OREGON INSTITUTE OF TECHNOLOGY

SMART STEPPER PLAN

PLAN BY:

Cari Blaker

Ismael Hernandez-Hernandez

Ian Gaffney

Zachary Elliott

TABLE OF CONTENTS

LISTOF FIGURES

LIST OF TABLES

1.0 Introduction

This document entails the detailed plan for the Smart Stepper 2023-2024 Junior Project for the Embedded Systems Engineering Technology Degree at the Oregon Institute of Technology. This plan breaks the design down into separate modules and submodules. This plan introduces and explains hierarchical block diagrams pertaining to each module and submodule. Each module's testing plan is included. Then, it details a list of parts, costs, and the project assignments to each member of the group project.

2.0 Modules

This section details the modules of the design plan for the Smart Stepper, and breaks down those modules into smaller submodules until every aspect of the design is described in depth. The Major Modules of this project are the Microcontroller, the SmartWatch, the SmartPhone, and the Encasement.

2.1 Microcontroller

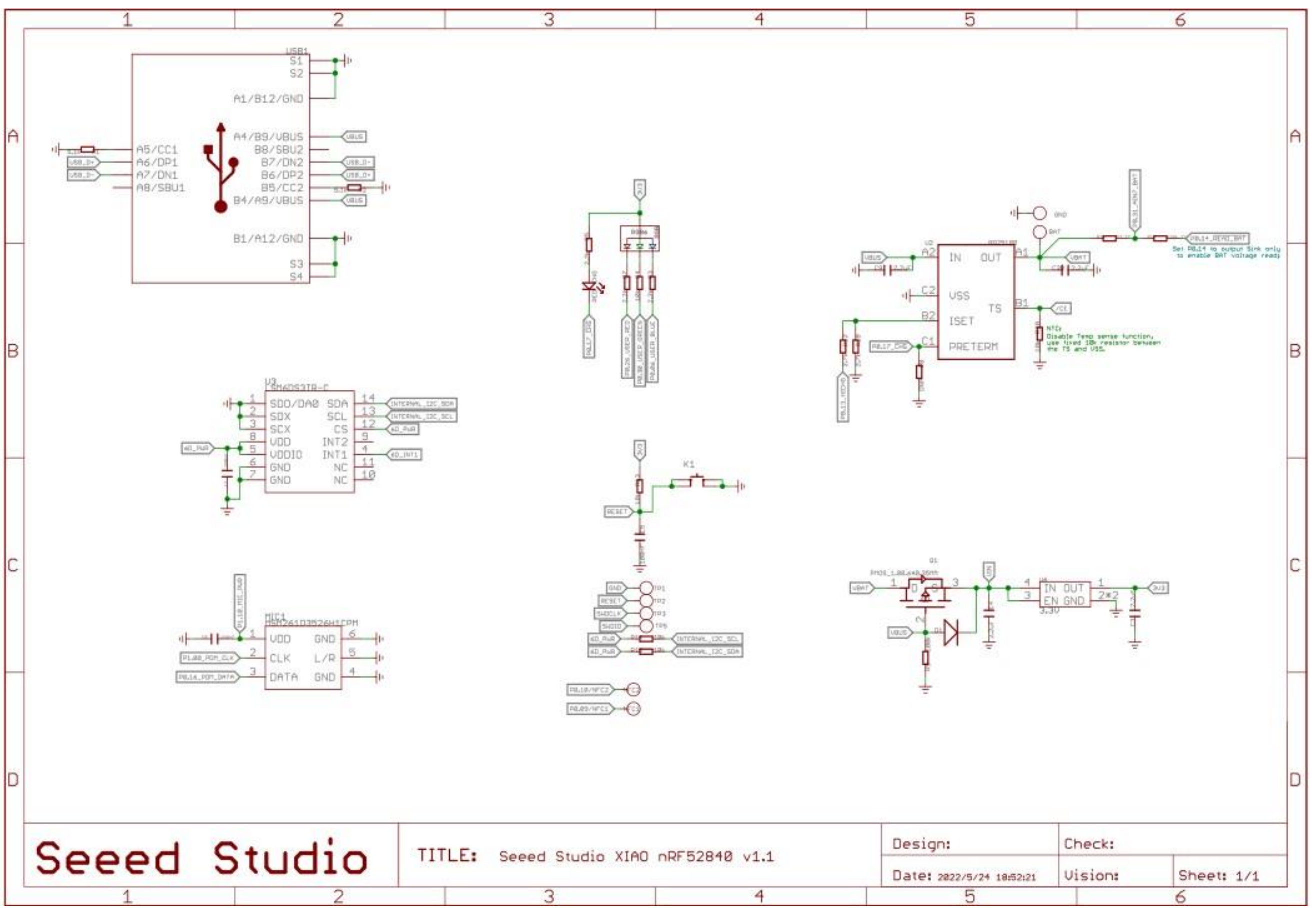

Figure 1: Seeed Studio XIAO nRF52840 Pinout Diagram

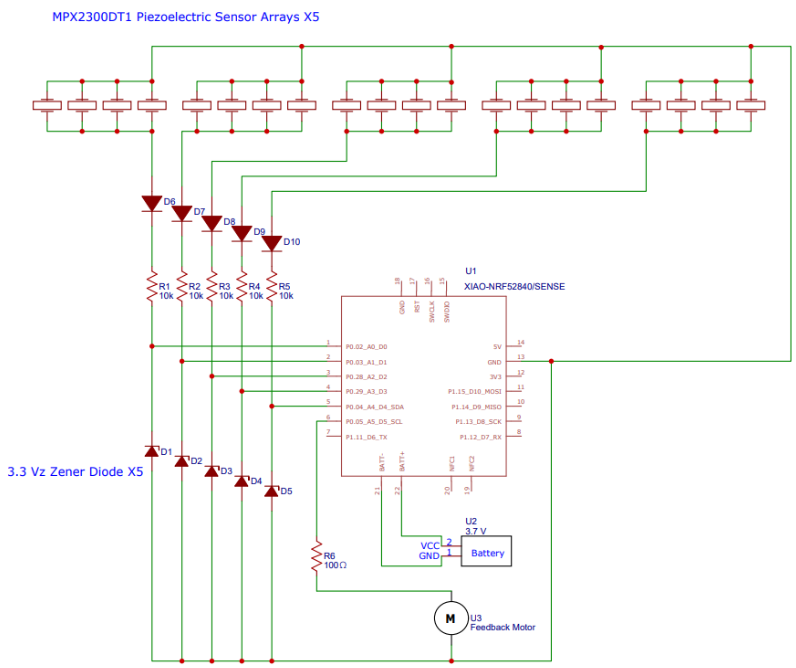

Figure 2: Hardware Connection Diagram

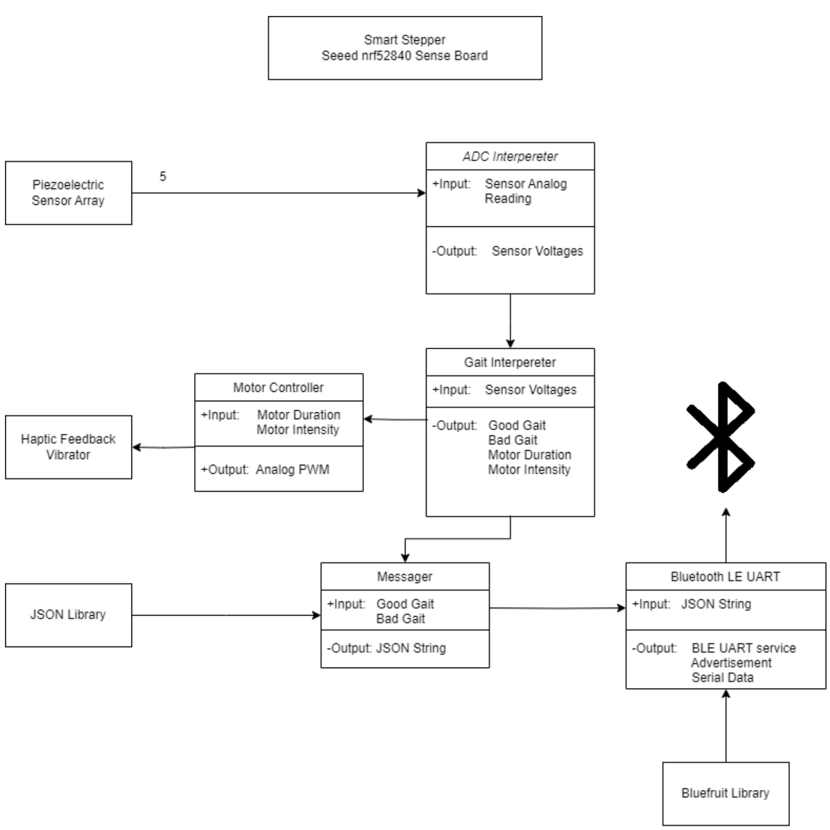

Figure 3 : Inside-Shoe Software Diagram

2.1.1 Microcontroller Power Supply

The system to be used is a LIPO785060 2500mAh 3.7V Lithium Ion Battery. It provides a nominal voltage of 3.7V. It provides a maximum voltage of 4.2V at a full charge with a minimum discharge voltage of 3.0V. It has a maximum constant charging/discharging current of 1500mA. It has a form factor of 50mm x 60mm x 7.3mm.

The battery is charged using the Seeed Board's onboard power management system. Power is delivered by a standard USB-C cable. The Vin is 5 volts, which is regulated down to 3.7V to charge the battery and power the board.

2.1.1.1 Design

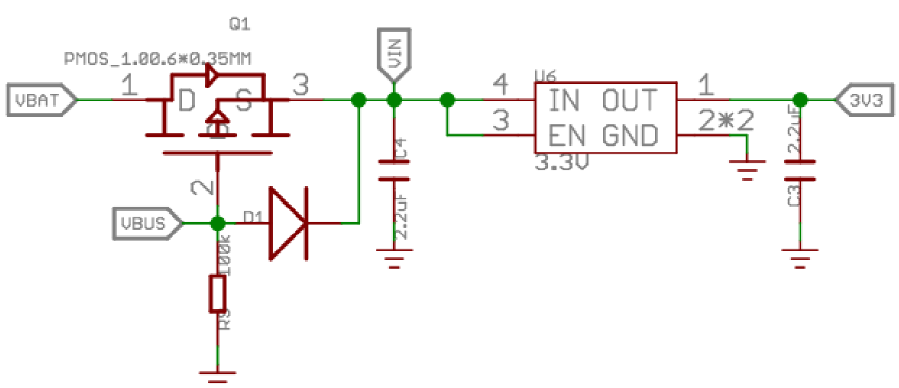

Figure 4 Power Supply Design

The built-in charging system connects the VBUS input from the USB-C port to the gate of a P-Channel MOSFET in enhancement mode, as well as a 100kΩ resistor (R9), which is in turn connected to ground. This ensures a large voltage across the resistor with minimal current to aid in controlling the gate signal. The VBUS also passes through a diode (D1), which in turn feeds into the source of the MOSFET. The drain is then connected to the battery. This section of the design allows recharging of the battery via USB-C.

The diode D1's output also connects to VIN, pins 3 and 4 of U6, and capacitor C4. C4 is a 2.2 µF capacitor, which is connected to ground, to smooth the potentially erratic voltage on VIN.

U6 is the voltage regulator. Voltage on pin 3 enables the regulator, and pin 4 is Vin for this component. Pin 1 is then connected to the 3.3V pin on the board, with another 2.2 µF capacitor (C3) connected to ground on its other lead—again, to smooth voltage.

2.1.1.2 TESTING

The power system's categories for testing are the battery's recharge times, discharge times, and the thermal characteristics thereof. The following tests in the following table need to be performed, and all results logged in the table. Each test must be repeated at least 3 times, and each time should be performed from a system starting from room temperature. These tests are to be repeated for any changes to the casing or shoe design, to ensure temperature remains in acceptable ranges. Multiple units should be produced to allow multiple tests to be performed simultaneously.

Table 1 Power Supply Testing

Test	**Time**	**Temperature at Start °C**	**Temperature at End °C**
Battery Discharge (Idle)			
Battery Discharge (0% PWM Motor)			
Battery Discharge (10% PWM Motor)			
Battery Discharge (25% PWM Motor)			
Battery Discharge Time (50% PWM Motor)			
Battery Discharge Time (100% PWM Motor)			
Full Recharge (System OFF)			
Full Recharge (System ON, Idle)			

2.1.2 Sensor Input

The Smart Stepper uses arrays of piezoelectric sensors wired in parallel to measure when pressure is applied to distinct parts of the foot. Each array is composed of four sensors, but there is no technical reason for any arbitrary number of sensors to be chosen within each array; so long as they are wired in parallel, any number may be used if testing determines an optimal number other than four. There are five arrays, each clustered to measure pressure in different parts of the foot.

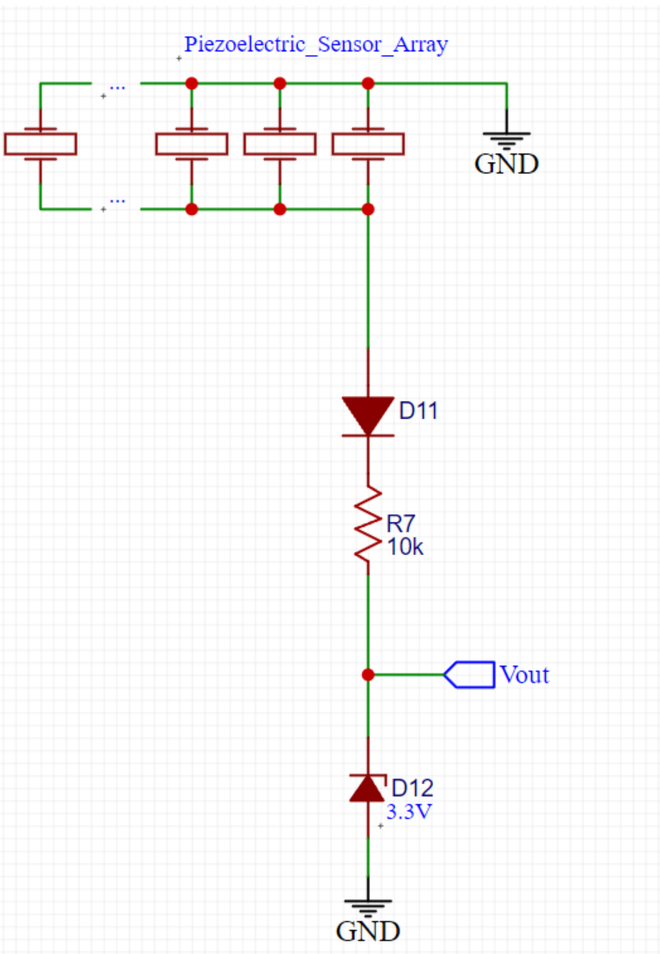

Figure 5 Piezoelectric Sensor Array

2.1.2.1 Design

Each sensor array is comprised of four piezoelectric sensors. The negative leads are wired to ground, and the positives are wired to a 10kΩ resistor in series with the anode of a diode. The cathode is wired to one of the analog I/O pins on the Xiao Seeed board, as well as a reverse biased Zener diode with a Zener voltage of 3.3V. The Zener diode's anode is then wired to ground. This arrangement limits the current and voltage for the input pins to safe levels.

The 10kΩ resistor limits any current created by the piezoelectric sensor array to safe levels; in prior testing, the piezoelectric sensors could generate up to 40 volts. This is an outlier to normal operation, but a 10kΩ resistor ensures that a maximum 4 mA current flows through these components. The Zener diodes ensure the voltage at the analog input pins never exceeds 3.3 volts. This is the maximum safe nominal voltage rating of the I/O pins.

2.1.2.2 Testing

The testing protocol for this section requires pressure to be applied to each sensor individually, and the voltage at the input pins to be measured. Each sensor is tested for two things; One, that they generate a measurable voltage at this point, and two, that this voltage never exceeds 3.3 volts. Additionally, the current coming out of each array when all sensors are triggered is measured to ensure no unexpected current spikes. The following checklist template should be filled out to verify all necessary initial testing has been done.

Table 2 Sensor Input Testing

	Sensor 1V at I/O pin	Sensor 2V at I/O pin	Sensor 3V at I/O pin	Sensor 4V at I/O pin	Max Array Current	Max Array Voltage
Array 1						
Array 2						
Array 3						
Array 4						
Array 5						

2.1.3 Sensor Input Processing

Sensor data processing occurs in two submodules. The first module is the ADC converter module, which reads the five analog inputs and converts the readings into a digital format for use by the Interpreter. The second module is the Interpreter, which reads this data from the ADC and correlates the timestamps when the pressure is applied, using the information to determine if a step was a toe- or heel- strike. The interpreter then stores the time and step information in memory for later use by other modules.

2.1.3.1 ADC Converter

The ADC interpreter reads a voltage between 0V and 3.3V from the following pins: P0.02, P0.03, P0.28, P0.29, P0.04. These pins, in order, are referred to by the Seeed board as A0 through A4. These pins have a maximum safe voltage of 3.3V.

The ADC converter is called from the main loop. It stores the current voltage values of each of the pins locally, with publicly available accessors to view this data. The purpose of this module is simply to log the current state of the pins and make this data available temporarily.

Testing of this sub-module is entirely comprised of providing variable voltages between 0 and 3.3V and recording that these are properly logged.

2.1.3.2 Gait Interpreter

This module is called by the main loop just after the ADC Interpreter. It monitors stored data therein, and once the rising edge of the input waveform is seen, this data is logged with a timestamp in a local struct, which is then pushed to one of five stacks, based upon which array was measured. In a healthy gate, each piezoelectric sensor array triggers sequentially, first at A0 and lastly at A4. In the event that the logged data breaks this pattern, for example by skipping one of the arrays, then it will invoke a function of the Motor Controller, instructing it to activate the motor for one second and at the minimum intensity. Each time this pattern is not matched, the incident is logged, and intensity increases by 10%. On a pattern match for a healthy gait, the intensity value is reset to 10%.

After the Gait Interpreter has measured a step, good or bad, it clears all sensor reading related to that step from the stacks, and pushes a timestamp to one of two additional stacks, representing the time the user took a good step or bad step. These are popped from the stack by the messenger module and communicated from there to the rest of the system.

The Gait Interpreter (GI) is broken down further into three sub-modules: Interpreter Algorithm, Data Storage Stacks, and Communication Module.

2.1.3.2.1 GI Algorithm

The Gait Interpreter Algorithm Submodule interprets the data stored locally. It accesses the data on the Data Storage Stacks submodules, decodes it by checking the associated timestamps on each data point, and determines if it represents a good step or a bad step. Once it has decided, it logs the current time and pushes this time into the GoodStepsStack or the BadStepsStack.

This sub-module requires the most testing of the Gait Interpreter System. Initial testing can be performed with dummy data to verify correct classification of steps, but further testing must wait for integration of each module.

2.1.3.2.2 GI Data Storage Stacks

The Data Storage Stacks Submodule encompasses all the stacks within the Interpreter Module. There are five sensor array stacks, storing timestamps where the ADC module logged the sensor array output crossing a trigger threshold. There are two additional stacks, storing the timestamps that the Algorithm submodule decoded a good step or a bad step: GoodStepsStack and BadStepsStack.

This module's testing ensures that the GoodStepsStack and BadStepsStack can be accessed by the messenger module, but this is a fairly non-volatile system.

2.1.3.2.3 GI Communication

The Communication Submodule reads the current voltages from the ADC module, and if they have crossed the minimum voltage threshold, the timestamp is logged in the Storage Stacks. The communication submodule logs the timestamp when the voltage crosses from below the threshold to above it. Currently, the threshold is set at .7 volts, though this may change during testing.

Additionally, once a step has been logged to GoodStepsStack or BadStepsStack, the communication module either resets the vibrator intensity to minimum or invokes the Motor Control Module to trigger the Haptic Vibrator for one second at the current intensity, then increment the intensity value. The intensity will start at 10% and increment by an additional 10% with each consecutive bad step.

Tests for this module measure correctly retrieved data from the ADC Interpreter module, properly invoked Motor Control Module, and properly escalated/reset intensity based on steps. Additionally, while this module dictates the duration and intensity of the haptic vibrator, these values are subject to change based upon further testing of optimal duration-intensity ratios. See section 2.1.4.2.2.

2.1.3.2.2 Gait Interpreter Testing

Testing of the Gait Interpreter will be the most important and extensive suite of testing. Ensuring that the Smart Stepper properly categorizes steps as either good or bad will be a significant challenge and will require extensive testing to that end. We currently do not possess an optimal algorithm for classification of steps from this data, and so a significant amount of development time will be spent manually testing the Smart Stepper in various environments to ensure quality and accuracy. This testing is also the production phase most likely to invoke changes in design, as this represents core functionality. Any other component, if it is causing misinterpretation here, can and will be changed if testing at this stage reveals a conflict.

The actual testing will comprise of having various volunteers walk in the shoe while being recorded by one of the team. Initially, we will visually confirm the time of each step, and if we would classify it as good or bad. Then we will check this data against what is reported by the interpreter, and optimize based upon the results.

2.1.4 Microcontroller Output

The microcontroller has two methods of providing feedback to the user. The first is the Haptic Vibrator, and the second is via Bluetooth LE transmission.

2.1.4.1 Motor Control Module

The Haptic Vibrator is PWM-controlled, so that the strength of the vibration can be adjusted. When the Gait Interpreter Module identifies an undesired pattern of input (indicating a toe-strike), it instructs the Motor Controller Module to begin PWM control of the Haptic Feedback Motor at a certain percentage duty cycle, for a specific number of seconds. Then, each time the Motor Controller Module is called from the main loop, it checks how many milliseconds have passed since it was last called to determine if it should swap the voltage on A5, until Motor Duration's value of time has passed. Once the time has elapsed, it returns A5 pin to its low state. If another instruction is received for a new duration and intensity, it overrides the old values and immediately begins Pulse Width Modulating the A5 pin at the new duty cycle, for the new duration.

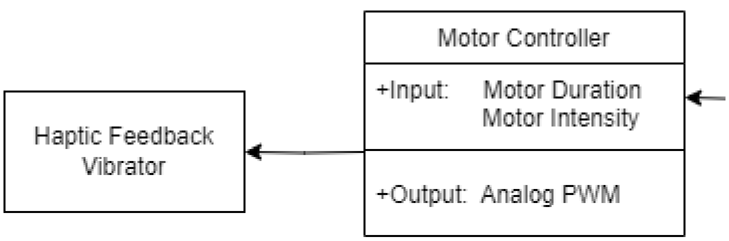

Figure 6 Motor Control

2.1.4.1.1 Motor Control Module Testing

Testing this module is comprised of hooking the A5 pin to an oscilloscope, passing the Motor Controller module various test values, and recording the duration/duty cycle of the resulting square wave. The following chart represents the minimum testing required.

Table 3 Haptic Vibrator Testing

Motor Duration	Motor Intensity	Expected HIGH time	Expected LOW time	Measured HIGH time	Measured LOW time	Measured Signal Duration
1	0.0	0 ms	100ms			
1	0.3	30 ms	70 ms			
1	0.6	40 ms	60 ms			
2	0.2	20 ms	80 ms			
3	0.9	90 ms	10 ms			
4	0.5	50ms	50 ms			
5	1.0	100ms	0 ms			

2.1.4.2 Haptic Vibrator

The haptic vibrator gives the user immediate feedback if they do not perform a proper gait. It vibrates to remind the user to correct their gait.

2.1.4.2.1 Design

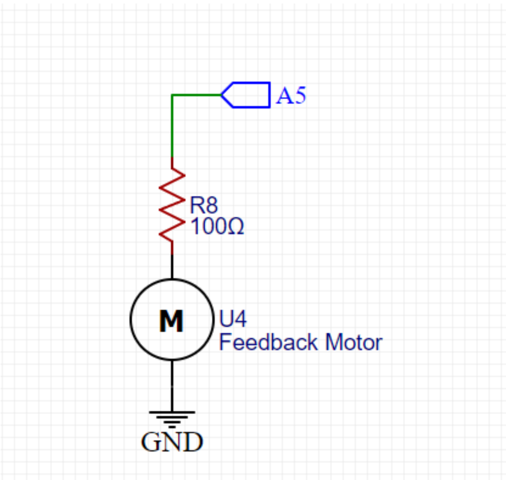

Figure 7 Haptic Vibrator Schematic

This component is a simple vibrator motor wired in series with a 100Ω resistor to the A5 output pin of the Seeed board. The 100Ω resistor is to ensure the current stays below the maximum current draw the analog pin is capable of providing.

2.1.4.2.2 Testing

Testing of the haptic vibrator has two major components. The first tests the vibrator at various strengths via PWM. The second tests are for the vibration strength and length of time themselves. This portion of the design should heavily involve the off-campus team, especially Dr. Marybeth GrantBeuttler, as she is the physical therapist, and her opinion/expertise/preference of how long/strong this vibration should be is paramount. Initial testing for the vibrator involves feeling its vibration on skin. Stopwatches measure length of time, and intensity is simply felt. Time only needs to be tested initially. The next round of tests must be performed with the haptic vibrator underneath the 3D-printed sole, and any cushion the design requires between the sole of the foot and the haptic vibrator. Testing is feeling the vibration through the foot. A final round of testing must be performed on the final product—with a fully encapsulated design inside of a sneaker, with all members of the team testing—wearing socks. Each of these phases of testing are

2.1.4.3 Bluetooth Transmission

Bluetooth transmission sends gait data from the microcontroller to the Smart watch, indicating a heel strike or a toe strike. The data is sent by the microcontroller via Bluetooth LE, so that both microcontrollers (one per shoe) may be transmitting via the same mesh connection at once.

2.1.4.3.1 Design

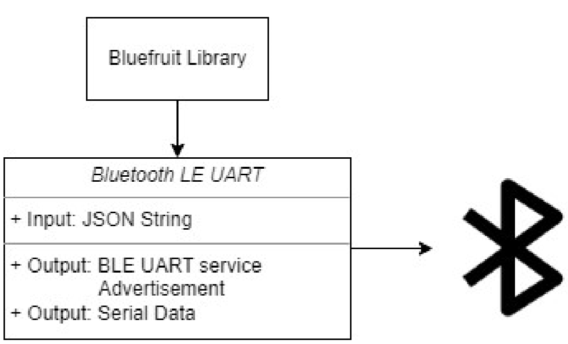

Figure 8 Microcontroller Bluetooth Transmission

The seeed Studio microcontrollers have built-in Bluetooth capabilities. The Bluefruit libraries provided by Adafruit/Arduino are implemented by the design to use these capabilities. The microcontrollers transmit data via Bluetooth LE UART. The data sent over the UART is formatted as JSON strings; JSON is easy to interpret and easy to read.

2.1.4.3.2 Testing

Initial testing is performed by connecting to the BLE UART using the Bluefruit Connect phone app to see the data sent by the microcontroller. The data is compared, if accurate, it is then changed, and the process is repeated to confirm the transmission's accuracy. Once the connection has passed this test for both microcontrollers per design, when the Smartwatch is configured to receive data over the BLE UART, each microcontroller is tested individually in tandem with the smart watch testing. Once the BLE connection is made for all three devices, it is tested with controlled dummy data. Then, purposeful toe/heel strikes are made using the sensors, and the received data is compared to the real input.

2.2 Smartwatch

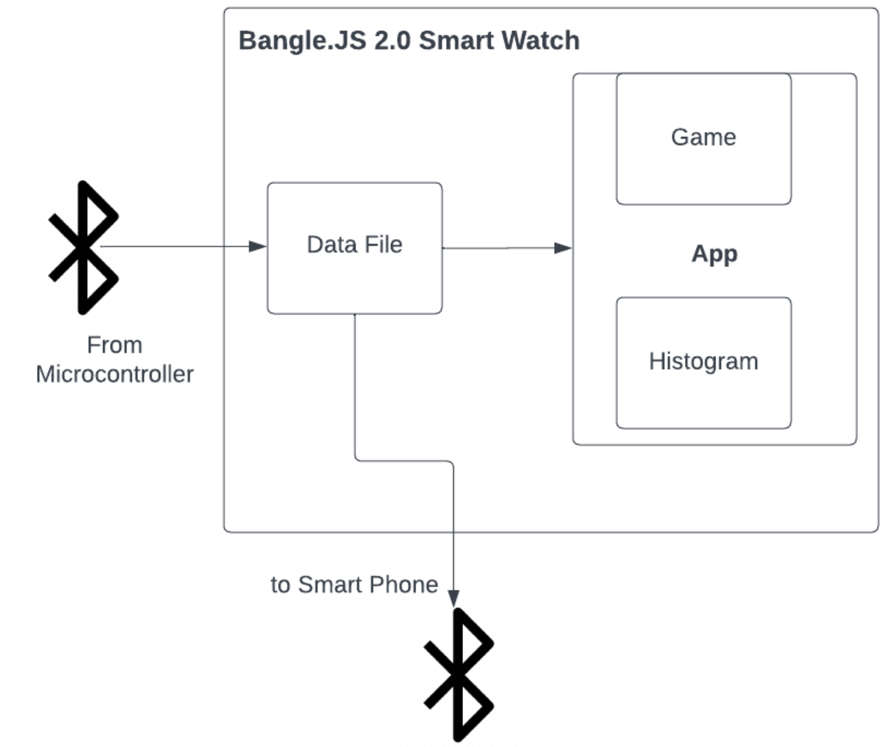

Figure 9 SmartWatch Softwarer

2.2.1 Bluetooth Reception

The Smartwatch is configured to receive data over the BLE UART, mesh connecting to both microcontrollers.

2.2.1.1 Design

The Bangle.js is configured to connect to both Microcontroller BLE UART's at the same time. It reads in the JSON strings sent over the UART, interprets them into a file, and then stores the data sent in the Bangle.js' on-device storage. This data is then used by the Smartwatch app and sent to the user's phone for longer term storage and analysis.

2.2.1.2 Testing

Testing is performed by connecting to the BLE UART using the Bluefruit Connect phone app to see if data can be sent to and displayed by the Bangle.js watch. Once this accuracy is confirmed, the connection is made to each microcontroller, which send dummy data to the watch. Once this accuracy is confirmed, the mesh connection is made between all three devices, where new dummy data is transmitted by the microcontrollers, and its received accuracy is confirmed in the smart watch. Finally, the accuracy must be tested with the sensor input, comparing a controlled number of heel strikes vs. toe strikes with each foot.

2.2.2 Smartwatch App

The Smartwatch App has two functionalities. One is a game, to encourage the user to take more heel strikes. The other is a graph of the week's heel strikes vs. toe strikes.

2.2.2.1 GAME

The SmartWatch app's game functionality is meant to encourage children to take more heel strikes. 2.2.2.1.1 Design

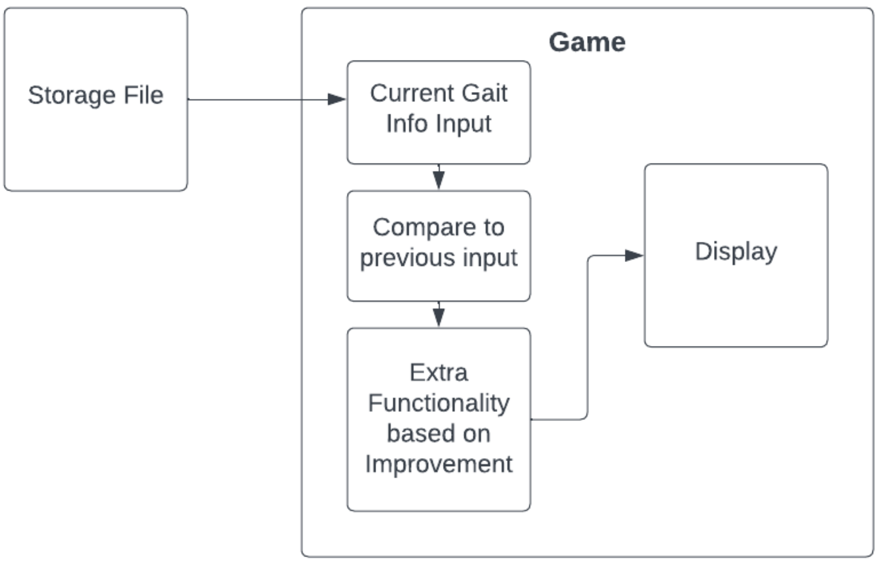

Figure 10 SmartWatch Game

The game takes its input from the file stored in memory, reading in all of the data since its last read. Comparing that data with previous data, the app determines how much progress has been made, and allows the user extra functionality (or not) based on progress (or lack thereof). This data is then used to update the graphics for display to the user.

2.2.2.1.2 Testing

The game is to be tested throughout its creation, using dummy data until the game is fully functional. Testing is done to be sure that when there are more toe strikes than heel strikes in comparison to the ratio of the last data read, no added functionality is given. And testing is done to be sure that the opposite is true, also. Once the game is fully functional, and Bluetooth connection is completed, testing is performed using the Bluetooth transmission of the microcontroller. A controlled amount of “heel strikes” is made using sensor input, and a controlled amount of “toe strikes” is made using sensor input. This control information is used to test the game’s functionality.

2.2.2.2 Graph

The graph submodule is a histogram, displaying a week’s worth of heel strikes vs. Toe strikes per day.

2.2.2.2.1 Design

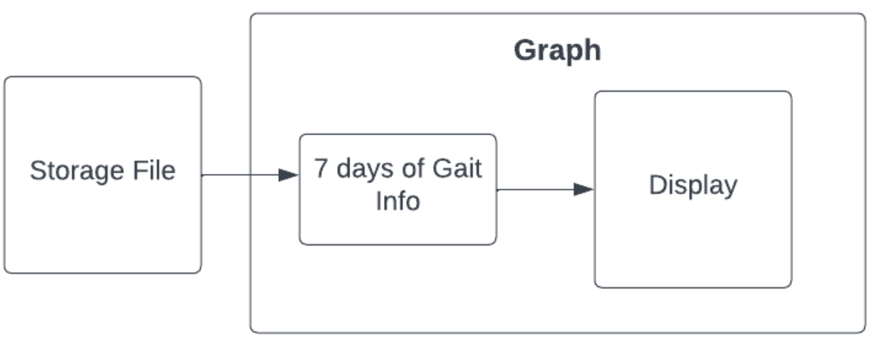

Figure 11 SmartWatch Graph

The graph portion of the app reads in seven days of data from the storage file in the watch. This data is used to display a histogram of the steps taken for each day, left foot on the left, right foot on the right for each day. The bar is red for toe strikes, and green for heel strikes, with the bar of whichever is greater (toe- or heel- strikes) for the day on top. This graph is displayed with the dates, so the user can see their progress over the last week.

2.2.2.2.2 TESTING

The current version of the graph uses the temperature sensors from the watch. Once the graph is displaying weekly left and right heel strikes vs toe strikes, the temperature can be divided into Celsius and Fahrenheit and displayed for each day and each foot to ensure that the graph is functioning properly. Once it is, dummy data can be entered an used to further ensure accuracy, using some dummy data that shows more toe strikes than heel strikes and vice versa, and different numbers for each day and each foot. Once the graph is proven to pass these tests, it can be hooked up to the real storage data file. The app can be modified to display over the course of seventy minutes, so that short-term testing can verify the functionality of the graph. The sensors feeding the microcontroller then must be forced to transmit a specific number of heel strikes vs. toe strikes for each foot over the course of that hour and ten minutes. The displayed information must match this control data to pass the test. The app must be modified to its intended week-long display, and the process must then be repeated with an entire week's worth of data.

2.3 SMARTPHONE APP

The smart phone app is getting developed by the off-campus portion of the Smart Stepper team. Its formal design won't be determined by this plan—however, a general outline of its expected functionality follows, as well as a testing plan.

2.3.1 DESIGN

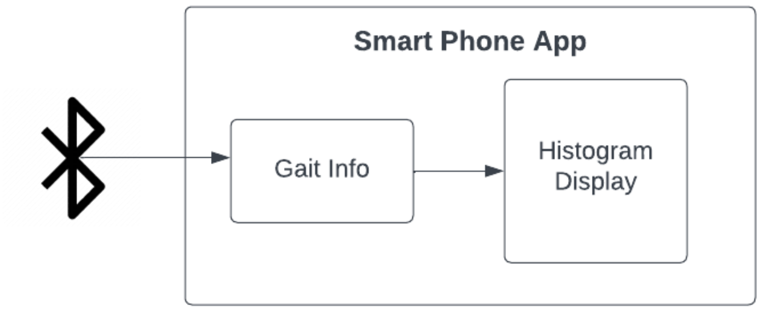

Figure 12 SmartPhone App

The Smart Phone connects via Bluetooth to the Smartwatch and collects/stores all of the information since the previous transmission. This information is displayed in a histogram which is the toe-strikes vs. heel-strikes as a measure of time, over the lifetime of the device.

2.3.2 TESTING

Initial testing is performed outside of this plan—however, final testing to ensure accurate data transmission and display must be done. All of the smart watches at hand are connected to each phone for which the app is available. A year's worth of data is sent to ensure accurate display, then the maximum amount of real data on hand is sent and compared for accuracy.

2.4 HARDWARE ENCASEMENT

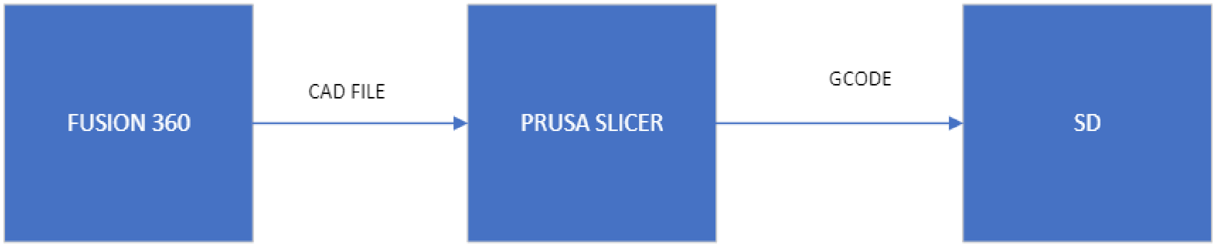

Figure 13 3D Printing Procedure

2.4.1 DESIGN

In the design phase, Fusion 360 plays a crucial role in the creation, editing, and improvement of 3D models for insoles. The iterative process involves constant refinement until the insole design reaches its final state. Subsequently, Fusion 360 is used to generate an STL (Stereolithography) file, encapsulating the 3D geometry and making it compatible with 3D printing software.

Moving to the Prusa 3D printing module, the next step involves importing the STL file into Prusa Slicer, a specialized slicing software for the Prusa 3D printer. Within Prusa Slicer configures slicing parameters such as layer height, infill density, and support structures. The software then translates the design into G-code, a language understood by 3D printers.

The final configuration step involves transferring the G-code file to the Prusa 3D printer.

2.4.2 TESTING

The team begins by executing the 3D printing process using the Prusa 3D printer to create a physical prototype. In addition to structural accuracy, the team focuses on calibrating infill density and temperatures during testing. This iterative process allows for optimization of the balance between strength and material usage. Functional tests are conducted to assess aspects like comfort, flexibility, and compatibility with footwear, ensuring the insoles meet their intended purpose.

3.0 PRELIMINARY COST

The following table lists the individual preliminary costs and the total preliminary cost of the Smart Stepper.

Table 4 Costs

Product	Units	Price Per Unit	Cost
Bangle.JS 2 Smart Watch	3	$93.22	$279.66
Seeed Studio Microcontroller	2	$15.99	$31.98
3.7V Lithium Ion Battery	2	$14.95	$29.90
Piezoelectric Sensor	10	$0.79	$7.99
Haptic Vibrator	2	$1.20	$2.40
Shoes (pair)	1	$29.99	$29.99
Filament TPU Roll	1	$52.99	$52.99
Total Physical Cost			**$434.91**

Each part has been ordered, received, and tested by the team, except for the shoes themselves. The Junior Project Team's partner will be deciding on the appropriate shoes. Once the team is ready to assemble all portions of the project, Dr. Marybeth GrantBeuttler is to provide the shoes she has selected.

4.0 TEAM ASSIGNMENT

Table 5 Team Roles

PERSON	PROJECT PORTION
Cari	Software: Smart watch apps
Ismael	Hardware: 3D printed sole Encasement design
Zach	Hardware/firmware: Piezoelectric sensor input/design/processing Haptic vibrator connection
Ian	Firmware: Bluetooth connection