfo Table

Element	Type
ID (Primary Key)	Incremented numerically until max amount of 9223372036854775807
Username	Text
Password	Text

At the option of importing data, decryption happens here where it checks if the same format that was being used when encrypting the data. The matching format has to be in AES 128 CBC mode along with the key and initialization vector for the process to go through. Although the user does not deal with this because decryption happens seamless within parsing the data for import, it is an important step when putting data into the database. When the user is done with the program, he/she can just log out of the application.

TABLE V
Client Info Table

Element	Type
ID (Primary Key)	Incremented numerically until max amount of 9223372036854775807
Month	Text
Day	Text
Year	Text
Time	Text
Activity	Text

C. Testing

1) File Decryption: The encrypted file that will be on the microSD card and will seamlessly be decrypted when opening the file on the computer. To test this feature, independent text files will be encrypted to ensure the decryption is working properly with the given private key. This test will be done by Reynald Garcia and should take about four weeks.

2) Graphical User Interface (GUI): The code will be organized into methods that are labeled by their specific function. Therefore making it easier for the programmer to find. After changing or adding new lines of code, the programmer must run it to check for errors. This makes it easier to figuring out where any problems are. The programmer also will continuously test the whole application to make sure that all of the functions work properly (importing, saving, displaying the correct information from the selected timeframe ranges, calculations of the overall usage of the arm, the user's selection of graphing of a bar, line, or pie), shows the all of data coming from the backend database, and is overall simple for the user to use. This test will be done by Ann Theriot-Thirakoune and should take about two weeks testing it.

3) Database: The programmer will ensure the database is connected and filled with patient information. Importing text files to tabulate the database can be tested by importing independent files to make sure the database is working. The connection between the database and the application is also of value. The programmer will make sure the query connected to the application is properly compiled to make sure the database will be filled. This test will be done by Reynald Garcia and should take about four weeks testing it.

D. Results

Testing the user interface is independent of the hardware limitations. Using arbitrary data to test the functionalities of the user interface allowed for complete testing.

1) File Decryption: To test the decryption end of the AES accelerator from the hardware side, debugging each step was important. The various steps taken to ensure that the content of the file was correct include: checking the hex dump of the cipher text to confirm it is divisible by 16, comparing given hex dump to that of a secondary online conversion tool, and confirming no data was left out from an array after decryption.

Overall, the testing resulted in the same contents as the original plain text file before it was encrypted. One problem that occurred during testing was that when importing an encrypted file, the file would be unable to be read. After rigorous debugging, we concluded that the function fputc() that writes to the microSD card was creating a new line with two bytes instead of one. Customizing the function, allowed for the decryption process to go through and was able to get the original contents of the encrypted file.

2) Graphical User Interface (GUI):

After any changes were made to the user interface, it was imperative that the new code passed compilation and operated as expected. Since the web application is running through a localhost, we are able to see the changes instantly by going to a link that was provided in the terminal (in my case, http://127.0.0.1:8000/#/). The first page that is seen is the login screen that asks the user for username and password. Entering the incorrect credentials would not allow access into the web application since it is case-sensitive. Once the user inserts the correct username and password, the page is redirected to the dashboard page where they are able to perform certain functions.

Based on the user's perspective, the user is able to do these functions: import data from the text file that was saved on the microSD card, select a time frame to view the data that they chose, and graph the selected data into a pie, bar, or line graph. The application was able to do all of those functions.

Figure 21 - Line Graph [34]

3) Database: The SQLite Database needed to import and save the original contents of the encrypted file and also fetch data for the UI to use for graphing purposes. To test both the saving function and fetching function, arbitrary values were used to import the data into the database. The results were as expected.

APPENDIX D - MECHANICAL

A. Prosthetic Arm

A simple representation of a prosthetic arm was 3d modeled in Autodesk Fusion 360. The purpose of the arm was to show how the monitor system would attach to a real prosthetic arm. To accomplish this, the arm was designed to have an open cavity where a force sensitive resistor could be placed, to simulate a prosthetic arm being attached. A hinge mechanism was also created to simulate fingers, with a flex sensor attached to detect when the fingers were being bent. A routing hole was created on the arm where the monitor system would sit, so that wires from the device could be routed through the model. This allows the device to look clean, while allowing a user to wire the device to an in-serial programmer for the purpose of debugging. The model was then 3d printed in two parts, the base and fingers, and assembled.

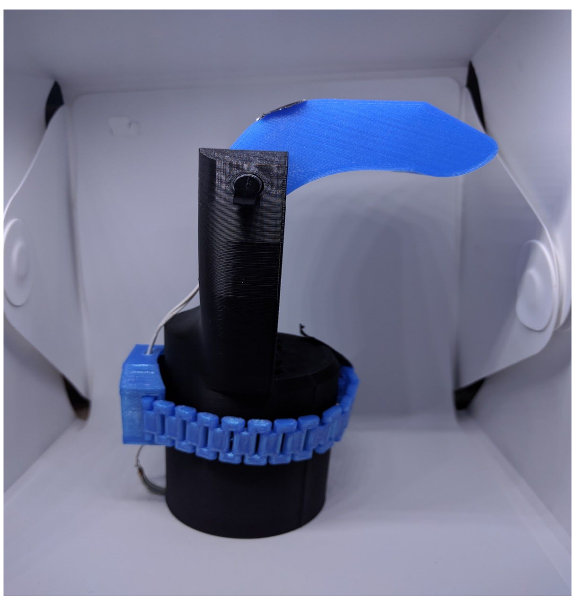

Figure 22 - Model of a Prosthetic Arm [35]

B. Case

The monitor system required a means to attach to the prosthetic arm. A watch-like design was model and printed to accomplish this task. The casing was designed to fit the PCB, while remaining as small as possible. For this reason, the shape is simplistic and closely conforms to the shape of the PCB, to ensure the design is as sleek as possible.

Slots were designed at the top and bottom of the case, to allow the wires for the sensors to pass through easily. The band is connected to the main case by inserting into pressure fit slots on both sides. This allows for the main case, which will always remain the same size, to be printed separately, while the band can be printed at different lengths depending on wrist size. The band can then be glued or epoxied.

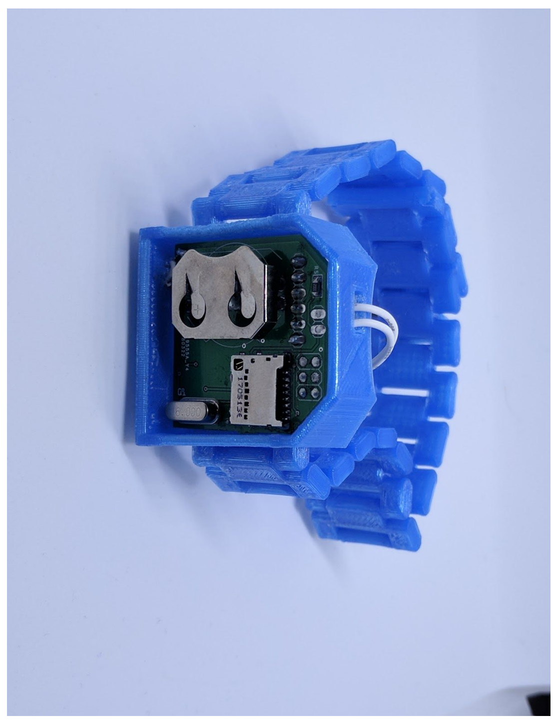

Figure 23 - Prosthetic Arm Monitor Case [36]

APPENDIX E - VENDOR CONTACTS

Contacts

There were a couple of people who have contributed to our project this semester. These contacts are Anita Bagley from Shriners Hospital and Dr. Warren Smith from Sacramento State University. Warren Smith is an amazing electrical and electronic engineering professor here at Sacramento State that specializes in biomedical engineering. He provided a list of projects for students in senior design and we were fortunate to select one of the projects. Not to mention, he gave us more ideas from former students that have done similar projects and let us borrow an older version of a prosthetic arm to study. Anita Bagley is the representative from Shriners that Dr. Smith put us in contact with. She is the co-director of the Motion Analysis Laboratory at the hospital. We have met Anita a couple of times to discuss details about the project and she was able to attend our technical review last semester and deployable prototype review this semester for feedback.

Thank You Letter

April 28, 2018

To: Dr. Warren Smith, Anita Bagley
Re: Thank you

Dear Dr. Smith and Anita,

Team Two from the 2017–2018 Senior Design class at Sacramento State University would like to give thanks for your guidance and support in designing the Prosthetic Arm Monitor (PAM). Your assistance had a huge impacted in the making of our hardware and software of the project. Thank you for taking the time to share your knowledge and experience with us. We really appreciate everything that you have done.

Sincerely,
Jimmy Tran, Ann Theriot-Thirakoune, Erik Gonzalez, and Reynald Garcia

APPENDIX F - RESUMÉS

ERIK GONZALEZ

OBJECTIVE

Seeking a position that will make use of my proven skills in programming, circuitry, and electronics.

EDUCATION

California State University, Sacramento 2012 - Present

Bachelor of Science in Computer Engineering (on-going)

Bachelor of Arts in Music (on-going)

-Current cumulative GPA of 3.539
-Dean's Honor Roll recipient 7/10 semesters.
-Expected graduation Spring 2018

PROJECTS

Bar-top Arcade Cabinet.

- Arcade emulation system
- Fully fabricated and designed by hand
- PC based
- Utilizes replacement front-end interface
- Batch scripts coded to enable parallel software cooperation

Sega VGM Audio Player

- Music recreated using authentic hardware of the Sega Master System video game console
- Utilizes TI SN76489 Digital Complex Sound Generator IC.
- Designed around the Arduino Mega
- Decodes original Sega VGM audio files and transmits proper data over 8-bit bus at regulated clock cycles
- Alternate Arduino sketch to allow use as software controller

MIDI Controller

- Manipulation of music-based sequencing software and audio synthesizers
- Designed around the Arduino Mega
- Utilizes multiple potentiometers (slide and rotary), tactile buttons/switches, and LEDs
- Communicates to devices through MIDI protocol

Portable Game Emulator

- Game emulation device housed within a small case
- Designed around the Raspberry Pi Zero
- Lithium Ion battery for portability

Custom Parts for Hobbyist "Gameboy Zero" Project

- Custom designed 3d modeled parts
- Custom designed PCBs
- Successful marketing and selling of custom parts

Laser Tag Game (School Group Project)

- Laser tag game utilizing IR sensors and Arduino
- XBEE radio communication

Foot Controller

- Controller for musical performance
- Communicates to device through MIDI protocol
- **Created for client: Music Major Undergraduate student**

Concert Attendance Cataloging Program

- Java program with Excel integration
- Excel sheets analyzed and relevant info cataloged
- Object oriented design
- Front-end GUI
- Results exported as Excel sheet / student search feature used through GUI for quick info retrieval
- **Created for client: Music Department – Event Coordinator**

KNOWLEDGE AND SKILLS

Languages: Java, C, C++, Python, JavaScript, Verilog, VHDL, x86 Assembly
Hardware: Arduino, Raspberry Pi, Parallax Propeller, Spartan 3E FPGA Board, 3d Printer
Software: Fusion360(Engineering 3d Modeler), Eagle Cad (PCB Design), Pspice (Circuitry Analysis)

PROFESSIONAL EXPERIENCE

Associated Students Incorporated(ASI) Sacramento, ca 2012 - 2016
Stage Manager –Interacted directly with both the performers as well as audience members to provide an enjoyable experience for all involved. Required quick, calm reactions when faced with unforeseen issues in a high-pressure situation, such as in front of a

large audience. Also, acted as supervisor and delegated duties to workers under me for large high school band festivals and competitions of over 500 people.

MEMBERSHIPS

Institute of Electrical and Electronics Engineers (IEEE) 2016- Present

- 3d Printing Officer

REFERENCES

References available upon request

Reynald Garcia

OBJECTIVE

- To obtain a position or an internship related to the field of Computer Engineering

EDUCATION

- Bachelor of Science, Computer Engineering,California State University, Sacramento Expected: Spring 2018

GPA: 2.9

COURSES

Advanced Computer Organization
Advanced Logic Design
CMOS & VLSI*
Computer Hardware System Design
Computer Interfacing & Microprocessors
Computer Networks and Internets*
Data Structures and Algorithms
Electronics
Operating System Principles*
Signals & Systems

**In Progress as of Fall 2018*

PROJECT EXPERIENCE

Laser Tag System *October 2016 - December 2016*

- Designed and constructed a two player laser tag game with a glove gun that uses an IR transmitter and receiver.
- Developed a glove that communicated through an Arduino system with an XBee.
- Assembled a system which included an LCD screen to scorekeep and a sound system to signal start, end, and increment.

Matrix Display *October 2015 - December 2015*

- Programmed a display using various instructions and interrupts in x86 assembly language to recreate a random character drop down from the movie, *The Matrix*.

Piano Keyboard with a Joystick Toggle *September 2016 - October 2016*

- Created a piano keyboard using multiple push buttons that played an octave of a piano using a speaker connected to a Propeller Board.
- Utilized a joystick to switch between octave scales of the piano.

Pipelined Datapath *February 2017 - May 2017*

- Simulated a datapath with a control unit for a pipelined system that can execute various types of instructions as well as handle hazards and exceptions.
- Validated its functionality with debugging individual components through a stimulus test bench then combined through a top level module.

KNOWLEDGE AND SKILLS

Computer Languages

- C/C++, Java, Python, System Verilog, VHDL, Assembly x86, MIPS(RISC)

Software

- Windows, Mac OS, Linux, VMware, Matlab, PSPICE,Multisim, Xilinx ISE Design Tool, Microsoft Office, Alteris,Visio

Machines/Microcontrollers

- Digital Multimeter, Oscilloscope, Function Generator, DEO Nano, Arduino, Raspberry PI, Propeller, FPGA

Communication/Organization

- Fluent in two languages: English and Tagalog.
- Possess analytical and technical skills to complete various hardware and software projects.
- Demonstrate an ability to efficiently complete multiple projects under strict deadlines.
- Debug and correct setbacks in projects such as an IR transmitter's limited distance.

Leadership/Management

- Lead a sole proprietorship by managing a virtual private network to buy and sell sneakers to customers.
- Coached a high school volleyball team and was able to construct a healthy facility for growth as a young adult and an athlete.
- Played in the Sac State League of Legends team through the American Collegiate Championship by Riot Games

WORK EXPERIENCE

High School Volleyball Coach — Van Nuys High School - Los Angeles, CA — 2010 - 2014

Ann Theriot-Thirakoune

OBJECTIVE To obtain a job or internship related to computer engineering

EDUCATION Bachelor of Science (In Progress), Computer Engineering – CSU, Sacramento

COURSES TAKEN

Computer Software Engineering	Network Analysis
Data Structure and Algorithm Analysis	Advanced Computer Organization
Systems Programming in Unix	Computer Networks and Internet
Computer Hardware Design	Discrete Mathematics
Electronics I	Signals & Systems
CMOS & VLSI*	Operating System Principles*

*Fall 2017

SKILLS

Great communication skills	Fast Learner
Great customer skills	Attention to detail
Team player	Works well under pressure
Self-motivated	

COMPUTER SKILLS

Programming Languages - Java, Assembly (x86), C, Objective C, Python, Verilog, VHDL
Operating Systems - Mac OS, Windows 7 & 10, Linux
Software - Microsoft Office, Adobe Photoshop, Cadence, AutoCAD, Arduino, Visio, XCode

PROJECTS

Software Engineering Project (January-May 2015):
On a team of five, we created a database (using Java) that recorded the attendance for California State Parks Off-Highway Motor Vehicular Recreation. We were able to take the given data and organized it into a larger functional program. The clients were able to view data by fiscal/calendar, calculate totals, compare parks, graph, import/export data, print, etc. The project included Software Requirement Specification (SRS), Software Design Document (SDD), and creating the application itself.

PROFESSIONAL ACTIVITIES

Member	Association for Computing Machinery chapter at CSUS	2014 - Present

Jimmy Tran

OBJECTIVE

Highly motivated individual seeking position where I will have the opportunity to further advance my professional career.

EDUCATION

California State University, Sacramento, **B.S. Computer Engineering**

Expected graduation **May 2018**

SKILLS

Programming Language Experience:

Java, C++, C, Assembly x86, Linux/Unix Systems, Verilog HDL, VDHL

Software:

Visual Studio, Eclipse, ModelSim, Multisim, MatLab, Adobe Photoshop, Microsoft Office, Arduino IDE, MPLAB, MPLAB X

Communications, Analytical, and Leadership

Leading a three member team in a high interactive, fast paced and competitive zone

PROJECT EXPERIENCE

Laser Tag Game

Working in a 4 member team where we created a 2 player laser tag game. The game included a scoreboard, two wireless gloves [guns], and a vest that can receive signals from the wireless gloves. This project enhance my communication and technical skills.

Programming Project — Game AI

Programs a decision making process with JAVA on an ongoing open source project. The script will decide what action the player should make depending on its surrounding and situation. This project gave me very strong cognitive skills.

High School Senior Project — Pool Table

Worked individually under the supervision of a building trade's professor to design and construct a professional 8-foot pool table, I drew plans and purchase the parts needed. The project gave me strong communication, technical, and time managing skills.

WORK EXPERIENCE

Produce Clerk **WinCo Foods** **February 2, 2012-Present**

Perform receiving duties including: writing orders, verifying orders against invoices; reporting shortages and claims to ensure proper billing.

AWARDS AND ACHIEVEMENTS

Dean's Honor List, California State University, Sacramento, fall 2016 Highest Honors,
Cosumnes River College, fall 2014 and fall 2015
Sheldon Building Trades Award: Recognition for outstanding craftsmanship 2010-2011
Working 32+ hours per week, while carrying 14+ units and maintaining a CSUS GPA of 3.64