California State University Sacramento

Engineering and Computer Science Department

EEE 193A/CPE 190

Deployable Prototype Documentation

Instrumentation with Diabetic Shoes

Team Member: Kyle Arango, Pratik Patel , Khoi Nguyen , Samuel Woldeyes

Team 10 – Sunrises Shoe

Kyle Arango, Pratik Patel , Khoi Nguyen , Samuel Woldeyes

College of Engineering and Computer Science
California State University, Sacramento

Abstract - Costs of diabetic treatment and risk of amputation have long been the unavoidable reality for many diabetic patients around the world. To this day, there are no affordable and efficient solutions to eliminate cost, risk and prevent amputation in the first place. To make this happen, we aim to integrate sensory technology with a diabetic shoe to reduce the likelihood of complications occurring. The technology integrated will reduce the need for face to face interaction between the patient and caring physician, while allowing monitoring of the patient via accurate data. The project has been split up into a number of tasks which have been split between team members with possible mitigation plans for tasks with a high likelihood of failure.

Keywords - sensory technology, monitoring, accurate data, tasks, mitigation plans, likelihood of failure

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

EXECUTIVE SUMMARY

Our project integrates sensory technology with a therapeutic shoe designed to prevent foot ulcers in diabetic patients. Our additions to the shoe will collect data to help monitor the pressure distribution across the feet throughout daily use to ensure the shoe is operating as designed. The data collected throughout each day of use will be sent to a server where it can be viewed for further analysis. We aim to have minimal user interaction for more convenient use by the patient, thus the shoe will be wirelessly charged and the data will be wirelessly and automatically sent to the server.

The main features and tasks are: Pressure Sensing (embedded within the shoe, amplification with op-amp, data acquisition), Charging/Battery (wireless power transfer to the parallax propeller through mutual inductance of inductor coils, power consumption of devices), and Data Capture/Transfer (storing data on micro SD, wireless transfer through use of Bluetooth, secure transfer), Server/User Interface (producing graphs from server data, patient search). Due to the scope of our project and the limited time and resources, the Raspberry Pi is the chosen device to handle a data relay between the shoe and the internet; using it allows our time to be invested less in low level tasks and more in features that directly address the societal problem at hand.

Pratik handles the design of the op-amp circuit to increase the signal detected by the pressure sensor so the propeller can more easily read and more precisely pinpoint the force applied to the sensor. Pratik also designed the AC/DC converter which will supply current to the devices in the docking station to allow the team's other features to receive power. Along with the op-amp circuit, Pratik is working on the hardware part of connected the Bluetooth to raspberry pi.

Sam's portion of the project involves power consumption of the parallax propeller, propeller to raspberry pi connection, and pressure sensors. The amount of current supplied by the AC/DC converter is used to calculate current consumption of each part of the project and then adjust accordingly. Sam also focuses on the specific type of lithium ion battery used to allows for multiple recharges when current is supplied to the battery. Sam is also responsible for using inductor coils to wirelessly supply current to the Propeller.

Kyle's main role is securing the data as it is transferred to and from each device. He is implementing data encryption and integrating secure protocols for transferring data between devices. He also works on implementing the connections between devices, the MC to PI to Server connection. In the Spring Kyle will work to improve the speed of the connections between the devices by increasing code efficiency. He will also work with Khoi to implement a feedback analysis feature which will analyze the data as it is collected and give the user more direct feedback.

Khoi's specific technical tasks in this project is to work with Microcontroller(MC) and User Interface. In Fall 2016, Khoi focused on MC and implemented code for data transfer process from MC-Pi-Server. Khoi also built a SSH server and developed prototype Database to keep track with patient's data. Khoi also created a Graph visualization program to generate graphs from raw data. In Spring 2017, Khoi will work with Kyle to add data feedback analysis feature and implement Bluetooth communication between MC-Pi while continuing polish his database and graph software.

I. INTRODUCTION

The societal problem that our team has decided to tackle is prevention of amputation and cost reduction of diabetic foot complications. In our world today, there are millions of people just in the United States diagnosed with Diabetes and a large amount that are undiagnosed. The source of issues for diabetics with foot problems begins with the formation of foot ulcers. Diabetics have an increased chance of growing foot ulcers compared to the average person due to their high blood sugar. Because of their high blood sugar, diabetics are more prone to develop neuropathy, which is the loss of feeling in a part of the body, in this case the foot. This growth occurs over a long amount of time and when untreated, can lead to loss of feeling in the whole leg, amputation and in some extreme cases, death. These problems can be avoided if the patient goes to their physician before issues become irreversible. Our goal here is to fabricate diabetic shoes with instrumentation to allow the caring physician to monitor the patient's weight distribution of their foot in the shoe throughout the day. This also reduces the amount of costly physician visits for the patient without too much inconvenience on their part. The first thing to consider in our design is other similar products that are already out in the market.

Currently in the market, there are shoes already designed to minimize the risk of skin breakdown for diabetic patients. Those shoes are wider and deeper than average to make sure that the patient's feet can comfortably move without scrubbing against the shoe's interior. Though, these designs still have many limitations.

First, when designing a diabetic shoe, the general goal is to make it wider and deeper to offload the pressures placed on the foot. However, not all the diabetic patients' feet shapes are the same. The current diabetic shoe cannot tell exactly which part of the plantar of the food bears the most pressure; thus, it cannot help to offload and balance out the pressure efficiently. Therefore, it is extremely important to get the shoe perfectly fitted due to this reason. If the pressure is not distributed correctly, or in other words, if the shoe cannot fit correctly, the ulcers cannot be treated properly.

Another problem of current diabetic shoes is it cannot detect infections from foot ulcers. Controlling infection is one of the most efficient treatment in foot ulcers. Once the infection is detected, an appropriate treatment will be prescribed in a timelier fashion that can reduce the risk of amputation as well as driving down healthcare costs.

Another limitation of the current diabetic shoes is it does not keep track of necessary information of patients' daily usage. The daily data collection such as the pressure data, daily steps count etc. is very useful and critical for doctor to analyze for further treatments.

One minor problem of the current diabetic shoe is patient compliance. Even though the patients are required to use diabetic shoes by the doctor, some patients don't wear it as needed. Therefore, it is quite important to determine if the shoe is being used or not. The benefits of it is not only to compel the patients to use the shoes for preventing worsening conditions but also to reduce healthcare costs.

There are a lot of problems our group is looking to solve for diabetics with foot problems and cost limitations. To implement all the features of our design, the project will need to be split up into varying tasks to be completed by May.

The idea of work breakdown structure (WBS) is to divide the project into manageable pieces of work to facilitate planning and control of cost, schedule and technical content. [1]. From the design idea document, the feature set will be broken down into subtasks which are assigned to specific members in the team. Each subtask will then be divided into work packages - the smallest blocks in the WBS that include the estimated resources and time needed to complete it with confidence [2]. In our group, there are 2 CPE and 2 EEE majors, so tasks will be divided based on concentration.

For pressure sensing feature, it will be divided into smaller subtasks: sensor circuit task, microcontroller task, and software task which is then further divided into work packages. These work packages will be described more in detail in assigned engineer section. This task will be split between all 4 members due to the involvement of software and hardware components.

For docking station feature, the task will be divided into battery charger task, and wireless communication task. Again, our 2 EEE's members Samuel and Pratik will be mostly working on this.

For data handling feature, the task will be divided into three smaller subtasks: microcontroller storage management, data transfer and data formatting. Since this feature involves more into software, each work package divided from subtasks will be assigned to 2 CPE's members Khoi and Kyle.

For user interface, the task will be divided into four main subtasks: pressure graph, database, application, and web page. Again, 2 CPE's members Khoi and Kyle will be in charged of this feature and they will work together in each specific work packages.

Based on the experience of working with sensors module, microcontroller and power, 2 EEE's members Pratik and Samuel will mainly work on the pressure sensing and docking feature while the 2 CPE's members Khoi and Kyle will be working on the data handling and user interface feature and any related software parts. However, members in the team can work interchangeably if needed to optimize and finish the workloads efficiently.

The work breakdown structure of the entire project will be visualized by the chart under work breakdown structure chart section. Before dates can be set to each task, a project budget estimation must be made to verify that the cost of the project doesn't surpass the amount that the group is willing to spend. Any areas that are considered to be too expensive will either be taken out of the project or changed with another feature. The table for the Product Budget Estimation is located under the Work Breakdown structure heading. Our project is sponsored by a business named Sunrise Shoes, but our funding is from internal team members. Once the budget of the project was agreed on, the project was split into smaller tasks. The assignment of team members to each of these tasks is based on their hardware/software experience. With the project then split up into different level subtasks and team members assigned to these tasks, a timeline to scope out the entire project can be made.

The project Timeline incorporates everything involved in the project from the start to final completion in May. The team leader role rotates between each team member once throughout the course of the project is marked on the days which rotation occurs. During the process of constructing the project timeline, dates for starting and ending tasks are also set. The tasks and subtasks assigned to team members in the outline of the work breakdown structure will be expected to be completed within their timeframe.

Throughout the course of the project, there are 12 milestones which represent intermediary objectives for the final project deployment. Some of these milestones are the completion of tasks and others are memorable moments in the project for the team members. The chart of milestones with dates is located under the project timeline. This project is one long journey toward our team's graduation which is the final milestone. The project began with the formation of the team which is the first milestone. There are also a number of subtasks involved in the project

Also incorporated into the project timeline is a description of each task, the resources needed to complete each task and the criteria needed to consider a task complete. The resources necessary for each task includes the number of hours needed, cost and devices/software to be implemented. For this, a Gantt chart is used to display the amount of time needed to complete each task and how time each subtask needs. During the start of our project, build time was very slow to compensate for starting senior design and focusing on how we wanted to implement our ideas into the project to construct a working product. We placed more tasks to be started and completed towards the end of the end of the first semester and beginning of second semester to allow for any time needed to troubleshoot before the final product demonstration in May.

Although our project timeline does a good job of assigning dates for each task to be completed, it doesn't account for possible failures that could occur and possible mitigation plans for failures. Our risk assessment judges which tasks have the highest chance of failure and how much that failure will impact the project. Based on the risk assessment, mitigation plans are devised to allow completion of the task in the allotted time.

Every task and sub task identified in the work breakdown structure has its set of associated risks. Risk refers to the uncertainty that can cause undesirable outcome on the project. When lacking information, or having ambiguity within the project, the likelihood of a problem occurring increases, and thus risk increases. For each risk event, two calculations are required: risk likelihood and risk criticality. The risk likelihood will determine the probability of failure (P) while the risk criticality will determine the impact of failure (I). Thus, the simplest way to model and compute the risk(R) is defining the risk mathematically as a function of $R = I * P$ [3].

If risks are not forecasted and evaluated at early stage of the project, the threat of risks increases as the project moves forward. For every project, a good engineer should know how to control, reduce, or eliminate risks. To do that, risk assessment and management is applied to most of the project at an early stage. This is a continuous process where we repeatedly evaluate risk and determine a way to mitigate the impact of that risk.

For our project, a perceived critical path will be identified to reduce the risk; at the same time, a possible mitigation plan will be proposed in case the problem associated with the risk occurs. There are total five critical paths are identified: Real Time Data Capture to MC Storage path, Data Handling Unit (DHU) to Server Upload path, microSD to the DHU path, AC/DC converter to the rechargeable battery path, and DB software and visual graph software to the webpage path. The risk assessment for each path are fully described in risk assessment section. In addition to the critical paths, the risk of each subtask in the Work Breakdown Structure will also be evaluated by its probability and its impact together with its risk mitigation plan. Planning for failure and mitigating it will help to construct a working product.

Once the various risks were assessed in the many tasks of the project, a budget had to be established, and we must find funding for the project. Sunrise shoes, our sponsor, does not provide monetary support, however, most of the parts bought were relatively inexpensive. The source of our funding came primarily from ourselves and family. We tried to use freeware and virtual ware where possible.

When everything is working correctly, the patient will wear the shoes throughout their day applying pressure onto the sensors. The sensors will capture the amount of pressure applied several times per second at different locations on the foot. When the shoes are placed on the docking station each night, they will recharge and send the pressure data to a server that the caring physician can view and interpret the data. Based on the data, the physician can prevent patients from getting ulcers or reduce pressure on certain parts of the foot that can cause permanent damage. The goal we are aiming for here is to reduce the number of physician visits for the patient to save money. By reducing cost, it will reduce the number of people with foot ulcers that deny any foot problems.

To achieve our goal of a working prototype and final product, many tests must be run to confirm that the product is working properly. To check whether the pressure sensors are sensing an accurate amount of weight is being applied, the sensors are mounted to the sole of a shoe without the excess material. The sole is placed on a wood frame to keep it in place while adjustable metal arms hang over it. Hanging from the adjustable arm is a spring with a loaded weight. The weight is set onto a sensor and the value is recorded on the microcontroller. If the pressure read by the microcontroller is inaccurate, the code is modified so that the formula used in the digital to analog conversion gives a more accurate value. For the prototype, our goal is accomplished when our team has implemented everything on our feature set. In our Pert chart beneath the risk assessment, we have all our perceived pathways along with each individual task involved in the project. For the prototype, data and signal flow through these pathways must be smooth and unsevered. We must be able to move between each pathway and show that it is working starting with pressure on the sensors and ending with the data visualization on the webpage.

In this month of November, our team began putting together each of the tasks to build our prototype. This included adding code to the microcontroller so that it moves data on the MC and sends it to the microSD card and then to the DHU. From there it goes to the server where it is visualized. An AC/DC converter supplies the devices in our docking station through wall connection to wall socket with 5V which is the minimum voltage needed to power the DHU and lithium battery charger. These features are what we are implementing for the prototype, but we plan on improving on the product for the final demonstration in May.

In the following semester, we plan on adding more features to our product to reduce the amount of user interaction and gather more useful data for the caring physician. For the working prototype, we used a wired data transfer between the microcontroller and the DHU which must be plugged in while the shoes are on the docking station. In the second semester, we plan on using Bluetooth connection so that this data transfer is wireless and the user involvement isn’t required to start the data transfer. Once the Bluetooth module in the shoes is within proximity of the module in the docking station, the data transfer will begin. Another feature to be added next semester is wireless charging. For the working prototype, wired charging is used to give the team more time to focus on other tasks that had a higher impact to the project. With wired charging, the shoes should be plugged into the lithium battery charger on the docking station to charge. With wireless charging included in the product, it further reduces involvement from the user. Adding these extra features that decrease user involvement throughout the use of the shoes increases the likelihood that they will continue to use the shoes until they show symptoms of relief. For the user to be able to wear the shoes while requiring little interaction on their part almost makes wearing them an unnoticeable change in their daily routine. To implement wireless charging, the AC/DC converter will send DC current to a delivery coil in the docking station. There will be capacitors between the coils and input/output to keep the signal DC and limit the current along with resistors. There will also be a receiver coil in the shoes that will receive the current from the delivery coil when the coils are close enough to each other which would be when the shoes are on the docking station.

The raspberry pi is used as our data handling unit which organizes the data it collects from the microSD and sends it to our server. The raspberry pi takes up about 400mA during the data transfer which is within the limit of our supply. The raspberry pi must be reset in order for data to be transferred by the Bluetooth so a hard switch was used here. When the shoe is placed on the docking station, the switch is closed and power goes to the raspberry pi. When the shoe is taken off, the switch is open and the pi remains off. The receiving Bluetooth module is built into the pi so this is one of the main reasons the pi must be reset for new data to be received.

Also in the docking station is the wireless charger. The wireless charger consists of 2 inductor coils; receiving coil is 14μH and delivery coil is 3.7μH. The wireless charger charges the battery at a rate of about 70mA per hour. The consumption of our battery is about 200mAh with Bluetooth running and the system runs for only the first 2 hours after pressure is sensed in the shoe so the expected consumption of the battery throughout the day would be 400mA. Our battery is rated at 2500mAh so we don't expect any drop-in voltage or current over time during the first two hours. Our results were verified during testing. At 400mA of consumption, it would take about 6 hours for the battery to recharge to full. The range of the wireless charger is about 1 inch so some supports and an outline were included on the docking station design to ensure the coils are aligned well enough to allow full current transfer from delivery to receiver coil. The schematic for the docking station layout is in the figure 7 in the design idea section.

The last feature we are implementing in the deployable prototype is tilt sensors in the inside linings of the shoe. The purpose of the tilt sensors is to measure, in this case, the vertical displacement of the two sides of the foot in the shoe. Going back to our original goal, we want to prevent neuropathy by finding the areas of the foot that are under the most pressure. Using tilt sensors, we will be able to see if the user's foot tilts to one side while they walk so a physician can then begin treatment by offloading pressure on a certain side of the foot.

This document goes on further to describe the societal problem, how we are solving it and the process we went through to produce a working prototype and later a deployable prototype. Included is data on our simulations and test experiments with our implemented hardware and software components.

II. SOCIETAL PROBLEM

Diabetic patients develop neuropathy over a long period of time by not monitoring their blood sugar levels as well as they should. The increased glucose levels harden patients' arteries, restricting blood flow and damaging nerves. When this complication with diabetes occurs, patients lose the feeling in their feet, meaning they can no longer feel any cuts or bruises on their foot. With the restricted blood flow, patients have a reduced ability to heal the wound, increasing the chance the wound might be infected. Injuries like can go unchecked for long periods of time which increases the likelihood of an infection occurring even higher. Patients with these complications tax the healthcare system disproportionately than those who do not have these complications. These costs are accrued through frequent doctor visits which are the result of amputations and foot ulcers (unchecked wounds). There are some treatment methods to for these types of complications. One of them is a shoe that offloads pressure from forefoot and heel to the arch of the foot. This allows for increased blood flow which helps the patient fight off infections.

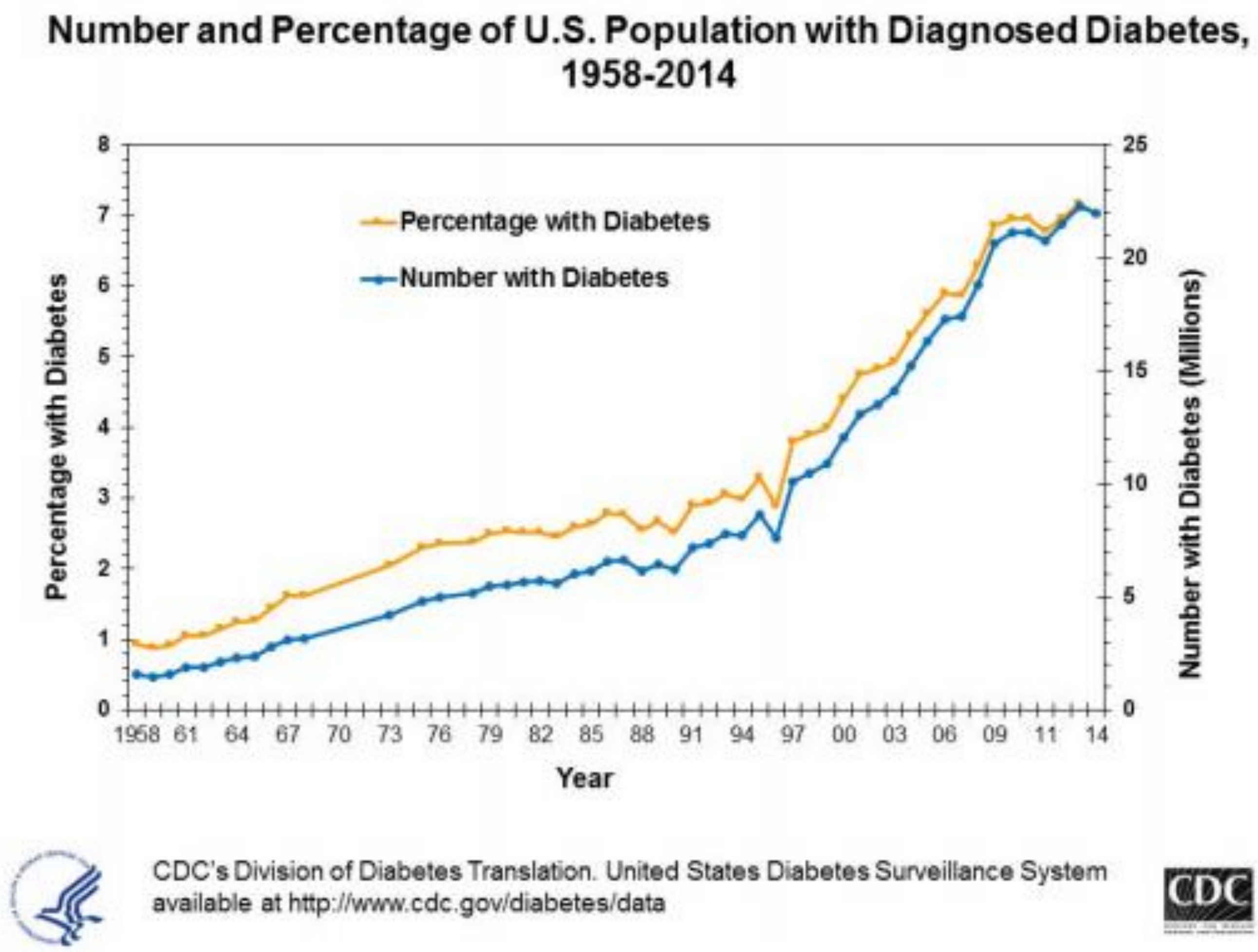

Fig. 1 A graph showing the increasing number of diabetes patients over the decades [4]

Other treatments include cleaning the wound regularly, but this leads to the problem of increased medical costs. Current treatment methods require patients to go relatively prolonged periods without a checkup which can cause their condition to worsen.

A. The High Medical Costs

There is a prevalence diabetes in the United States. Figure 1 shows the trend of diabetes in the united states for the past few decades which seems to be going up. [4]

With the number diabetes patients rising every year, that means the cost of treating it will rise every year as well. Studies show that patients with foot ulcers costs double the amount of a patients without foot ulcers, even though patients with foot ulcers tend to only be 8% of the diabetes population. The reason for his is because patients with diabetic foot ulcers have 138.2% more days hospitalized, 85.4% more days of home health care, 40.6% more emergency doctor visits, and 35.1% more outpatient physician office visits [5]. Patients with foot ulcers are having to constantly be hospitalized and cause more payments out of private insurance programs and Medicare, since the majority of the population that experience foot ulcers are senior citizens, Medicare incurred a cost of $11,710 per patient, and private insurers a cost of $15,890 per patient [6]. Figure 2 shows a breakdown of how each dollar is spent on a patient with regards to Medicare and private insurance. [7] The total cost in this figure are different due to them being from a different study, however the figures are close.

Because foot ulcers are the leading cause of amputations for diabetic patients, the cost of care that these patients need as a result of any hospitalization is much higher. Amputations cost anywhere from $7000 to $10000. The aftercare adds on another $60000 as another cost to the patient and insurance.[8]

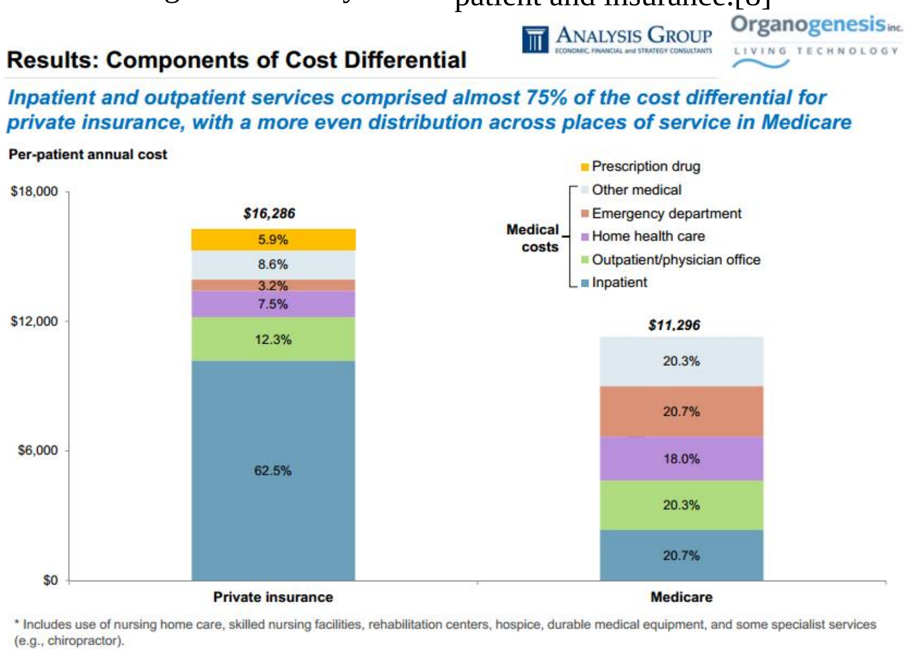

Fig. 2 Components of Cost Differential [11]

In fact, diabetic foot ulcer amputation costs comprise a majority of the total costs for amputations. Of the 21,305 major amputations cases studied of a 6 year period showed that 82.9% of these major amputation cases were associated diabetes. The numbers look similar for minor amputations as well. Of the 106,592 cases studied over the same period, 96.1% of the minor amputations performed for were diabetes patients. As mentioned previously from another study, the cost of each diabetic foot ulcer hospital admission is $9,937 with an average yearly cost of $1.38 billion. Those foot ulcers that do not result from diabetes costs approximately the same for a per admission price of $9,794, however the average yearly cost is only $.13 billion dollars. [9] It is very important to know that about 30% of all costs incurred by diabetic foot ulcers were due to infections which are preventable. [10] Figure 3 is a chart that shows the occurrence of hospital admission due to foot ulcers. It is broken down into diabetics and nondiabetics, and further into the categories in which caused the hospitalization to occur. [11]

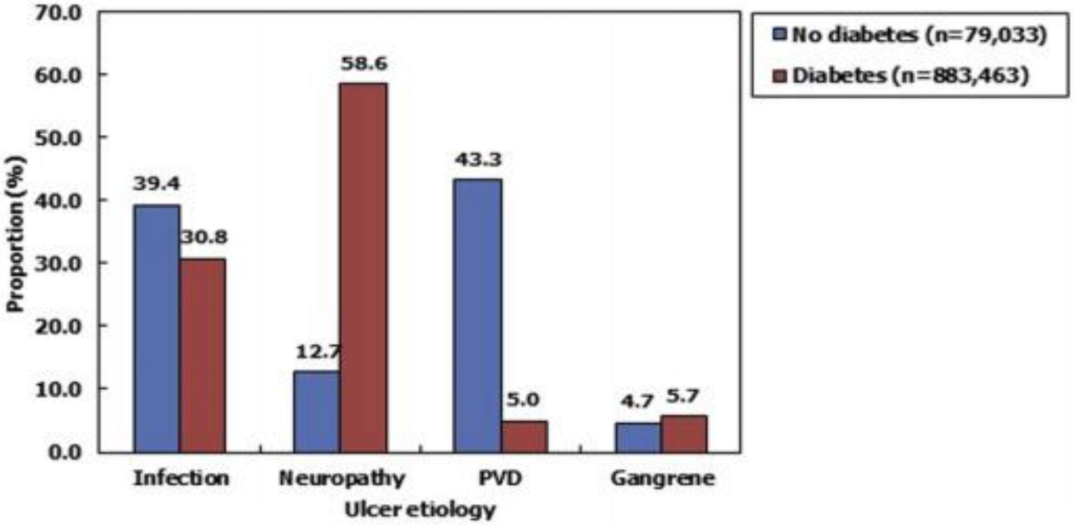

Fig. 3 Hospital Admission [13]

What's more is that the patients experience losses as well. When studied, foot ulcer patients experienced a loss of $3,259 in wages due to the increased hospital visits [12]. Even when the patient goes to see a doctor before the ulcer requires amputation, the average cost of treatment can be up to $10,000 if it's not infected. In the extreme cases where amputation is necessary, 50% of patients develop an ulcer in another part of the body. [13]

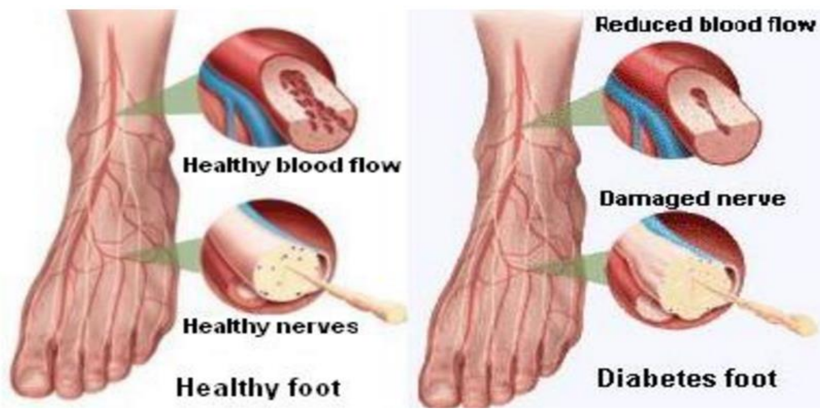

Fig. 4 A healthy foot vs Diabetic foot [15]

B. The Ease of Contraction

There are many different factors that go into the what causes a foot ulcer. One of the main aspects is that diabetic patients have higher than normal levels of sugar in their blood. These elevated levels of glucose in the blood over time work to harden the arteries and constrict the blood vessels.[14] Figure 4 shows the difference between a healthy blood vessel and a constricted blood vessel. [15]

These constricted blood vessels lead to a multitude of problems for diabetes patients, one of the major ones being poor blood circulation. With poor blood circulation, patients have an increased risk for peripheral arterial disease (PAD). This condition restricts blood flow to the legs which deprives the legs of essential blood and oxygen to help heal any cuts or infections. [16]

In addition to PAD, patients can also develop neuropathy, the condition that exacerbates the effects of foot ulcers. Neuropathy causes patients to lose feeling in their legs, this makes it so that any new cuts, bumps, or bruises that go unattended increases the likelihood of infection. [17]

A diabetic foot ulcer does not develop

overnight. All of these symptoms combined are what makes diabetic foot ulcers a stealthy, costly, and deadly diseases . To begin, elevated blood sugar levels damage the body over time and constrict blood flow. The constricted blood flow than can cause PAD which can reduce the legs ability to heal itself. The constricted blood flow also causes neuropathy which makes the patient losing the ability to feel their legs. Also diabetics have a slower immune system which makes it slower at fighting off infections.

The disease is stealthy due to the fact that a diabetic would not even feel a cut on their leg due to the the onset of PAD and neuropathy caused by prolonged blood sugar levels. The cut goes unattended and an open wound develops which can be considered the diabetic foot ulcer. If this ulcer goes even further unattended due to the inability to feel the pain that it is causing, it will develop into an infected ulcer, this is shown in figure 5. [18]

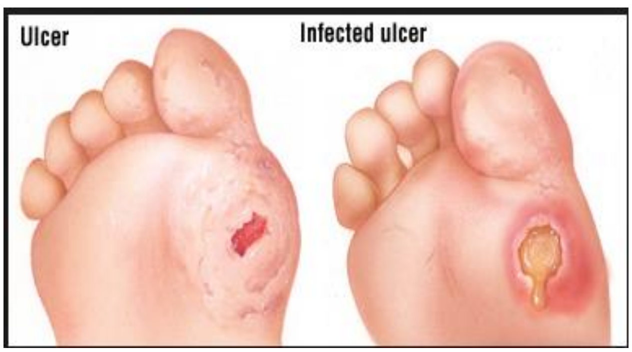

Fig. 5 Infect Ulcer foot [19]

It is costly because diabetic foot ulcer patients healthcare costs are double than those patients who do not develop the foot ulcers. And it is deadly because the mortality rate for this disease is 46% with or without amputations. [19]

The only way to ensure that a patient doesn't grow a foot ulcer is to make regular trips to a doctor which most are not willing to do. Diabetes patients are more prone to foot ulcers because they have low blood sugar. When patients bodies don't make enough insulin to break down sugars for the body, they are diagnosed with Diabetes. Anyone with diabetes is susceptible to foot ulcers which is why it is so important to have regular visits with a doctor which not all patients do. Foot ulcers first begin to form because of lack of blood circulation in the foot which causes a lack of feeling. On top of low blood circulation, increased pressure on the foot increased the risk of ulcers. Patients who have diabetes for an extended number of years usually develop neuropathy, the inability to feel pain because of nerve damage. This nerve damage occurs because of increased glucose levels over the increased length of diabetes. For those with nerve damage, a small cut or a wound which if left untreated can lead to infection which also leads to the growth of ulcers. One of the main side effects of a foot ulcer is pain when walking, but those with neuropathy continue walking with their ulcer not knowing they have one until it reaches a critical condition.

Patients also unintentionally place themselves at further risks by not revealing all their information to their health care physician. The Wall Street Journal found that 28% of patients lie or don't reveal everything to their doctor about their health.[20]

C. Lack of Good Treatment

There are several forms of treatment available for foot ulcers today. One of them is debridement, which is the process of removing all necrotic tissue, periwound callus, and foreign bodies down to viable tissue. The main objective of debridement is to reduce the risk of infection, as mentioned, infections account for 30% of all medical costs associated with diabetes foot ulcers.[21] The reason why debridement is considered an important treatment because an open wound or ulcer can not be properly evaluated until the dead or necrotic tissue is removed. Not only that, the ulcer tends to heal faster once the dead tissue is removed. Debridement is also used to treat abscesses-a collection of pus that has built up within the tissue; if left untreated, an abscess can develop into a serious infection that may invade the bloodstream leading to amputation and even death.

Another treatment that can address the foot ulcers problem is offloading. When you walk, the amount of force generated by your weight will be put on your feet and ankles. This amount of pressure can inhibit blood flow preventing the development of new blood vessels as well as tissue on the bottom of your foot. If diabetic patients don't keep offload, it can make wounds slower to heal and even worse it may cause infection that can lead to amputation. Thus, for those diabetic patients with foot ulcers, it is best for them to keep all their weight off by using a wheelchair, clutches and reduce walking with their bare feet. However, in some circumstances, some diabetic patients have to walk as their job requires; custom insoles, pads, shoes are designed to equalize the distribution of plantar pressure throughout the sole of the foot. Those designs provide extra support to the ankle, extra depth allowing more room for foot orthoses. Several studies showed that the total contact cast (TCC) is the most effective treatment used for offloading to prevent foot ulcers.[22] This device consists of a fiberglass that fits around your leg and foot very closely and the whole foot is wrapped in bandages prior to the application of the cast. A soft layer of foam is placed on the ulcer which makes a room so that no pressure is put on the wounds; therefore, it can heal more quickly. However, practical offloading method still remains as a critical challenge in foot ulcers treatment due to improper cast application or TCC does not allow patients to access the wound on a daily basis.

Infection control is also an important treatment to address foot ulcers. A diabetic patient with any foot infection if left untreated can potentially lead to serious problem such as amputation. Types of infection can range from mild to moderate including cellulitis, myositis, abscesses, necrotizing fasciitis, septic arthritis, tendinitis, and osteomyelitis. [23] Regardless the serious complication of the infections, they all are required to receive an immediate and proper treatment. Moreover, not all of infections should be treated the same way. For some mild cases, a patient can prevent the infection of foot ulcer by foot bathing, disinfecting the skin around an ulcer, or keeping the ulcer dry with frequent dressing changes. However, in most cases, especially for serious infections, a tissue surrounding the ulcer may need to be sent to the lab to determine which antibiotic will help by the doctor; or in extreme cases, the doctor will have to order an X-ray to look for signs of bone infection.

D. Problems with the Treatment

The current solution to help diabetic patients avoid foot ulcers, is a custom fitted pedorthic shoe designed to prevent ulcers and facilitate the healing process. However, some issues can arise even with the diabetic shoe.

Currently, for those with high risk diabetic conditions, the standard monitoring rate is in 3 month intervals.[24] Because of the nature of diabetic/neuropathic conditions it can be difficult for a patient to feel, or notice if any problems are occurring. Commonly ulcers, and other detrimental conditions can rapidly and unexpectedly onset. More frequent monitoring can help catch complications early so that corrections to treatment can be made. There is a limit to how frequent a physician or specialist can monitor the condition of the patient. The cost of frequent visits can be quite high, not to mention the inconvenience to the patient as well. [25] A way to remotely monitor and check up without inconveniencing the patient would be desirable. This would lower the chances of painful, time consuming, and costly complications and reduce the need for physician care.

An improperly fitting shoe can be disastrous, as it could provide the correct conditions for an ulcer to develop. It's extremely important to get the shoe correctly fitted. Specialists need to locate where the ulcer is located on the foot, then a custom shoe must be made as to alleviate the pressure from that area of the foot to prevent further damage. Current solutions involve properly measuring and fitting a shoe to the patient, but don't necessarily monitor the fit of the shoe during use or over time. Following visits and checkups to monitor the patient's foot are the main way to ensure a quality fit.[26]

Even if the shoe works as it should and there are no complications caused by the shoe, there still remains the issue of patient compliance. About 28% of patients lie or don't reveal everything to their doctor about their health.[27] Patients who are given the shoe may end up not using it or may use it less frequently than needed. In a similar product, compression stockings used for medical purposes, total patient compliance was 21%, and partial compliance was 16%.[28] This is an issue that can make the foot and ulcer conditions worsen even if the shoe would otherwise properly heal the patient. A patient might then lie to the specialist which can cause confusion and slow down the healing process and decrease the quality of health care.

Even worse, if a treatment requires more self-administered treatment, compliance rate drops.[29] This can mean that if the shoe is improved, but the difficulty of using the shoe increases, then patient compliance decreases. This means that any improvements or added features to the shoe must effect the patient minimally to retain the highest number of compliant users. Ideally any improvements to the shoe would cause no apparent differences to the user of the shoe.

III. DESIGN IDEA CONTRACT

In the problem statement, two treatment methods were considered, keeping the wound clean, and offloading pressure from one area of a foot to another. Our team considers offloading a better solution because it tries to address the main problem with patients with these types of complications, which is the restriction of blood flow to the feet. Our sponsor, Sunrise Shoes, has developed a patented shoe, that helps address the problem of blood flow, but does not address the problem of patient compliance or patient monitoring. We will be trying design a solution for Sunrise Shoes that will help address the problem of patient monitoring without having an effect on patient compliance. By helping address the problem of patient monitoring we hope to address the underlying effect of this affliction, high medical costs. By increased monitoring we hope that doctor will be able to catch abnormalities faster while also being able to determine if the treatment is helping the patient.

To achieve our design goal, we will include technology so that it requires as little user interaction as possible. Wireless technology will be used to send the data collected from the sensors and send it to a Data Handling Unit(DHU) in the charging
dock. The DHU will then send the information to a server online where it can be accessed by Sunrise Shoes and physicians as needed. Wireless technology, such as Bluetooth, are small enough to be placed in the tongue of the shoe to allow for communication to a DHU. The DHU is too large and uses too much power to be placed in the shoe, so it’s placed in the loading dock. The pressure sensors will largely be located in the sole of the shoe to monitor the bottom of the foot because that is where ulcers and infection occur. Data from the shoe will be collected for the first two hours each day when pressure is sensed in the shoes. The shoes will begin collecting data again the next day once they are charged and pressure is felt in the shoe. Figure 6 shows a breakdown of different components involved in the design. [30]

A. Feature Set

1. *Pressure Sensing*

By using pressure sensors at multiple contact points in the shoe or pressure sensor pads, along with foot tilt information, data can be gathered to help assess the medical viability of the shoe.

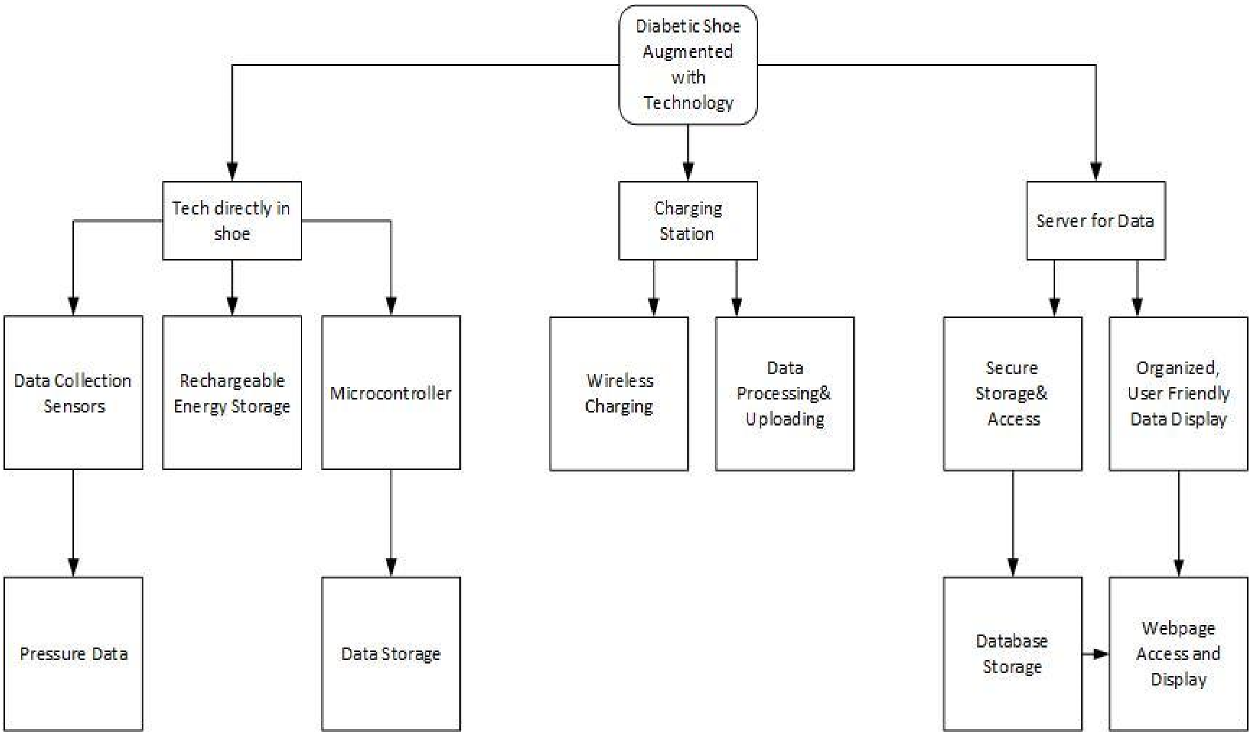

Fig. 6 Design Overview [30]

*i. **Hardware Needed to Implement***

a. Pressure sensors - the pressure sensors chosen will be determined by the resolution of data needed by the sponsor. Our sponsor specified that having a sensing diameter of about a quarter is acceptable. The sensors we choose will try to meet this criteria of capturing force distributed over an inch or less. There wille pressure sensors such as the Flexiforce A301 that may require additional circuitry to operate, these hardware components also need to be considered

b. Tilt Sensors - this sensor will be used to determine the foot's orientation with respect to the horizontal.This will help determine a rough estimation of which point in the gait cycle the corresponding pressure data was acquired from. Just like the pressure sensors, tilt sensors may also have associated circuitry that may need to be considered.

c. Microcontroller (MC)- a MC will be needed to handle the incoming analog data from the sensors. Because of the space limitations of the shoe, the MC needs to be able to handle multiple processes at the same time. The processes relating to pressures sensing that our MC needs to be able to handle are collecting the data, and saving it onto a memory unit inside the shoe without interruption.

ii. Software Needed to Implement

a. MC Integrated Design Environment (IDE) - any MC chosen will have its own IDE to work with as well its own programing language.

2. Docking Station

A docking station will be a 3-D printed box that the shoes will be placed on to help provide power for the various electronics in the shoes and to help reduce patient interaction in terms of replacing batteries or data management.

One of the main aspects of the charging station is wireless charging. Again a key focus to this issue is giving the patient as little to do as possible so once the shoes are placed on the platform, they will begin charging wirelessly and begin transferring the data. The wireless charging will work through inductive power transfer [31]. The power source will be the AC signal coming from a wall outlet, from there the signal goes to a coil in the charging station with a transmitter circuit. There will be a coil in the shoe also so that when the charging station and shoe are close enough in contact, a magnetic field is formed which makes a current going through the coil of the shoe. The AC current that is formed in the shoe is then converted to DC to then power the shoe.

The charging station charges the shoes and also sends the data to the internet. For it to work correctly, it must firstly be able to charge the shoe. The DHU is also inside the charging station so it must be able to retrieve the information from the shoe. From there, the DHU will have to send the information via internet to a server. For the charging station to be completely implemented data must be extracted from the shoe and send to the server. Figure 7 shows how components are placed in a docking station. [32]

i. Hardware Needed to implement

a. Rechargeable Batteries

These batteries will need to be able to able provide enough voltage to drive the MC's and sensors in the shoe as well be able to be recharged by low currents like ones provided by wireless power. Even though these batteries will be placed in the shoes, they will be charged by the docking station which is carrying most of charging circuitry.

b. AC/DC Converter - a portion of the electronics in the docking station will use DC to operate. This includes the DHU and the wired charging module of the docking station. The wireless charging will use AC to operate.

c. DHU

The DHU will be in the docking station. A suitable device that allows for easy access to the internet and has multiple connectivity methods such as wireless needs to be used.

d. Charging Circuit/Receiver Coil

These refer the hardware required to send power wirelessly to the shoes. An inductor circuit in the station will resonate with the receiver coil in the shoe to transfer power to charge the batteries in the shoe.

ii. Software to implement

Python code for establishing a handshaking communication between Raspberry Pi and Propeller.

3. Data Handling

The data will be collected from the pressure sensors for the first two hours after foot pressure is sensed in the shoe and temporarily stored on the microcontroller's memory. A timer will be used on the microcontroller to limit data collection to two hours. The timer will be reset when the batteries in the shoe are fully recharged.

We estimate about 12-16 Kbps in collected data. The microcontroller has a limited data storage space and the available memory is volatile and would be lost if the device were to lose power. So the data will be transferred to a non-volatile storage device connected to the microcontroller, such as a microSD card.

When the shoe begins charging, the data will be transferred off of the microcontroller wirelessly to the DHU in the charging station. The DHU will transform and reformat the data to be more useful and easier to store and read on the database. A backup copy of all the data collected will be stored on the DHU to protect against accidental data loss.

The Data will then be sent to a server where it will be accessible to those who need it through a UI.

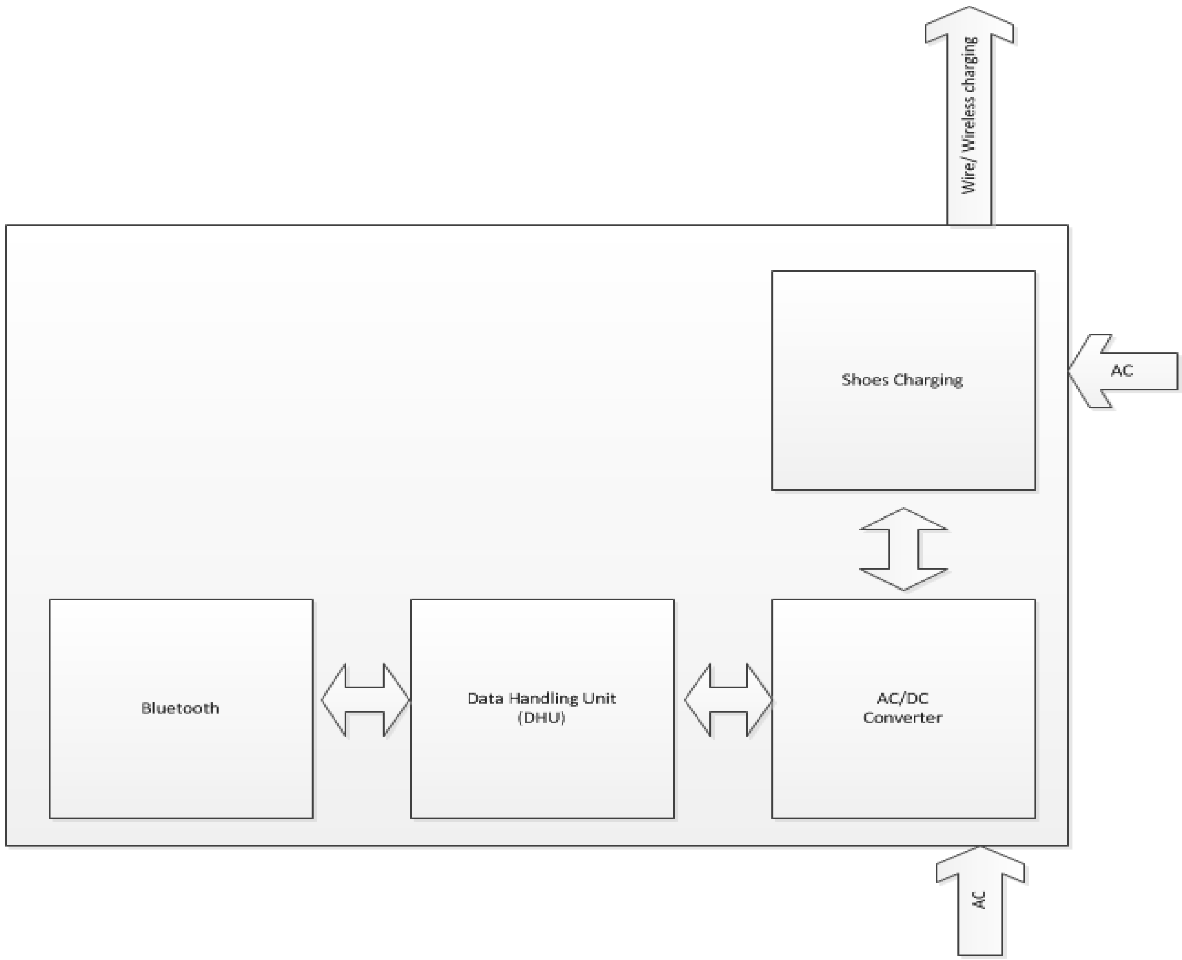

Fig. 7 Docking Station Diagram [32]

i. Hardware to implement:

a. Microcontroller Take in inputs from each pressure sensor, convert analog value to pascal and store data in memory device on microcontroller.

ii. Software to implement

Simple IDE software tool with C language.

4. User Interface

SSH Server is setup for the Pi to communicate. This SSH server allows files transferring between Pi(Linux based) and PC(Windows based) using scp (secure copy) command.

Database will be used to store the collected patient's data. SqlLite language is used.

An application will be what the user interacts with, and would call stored procedures in the SQL database to manipulate data for best ease of use, performance, and security. This application will be a friendly user interface environment could be written in C# for both end users (doctor, and developer) will be stored in our physical server. This software will be deployable into any other PC such as doctor's PC to help to organize data and keep track with all the patients' information. The doctor can easily search, edit, insert, delete or retrieve any essential information such as pressure mapping data, temperature data of the patients ect..

A graph visualization will transform collected data into graphs.

A simple diagram of Database feature is shown in Figure 8. [33]

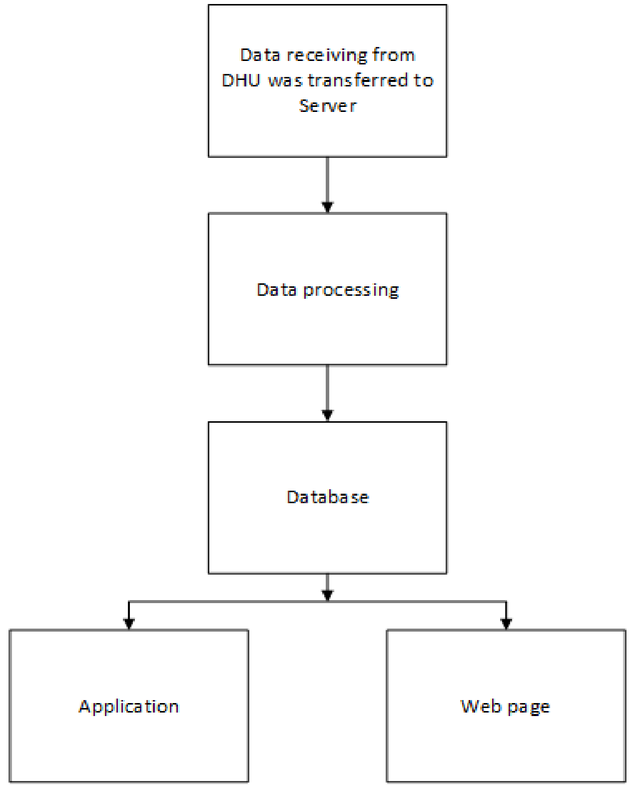

Fig. 8 User Interface Diagram [33]

B. Fall and Spring Expectations

There is no possible way that we will be able to fully implement the feature set for a deployable prototype in December, especially with features like wireless charging, and wireless communication with limited power. For the working prototype, the group feels that having the data path established with wired connections, the pressure sensors working, and having wired charging will be sufficient enough to show proof of concept for a working prototype.

For the spring semester, we will establish the data paths with wireless connections. We will add bluetooth to allow communication between the shoe and the DHU. We will implement the tilt sensors to show the orientation of the foot at moment of pressure data capture. We will also implement wireless charging to help with patient compliance rates.

IV. FUNDING

To meet the feature set of the project, we needed to buy parts such as microcontrollers, pressure sensors etc. For the first semester, the laboratory prototype was entirely self-funded by team members. A summary of the estimated costs, and source of funding is listed in Table I. [34] For the second semester, the deployable prototype was also entirely self-funded by team members.

TABLE I
Internal Funding Proposal

List of Parts	Qty.	Cost	Source
Microcontroller - Parallax Propeller	2	$80	Sam and Khoi
Pressure Sensors	2 pack	$80	Sam
Microcontroller - Feather 32ux	1	$26	Khoi
Feather Battery 2500mA	1	$14	Khoi
Data Handling Unit	1	$35	Kyle
MicroSD card	1	$5	Kyle
Transformer	1	$16	Sam
Rechargeable battery	1	$10	Sam
Receiver/delivery coil	1	$15	Sam
Fiber Wood	1	$4	Family
Wood, 2x4x6ft	4	Free	Family
Bolts	2	$1	Family
Screws	6	$3	Family
Drawer Slide	1 pack	$17	Family
Springs	2 packs	$11	Family
Nylon Rope	1	$5	Pratik
Single Pulley	2	$10	Pratik
Washers	1 pack	$1	Pratik
3D printed housing	2	Free	School
Total Cost		333	

V. PROJECT MILESTONES

Just like how every engineering project costs money and needs a source of funding, every project has milestones that signify the completion of major goals in the project where significant progress has been made. Our milestones chart in Figure 9 below summarizes the intermediary goals of the project. [35] Each milestone was chosen so that it signifies a major accomplishment in the life cycle of the project.

The first milestone is reached on 8/29/2016 when the team first forms and establishes a problem statement and begins to formulate a design idea and contract. The next milestone is when the design idea is approved. This signals the beginning of the build phase of the project, this will be reached on 10/15/2016.

In the build phase, one of the biggest milestones will be to complete work on pressure sensors so that real data can be captured for data transfer. This will be the third milestone. The work on the pressure sensors is crucial, not only is it the main indicator for determining the viability of the shoe, but also the foundation in which the rest of the project is built upon. The pressure sensors should be up and running by 10/31/2016.

The fourth milestone is getting ready the wired charging capabilities. For whatever reason, if the team fails to implement wireless charging, wired charging will need to be implemented to recharge the batteries in the shoes. This milestone should be reached by 11/18/2016.

The 5th and and 3rd milestones will be worked on simultaneously, the 5th milestone depends on the 3rd milestone. To make working on these possible at the same time, dummy variables to represent pressure will be used to help simulate data transfer from MC to server. The data transfer milestone is scheduled to be completed by 11/28/2016.

The next milestone, milestone 6, is the halfway point, this is when the working prototype will be ready for review. The prototype should be able to collect pressure data and send it to a server. The prototype is scheduled to be ready by 12/5/2016.

Reaching to spring will be another milestone for the project, milestone 7, which will be reached on 1/23/2017. Preparations for the finals tasks of the project will need to be made. The first two milestones thereafter are 8 and 9, which are working tilt sensor and working wireless charging respectively. Milestone 8, wireless charged will be implemented by 2/29/2017 and milestone 9, the gyroscope implementation will be reached on 3/1/2017. Milestone 10, the communication between MC - Pi need to be finished since it is crucial path in the risk assessment.

Next is milestone 11, mid-term review, the test result of integrated test.

Milestone 12 on April 17, is the last day for credit in term of hardware.

The milestone 13 on May 2, is achieved when the data from the server is visualized.

The final milestone is final demonstration of deployable prototype on May 12, 2017.

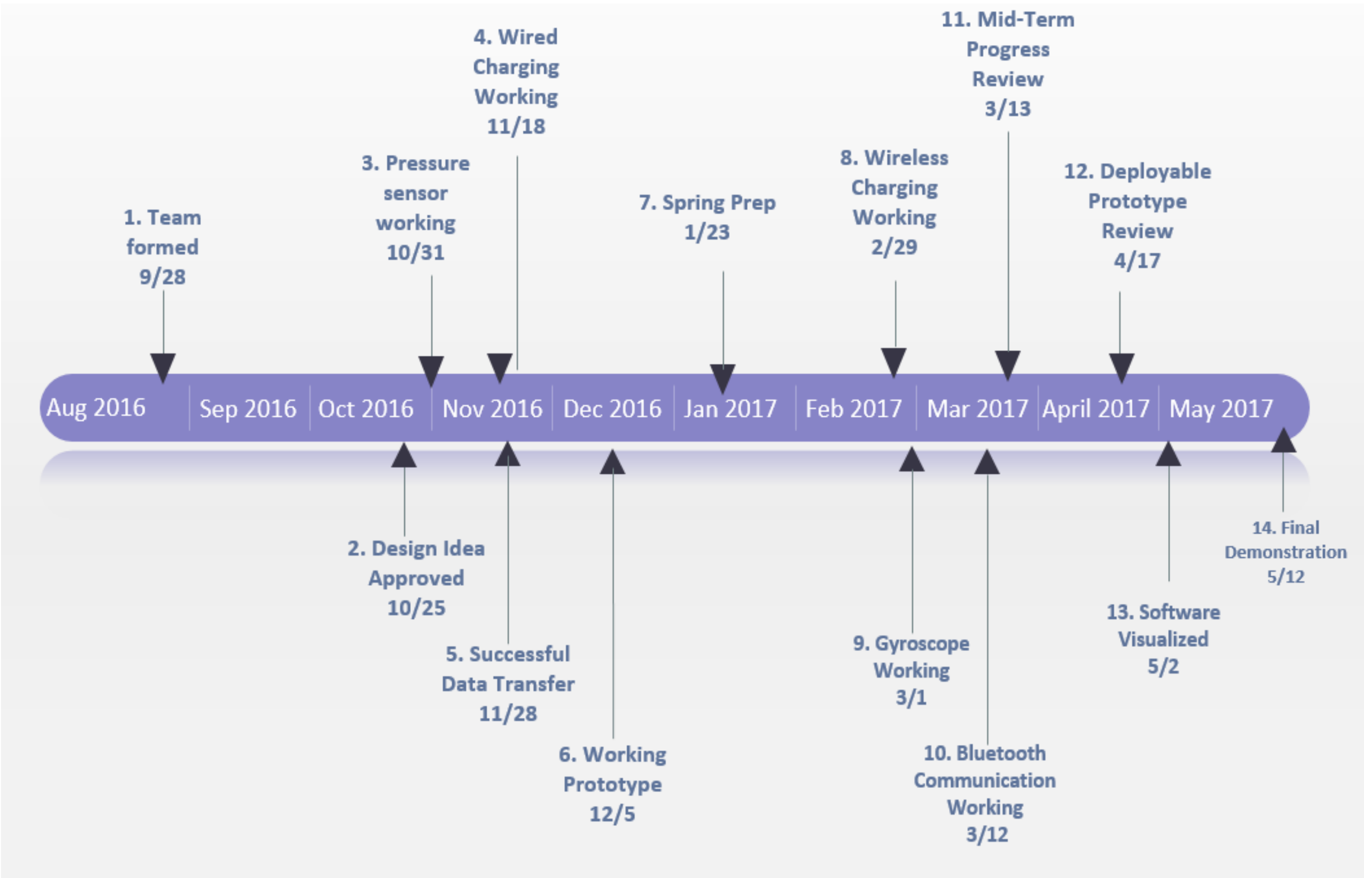

Fig. 9 Milestone Chart [35]

VI. WORK BREAKDOWN STRUCTURE

A. WBS Overview

While the milestone and project timeline provide an overall timeframe of how long the project will take; the goal of work breakdown structure (WBS) is to facilitate planning and control of cost. Thus, it makes the project manageable. The WBS splits the project into four main tasks: Pressure Sensing, Docking Station, Data Handling and User Interface.

These tasks are divided into lower level subtasks, seen in figure 10, to better gage how long the entire project will take and make it easier to track progress. [36] Our team has 2 EEE and 2 CPE majors. Based on background, the tasks and subtasks are distributed

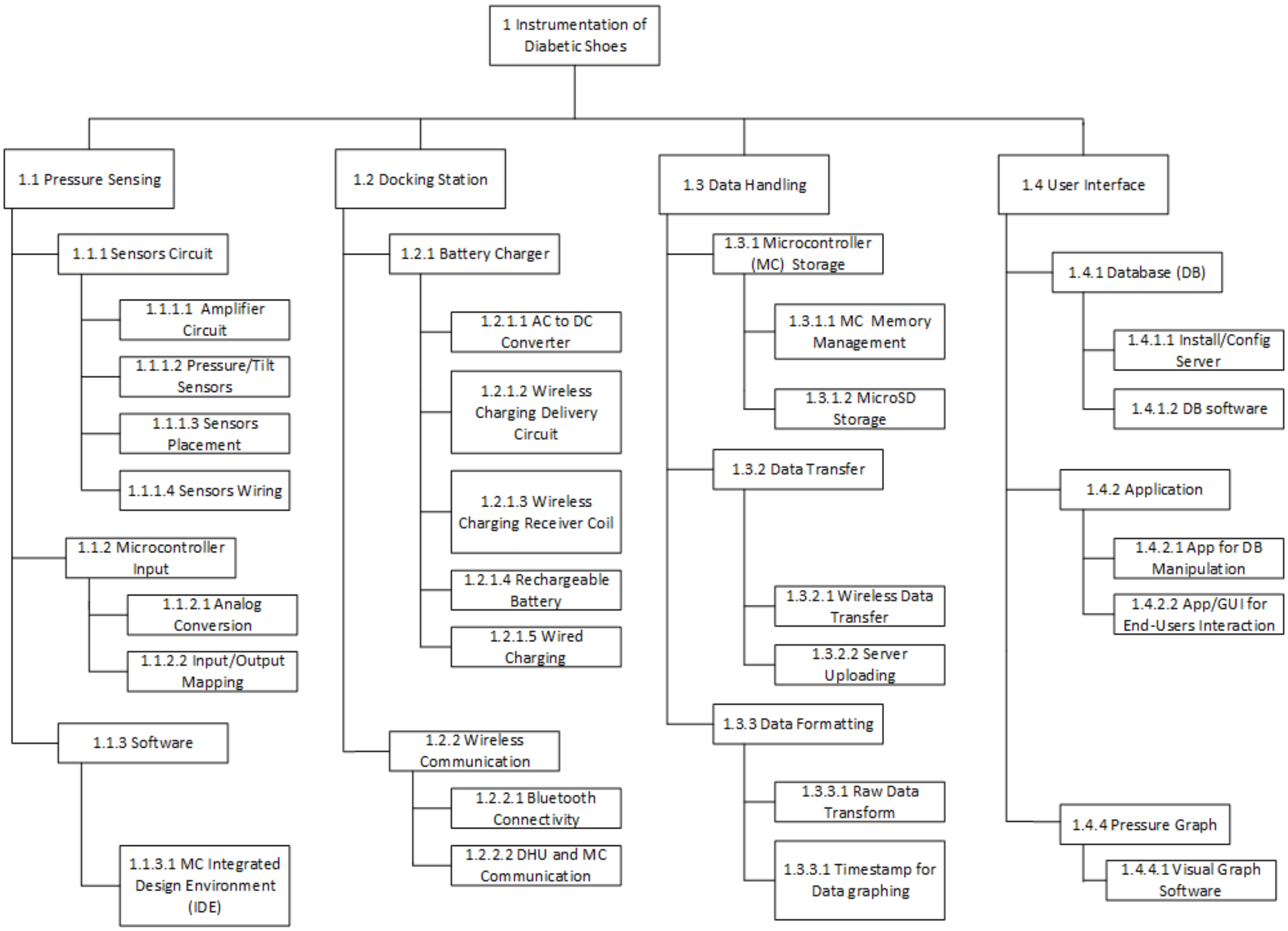

Fig. 10 Work Breakdown Structure [36]

based on hardware and software components. The CPE team will tackle the software components and the EEE team will tackle the hardware components. For the tasks that are both software and hardware, the tasks have been split up according to past experience and current specialization. The tasks that are both hardware and software, like the microcontroller and programming, are core components of the project. We will start by working closely together to ensure that the core components are working, like the pressure sensing, and will then divide and work on tasks that can be accomplished in parallel.

The task of pressure sensing will be split up into Microcontroller, Sensor circuit and software. Microcontroller will be split between all 4 members because it includes hardware and software aspects. For the microcontroller, the Digital signal from the sensors must be converted to analog then to Pascals for the user interface. The input/output mapping is another aspect in the microcontroller that will be handled by the 2 EEE's. The 2 CPE's will handle the software needed to program the microcontroller to perform as needed to capture real time data from the sensors. This will be done using a microcontroller Integrated Design Environment (IDE). The sensor circuit going from the pressure sensors to the microcontroller will be tasked to the 2 EEE's. The signal from the sensors will need to be amplified to increase precision of measurements. Tilt sensors will track angle of the foot while it's mid step. The data from all the sensors will be captured simultaneously. It will then need to be stored on the MC storage so that it can be data stamped and sent via Bluetooth to the Microcontroller unit (MCU) where it will upload the data to a server.

The task of the docking station will be split up into the Battery Charger task, and Wireless Communication task. For the battery charger the engineers tasked will have to set up a wired and wireless charging circuit. There will be two circuit associated with the wireless charging which are the receiver and sender circuits. Another component is picking the proper batteries that will deliver the needed power, small enough for the shoe, and be able to recharge. An AC/DC converter will be needed for the DHU, wireless communication device, and the wired charging circuit. For the wireless communication task, a Bluetooth communication unit needs to be set up to allow communication between the DHU and the MC to allow for data transfer between the two devices. The DHU will handle the organization of the data and will upload it to a server.

The task of Data Handling has be broken into: microcontroller storage management, data transfer, and data formatting. The engineers tasked with this must a larger storage than a stock microcontroller would allow to hold large amounts of data. The data must be pulled from the MC internal memory and moved on a MicroSD storage unit. A CPE must handle the interfacing between these two technologies. One the data has been stored, it must be formatted for time and date to allow easy understanding of the data. Once this information has been attached to the data point, it needs to be transferred from the MicroSD to the DHU on the docking station via Bluetooth. Tasked engineers must set up the communication between the Bluetooth receiver/sender pair in the shoe and docking station and initiate data transfer from MicroSD to DHU.

To set up the User Interface, a database needs to be established. A server needs to be setup to hold the data received from the DHU. The data will then be sorted into a database using software that will be determined by the tasted engineers. To access this information, a web page needs to be developed for the end user. The web page will show the organized data in a visual format. The end user will be able to select the patient and view the data from the web page. Visual graph software will be utilized to help form the graphs for the end user interaction.

B. Fully Description of Each Task

In the table below, the name of every task in the project, along with its description, assigned engineer, estimated hours and completion criteria is listed below in Table II. [37]

TABLE II
Task Descriptions

Task	Description	Assigned Engineer	Estimated hours
Amplification Circuit	An operational amplifier will be used to amplify the analog signal from the pressure sensor. This will make smaller variations in pressure more precisely readable. The completion of this task is determined by the successful capture of accurate pressure information.	Sam/Pratik	5
Pressure/Tilt Sensors	Establish working sensors that will capture data and store them on a memory unit attached to the MC. The completion of this task is determined by the successful calibration and testing of the sensors which will yield accurate pressure values.	Sam/Pratik	3
Sensor placement	The pressure sensors will be placed at the high pressure points on the foot; the arch, forefoot, and toe. The completion of this task will be determined by capturing pressure data while the sensors are mounted and placed in various locations of the shoe while walking.	Sam/Pratik	2
Sensor wiring	The sensor circuit must be placed in the shoe so that it doesn’t impede the user's ability to walk, doesn’t damage the circuit and can be embedded in the shoe.	Sam/Pratik	3
Analog Conversion	The voltage output from the sensor amplifying circuit will be converted to an analog value and then converted to a force value in the microcontroller. Again, the success of this task is determined by accurate reading of the pressure.	Sam/Pratik	2
Input/output Mapping	Determine which pins to use on the MC that will allow for easy wiring of the circuits inside the shoe. This completed when a sketch is made that will guide the wiring of the shoe.	Sam/Pratik	4
MC Integrated Design Environment(IDE)	The microcontroller will need to be programmed and the IDE will expedite this process. This task will be complete when a MC is chosen and the associated IDE is acquired.	Sam/Pratik	6

AC to DC Converter	The AC to DC converter will convert the AC signal coming out of the wall to DC to power the devices in the docking station. This task will be complete when all components in the docking station have met their power requirements.	Sam/Pratik	15
Wireless charging delivery circuit	A circuit will be made to wirelessly deliver current from the wall to the receiver circuit to charge the batteries. This task is considered complete when the battery in the shoe charges in the allotted 8 hrs.	Sam/Pratik	20
Wireless charging receiver coil	This circuit will be made to receive the current from the delivery circuit. This task is considered complete when the battery in the shoe charges in the allotted 8 hrs.	Sam/Pratik	15
Rechargeable Batteries	These batteries will power the microcontroller in the shoe and recharged through the charging circuit. This battery will need to be small and meet the current requirements of the circuits in the shoe. This task is complete when a battery is chosen that meets the requirements, and when the battery charges from wired and wireless sources.	Sam/Pratik	15
Wired Charging Circuit	There will be a circuit supplying the DC batteries with power from the AC to DC converter through wired connection. This task is complete when the rechargeable battery is recharged from this circuit.	Sam/Pratik	8
Bluetooth connectivity	A circuit powered by DC power will gather the data from the memory storage on the microcontroller and deliver it to the data handling unit. This task is complete when data is successfully transferred from the shoe to the DHU.	Khoi/Kyle	40
DHU and MC Communication	A program will run on the DHU that initiates connectivity and facilitates data transfer from the MC. This task is complete when data is successfully transferred from the shoe to the DHU.	Khoi/Kyle	20
MicroSD Storage	The data will be stored on a microSD at regular intervals. This task will be complete when an SD card is chosen that will meet the	Khoi/Kyle	1

	minimum data storage requirements.		
DHU - Server Communication	The DHU connect to Server and upload the data. The completion of this task is dependent on storing and accessing the data on the server for use in data visualization.	Khoi	15
Raw Data Transform	The raw data will be formatted to suit a database model. This task is complete when the data is able to be placed in the database.	Khoi	5
Install/ Config Server	A laptop will be used as a server that will have all the application and Database management software (DBMS). The task is complete when the DHU is able to store data onto the server.	Khoi	5
App for DB Management	A platform for database management to create and manage database such as creating/dropping tables. The task is complete when graphs and tables can be produced in the database.	Khoi	2
App/GUI for End-User Interaction with DB	An application would be what the user interacts with, and would call stored procedures in the SQL database to manipulate data for best ease of use, performance, and security. This task is complete when the end-user can access the formatted data in easy to ready graphs.	Khoi	50
Visual Graph Software	An application that gets input from raw transferred from DHU and output the according graphs (Pressure vs Time). The task is complete when graphs are created with pressure and time data.	Khoi	20

To fully implement the Pressure sensor feature, the tasked engineers Pratik and Sam will have to put in a total of 19 hours. Setting up the pressure sensors will take the most amount of time at 13 hours in this feature. The other task these engineers will do is that of the Docking Station which will take a total of 57 hours. The bulk of these hours are taken up in the wireless charging tasks which take up about 45 hours alone. The wireless communication task will only take about 9 hours.

The tasks that the CPE's Kyle and Khoi have taken up are: the User Interface, and Data Handling. The total estimated time to implement both of these features is 227 hours. However, for the prototype in Fall 2016, the team decided to postpone the web page feature and Bluetooth connection for next Spring 2017. Khoi and Kyle also decided to develop a simple graph software for the prototype; thus the actual total time to implement is 147 hours. Moreover, the team spent roughly average 16 hours per report, that total up 96 hours for entire semester.

The total time to complete the entire project is about 888 hours. The time allocation is summarized in Table III below.[38]

TABLE III
Hour per Team Member Fall 2016 and Spring 2017

Task	Khoi Nguyen	Kyle Arango	Pratik Patel	Samuel Woldeyes
Documentation	**96**	**96**	**96**	**96**
Tasks/testing	**147**	**147**	**105**	**105**
Total hours	**243**	**243**	**201**	**201**
Total team hours	**888**			

VII. RISK ASSESSMENT AND MITIGATION

Along with being able to manage our time wisely, the team also will need to be able to prepare for potential risks that are associated with all the tasks.

Sources of risk can arise from complexity, limited resources, lack of experience, unexpected breakages and more. Our design incurs risk due to all the sources mentioned and additionally has several interconnected components which adds to the complexity of the project and further increases the risk.

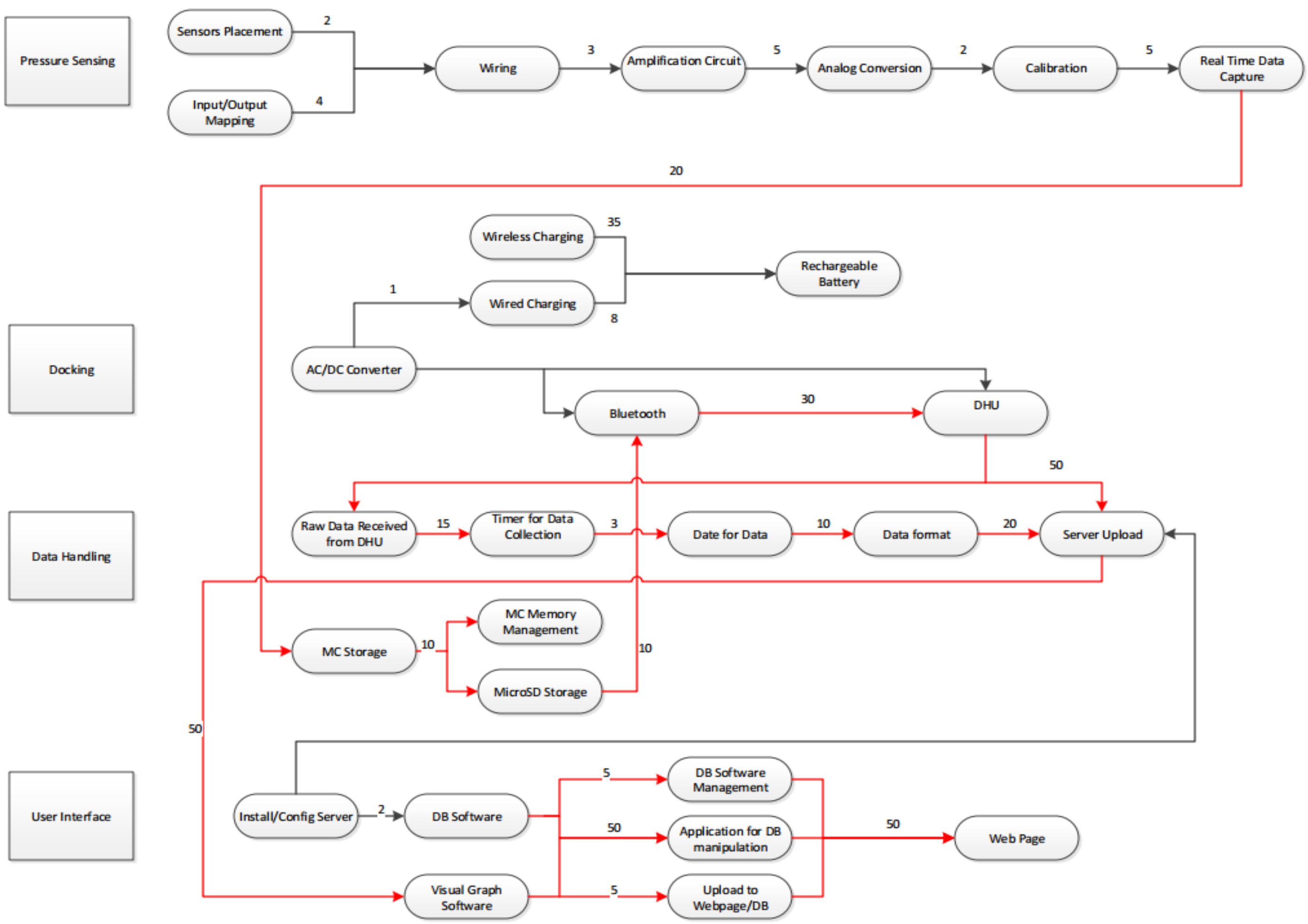

Fig. 11 Pert chart of diabetic shoe [39]

A perceived critical path has been identified in order to reduce the the risk; at the same time, a possible mitigation plan has been proposed in case the problem associated with the risk occurs. There are total five critical paths are identified: Real Time Data Capture to MC Storage path, DHU to Server Upload path, microSD to the DHU path, AC/DC converter to the rechargeable battery path, and DB software and visual graph software to the webpage path. The risk assessment for each path are fully described below.

A. Perceived Critical Path

In order to reduce the risk, it is important to identify the critical path of our project. Because of the nature of our design, there are many aspects of our design that are critical to the completion of our project.

From figure 11, there major risks we have identified that could potentially jeopardize our project if not address properly and in timely manner. [39]

First, the path from the Real Time Data Capture to MC Storage. The Real Time data transfer is dependent on several factors like having enough power to the MC, proper wiring of the sensor circuit and amplification circuit. Without being able to capture real time data, the entire feature set of data handling will not be possible, as there would be no data to handle. Steps to help facilitate the transfer of data can still be implemented however.

Second, the path from DHU to Server Upload. This link is critical because it will allow us to store the collected data to a server where it can be accessed later. Without this link, the user interface feature will not be possible. The DHU is dependent on having power from the AC/DC converter as well has having data received from the shoe.

The next path is from the microSD to the DHU. This path will allow us to bring off load the data from the shoe to the DHU via bluetooth. No one in the has ever worked on bluetooth and poses a problem. We may not get two bluetooth modules to communicate, which will be detrimental to fully implementing the feature set of the project. Our wireless data transfer depends on the crucial link in the chain.

Another critical path is from the AC/DC converter to the rechargeable battery, the bluetooth module, and DHU module. If we are unable to power or recharge any of these components, we will not be able to facilitate any data transfer from the shoe to the server.

The last path is from the DB software and visual graph software to the webpage. The web page depends on the the on the visual graph software to be up and running on the server, as well as a database.

B. Risk Assessment Chart

The Risk Assessment chart, seen in figure 12, compares the probability of failure on the y-axis with the impact to the project on the x-axis if the failure occurs. [40] The green zone represents a low probability of failure and low impact to the project. The yellow zone represents a high probability of failure low impact, high impact and low probability of failure, medium impact and medium probability of failure. The red zone represents high probability of failure and high impact to the project. This chart helps helps assess which tasks in the project need the most attention due to impact of the project and probability of failure.

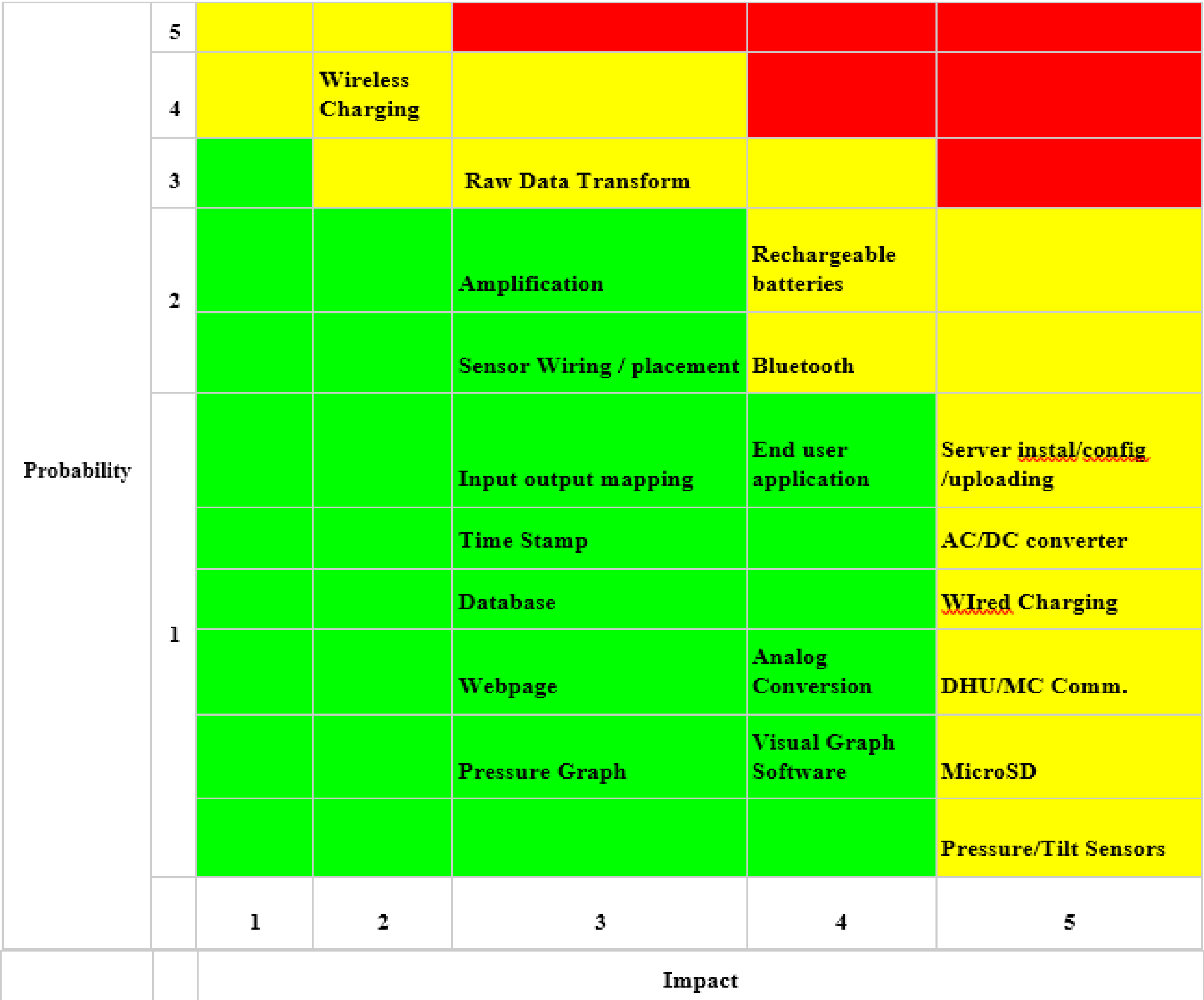

Fig. 12 Risk Assessment chart of diabetic shoe [40]

C. Mitigation for Critical Path

For the Real Time Data Capture to MC Storage path, the risk is getting the wiring and amplifications circuits to work properly as well as having the ADC on the MC to work properly as well. The probability of improper wiring is P = .3 due to the fact it's unreasonable to expect wiring to 100% right on the first try. The faulty wiring could result in shorts in the circuit they may fry an electrical component. The impact of the faulty wiring we believe is I = 3 because if a component does fry, we can replace them with relatively low cost. The risk of failure is R = .9. To mitigate the impact, we should have two engineers look over the wiring to make sure it's correct. Another thing would be to cut power immediately when see see an unexpected or unwanted behaviour from the circuit. Calibration of the sensors is not necessary to continue the work of data collection, as long as the sensors are responsive to change. The analog conversion depends on the MC and it's ADC working properly. There are a few things that can go wrong here, first the signal we need to pick up may be too small to read for the MC because the signal may be smaller than the noise the pin picks up. If this happens we can use forms of digital to clean up the signal so that we can view the changes in voltage from the pressure. We feel the probability of this happening is P = .1, and the impact being I = 4 with risk being R = .4. The probability is low because the signal we are dealing with is in the 30 mV to 5V DC range, which is supported by the microcontroller we are using. [41] If for whatever reason this is not the cast, it may jeopardize the project, as there would be no way of knowing if the pressures we are measuring are accurate or even working without proper instrumentation. The amplification circuit has the same probability and impact as wiring the circuit. The op-amp that we are using is specifically called in the documentation of the sensor, the MCP6004. [42] If a chip fails, we can try to replace the chip with the same model. If that still doesn't not work we can use another model with similar specs to of the one called out in the document.

For the DHU to Server Upload path, without this connection, we will not be able to upload any data to the server for access by the end user. We plan use a virtual server on a laptop. The probability the virtual server failing is P = .1, the impact of the server failing is I = 5, with the risk of failure being R = .5. Without the server we will not be able to have a database to store the information, however work on the data can start and can be integrated into the project at later time. If we cannot get a server connection up and running, we may need to manually move the data from the DHU to a computer via USB drive. If the server fails to run, and communication is not possible, we will try to move to cloud based storage like dropbox or google drive and have the DHU handle the data transformations and graphing.

The next path is from the microSD to the DHU. This is where bluetooth will be involved. Microchip's device [43] uses 70mA to transmit and receive, and is designed for our application of wearable technology, is high, but we will be charging the shoe during transmission of data, we will need to meet the current requirements for charging the batteries and transmitting data. The probability of Bluetooth failing P = .3, and the impact is I = 4 with the risk of failure bing R = .8. No one on the team has ever worked with bluetooth and poses a challenge, and is where a majority of the risk comes from. If we fail to have bluetooth communication, we will have to have a wired connection to transfer data similar to that for the rechargeable batteries. USB is capable of transferring power and data simultaneously.

For the DB software and visual graph software to the webpage path, we need to have a database up and running, as well as a server to help manipulate the data. The task is time consuming but not impossible, the probability of not having visual graph software is P = .1, and the impact is I = 4, with risk of failure being R = .4. Without the visual graph software step, understanding the data will be difficult as there will be no graphs, just numbers in a table. It would take the user a long time to understand the data, which is antithesis to the goal of the project; however at least some form of data is able to be read, the data tables can be put into an Excel worksheet to be graphed manually by the user if we cannot get visual graph software ready. If we cannot get the webpage ready, we can have the text form of the data for the user to download, then they can put it in an excel worksheet to visualize the data.

A lot of the data transfer relies on the AC/DC converter doing its job and providing the right amount of power for the electronics in the docking station. The wired and wireless charging capabilities rely on this as well. The probability of the AC/DC not providing power is P = .1, however the impact I = 5, with risk of failure being R = .5. There will be no way of powering the shoe for the working prototype. To help limit the impact during build phase, we can use batteries to power individual components, but ultimately the AC/DC needs to be operational to provide the wireless data transfer and charging capabilities, without it there is no working prototype that fully meets the design features.

D. Mitigation for Sub-Tasks

Each subtask has its own associated risks. They are detailed below. Tasks relating to the critical paths are also listed below.

1. Sensor Risk Assessment

a. Amplification Circuit

The amplification circuit relies on the fact that none of the wires or op-amp are damaged or moved out of place due to motion of the patient's foot during movement. To avoid damage or movement of wires out of place, a circuit cage will be placed over the circuit so it won't come into contact with external forces. Another risk of failure is improper wiring of the circuit. This can lead to damaged components and cause parts to be replaced. The probability of this happening is P = .3 and the impact is I = 3, where the risk is R = .9.

b. Pressure/ Tilt Sensors

The risk with sensors is if internal components within the sensor stop working properly. If the sensors begin to stop working properly, they will be replaced with other sensors. This can be due to any number of reasons, mainly being human error in wiring the the circuit properly. Another possible source is applying a load that is greater than indicated by documentation The probability of improperly wiring the sensors is P = .3 and the impact is I = 3, where the risk is R = .9.

c. Sensor Placement

The task of the pressure sensors is to sense the pressure at key points of the foot. This can't be done if the sensors are moving around within the shoe. Also if the pressure sensors are under more pressure than they can withstand, they won't be able to function properly. To prevent the sensors from moving around under the foot in the shoe, the sensors will be embedded in the soles of the shoe. To prevent sensors from being under more pressure than they can withstand, the user must be kept below a certain weight limit. The probability of improperly wiring the sensors is P = .3 and the impact is I = 3, where the risk is R = .9.

d. Sensor Wiring

The wiring used for the sensor circuit must be made so that none of the nodes are made open during motion of the shoe. The wires can come loose from the circuit while the user is walking causing lose of data, or unwanted shorts in the circuit.

The wires used for the circuit will be embedded in the material of the shoe to reduce opening any of the nodes. The probability of failure is low, as we will be working with the vendor to accomplish this P = .3. The impact is high, I =3. The risk is R = .9. Any faulty wiring will jeopardize the project causing lose of data or frying electrical components.

e. Analog Conversion

The analog conversion converts the voltage from the output of the amplification circuit to an analog value that the microcontroller reads. If this isn't done the data can't be properly converted to pressure values. To prevent the risk of the analog conversion not working, we need to stay within the limitations of the MC and amplifier we use so that a signal can be acquired. The probability of the ADC not working is P = .1 with I = 4 and R = .4.

f. Input / Output Mapping

The input/output mapping displays how the ports in the microcontroller are being used. This mapping will be used by the vendor to help wire the shoe for the deployable prototype. Any failure in this will result in more time being added due to the fact the shoe needs to be refabricated with the correct wiring. To help mitigate this, two engineers should look over the mapping to make sure that it is correct, also apply a dry run by following the schematic to make sure the mapping is correct. The probability of this being wrong is P = .1, I = 3, and R = .3.

2. Docking Risk Assessment

a. AC to DC Converter

The AC to DC converter powers the DC devices in the docking station. If it doesn't work properly, data can't be transferred to the server and the shoes can't be recharged. If the AC to DC converter doesn't work, then we will buy a premade converter rather than make one. For the lab prototype, we can also use batteries as a power source, however for the deployable prototype, it needs to be functioning. The probability of this happening is P=.1 and the the impact is I = 5 with risk being R = .5.

b. Wireless charging delivery circuit

The wireless charging receiver coil must supply enough current to the batteries so they can charge overnight. If the inductor doesn't deliver enough current then the shoes won't charge. If the wireless charging delivery circuit doesn't work then wired charging will be used to charge the batteries while the shoes are on the docking station overnight. The probability of this not working is P = .7, with I = 2 and R = 1.4. The probability is high due to the space constraints of the shoe. If the circuit required to charge is too large, wireless charging will not be possible.

c. Wireless charging receiver coil

The wireless charging receiver circuit will receive the current being delivered by the delivery circuit. If the receiver circuit isn't receiving any circuit, then the shoes won't charge. If current isn't being received by the receiver coil, then a wired charger will be used to charge the batteries while the shoes are on the docking station overnight. The probability of this not working is P = .7, with I = 2 and R = 1.4

d. Rechargeable Batteries

Using rechargeable batteries depends on the current the battery can deliver, and how much current our project requires. Not only this, but the batteries have to be meet the space constraints of the shoe. If they can't be recharged, then the shoe won't collect data. If the batteries can't be recharged, then interchangeable batteries will be used. The probability of this not working is P = .3, with I = 4 and R = 1.4. Not being able to find right batteries will limit the implementation of the full feature set. If the patient has to replace batteries, then the goal of minimal interaction is not met.

e. Wired Charging Circuit

The dc batteries in the shoe will be charged by a wired battery charger while the shoes are placed on the docking station. If the AC/DC converter in the docking station doesn't work properly, then the batteries won't be able to charge.
Batteries will be changed once they die. No recharging will be done. The probability of this not working is P = .1, with I = 5 and R = .5

g. Bluetooth connectivity

The data from the memory storage is sent to the data handling unit by Bluetooth connection. If the bluetooth doesn't work, the data can't make it to the server so this task has a high impact. If the bluetooth doesn't work, then a wired data transfer will be used to send the data to the data handling unit. The probability of this not working is P = .3, with I = 4 and R = 1

f. DHU and MC Communication

A program will will run on the DHU that initiates connectivity and facilitates data transfer from the MC. This task is entirely based on software so the risk is very low but the impact is high. If the software that is fabricated doesn't enable contact between the DHU and MC, then software from a open source site will be used. We will have to use wired data while the shoe is charging to help mitigate the failure of this task. The probability of this not working is P = .1, with I = 5 and R = .5

3. *Data Handling Risk Assessment*

a. MicroSD Storage

The data needs to be stored on an SD card until it can be transferred off the microcontroller. If we can't get the processes running to store and retrieve data off the MicroSD, then only a few seconds can be recorded before the microcontroller's memory fills up. If this task cannot be done, then we would have to be very selective of what we record and it would become very important that the battery life is long enough to last until the next data transfer off of the device. The probability of this not working is P = .1, with I = 5 and R = .5

b. Server Uploading

An automated upload to the server is wanted for minimal user interaction. The process of automating, however, could be tricky because there are may scenarios that need to be considered including cases where unintended or unexpected events occur. To reduce the risk of this particular task we need to make sure that the data is backed up and can at the very least be accessible on the DHU. So all the data collected by the DHU should be copied and backed up on the DHU. This will also reduce the risk of deleting large amounts of data with no hope of recovery. The probability of this not working is P = .1, with I = 5 and R = .5

c. Raw Data Transform

The raw data needs to be organized for the server to read and recognize it. Bugs and debugging could cause delays and we need to insure that the data is unchanged by the transformation process. The code that performs this function should be simple and kept very small, if we keep the code small then debugging will be faster. To ensure the data integrity we can provide two ways of transforming the data and run both ways and make sure the data matches. If data cannot be transformed, we will use raw data, and we can use the sampling time to make a graph of the pressure vs time. The probability of this not working is P = .1, with I = 3 and R = .3.

d. Time Stamp for Data Graphing

Timestamps will be used to help graph the data over time and indicate to the viewer when data was collected. A timestamp should be fairly easy to implement by taking time data at the same time we take sensor data, so the risk is low, however problems with taking time and sensor data at the same time could arise. The data will most likely still be stored in chronological order though, so even without timestamps the data can still be graphed, just not accurately and not at specific times. The probability of this not working is P = .1, with I = 3 and R = .3. If data cannot be transformed, we will use raw data, and we can use the sampling time to make a graph of the pressure vs time

4. *User Interface Risk Assessment*

a. Database Server

The database relies on the server to operate. If the server goes down, no information from the pressure sensors can be recorded and the transmitted data will be lost. Another probable risk could be the hard drive failure. Since the server is required to operate 24/7, the hard drive used as DB storage can be failed anytime resulting in permanent data lost. To mitigate the risk, an easy and non-cost solution is to backup data manually on to external hard drive daily or weekly depending on how often the patient uses the shoes. Another solution is migrating the on-site server to the virtual server on cloud. This solution requires monthly cost to maintain the server and maybe challenging due to the platform disparity. The probability of this not working is P = .1, with I = 3 and R = .9.

b. End-User Application

The End-User application requires data stored on the Server as an input so that it can retrieve and manipulate. If the data stored on the Server is corrupted or not available, it may cause the application to be crashed. However, the chance for the data to be corrupted is very unlikely to be happened, the risk of this task is very low. The only way to mitigate this risk is to re-capture the data from the sensors, but again, the chance for data to be corrupted is very rare. The probability of this not working is P = .1, with I = 4 and R = .4.

c. Pressure Graph Software

The visual graph software depends on raw data transferred from DHU to generate according pressure graphs. If raw data is not in the correct format (i.e pressure data with timestamp), the graphs will not be generated correctly. The idea for using the graph is to make the data easier to read and analyze. If the graph is not generated as expected, creating a table sheet with pressure data and timestamp can considered a good work around solution. The table sheet can be put into an excel sheet to graph for visualization. The probability of this not working is P = .1, with I = 3 and R = .3.

VIII. DESIGN DOCUMENTATION

A. Design overview

The guiding design philosophy for this project was to minimize the interaction between patient and tool. There is a problem with patient compliance with treatments like these, the patient often does not follow through on adhering to the treatment, therefore needing to minimize the interaction with patient and treatment is essential. The only interactions the patient should have with this treatment is for them to wear the shoe, and place it on the docking station for charging. To accomplish this, the entire process of data collection, storage, transfer, and presentation needs to be automated.

Starting with data collection, as soon as the shoe is put on it will need to start gathering data. Having any sort of manual switch would require patient interaction, which is against our design goal. From here data will be collected nonstop for approximately 2 hours. After the two hours, the system will wait for the battery to be in charging mode. This will happen when the patient places the shoes onto the docking station. At the docking station, the data will begin to unload from the SD card onto the DHU where it will send it off to the server where it can be viewed later.

The only portions of this project that are not automated are analyzing and the acquiring the by a use from the database. Once the sensors pick up data, everything from capture to presentation is automated through code. Further details into the hardware, software and mechanical aspects will be described in their respective sections.

Top down level of Design Overview is shown in figure 13. [44]

Figure 13- Top Down Design Overview

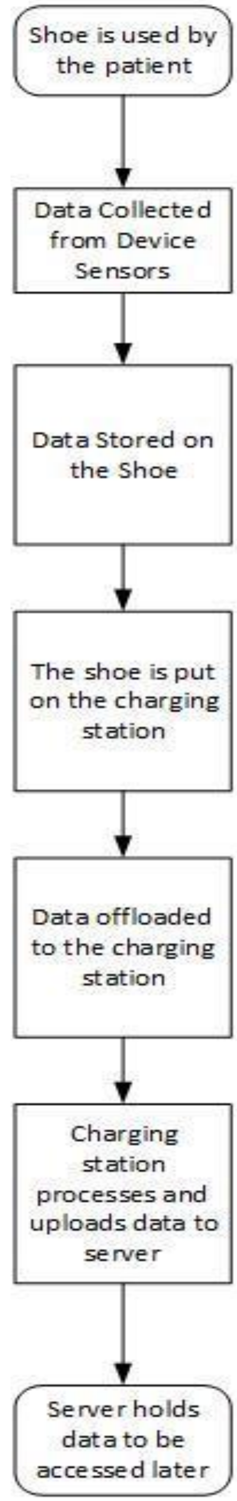

Fig. 13 Top down level of Design [44]

B. Implementation of Design

1. Data collection

We first started work on the pressure sensors.

We found out from our vendor that the resolution of the pressure data we capture does not need to be finely resolute. With this in mind we decided to use Flexiforce A304 sensors with a sensing diameter of 9.7 mm. This sensor works very much like a photo-resistor. When a force is applied to the sensing area, the resistance of the sensor decreases, the circuit closes and allows the flow of current through the sensor. The sensor has a very high resistance, even when it is closed, and will yield very small voltage values from the sensor. To solve this problem, we used the linear op-amp MCP6004. The op-amp is set in the inverting configuration, the sensor is supplied with a negative voltage so that the output will be a positive voltage. Figure 14 shows a picture of the sensor (a) and the op-amp (b). [45][46]

Figure 14 - Op-amp and A301 sensor [45]

To capture and store the signal, we use the Parallax Propeller (Fall 2016) and Adafruit Feather 32u (Spring 2017) in figure 15 as our MCU. [47] The MCU has ADC converters that we will used to convert the analog signals coming from the sensors into digital. The MCU's ADC captures voltage values from 0 to 5V spread across 0 to 4096 bits. One of the main reasons the propeller was used is because of built-in microSD capabilities.

Further detail of MC code will be described in Appendix C.

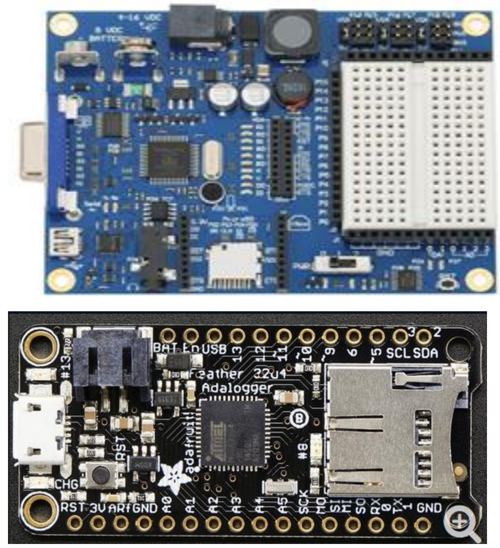

Figure 15 - Propeller and Feather 32u MCU [47]

Another aspect of proper data collection is the placement of the sensors. The sensors will be placed in the common areas that foot ulcers are known to develop, this is shown in the figure 16. [48]

Figure 16 - Common foot ulcers areas [48]

A sensor will be placed in the toe, 3 will be in the pad of the foot, and one in the heel. The arch of the foot is not included for measurement. The locations chosen for these sensors are in line with similar other products that will be in the market. Also, a study done in 2013 that tries to prove the concept of our own product, chose similar locations for their own testing. The conclusions of their testing were that these locations can be used for ulcer detection on a day to day basis. Any slight change in pressure values detected by the sensors will indicate the onset of foot ulcers. [48]

To doctors further analyze the data they are receiving from the product we also used an accelerometer to help determine the orientation of the foot in space with respective to each force measurement attained. This helps determine motion of the foot as the pressure is being measured to rule out any anomalies of high pressure like picking up an object.

2. Docking Station

The docking station serves as a recharging station and a place for the data to be offloaded to the DHU. Power is supplied to the DHU and wireless charging circuit through the AC/DC converter which supplies 5V and about 1A of current. The AC/DC converter consists of a step-down transformer of 120VAC to 24VAC, then goes into a full wave rectifier made of 1n4004 diodes. The supply then goes into a voltage regulator (LM2596) which steps is down to 5V and includes a filter capacitor to get rid of any ripple at the output of the full wave rectifier.

The raspberry pi is used as our data handling unit which organizes the data it collects from the microSD and sends it to our server. The raspberry pi takes up about 400mA during the data transfer which is within the limit of our supply. The raspberry pi must be reset in order for data to be transferred by the Bluetooth so a hard switch was used here. When the shoe is placed on the docking station, the switch is closed and power goes to the raspberry pi. When the shoe is taken off, the switch is open and the pi remains off. The receiving Bluetooth module is built into the pi so this is one of the main reasons the pi must be reset for new data to be received.

Also in the docking station is the wireless charger. The wireless charger consists of 2 inductor coils; receiving coil is 14µH and delivery coil is 3.7µH. The wireless charger charges the battery at a rate of about 70mA per hour. The consumption of our battery is about 200mAh with Bluetooth running and the system runs for only the first 2 hours after pressure is sensed in the shoe so the expected consumption of the battery throughout the day would be 400mA. Our battery is rated at 2500mAh so we don’t expect any drop-in voltage or current over time during the first two hours. Our results were verified during testing. At 400mA of consumption, it would take about 6 hours for the battery to recharge to full. The range of the wireless charger is about 1 inch so some supports and an outline were included on the docking station design to ensure the coils are aligned well enough to allow full current transfer from delivery to receiver coil. The schematic for the docking station layout is in the figure 7 in the design idea section.

3. Data Handling

a. MCU and DHU Communication

For the Microcontroller and Raspberry Pi to communicate via USB connection (Fall 2016) or via Bluetooth (Spring 2017), a python code is implemented and run on the Pi to read data from Serial port.

The idea of this python code is to make a handshake communication between two devices via serial port.

For Fall 2016, the USB port (serial port) was established. The pin Tx>P31 and pin Rx<P30 on Propeller. See Fig 17. [50] and the udev/USB port on the Pi were initialized for data transmit and receive. Both baud rates of 2 devices also were set to 115200 meaning the serial port can transfer a maximum of 115200 bit a second. The baudrate of the Propeller and the Raspberry must be identical for read and write.

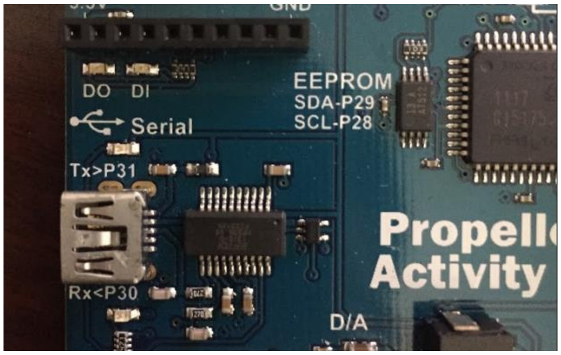

Figure 17 : Serial Port on Propeller Board [50]

In spring 2017, we revised the code so that both devices can communicate via Bluetooth. The pin Tx(transfer) and Rx(read) are now set to match the Tx and Rx pins of the Bluetooth module (See Fig. 18 [51]. The baudrate is the same 115200 on both devices. we revised the python code to initialize with **/dev/rfcomm0** which is reserved for Bluetooth port on the Pi.

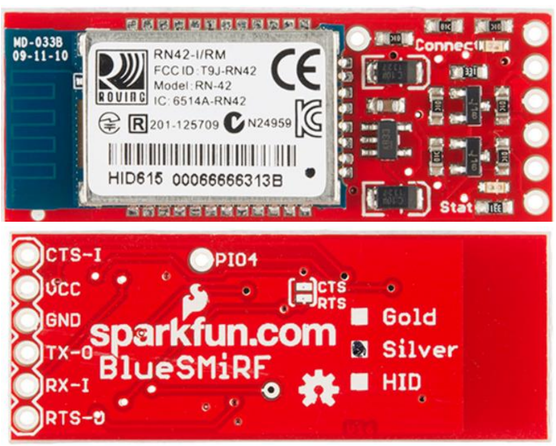

Figure 18- Bluetooth RN42 module [51]

After the communication channel is established, the data should be freely flow from Propeller to Raspberry Pi.

b. Automated Transfer Process

The transfer process takes the data saved on the shoe and transfers it to a server where it can be accessed by the client software that can graph the data. The path to get to the server involves an intermediate transfer to the DHU in the docking station.

The data is transferred off the shoe through the Bluetooth connection to the DHU. The DHU receives the data and scan the data to see if there are any potential errors in communication. If there are any errors, it throws out the invalid data automatically. After the data has been cleaned it is sent to the server via SSH connection.

The whole process is initiated by setting the shoe on top of the docking station. Two signals are triggered when the shoe is set in place: the signal to the microcontroller and the signal to the DHU. The signal to the MC comes from the wireless charging system. When the shoe is in range to be charged the charging coil outputs 5v to which is used as a signal to put the MC to initiate transfer. On the docking station a switch is depressed by the heel of the shoe which gives power to the DHU and it automatically starts up. On startup, the DHU runs scripts to receive the data, clean up the data, and send it all to the server. See Figure 19 [52]

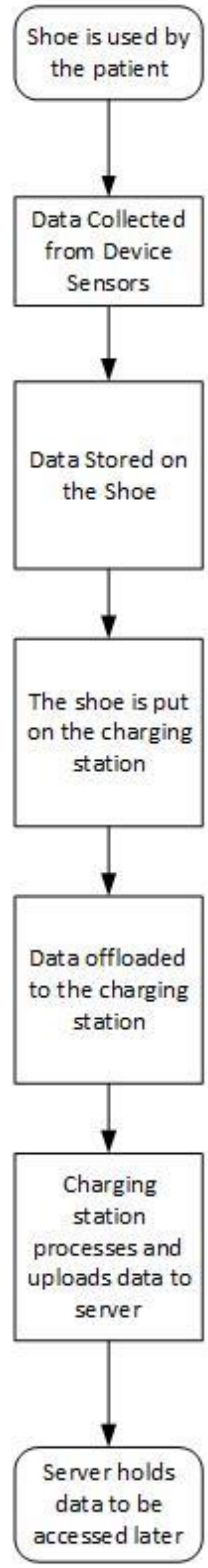

Figure 19. Dataflow Overview [52]

A. DHU

The overview of the DHU shown in figure 20.[53]

The DHU is a Raspberry Pi 3 Model B inside of the docking station. It is in charge of collecting data from the shoe once the shoe is docked, analyzes the data to reject any malformed data, and transfers the data to the server.

When the user is done with the shoes for the day they put their shoes on the docking station; this is the trigger for automatic data transfer to occur. A touch sensor on the docking station senses when the shoes are placed on the docking station and a signal is sent to the DHU to prepare to receive data. At the same time the shoe begins to charge wirelessly and a signal from the charging coil is sent to the microcontroller(MC) on the shoe to prepare to send data. The DHU prepares by restarting and running scripts to establish Bluetooth connection and listen for sent data. The MC prepares by going into a delayed sending state. The state has a delay to allow the DHU to fully restart before the data attempts to send.

The DHU waits until data is collected and once it receive a "Done" signal over the Bluetooth communication line it know to stop collecting data and to begin analyzing the data for malformed portions. Bad data can cause bugs in the graphing functions on the server client so the DHU looks at every line of data and ensures there is the expected number of data objects and the expected type of data. If any bad lines are found they are thrown out.

After the data has been analyzed it is sent to the server through SSH.

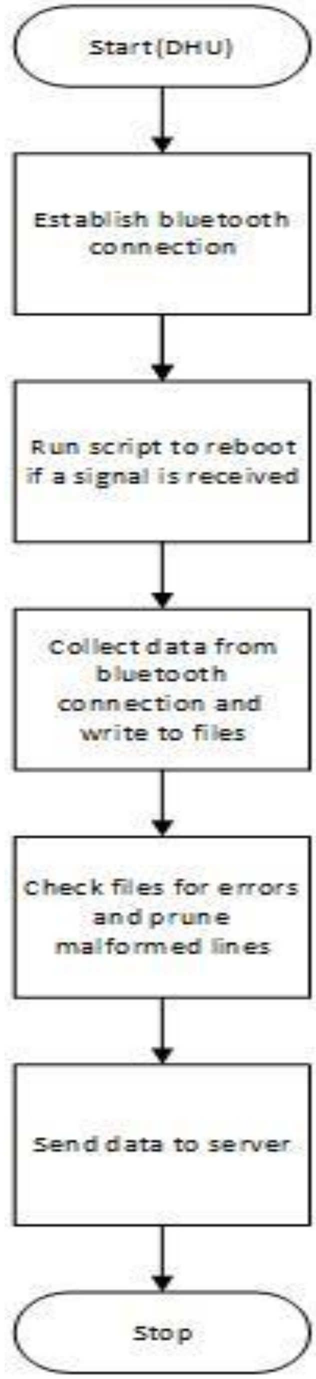

Figure 20. DHU Overview [53]

B. A More Detailed Look

On start up the DHU runs a script (see figure 21) that starts up three other processes. [54] The first is process that runs in the background to connect to the bluetooth module and sustain that connection using the command "rfcomm connect 0 [MAC:ADDR]" where [MAC:ADDR] is the MAC Address of the bluetooth module connected to the MC. This establishes a connection on rfcomm channel 0 and is accessible via "/dev/rfcomm0". Part of the reason why we need to restart is to ensure this channel is not inaccessible, say from a bad connection previously.

The second process is a script that runs in the background continuously to see if a signal has been sent to reboot. If the signal is received this process starts the reboot sequence immediately.

The third process is another script which has three sub processes: collecting the data and putting it into files, parsing and analyzing the files to prune any malformed data, and sending all files to the server.

To collect data the script starts a serial connection with Python serial over the earlier establish Bluetooth connection. The serial port is opened on the "/dev/rfcomm0" port and the script continuously listens for any sent data. One it does receive data is starts logging what was sent until an end-of-file marker is reached. At this point it will increment the file number and continue collecting until the expected number of files are sent,

One file is sent for each pressure sensor and two files are sent for the gyroscope data collected. The data format for the pressure sensors is "time, pressure reading". The data format for the gyroscope is "x, y, z". This expected format is used in the next part to parse the files and find any pieces of data that don't match up and disregard that data.

The parser read each file line by line and determines the number of data objects, if they is an incorrect number that line is disregarded. If there is a correct number of data objects it makes sure every data object is the correct format expected (integer or float). If anything is incorrect the line is disregarded. All lines not disregarded are rewritten into a now pruned file free of malformed data.

These new pruned file are then sent to the server over the internet through the command "sshpass -p 'passw' scp -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no [fileToSend] [Server]:[Destination]". The command sshpass is used to pass the server password into the command "scp", which copies the [fileToSend] to the [Server] at the [Destination]. The other command modifiers are used to allow this process to be automatic otherwise user interaction is required.

All software portions have been confirmed to work as intended in the deployable prototype.

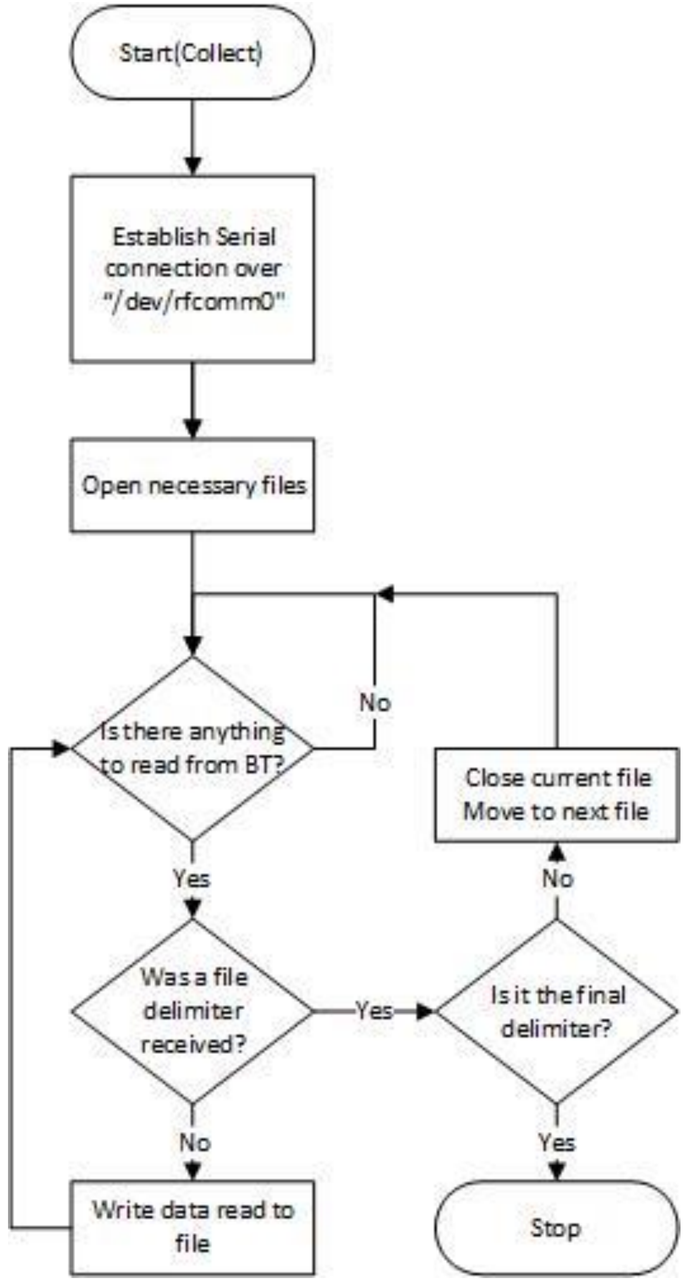

Figure 21. Serial Collection and File Writing [54]

4. User Interface

a. SSH server

The idea of setting up the SSH(secure shell) server is to transfer file from the Pi (linux based) to server (windows based) using scp (secure copy protocol) command. That means you can copy, edit, move and delete files from a remote to a host computer or vice versa. This SSH network protocol secures the connection between the two connected computers and allow you to access and manage your PC from another computer using the Linux/Windows command line.

Further detail on setting up SSH server described in section IV, appendix C.

b. Database

The Database was written in sqlite language and SQlite studio(free software) was used to create and manage our database such as creating tables, delete tables or adding attributes.

An application was developed for end users to interacting with data information written in C#.

Further detail about database schema described in section V, appendix C.

c. Graph Application

Collected data is also presented in a graph form.

To simplify and make the program to be more friendly to use, we develop a graph application which is built-in the database application.

This graph application will generate a graph based on the output text file receiving from the Raspberry Pi. See fig.22 for a sample successful generated sample graph.[55]

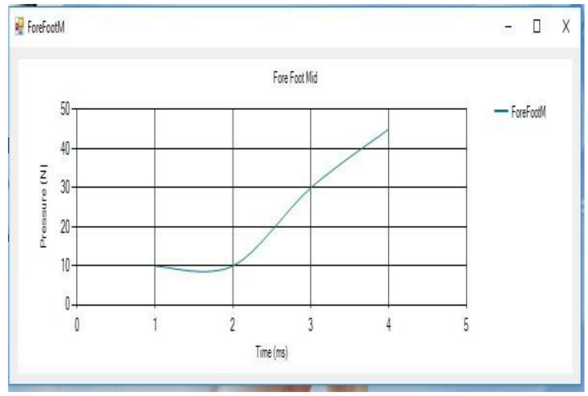

Figure 22 - Sample of Data in Graph [55]

IX. DEPLOYABLE PROTOTYPE STATUS

A. Summary

The deployable prototype performs all the core tasks in our design. These core tasks facilitate data acquisition and data transfer. The laboratory prototype can accurately gather accurate pressure data, store that data for later transfer, and transfer the data off the device and onto a server. Our deployable prototype includes wireless charging, wireless data transfer, tilt sensing, and high frequency data gathering, which wasn't utilized in the laboratory prototype. All the features currently implemented were polished and/or adapted to meet design features to improve upon the previous semester's prototype.

For pressure sensing, there are 5 sensors mounted in the shoe approximately where they need to be to obtain the most useful data. A variable force apparatus has been created to help calibrate the sensors to ensure that the data acquired is accurate and reliable. The accuracy is important in the end product and the apparatus can be used to calibrate again after finalized mounting positions and methods are decided. Each pressure sensor is connected to an amplification circuit so the microcontroller can more precisely detect the amount of pressure.

The pressure sensors are connected to the microcontroller which facilitates data transfer. After data acquisition the data is able to be stored on a microSD card to be held until data transfer. The deployable prototype can gather as much data as the microSD card can hold. A hard switch is used to supply power to the DHU when the shoe is placed on the docking station. When the shoe is placed, the wireless data transfer will begin and the raspberry pi will run through its script. The benchmark of 100 measurements per second has not been obtained yet.

The docking station that the shoes are placed on while not in use serves as a charging and data offloading station. Our completed deployable prototype portion of the docking station is powered using an AC/DC converter which supplies up to 1A limited by the 1n4004 diodes used. The AC/DC is connected to the hard switch which is connected to the raspberry pi. When the switch is closed, the script on the raspberry pi starts and collects the data from the microSD. When the shoes aren't on the docking station, the raspberry pi is off. Also in the docking station is the delivery coil for wireless charging. From testing, we found that the wireless charger has a range of about 1 inch and charges the battery with a constant current of 100mA.

Time stamps are also added on the MC; however, we are currently using a simple counter to approximate the amount of elapsed time. Later on we need to decide on the resolution to use for the time stamp, which will depend on how short the delay between sensor reads can be while still acquiring accurate data. The shoe will collect data for the first two hours are pressure is sensed in the shoes through use of a timer.

Data is transferred from the MC to the DHU. For the transfer to happen a Bluetooth module is used and the DHU automatically recognizes the device and prepares to receive data. A script on the DHU is run which reads data in from the Bluetooth module and saves the data to a file to transfer to the server and at the same time is backed up on the DHU for redundancy. By the end of Fall 2016 we had a wired transfer and we modified the existing protocols to be used over Bluetooth instead of a wired USB connection.

After backing up the data on the DHU it is sent to the server. Currently, the server is run on a laptop so we can easily modify and test the server as needed. The data can successfully be received and stored on the server where is can be accessed remotely if the correct permissions are granted. A graph of pressure over time can be obtained when accessing the server.

B. Device Test Results

Device test results shown in Table IV [55]

TABLE IV
Test Plan Results

I	Description	Expected results	Actual results	Pass/ Fail	Additional comments
P01	**Amplification circuit** -check pressure value detected by microcontroller -compare microcontroller pressure value to actual value	-pressure values detected by microcontroller are within 5% of actual values	values compared with spring loads; 5 tests were done, within 4%	Pass	acceptable amount of noise at output
P02	**Pressure Sensors** -check whether resistance across sensor fluctuates as different amounts of pressure are applied	-resistance of the A301 sensor changes as force is applied	resistance fluctuates inversely to amount of force applied	Pass	
P03	**Sensor placement** -verify sensors are placed in the shoe so that wires don't come loose while user is in motion -verify that sensors are being placed on parts of the foot that apply the most force on the ground	-sensor connection remains secure -sensors are placed in high pressure points	sensor is still able to collect data when placed under foot	Pass	unaffected by heat when foot placed inside shoe
P04	**Gyroscope** -place gyroscope on surface to simulate a shoe and rotate the surface and observe the changes in angle values	-gyroscope angle values displayed are within 2 degrees of actual angle values	able to sense change in axis, values are within 2 degrees	Fail	implemented on Arduino, not propeller yet
S01	**AC/DC Converter** -apply load to AC/DC converter that is equivalent to all devices in docking station -measure current leaving the AC/DC converter -verify voltage is large enough to supply other devices in docking station	-AC/DC converter outputs regulated 5V - AC/DC provides enough current to drive all the electronics	able to continuously supply 1A of current to resistive load	Pass	devices in docking station won't take more than 750mA so tested for 1A.
S02	**Wireless charging circuit**	-current is	Output of	Pass	coils should be

	-delivery coil supplies current to receiver coil connected to a load -battery receives current while within range and recharges	received by load	wireless charging circuit is kept at 5V so current is based on load. Max current at 400mA with battery		aligned perfectly to achieve max current
S03	**Bluetooth circuit** - Verify that Bluetooth module connected to Propeller works properly - Connect propeller with different Bluetooth devices such as: Pi, iPhone, android etc.	- Before pairing the Bluetooth module with the DHU it should be able to connect and be detectable by other devices.	Paired module with raspberry pi and able to transfer data	Pass	can also connect to android devices and display data
D01	**MicroSD test** -Verify that all data collected from pressure sensors are stored onto microSD in a right format. - Power on Propeller for ~ 30 secs for data collection, then stop.	Data should be transferred and saved onto a text file in MicroSD in the right format.	Able to store collected data on 4 output files on microSD	Pass	may need to use MicroSD-SD adapter to read output files
D02	**MC-Pi data transfer test** - Verify that data is transferred successfully from MC to Pi via Bluetooth connection.	- Collected data from MC should be present on Pi after Bluetooth transferring	Able to transfer data from Propeller to Pi	Pass	
D03	**Pi-Server data transfer test** -Verify that data is transferred successfully from Pi to Server via Wi-Fi connection - The transfer process will be run sequentially after the Bluetooth transfer process is done.	Data from the Raspberry Pi will be present on the Server as a text file.	Able to transfer output files from Pi to Server	Pass	
D04	**Data transfer automation test** -Verify that all the data transfer process is done automatically without user interaction - Place the shoe on the docking and the transfer	Data should be present on the Server after all the transfer process is done	Process is done automatically w/single button push. Data is present on server.	95%Pass	One button push is needed, but the data transfer is successful

	process should be start automatically from MC-Pi-Server				

X. DEPLOYABLE PROTOTYPE MARKETABILITY FORECAST

Our project is not a completely new idea, there exists companies with similar products in the market. We have not invented something that is completely new to the market but instead used existing technologies to implement our own diabetic shoe with unique features to compete with other competitors. The targeted recipient of our product are people with foot ulcers. Our product isn't just specifically for diabetic patients, but also for anyone that struggles with foot ulcers. There is competition in our current market due to the fact that other similar products have been introduced. The applications of our product can go outside just the realm of people with foot ulcers like monitoring military personnel with foot problems. We aim to enter the market with our product serving customers with foot ulcers and then expanding to other markets like military use and/or athletic training.

Our largest competitor in this market will be the company ORPYX - Figure 23.[56] They have a product called the SurroSense Rx which for the most part performs the same way as our product. Our target market is the same as our competitors so this is a main reason why we view ORPYX as our largest competitor. Most of the other competitors in the market use boot-type product to relieve pressure off the foot.

Fig. 23- SurroSense Rx, competing product [56]

The SurroSense Rx released by our competitor is a pressure pad that is adaptable to the shoe that the customer is wearing.[1] The battery is attached to the laces on top of the shoe. The data is gathered on the pressure pad and sent to a smartwatch that comes with the purchase. Some strengths of this product over ours is that it is compatible with most shoes while our product requires that the customer wears the provided shoe. Another strength is that this product provides the user with feedback on their walking via the smartwatch which provides a pressure map. It's weakness however is the high cost of $3400.

Our prototype performs all the core tasks in our design. To refine the deployable prototype to be more desirable in the market, user involvement in the use of the product must be reduced to a minimum, more data should be gathered to give the caring physician more knowledge on the problems the patient encounters, and the presentation of data should be cleaner and easier to read.

Currently a physical switch needs to be depressed by the shoe in order to initiate transfer. This relies on the user correctly placing the shoe onto the docking station. To improve this we would need to have a triggering mechanism that relies on the Bluetooth connection instead of a physical switch. This would require software changes to provide reliable signals between the microcontroller and the DHU. Providing software synchronization should be a high priority for any continued development because it would increase reliability, speed of transfer, and would create less interaction from the user.

On the hardware end we would like to be able to take a higher resolution pressure map of the foot. This would require more sensors or different types of sensors. Our competitor has a very high resolution pressure pad, however this resolution comes at a high price. Our shoe is low cost in comparison, and this allow us to compete in the market. The resolution we have now is sufficient but we would like to add a few more sensors to increase the coverage area. With more sensors however we run into the problem of not being able to take enough samples, especially if an interface is needed to expand the number on analog pins. This would require more sensors and would require a faster microcontroller and hopefully one with a large number of analog pins.

On the software end, data graphing and a web page could be used to better visualize data. Currently we have the data display as a graph of pressure vs, time for each sensor. This is fine for small time slices but quickly becomes useless as a way of visualizing the data and the walk cycle. Improvement would need a graphical pressure map of the foot and a user interface to help navigate through time. With enough development time, it would be nice to see an analysis of the gait cycle as well as an approximate animation of the gait cycle. The animation could greatly improve readability and reduce the time necessary to find bad or damaging walking habits of the user.

Another feature that could be added to help our shoe compete is an interface with a phone or smartwatch. This could be used to provide immediate feedback to the user and could greatly reduce the time needed to correct gait and further prevent the possibility of foot ulcers.

To move this device to manufacturing would require additional planning with Sunrise shoes to determine where the sensors would be most effective and easiest to embed. This could require a slight redesign of either the shoe or the sensor placement. Additionally, easy access to debug, update, and change configuration files needs to be provided for quick manufacturing.

Improvements are definitely needed to provide a greater chance of success on the market. Currently we have the core components required to compete, functionality with reduced cost, but the device overall needs polish on every component.

XI. CONCLUSION

By implementing pressure sensing technology into a diabetic shoe and providing accurate data, we are addressing the problem associated with ulcers caused by uneven or undesired pressure on the foot. Extensive measures have been taken to ensure that the data obtained is accurate and that the data can be transferred off the shoe to be accessible by experts who can assess the patient's condition.

The design was split into manageable work tasks. Tasks fall into two subcategories, data acquisition and data transfer. These categories suited the separation of tasks among two teams, the data acquisition was assigned to the electrical engineering team and the data transfer was assigned to the computer engineering team. Each team has completed the main components of their respective tasks.

The task were further divided into work packages that each person could work on an these tasks were prioritized by criticality to the project as a whole. The time to complete each work package was estimated by the person with the most expertise and knowledge about the task. A total time was estimated and a timeline was created to keep the team and project on track; Each task is listed in a Gantt chart and needs to be complete by a certain time as indicated on the timeline. If the team sees that a task is taking longer than expected we can help each other complete the task or take mitigative actions if necessary. More critical sections were put in a higher priority and were worked on first in order to minimize risk.

Risk assessment determined what risks we faced as part of the project. The risks were analyzed and mitigation plans were crafted to minimize the impact if an issue were to occur. The probability and risk of each task was determined and an associated degree of risk was calculated. Doing so allowed us to see very quickly what tasks needed to be worked on early to reduce the risk of those riskier tasks early. We determined that the most critical sections were the most important and the most risky so we worked on the critical tasks and components first. As the project has progressed we have had to reevaluate the risk and importance of each task and the process of running a risk analysis on the whole project has helped the team see what the ranking of risk is among the tasks.

For our project the perceived critical path has been determined to be the anything that could have a detrimental effect on the project as a whole if the particular task fails. There are total five critical paths are identified: Real Time Data Capture to MC Storage path, Data Handling Unit (DHU) to Server Upload path, microSD to the DHU path, AC/DC converter to the rechargeable battery path, and DB software and visual graph software to the webpage path. These are the paths we focused on when making the laboratory prototype.

Right now our prototype can accurately gather accurate pressure data, store that data for later transfer, and transfer the data off the device and onto a server. This is the core of our design and contains the most critical and complex features to be implemented in the final prototype. We will continue to work on the the core to allow more data to be taken. More sensors will be added and collection algorithms will be optimized to allow more frequent acquisition of data points. Auxiliary features, such as wireless charging and data transfer, will be added in the coming months to meet our full feature set.

When the full feature set we have laid out is implemented the user should be able to use the shoe with little user interaction and minimal added complexity from the user's point of view. The user should be able to put on the shoe and benefit from the therapeutic properties of the shoe all while pressure data is being taken. When the user is done using the shoes they should be placed on the docking station and the shoe will automatically charge it’s batteries and transfer the data off the shoe and onto a server accessible by experts who can assess the user’s condition.

Our goal when starting the project was to use our knowledge and our ability to gather new knowledge and information to address a societal problem we face in the world today. Diabetics face a debilitating, even life-threatening problem when dealing with the neuropathy that can occur due to their condition and we saw the opportunity to help alleviate this problem. We hope that with our laboratory prototype, we are one step closer to a deployable prototype that can help, in some way, those who suffer from these dangerous conditions.

REFERENCES

[1] Lecture Slide Work Breakdown Structure, 2016

[2] Lecture Slide Work Breakdown Structure, 2016

[3] Lecture Slide Risk Assessment, 2016

[4] "Long-Term Trends in Diabetes," in CDC, 2016. [Online]. Available: https://www.cdc.gov/diabetes/statistics/slides/long_term_trends.pdf. Accessed: Sep. 12, 2016.

[5] B. McCall, "Huge Burden of Foot Ulcers Doubles Diabetes Costs in US," in Medscape, 2014. [Online]. Available: http://www.medscape.com/viewarticle/821908. Accessed: Sep. 12, 2016.

[6] B. J. Rice et al., "Burden of diabetic foot ulcers for medicare and private insurers," Clinical Care/Education/Nutrition/Psychosocial Research, vol. 37, no. 3, pp. 651–658, Mar. 2014. [Online]. Available: http://care.diabetesjournals.org/content/37/3/651. Accessed: Sep. 11, 2016.

[7] "Medical, Drug, and Work-Loss Costs of Diabetic Foot Ulcers," 2013. [Online]. Available: https://www.ispor.org/regional_chapters/Mexico/documents/SB2%20Presentation.pdf. Accessed: Sep. 12, 2016.

[8] H. Nyugen, "H-Fit Healing Shoe with Sensor Technologies,".

[9] C. Hicks, S. Selvarajah, R. Sherman, K. Hines, J. Black, and C. Abularrage, "Burden of Diabetic Foot Ulcers on Hospital Admissions and Costs," p. 10, Feb. 2016.

[10] C. Hicks, S. Selvarajah, R. Sherman, K. Hines, J. Black, and C. Abularrage, "Burden of Diabetic Foot Ulcers on Hospital Admissions and Costs," p. 10, Feb. 2016.

[11] C. Hicks, S. Selvarajah, R. Sherman, K. Hines, J. Black, and C. Abularrage, "Burden of Diabetic Foot Ulcers on Hospital Admissions and Costs," p. 10, Feb. 2016.

[12] B. J. Rice et al., "Burden of diabetic foot ulcers for medicare and private insurers," Clinical Care/Education/Nutrition/Psychosocial Research, vol. 37, no. 3, pp. 651–658, Mar. 2014.

[13] I. Kruse and S. Edelman, "Evaluation and treatment of diabetic foot ulcers," Departments, vol. 24, no. 2, pp. 91–93, Apr. 2006. [Online]. Available: http://clinical.diabetesjournals.org/content/24/2/91. Accessed: Sep. 11, 2016.

[14] "Top 5 causes of non-healing diabetic foot ulcers (and how to prevent them)," in Unity Point, www.unitypoint.org/livewell/, 2014. [Online]. Available: https://www.unitypoint.org/livewell/article.aspx?id=7a88ede4-3631-4063-a4ba-cd0b98b738ca. Accessed: Sep. 11, 2016.

[15] S. Insole, "Tag: Diabetic foot ulcer," 2016. [Online]. Available: https://footpressurediabeticfootulcersmartinsole.wordpress.com/tag/diabetic-foot-ulcer/. Accessed: Sep. 12, 2016.

[16] C. Hicks, S. Selvarajah, R. Sherman, K. Hines, J. Black, and C. Abularrage, "Burden of Diabetic Foot Ulcers on Hospital Admissions and Costs," p. 10, Feb. 2016.

[17] C. Hicks, S. Selvarajah, R. Sherman, K. Hines, J. Black, and C. Abularrage, "Burden of

Diabetic Foot Ulcers on Hospital Admissions and Costs," p. 10, Feb. 2016.

[18] Drugs, "Foot ulcers guide: Causes, symptoms and treatment options," 2000. [Online]. Available: https://www.drugs.com/health-guide/foot-ulcers.html. Accessed: Sep. 12, 2016.

[19] [Online]. Available: https://www.researchgate.net/publication/23482085_Mortality_Rates_and_Diabetic_Foot_Ulcers_Is_it_Time_to_Communicate_Mortality_Risk_to_Patients_with_Diabetic_Foot_Ulceration. Accessed: Sep. 11, 2016.

[20] L. Esposito, "Health Buzz: Many Patient lie to their doctor, survey finds," in U.S. News, 2015. [Online]. Available: http://health.usnews.com/health-news/health-wellness/articles/2015/11/10/many-patients-lie-to-their-doctors-survey-finds. Accessed: Sep. 11, 2016.

[21] C. Hicks, S. Selvarajah, R. Sherman, K. Hines, J. Black, and C. Abularrage, "Burden of Diabetic Foot Ulcers on Hospital Admissions and Costs," p. 10, Feb. 2016.

[22] American Diabetes Association. Consensus Development Conference on Diabetic Foot Wound Care, Boston, Massachusetts. American Diabetes Association. Diabetes Care. 1999;22(8):135–1360.

[23] M. S. Bader, "Diabetic foot infection," in AAFP, 2008. [Online]. Available: http://www.aafp.org/afp/2008/0701/p71.html. Accessed: Sep. 11, 2016.

[24] "The Primary Role of Gait Restoration,".

[25] B. McCall, "Huge Burden of Foot Ulcers Doubles Diabetes Costs in US," in Medscape, 2014. [Online]. Available: http://www.medscape.com/viewarticle/821908. Accessed: Sep. 12, 2016.

[26] Drugs, "Foot ulcers guide: Causes, symptoms and treatment options," 2000. [Online]. Available: https://www.drugs.com/health-guide/foot-ulcers.html. Accessed: Sep. 12, 2016.

[27] S. Insole, "Tag: Diabetic foot ulcer," 2016. [Online]. Available: https://footpressurediabeticfootulcersmartinsole.wordpress.com/tag/diabetic-foot-ulcer/. Accessed: Sep. 12, 2016

[28] S. Raju, K. Hollis, and P. Neglen, "Use of compression stockings in chronic Venous disease: Patient compliance and efficacy," Annals of Vascular Surgery, vol. 21, no. 6, pp. 790–795, Nov. 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0890509607002865. Accessed: Sep. 12, 2016.

[29] A. H. P. Paes, A. Bakker, and C. J. Soe-Agnie, "Impact of dosage frequency on patient compliance," Original Articles, vol. 20, no. 10, pp. 1512–1517, Oct. 1997. [Online]. Available: http://care.diabetesjournals.org/content/20/10/1512. Accessed: Sep. 12, 2016.

[30] K. Arango, Figure 6 - "Design overview flow chart."

[31] PowerbyProxi, "Wireless charging & how Inductive Chargers work • PowerbyProxi," PowerbyProxi, 2016. [Online]. Available: http://powerbyproxi.com/wireless-charging/. Accessed: Oct. 3, 2016.

[32] K. Nguyen, Figure 7 -"Docking Station Diagram."

[33] K. Nguyen, Figure 8 - "User Interface Diagram."

[34] P. Patel, TABLE I - "Internal Funding Proposal"

[35] P. Patel, Figure 9. "Milestones Chart"

[36] K. Nguyen. Figure 10. "Work Breakdown Structure"

[37] K. Nguyen, P. Patel, Table II "Task Descriptions"

[38] K. Nguyen, Table III - "Hours per team member for Fall 2016 and Spring 2017"

[39] K. Nguyen. Figure 11 - "PERT Chart"

[40] W. Sam. Figure 12 - "Risk Assessment Chart"

[41] Parallax. (2014, May). "Propeller Board of Education" [Online]. Available:

https://www.parallax.com/sites/default/files/downloads/32900-Propeller-BOE-Documentation-v1.1_0.pdf. Accessed: Oct. 26, 2016
[42] "FlexiForce A301 Sensor." Tekscan. N.p., 2016. Web.[Online]. Available: https://www.tekscan.com/products-solutions/force-sensors/a301. Accessed: Oct. 31,2016
[43] "RN4677 Bluetooth 4.0 Dual Mode Module Data Sheet." N.p., n.d. 2016.. Available: http://www.mouser.com/ds/2/268/50002370A-587936.pdf. Accessed: Oct 31,2016
[44] K. Arango. Figure 13 - Top Down Design Overview
[45] Tekscan, Figure 14. Pressure Sensor and op-amp. https://www.tekscan.com/products-solutions/force-sensors/a301
[46] Microchip, Figure 14. Pressure Sensor and op-amphttp://www.microchip.com/wwwproducts/en/MCP6004
[47] Parallax and Adafruit . Figure 15. Propeller and Feather 32ux Board. https://www.parallax.com/product/32912. https://www.adafruit.com/product/2795
[48] Gregory Thompson. Figure 16 - Common foot ulcers areas http://www.webmd.com/diabetes/locations-of-foot-ulcers
[49] A.N. Nithyaa. “A Real Time Foot Pressure Measurement for Early Detection of Ulcer Formation in Diabetics Patients Using Labview”
[50] Parallax. Figure 17. “Serial Port on Propeller Activity Board “
[51] Sparkfun. Figure 18. “Bluetooth RN42 module”
[52] K.Arango . Figure 19. “Dataflow Overview”
[53] K.Arango . Figure 20. “DHU Overview”
[54] K.Arango . Figure 21. “Serial Collection and File Writing”
[54] K.Nguyen . Figure 22. “Sample of Data in Graph”
[55] Team 10 . Table IV. “Test Plan Result”
[56] Orpyx Medical Technologies Inc.. Figure 23.. “SurroSense Rx, competing product”. https://orpyx.com/pages/surrosense-rx
[57] K.Nguyen. Figure 23. “Shoe on Dock”
[58] P. Patel. Figure 24 - “Block diagram of hardware”
[59] P. Patel. Figure 25 - “Device Weight Measurement using Spring ”
[60] P. Patel. Figure 26 - “ Weight Measurement using Scale”
[61] Tekscan Inc - Figure 27 “ Tekscan FlexiForce A301”. https://www.tekscan.com/
[62] Microchip Inc - Figure 28“ MCP6004 Op-Amp”. http://www.microchip.com/wwwproducts/en/MCP6004
[63] P. Patel - Figure 29 “ Inverting op-amp configuration”.
[64] P. Patel - Figure 30 “ Voltage divider with unity buffer”.
[65] P. Patel - Equations (1), (2), (3), (4)
[66] P. Patel - Figure 31 “Analog to digital conversion”.
[67] P. Patel - Figure 32 “Data storage path”.
[68] P. Patel - Figure 33 “MPU-9250”.
[69] P. Patel - Figure 34 “MPU - Feather Pins Connection”.
[70] P. Patel - Figure 35 “Bluetooth - Feather Pins Connection”.
[71] P. Patel - Figure 36 “Wireless charging circuit”.
[72] P. Patel - Figure 37 “DHU power circuit”.
[73] P. Patel - Figure 38 - “Block level diagram for the AC/DC converter”.
[74] Team 10 - Table V- “Hardware Test Plan Results”.
[75] K. Nguyen- Figure 39 “Flowchart of the MC code”.
[76] K. Nguyen- Figure 40 “Configuration of SSH”.
[77] K. Nguyen- Figure 41 “Database Schema”.
[78] K. Nguyen- Figure 42 “Database App interface”.
[79] K. Nguyen- Table VI “Software Test Plan Results”.

[80] P. Patel - Figure 43 “Bottom Shoe Sensors Assembly”.
[81] P. Patel - Figure 44 “Shoe Wires Assembly”.
[82] P. Patel - Figure 45 “Finish Shoe Assembly”.
[83] P. Patel - Figure 46 “Protoboard by Ardafruit”.
[84] P. Patel - Figure 47 “Docking top view”.
[85] P. Patel - Figure 48 “Force vs Voltage of Top Sensor”.
[86] P. Patel - Figure 49 “Force vs Voltage of Mid Sensor”.
[87] P. Patel - Figure 50 “Force vs Voltage of Heel Sensor”.

GLOSSARY

AC: Alternating Current. We will encounter this in the form of conventional home outlets.

AC to DC Converter: A circuit designed to change AC to DC.

Bit: Smallest unit of digital information. Represented as either a ‘0’ or a ‘1’.

Charging Station: A device that facilitates charging of the product. Ours will also facilitate data transfer.

Data: Collection of information. In this context, information received from sensors embedded in the shoe.

Database: A computer used for information storage, accessible via network.

DC: Direct Current. Our devices will mainly use this for power and signals.

Device: The therapeutic shoe with all of its electronic components

Docking Station: A device that facilitates wireless charging and data transfer. Contains and AC to DC converter and the DHU. See Charging Station.

DHU: Data Handling Unit. Unit located in the docking station that transfers and processes data. See also Raspberry Pi.

Gait Cycle: Walking cycle: Heel Strike, Foot Flat, Midstance, Heel Off, Toe Off, Mid- Swing, Repeat.

Kbps: KiloBit Per Second - Information transfer rate.

Microcontroller: A small, low power computer used to control devices.

Pressure mapping: Data collected from pressure sensors reformatted easily readable.

Product: See Device.

Raspberry Pi: A small low cost computer. Used in our docking station to process and transfer data. See also DHU.

Sunrise Shoes: Company the develops pedorthic shoes and diabetics shoes. Interested in augmenting their current design with technology. Our sponsor.

Appendix A: User's Manual

Instrumentation of Diabetic Shoe

April, 2017 by Team 10

1. System Overview

The shoe is designed to be as intuitive and unintrusive as possible to use. That said, there are still some things to be aware of. There are two parts to the device as a whole: the docking station, and the shoe. See Figure 23.[57]

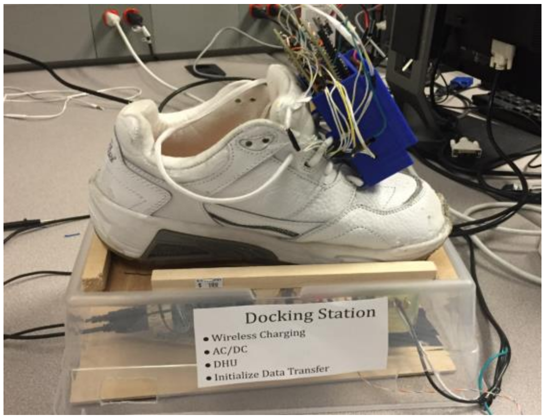

Figure 23 - Shoe on Dock [23]

The shoe can be used like any other conventional shoe, slip the shoe on your feet, tie up the laces, and go. There is however a plastic encasing that the laces are looped through. This encasing houses all the electronic components necessary to log data from the shoe and care should be taken to ensure that it is securely attached to the laces. If the encasing is not firmly attached to the laces, the device should not be used because it could detach and cause harm to the user and damage the internal components of the shoe.

The docking station is where the shoes should be placed when not in use to charge its battery and offload the logged data. The docking station does require an initial setup. We recommend you place it on a relatively flat surface where it can be plugged into the wall. It is designed to be plug and play; plug in the power cord and plug in an ethernet cable for internet connection. Internet connection is needed to communicate with the server and upload logged data. In order to establish a wireless connection you will have to perform a one time setup process. To establish the wireless connection plug in a monitor, keyboard and mouse to the docking station connect to the desired wireless access point. If the monitor does not display anything then place the shoes onto the docking station.

After using the shoe place it onto the docking station within the outlined area. The light LED on the corner of the docking station should light up after a few seconds, if this does not happen then remove the shoe and try again while being sure that the heel of the shoe depresses the switch on top of the dock.

2. Getting Started

2.1. Installation

The docking station should be placed in place with room temperature. It should be kept out of reach of children and pet. For proper use of the product, an ethernet cable with an internet connection is needed for connection to the internet. Plug the power cable into a power the electronics, a green light will turn on when the device is powered. Plug an ethernet cable into the ethernet port after the device has been powered, blinking lights on the ethernet port will indicate internet connection.

3. Operation

3.1. Power on

The devices in the shoe will always be powered on. Indicated by a green led on the Feather board.

3.2. Collecting

Data collection will begin after 20 secs delay as the shoe is taken off the docking station.

3.3. Charging

Charging will begin as soon as shoe is placed on the docking station and is properly aligned with the guides, indicated by a light.

3.4. Data Offloading

Data off loading will begin as soon soon as shoe is placed on the docking station and is properly aligned with the guides Data transfer indicated by a red light on the Docking station.

Data transfer completion is indicated by a blinking red led on the Feather board.

4. Software

4.1. Database

4.1.1. Run application

4.1.2. Enter login credentials

4.1.3. Manipulate data by using search/edit/delete button

4.2. Graph Application

4.2.1. Open graph application within the Database application

4.2.2. Select individual graphs or all then click on OK to generate graph

Appendix B
Hardware

There hardware can be broken down into categories. The first category is the sensors. The sensor hardware encompass the FSR's, the IMU"s, the MC's, op-amps, and bluetooth transmitter were used to for data collection and transmission. The next category is power. For power, there was an AC/DC that was designed, wireless charging and the battery used.

The sensors can be represented in block diagram form as shown in the figure 24. [58]

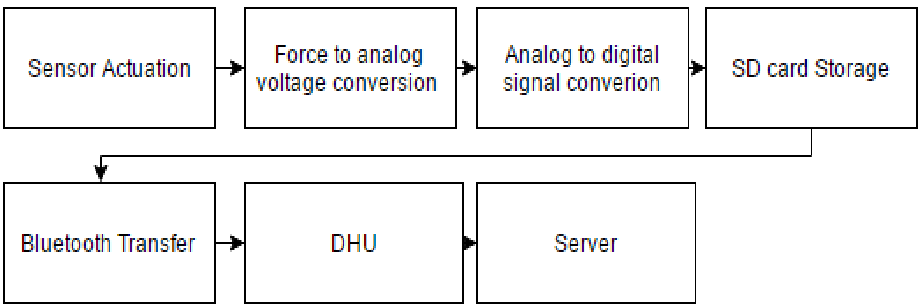

Fig. 24 - Block diagram of hardware [58]

The actuation refers to how the sensor will be loaded by a force. To have consistent and accurate readings at the output of the sensor circuit, the means to which to load the sensor for each measurement must be consistent in between each measurement.One thing to avoid while loading the sensors is to avoid any curve in the sensor area. This means the sensors need to be mounted against a flat hard surface. The base of the shoe we are using is a hard-enough surface for us that bending will not be an issue.

There are a few actuation methods that we have used to attain various results of a force vs output voltage relationship. This relationship is not only dependent on the actuation method but circuitry used. The gain can be different values for the same weight measurement.

The first actuation method was a device built by ourselves that involved pulling down on a spring to measure the weight on the sensor. The spring hangs on a string, the string pushes down on a block that pushes down on the sensing area of the FSR when the spring is pulled down. The stretch of the yields how much force was placed on the sensor. Figure 25 [59] shows the setup that was mentioned.

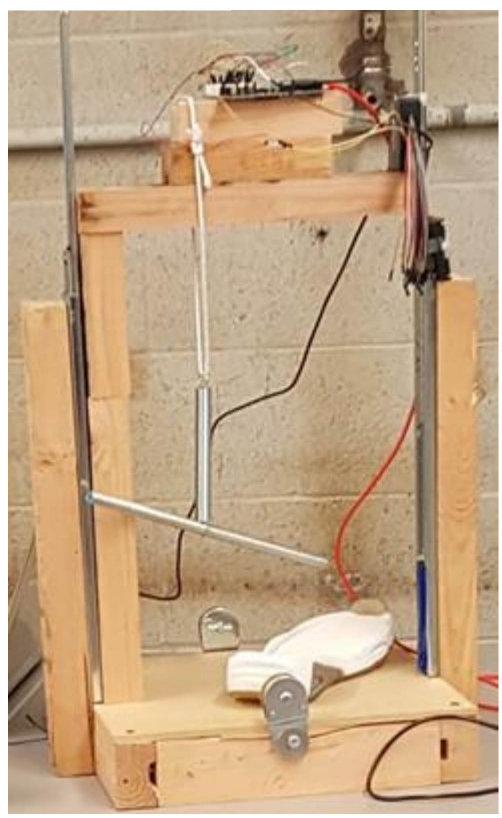

Fig. 25a - Weight Measurement- Spring [25]

The raspberry pi is used as the data handling unit. The data on the microSD is transferred through Bluetooth connection to the raspberry pi. During testing, it was found that the raspberry pi consumes steadily about 400mA during data transfer.

The raspberry pi is powered by the AC/DC converter which supplies 5V and about 1A of current. The AC/DC converter consists of a step-down transformer of 120VAC to 24VAC, then goes into a full wave rectifier made of 1n4004 diodes. The voltage drop across the full wave rectifier is about 0.8V (2Vmax/pi). The supply then goes into a voltage regulator (LM2596) which steps is down to 5V and includes a filter capacitor to get rid of any ripple at the output of the full wave rectifier. During testing, we found that we needed diodes that could handle 1A of current which is why we chose the 1N4004.

To recharge our lithium ion battery, we are using a wireless charger consists of 2 inductor coils; receiving coil is 14µH and delivery coil is 3.7µH. The wireless charger charges the battery at a rate of about 70mA per hour. The consumption of our battery is about 200mAh. We only have the system in the shoe running for two hours so the expected battery consumption throughout the day is 400mA which is within the limit of the 2500mAh rating of our battery. Based on our tests, the battery will go about 3 hours before there is any drop-in voltage. At 6 hours, the voltage will drop low enough to where the battery will stop current flow and turn off the microcontroller. The graph can be seen in the figure below. [59]

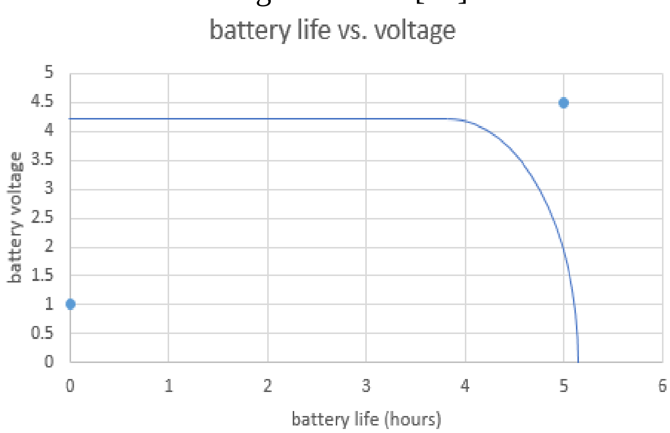

Fig 25b Battery life graph [59]

This method provides consistent actuation of the sensor. The downside of this method comes from the limitation of the spring. The spring has a load limit and is only accurate up to the specified limit on the product. The next is the length limit which can limit how much force can be put on with certain springs.

The next 2 methods that were used to gather data was done with a mechanical step on scale. With this, we have a better measurement of the weight being put on the sensor. Figure 26[60] shows the scale that was used.

Figure 26 - Weight Measurement - Scale [60]

Hardware Test Results

Even though we can get a better reading of the weight, and have a larger range of weights to apply, the actuation method is inconsistent.

The actuator for this method was my foot. We can get linear relationship curves, but due to shaking and human error in measurement reading, the curves are not as accurate as they can be.

Another reason why there are errors in this type of actuation method is that weight distribution across the sensing area changes giving slight variations in the output reading.

The next item on the block is the force to voltage conversion. The technology used for the measure force is an FSR, or force sensitive resistor. The model we are using is a Tekscan FlexiForce A301. As force is applied to the small sensing area, the resistance of the material is decreased allowing current flow through the FSR. The resistance of the material is inversely proportional to the force applied, and directly proportional to the output voltage, maintaining a linear relationship. Figure 27 below shows the sensor that was used. [61]

Fig. 27 Tekscan FlexiForce A301 FSR [61]

The op-amp that will be used is an MCP6004 linear, low-power op-amp, shown in figure 28. [62]

Fig. 28 MCP6004 Op-Amp *[62]*

This device was specifically called out by the documentation of FSR.

For the project, there are 4 total sensors that will be placed into the shoe to help monitor the pressure profile of the patient's foot. These sensors will be placed in areas of the shoe that are known to help develop foot ulcers.

To determine how much pressure is being applied to the FSR, a Force vs Voltage graphs needs to be developed. To accomplish this, additional electronics are needed to help translate the force signal into a voltage signal.

There were several options we considered for convert the force signal to a voltage signal. The first option that we tried was the circuit recommended circuit, which was placing the FSR in inverting op-amp configuration. A schematic of the configuration is shown in figure 29.[63] For future diagrams, MCP6004 will be replaced by U1.

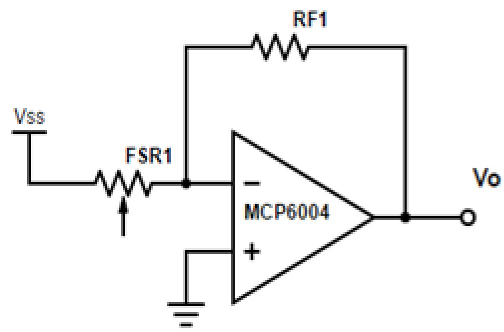

Fig. 29 Inverting op-amp configuration. [63] FSR1 is the force sensitive resistor and RF1 is the reference resistor.

he gain of this circuit can be characterized by the equation below:

$$\frac{V_o}{Vss} = -\frac{RF1}{FSR1} \qquad (1)$$

Therefore, the equation for the output voltage can be written as.

$$V_o = -\frac{RF1}{FSR1} V_{ss} \qquad (2)$$

For this configuration, is set to -1 V, however it can have range of -0.5 V to -1.0 V. One of the first problems with this configuration is that if the FSR were to break down or a short is created from to, all the current would get dumped into the microcontroller that we are using to log and transfer the data, essentially damaging the ADC pins of the microcontroller. This scenario is a likely because as mentioned previously, a higher force on the FSR yields a lower resistance in the FSR.

If we wanted to keep using this circuit, we would have had to use a unity gain buffer attached to the output of the inverting stage. This would have prevented current flowing into the MC, which would be reading the voltage at the output of the unity gain stage; however, this would have increased the number of components we needed to use, which we are trying to minimize.

The next aspect of this circuit that we did not like was the use of a negative voltage supply. We could use an inverter for get a negative power supply to the sensors, but as mentioned before, we wanted another solution that would help minimize the components.

After further research into better ways to get force information out of FSRs, the circuit in figure 30 is shown below. [64] It is a voltage divider with a unity buffer whose output will be measured by a MC

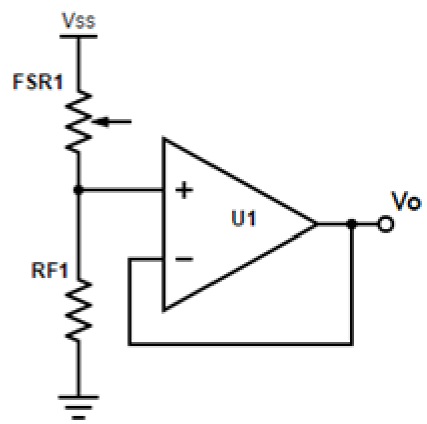

Fig. 30 Voltage divider with unity buffer. U1 is an MCP6004 op-amp.[64]

The gain and output for this configuration are described by equations 3 & 4 below.

$$\frac{V_O}{V_{SS}} = \frac{RF1}{RF1+FSR1} \qquad (3)$$

$$V_o = \frac{RF1}{RF1+FSR1} V_{ss} \qquad (4)$$

The unity buffer gain here is used to stop any current flow to the MC when the FSR1 resistance gets low. The resistor RF1 performs two functions. The first is that it will help set the force range of the sensor. A larger value on RF1(> 100K Ohms) yields a larger gain, which means the output will rail to faster, capturing only a short range of force. The second function of RF1 is to limit the current draw from the power source. The resistance in the FSR can reach low values (<100 Ohms), therefore RF1 must have a value of 1K Ohms or greater to limit the current to 5 mA with a 5 V battery.

The benefits of this circuit are that it uses very little components, and helps protect MC from anything that may happen on the other side of the buffer.

The output of this force to voltage conversion circuit is a voltage signal with an output range of 0-3.3V. To be able to determine the force outputted, calibration needs to be done. The value of RF1 chosen for this configuration is 450 kohms. A large value was chosen to detect small changes in pressure. The calibration process will be explained in its own appendix.

The next block in the diagram is the analog to digital conversion. Our MC supports an input from 0-3.3V. To get the proper output reading, the raw data polled by the MC's ADC must be divided by 1024 as the 3.3V are spread across 10 bits.

The next blocks in figure 32[65] for SDcard storage, Bluetooth transfer to DHU and then from there to server had device drivers written for them and will be explained in the software section.

For the gyroscope, the block level diagram follow nearly an identical path. The gyroscope acceleration combo that was used was

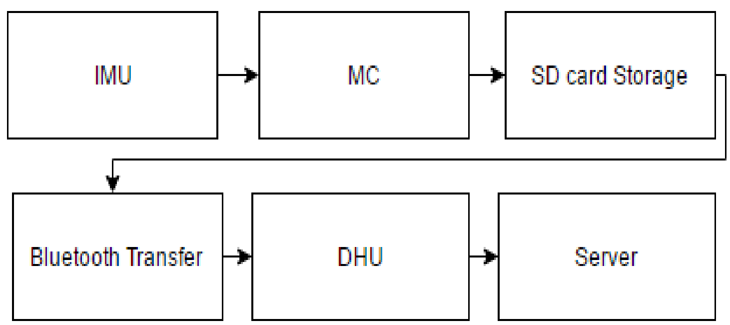

Figure 31 - Data storage path [65]

an MPU-9250. A figure 33 of it is shown below. [66]

Fig. 32 MPU-9250 gyroscope-acceleration-magnetometer combo. [65]

The SLC, and SDA pins are hooked to the MC for I2C communication.

The IMU is hooked up to the MC as shown in the figure 34.[67]

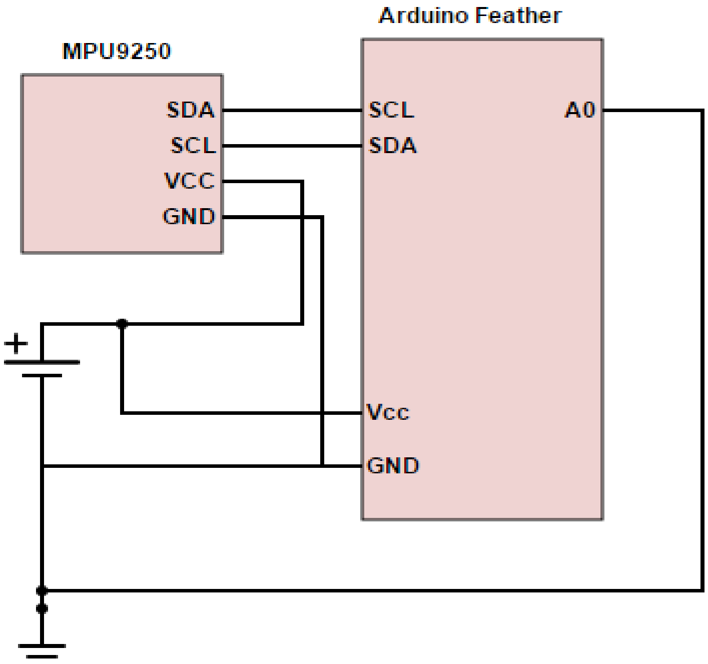

Figure 33 - MPU - Feather Pins Connection [66]

The SDA pin transfers data from the registers while the SCL syncs the clocks of the devices. For the Arduino Feather 32uA MC unit, the A0 pin has to be grounded in order for proper operation of the IMU. The device is powered by a 3.3V LiPo battery.

The code for the gyro was formed by using the Wire.h library on the MC which allows I2C communication between the MPU and MC.

The unit is configured by writing 7-bit binary numbers onto registers of the MPU. There are registers that govern the power management options and there are options to control the gyroscope accelerometer options.

To get the actual accelerations and scope values, the correct registers must be read and processed based on the settings of the gyroscope and accelerometer.

With the accelerometer values yielding correct acceleration information yaw, pitch, and roll can be calculated based on the values read. The commented code for the gyroscope is in the appendix.

The bluetooth circuit is shown in the figure 35 below. [68]

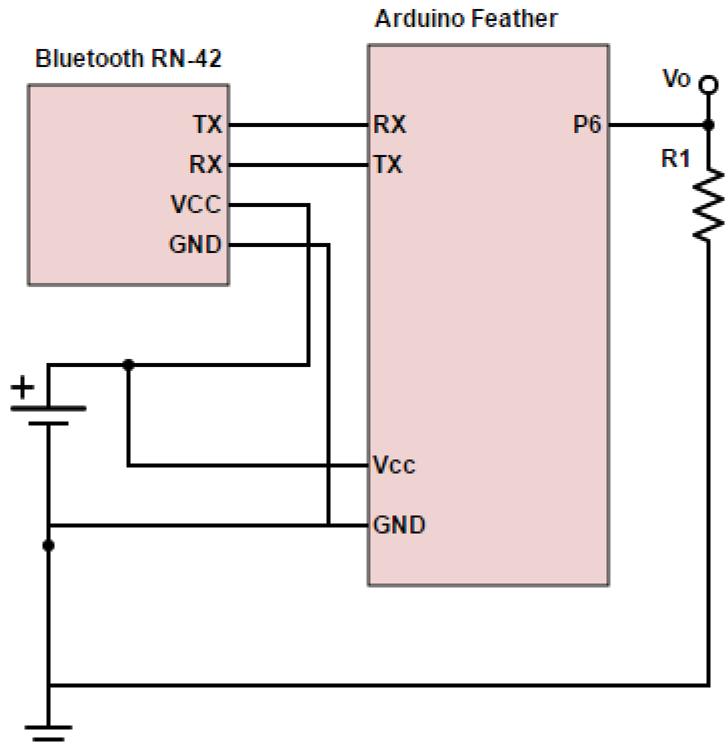

Figure 34 - Bluetooth - Feather Connection [67]

For this circuit the bluetooth communicates to the feather through the RX and TX pins.. The general I/O pin 6 waits for Vo to go high, then it will initiate data transfer to the DUH. THe resistor R1 is a pulldown resistor with a value of 5 kohms. The Vo signal comes from the wireless charging circuit shown in the figure 36. [69].

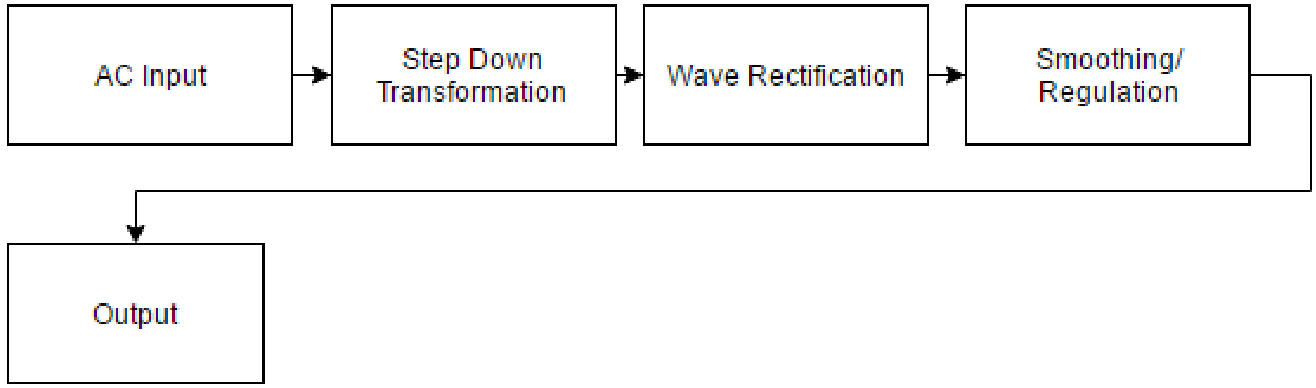

Figure 37 - Block level diagram for the AC/DC converter [70]

The signal Vo is read through a buffer so that it will not siphon any current from the battery, Vbb is the voltage of the battery. Vo will go high when the wireless receiver coil in the shoe and charger coil in the docking station are aligned, which signifies that the shoe is placed on the station to start data transfer. Whe n the shoe is placed on the station a motion sensor trips the power to the DHU, initiating its data transfer to the server. A circuit for this figure is 37. [70]

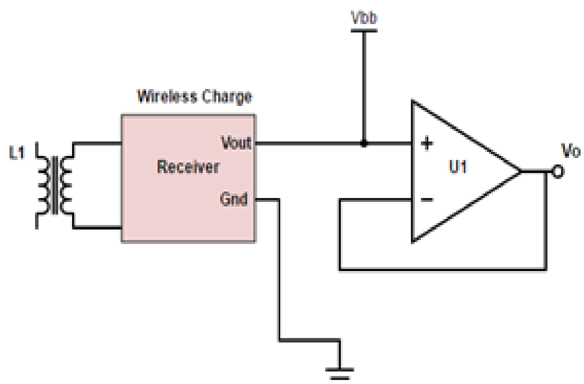

Figure 35 - Wireless charging circuit [68]

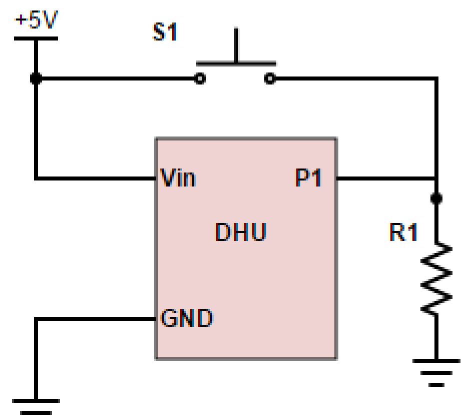

Figure 36 - DHU power circuit [69]

The DHU receives power from the AC/DC converter as well as the switch. Once the switch detects motion it will close, resetting the pi so the automation script can start.

The block level diagram for the AC/DC converter is shown 38. [71]

Test Plan Result

There are various components that were tested before being implemented into the the system. The sensing portion of the hardware has its own set of tests and the AC/DC portion has its own set of tests.

For the pressure sensor the amplification circuit needs to bested and made sure the detected voltages are within 5% of the voltmeter value. This will be test P01. Test P02 will be to make sure the sensors are in good operating condition. To check this done through calculations and measurements. Test P03 will test sensor placement. Once the sensor are in the shoe, we need to make sure the signals are getting through all the solder and connected wires that run in between the shoe to the MC. Test P03 will also check other conditions of sensor operation. Test P04 will test the accelerometer. The accelerometer will be tilted in various directions to make sure the force of gravity is distributed across the axis.

Test plan result for hardware shown in Table V.[74]

TABLE V
Hardware Test Plan Result

Test ID	Description	Expected results	Actual results	Pass/ Fail	Additional comments
P01	**Amplification circuit** -check pressure value detected by microcontroller -compare microcontroller pressure value to actual value	-pressure values detected by microcontroller are within 5% of actual values	values compared with spring loads; 5 tests were done, within 4%	Pass	acceptable amount of noise at output
P02	**Pressure Sensors** -check whether resistance across sensor fluctuates as different amounts of pressure are applied	-resistance of the A301 sensor changes as force is applied	resistance fluctuates inversely to amount of force applied	Pass	
P03	**Sensor placement** -verify sensors are placed in the shoe so that wires don't come loose while user is in motion -verify that sensors are being placed on parts of the foot that apply the most force on the ground -verify sensors work in the shoe	-sensor connection remains secure -sensors are placed in high pressure points	sensor is still able to collect data when placed under foot	Pass	unaffected by heat when foot placed inside shoe
P04	**Gyroscope** -place gyroscope on surface to simulate a shoe and rotate the surface and observe the changes in angle values	-gyroscope angle values displayed are within 2 degrees of actual angle	able to sense change in axis, values are within 2 degrees	Pass	

		values			
S01	**AC/DC Converter** -apply load to AC/DC converter that is equivalent to all devices in docking station -measure current leaving the AC/DC converter -verify voltage is large enough to supply other devices in docking station	-AC/DC converter outputs regulated 5V - AC/DC provides enough current to drive all the electronics	able to continuously supply 1A of current to resistive load	Pass	devices in docking station won't take more than 750mA so tested for 1A.
S02	**Wireless charging circuit** -delivery coil supplies current to receiver coil connected to a load -battery receives current while within range and recharges	-current is received by load	Output of wireless charging circuit is kept at 5V so current is based on load. Max current at 400mA with battery	Pass	coils have to be aligned perfectly to achieve max current
S03	**Bluetooth circuit** - Verify that Bluetooth module connected to Propeller works properly - Connect propeller with different Bluetooth devices such as : Pi, iPhone, android etc.	**-** Before pairing the Bluetooth module with the DHU it should be able to connect and be detectable by other devices.	Paired module with raspberry pi and able to transfer data	Pass	can also connect to android devices and display data

Appendix C
Software

1. MCU

1.1. Flowchart

Figure 39 shows the flowchart of the MCU code.[74] It includes 2 states running in an infinite loop: collecting and sending state.

For collecting state, it captures data from the pressure sensors and stores them in the microSD card. This state will continue to run until it receives the signal HIGH from the button (for Fall 2016) or signal HIGH from the wireless charging(for Spring 2017) to go to the next state – sending state. Once receiving signal HIGH, the sending state will start. The MC will read all the data stored in the microSD and send them over the Raspberry Pi via the serial port (Fall 2016) or Bluetooth (Spring 2017).

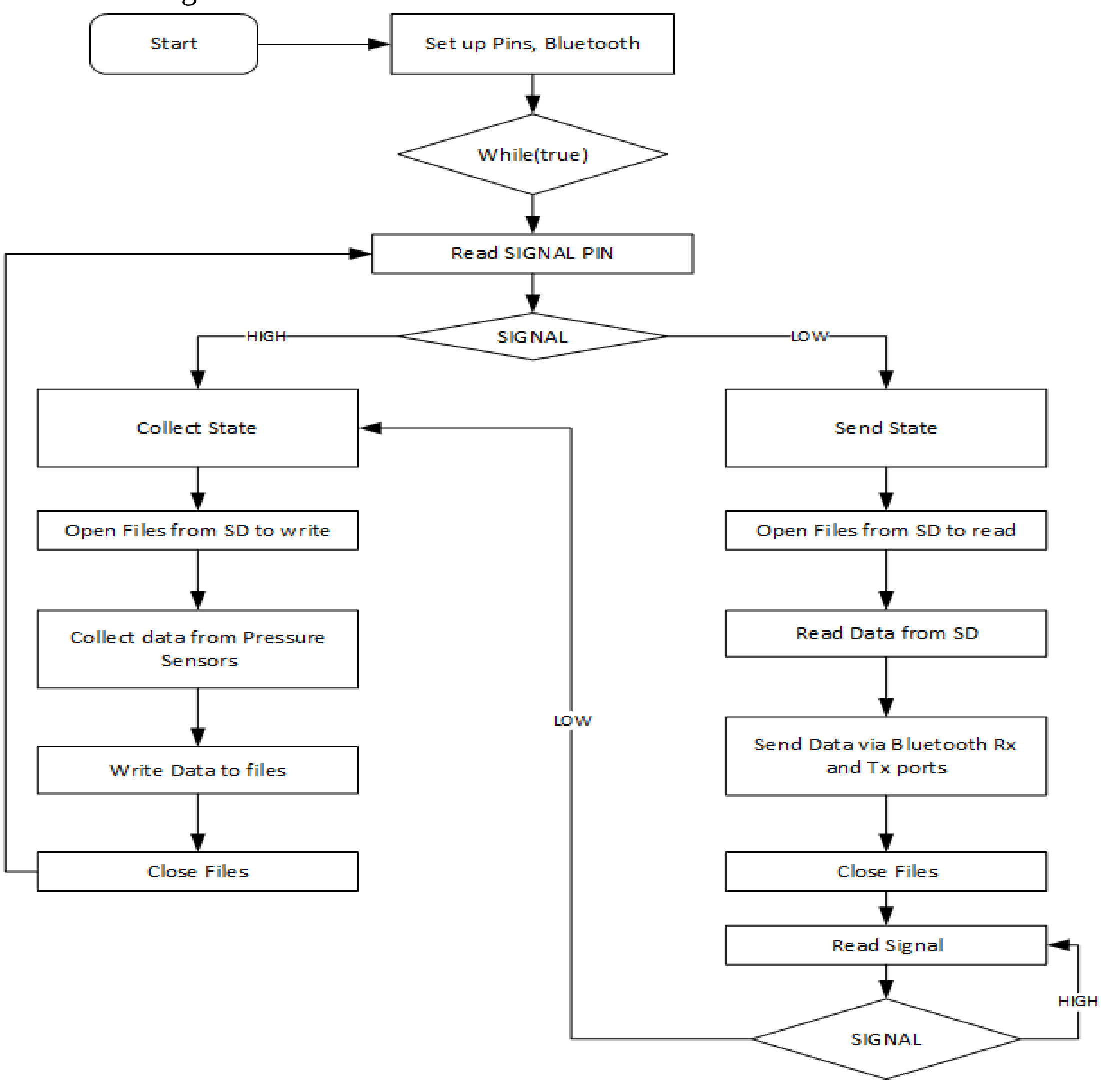

Figure 38 - Flowchart of MC code [71]

1.2. MCU C Code

```
//Created by: Khoi Nguyen
//Revised: 04-13-2017
//Last edited: 04-20-2017 By Kyle Arango

#include <SD.h>
#include <SPI.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#define signalPin 5
#define collectLed 8
#define sendLed 13
#define FILE_NO 2
#define rxPin 0
#define txPin 1
SoftwareSerial BTSetup (rxPin, txPin);

//Global variables

///SD card varibles///
int time_index=0;
int signalState=0;  //use to signal Sending State
File fp[FILE_NO];      // Create 7 files in SD
char filename[10];  // dynamic files name

int fileIndex = 0; //increment and open new file once the file is
too large

///Gyrop variables///
long accelX, accelY, accelZ; // Accel register values
float gForceX, gForceY, gForceZ; // Accel in gforces
long gyroX, gyroY, gyroZ; // Gyro register values
float rotX, rotY, rotZ; // Gyro in angular velocity


void setup()
{
    pinMode(signalPin, INPUT);  // Pin to signal the Send State
    pinMode(collectLed, OUTPUT); //collectLed pin
    pinMode(sendLed, OUTPUT); //sendLed pin
    BTSetup.begin(115200);

    //Create 8 new file on Set up, otherwise it will be halted !!!
    for (int i = 0; i <FILE_NO; i++)
  {
    sprintf(filename, "test%d_%d.txt", i, fileIndex);
    fp[i] = SD.open(filename,O_CREAT);
  }
```

```
  closeFile();

  //while (!Serial){;} // wait for serial port to connect, for
testing

  // Mounting SD card
    if (!SD.begin(4))
  {
    Serial.println("initialization failed!");
    return;
  }

  //Gyro setup///
  Wire.begin();
  setupMPU();

}

void loop()
{

    char buff[50];       // buff for sprintf format
    int intVoltage[6];  // voltage in int
    float voltage[6]; // voltage in float
    char str_voltage[6]; // convert float for sprintf
    int A[] = {A1, A2, A3, A4, A5, A7};    // analog pins


  digitalWrite(collectLed, HIGH);
  digitalWrite(sendLed, LOW);


  //////////////////////*****COLLECT STATE****** ///////////////

     // open the file for write.
    for (int i = 0; i <FILE_NO; i++)
  {
    sprintf(filename, "test%d_%d.txt", i, fileIndex);
    fp[i] = SD.open(filename,FILE_WRITE);
  }


  // if the files opened okay, write to it:
  if (fp[0] && fp[1] )
  {
    //Read data from 4 analogs Pin A1-A4
    for (int i = 0; i < 6; i++)
    {
```

```
      intVoltage[i] = analogRead(A[i]);     //read from A1-A4
      voltage[i] = intVoltage[i] * (3.3 / 1024.0); // Convert the
analog reading
      dtostrf(voltage[i], 4, 5, str_voltage);    //convert float
voltage to string
      //sprintf(buff,"%d , %s\n",time_index, intVoltage[i]);
//format string to buff, to print to file
      //fp[i].print(buff);              //Write to file in SD
    fp[0].print("File");fp[0].print(i); fp[0].print(", ");
      fp[0].print(time_index); fp[0].print(", ");
fp[0].print(str_voltage); fp[0].print("\n");
    }//end for

    time_index++;


  //Read Accel and print to File5
  recordAccelRegisters();

  fp[1].print("File6"); fp[1].print(", ");
  fp[1].print(gForceX); fp[1].print(", ");
  fp[1].print(gForceY);fp[1].print(", ");
  fp[1].print(gForceZ);
  fp[1].print("\n");


    // Close all files every write
    if(fp[0].size() > 65536)
    {
      fileIndex++;
      closeFile();
      for (int i = 0; i <FILE_NO; i++)
      {
        sprintf(filename, "test%d_%d.txt", i, fileIndex);
        Serial.print("File exceeded max size, opening new file: ");
        Serial.println(filename);
        fp[i] = SD.open(filename,O_CREAT);
      }
    }
    closeFile();
    //delay(1000);

  }
  else
  {
    // if the file didn't open, print an error:
    Serial.println("error opening test.txt");
  }
```

```
  //delay(100);

  //////////////////////*****SENDING STATE****** ///////////////
  signalState = digitalRead(signalPin);
  if(signalState==1)  //go to send state if signal =1
  {
    //closeFile();
    digitalWrite(collectLed, HIGH);
    digitalWrite(sendLed, HIGH);
    delay(55000); //use this for sync with the Pi (restart)

    digitalWrite(collectLed, LOW);
    // Open each file and print to terminal/Bluetooth
    for (int i = 0; i <FILE_NO; i++)
      {
        for(int j = 0; j <= fileIndex ; j++)
        {
          int d=i+1;
          sprintf(filename, "test%d_%d.txt", i, j);
          fp[i] = SD.open(filename,FILE_READ);

          while (fp[i].available())
          {
            BTSetup.write(fp[i].read());
          }
          fp[i].close();
        }
      }

    BTSetup.println("Done");
    BTSetup.println("Done");
    BTSetup.println("Done");BTSetup.println("Done");
    //Remove files after read
    for (int i = 0; i <FILE_NO; i++)
      {
        for(int j = 0; j <=fileIndex; j++)
        {
          sprintf(filename, "test%d_%d.txt", i, j);
          SD.remove(filename);
        }
      }

     fileIndex = 0;
    time_index = 0; //reset time_index
    //delay(2000);

    //After upload, signal pin ==1, or the shoes is on the docking,
it will stay in this state, not collecting
    while(1)
```

```
    {
      digitalWrite(sendLed, HIGH);
      delay(100);
      digitalWrite(sendLed, LOW);
      delay(100);
      signalState = digitalRead(signalPin);
      if(signalState ==0)   //if signal =0, or start using the shoe
      {
        delay(5000);
        break;      //go back to collect state
      }//end if
    }//end while(1)
  }//end if-SendState

}//end forever loop

void setupMPU(){
  Wire.beginTransmission(0b1101000); //This is the I2C address of
the MPU (b1101000/b1101001)
  Wire.write(0x6B); //Accessing the register 6B for power management
  Wire.write(0b00000000); //Setting SLEEP register to 0 to take off
of sleep mode
  Wire.endTransmission();
  Wire.beginTransmission(0b1101000); //I2C address of the MPU
  Wire.write(0x1B); //Accessing the register 1B  for gyro config
  Wire.write(0x00000000); //Setting the gyro to full scale +/-
250deg/s will cap movement up to 250 deg/s
  Wire.endTransmission();
  Wire.beginTransmission(0b1101000); //I2C address of the MPU
  Wire.write(0x1C); //Accessing the register 1C - Acccelerometer
Configuration (Sec. 4.5)
  Wire.write(0b00000000); //Setting the accel to +/- 2g
  Wire.endTransmission();
  }

  void recordAccelRegisters() {
  Wire.beginTransmission(0b1101000); //I2C address of the MPU
  Wire.write(0x3B); //Starting register for Accel Readings
  Wire.endTransmission();
  Wire.requestFrom(0b1101000,6); //Request Accel Registers (3B - 40)
  while(Wire.available() < 6);
  accelX = Wire.read()<<8|Wire.read(); //Store first two bytes into
accelX
  accelY = Wire.read()<<8|Wire.read(); //Store middle two bytes into
accelY
  accelZ = Wire.read()<<8|Wire.read(); //Store last two bytes into
accelZ
  processAccelData();
  }
```

```
/* Divide the raw accel values by the LSB ofr the +/- 2g */
void processAccelData(){
  gForceX = accelX / 16384.0;
  gForceY = accelY / 16384.0;
  gForceZ = accelZ / 16384.0;}

/* Record raw gyro values from registers of the gyro*/
void recordGyroRegisters() {
  Wire.beginTransmission(0b1101000); //I2C address of the MPU
  Wire.write(0x43); //Starting register for Gyro Readings
  Wire.endTransmission();
  Wire.requestFrom(0b1101000,6); //Request Gyro Registers (43 - 48)
  while(Wire.available() < 6);
  gyroX = Wire.read()<<8|Wire.read(); //Store first two bytes into
accelX
  gyroY = Wire.read()<<8|Wire.read(); //Store middle two bytes into
accelY
  gyroZ = Wire.read()<<8|Wire.read(); //Store last two bytes into
accelZ
  processGyroData();}

/* Process the gyro data by dviding by lsb for 250 deg/sec limit */
void processGyroData() {
  rotX = gyroX / 131.0;
  rotY = gyroY / 131.0;
  rotZ = gyroZ / 131.0;
}

void closeFile()
{
  for(int i =0; i<FILE_NO; i++)
  fp[i].close();

}
```

2. Propeller and Pi Sending / Receiving Data via Bluetooth - Serial Python Code

#Created by Khoi Nguyen and Kyle Arango
#Last edited: April 20, 2017

```
import serial
sd = serial.Serial()
device="/dev/rfcomm0"
sd.port=device
while(1):
	#print("In loop")
	line = sd.readline()
	#time.sleep(0.1)
	#if line and not line.isspace():
	#testfile.write('Reading First line:' + line + '\n')

	#print(line)
	if 'Done' in line:
		GS_Serial.write('Closing Files\n')
		GS_Serial.close()
		for i in range(0,7):
			fp[i].close()
		sd.close()
		break
	x=line.split(', ')
	#print len(x)
	if (len(x) == 4):
		fp[6].write(x[1] + ', ' + x[2] + ', ' + x[3] )
	else:
		for i in range(0,6):
			#print('File' +str(i))
			if 'File' + str(i) in x and len(x) == 3 :
			#print('Found File' + str(i))
				fp[i].write(x[1] + ', ' + x[2])
				Break
```

3. Linux auto shell script

```
 #!/bin/bash
sudo rfcomm connect 0 00:06:66:8C:D3:F8 >/dev/null &
sudo python /home/pi/button.py &
echo "Sleep" >> /home/pi/bluetooth.log.txt
sleep 10
echo "Done Sleep" >> /home/pi/bluetooth.log.txt
sudo /usr/local/bin/script.sh
```

4. Server SSH setup

To set up SSH server on windows, we use FreeSSHD software using SFTP protocol. This software allows windows based server remote connections via IP address. Users will be authorized prior connecting to server using its windows username and password. Port 22 is used for files transferring protocol. This way we can transfer any files data from Pi (as a remote) to our server(host).

The listen address is the IP address of the host with port 22 for file transferring protocol. Other values should kept as default. Figure 40 is the configuration of the Server.[76]

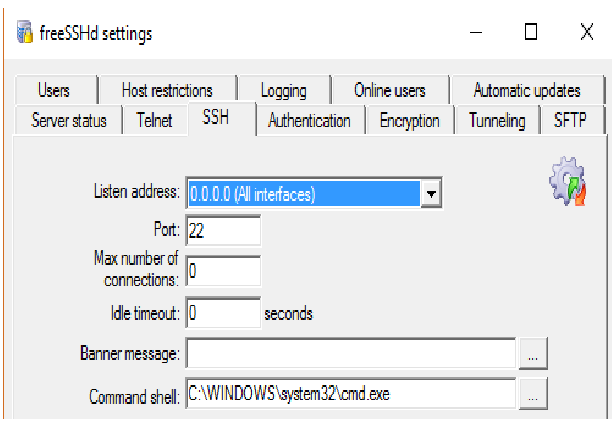

Figure 39-Configuration of SSH Server-1[75]

We then can add user and authorized remote PC to connect to server PC via its username and password.

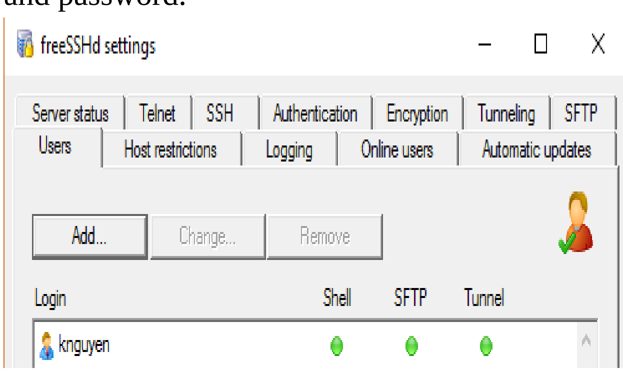

Figure 40-Configuration of SSH Server-2 [76]

5. Database

Figure 41 shows the database schema. [77]

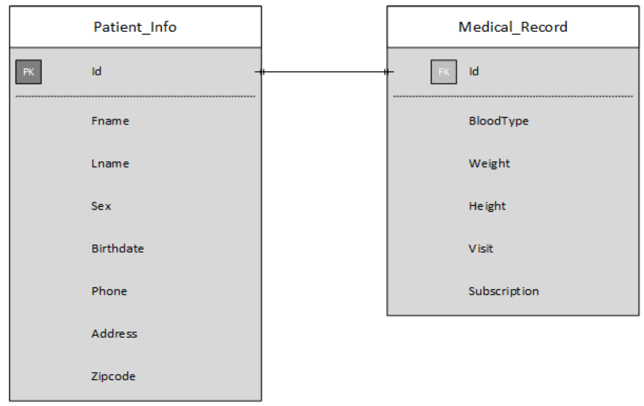

Fig 41 - Database Schema [77]

For our database, we simply create two tables: Patient_Info and Medical_Record to record patient's information. The Patient_Info table will keep track with patient's general information while Medical_Record table will record patient's medical details.

We use Visual Studio Community (free version) to develop an application which will be what the user interacts with, and would call stored procedures in the SQLite database to manipulate data for best ease of use, performance, and security. This application will be a friendly user interface environment could be written in C# for both end users (doctor, and developer) and will be stored in our physical server. This software will be deployable into any other PC such as doctor's PC to help to organize data and keep track with all the patients' information. The doctor can easily search, edit, insert, delete or retrieve any essential information such as pressure mapping data, temperature data of the patients ect.

Figure 42 shows user interface application.[78]

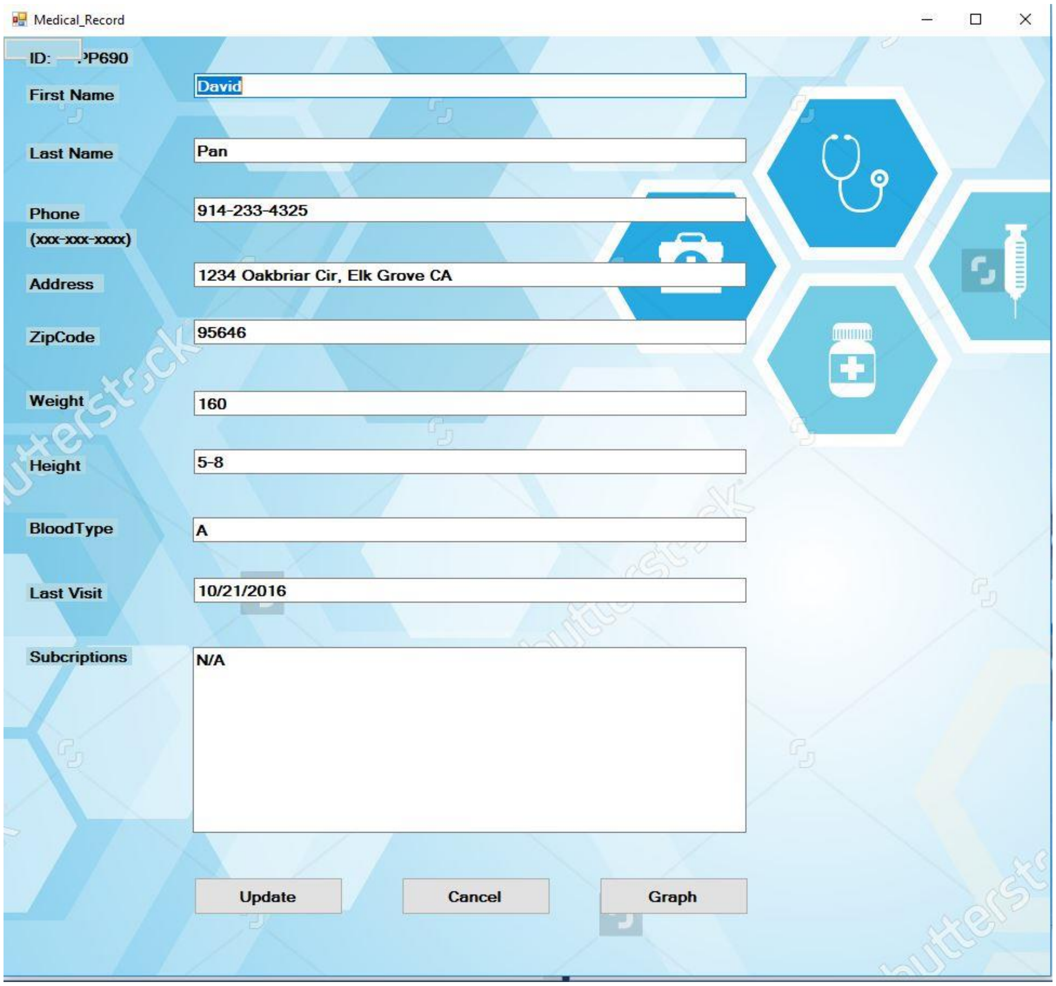

Figure 42a- Database App interface-1 [78]

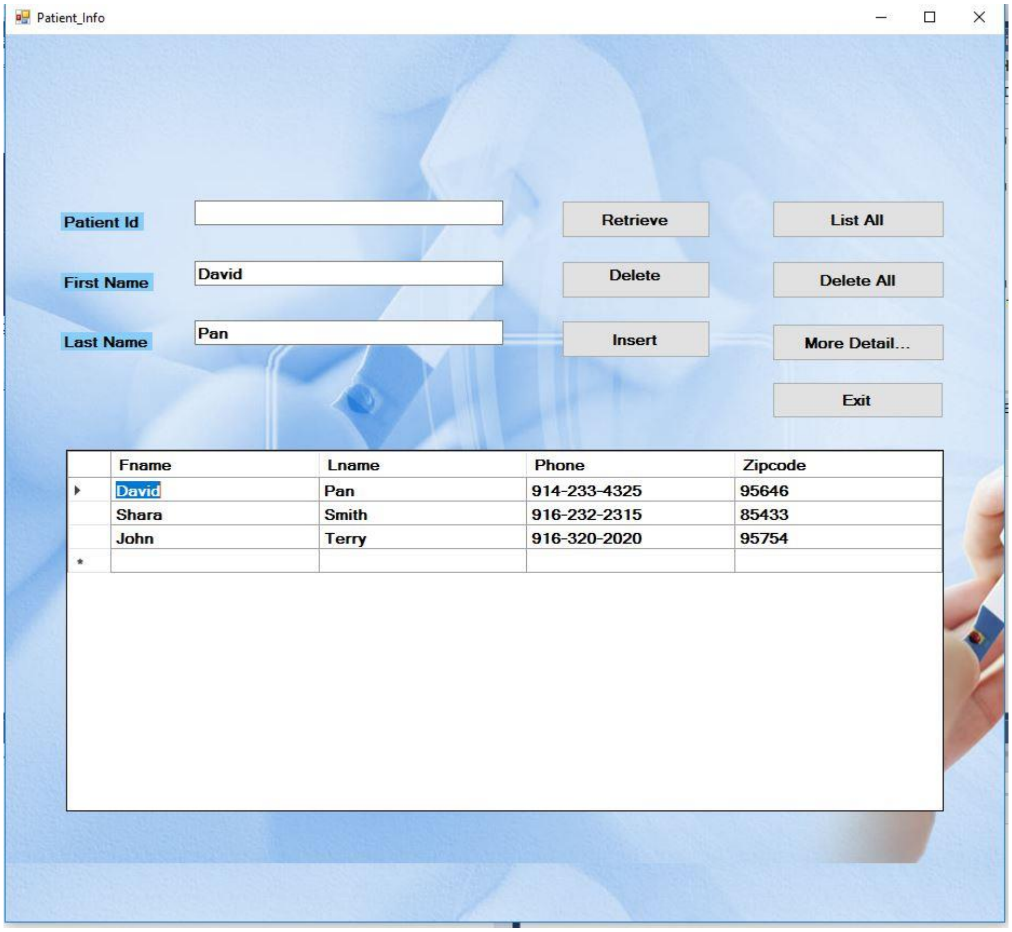

Figure 42b- Database App interface-2 [78]

6. Test Plan Result of Software

Test plan result for software shown in Table VI.[79]

TABLE VI
Software Test Plan Results

D01	**MicroSD test** -Verify that all data collected from pressure sensors are stored onto microSD in a right format. - Power on Propeller for ~ 30 secs for data collection, then stop.	Data should be transferred and saved onto a text file in MicroSD in the right format.	Able to store collected data on 4 output files on microSD	Pass	may need to use MicroSD-SD adapter to read output files
D02	**MC-Pi data transfer test** - Verify that data is transferred successfully from MC to Pi via Bluetooth connection.	- Collected data from MC should be present on Pi after Bluetooth transferring	Able to transfer data from Propeller to Pi	Pass	
D03	**Pi-Server data transfer test** -Verify that data is transferred successfully from Pi to Server via Wi-Fi connection - The transfer process will be run sequentially after the Bluetooth transfer process is done.	Data from the Raspberry Pi will be present on the Server as a text file.	Able to transfer output files from Pi to Server	Pass	
D04	**Data transfer automation test** -Verify that all the data transfer process is done automatically without user interaction - Place the shoe on the docking and the transfer process should be start automatically from MC-Pi-Server	Data should be present on the Server after all the transfer process is done	Process is done automatically when receiving signal from wireless charger Data is present on server.	Pass	
U01	**SSH Server test** -Verify that Server is on and able to communicate via SSH command -Using Putty program and connect to Server via Linux command	User should be connecting and able to retrieve any data on the Server	Server is running and user is able to connect and access data	Pass	
U02	**DB test** - Verify that DB software can retrieve, insert, delete data - Insert some test information then do search, and delete.	- Software should be able to meet all data manipulation from the user.	Able to retrieve and manipulate data	Pass	

U03	**Graph visualization test** -Verify that graph application can transform collected data into graphs -Open DB software, retrieve any patient and click on graph button.	**-**The graph should be generated as Pressure vs Time graph.	Able to generate 5 graphs	Pass	No graph for Gyroscope
A01	**Overall Test** -Verify all pieces are functional together -Use the Shoe as intended for incremental periods of time (2min, 5min, 10min, 30min etc.) -Place on docking station to test all automated features	**-**The shoe should stay on and collect data. -The battery should be recharging and transferring data while docked. **-**Data should reach the server and accessible to authenticated users.		Pass	

Appendix D
Mechanical

There are several mechanical portions in this project. The main one is the pedorthic shoe developed by Sunrise Shoes. This shoe has a patent pending on its offloading technology. It is in this shoe that all the electronics were placed and sensors were placed.

The sensors were placed in the lowest platform on the shoe and held down by double sided scotch tape. The sensors themselves have female to male wires on them that we soldered on extensions onto. The wires were routed to the side of the shoe where they will be brought up through the fabric of the shoe to the top of the tongue. In the figure 43 the assembly for the very bottom of the shoe. The wireless receiver coil is embedded into the shoe. [80]

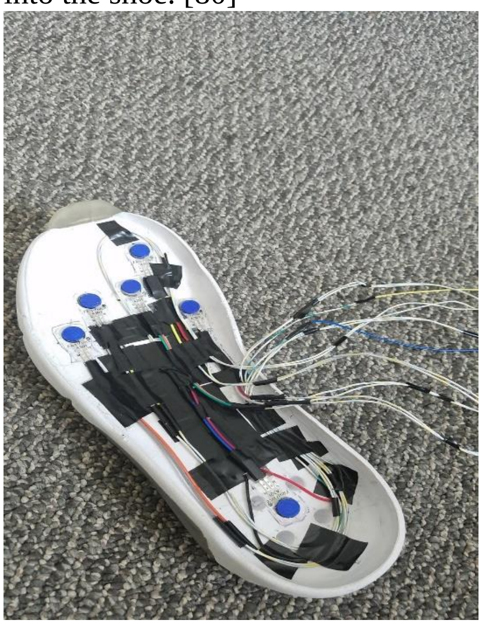

Fig. 43 - Bottom Shoe Sensors Assembly-1 [80]

The figure 44 shows the material that will be placed on the this portion of the shoe. [81]

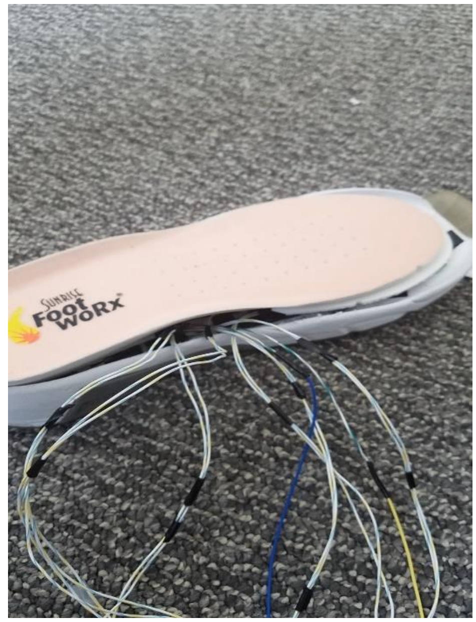

Fig. 44 - Shoe Wires Assembly [81]

The blue pucks visible in the picture is a small disk that was 3d printed to help distribute the weight evenly across the sensing area.

The shoe was put together using shoe glue and was set to dry over a 48 hours. See fig 45.[82]

Fig. 45 - Finish Shoe Assembly [82]

The wires were brought up through the fabric of the shoe.

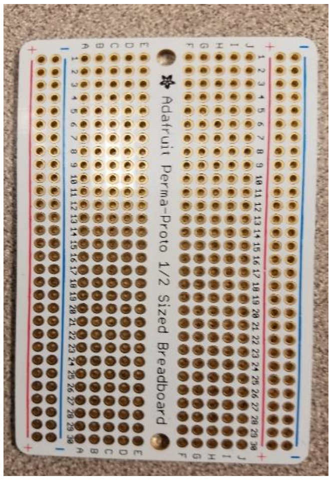

Figure 46 - Protoboard by Ardafruit [83]

The electronics were soldered onto a protoboard shown in the figure 46.[83]
These electronics were then housed in a 3d designed pcb box. The box houses the batter and all other electronics not in the sole of the shoe already. The box is designed to hook onto the shoelaces of the the shoe with four points of contact for stability. See Figure 47.[84]

The docking station houses the AC/DC converter and the DHU unit. The top of the station where the shoe rests has a small rest for the charger coil to sit exposed. This will help minimize the distance between charger and receiver. The top also has guides to make sure shoe is placed in the right spot every time.

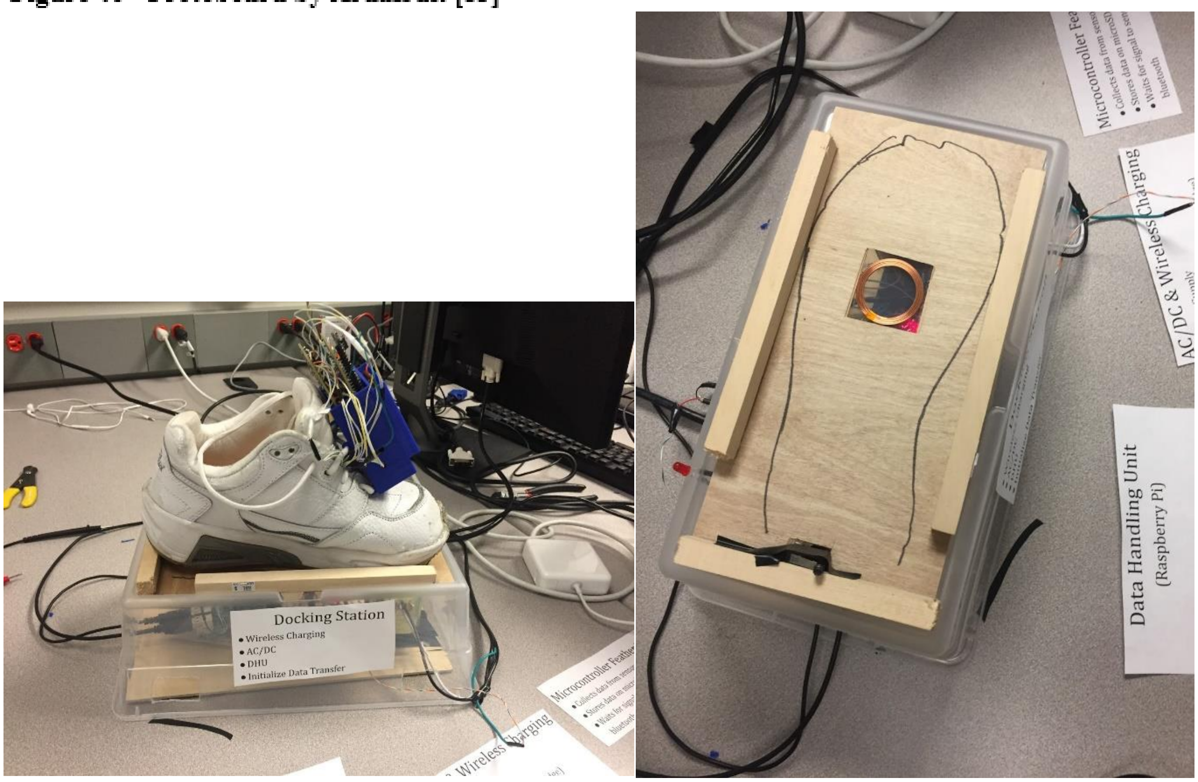

Figure 47 - Shoe on Dock (left) and Top view of Docking (right) [84]

There were also 3d designs printed for the docking station an pcb box. The mechanical drawings for these are uploaded on the hive.

Appendix E
Vendor Support and Industry Contacts

Sunrise shoes was our sponsor on this project. They helped us in the planning stage by providing information and documents to help us understand the societal problem we faced. They also provided us with feedback on our design ideas and provided the expertise in the medical field where we do not have expertise. They donated shoe samples for us to us to consider while deigning our project and for use for the final deployable prototype.

Appendix F
Team Members Resume

KHOI NGUYEN

Email: knguyen.academy@gmail.com

OBJECTIVE:
Seeking a position in the technology industry where I can utilize my programing and computer technical skills.

EDUCATION:

2014 – Present

Sacramento State University, CA

Degree Expected: Bachelor of Computer Engineering

Classes taken: CPE 64, CPE 142, CPE 151, CPE 166, CPE 185, CPE 186, CPE 159, CSC 28, CSC 60, CSC 130, CSC 134, CSC 138, CSC 139, EEE 117, EEE 108, EEE180, ENGR 120

2011 – 2014

Cosumnes River College, CA

Classes taken: CISC 300 - Computer Familiarization, CISP 300 - Algorithm Design/Problem Solving, CISP 360 - Introduction to Structured Programming (C++), CISP 400- Object Oriented Programming with C++), CISP 430- Data Structure with C++, CISP 310 – Assembly Language

VOLUNTEER AND EXPERIENCE HISTORY:

FPPC- Fair Political Practices Commission
Jun 2014 – Present
Position: Computer technician, Student assistance
Job duty: maintain website, company VB application and I.T.

CNService
Jan 2014 – 2015
Position: Computer technician
Job duty: on call service, setting up new computers, printers, cables for companies when called.

Math and Computer Lab Tutor
July 2013 – Present
Math /Computer lab Center – Cosumnes River College, Sacramento, CA
Position: Math, computer lab Tutor
Job duty: assisting and tutoring students with math and computer problems.

Computer technician: 3 years experiences
Jan 2010 - Jan 2013
Servatek Computer, Elk Grove, CA.
Position: Computer Technician
Job duty: Maintaining and repairing equipment, installing hardware and software systems, troubleshooting a variety of computer issues, offering technical support on-site or via phone calls.

Samuel Woldeyes

11768 Dionysus Way
Rancho Cordova, CA 95742
Phone: 916-889-6371
E-mail: Samuel.woldeyes217@gmail.com

Objectives

To receive an entry level job with possibility of advancement

Education

- High School graduate in May 2013
- California State University, Sacramento
- Planned graduation date: May 2017
- Degree objective: Bachelor of Science, Electrical Engineering
- G.P.A: 3.15
- Engineering G.P.A: 3.4
- Planning to attend graduate school after receiving bachelor's

Courses Taken: Electronics I, Electronics II, Advanced Logic Design*, Semi-conductor device physics*, Electromechanical Conversion, Digital Signal Processing, Computer Hardware Design, Feedback Systems, Microprocessors, Applied Electromagnetics, Power Electronics, Communication Systems
*in progress

Experience

- Assisted non-profit organization raising funds by selling food and drinks to the public
- Shift Supervisor at Sacramento State Dining Commons
- Attended Intel Ultimate Engineering Experience class learning new ways to work with others and time management on projects

Projects

- Patch Antenna – designed a simulation of a patch antenna with Rt/duroid 5880 laminate at 7.4GHz
- FM Receiver – constructed FM radio with lab group on a breadboard and received radio signals between 88-108 MHz
- Dual Axis Solar Tracker – constructed solar panel that rotates on two servos toward area of brightest light
- Diabetic Footwear with Instrumentation – Our team is designing footwear with sensing capabilities to detect foot ulcers without user interaction and present real time data to a physician through use of a server

Skills

- Programming (C, arduino, Verilog, VHDL)
- Computer Applications (Microsoft Word, Excel, PowerPoint, PSPice, Multisim, Quartus, Xlinix)
- Hardware (Oscilloscope, voltmeter, FPGA, SRAM, OTA, feedback amplifiers)
- Good collaboration/teamwork with others

Interests/Achievements

- Member of Mesa Engineering Program(MEP)
- Member of IEEE, NSBE, Tau Beta Pi Engineering Honor Society
- Dean's Honor List: Spring '16, Fall '16

Pratik Patel

apatel@csus.edu

OBJECTIVE: To Obtain a Career in EEE

EDUCATION:
Bachelor of Science, Electrical and Electrical Engineering, CSU Sacramento
COURSES:

Senior design*	Electronics II Lab
Robotics*	Electronics I
Introduction To Feedback Systems	Electronics I Lab
Introduction to Digital Controls	Network Analysis
Introduction Signals and Systems	Electromechanical Conversion
Introduction to Microprocessors	Modern Communications Systems
Electronics II	Engineering Economics

** In progress as of Fall 2016*

KNOWLEDGE AND SKILLS
Leadership/Management:
Skilled at motivating team members and identifying individual strengths to delegate responsibilities. Can effectively coach individuals and groups to improve performance and skills.

Problem Solving:
Able to recognize the concepts in any given problem and able to research unknown information. Can use tools like Pspice and MatLab to make solving problems easier. Solid circuit analysis techniques.

Hardware/Software:
Excel, Word, Powerpoint, MatLab, Visual Studios C. Have worked with Arduinio, Propeller, Raspberry Pi, Amani GTX. Can adapt to various different IDE's for microcontrollers, CLPD's and simulation tools like Pspice.

Communication:
Effective in social situations, interact easily with new people and establish rapport. Experience in explaining complex ideas and concepts in easily understandable language. Awesome customer service skills. Fluent in English and Gujarati.

PROJECT EXPERIENCE:
Sunrise Shoes Project
With our senior project, we aimed to reduce medical costs due to complications with diabetes by integrating technology with current treatment methods. We worked with a local company to implement sensors that in clinical shoes for patients that help remotely monitor the pressure in the patient's feet and track treatment compliance. Data was gathered and sent to a server where it was sorted into a database to be accessed by a physician at a later time.

/ORK EXPERIENCE:	

IESA Tutor College of the Sequoias, Visalia CA helped tutor Math, Physics, and Chemistry to students in the STEM field.	**/12 - 5/13**
lath Lab Tutor Library, College of the Sequoia's	**/12 - 5/13**

I helped tutor any student taking any math class

ACTIVITIES AND ACCOMPLISHMENTS:
Member: Member of the MESA Engineering Program
Award: 1st place in the math proportion for high school Academic Decathlon regional competition

Kyle Arango

	karango1992@gmail.com
	(530) 867-7727

EDUCATION:

In progress: **BS, Computer Engineering**, CSU Sacramento, 3.818 CSUS GPA, 3.255 Cumulative GPA, expected June. 2017

KNOWLEDGE AND SKILLS:

Computer Languages: C · C++ · Java ·Verilog · VHDL · Matlab · BASIC · x86 Assembly
Software: Visual Studio · Eclipse · Multisim · XilinxISE · Matlab, · PSPICE
Tools: Oscilloscope · Arbitrary Waveform Generator · Logic Analyzer

RELAVENT COURSES:

Advanced Computer Organization	
Advanced Logic Design	
	· Field Programmable Gate Array(FPGA) Usage
Computer Hardware Design	
	· PCIe interface, arbitration, bus structures, interrupts
Computer Interfacing	
	· Design of microcomputer systems including memory systems, parallel and serial input/output · Microcontroller programming

Computer Network and Internet		
	· Study and application of common internet protocols: TCP, UDP, HTTP, FTP, SMTP, etc.	
CMOS & VLSI		
Data Structure and Algorithm Analysis		
	· Memory management and efficient algorithm design	
Electrionics		
	· Ideal Op-amps, BJTs, MOSFETs, single stage amplifier	
Intro to Systems Programming in UNIX		
	· UNIX environment programming and shell script programming	
Network Analysis		
	· Laplace transforms, Bode plots & Fourier Series	
Operating System Pragmatics		
Operating System Principles		
	· Process and thread, concurrency, scheduling, interprocess communication and synchronization	
Signals and Systems		
	· Time-domain and frequency-domain analysis of linear continuous-time and discrete-time systems	

ACTIVITIES AND ACCOMPLISHMENTS:

	· Tau Beta Pi Engineering Honor Society Member · Dean's Honor List, Fall 2014, Spring 2015, Fall 2015 · President's Honors, Spring 2013 and Fall 2012 · CCNA Discovery: Networking for Home and Small Businesses Certified · ACM Programming Contest 2nd Place (CSUS Lower Division)

Appendix G
Sensor Calibration

In addition to having a force to voltage converting circuit in the project, we also needed to condition and calibrate the sensors.

To condition the sensors, we would press on the sensor sitting on scale until we reached a 140 lb on the scale momentarily then unpressed. This was done about five times, and conditions the sensor for repeatability. Each consecutive press would yield close to same output voltage, which is the expected result.

To calibrate the sensors we load the sensor and read output voltage for a range of forces. For our experiments we used a range of 5 lbs to 80lbs. This was do to the fact that consistent actuation could not be maintained with the motion of our foot. After recording the output voltage that corresponding to a force value, we put these in an excel sheet and found the expected linear relationships between the force applied and voltage output of the circuit. The summary of these results are shown in the figures 48,49,50. [85],[86],[87]

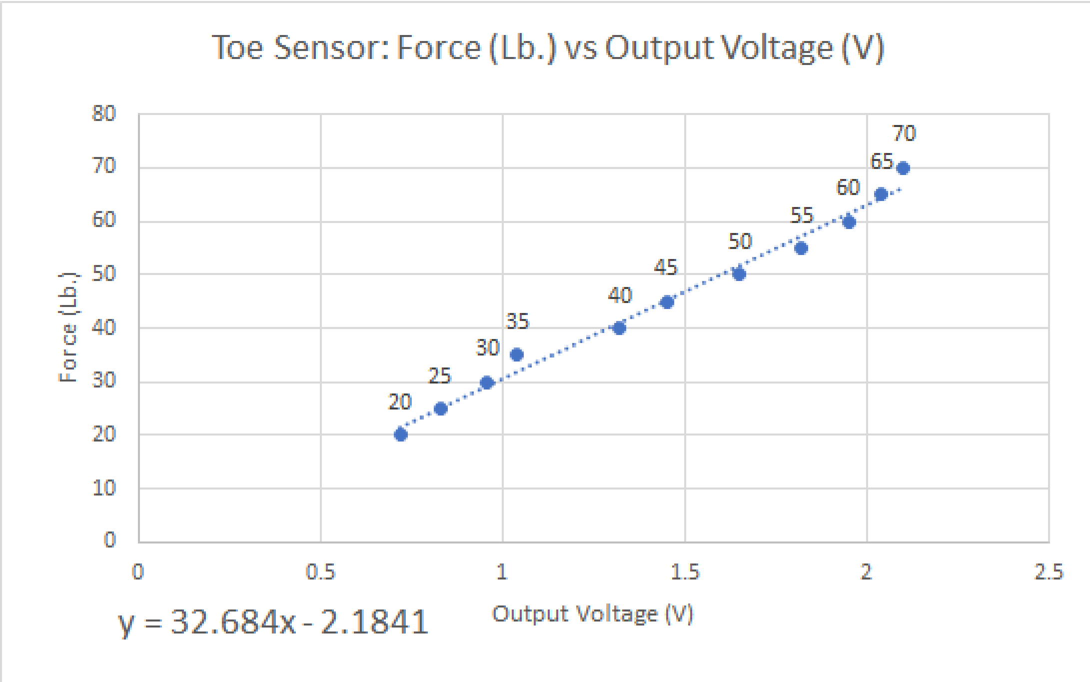

Fig. 48 - Force vs Voltage of Top Sensor [85]

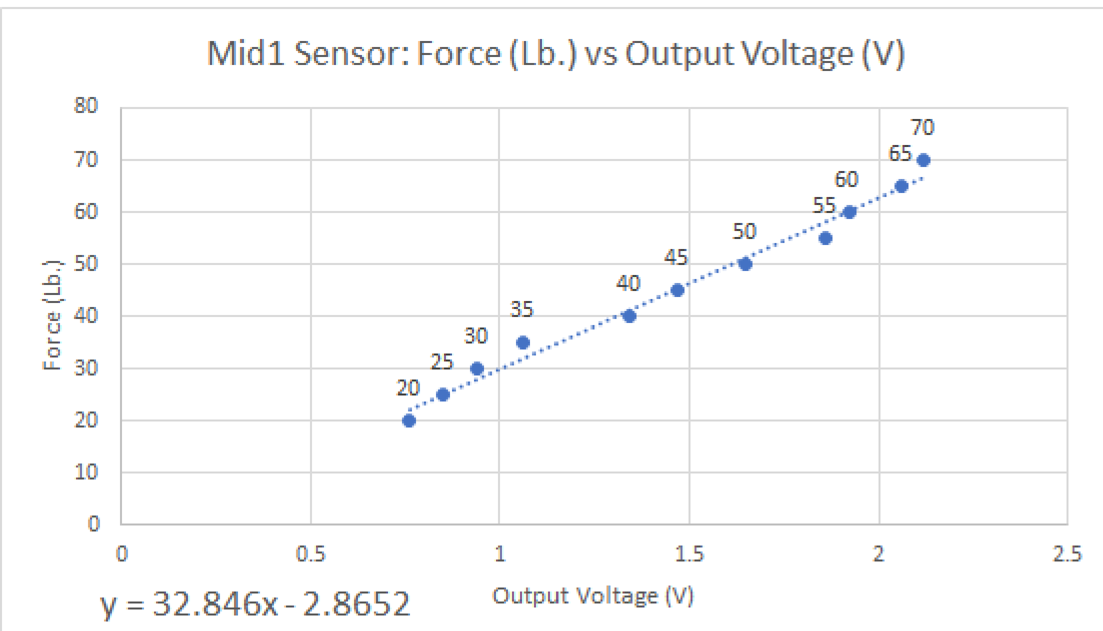

Fig. 49a - Force vs Voltage of Mid Sensor (1) [86]

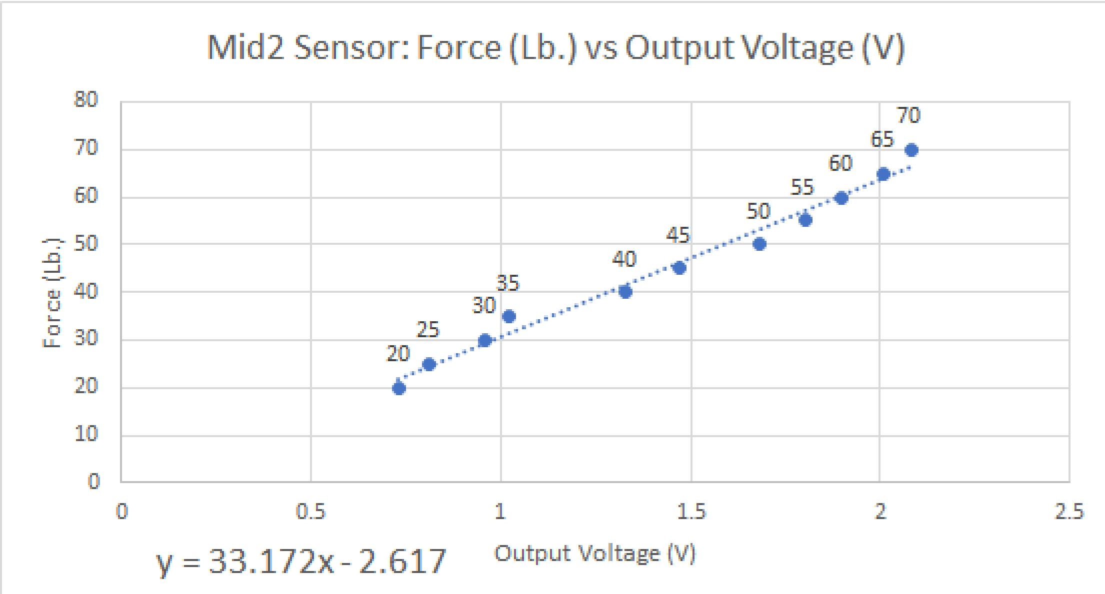

Fig. 49b - Force vs Voltage of Mid Sensor (2)[86]

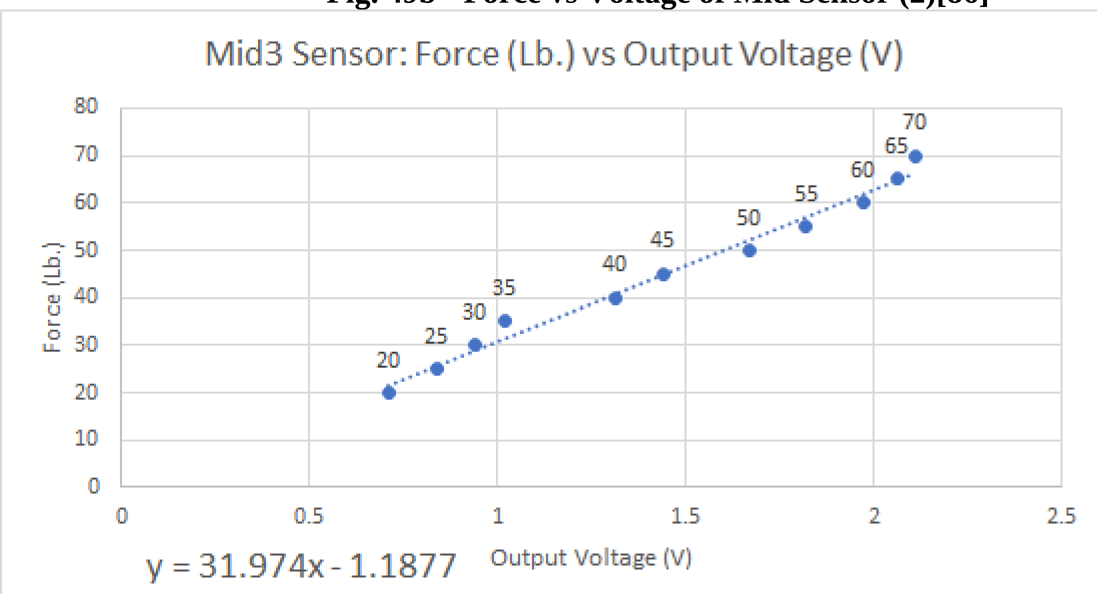

Fig. 49c - Force vs Voltage of Mid Sensor (3) [86]

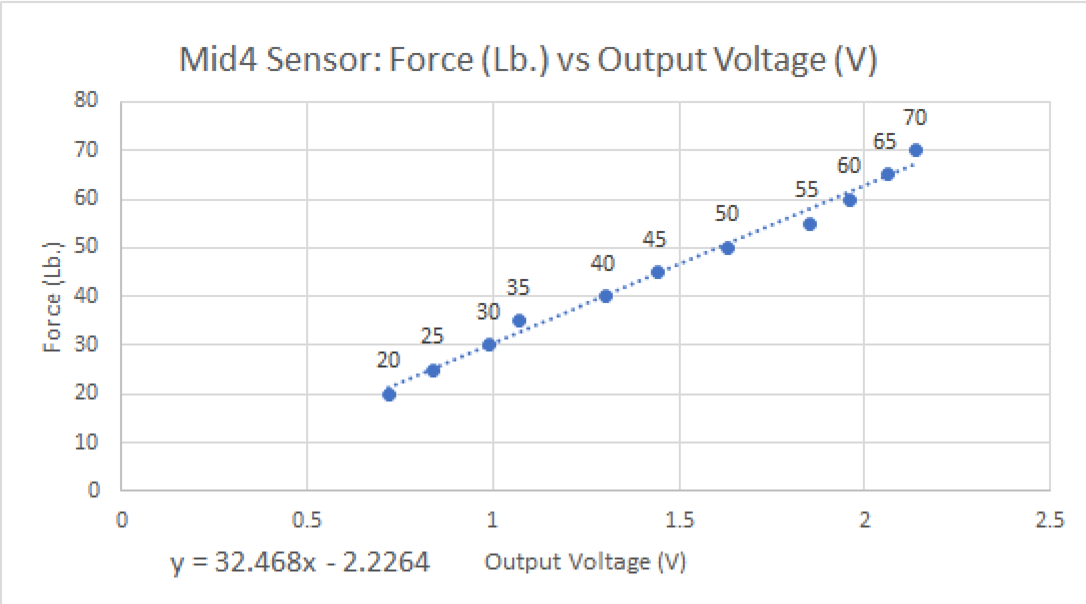

Fig. 49d - Force vs Voltage of Mid Sensor (4) [86]

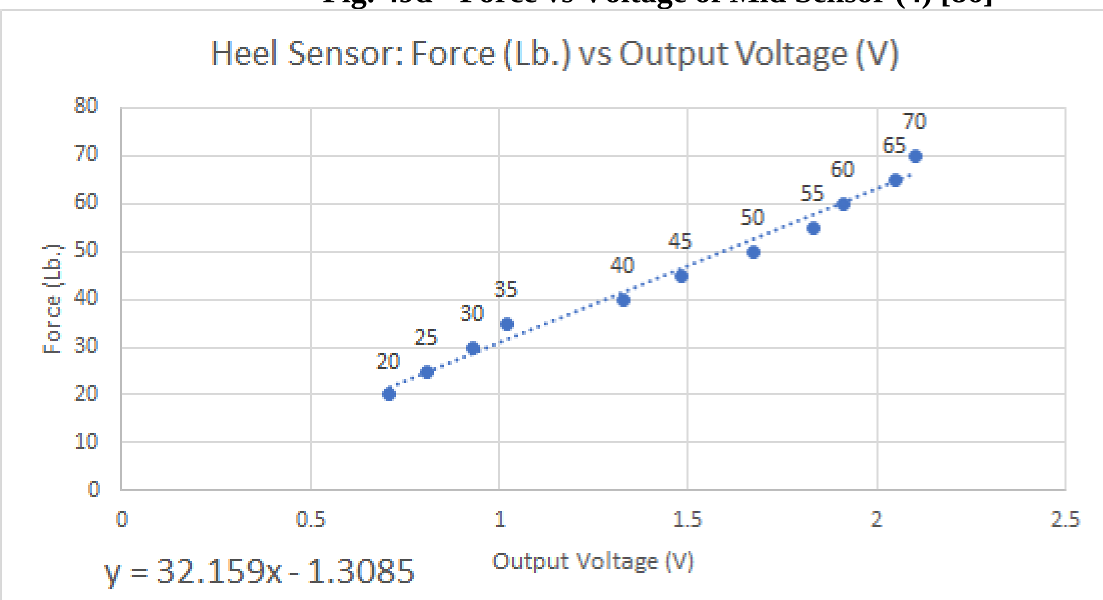

Fig. 50 - Force vs Voltage of Heel Sensor [87]

The calibration curve for each sensor are very close in value to each other, which is another expected result. Each sensor has similar actuation and same reference resistance, which means the calibration curve should almost be identical.

We calibrated 6 sensors to use, however only 5 ended up use. The reason for this was the placement of some electronics on the shoe prohibited placements of some sensors in the shoe, and adding the 6 sensor on our final hardware cause unforeseen issue of introducing noise on a multi purpose pin that can be used as an analog input.