



Sense Yo Soles©

Aimee Davis

Alex Bravo

Anthony Childers

Client: Ronny Elor

Advisor: Dr. Ben Hawkins

Table of Contents

Executive Summary	2
Introduction	2
Project Overview:	2
Clients and Community Partners:	3
Stakeholders:	3
Framed Insights and Opportunities:	4
Background	8
Engineering Specifications	11
Design Development	13
Final Design	17
Functional Decomposition	17
Design Verification	21
Design Verification Plan	22
Verification Test Results	23
Sensor Linearity	23
Battery life	25
Transmission Range	27
Wireless Pairing	29
System Analysis	29
Management Plan	30
References	36

Executive Summary

The goals for the Sense Yo Soles shoe insert is to port the previous Java application to an Apple phone application, to implement a sensor communication system that allows data from two sensor systems (two insoles) to be sent to a single Apple device via bluetooth, to receive and display both standing and striding data, and to implement power management through the use of Bluetooth Low Energy. The results of our final design are two packaged shoe insoles each with their own bluetooth modules that connect to an Apple device. Each insole sends a stream of 3 pressure sensor data values. An Apple phone that has the application downloaded can connect via Bluetooth to each insole to receive the data stream which then appears on the phone in a user friendly way.

Introduction

Project Overview:

The Sense Yo Soles project is a system that is embedded into a shoe insert; the plantar pressure distribution is measured by calculating the ratio of pressure on 6 individual pressure sensors located on the lateral, medial, and heel areas of both the left and right foot to the total pressure distributed across both feet. The alpha prototype created by our client and her Biomedical Engineering senior project team at San Jose State University, measured and analyzed data using the above method while connected to power and a computer. In CPE 350, we made the system stand alone and wireless allowing for the product to be more usable, and reduced the overall cost of the system. We achieved

this by using an ATtiny84 microcontroller to control these peripherals and to send data wirelessly using Bluetooth to an external device for processing. In CPE 450, we ported the Java application to an Android application, implemented sleep cycles with interrupts for power management, and implemented a dual sensor communication, allowing 2 sensors to send data to a single Android device. Our main goals for CPE 461/462 were to use Bluetooth Low Energy for power management, to display standing and stride data in a user friendly way on an iOS application, and to package hardware into the insole.

Clients and Community Partners:

The client is Ronny Elor; the Sense Yo Soles project was started as Elor's Biomedical Engineering senior project at San Jose State University and is in the beta prototype stage. Ronny Elor's team developed the initial plantar pressure shoe sole product.

Stakeholders:

Dr. Benjamin Hawkins is the only stakeholder with whom we have direct contact. Dr.

Hawkins was Ronny Elor's advisor for the initial iteration of the Sense Yo Soles project at San Jose State University and is our team's advisor.

Framed Insights and Opportunities:

During the Fall 2016 and Winter 2017 quarters, we had weekly video chat meetings with Ronny, excluding holiday breaks. During the first conversation Fall quarter we shared a little about ourselves

as well as asked a series of questions regarding the current prototype and the changes Ronny wants to see at the end of that quarter. It was decided that the three priorities that need to be accomplished by the end of Fall quarter are wireless connection, standalone battery, and the improvement of the sensors (primarily cost reduction).

One goal mentioned during the project introduction week 1 was having a mobile app where data that is collected from the sensor can be viewed for analysis. During this first meeting, Ronny helped us to prioritize, and together we decided that having the data be displayed on the computer as it is for Elor's prototype, rather than a mobile app was okay for our iteration of the alpha prototype. While the end goal of this project (end of CPE 450) is to have the sensors, battery, and microcontroller all contained within an insole, Ronny decided that having a dongle of some sort that clips on to the ankle or shoe laces for this alpha prototype is acceptable. Our next meeting went a little more into depth as we studied the goals and objectives of this project and researched some solutions. Ronny stated that she wants the soles to take some readings so we can ensure that accurate data is still being collected. In addition, she suggested we begin to think about what materials to use for casing of the device as well as how to distribute force over the sensors correctly. We had one official meeting with our stakeholder, Dr. Hawkins, to ensure we were making progress in the right direction. Dr. Hawkins has background knowledge in this project and was able to assist in declaring a need statement, use cases, and personas that aligned with that of the original designers of this project. The meetings with Ronny throughout the rest of the quarter were mainly updates, our opportunity to let her know of the current status of the project. In addition, we were able to ask questions regarding how her team got the serial data to display on the visual image of the foot. Her

prototype collected and stored serial data in a log, to be viewed at a later time. This was stale data. Comparatively, our alpha prototype collected, transmitted, and displayed data all at the same time, allowing for real time data to be analyzed.

At the beginning of Winter quarter in CPE 450, we worked with Ronny and Dr. Hawkins to establish our Beta Prototype requirements. Each week we had team meetings with Dr. Hawkins and Ronny, during which discussed the progress on the previous weeks tasks, as well as the goals for the upcoming week. During the first few meetings the software team took the time to research what would be the best platform to create the application on. Upon Dr. Hawkins suggestion, we eventually decided to use Android Studio, to dive into the work instead of researching too much about the platforms. The software team had significant troubles with establishing a bluetooth connection over the course of the next few weeks, which was discussed in each weekly meeting. The learning curve for Andriod Studio was steep - no one on the Sense Yo Soles team has worked with it before, and online documentation is difficult to follow and implement. A course in Android application development would have been very beneficial. Learning on our own as we went took time. In the beginning of Winter quarter Ronny had requested that hardware team implement gait data collection in addition to standing. The hardware team made an executive design decision, that given the timeline we had to complete our tasks, gait would not be able to be implemented for beta prototype but could for the Senior Project portion of this project, this quarter. The Sense Yo Soles team continued to keep Ronny updated regarding progress and problems throughout the duration of the quarters.

This quarter Aimee Davis, Alex Bravo, and Anthony Childers continued development as a Senior Project, the focus of this report. The goals for this quarter were to miniaturize the hardware such that the entire sensor system would be contained in two shoe inserts. These inserts would have expanded battery life, improved communication specifically bluetooth low energy, and the sensors would be better supported to improve accuracy of their readings. The hardware would also implement gait data collection. At the beginning of this quarter it was decided to move the software implementation to an iOS platform which all members had more experience with. The software would display current standing and striding data, communicate with the sensor devices via bluetooth low energy, and have a method to record and export that data. The initial plan for hardware implementation was to build custom hardware. A design was made, however, before components arrived a fully built solution was found. Adafruit produces a feather board that was of the correct dimension and has battery charging and bluetooth low energy built in. The drawback was that all of the firmware used to run the sensors had to be rewritten also the bluetooth low energy libraries were unknown and had little documentation. Still all features were remade to support the new hardware as well as bluetooth low energy communications and gait data collection. This quarter there was no contact with Ronny, however we had bi-weekly meetings with Dr. Hawkins. In these meetings we discussed the progress of the project and Dr. Hawkins was able to provide insights in, for example, how to best graphically display the data being received.

Goals and Objectives:

Goal 1: The device should be wireless.

Objectives: Add battery power supply, replace corded communication with wireless communication through Bluetooth.

Goal 2: Reduce cost compared to present alpha prototype created by Ronny and team.

Objectives: Replace alpha prototype components with more common, less costly, and better suited analogs, primarily less expensive sensors that are better protected so they are less likely to break causing the need to purchase more.

Goal 3: Increase usability.

Objectives: Reduce size of the product by removing unneeded components such as breakout boards, cords, jumpers, etc. Package the product such that it is a self contained unit that can be inserted into any pair of shoes. Goal 1, making the device wireless, will increase usability as the need of a wired connection to a computer to communicate will not be necessary. Port the Java Application to a mobile iOS application, allowing the user to view data anywhere.

Goal 4: Send data from two insoles.

Objectives: Implement a sensor communication system that allows two sensor sets to communicate with a single iOS device. Both sensors must be fully implemented using bluetooth.

Goal 5: Power Management.

Objectives: Implement Bluetooth low energy support. Increase battery capacity.

Outcomes and Deliverables:

- We will deliver an iterative prototype that is wireless, self contained, and battery powered at a greatly reduced cost.

- We will deliver an iOS phone application that will receive wireless data from the prototypes, process, and display the results visually without numbers.
- We will deliver a system that receives data from 2 insoles with implemented sleep cycles and interrupts for power management.
- We will deliver 2 insoles that can be placed into any shoe that will collect and transmit standing and gait data for a minimum of 12 hours.

Background

The Sense Yo Soles project is a system that is an extension upon and an improvement of an alpha prototype produced by Elor's senior project team. Ronny and her team did a significant amount of research on standing conditions and what ranges of ratios of pressure on each location in the foot is considered healthy and abnormal. Moving forward with Ronny's prototype, we did not evaluate the truth of this research, and rather trusted that if our prototype collects pressure the same way that her prototype did, then the analysis of it (ie. the calculations of pressure ratios) would be accurate as far as what they had originally discovered. The end system has two states during which it will gather data, standing and striding. Using priority specifications provided by our customer we decided to first implement the standing state, wireless operation, and wireless communication during CPE Capstone 350 and 450. Once the system had detected a standing state it began taking sensor measurements via 10 bit ADC built into the ATtiny MCU. Readings were taken at a set time interval and sent via bluetooth module to a laptop where the data was be processed. This quarter, we focused on stride data collection, miniaturization, battery life, and mobile data collection. In this

phase an AT Mega 32-u MCU was used to collect analog data, translate it to digital data via 10 bit ADC, and transmit that data via bluetooth low energy to the mobile application built on an iOS device.

The Bluetooth connection for our first prototype was accomplished using two HC-05 Bluetooth modules. These modules are backup devices to be used in case the more complicated Bluetooth low energy modules could not be integrated. These modules are “hard paired” meaning that they will only connect to each other and will automatically connect when ever in range of one another. Once the bluetooth connection is established the ATTiny 84 begins to take reading using an on board analog to digital converter. The sensor readings range from 0-1023 and are sent to the bluetooth module for transmission as a single string. The current prototype uses nRF51822 chipset bluetooth low energy modules that are on board each insole. The iOS application uses its device sensors to detect the user’s motion and assigns a bluetooth low energy characteristic indicating which type of data to collect, standing or striding. This data is collected and assigned to a different bluetooth low energy characteristic which the iOS app monitors for updates. The iOS app connects to the sensors by name “SenseYoLeftSole” and “SenseYoRightSole”. This connection is made automatically on launch of the application. The data is still in a range from 0 to 1023 and sent as a string.

The backend for the original prototype was developed in Java and was receiving data via serial connection. For our initial prototype this serial connection was adapted to use bluetooth instead of a physical (wired) connection. Doing this required writing a serial connection program in C to connect to the bluetooth device through a bluetooth serial port, retrieve data and then write the serial data to

first-in-first-out (FIFO) file, which was opened by the Java application developed by Elor's team and used in her original prototype. For our final prototype we abandoned the use of serial data connection and the original Java backend code. The required backend was rebuilt for iOS using Swift, the primary language for iOS development. Swift contains built in libraries for creating bluetooth low energy connections and the communication was accomplished via the Apple device's built in bluetooth low energy module. The Java code provided to us by Ronny for the analysis and display of data, was ported to Swift, and maintains the same functionality as before. The final system implemented a bluetooth low energy service and characteristic scheme. In this scheme the sensors are alerted when a data selection characteristic is updated by the application and the application is alerted whenever the data characteristic is updated. This scheme enables extension of battery life. The microcontroller spends idle time in a sleep mode; it will only 'power on' for a few moments at a time to gather and transmit requested data. The analysis of this data is done in the mobile iOS application.

This prototype is designed with a few personas in mind. The first is someone who knows that they have a problem with how they stand however do not have the resources to see a doctor or specialist for help. These insoles are a product that can be purchased in a local store that can provide assistance in making corrections on these habits formed in order to reduce pain and further complications. Another person that could benefit from this device is someone recovering from injury who has completed physical therapy however needs to continue practicing proper stance and weight distribution so as to not injure themselves again. This product can be used as a diagnostic

tool for doctors and physical therapists. Patients can wear this device and raw data can be sent to their medical professional who can analyze the data and provide further corrective measures. Sense Yo Soles will have a few competing products. The company Moticon has a product, OpenGo science, that is similar to the Sense Yo Soles project. It is also an insole that analyzes plantar pressure distribution; it uses thirteen Moticon produced pressure sensors per sole, versus our three. Like our project, the data analysis is done outside the embedded system in either a mobile or desktop application. The system is entirely inside the insole with no outside wires or connections. iShoe is another similar product; it uses FlexiForce force sensors to gather data for a balance diagnostic. This information can then be analyzed to help prevent falls.

Engineering Specifications

The overall goals are that the device should be wireless with sufficient range to allow a user to move freely, the device must be internally powered with sufficient capacity to allow a user to collect data for a large percentage of a day, lower overall cost, display the data in a user friendly way on a mobile application, have dual sensor communication and power management, and package the device in a way that allows for users to be mobile while the device is collecting data.

The engineering specifications that are derived from the overall goals include reducing the cost of the prototype by replacing alpha prototype components with more common, less costly, and better suited analogs. Increasing usability is done by reducing the size of the product by removing unneeded components such as breakout boards, cords, jumpers, etc., creating a wireless connection,

as well as packaging the product such that it is a self contained unit. All of this is shown in Table 1 below.

Table 1: Engineering Specifications

Spec Number	Parameter Description	Requirements or Target with units	Tolerance	Risk	Compliance
1	Bluetooth Communication	10 ft	Min	L	T,S
2	Battery Operated		Absolute	L	A,T,S
3	Battery Life	6 Hours	Min	M	A,T
4	Production Cost	\$200	Max	M	A
5	Size	Self contained in an insole	Max	M	A, I
6	Maintain Functionality	Detect improper standing posture (up to 100 lbs per pressure sensor)	Absolute	H	A,T,S
7	Mobile App	Display data	Min	H	T,S
8	Data collection for both feet	Two sensor arrays must collect data and pass it to application in a single data stream.	Absolute	H	A,T,S

Design Development

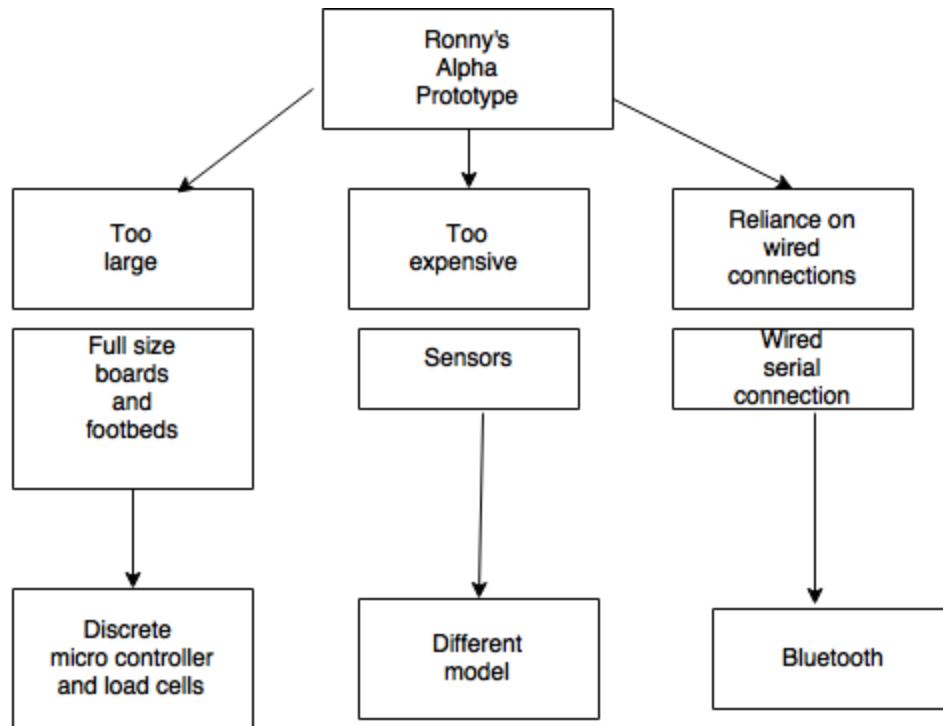


Figure 1: Design Development, Approach

With the overall goals of advancing the provided prototype to be wireless with a price reduction in mind, we were not given a specific route to take. Our team made it a priority to lower price, therefore, we researched parts that were cheaper than what had been used in the previous iteration. In Figure 1 above, the schematic starts off with Ronny's alpha prototype which has three major issues, listed in the tier below. The three issues that we set out to fix, included the prototype being too large, too expensive, and having a reliance on wired connections. Tier 3 goes into details regarding why those problems were present in Ronny's alpha prototype. The system was too large due to the use of full size microcontrollers. The system was too expensive primarily because of the sensor's that Ronny's team used. It turns out that our team used the same sensors in our alpha

prototype however, the sensors were much cheaper now than they were at the time Ronny purchased them. In addition, Ronny's team did not properly protect the sensors, so they broke very easily, which requires purchasing more sensors, increasing the overall production cost of the prototype. Lastly, the wired serial connection made the prototype unable to be used. The 4th tier shows our solutions to these problems including using discrete microcontrollers and load cells, and bluetooth connection to transmit data.

Figures 2, 3, and 4 below show our overall design schematics for sensor integration, bluetooth integration, and data flow from sensor to the the display software for both the capstone design and the final design.

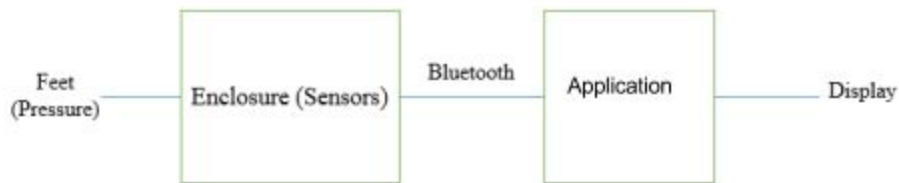


Figure 2: Level 0 Black Box Diagram

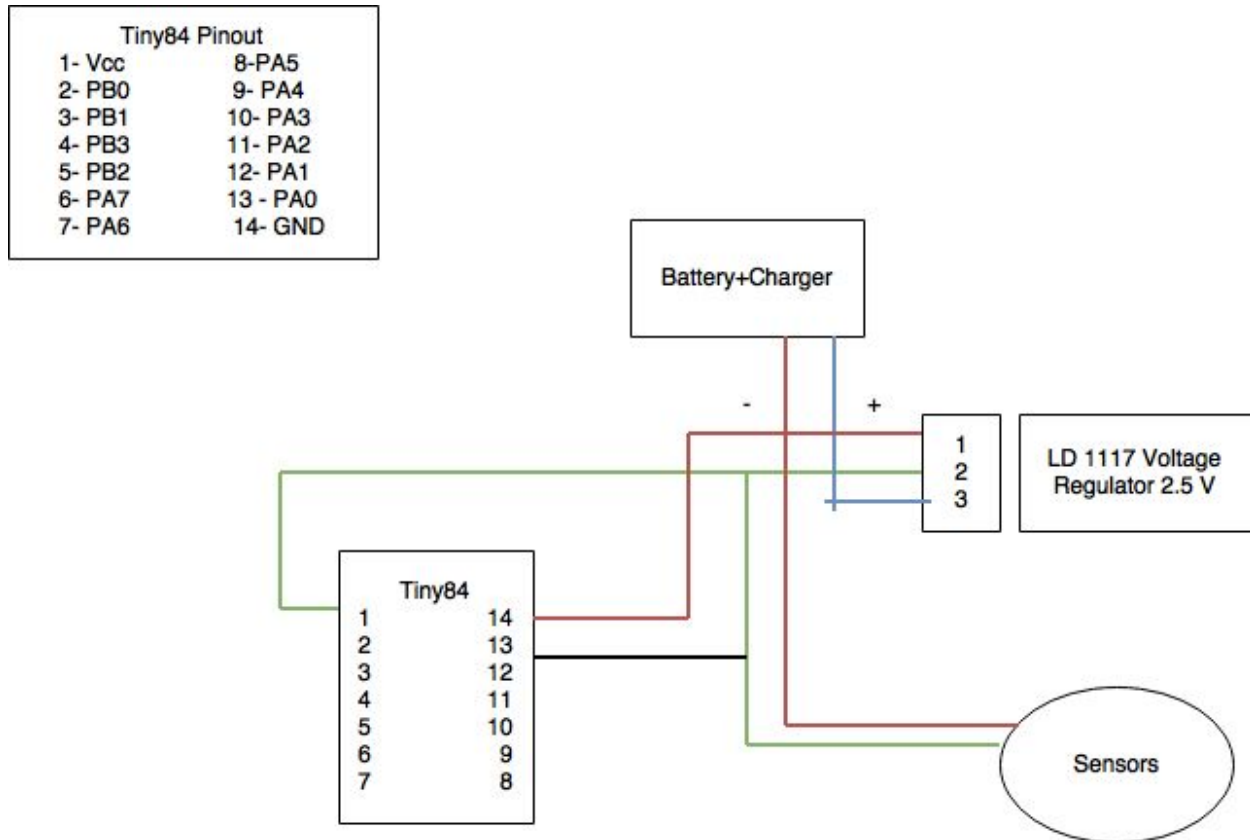


Figure 3: Sensor Wiring and Microprocessor Integration Capstone design

In order to evaluate these schematics, tests were created and used. To test our bluetooth module, a simple sketch file was made to transmit sample data from the sensor module Figure 4, to the receiving bluetooth module Figure 5 connected to the host laptop. Using the serial port monitor built into the Arduino IDE we checked to see if data was being sent properly. To test the pressure sensors, a voltage divider was created and set to 2.5V as shown in Figure 3. Weights were added in 2.5lbs increments to be used as data points. A potentiometer was used to find the correct bias resistance needed to produce a linear change in voltage in the range of 0 ~ 100 lbs. Finally, to test our battery, direct readings were taken from the assembled prototype and we calculated a 7 hour battery life. A practical test was then performed according to the specification sheet for the battery. The battery was fully charged, allowed to rest for 5 minutes, then the device was powered on and

left in a continuous data collection and transmission loop until the battery discharge protection circuit cut power. The device transmitted for greater than 7 hours. Transmission range and sensor testing was conducted by collecting data with the sensor installed in a shoe as the user walked around our lab. Continuous data was received at greater than 20 feet through the lab's cinderblock walls and the sensors show no sign of wear or damage. Sensor readings are still at calibrated levels from the biasing resistor selection test.

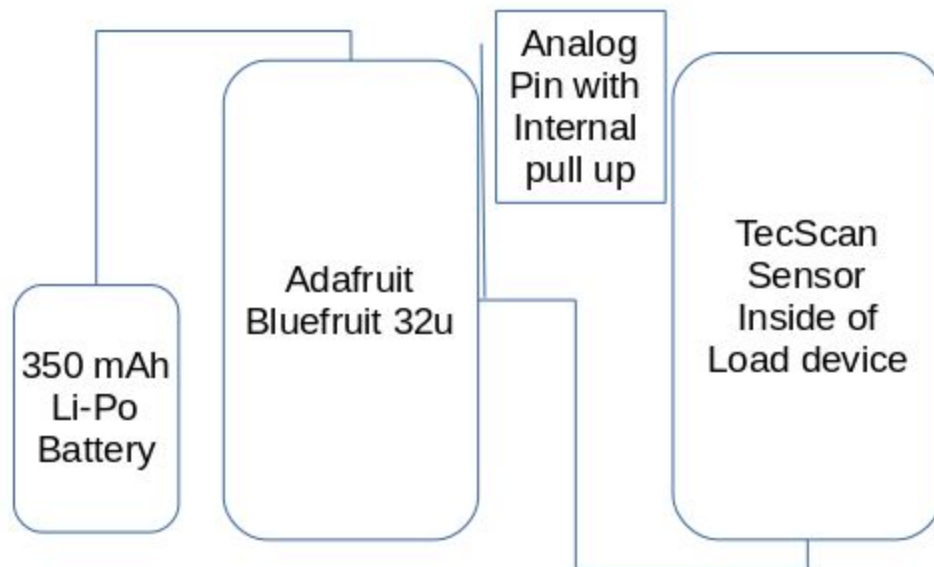


Figure 4: Sensor Wiring and Microprocessor Integration Final design

The final design uses an Adafruit Bluefruit Feather board that has all required circuitry built in. There is no longer a need for an external voltage divider to run the sensors, a charging circuit for the lithium polymer battery, an external voltage regulator to power the system, or an external bluetooth

module. The use of this component allowed for significant miniaturization of the final prototype as well as a 133% increase in battery capacity nearly doubling battery life.

Final Design

Functional Decomposition

In Figure 5 below you see the level 0 functional decomposition of our project. This is a basic look into the system, where pressure is applied to the sensors and the weight distribution status is outputted on an application.

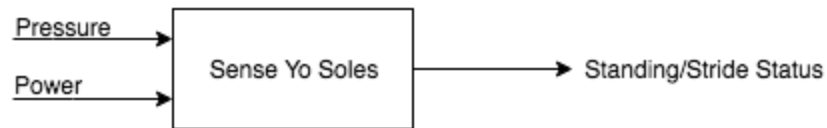


Figure 5: Level 0 Sense Yo Soles Functionality

Table 2: Level 0 Sense Yo Soles Functional Requirement

Module	Sense Yo Soles
Inputs	<ul style="list-style-type: none"> - Power: 3.3 VDC - Pressure: User puts pressure on sensors
Outputs	<ul style="list-style-type: none"> - Standing/Stride Statues: Displayed on screen
Functionality	Monitor the status of the user's standing position based on the ratio of three points of the foot. Output should display the user's compliance to a customer provided model based on that ratio at these points.

In the level 1 functional decomposition as shown in Figure 6, the user puts pressure on the pressure sensors which send serial data to the application that analysis the data and provides a status update for the user.

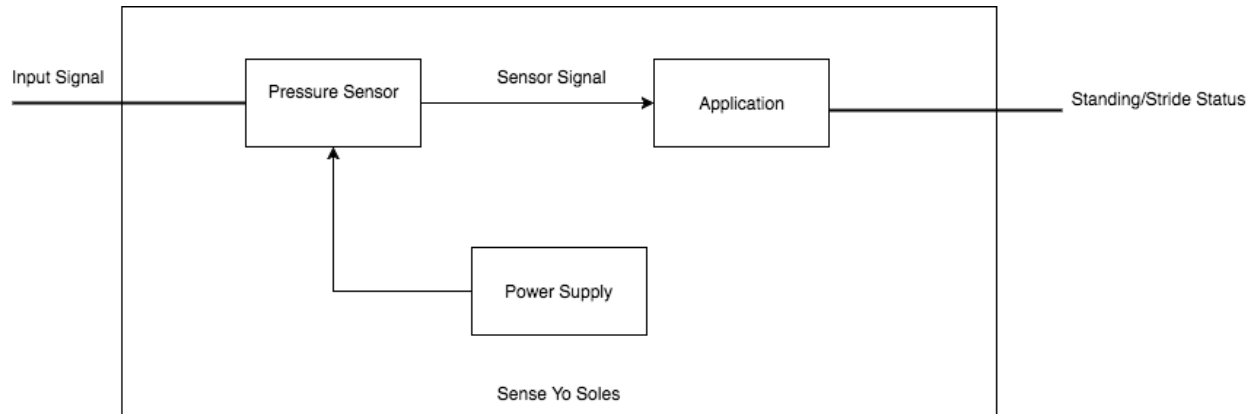


Figure 6: Level 1 Sense Yo Soles Design

Table 3: Level 1 Pressure Sensor Functional Requirement

Module	Pressure Sensor
Inputs	<ul style="list-style-type: none"> - Force applied to sensor surface - Power 3.3V
Outputs	Sensor Signal: Bluetooth Low Energy, Digital Sensor signal
Functionality	Convert analog sensor readings to digital values and transmit to application

Table 4: Level 1 Application Functional Requirement

Module	Application
Inputs	Data from sensors in ASCII
Outputs	Indication of posture correctness
Functionality	Process sensor data to evaluate level of compliance to the provided model

Table 5: Level 1 Power Supply Functional Requirement

Module	Power Supply
Inputs	
Outputs	3.3 V
Functionality	Provide power to sensor and reference voltage for analog to digital conversion operations.

Figure 7 shows a level 2 functional decomposition for the pressure sensor design. Data from the pressure sensors comes in as a voltage. That voltage goes through an ADC which converts it to a digital value that is sent to the bluetooth module for the application to evaluate.

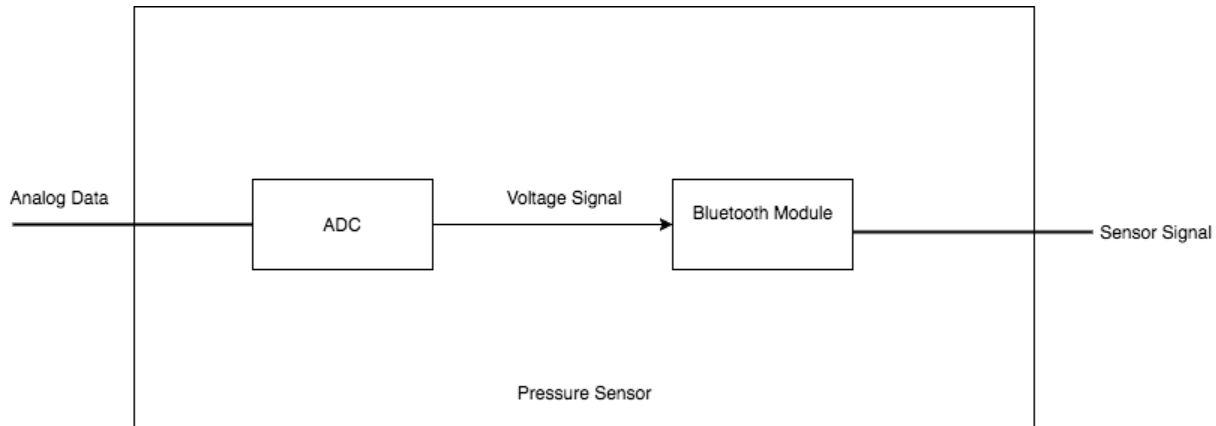


Figure 7: Level 2 Pressure Sensor Design

Table 6: Level 2 Pressure Sensor Functional Requirement

Module	ADC
Inputs	Analog data reading 0 - 3.3 V
Outputs	Digital value 0-1023
Functionality	Convert analog voltage data to a digital value such that: $(\text{analog value} / 3.3\text{V}) = (\text{digital value} / 1024)$

Table 7: Level 2 Pressure Sensor Functional Requirement

Module	Bluetooth Module
Inputs	Digital sensor value
Outputs	Sensor Signal: Bluetooth Low Energy, Digital Sensor signal
Functionality	Transmit digital sensor reading to the application module using bluetooth protocol

Figure 8 shows the level 2 functional decomposition of the application. Serial data is received and the application performs calculations on the serial data. These calculations include parsing the input

stream and then calculating the ratio of pressure on each sensor to the total pressure distribution across all 6 sensors. These ratios are then compared to values that were determined by Ronny and her team as “correct” ratios for proper stance. The status of each sensor is then displayed to the user.

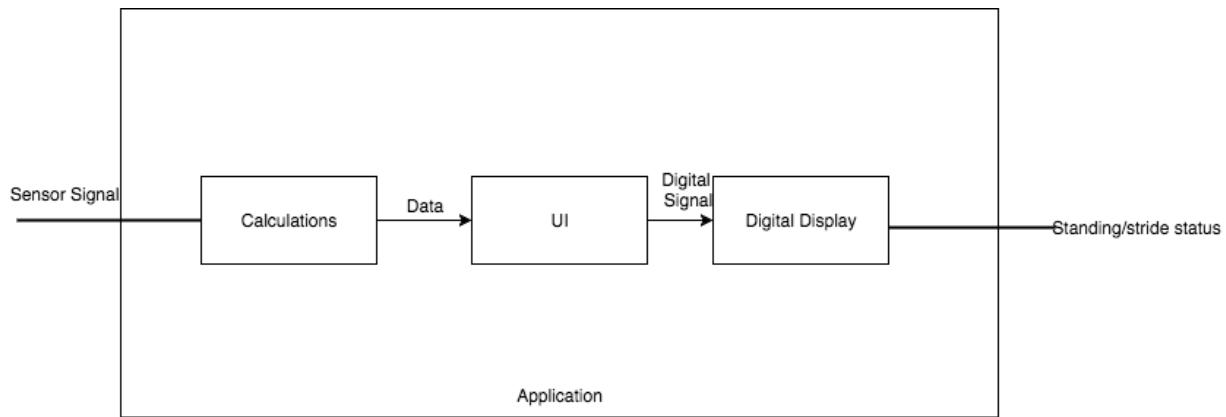


Figure 8: Level 2 Application Design

Table 8: Level 2 Application Functional Requirement

Module	Calculations
Inputs	Sensor Signal: Bluetooth Low Energy, Digital Sensor signal
Outputs	Analysis of data compared to provided model
Functionality	Compare data to provided model

Table 9: Level 2 Application Functional Requirement

Module	UI
Inputs	Analysis of data compared to provided model
Outputs	Colored display data indicating fit to model
Functionality	Convert analyzed data to graphical display steam

Table 10: Level 2 Application Functional Requirement

Module	Digital Display
---------------	-----------------

Inputs	Graphical display stream
Outputs	Visible display
Functionality	Convert data stream to pictures

Design Verification

FMEA

Table 11: Failure Mode and Effects Analysis

FAILURE MODE AND EFFECTS ANALYSIS																
Item:	Sense Yo Soles System				Responsibility:	Core Team				FMEA number:	1					
Model:	Current				Prepared by:	Core Team				Page :	1 of 1					
Core Team:	Aimee Davis, Alex Bravo (Software) Anthony Childers (Hardware)									FMEA Date (Orig):	6/15/2017 Rev: 1					
Process Function	Potential Failure Mode	Potential Effect(s) of Failure	S e v	C l a s s	Potential Cause(s)/ Mechanism(s) of Failure	O c c u r	Current Process Controls	D e t e c t	R e p a r	Recommended Action(s)	Responsibility and Target Completion Date	Action Results				
												Actions Taken	S e v	O c c	D e t	R e p a r
Wireless Data Transfer	Loss of Bluetooth Connection	Disruption of data flow	2		Radio interference, sensor out of range	3	Firmware Automatically reconnects	1	6	Remark in the user manual	team; product marketing/ release				0	
	Failure of Bluetooth Module	total system failure	7		environmental exposure, water damage, dynamic stress	2	component mounted in protective enclosure	1	14	Remark in the user manual						
	Data Packet Corruption	Slow down of data stream	1		Radio interference, sensor out of range	3	Bluetooth Protocol automatically corrects	1	3						0	
Sensors read pressure	Sensor deviates from linear readings	invalid data to user	7		Improper use or premature sensor failure	2	None	1	14	Remark in the user manual	Sense Yo Soles team; This should be completed before product marketing/ release				0	

Software analyses data	Incorrect calculations are performed	invalid data to user could result in user incorrectly fixing standing posture and gait	5	The algorithm used to analyze data, provided to use by Ronny, could be incorrectly functioning with the format of the data being received	1	1	5	Confirm Ronny's research on data analysis by consulting her provided sources.	Sense Yo Soles team; should be completed during the testing and evaluation phase of this project (after beta prototype)					0
battery power	Failure of Charging System	Total system failure	7	environmental exposure, water damage, dynamic stress	2	1	14	Create User Manual to prevent misuse	Sense Yo Soles team; This should be completed before product marketing/release					0
	loss of battery capacity	Reduced period of service	4	overcharge, undercharge	1	1	4							0
Microcontroller operation of sensor	Failure of Microcontroller	Total system failure	7	environmental exposure, water damage, dynamic stress	2	1	14	Create User Manual to prevent misuse	Sense Yo Soles team; this should be done before product marketing/release					0
	broken physical connections of sensor components	Total system failure	7	environmental exposure, water damage, dynamic stress	2	1	14	Create User Manual to prevent misuse	team; this before product marketing/release					0

Design Verification Plan

Table 12: Test Verification Plan

Item No	Specification	Test description	Acceptance criteria
1	pressure sensors must produce linear change in voltage for weights between 5 and 50 lbs	Establish a theoretical slope using sensor data sheet. Measure voltage return from physical sensor at multiple points in the range to establish a real slope. Repeat test with all sensors	no more than 5% deviation from theoretical linear return
2	battery life must be > 6 hours	device used for multiple days charged completely every day battery data recorded in application	battery life exceeding 6 hours
3	range of connected sensor must be at least 10ft	test data collection at ranges > 10 feet	data collection uninterrupted at 10+ feet

4	sensor must wirelessly pair with android device via bluetooth	test paring with multiple devices from as many manufactures as possible to verify compatibility with serial bluetooth protocol	device will reliably pair with all devices
5	verify that sensor data is sane	compare raw data to original data collected by Ronny's team	data collected is of similar range and magnitude to the original results
6	test software algorithm for proper display of posture	directly display raw data along with ratios produced by the algorithm and compare outputs with expected output provided by ronnie	application displays expected results
7	test hardware interrupts and procedural code	monitor output and current consumption during operation for proper sleep state and data throughput	sensor outputs recorded data and enters a low power mode between readings

Verification Test Results

Sensor Linearity

To test the sensor linearity known weights ranging from 3 - 50lbs were placed on sensor cells. The test was run five times each on one brand new sensor, one used sensor from the same lot, and one used sensor from a much older lot.

The test results show fine individual sensor linearity however the individual sensors do not report the same output at the same pressure. The problem was traced to sensor alignment on the load distribution cell. Consistent sensor alignment is nearly impossible to achieve with the current load cell design. This data led to a redesign of the load cell.

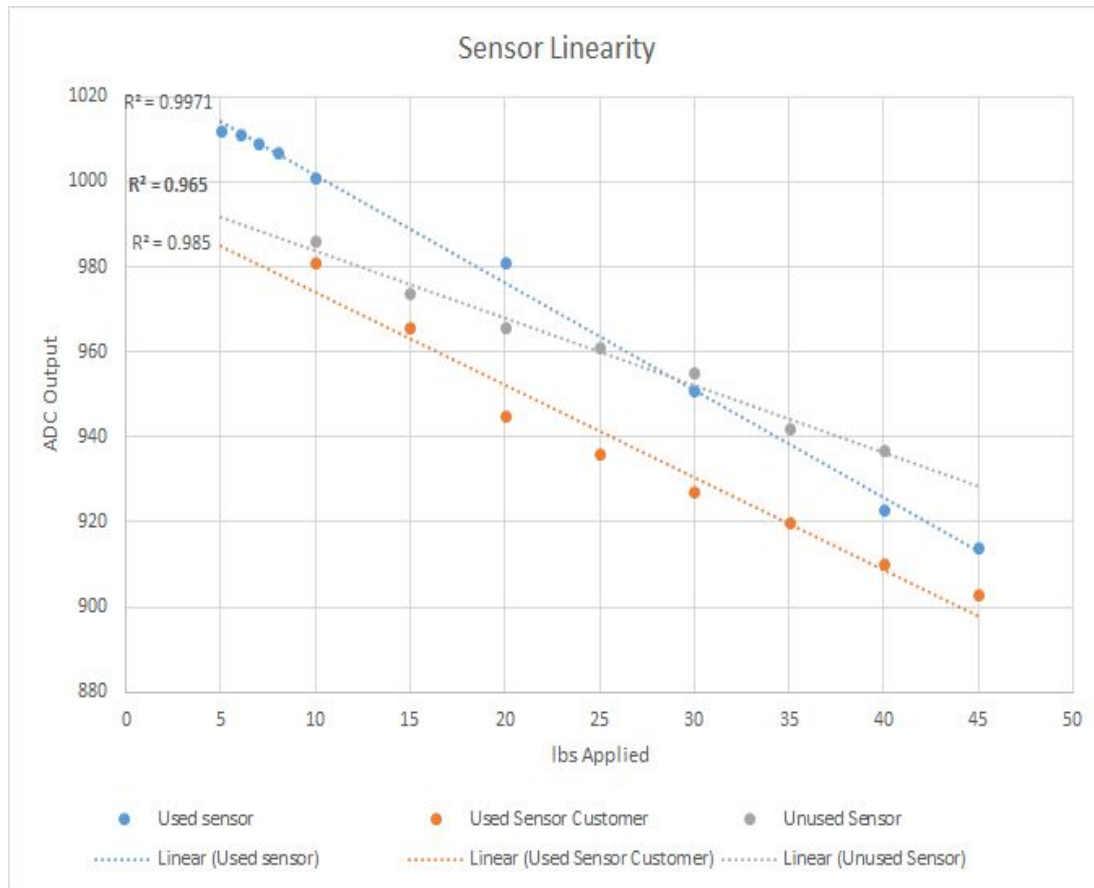


Figure 9: Sensor Linearity Test Results

After the load cell redesign this quarter, sensor data is much more precise. However each sensor seems to have an independent offset and some difference in linear response. After more research it seems that the sensors must be calibrated before each use. Also exact placement of sensors on the insole is key to gathering good data. Ideas for a calibration function that could be easily performed by the user were explored as well as the idea that the current sensors may not be a good fit for this application. The figure below shows sensor linearity data using three sensors from the set of new

sensors ordered for the final prototype.

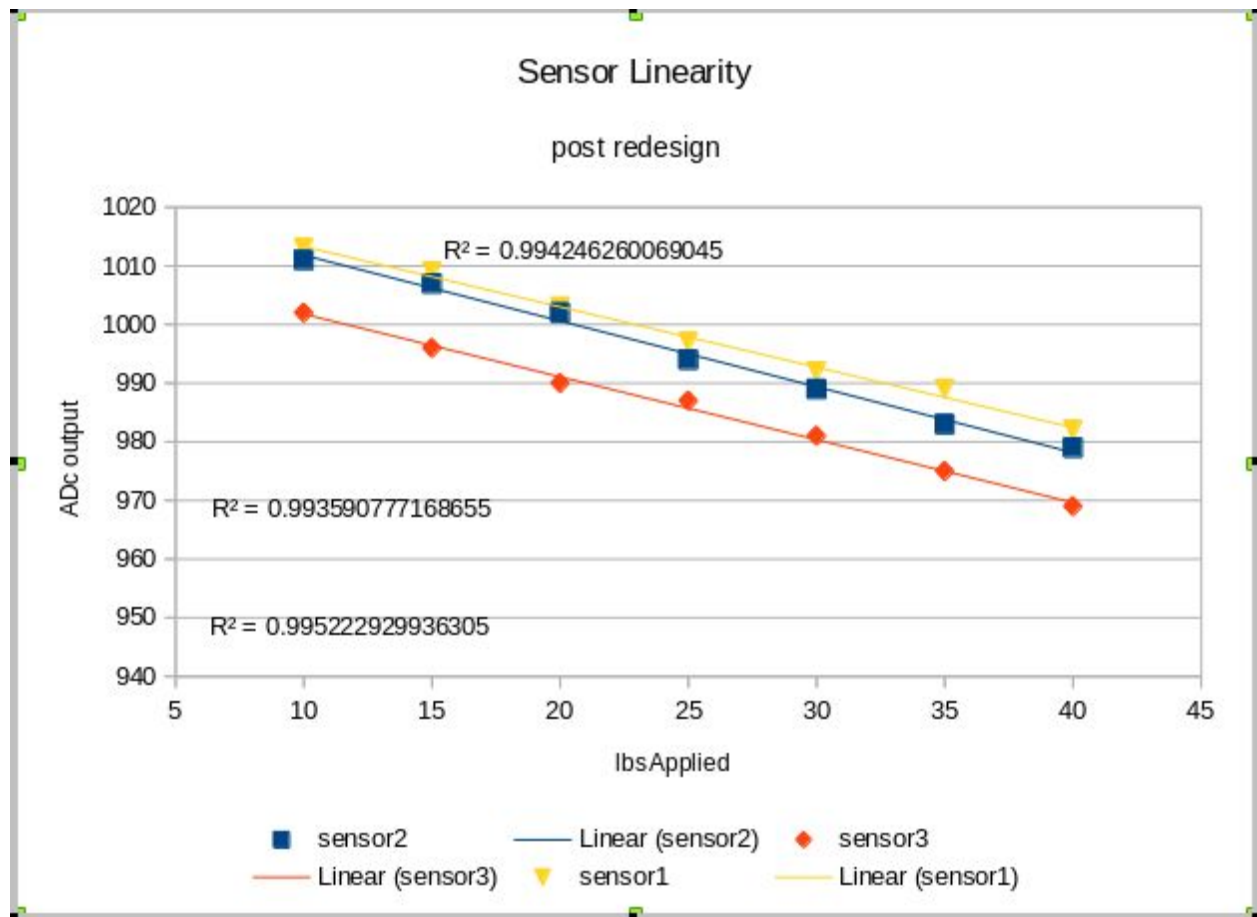


Figure 10: Final Design Sensor Linearity Data

Battery life

Battery life of the prototype was tested by charging up the equipment to the full power, and then letting the system run until it died. This was done twice in order to ensure accuracy. The results of the two test runs are shown below.

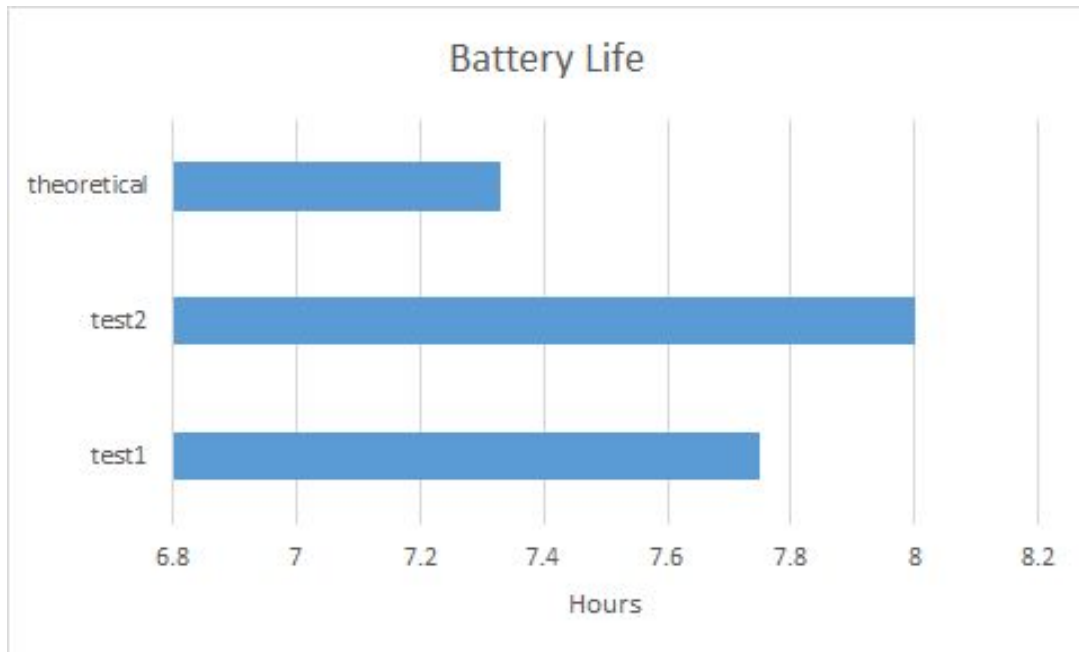


Figure 11: Battery Life Test Results

Battery life testing for the final prototype was done in the same way however the battery capacity was increased from 150mAh to 350mAh when combined with the implementation of bluetooth low energy and an optimized professionally manufactured circuit board with many built in power saving features battery life was increased. Due to long running times results are shown to nearest half hour in which the device was found still operational Results shown below.

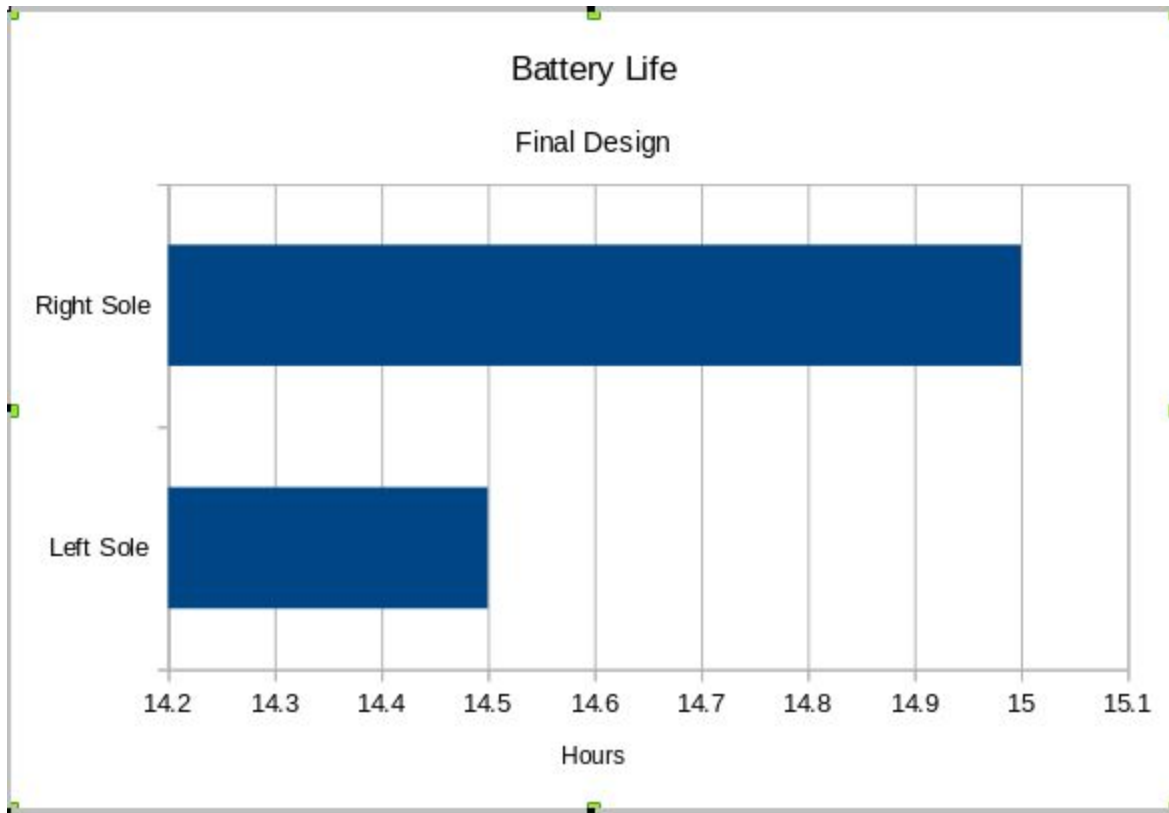


Figure 12: Final Design Battery Life Tests

Transmission Range

To test the transmitter range, the transmitter was placed and the receiver was moved away until loss of data or 100 feet was reached, which is 10 times the required range. Some possible sources of interference were placed near transmitter in subsequent tests, these include a structure, a bluetooth mouse and a bluetooth speaker with multiple modes. As you can see from Figure 13 below the bluetooth modules performed very well with the exception the speaker in party mode. This mode accepts all requests and some interaction between the speaker, the module, and the paired phone caused the connection to be dropped.

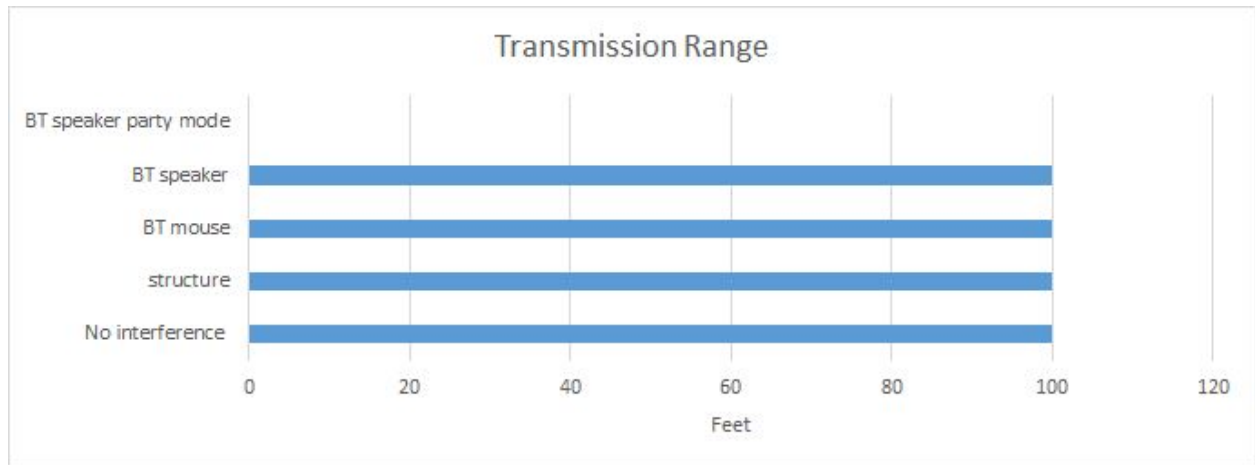


Figure 13: Transmission Range Test Results

Transmission range was tested for the Final design as well. We found that the range of the nRF51822 module did not have the same range performance at the HC-05 module. However because of all of the added features and lower energy consumption we feel that range is a fair trade, also the transmission power of the module can be turned up to extend range if required. Considering that our minimum range is 10ft we did not feel this was necessary. One interesting result is that devices that caused interference with the HC-05 did not seem to affect the nRF51822. Results shown in Figure 14 below.

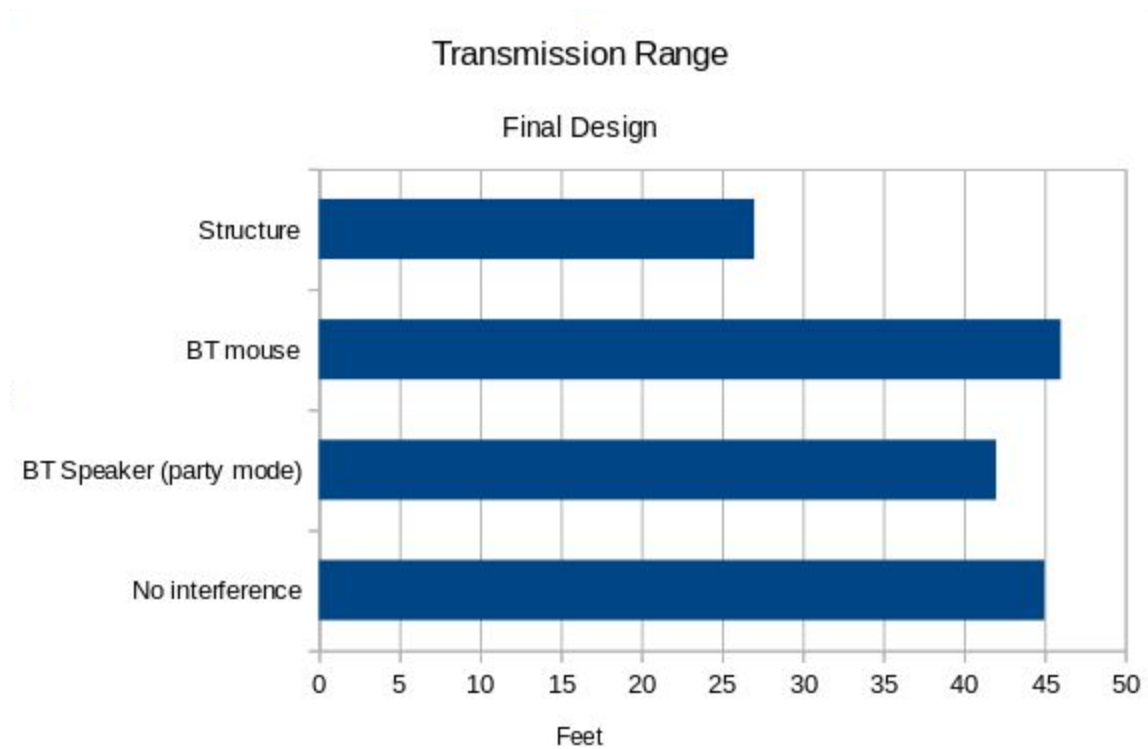


Figure 14: Transmission Range Final Design

Wireless Pairing

To ensure that there could be successful pairing and connection with all devices, we tested with Dell Latitude 6440, HTC 1 M8, Apple iPhone 6s, Macbook Pro, Motorola Moto G, Samsung Galaxy 5, Samsung Galaxy 6, Samsung Galaxy 7. All of these devices successfully paired.

System Analysis

In CPE 450 winter quarter, many of the prototype requirements were met. These included the implementation of a sensor communication system, which sends data from two sensors to a single Android device via single Bluetooth pairing, and the implementation of sleep cycles with interrupts

for power management. However, due to the difficulty of learning new applications, the software application requirement was not fully met. A bluetooth connection was made and a serial data stream is received on the phone, however the data is not displayed in a user friendly way, as it was in the previous iterations Java application. We decided to build the Android app using Android Studio which had a huge learning curve for the team. Every step along the way we encountered some sort of issue.

For the final design we implemented the backend in iOS which went much more smoothly. Bluetooth low energy integration on the hardware side was impacted by lack of library documentation. Also due to the division of labor integration between hardware and software was tricky and time consuming. One design flaw that should be pointed out is that the sensors used must be calibrated often this would require the user to move thru some set of possibly complicated instructions in order to use the device properly. Due to this we think that the sensors selected may not be the best fit for this application.

Management Plan

Aimee Davis is the Communications Officer, System Designer, Software Designer, System Interface, Product Reliability and Serviceability. Alex Bravo is the Project Manager, System Designer, Software Designer, Product Reliability and Serviceability. Anthony Childers is the Procurement Officer, System Designer, Hardware Designer, Product Reliability and Serviceability.

The estimated total budget for our initial design was about \$120 per sole, broken down in Table 13 below.

Table 13: Budget

Part	Quantity	Price
Lithium Ion Polymer Battery	1	\$5.00
Lithium Ion Polymer Charger	1	\$4.00
Enclosure	1	\$12.00
FlexiForce A301 Sensor	1	\$49.00 (4)
Dr. Scholl's	1	\$10.00
CP2101 USB->UART adaptor	1	\$2.99
HC-05 Bluetooth modules	2	\$3.99
Discrete Microcontroller ATtiny 84	1	\$1.99
Miscellaneous (op amps, filters etc).	NA	~\$20.00

The estimated total cost for our final Senior project design is about \$200 for a set of two soles.

Broken down in table 14 below

Table 14: Budget

Part	Quantity	Price
Lithium Ion Polymer Battery	2	\$6.95 each
FlexiForce A301 Sensor	6	\$11.47 each
SuperFeet Premium Insoles	1	\$50.00
Adafruit Feather 32u4 Bluefruit LE	2	\$29.95 each

Miscellaneous (3d print materials, wire, solder, etc).	NA	~\$10.00
--	----	----------

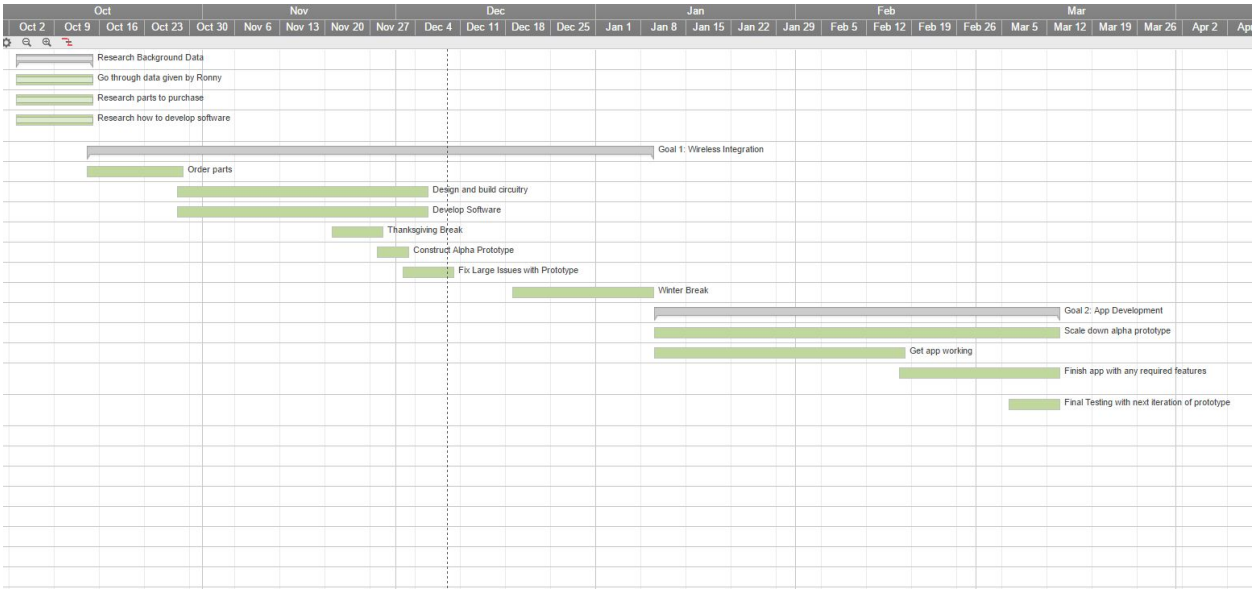


Figure 15: Gantt Chart (First 2 Quarters)

Task Name	Start Date	End Date	Duration
Research Background Data	10/03/16	10/14/16	10d
Go through data given by Ronny	10/03/16	10/14/16	10d
Research parts to purchase	10/03/16	10/14/16	10d
Research how to develop software	10/03/16	10/14/16	10d
Goal 1: Wireless Integration	10/14/16	01/09/17	62d
Order parts	10/14/16	10/28/16	11d
Design and build circuitry	10/28/16	12/05/16	27d
Develop Software	10/28/16	12/05/16	27d
Thanksgiving Break	11/21/16	11/28/16	6d
Construct Alpha Prototype	11/28/16	12/02/16	5d
Fix Large Issues with Prototype	12/02/16	12/09/16	6d
Winter Break	12/19/16	01/09/17	16d
Goal 2: App Development	01/10/17	03/13/17	45d
Scale down alpha prototype	01/10/17	03/13/17	45d
Get app working	01/10/17	02/17/17	29d
Finish app with any required features	02/17/17	03/13/17	17d
Final Testing with next iteration of prototype	03/06/17	03/13/17	6d

Figure 16: Gantt Chart Data (First 2 Quarters)

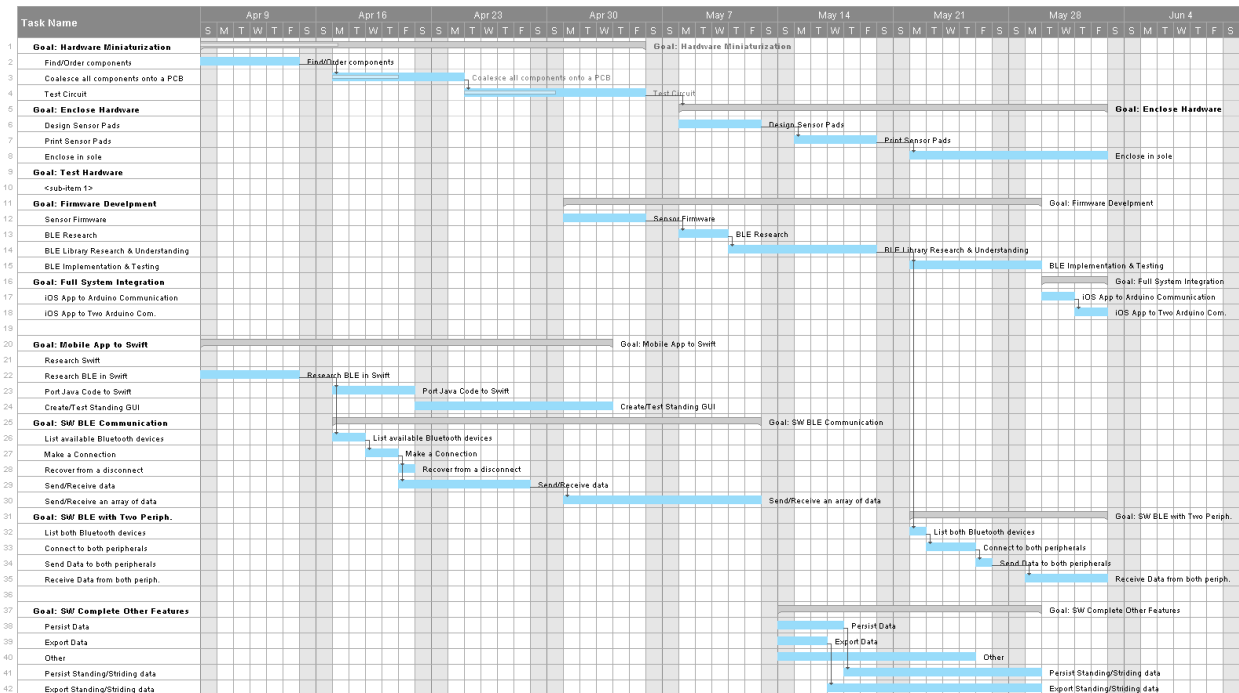


Figure 17: Gantt Chart (Spring Quarter - Senior Project)

Figures 15 and 16 above discuss the breakdown of workload over the course Capstone I and the first quarter of senior project. The first 7 weeks were broken into three weeks of design and four weeks of implementation. During the design phase the team was working together to gather background information, formulate a need statement, and develop use cases and personas to help guide the remaining 4 weeks of implementation. Upon the implementation phase, the team broke into two groups: one for hardware and one for software. During this time, the software team familiarized themselves with the code that Ronny had provided. The hardware team created steel casings for the sensors, assembled the prototype, and got bluetooth modules connected. At this point, the software team stepped in and wrote the programs to receive the serial data from the bluetooth module and present the data on a GUI. In the next quarter, the software team worked hard to develop an application for a mobile device to display the data that is currently being displayed on the computer. The bluetooth connection was established however the data was not

able to be displayed in a user friendly way as it was for the alpha prototype. The hardware team worked on getting data sent from two insoles, as well as implementing sleep cycle with interrupts for power management. In addition, hardware team designed a new sensor layout and cells for future iterations of the project.

Figure 17 discusses the workload over the course of the final quarter of senior project; the team is broken up into a software and hardware team. The hardware team focused on miniaturizing the existing hardware to fit within an insole and protecting that hardware, while the software team rebuilt the mobile application on the iOS platform.

References

Tran, Aurorae, Bo-Hsen Hsu, Ronny Elor, and Amanda Padilla. "Sense Yo Soles: Biofeedback Orthotic Insoles for Corrective Self-Treatment of Gait and Posture." Final Report. San Jose State University, 2015. Print.