ECE 2799 - Habit Helper

Team 14

Homework 6 - Final Report

April 30, 2019

Responsible Engineers

Fivos Kavassalis

Faith Kurtz

Catherine Roberts

Course Managers

William R. Michalson

Jennifer Stander

Table of Contents

Table of Figures

1. Introduction

Our objective is to design a wearable health device which aims to help the user live a more healthy lifestyle. The product will aid the wearer in correcting poor posture and low physical activity. This will be achieved by collecting measurements and offering the user helpful reminders and statistics about their habits. The user will also be able to view their daily habits and long term progress through a mobile phone application. The device will be worn over their shoulders with evenly distributed weight to keep it secure. It will contain an IMU sensor to detect “slouching” (in other words poor posture) and movement (or lack thereof).

1.1 Problem Statement

Long hours of physical inactivity such as sitting in front of our computers is a modern day menace that threatens the physical well-being of many of us, irrespective of age or gender. The so called “modern living style,” the evolution of the global economy and the consequent development of computer-based professions is at the root of this problem. At the same time, more and more people are aware of the threats posed to their health precisely because of this life-style and are prepared to make an effort towards changing it. Our device is designed to provide a step in this direction.

1.2 Market Research

In this section, we will discuss our market research. This will include the methods used to perform the research, the relevant results, and the conclusions we have drawn from these results.

The two methods used to perform market research were internet searches and surveys. First, we looked at the market for similar products and verified that our design was not an exact copy of an existing design. Our findings were then used to create a market research survey specific to our project. The resulting survey gave us the majority of our market information.

Our first step was to search the internet for any similar products. We found some devices with similar functions, such as a product to notify the user when they are slouching, but no products had the exact same functionality as our proposed device. For example, the “slouching” device stated above does not measure physical activity, which is something that we are planning to do.

The survey was the most useful tool for performing our market research. We surveyed 51 individuals of various ages and genders using Google Forms as a platform. We gathered data on the following topics:

- Hours spent sitting per day
- Interest in the product
- Interest in correcting various habits
- Desired features
- Inhibiting factors
- Expected cost

The survey yielded useful information which helped us determine our market. The results also guided us in defining which features were most important to potential users.

The market research survey also produced very useful results. As shown in Figure 1, the majority of individuals surveyed expressed high interest in the product. On this scale, 5 is very interested and 1 is not at all interested. Color represents interest and the numbers on each section represent the number of people.

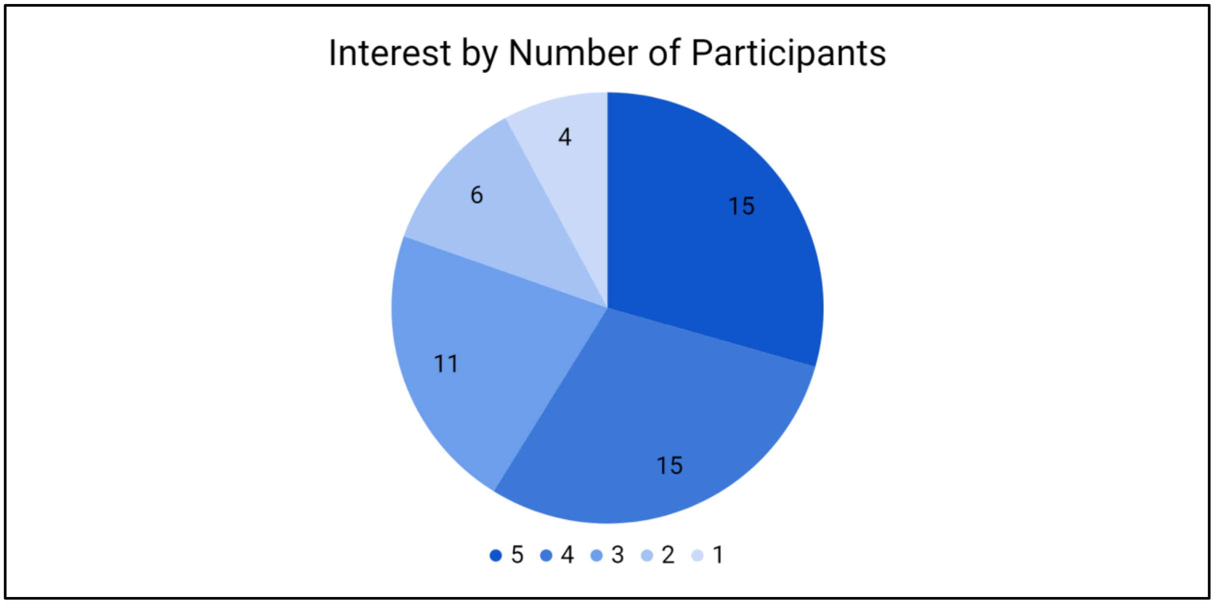


Figure 1: Interest Level by Number of Participants

In addition, there is a clear trend between the number of hours spent sitting each day and interest in the product. As shown in Figure 2, individuals who spent more time sitting each day expressed greater interest in using the product.

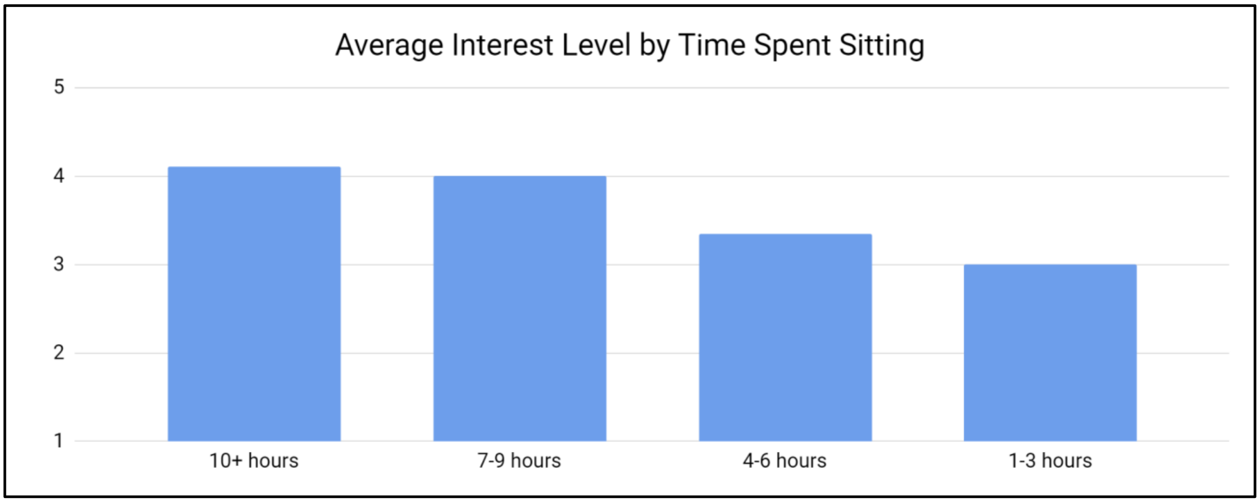


Figure 2: Average Interest Level by Time Spent Sitting Per Day

Another important result from the survey was the expected cost of the product. Figures 3, 4, and 5 show what potential users expected to pay for the product based on their interest level. High interest is an interest score of 4–5 out of 5, mid interest is 3 out of 5, and low interest is 1–2 out of 5. These graphs are based on 50 of the 51 responses collected because one respondent did not specify an expected price.

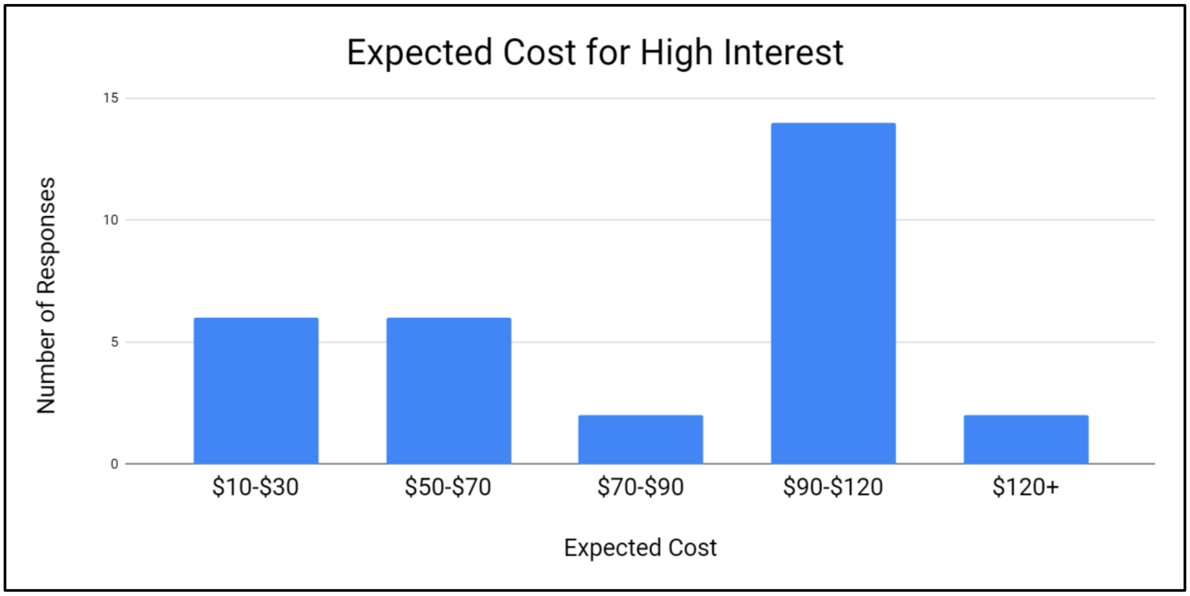


Figure 3: Expected Cost for High Interest Individuals

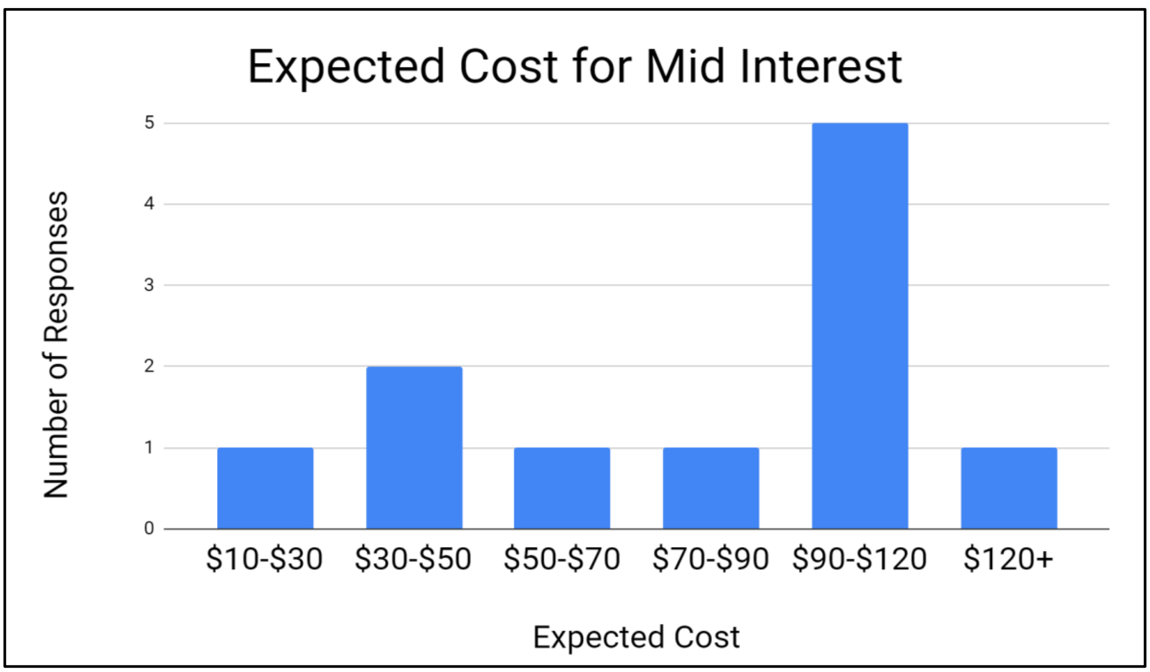


Figure 4: Expected Cost for Mid Interest Individuals

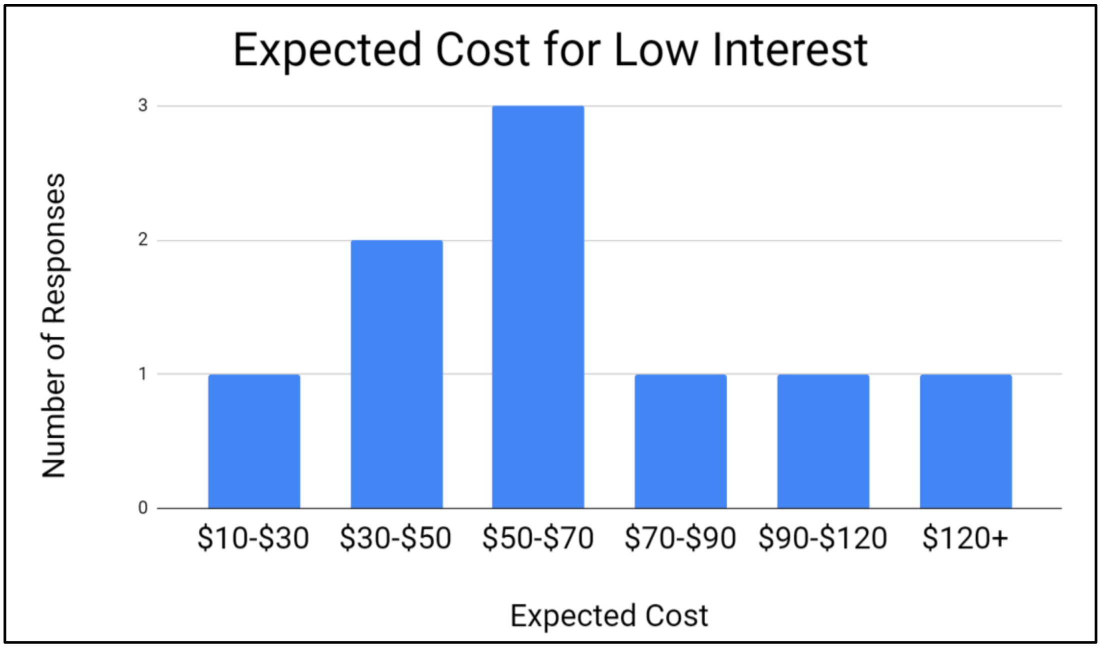


Figure 5: Expected Cost for Low Interest Individuals

For mid and high interest individuals, who would be our primary market, the typical expected product cost is $90–$120.

The interest levels that were measured with our survey shows that people who sit long periods of time everyday would be very interested in our product. As a result, a promising market would be people who have sedentary careers like office or computer jobs and students.

In addition, within this market there is a big push for corporate wellness in many companies across the country. In fact, the official journal of the American College of Sports Medicine's November report placed wearable technology as the top priority for corporate wellness in 2019. Many businesses are introducing wellness programs for their employees as a way to lower the amount of money spent on health insurance. Many of these programs have benchmarks that employees have to meet in order to receive certain rewards. In order to prove that these benchmarks have been met, companies have been investing in wearable health technology that provide reliable data about an employee's progress and provide many other health benefits. Price Waterhouse Cooper predicts that over 75 million wearables will have entered offices worldwide by 2020. Companies such as UnitedHealthcare, Humana, Lowes, Coca-Cola, Ford, and Air New Zealand have already purchased wearable technology for their employees. The wearable market is growing rapidly and there is an abundance of money in it to make reliable revenue with our new and versatile product.

Collectively, the results of our market research have led to the following conclusions.

- The overall interest in this product is fairly high, suggesting that there is a viable market for this product.
- The target market of this product consists mainly of individuals who spend a significant amount of time each day sitting and companies who want to purchase wearable technology for their wellness programs.
- $90–$120 would be an appropriate price for the product.

2. Product Research

2.1 Customer Requirements

The information derived from our market research not only provided an understanding of the extent of consumer interest for our product, and therefore an indication of whether a market for our device exists, but it also clarified the specifics of what customers are looking for. The following are the customer requirements for our device:

- Durable
- Long lasting battery life
- Cost of approximately $90–$120 (or less)
- Small and compact
- Comfortable
- Aesthetically pleasing
- Lightweight
- Accurate
- Easy to use
- Waterproof

Meeting as many of these requirements as possible should increase customer desire for the product.

2.2 Product Specifications

Based on the value analysis for our specific module design options, we plan on using the following components in our design:

- Case: Crescent shaped flexible case placed around the neck from behind
- Sensors (Posture and physical activity): Adafruit BNO055 Absolute-Orientation Sensor (IMU)
- User input (on/off): SPDT Slide Switch
- User notification system: Vibrating Mini Motor Disc
- Microcontroller: Arduino Nano Microcontroller
- Wireless Transceiver: HC-05 Wireless Bluetooth Serial Transceiver Module
- Power source: Two 3.7V 4800mAh rechargeable lithium battery cells connected in series (80 hour battery life-5 days at 16 hours per day)

We have chosen to use the Adafruit BNO055 Absolute-Orientation Sensor as an IMU in our prototype because it meets all of our requirements and we already have access to one. For the actual product, different sensor(s) could be used which might be a more optimal design choice. All of these modules are located inside the crescent shaped case shown in Figure 6.

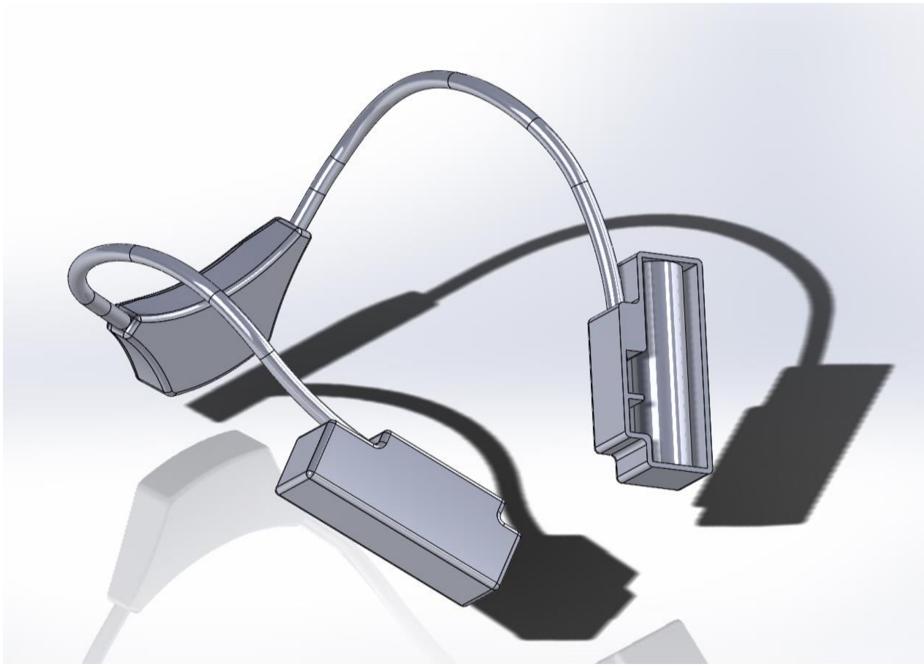


Figure 6: Case Design

The device's functionality is analytically explained in the flowchart below (Figure 7).

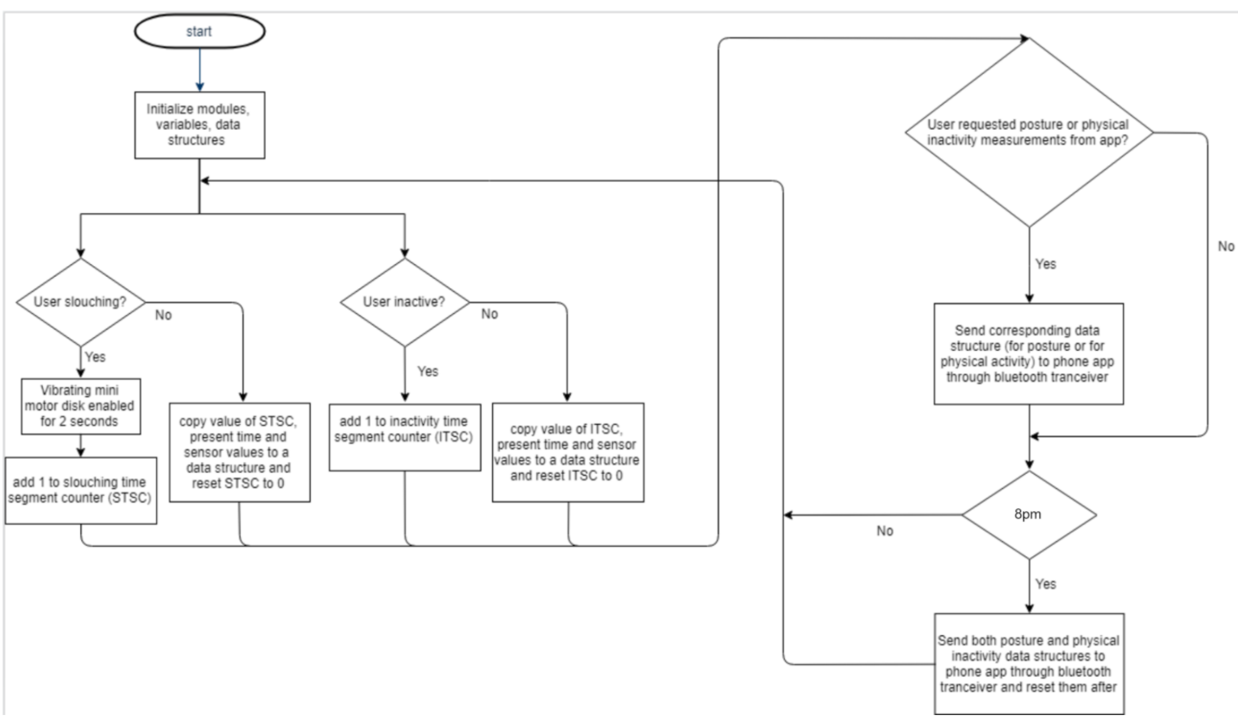


Figure 7: Device Flowchart

With reference to the implementation of the app, we created a Bluetooth Terminal by establishing a link between the user's phone and the HC-05 module connected to our device. Through this terminal, the user will be able to see the amount of time they have been slouching and/or have been physically active any time of a given day. This information will also be displayed in total time in minutes and with graphs for visual feedback. Figure 8 below is a flowchart displaying our app's functionality.

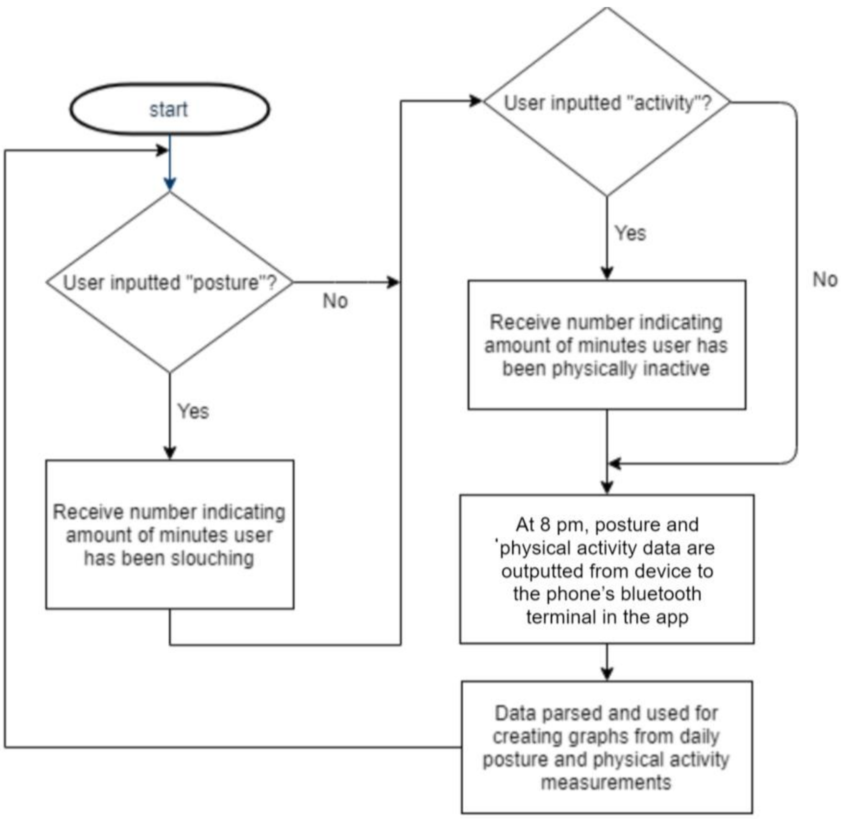


Figure 8: Phone Application Flowchart

2.3 Competitive Value Analysis

The companies who are also selling products attempting to correct posture and/or increase physical activity span the range of very established and reputable products to some that are not yet available to purchase. In order to accurately compare our product with the existing competition we evaluated each one using our customer criteria. Each of the sections is graded on a scale from 1 to 10 with 10 being the best and 1 being the worst. The only exceptions here are the cost scale and battery scale. Cost is still graded from 1-10, but 10 is the least amount of money and 1 is the highest cost. Battery life is scored using the number of days that the battery lasts. The list of products found to be in competition with our device are listed below:

- FitBit Flex:
  - Cost: $99.95
  - Size: The bracelet relatively thick making it extremely noticeable.
  - Functionality: Tracks physical activity very well, but does not do any posture tracking.
  - Comfort: Fairly comfortable as is has a sturdy, adjustable band that will keep it from falling off or sliding around. However, he band could get caught on things and can irritate the skin when wet.
  - Ease of Use: It has a user friendly interface on the bracelet itself and has a well refined computerized interface on the user's phone. The band is also completely waterproof so the user can wear it in the shower and can track physical activity when swimming.
  - Battery Life: Up to 5 days and is rechargeable.
  - Aesthetics: Comes in different colors to make up for the fact that the product is always in plain view. Sleek and uses quality materials.
- UpRight Go:
  - Cost: $59 (plus replacement adhesive pads)
  - Size: Product is small and can be hidden under clothes if placed properly.
  - Functionality: UpRIght Go is a accurate posture measuring device… when it stays on. There are many reviews from customers who complain that the device either did not stick to their bodies or it would not stay on. Due to the nature of this device if it did not adhere to the correct location on the body it was more or less useless.
  - Comfort: Due to the method of adhesion this product must be ripped off the body at least once per day which is very uncomfortable and even painful. Additionally, it is very awkward to place this product in the optimal location on the user's back.
  - Ease of Use: This product has a very straight forward user interface that includes turning the product on, placing it in the right location, and that's pretty much it.

The app allows you to see your progress and change the device from “tracking” mode (which only tracks your posture and does not use the vibrator to notify the user when they are slouching) to “habit forming” mode (which does use the vibrator)

- Battery Life: 8 - 12 hours
- Aesthetics: The device is sleek and a nice shiny white color. However, the device would look strange if not covered using a completely closed back shirt.

- Lumo Lift:
    - Cost: $249.95
    - Functionality: This device is the most functional among the competitors. The vibration function can be turned on and off simply by double tapping the magnetic square.
    - Size: This device is separated into two parts. The sensor and the majority of the electrical components are housed in the largest piece housed beneath the clothes. This piece is quite thin and can easily be hidden with clothing. The second piece, the magnetic square, is quite small and is worn on the front of the shirt or on a bra strap. There is a downside aspect here as the product cannot be totally hidden for men and women who would prefer not to clip anything to their bra.
    - Comfort: The Lift is placed on the front of the body, so there is no awkward reaching to get just the right spot on the back like the UpRight Go. Additionally, the device connects to the body via a magnet, so there is no painful sticker removal and no need to buy replacement adhesive pads. There might be some discomfort with the device possibly bumping against the chest of the user when they move as the device is not directly connected to the body. Lumo suggest wearing tight clothes when using this device, but it is naive to think that customers will only wear tight clothes from now on to accommodate the device.
    - Ease of Use: There are more features associated with this device making the user interface more complex than the other products discussed, however; it is more convenient once the customer knows how to operate the device and the app that goes with it.
    - Battery Life: Up to 5 days and is rechargeable.

    - Aesthetics: It is evident that the aesthetics for this device were considered carefully. The magnetic square meant to be worn on the front of the customer's top comes in ten different colors and can be mixed and matched to accommodate the customer's stylistic needs. Additionally, even though the second piece of the device is housed beneath the clothes it comes in three colors and is in sleek, visually appealing metallic colors.
- Zikto Fitness and Activity Tracker:
    - Cost: $81.24
    - Functionality: The Zikto Fitness and Activity Tracker tracks balance and walking posture when the customer exercises. From customer reviews it appears this device has a lot of bugs, for example, many users frustrated because the bracelet buzzes when performing common actions (such as looking at phone).
    - Size: Rather bulky and not discreet to wear.
    - Comfort: Fairly comfortable to wear, has an adjustable strap.
    - Ease of Use: Customers indicate that there are a lot of space for improvement in this product which includes creating a more durable product (the wristband has a tendency to snap). The smartphone app also needs improvement as many customers have called it useless.
    - Battery Life: Up to three days, not rechargeable.
    - Aesthetics: The bracelet is rather ugly. It comes in two colors, but it is bulky and not visually appealing at all.
- Prana:
    - Cost: Unknown. This product is not for sale yet and their patent is currently pending.
    - Functionality: Primarily tracks breathing, but it also includes posture tracking and step tracking.
    - Size: Super small and discrete. The only piece that can be seen on the front of the clothes is the back of the clip that holds it to the waistband of the wearer's pants.
    - Comfort: This device does not touch the skin very much and it is designed to be placed on the waistband of the pants which are generally tightest piece of clothing that a person wears. However, this is not always the case as people could wear

pants that are too large. This would skew the data as this product relies on measuring breathing patterns on the lower back.

- Ease of Use: Easy set up, the customer needs only to place the sensor and communicate with it using a smartphone app.
- Battery Life: Up to 7 days with wireless charging capabilities.
- Aesthetics: The sensor has no special visually pleasing features because it is mostly hidden within the waistband of the wearer's pants.

The value analysis performed to compare our product with successful existing products is shown in Figure 9 below.

	Cost	Functionality	Size	Comfort	Ease of use	Battery Life	Aesthetics	Weighted Total
Weight	80	95	90	100	65	70	40	
Our Product	8.5	7.5	5	9	10	5	7	4022.5
FitBit Flex	7	1	6	6	9	5	7	3010
Upright Go	10	7	7	4	4	0.5	10	3190
Lumo Lift	1	10	8	7	9	5	9	3745
Zikto Fitness and Activity Tracker	8	3	6	6	5	3	5	2800
Prana	1	8	10	9	8	7	6	3890

Figure 9: Competitive Value Analysis

Our conclusions here are rather speculative; however, they were based on the requirements obtained from market research. There could be some discrepancies in grading our competitors because what we and our survey group want might not be what the market as a whole is looking for. As a result, what we perceive as a major asset or downfall might be disputed by others. On the other hand, we thoroughly researched these competitors and read both what the company says the device should be able to do, along with customer reviews detailing what the product actually does. We listened to the customer reviews intently to see what aspects about the product that people liked and disliked in order to improve our own design. Overall, the value analysis above may have a few errors, but the outcome still gives a sufficiently accurate reading on our product's potential for success.

3. Design Approach

3.1 System Architecture

In this section, we will provide an architectural description of the device. This will include block diagrams of the device as well as detailed description of each module. The technical block diagram for the device is shown in the Figure 10 below.

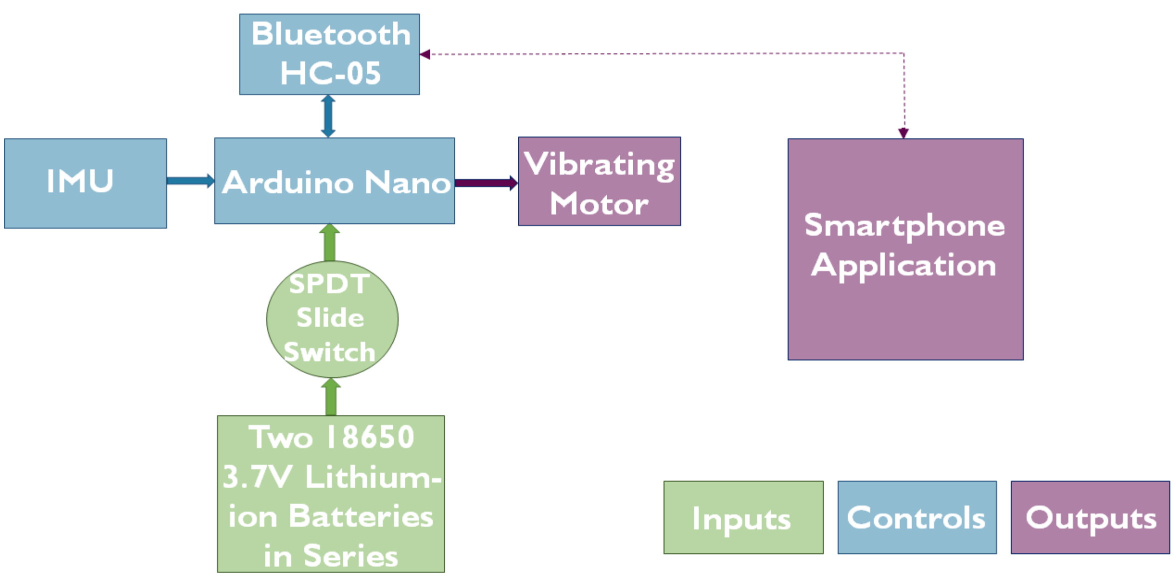


Figure 10: Technical Block Diagram

3.1.1 Power Source

The method we are using to power our device is two 3.7V Lithium-ion batteries connected in series with a battery capacity of 2600mAh. In addition, these batteries are rechargeable and will hold a charge for many days (approximately 80 hours), which will make our device more convenient and user-friendly. These batteries come in a long cylindrical shape that will be a good fit for our product. Lastly, two batteries instead of one will evenly distribute the batteries' weight and make the device more likely to stay on and be more comfortable.

3.1.2 User Input

In order to turn the device on and off there will be a SPDT slide switch connecting the batteries to the microcontroller. This will allow the user to conserve power when the device is not in use.

3.1.3 Microcontroller

The microcontroller model we chose is the Arduino Nano. Out of all the microcontrollers on the market, the Nano is the most feasible model for our application because it is small but it still contains all of the pins we need. We were originally going to purchase a microcontroller with a built-in wireless transceiver, but these boards put the Bluetooth into the peripheral. This seemed like it could cause problems, so we decided to get a separate Bluetooth module.

3.1.4 Posture and Physical Activity Sensor

In order to measure posture and physical activity we are using an IMU (Inertial Measurement Unit). The IMU measures posture by sending the value of the X, Y and Z-axis from its accelerometer to the microcontroller. Bad posture is detected by calculating the difference between the original value when calibrated and the value at latest data point. Physical Activity will be measured using linear acceleration data. Since sitting and participating in non-physical activity will not register as linear acceleration, we can determine if the user is inactive by past linear acceleration data.

3.1.5 User Notification System

In order to notify the user that they are slouching or have not moved in a while we are implementing a small vibrating motor. The motor will be located on the user's back/shoulder area and will provide a brief vibration to remind the user to sit up straight or to go for a walk. This notification method is instantaneous and does not require the user to interrupt any activities in order to interact with the device. The motor will be controlled by the Arduino Nano and will be triggered using data collected by the IMU.

3.1.6 Wireless Transceiver

In order to transmit data to our application, we will be using a Bluetooth transceiver. The transceiver will be a HC-05 Bluetooth module in our design because we would like fast transmission speeds without having high power consumption. Additionally, we decided to choose Bluetooth SPP (Serial Port Profile) over BLE, because the former is designed for continuous two-way data transfer with high application throughput (up to 2.1 Mbps). Overall, the transceiver will facilitate the communication between the app and the rest of the physical parts.

3.1.7 Smartphone Application

The smartphone application makes up the software side of our project. It stores history of the user's posture and physical activity data and displays them in a user-friendly graph. The application was programmed using the "App Inventor" for Android, which is an open-source web application originally provided by Google and now maintained by MIT, and communicates with the physical device via Bluetooth.

3.2 Module Definitions

The full hardware schematic of the device is shown in Figure 11 below.

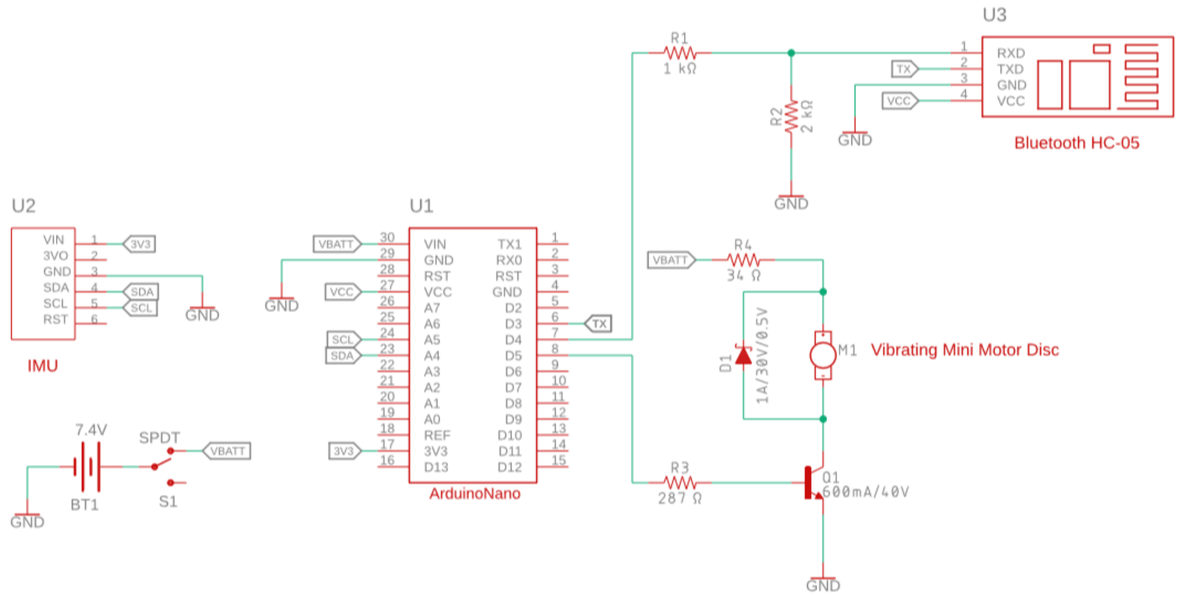


Figure 11: Schematic Diagram of the Hardware Device

In this section, each of the modules of the device will be discussed individually. For each module, we will describe the module functionality, the specifications and calculations used to reach the current design and ensure functionality, the inputs and outputs of the module, and the plan for testing the module individually (before integration).

3.2.1 Arduino Nano

The Arduino Nano is our central microcontroller which includes the central processing unit that will be used for our system. More analytically, the microcontroller's functionality is to instruct the peripherals, like the IMU, the Bluetooth HC-05 and the vibrating motor to execute an individual set of commands. These commands could be to either request for sensor data gathered, to instruct transmission of data, or to enable a peripheral output. Furthermore, it is the microcontroller's responsibility to effectively process the data received from its peripherals using its own resources. For our device, we will program the microcontroller to initialize each of the other modules, along with variables and data structures that will be used, to check periodically if the user is slouching or if they are physically inactive, and to process this information. Moreover, whenever the Bluetooth HC-05 module receives a valid request through our mobile application, the microcontroller will attempt to satisfy this request by instructing the HC-05 to transmit data coming from the program's data structures. Lastly, the microcontroller will command the Bluetooth module to automatically transmit the daily posture and physical activity data to the user's phone app at 8 pm each day. The device's functionality, which will be executed through the Arduino Nano microcontroller, is also explained in the flowchart in Section 2.2 (Figure 7).

The following specifications and calculations were used to ensure (to a reasonable degree of certainty) that the Arduino Nano module would function as planned. Many of these specifications were also used to make key design decisions for the module, particularly with regard to the inputs and outputs.

To verify that the Nano could be powered properly as well as supply power to its peripherals during implementation and testing, the following specifications were found in the data sheets. First, the recommended input voltage limit of the Nano is 7–12V. This means that the Nano can be successfully powered by the battery module we intend to use. Another power option which will be used during the programming and testing phase, is USB power via a computer. According to the data sheet, the total maximum current draw from the Arduino when powered from a USB port or power supply is 500mA. This means that even when powering the Nano by USB, it can supply more than enough current for all of the connected modules.

Next, the input/output (I/O) pins were examined to ensure that they will be capable of the functionality we need. We found the following specs.

- The voltage limits on I/O pins are -0.5 to +5.5V max
- The total max current per I/O pin is 40mA
- The sum of output currents of all I/O pins combined must not exceed 200mA

Since none of the peripherals connected to the Nano module will exceed these specs, the module should work as expected. The calculations used to confirm this are explained in the paragraphs below.

The Arduino Nano voltage values on I/O pins can span between -0.5V to 5.5V. Consequently, there is the possibility of pairing the Bluetooth module with the microcontroller and controlling the vibrating mini motor disc if fitting configurations are applied. More analysis on these topics will be provided in the sections below. For the case of the motor, it should be noted that one of the limitations of the Arduino Nano is that the maximum current that an I/O pin can provide is

40mA. Unfortunately, our vibrating motor draws from 40mA to 100mA when working from 2V up to 5V. This means that the startup current will be larger than the motor's rated current, which could severely damage if not permanently destroy the pin. To avoid this issue, we included a transistor to limit the current coming out of the Nano I/O pin. We also added a few other components to complement this change (these components are shown in the schematic in figure 11 of this report).

Another important consideration regarding the Arduino Nano is that the sum of all currents out of all the I/O pins combined cannot exceed 200mA. Despite the fact that the Nano voltage regulator may allow up to 500mA current draw across the "5V" and the "GND" pins, all analog and digital I/O pins must not amount to more than that limit. Therefore, we had to keep this specification in mind in order to not damage our microcontroller. The maximum current draw that is reached when adding up all of the current values in our design is 67.3mA. This calculation is shown in equation (1) below.

$$\text{IMU Max} + \text{Bluetooth Max} + \text{Motor Max} = 12.3\text{mA} + 40\text{mA} + 15\text{mA} = 67.3\text{ mA} \quad (1)$$

Please note that the equation above is taking into account the absolute maximum current draw possible at any given time. The actual draw will be a much lower value (for example, the average draw for the bluetooth module is only 8mA).

In conclusion, the Nano's limitation with respect to the current sum of its I/O pins being below the 200mA limit is fulfilled in our circuit design. The recommended input voltage for the Arduino Nano is 7-12VDC. We will supply the microcontroller with 7.4VDC through its "Vin" pin using the battery arrangement explained in section 3.2. However, the operating voltage of this microcontroller is 5V, meaning that it has a built-in voltage regulator. Furthermore, since the Nano is powered, it must also be grounded, which is done through the "GND" pin. The microcontroller is connected to the IMU through I2C. More specifically, the SCL (Serial Clock) and SDA (Serial Data) busses are connected to two of its analog pins that were designed for the I2C communication protocol, namely "A5" for SCL and "A4" for SDA. The Nano also powers the IMU with 3.3VDC through its "3V3" output pin (detailed characteristics of IMU are discussed in section 3.4). It is also connected to the Bluetooth HC-05 module through UART in order to transmit and receive serial data. The microcontroller powers the HC-05 using its "5V" output pin and uses its "Tx1" pin to receive data from the HC-05. The transmit and receive logic signal of the HC-05 uses 3.3V. Therefore, receiving signals from the HC-05 to the Arduino Nano microcontroller is safe, since the I/O pins can receive signals from -0.5V to +5.5V (in actuality, it is "Vcc+0.5V", which for a 5V Arduino is +5.5V). However, the Arduino will transmit signals to the receiving pin of the HC-05 which has 3.3V logic, using 5V logic through the "Rx0" pin. To combat this discrepancy, which could potentially destroy our Bluetooth module, we created a voltage divider which will be described in section 3.3 along with other details concerning the Bluetooth HC-05 module. Finally, the Arduino Nano controls a vibrating motor through the "D3" digital I/O pin which also has PWM functionality. The configuration surrounding the vibrating mini motor disc is explained in section 3.2.5 of this report.

3.2.2 Battery and Switch

After implementing a Value Analysis based on the estimation data, we decided to pick two 3.7V, 2600mAh Lithium-ion batteries (rechargeable) in series to power our system. We calculated this using equation 2 shown below. The reason behind multiplying the division of battery capacity with current draw with 0.80 is so we can receive a more realistic number of the battery life provided to our project and not base our solution off the ideal battery life.

$$\frac{Battery\ Capacity\ (Amphours)}{Current\ Draw\ (Amps)} \cdot 0.8 = Battery\ Life\ (Hours) \quad (2)$$

On average, the current draw of our system is estimated to actually be around 25mA (approximately 15mA for the Arduino Nano while running, around 2mA for the IMU and about 8mA for our Bluetooth transceiver). Our microcontroller (Arduino Nano) requires an input voltage of 7V to 12V, but its built-in regulator lowers the operating voltage to 5V. The maximum total current we can draw from an Arduino Nano when it is powered via an external power supply is 500mA. The IMU will be powered through the Nano's 3.3V pin and will require a maximum of 12.3mA. We plugged in a value close to that in our initial estimation (10mA). However, we saw that the IMU will draw about 2mA on average, and thereby we replaced the first value (12.3mA) with 2mA in our calculations for our revised design. The Bluetooth transceiver (HC-05) will be powered through the 5V pin and we said that it will require about 30mA, but after looking more closely into its datasheet, we realized that it will consume a maximum of 30-40 mA only while the module is attempting to pair with another device. Under normal circumstances (i.e. when Bluetooth module is connected to the user's phone), the Bluetooth module's current consumption falls to 8mA. Furthermore, the Arduino Nano would consume about 15mA. Finally, the current draw from our vibrating mini motor disc is not included in the circuit's average current consumption, since it is enabled for a relatively short period of time whenever it is required. The current draw of our system is estimated to be around 60mA (approximately 20mA for the Arduino Nano while running, around 10mA for the IMU and about 30mA for our Bluetooth transceiver).

With 80 hours of battery life as a requirement (5 days at 16 hours per day), equation (2) above gives us a battery capacity of 2500mAh. Therefore, we derived that the batteries that were the most applicable to our product are Lithium-Ion batteries with specifications of 3.7V, 2600mAh (rechargeable). We will be using two batteries in series, which will increase the voltage output to 7.4V in order to satisfy the recommended input voltage of the Nano (7-12V). These batteries will power each of the physical modules. With the exception of the vibrating motor module, current will be sourced using the microcontroller and will be distributed with a series of resistors, transistors, and diodes at each module.

As mentioned earlier, our design uses two batteries that are connected in series. The power source is controlled by a SPDT slide switch, which is connected to the vibrating motor, the microcontroller and all other modules by extension. The switch has a common terminal that is connected to the positive terminal of the batter. With respect to the other two terminals of our switch, one is connected to the the rest of our circuit through the "Vin" pin of the Arduino Nano and the positive terminal of the vibrating motor, whereas the other is not connected to anything. The side whose pin is connected the microcontroller will be the 'on' position and the side whose pin is not connected to anything will be the 'off' position.

3.2.3 Bluetooth HC-05 Module

The bluetooth HC-05 module of our device will provide the means for connection to the user's smartphone. It will send data from the Arduino Nano microcontroller to the phone application module and vice versa. The data being sent from the HC-05 to the phone will include information on the user's posture, physical activity level, and the time at which these measurements were taken. When the phone transmits to the HC-05, it will be a request for the aforementioned data to be sent.

The following specifications and calculations were used to ensure that the Bluetooth module would function as planned. Many of these specifications were also used to make key design decisions for the module, particularly with regard to the resistors connected to the HC-05. The main concern with this module was making sure the correct logic voltage level was being sent to the HC-05 from the Arduino Nano and vice versa. The logic level of the Nano is approximately 5V and the logic level of the HC-05 is approximately 3.3V. To accommodate this difference, we will implement a simple voltage divider between the RX pin of the HC-05 and the RX0 pin of the Nano. This voltage divider needs to deliver only 3.3V to the HC-05. We found that we will need proportional resistor values of R1 = ½*R2 using equation (3).

$$3.3V = 5V*R2/(R1 + R2) \rightarrow R1 = R2*(5V/3.3V - 1) \simeq ½*R2 \quad (3)$$

Based on this, we chose R1 to be 10kΩ and R2 to be 20kΩ (Both 1% tolerance). No additional components are needed on the TX pin of the receiver because the Nano will accept any voltage between -0.5V and 5.5V at the TX1 pin.

Another concern was the current draw of the HC-05 on the Nano. The HC-05 will be connected to the 5V output pin (VCC) of the Nano. We had to be sure that the current draw would not exceed the limit for the connected Nano pin. According to the HC-05 specs, the current fluctuates in the range of 30-40mA during pairing. After pairing, regardless of whether or not the HC-05 is processing data, the current is 8mA. Based on this, the maximum amount of current draw on the VCC pin in 40mA. This will not cause any issues because the VCC pin on the Nano can handle up to 500mA.

We decided to choose Bluetooth SPP (Serial Port Profile) because it is designed for continuous two-way data transfer with high application throughput (up to 2.1 Mbps). Consequently, it is ideal for sending bursts of data between two devices. During prototyping, we would like to make sure that performance is guaranteed for short distances by having a continuous broadband link (multiple data will be received/transmitted). Figure 12 below shows some of the basic characteristics of the Bluetooth profile being used in our device.

Name	Bluetooth Classic
IEEE Standard	802.15.1
Frequency (GHz)	2.4
Maximum raw bit rate (Mbps)	1-3
Typical data throughput (Mbps)	0.7-2.1
Maximum (Outdoor) Range (Meters)	10 (class 2), 100 (class 1)
Relative Power Consumption	Medium
Example Battery Life	Days
Network Size	7

Figure 12: Bluetooth Classic Basics

Because of these characteristics and the standardized nature of Bluetooth SPP, there should be very few issues related to the transceiving of data between the HC-05 and the user's smartphone. This section will cover each of the inputs and outputs for the Bluetooth module (as shown in the schematic in Figure 11). The inputs for the HC-05 module are the 5V input, which powers the HC-05 itself, and the 5V logic signal from pin RX0 of the Nano. This logic level is reduced to 3.3V at the RX pin of the HC-05. The HC-05 module has only one output, a 3.3V logic signal at the TX pin. In addition to these hardware inputs and outputs, the HC-05 also has the input and output associated with receiving and transmitting wireless signals, but these will not be considered because they do not affect the other modules.

3.2.4 BNO055 Absolute Orientation Sensor (IMU)

We will be utilizing absolute orientation to measure posture because it can detect subtle movements like slouching. Additionally, we will be using the linear acceleration vector to measure physical activity. The linear acceleration vector will work well for this application because if the value increases we will know the user is moving around, but if it has been zero or close to zero for a while we will tell the user that they should go for a walk. The IMU will be controlled using the Arduino Nano and the information it gathers for both absolute orientation and the linear accelerator vector will be transmitted to an application via the Bluetooth module.

According to the specifications, the IMU can be powered between 3.3V to 5V. We chose to power it with 3.3V in order to reduce power consumption. Another important aspect of the IMU which had to be calculated was the duty cycle—in other words, how much of the time it would actually be running. Taking this into account drastically reduced our required battery size, because the IMU only needs to take measurements periodically, not constantly. The BNO055 Absolute Orientation Sensor has multiple power modes. We are hoping to use both normal mode and suspend mode. In normal mode, the maximum current draw is 12.3mA. In suspend mode, the maximum current draw is 0.04mA. If the device takes a measurement for 10 seconds per minute, it will only be in normal mode 16.67% of the time. This means that the approximate average current draw is 2.08mA. Based on these calculations, the battery size needed for the device is significantly smaller than if the IMU was in normal mode all the time.

The IMU is connected to the microcontroller by three pins: VIN, SDA, SCL. The IMU is powered by its VIN pin connected to the 3.3V pin on the Nano. The data collected by the IMU is transmitted through the SDA and SCL pins that connect to the A4 and A5 pins on the microcontroller, respectively. The SDA pin is the primary data pin that transmits important information to the data line on the Nano (pin A4). The SCL pin is the IMU’s clock that transmits when the data was collected in milliseconds to the microcontroller clock line (A5). Lastly, the GND pin on the IMU is connected to ground.

3.2.5 Vibrating Mini Motor Disc

The vibrating mini motor disc is used as a direct user notification system for our application. More precisely, the vibrating motor is enabled through our microcontroller for a couple of seconds whenever our device senses that the user is slouching. If the user keeps slouching even after they are notified from the vibrating motor once, the motor will be activated again after a longer period of time as long as bad posture position is being sustained throughout that time. The amount of time until the vibrating motor is able to vibrate again will be a multiple of the time between two checks of the user’s posture. We decided to apply this implementation because we want to make sure we directly remind users on their posture position, without overly disturbing them.

The following specifications and calculations were used to ensure that the vibrating motor module would function as planned. Many of these specifications were also used to make key design decisions for the module, particularly with regard to the circuit elements connected to the motor itself.

In our application, we will control the vibrating motor through a PWM pin of the Arduino Nano. However, for this to happen without burning the PWM pin, we have to add some extra components to our configuration since the motor itself draws 100mA at 5V (minimum current draw is 40mA at 2V) while the maximum current an Arduino Nano digital I/O pin can source, is 40mA. To remedy this, we used an NPN transistor and voltage divider circuit shown in the schematic (low-side switch circuit). More specifically, we selected the P2N2222A amplifier transistor with a max reverse DC voltage of 30V, 1A of average current rectified and a forward voltage of 500mV, which fits the standards of our application based on our calculations below. The transistor will function as an electrical switch which will be controlled through the microcontroller PWM pin. A resistor must be added between the two to enforce current limitation. Furthermore, a flyback diode was placed across the motor to provide a safe path for the inductive kickback of the motor. Instead of powering the motor through the Arduino Nano, this circuit powers it directly from the battery, with the Nano turning it only on and off. To find the component values for this circuit, we calculated them using basic circuit analysis, then tested them in Multisim.

First, we calculated the value of the resistor connected between the Arduino Nano and the transistor. This resistor is used to limit the current the digital output must source and the transistor base must handle. We wanted a resistor that would draw a current of approximately 15mA from the I/O pin of the Nano with a 5V digital output. We found a value of 287Ω using equation (4) below.

$$R = (V_{in} - V_{be})/I = (5V - 0.7V)/15mA \simeq 287\Omega \qquad (4)$$

Next, the resistor between the battery and motor was calculated. The purpose of this resistor is to adjust the current draw of the motor on the batteries. It is also necessary because the motor is meant to function between 2-5V and the batteries supply 7.4V. It acts as a voltage divider with the resistive load of the motor. Since the motor has a linear V-I characteristic (5V current draw: 100mA, 4V current draw: 80mA, 3V current draw: 60mA, 2V current draw:

40mA), it can be easily modeled as a resistive load. By Ohm's law, the motor can be modeled as a 50Ω resistor, as shown in equation (5).

$$R = V/I = 5V/100mA = 50\Omega \quad (5)$$

Once the motor resistance was known, the voltage divider resistor value was chosen. In order to stay safely within the 2-5V range across the motor, we chose a resistor value (resistor with 1% tolerance) that would leave just below 4.5V across the motor. For simplicity, we used 4.4V. The calculation is shown in equation (6) below.

$$4.4V = 7.4V*50\Omega /(50\Omega +R) \rightarrow R = 7.4V*50\Omega/4.4V - 50\Omega \simeq 34\Omega \quad (6)$$

Finally, we had to integrate a flyback diode in our design. The reason for adding this component to our circuit, is because the motor is partially an inductor. Consequently, without a diode across the motor, if the transistor shuts off quickly, then the voltage across the motor will get larger to ensure the continuity of current flow. This will probably result in destroying the transistor. However, if a flyback diode is used when the transistor shuts off rapidly, the current that is still needed to flow through the motor (inductor) for some time will flow through the diode instead. For our application, we decided to use a Schottky diode, namely the 1N5818. We decided to use this type of diode because, when the motor is controlled through a PWM pin, we need a diode with fast reverse recovery at this low voltage. Schottky diodes have fast twitching speeds, meaning that they are great with higher frequencies since they still succeed in rectification (i.e. preventing reverse flow of current). Therefore a Schottky diode suits our application better than general purpose diodes, because we will be turning the motor on and off rapidly, as implied by "PWM", whereas general purpose diodes are power rectifiers intended for normal power line frequencies (e.g. 50-60 Hz). The lower forward voltage of the 1N5818 will also cause less backwards EMF on the inductance during off-time and thereby, will make the system more efficient. To conclude, the 1N5818 fulfills all of our requirements regarding maximum reverse DC voltage, average current rectified and maximum forward voltage of our application, based on our calculations and the results from the Multisim simulation.
The schematic demonstrating the calculations above is shown in Figure 13.

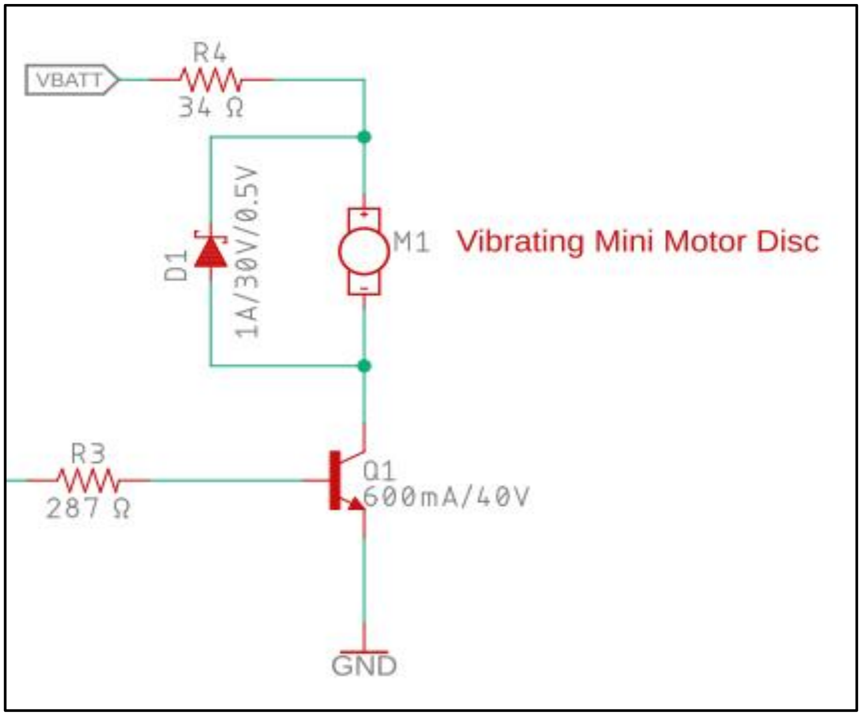


Figure 13: Vibrating Motor Module Schematic

The vibrating motor will be controlled through the “D3” PWM pin of the Arduino Nano microcontroller. Therefore, based on our calculations and the specs of the modules used, we ended up with the following configuration. The “D3” pin will be connected to the base side of the P2N2222A (NPN) transistor with a 287Ω resistor in between the two, to limit the current sourced by the pin. The negative terminal of the vibrating motor will be connected to the collector side of the transistor, while its positive terminal will be connected to the positive leads of the battery with a 34Ω resistor in between to decrease the voltage of the battery to an acceptable value with respect to the vibrating motor. Moreover, the 1N5818 Schottky diode is connected across the vibrating motors as a flyback diode to eliminate the voltage spike created when the field in the motor collapses and thereby, avoid burning the Nano or the transistor. Lastly the emitter voltage of the P2N2222A transistor is connected to ground.

3.2.6 Software Application

The smartphone application provided along with our device will assist the user in receiving useful feedback with respect to their posture and physical activity on a daily manner. The phone app will communicate with our device through Bluetooth. The data that will be transmitted from the app are through the buttons it includes while data that will be received from the app will be displayed on the app's screen in different formats. The user will be able to "ask" the device on how much time they have been slouching and/or have been physically inactive on that day till the time they requested for that information. Any data received from our device will be processed, resulting to the display of minutes and graphs regarding posture and physical activity. More precisely, the graphs will show the amount of time the user has had a good vs a bad posture and for how much time a user has been physically active vs inactive during the day. Furthermore, the application will be intuitive to navigate and the data will be presented in a way that they will be able to give a clear message based on the sensors' readings. The phone app is being developed in "App Inventor" for Android devices. The app's functionality is also displayed in the flowchart in Section 2.2 (Figure 8).

The data collected by the IMU will be transferred and processed by the microcontroller. Then, the Bluetooth module will transmit the processed data to the application upon the user's request (output from app to device). These input data will not be shown in the phone application and will also be parsed in order to output user friendly graphs that will depict the user's data for the day. Additionally, the app will have other navigation and user interface graphics that will allow the customer to choose the information they are viewing and allow them to request more up to date data from the device.

3.2.7 Physical Case

The case of the device was 3D printed from black ABS plastic. We used the Solidworks model shown in section 2.1 of this report. We then hardwired/soldered each our the components of the final design into the case. The finished device is shown in figures 14-17 below.

Figure 14: Inside of the Device

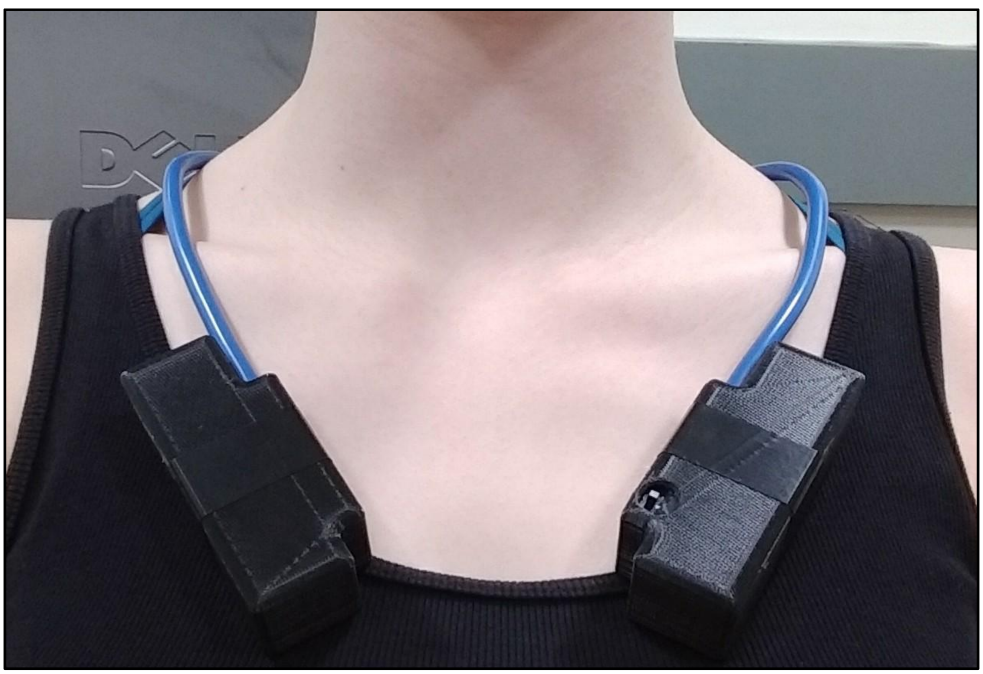


Figure 15: Front View

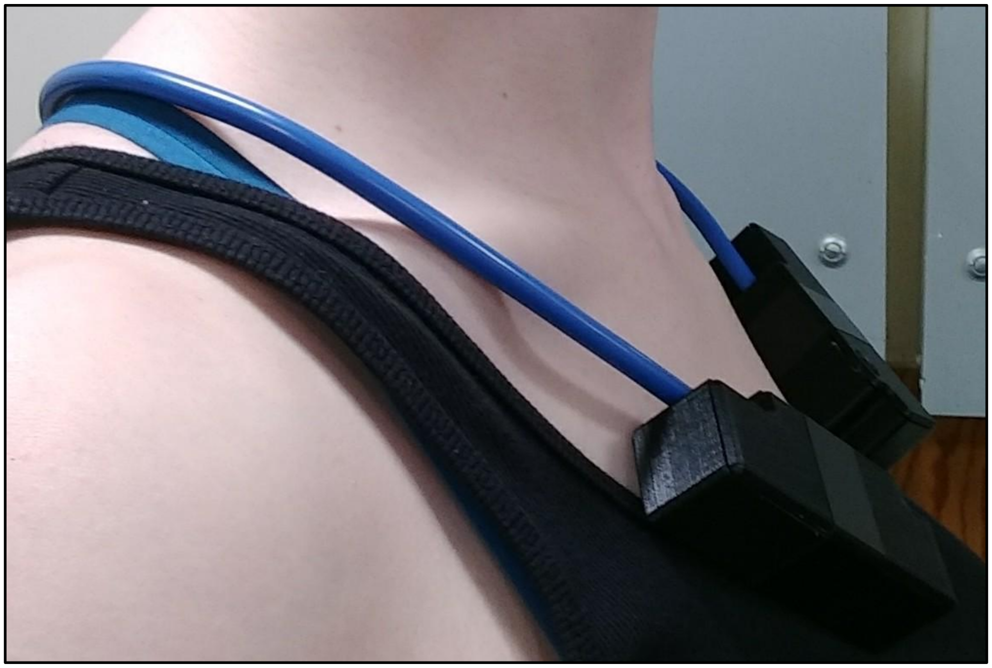


Figure 16: Side View

Figure 17: Back View

4. Product Results

The finished product is working very well and includes most of the features we wanted it to have when we originally set out to do this project. The functionality of the device, namely how well it tracks bad posture and low physical activity, is very accurate and does not need calibration. We originally thought that we might put in a feature that would allow the user to calibrate the device to their own unique type of movement, but it appears that this is not necessary after all. The IMU and our code have worked together in order to cause the device to start anywhere in space and still be able to capture the subtle movements that indicate the bad habits we are trying to help.

We were having trouble getting the batteries to make a strong enough connection to power the device. This problem was solved when we replaced a wire in one of our battery holders that was broken. The product is now completely wireless and can be used anywhere. The vibrating mini motor disc is currently working to the specifications we outlined earlier. We were worried that if the IMU was inaccurate or needed calibration often then the vibrator would turn on when there was no need for it. This was a big concern for us because if the motor was activated too much it would disturb and potentially annoy the user enough so that they would not want to wear the product. On the other hand, we were worried that our calculations for the motor's current draw would be off and the vibration would be so small that the user couldn't feel it or so strong that it would be uncomfortable and noisy. By testing the motor at different voltage outputs it became clear that our original calculations were correct. Lastly, the motor stayed on only when it was needed and from what we could measure it did not draw unnecessary power from the batteries or stress the system when it was running meaning that the other circuit elements surround the motor were doing their jobs correctly.

We 3D printed a case that is shown as a model in Section 2.2 (Figure 6) and printed in Section 3.2.7 (Figures 14-17). The rear case was meant to house all circuit elements and the two front cases housed the batteries. All of the elements fit well inside their respective areas and the case itself allows the device to be wireless and mobile. All three cases are connected by two straps made out of plastic tubing with copper wire inside them. The wire no only acts as a connection between the main circuit and power source, it also causes the tubing to hold its shape yet be flexible enough for the user to shape it to their bodies. We would have liked the case to be more discreet, but for an initial prototype we are very happy with what this case was able to accomplish.

The bluetooth link has been reliably set up between the device and a smartphone and the pairing process is smooth. The product shows up every time on a phone's bluetooth search screen and is able to sync data at the press of a button.

The application includes all of the information that we were hoping to implement including minutes slouching and inactive, graphs of this data, and the ability to sync data whenever the user requests it. Additionally, five different buttons were implemented so the user can choose the type of data they want to see. The information the user can choose to receive is their posture or inactivity time in minutes (top two buttons) or a graph of their posture or inactivity as it was measured throughout the day (Figures 18 and 19 below).

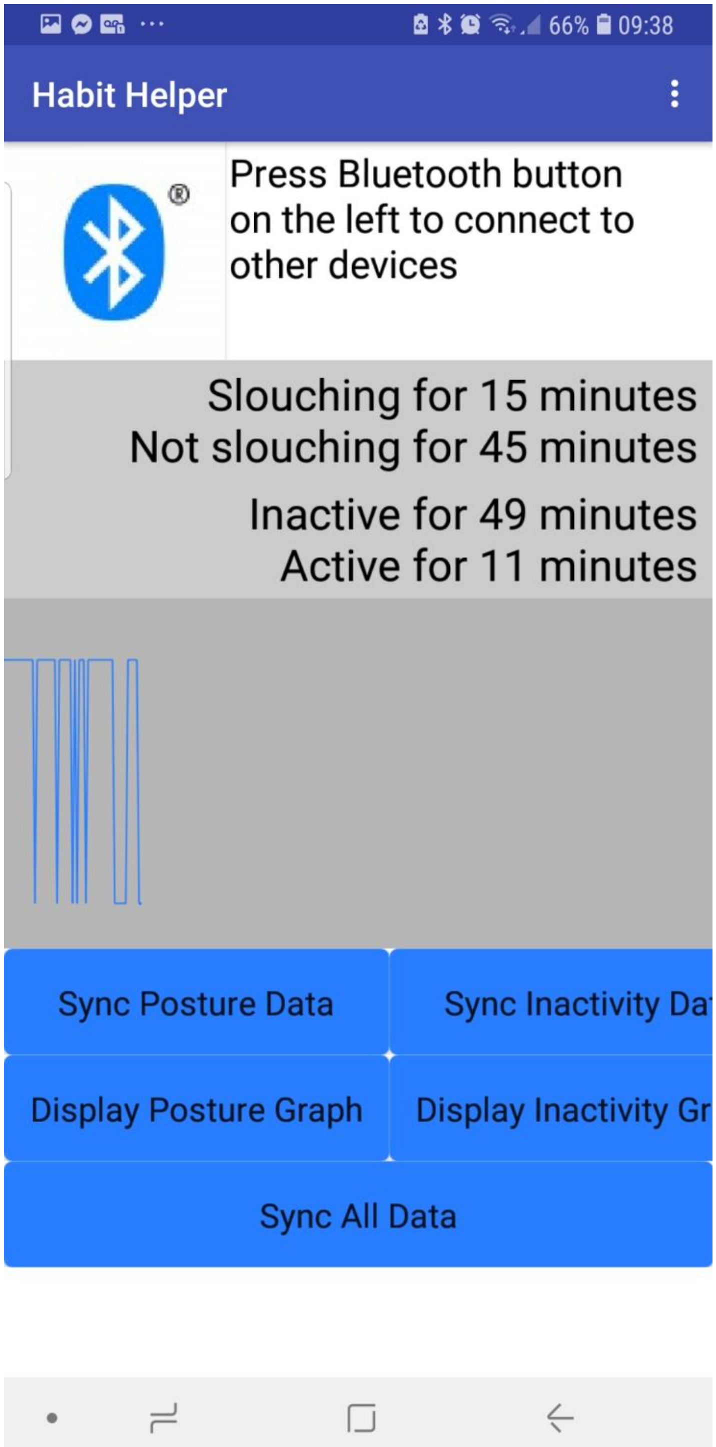


Figure 18: Application Displaying Posture Graph

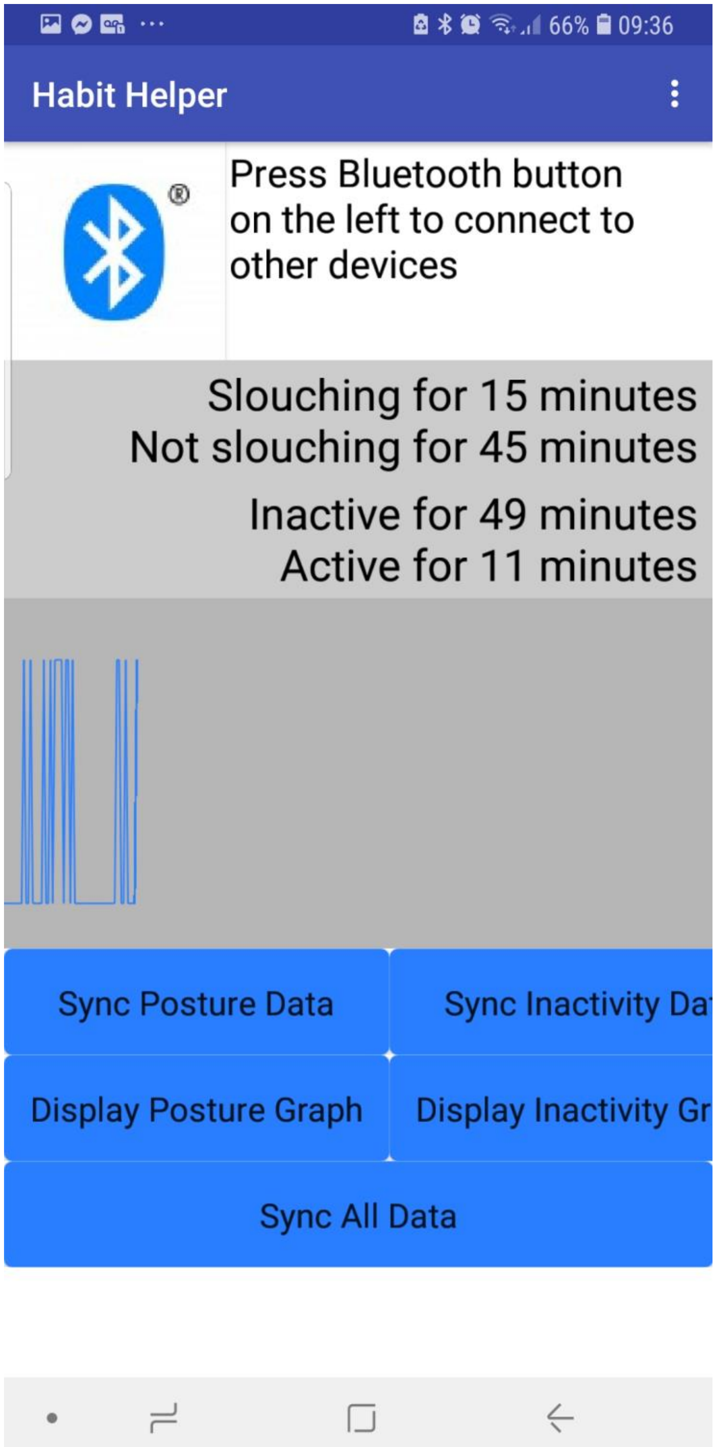


Figure 19: Application Displaying Inactivity Graph

5. Cost Analysis

5.1 Initial Investment

This section is very vague. Little to no explanation about how the numbers were derived.

Bill of Materials

Arduino Nano: $23.50

IMU: $34.95

Vibrating Mini Motor Disc: $1.95

Bluetooth Module: $8.49

Batteries(2): $11.90

Switch: $0.61

P2N2222A Transistor: $0.47

Resistors: $0.02

Schottky Diode: $0.05

Case Materials: $3.00

Total: $84.94

Use formal language for reports.

While the total amount of money we spent on the prototype was $84.94, that is not what the final product will cost per unit. First, we will be buying everything in bulk which will take a lot of money off that price. Additionally, since we are only using the IMU module to measure acceleration and linear acceleration (two of its eight functions) we will find cheaper modules that perform these same things. The reason we did not get these things in the first place was just to save money because we already had an IMU in our possession. Additionally, we will be making our own PCBs that will contain only the essential aspects contained in the IMU (or equivalent sensors), the Arduino Nano, and the Bluetooth module. This will no only save us money, but will make the device itself smaller and more aesthetically pleasing. Lastly, by the time this product is ready to be sold we will have implemented some kind of battery saving mode that we did not have time to implement in this seven week project. This means we will not have to get such powerful batteries, thus making the device cheaper (and smaller still). Overall, we estimate that the device will cost around $25 after all is said and done. With the case and the packaging that will be needed for shipping the total cost to make one unit will be around $30. If we sell each unit for $90 we will be making twice what the product costs in profit and three times the product's cost in revenue.

5.2 Return on Investment

Better You Inc. Plans to Produce the Habit Helper. From the market research outlined in Section 1.2 it is clear that our product will be most profitable marketed to people who sit for many hours per day. This market includes office workers, students, and employers who want to promote company health. In the United States alone, 80% of citizens have careers that require light to no physical activity on a daily basis. As a result, our potential customer base consists of 261.76 million people. This number is very broad, but it is a good starting point for discovering how much potential our product has.

5.2.1 Estimates

Fixed Costs (FC):

- Development: $100,000
- Production: $10,000
- Advertisement: $100,000

How did you arrive to these numbers?

FC = $210,000

Unit Costs (UC):

- Cost of One Unit: $25
- Packaging: $5

UC = $30

Number of Products Produced (N): 100

Total Production Costs (TC): $213,000

Total Revenue(TR): $9,000

Selling Price: $90 per unit

Using the number of potential customers stated above, if only 1% decided to purchase our product we would make over $200 million in revenue. See Figure 20 below for a comprehensive view of future costs and revenue.

5.2.2 Definitions

Break Even: TC = TR

Profit/Loss = 9,000 - 213,000 = -204,000

Return on Investment = (9,000 - 213,000)/213,000 = -0.96

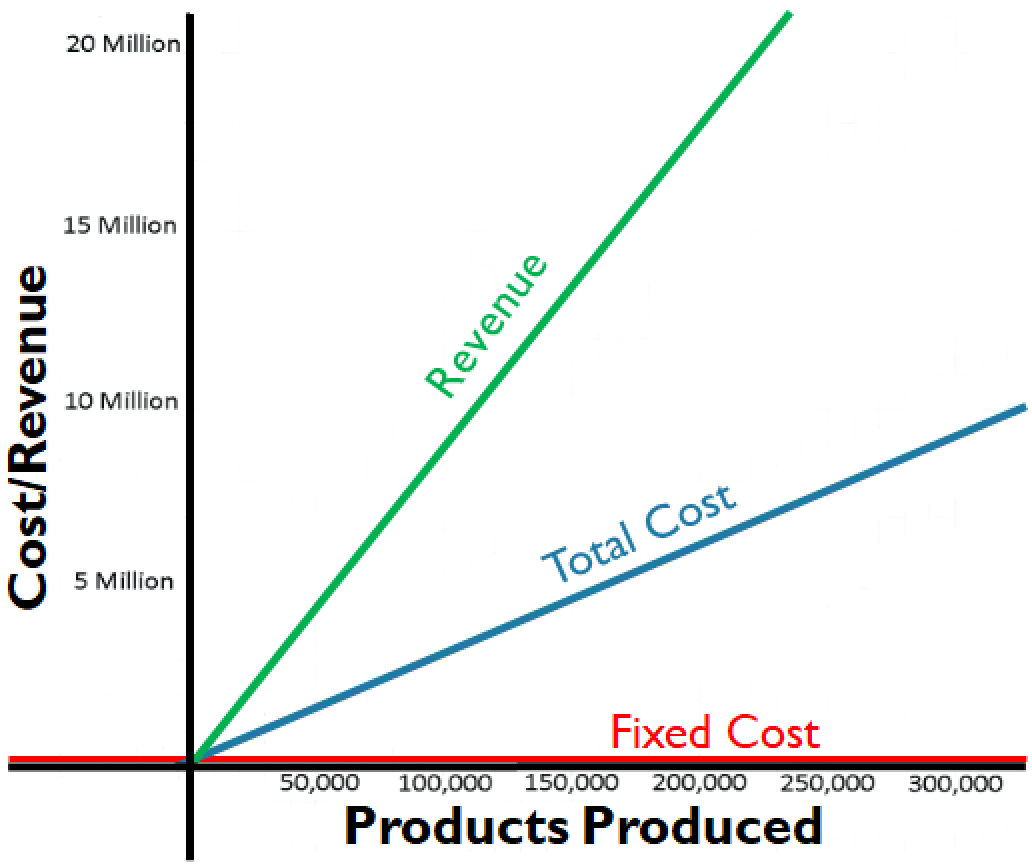


Figure 20: Return on Investment Chart

6. Plans for The Future

The biggest aspect of our design we would like to change is the size. Ideally, we would like the device to be completely hidden under clothing, but with the boards and batteries we are currently using that is not feasible. As mentioned in the previous section, there are many options that we could use in order to make the device smaller. If we were to continue this project, creating our own PCB that only contained the components we need would be a main priority. Additionally, of all the components we would like the batteries to be smaller. In order to achieve this we would implement and power saving mode that would require a smaller battery for the same battery life requirements.

While we have an application that does all of the things we originally wanted to implement, there are so many more features it could have. First, the graphs could be shown in a less square fashion to give the user more information about the severity of their slouching or the intensity of their exercise. Additionally, aesthetics have not been a priority for this project because we needed to focus on functionality first. As a result, making the application prettier and giving it more intuitive controls would make customers much more satisfied with it. Lastly, the application could have more interactive features like goal setting, competitions, and collaborations with other wearable technologies like FitBit.

7. Conclusions

Overall, this project was very successful. While there is so much more we could do with this product, we were able to establish a working bad posture and low physical activity detector in only seven weeks. There is so much more potential in this product and the project could be continued far beyond this class. In the scope of what we completed within the allotted time, we were still able to complete everything we wanted to implement by the end of the term resulting in a finished prototype that could be used for its intended purpose by a customer.

Researching and creating all of the information in this report has been a very important learning experience for us. Going through the entire process of researching, designing, and building a product has given us a very clear insight into what a career in electrical and computer engineering will be like. It also provided invaluable experience in applying the knowledge we learned from our past ECE classes that will help us get employment in the future. Overall, this was a very beneficial experience that we will take with us to our future classes and careers.

Bibliography

B. Recny, “Bluetooth UART - SPP vs. BLE implementations,” Rigado, Feb-2019. [Accessed: 26-Mar-2019]

BOSCH, “BNO055 Intelligent 9-axis absolute orientation sensor,” BNO055 datasheet.

C..BasuMallick, “Why Wearable Technologies Will Drive Workplace Wellness in 2019,” HR Technologist, 18-Jan-2019. [Online]. [Accessed: 27-Mar-2019].

D. Kliszowski, “Bluetooth Classic vs. Bluetooth Low Energy on Android - Hints & Implementation Steps,” Droids or Roids , 14-Aug-2018. [Accessed: 26-Mar-2019].

D. Schatsky and N. Kumar, “The digitally enhanced workforce,” Deloitte Insights, 25-Jul-2018. [Online]. [Accessed: 23-Apr-2019].

G. Staples, “Arduino Power, Current, and Voltage Limitations,” Electric RC Aircraft Guy, LLC. [Online]. [Accessed: 5-Apr-2019].

Sparkfun, “CR18650 2600mAH datasheet” CR18650 datasheet [18-Jul-2012].

ITead Studio, “HC-05 -Bluetooth to Serial Port Module,” HC-05 datasheet.

J. Blom, “Bluetooth Basics,” SparkFun. [Online]. [Accessed: 26-Mar-2019].

J. Dee, “How to Power a Project,” SparkFun. [Online]. [Accessed: 26-Mar-2019].

L. Ada, “All About Batteries,” Adafruit. [Online]. [Accessed: 26-Mar-2019].

L. Ada, “Vibrating Mini Motor Disc,” Adafruit. [Online]. [Accessed: 27-Mar-2019].

Microchip, “megaAVR datasheet,” ATmega48A/PA/88A/PA/168A/PA/328/P datasheet.

M. Maslakovic, “Posture corrector: Wearables that help you with your posture,” Gadgets & Wearables, 15-Oct-2018. [Online]. [Accessed: 25-Mar-2019].

Mouser, “Low drop power Schottky rectifier ,” 1N5818 datasheet.

N/A, “NPN Transistor,” Electronics Tutorials. [Online]. [Accessed: 6-Apr-2019].

N/A, “SPST, SPDT, and DPDT Switches Demystified,” Music from Outer Space. [Online]. [Accessed: 6-Apr-2019].

N/A, “Toggle Slide Switch Breadboard & Perfboard Compatible: SPDT,” Tinkersphere. [Online]. [Accessed: 28-Mar-2019].

Nate, “Battery Technologies,” SparkFun. [Online]. [Accessed: 26-Mar-2019].

O. Lathrop, “Purpose of the diode and capacitor in this motor circuit,” Electrical Engineering Stack Exchange. [Online]. [Accessed: 7-April-2019].

ON Semiconductor, “Amplifier Transistors NPN Silicon,” P2N2222A datasheet.

“Physical inactivity U.S. adults by state 2018 | Statistic,” Statista. [Online]. [Accessed: 27-Mar-2019].

“The Wearable Life 2.0,” pwc. 2016. [Online]. [Accessed: 23-Apr-2019].

“Willingness to wear health tracking technology U.S. 2018 | Statistic,” Statista. [Accessed: 27-Mar-2019].

Appendix

Arduino Nano Code:

```
#include <EEPROM.h>
#include <SoftwareSerial.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BNO055.h>
#include <utility/imumaths.h>

#define ONE_SECOND (1000)

SoftwareSerial BTSerial(3, 4);
char BluetoothData; // the data given from Computer

//Global Variables used in ISR
volatile byte cnt = 0; //counts through minute
volatile byte prevcnt = 0;

volatile byte yess = 0; //number of minutes spent in a row slouching
volatile byte nos = 0; //number of minutes spent in a row not slouching
volatile byte yesa = 0; //number of minutes spent in a row inactive
volatile byte noa = 0; //number of minutes spent in a row not inactive
volatile byte prevs = 0;//previously slouching
volatile byte preva = 0;//previously inactive

volatile byte scnt = 0; //slouch counter within minute
volatile byte acnt = 0; //Inactivity counter within minute

volatile byte iss = 0; //is slouching. 1 if slouching, 0 if not.
```

```
volatile byte isa = 0; //is inactive. 1 if inactive, 0 if not.

volatile unsigned int i = 0; //unsigned number from 0-65535, since we must be able to count (at most) the number of minutes in a day (1440)
volatile unsigned int j = 0; //same as above

//int vib = 0;

byte inactivityArr[500] = {0};

byte one = 1;
byte zero = 0;

Adafruit_BNO055 bno = Adafruit_BNO055();

byte measureS(imu::Vector<3> accelerometer);
byte measureA(imu::Vector<3> linearaccel);

void setup() {

  Serial.begin(9600);
  BTSerial.begin(9600);
  pinMode(5, OUTPUT); // for vibrating motor...

  /* Initialise the sensor */
  if (!bno.begin())
  {
    /* There was a problem detecting the BNO055 ... check your connections */
    Serial.print("Ooops, no BNO055 detected ... Check your wiring or I2C ADDR!");
    while (1);
  }
```

```
  delay(1000);

  bno.setExtCrystalUse(true);
}


void loop() {

  Adafruit_BNO055 bno = Adafruit_BNO055(55);

  imu::Vector<3> accelerometer =
bno.getVector(Adafruit_BNO055::VECTOR_ACCELEROMETER);
  imu::Vector<3> linearaccel =
bno.getVector(Adafruit_BNO055::VECTOR_LINEARACCEL);


  if (BTSerial.available()) {

    BluetoothData = BTSerial.read();

    if (BluetoothData == 'p') { // request for posture data

      int k;
      int minSlouch = 0;
      int minNotSlouch = 0;

      for (k = (i-1); k >= 0; k--) { // i ????????????????

          if ((minSlouch + minNotSlouch) > 1440) {
            break;
```

```
      }

      if (k % 2) { // (k mod 2) == 1
        minSlouch = minSlouch + EEPROM.read(k);
      } else {
        minNotSlouch = minNotSlouch + EEPROM.read(k);
      }

  }

  BTSerial.print("Slouching for ");
  BTSerial.print(minSlouch, DEC);
  BTSerial.println(" minutes");
  BTSerial.print("Not slouching for ");
  BTSerial.print(minNotSlouch, DEC);
  BTSerial.println(" minutes");

}

if (BluetoothData == 'a') { // request for pgysical inactivity data
      int m;
      //byte printArr[1000] = {0}; //placeholder!
      int minInactive = 0;
      int minActive = 0;

      for (m = (j-1); m >= 0; m--) { // i ????????????????

          if ((minInactive + minActive) > 1440) {
            break;
          }
```

```
        if (m % 2) {

          minInactive = minInactive + inactivityArr[m];

        } else {

          minActive = minActive + inactivityArr[m];
        }

      }



      BTSerial.print("Inactive for ");
      BTSerial.print(minInactive, DEC);
      BTSerial.println(" minutes");
      BTSerial.print("Active for ");
      BTSerial.print(minActive, DEC);
      BTSerial.println(" minutes");

}

if (BluetoothData == 'g') { // For sloucing graph

    byte slouchVal;
    byte noSlouchVal;

    int f;
    for (f=0; f<i; f++) {

      if((f%2) != 0) {
        slouchVal = EEPROM.read(f);
```

```
      while (slouchVal > 0) {
        BTSerial.write(130); // considered HIGH value in graph
        slouchVal--;
      }

    } else {

        noSlouchVal = EEPROM.read(f);



      while (noSlouchVal > 0) {
        BTSerial.write(20); // considered LOW value in graph
        noSlouchVal--;
      }

    }
  }
}




if (BluetoothData == 'h') { //

  byte inactivityVal;
  byte activityVal;

  int g;
  for (g=0; g<j; g++) {

    if((g%2) != 0) {
```

```
        inactivityVal = inactivityArr[g];


        while (inactivityVal > 0) {
          BTSerial.write(130);
          //Serial.print(1);
          inactivityVal--;
        }

      } else {

        activityVal = inactivityArr[g];
        Serial.print("Init activity value: ");
        Serial.println(activityVal);
        Serial.write(activityVal);

        while (activityVal > 0) {
          BTSerial.write(20);
          activityVal--;
        }

      }
    }
  }

}


//in the first 10 seconds
if (cnt < 10) {
```

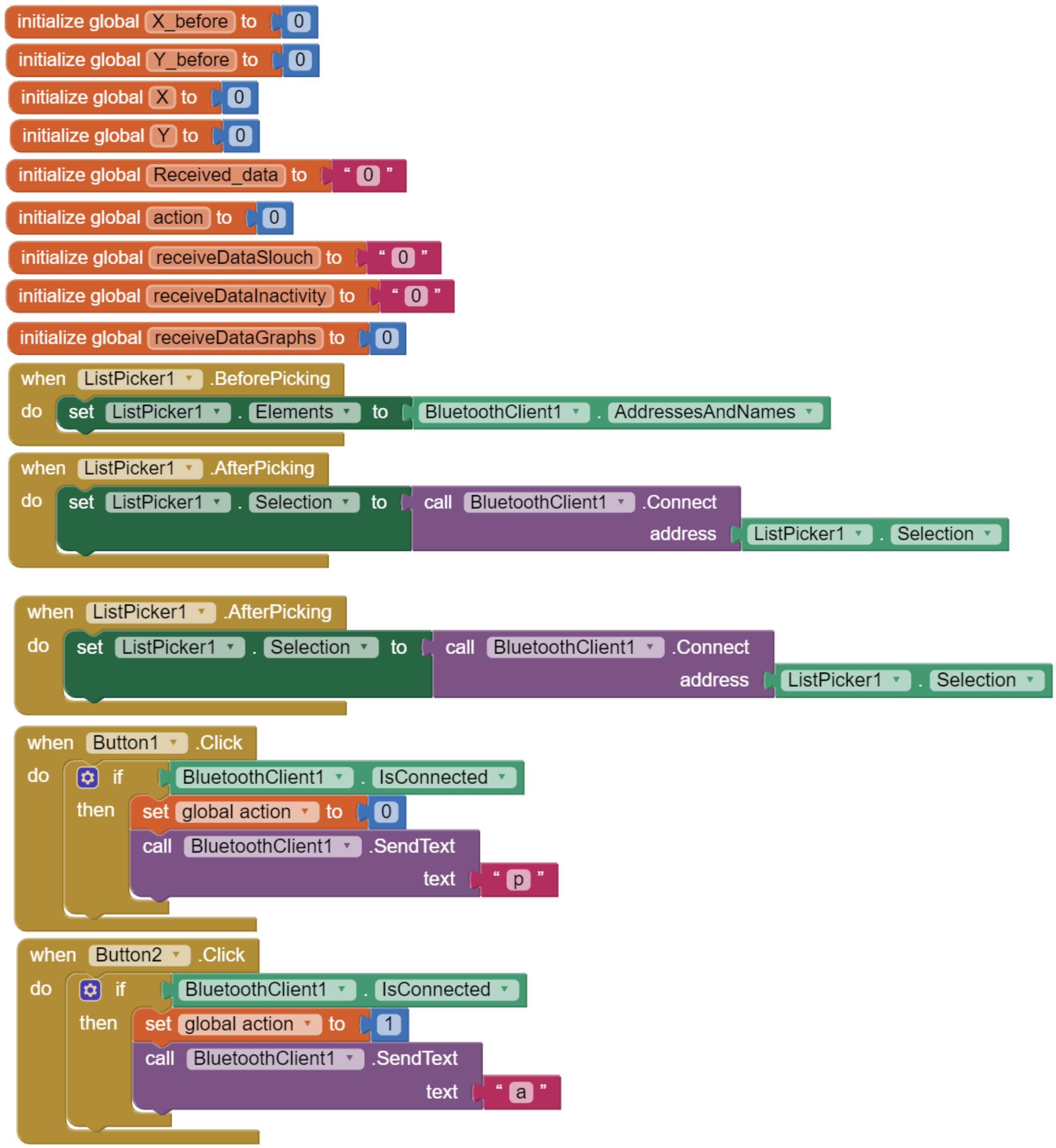
```
  //get measurement(either 1 slouching or 0 not slouching)
  iss = measureS(accelerometer); //measureS returns 1 if slouching, 0 otherwise
  //Serial.print(iss);
  isa = measureA(linearaccel); //measureA returns 1 if inactive, 0 otherwise
  Serial.print(isa);
  scnt = scnt + iss;
  acnt = acnt + isa;
  iss = 0;
  isa = 0;
}


//If the 11th (10) to the 59th (58) second, do nothing (except potential motor vibration).

// At 12th second, vibrate if scnt > 5
if ((cnt == 12) && (scnt > 5)) {

    digitalWrite(5, HIGH);
    cnt++;
    delay(1000);
    digitalWrite(5, LOW);

}



if (cnt == 59) {



  //If the user is slouching this minute
  if (scnt > 5) {
    if (prevs == 1) { //If was slouching last minute
```

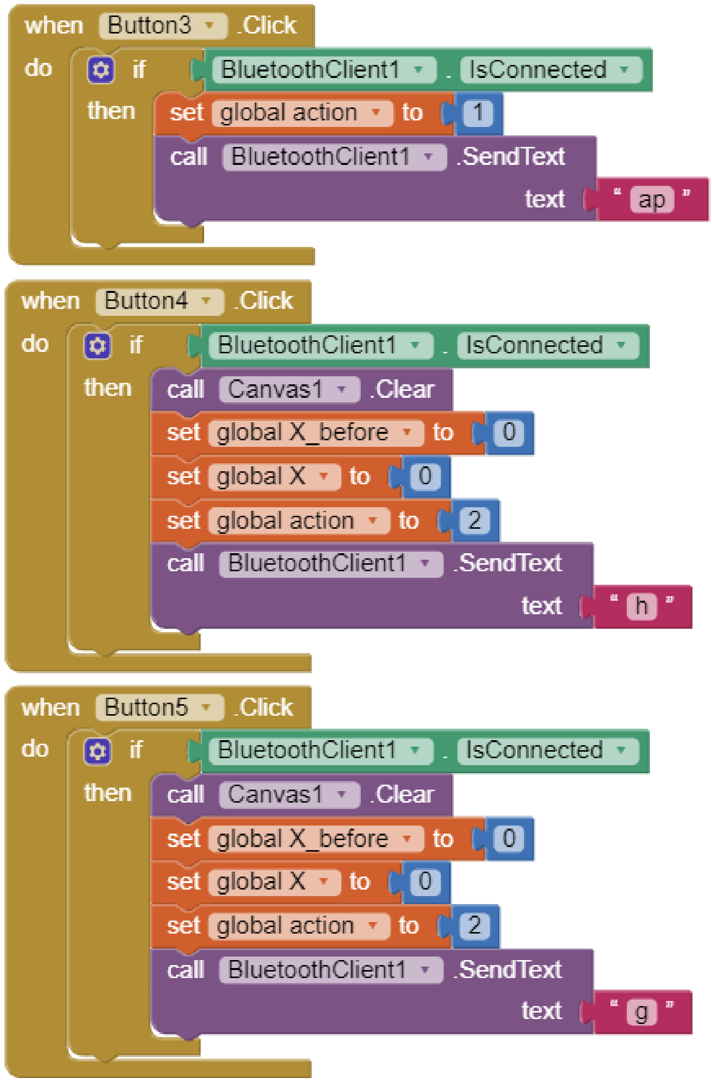
```
          yess++;//add 1 minute spent slouching
        } else { //If was not slouching
          //SEND NOS TO THE ROM ARRAY, this stores the number of consecutive minutes spent not slouching
          EEPROM[i] = nos;
          i++;//move to the next cell for next time
          nos = 0;
          yess++;//add 1 minute spent slouching
          prevs = 1; //was slouching this minute (this info will be used in the next minute)
        }

      } else { // if the user is not slouching this minute (if scnt<=5)

        if (prevs == 1) { //If was slouching last minute
          //SEND YESS TO THE ROM ARRAY
          EEPROM[i] = yess;
          i++;//move to the next cell for next time
          yess = 0;
          nos++;
          prevs = 0;
        } else { //If was not slouching
          nos++;
        }

      }





      if (acnt > 5) {
        if (preva == 1) {
```

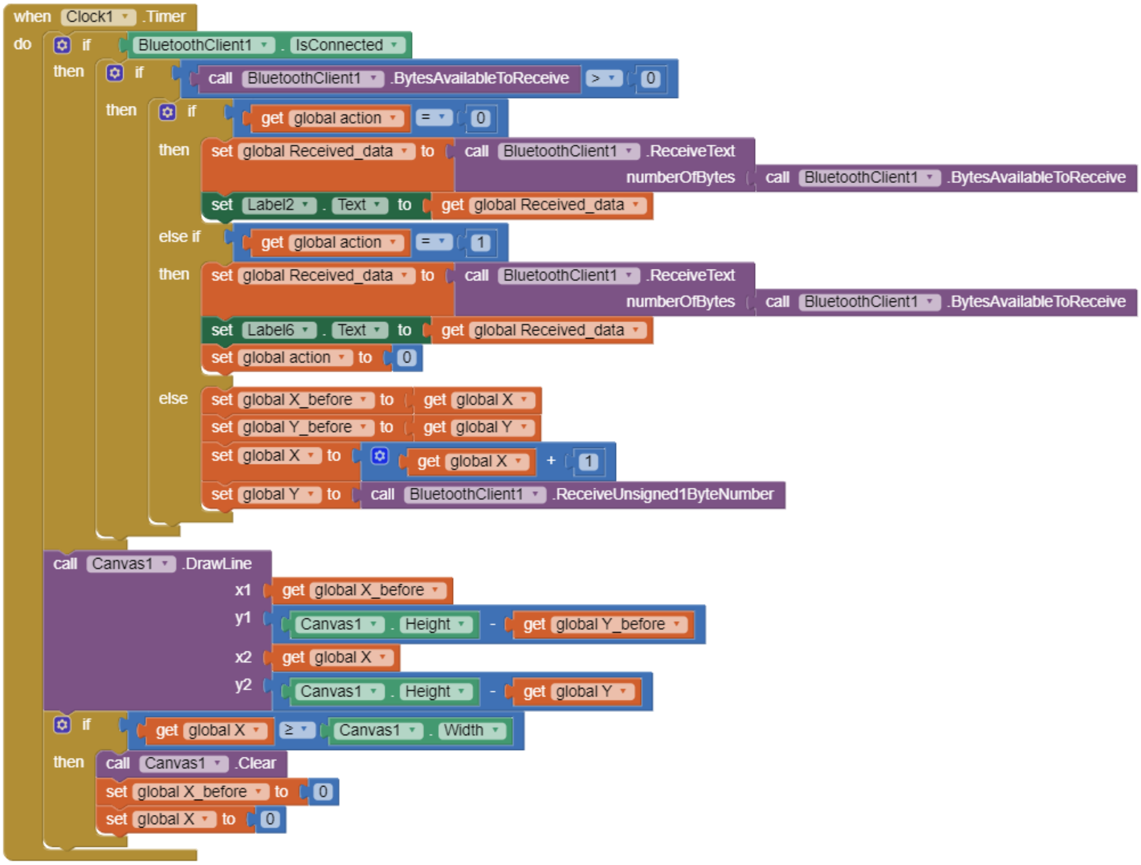
```
      yesa++;//add 1 minute
    } else {
      inactivityArr[j] = noa;
      j++;//move to the next cell for next time
      noa = 0;
      yesa++;
      preva = 1;
    }

  } else {

    if (preva == 1) {

      inactivityArr[j] = yesa;
      Serial.print("YESA ");
      Serial.print(yesa);
      Serial.println(" YESA");
      j++;//move to the next cell for next time
      yesa = 0;
      noa++;
      preva = 0;
    } else {
      noa++;
    }
  }



  scnt = 0; //reset slouch counter within minute
  acnt = 0; //reset inactivity counter within minute
}
```

```
    cnt = (cnt + 1) % 60; // tranverses from 0 to 59 as a representation of a second within a minute
    if (i == 999) i = 0; // MAY CHANGE(CHECK EEPROM)!!!!!
    delay(ONE_SECOND);
  }


  byte measureS(imu::Vector<3> accelerometer) { //function to check if the user is slouching with IMU, returns 1 if slouching 0 if not
    byte slouch;
    if ((accelerometer.y() < 8) && (accelerometer.z() > 6))
      slouch = 1;
    else
      slouch = 0;

    return slouch;
  }

  byte measureA(imu::Vector<3> linearaccel) { //function to check if the user is inactive with IMU, returns 1 if inactive 0 if not
    byte inactive;

    if ((linearaccel.x() <= 0.85) && (linearaccel.x() >= -0.85) && (linearaccel.y() <= 0.85) && (linearaccel.y() >= -0.85) && (linearaccel.z() <= 0.85) && (linearaccel.z() >= -0.85))
      inactive = 1;
    else
      inactive = 0;

    return inactive;
  }
```

MIT App Inventor Code:




when Button3 .Click
do if BluetoothClient1 . IsConnected
then set global action to 1
call BluetoothClient1 .SendText
text " ap "
when Button4 .Click
do if BluetoothClient1 . IsConnected
then call Canvas1 .Clear
set global X_before to 0
set global X to 0
set global action to 2
call BluetoothClient1 .SendText
text " h "
when Button5 .Click
do if BluetoothClient1 . IsConnected
then call Canvas1 .Clear
set global X_before to 0
set global X to 0
set global action to 2
call BluetoothClient1 .SendText
text " g "


when Clock1 .Timer
do if BluetoothClient1 . IsConnected
then if call BluetoothClient1 .BytesAvailableToReceive > 0
then if get global action = 0
then set global Received_data to call BluetoothClient1 .ReceiveText
numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
set Label2 . Text to get global Received_data
else if get global action = 1
then set global Received_data to call BluetoothClient1 .ReceiveText
numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
set Label6 . Text to get global Received_data
set global action to 0
else set global X_before to get global X
set global Y_before to get global Y
set global X to get global X + 1
set global Y to call BluetoothClient1 .ReceiveUnsigned1ByteNumber
call Canvas1 .DrawLine
x1 get global X_before
y1 Canvas1 . Height - get global Y_before
x2 get global X
y2 Canvas1 . Height - get global Y
if get global X ≥ Canvas1 . Width
then call Canvas1 .Clear
set global X_before to 0
set global X to 0