Project #65
Health-Care Neckband

ECE445 Spring 2017

Final Report

TA: John A. Capozzo

Jue Gong

Jiahao Wen

Abstract

The purpose of this project is to build a neckband that users can put on around their necks. A small box containing all the circuits will be attached to the back of users' neck. The neckband will keep track of users' heads tilting angle both forward and backward using an IMU chip, alarming the user via vibration once the user stays in unhealthy posture for a period of time. It also includes a smartphone app that can connect to the neckband, receiving data from the Bluetooth module and storing data on the smartphone. After test and calibration, we managed to measure users' head posture accurately and provide feedback to users when they are at unhealthy postures. In the end, we will discuss some of the future work of this project.

Table of Contents

1.Introduction

1.1 Purpose

Cervical spondylosis, also known as neck arthritis, is a common, age-related condition that affects the joints and discs in your neck [1]. Even though cervical spondylosis is more likely to be present in the elderly, more and more young people are now being diagnosed with cervical spondylosis because of sedentary lifestyle, incorrect posture and overuse of neck muscles. While it is very hard to change people's lifestyle, what we could tackle here is to monitor users' incorrect postures, and remind them when they are in unhealthy postures. Out project targets to help users to understand their posture habits and improve their neck health conditions.

When users put the neckband on around their necks, the sensors will be attached to the back of their necks. The IMU measures users' tilting angles both forward and backward, and keeps track of how much users' necks move for each second. The microcontroller computes the data from IMU, and contacts the motor to give feedback to users. The mobile app keeps track of the collected data, and makes them nicely organized to users.

1.2 Features

1.2.1 Block Diagram and description

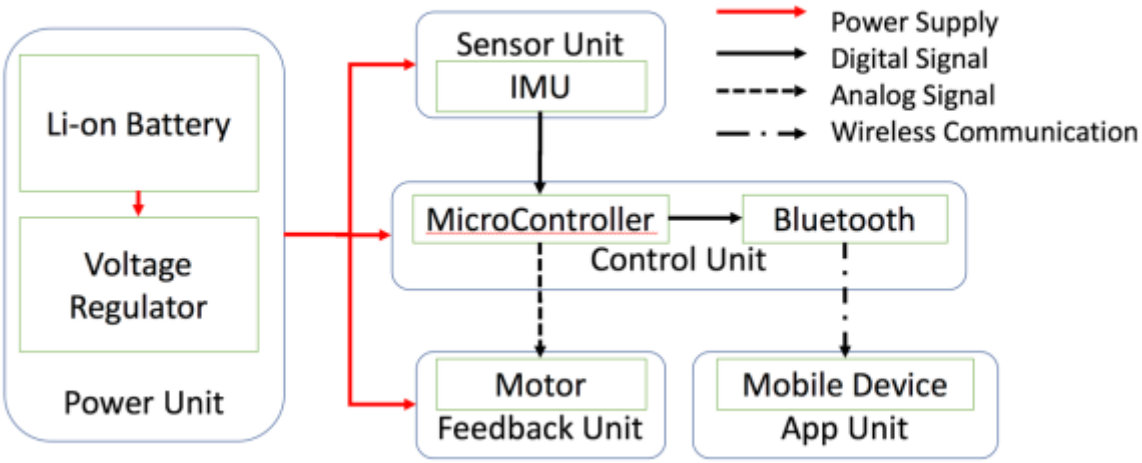

Figure 1.1

Figure 1.1 shows the high-level design composition which mainly includes five parts: Power Unit, Sensor Unit, Control Unit, Feedback Unit and App Unit. We'll discuss the purpose of each unit in the next chapter.

1.2.2 Goals

- Measure users' heads tilting angles both forward and backward accurately with the measurement error less than five degrees
- Alarm users via vibration in three intensity levels when users are in unhealthy posture
- Data collected and stored in smartphone app to help users understand their posture habits

1.3 Summary

The key factor in the performance of our project is how accurate our IMU can be in measuring users' heads tilting angles. We initially used a gyroscope to perform the measurement, but then found out that the drift created during the calculation makes the measured data drifts away from the true value. In order to compensate and correct the drift, we changed the gyroscope into an IMU which integrates a gyroscope, an accelerometer and a magnetometer onto one board. By using sensor fusion, we managed to eliminate drift and get accurate output from the IMU. Because of the change we made to our sensor unit, we were able to meet our initial requirement.

In the following parts of this report, chapter 2 mainly describes our design process. Chapter 3 describes our design verification and results. Chapter 4 describes the design costs. Chapter 5 describes our accomplishments and future works.

2. Design

2.1 Sensor Unit

2.1.1 IMU

We are choosing the MPU-9250 IMU breakout as our sensor unit which contains a three-axis gyroscope, a three-axis accelerometer, and a three-axis magnetometer. This sensor is essential to our design since it helps to measure user neck's current angle versus the neutral position angle.

We initially chose a gyroscope as our sensor unit. Since gyroscope measured the angular velocity, the integration of angular velocity will give us the current angle of the orientation. However, we found out that during the integration process, drift will be created resulting that the measured angle will drift away from the real angle. One solution to this problem is to use additional accelerometer and magnetometer as reference so that drift can be corrected in the three axes of the gyroscope. That's the incentive for us to use IMU as our sensor unit.

Another difficulty is the implementation of sensor fusion. After doing some research on the topic of sensor fusion, we realized that it's rather a complex task to implement our own sensor fusion algorithm. Therefore we decided to use the work written by Kris Winer[2].

2.2 Control Unit

2.2.1 Microcontroller

2.2.1.1 Objective

We chose an ATmega328P microcontroller for our design. The microcontroller will receive digital signal from the IMU sensor and do calculations to get the user's head orientation data. It should monitor the user's posture, and should send signals to the motor to control its vibration state when the user is in an unhealthy posture for a long time. Upon the request of communication from the App, the data will then be sent to the Bluetooth module by the Microcontroller for transmission.

2.2.1.2 Design details

After reading data from the IMU, the Microcontroller could calculate the orientation angle of the head. Now we need to connect the tilting angle to one's health. According to the research we found, the bad posture is defined as a tilting angle of neck that is larger than 15 degrees forward or backward, and one should relax after being in bad posture for 20 to 30 minutes [5]. Also, there is a relationship between tilting angle of the neck and weight applied to the neck, as shown in table 2.2.1.

Tilting angle(degrees)	15	30	45	60
Weight to Cervical Spine(lbs.)	27	40	49	60

Table 2.2.1 Relationship Between Tilting Angle and Weight

Now it is not hard to understand the effect of bad posture, as you can imagine holding a 27-pound mass for 20 minutes. From table 2.2.1, we deduct the polynomial formula which gives the relationship between tilting angle and weight applied to the neck.

$$\text{Weight} = 2.962963 * 10^{-4} * angle^3 - 3.555556 * 10^{-2} * angle^2 + 2 * angle + 4$$

Figure 2.2.2 shows the graph of the formula.

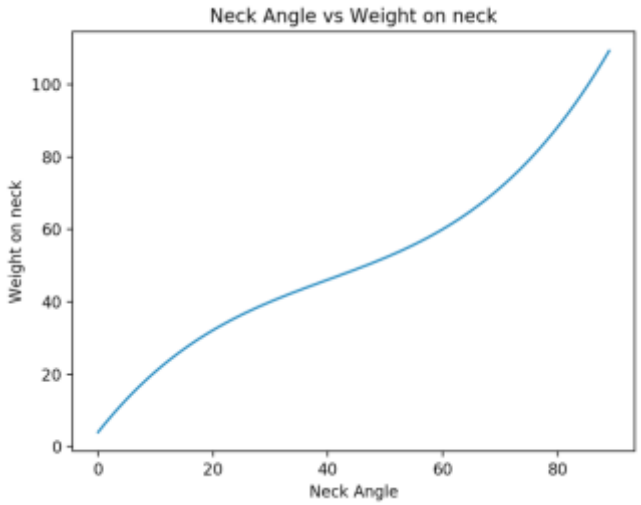

Figure 2.2.2

Because the more weight you bear on your neck, the worse it could be for your neck health condition, we can directly use the weight as score to do further calculation. Below is the flow chart of our algorithm.

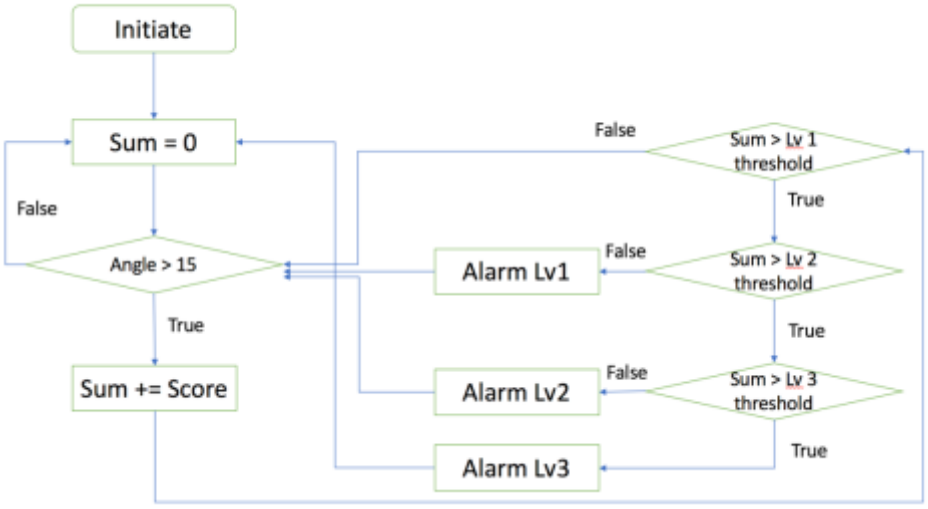

Figure 2.2.3

As mentioned before, it is recommended to relax after being in a bad posture for 20 to 30 minutes. Here we use 3 levels to tell the user how urgently he or she should change to a healthy posture: being in 15 degrees for 20 minutes, 25 minutes, and 30 minutes.

Because the sampling rate of the IMU is 2S/s, we can calculate the value for all three thresholds:

Lv1 threshold = 20min * 60sec/min * 2S/s * 27pt/S = 64800 pts

Lv2 threshold = 25min * 60sec/min * 2S/s * 27pt/S = 81000 pts

Lv3 threshold = 30min * 60sec/min * 2S/s * 27pt/S = 97200 pts

The signals the Microcontroller sends to the motor are analog signals, or Pulse Width Modulation(PMW). The first level PWM has a duty cycle of 33% and is equivalent to a 1.1V input voltage to the motor. The second level PWM has a duty cycle of 45% and is equivalent to a 1.5V voltage. The third level PWM has a duty cycle of 60% and is equivalent to a 1.9V voltage. Figure 2.2.4, 2.2.5, 2.2.6 are testing results of the PWM output.

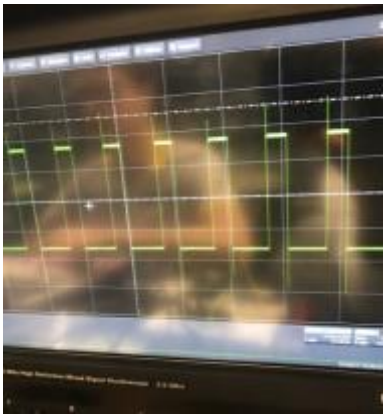

Figure 2.2.4

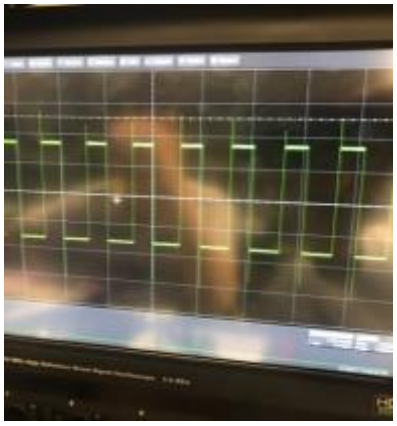

Figure 2.2.5

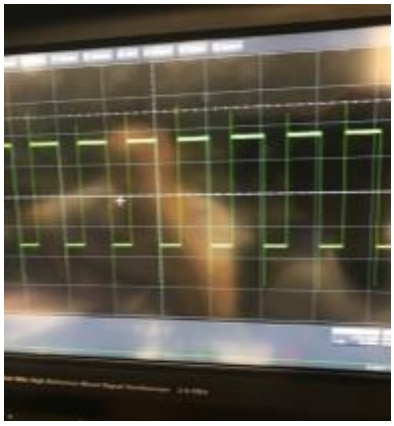

Figure 2.2.6

2.2.2 Bluetooth

The major functionality of the Bluetooth module is to make communication possible between the Microcontroller and the Mobile App. We chose RN-42 XV Bluetooth Module because of its simplicity on implementation and programming [4]. After connecting serial pins on ATMega328p, the Bluetooth module can perfume serial communication with the microcontroller.

2.3 Power Unit

The power unit consists of two 3.7V Li-ion batteries with protection circuit on them. When the battery discharges to below 2.5V, the protection circuit will force the output of the battery to 0V. A linear voltage regulator is used to reduce the input voltage to 3.3V before it supplies power to all other units.

2.3.1 Power Consumption

The main reason that we chose two 3.7V Li-ion batteries each at 400mAh is that they are lightweight with high energy density. Thus, they can sustain the neckband for reasonable amount of time while still fit into our design of a lightweight device. In this section, we will calculate the average usage time that the battery pack can provide to our device.

Battery	Number	Total Power
3.7V Li-ion battery	2	400mAh * 2 = 800mAh

Table 2.3.1 Battery Power

Parts	Voltage(V)	Current(mA)	Power Consumption(mW)
IMU	3.3	3.2	3.3 * 3.2 = 10.56
Microcontroller	3.3	0.3	3.3 * 0.3 = 0.99
Bluetooth	3.3	30(connected)	3.3 * 30 = 99
Motor	3.3	60(in use)	3.3 * 60 = 198

Table 2.3.2 Total Power Consumption

Total energy provided by battery is

$$3.7V * 800\ mAh = 2960\ mWh.$$

Since the parts that always remain active are IMU, Microcontroller and Bluetooth. The constant power consumption per second is

$$10.56 + 0.99 + 99 = 110.55\ mW/s.$$

Suppose our motor vibrates at average 5 times for every hour, and each time for 2 seconds. The approximate average power consumption per second is then

$$60\ mW * 10\ s / 3600\ s = 0.17\ mW/s.$$

Under such assumption, our approximate time for usage is then

2960 mWh / 110.55 (mW/s) = 26.7 hours.

2.3.2 Voltage Regulator

The voltage provided by the battery pack is 7.4V. By using the voltage regulator whose schematic is shown in figure 2.3.3, we manage to get a steady output of 3.3V to all of other units on the PCB.

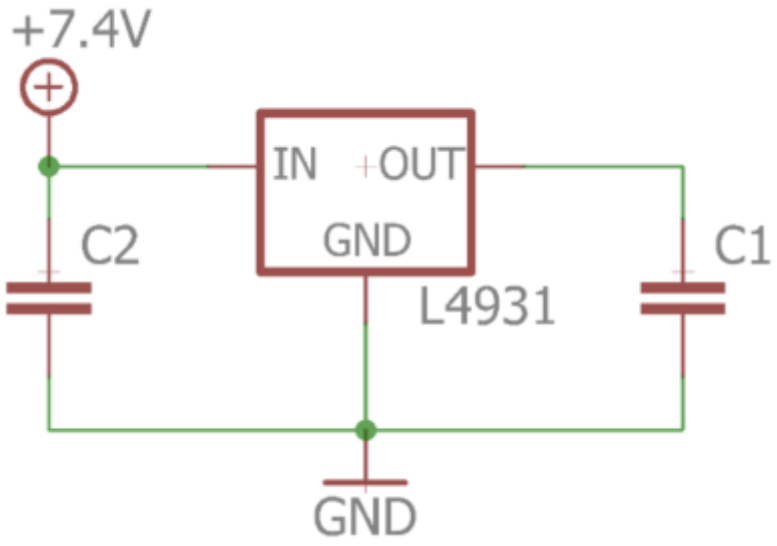

Figure 2.3.3

2.4 App Unit

2.4.1 Objective & Design Procedure

Our Mobile App has four functionalities. Figure 2.4.1 shows a screenshot of the application. The first is to show the user's current neck angle. The second is to tell the user whether he/she is in a healthy posture. The third is to record the time the user is in a bad posture in a day. The last is to record the total tilting angle of the user's neck like a pedometer. For Bluetooth communication, the socket provided by Android is similar to TCP, reliable and ordered, thus only simple messages are needed [6].

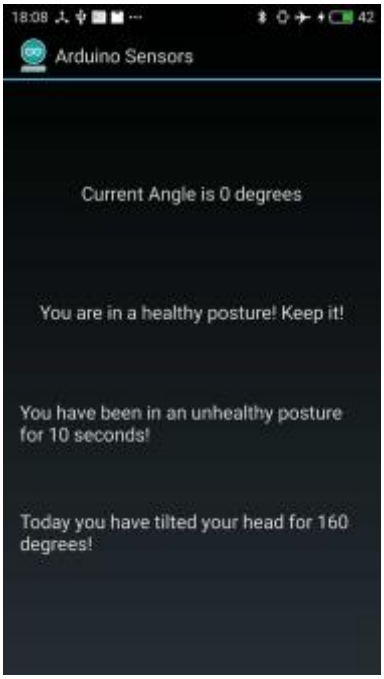

Figure 2.4.1

2.4.2 Design Details

Android use Handler to achieve inter-process communication [7]. A handler object receives messages and runs code to handle messages. Figure 2.4.2 shows a simple flow chart of our application.

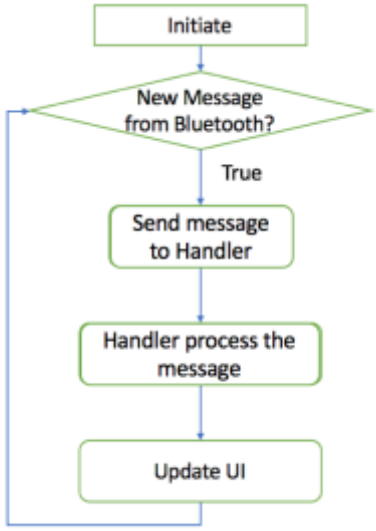

Figure 2.4.2

For message processing within the Handler, we use strings with format “#data~”. “#” indicates the start of the string and “~” indicates the end of the string. The middle data

consists only number strings, so no confusion will be made. By fetching the middle part of the data and parse it back into integer type, we can log the data.

2.5 Feedback Unit

Our feedback consists of a vibration motor that will vibrate when the user stays in unhealthy posture for a period of time. The motor will be able to vibrate at three level of intensities that human beings can perceive.

3.Design Verification

3.1 Power Unit

The power unit consists of two 3.7V battery in series connection, a 3.3V linear voltage regulator, and protection circuit on the batteries.

First of all, we use the multimeter to verify that our battery pack does provide 7.4V as the input voltage when fully charged. We want to exclude any flaws in the power unit that could influence the performance of our project.

Then, to make sure that the 3.3V linear voltage regulator performs as expected, we tested the voltage regulator using inputs from 6V to 9V. The output of the voltage regulator is steadily around 3.3V as shown in figure 3.1.1.

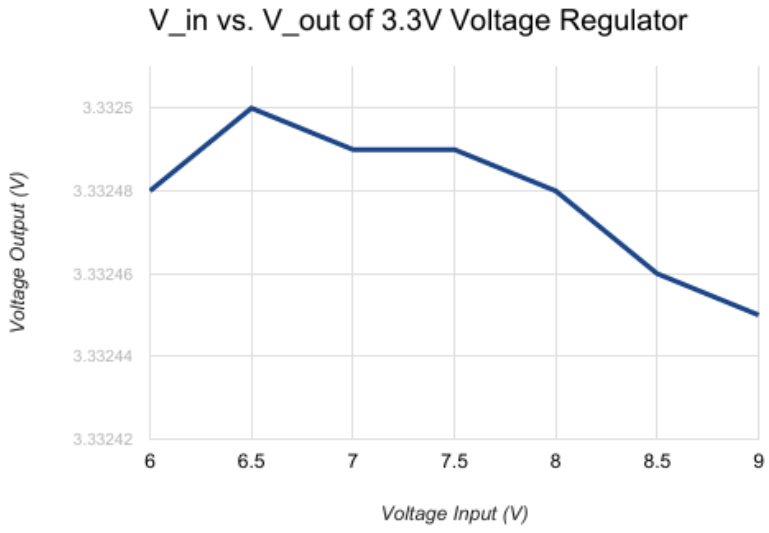

Figure 3.1.1

Finally, we tested the protection circuit by connecting is with DC input from 3.7V down to 0V. As we can see from figure 3.1.2, when the input voltage is larger than 2.5V, output voltage always equals to input voltage. And when we reduce the input voltage to below 2.5V, the output voltage will be regulated below 0.5V, therefore protecting the battery from over discharging.

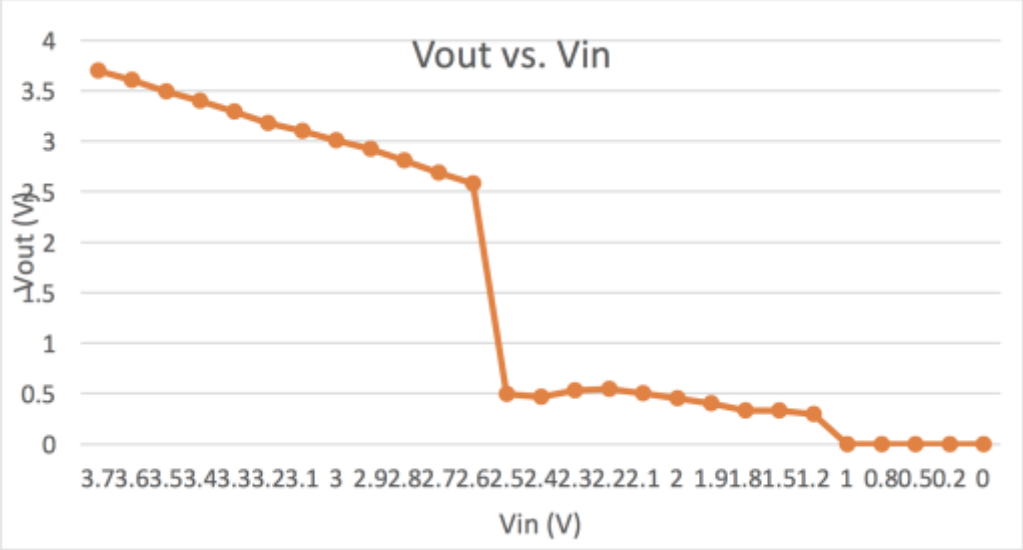

Figure 3.1.2

3.2 IMU sensors

Real Angle from protractor	10°	20°	30°	40°
Measured Angle from IMU	11°	20°	31°	42°
50°	60°	70°	80°	90°
53°	64°	74°	85°	85°

Table 3.2.1 Testing Result of IMU

After we integrates the IMU with a microcontroller, we tested the accuracy of the IMU as figure 3.2.2 shows. As we tilted the IMU chip from 10 degrees to 90 degrees, we use a protractor to measure the real angle. Then we compared the measured angle from IMU

with the real angle. Table 3.2.1 shows the results. As the tilting angle becomes larger, the error also grows larger but at most to 5 degrees which meets our requirement.

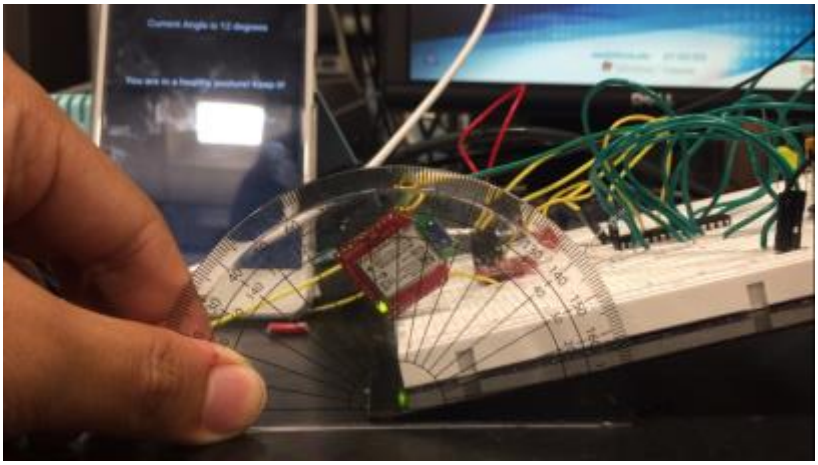

Figure 3.2.2

3.3 Microcontroller

In our verification test, we set the threshold for three levels to be 1000pts, 2000pts and 3000 pts. After connecting the IMU and the Microcontroller, we use an oscilloscope to monitor the output signal of the Microcontroller. We place the IMU at different angles and compare the difference of the theoretical time and the real time to the first alarm (first PWM signal). The results are shown in table 3.3.1.

Angle(degrees)	20	30	40	50	60	70	80	90
Theoretical time to first alarm(s)	16	13	11	10	8	7	6	4
Real time to first alarm(s)	16	12	11	9	8	6	5	5

Table 3.3.1 Testing Result of Microcontroller

According to table 3.3.1, if we use the original threshold, the error will be within 1 minute, which meets our requirement.

3.4 Software

First, we tested whether the App could connect to the Microcontroller through Bluetooth. We place the IMU at different angles and see if the data is correctly displayed on the UI. Then we tested whether the data is synchronized between the App and the Microcontroller. We send the total score to the app and display it on the screen.

By changing the angle of the IMU, we read the total score from the UI and see if the Microcontroller is output the correct PWM signal when the total score reaches each threshold.

3.5 Feedback Unit

As we use the PWM to control the different magnitude of vibration intensities of the motor, we want to illustrate that these three levels of intensity can be distinguished by human. By using PWM to control the motor, the microcontroller will drive the motor using voltage of 1.1V, 1.5V, and 1.9V.

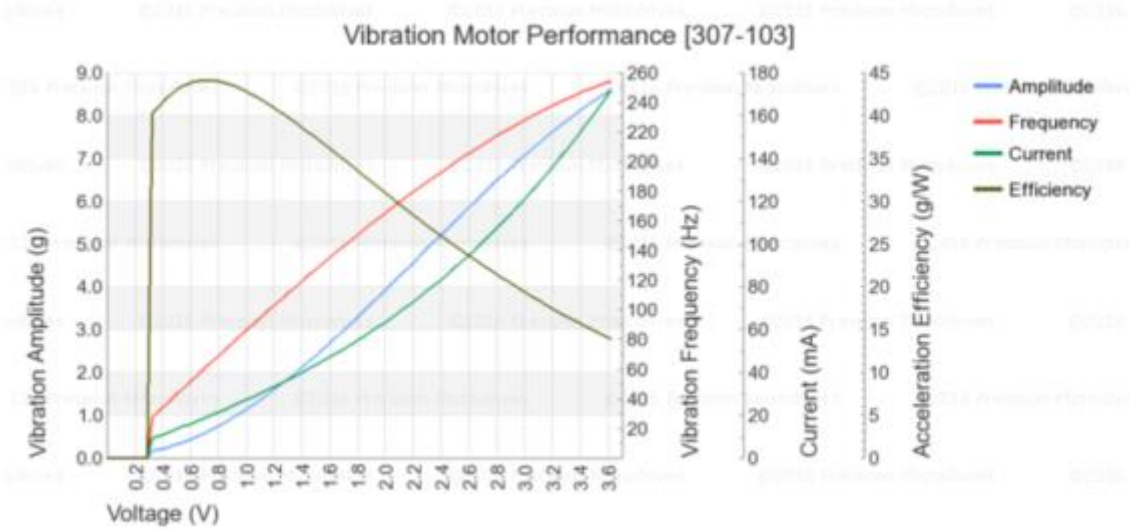

Figure 3.5.1

Figure 3.5.1 shows the motor performance in different driving voltages and frequencies [3]. We can see that 1.1V, 1.5V and 1.9V driving voltage equal respectively vibration amplitude of 1.3g, 2.5g and 3.9g. Because we cannot find any related paper about human perception of vibration, we tested the three different vibration amplitudes using our fingers, and the three level of intensity can be distinguished easily.

4.Cost

4.1 Labor Cost

Total = 2.5 * Hourly Rate * Hours in Total

Name	Hourly Rate	Hours in Total	Total
Jiahao Wen	$30	200	$15,000
Jue Gong	$30	200	$15,000
Total			$30,000

Table 4.1 Labor Cost

4.2 Parts

Vendor	Part Number	Part Description	Unit Cost	Quantity	Total Cost
Sparkfun	COM-09061	Microcontroller	$4.3	1	$4.3
Sparkfun	PRT-10401	Lipo Charger	$7.95	1	$7.95
Sparkfun	PRT-13853	Lithium Ion Battery	$4.95	2	$9.9
Sparkfun	SEN-13762	IMU	$14.95	1	$14.95
Sparkfun	WRL-11600	Bluetooth Module	$29.95	1	$29.95
Sparkfun	COM-00526	Voltage Regulator	$1.95	1	$1.95
Sparkfun	COM-14087	Flip-pin	$1.95	2	$3.9
Total					$72.9

Table 4.2 Parts Cost

4.3 Total Cost

Section	Total
Labor	$30,000
Parts	$72.9
Total	$30,072.9

Table 4.3 Total Cost

5.Conclustion

5.1 Achievements

We managed to achieve all functionality requirements of this design. The power unit works properly and steadily, supplying power to the neckband for over 24 hours. The measurements taken from the IMU provides sufficient accuracy to support the algorithm of our design. The microprocessor can communicate with IMU and gives appropriate output signal to control the vibration motor. The Bluetooth module managed to communicate with the android phone app we built, and the app can collect and store data in mobile phone as required.

5.2 Uncertainty

The neckband we created is intended to measure users head posture. If users bend over their backs without tilting their head forward or backward, the neckband will still show that users were in unhealthy postures. This is due to the nature that our neckband works at a global reference coordinates regardless of the tilting angle of the back. In the future work section, we will address this issue. However, this will not influence the functionality of our design, since people are likely to bend over their back for a long time. The problem aroused from this issue is that the data collected may be inaccurate due to the mistakenly taken data from bending over one's back.

5.3 Ethics and Safety

One of the greatest safety concern for our project is the battery. Just as every other wearable technology application, the safety of the users is always the priority. And because our device will possibly have direct contact to human skin, one minor mistake in the design could lead to severe harms to users' health. Thus, we are dedicated to build a device that ensures the least possibility of electrical shock, burns and fire. To

achieve the safety goal and obey the IEEE Code of Ethics #1, the power supply we use must adhere to the US safety standard UL60950-1[8].

Another concern is the accuracy of the data and computed result. Because our product aims to monitor the user's neck movement and sitting posture, inaccurate data will not only be ineffective for improving users' health condition but also misleading and potentially harmful. In order to address this issue, we will test and calibrate our product carefully and strive to minimize error within 5 degrees. If we are unable to achieve the expected accuracy, we will re-design our sensor system by adding more sensors or place them in different positions.

We understand that privacy is a crucial part to users, and to ensure we are following ACM code of Ethics and Professional Conduct 1.7, we'll take appropriate means to protect the data collected from illegal access or accidental disclosure to inappropriate individuals [9]. All the data collected from users will not be used for other purpose without users' consent.

Moreover, in order to follow IEEE Code of Ethics #5, we need to inform users of what groups of people our product is suited to and the potential consequence of using our product [8]. We'll provide proper documentations and warning labels to users to help them better understand and use our product.

5.4 Future Work

First, the basic of building a wearable device is that people can wear the device comfortably and safely. The prototype we built is far from the perfect size we want for a wearable device around users' necks. The reason that we cannot reduce the size of our project is that we bought high-level chips from different manufactory. The chips themselves already take too much space. Moreover, integrating these chips onto a PCB requires more space for the wires and connections. Thus, one thing we can do in the future is to build a PCB with more basic chips that fit specially for our needs. Currently we are not using every chip to the fullest. Some functionalities of the chips are not used at all. By creating a specially designed PCB, we can reduce the size of our project, and make it more user friendly.

Moreover, the neckband currently only measures the user's tilting angle of the pitch axis

as figure 5.4.1 shows. In the future, we want to expand our functionality to simultaneously measure user's head in yaw, pitch, and roll axes.

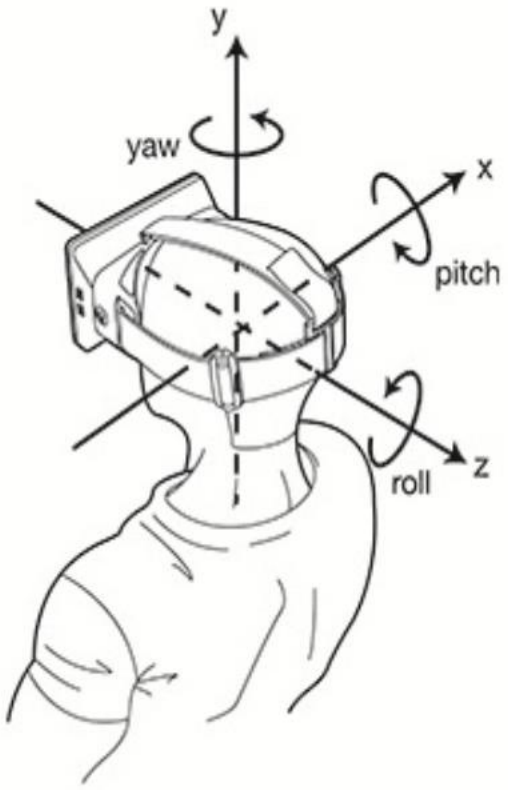

Figure 5.4.1

Another deficiency of our design is the ability to differentiate between user neck's tilting angle and user back's tilting angle. One way to solve this problem is to attach one more sensor unit to user's back, and by comparing data from user's neck and user's back, we will be able to calculate the real angle of how much degree the user tilted his head.

Appendix A Requirement and Verification Table

Module	Requirement	Verification	Score
IMU	(1) Output range of the angle should be +/- 90 degrees in y-axis of the IMU; (2) Measure the angle with accuracy of +/- 5 degrees;	(1) Connect to an Arduino chip with testing code. Real angle is measured with a protractor. Place the IMU in positive and negative 0°, 30°, 60°, 90° in three axes and read data from the Arduino chip. Compare the data to see the accuracy.	15
Microcontroller	(1) With input data from the IMU, it can output correct and accurate digital signal from 2 output pins to the MUX to control the intensity of the motor. (2) The error of the time before the motor alarms the user is within 1 minute (3) Must be able to send serial data of the current angle of IMU to Bluetooth at baud rate 37400.	(1) (After verifying the IMU) Connect the IMU to the Microcontroller. Place the IMU at 15 degrees for a period of time, and the microcontroller should output a digital signal designating level 1 of intensity to the MUX. Continue Holding the IMU at the degree of 15, the microcontroller will then send digital signal denoting level 2 and level 3. (2) Place the IMU at 15 degrees, record the time when the motor reaches level 1, level 2 and level 3 intensities. Then place the IMU at 30 degrees, record again the time when the motor reaches level 1, level 2 and level 3 intensities. The time taken should be much shorter. (3) Connect the smartphone to the Bluetooth module, if the smartphone App can receive measurements from IMU, then the microcontroller is correctly sending data to the Bluetooth module.	10

Motor	(1) The rotation speed of the motor should be able to vibrate at 3 distinguishable levels at 5000rpm, 7500rpm, 10000rpm.	(1) Test the motor at different voltage, record the vibration amplitude. In real circuit, we will use digital signal from the Micro to MUX to control the input voltage of the motor.	5
Power Unit	(1) Provide 3.3+/-0.3 V voltage to all modules of the system. (2) Can supply power to the system for over 18 hours. (3) For protection circuit, if the input voltage is higher than 4.0V or lower than 2.7V, the protection circuit will cut of the circuit.	(1) Connect the output of the Power Unit to a voltage meter. Read from the voltage meter to see if the output voltage is within the range. (2) Calculate the power consumption of or product and do calculation. (See below) (3) Test the protection circuit individually. Connect the input of the protection circuit to a voltage generator. Connect the output of the protection circuit to a voltage meter. Given different input in range 2.7V to 4.0V, see if the circuit still works by reading from the voltage meter. Given input less than 2.7V and input higher than 4.0V, see if the circuit is cut off by reading from the voltage meter.	5
Bluetooth	(1) Must be able to receive data from the microprocessor. (2) Must be able to communicate with Android device.	(1) Connect the Bluetooth module to the microprocessor by wire, and use a smartphone to connect the Bluetooth module. Then use the smartphone APP to check whether the Bluetooth module is sending the correct data.	10

Mobile App	(1) Must be able to receive data from the Bluetooth module, and show the data on the smartphone	(1) Connect the smartphone to the Bluetooth module, and check if there are any incoming data from our IMU sensor	5

Table 6 R&V Table

Appendix B Core Mobile App Code

```
bluetoothIn = (Handler) handleMessage(msg) → {
        if (msg.what == handlerState) {                                          //if message is what we want
            String readMessage = (String) msg.obj;                               // msg.arg1 = bytes from connect thread
            recDataString.append(readMessage);                                   //keep appending to string until ~
            int endOfLineIndex = recDataString.indexOf("~");                     // determine the end-of-line
            int startOfLineIndex = recDataString.indexOf("#");                   // determine the start-of-line
            if (endOfLineIndex > 0) {                                            // make sure there is data before ~
                String dataInPrint = recDataString.substring(startOfLineIndex + 1, endOfLineIndex);    // extract string
                sensorView1.setText("Current Angle is " + dataInPrint + " degrees");
                int angle = Integer.parseInt(dataInPrint);
                if (angle > 15) {
                    sensorView0.setText("You are in an unhealthy posture!");
                    time += 0.5;
                }
                else
                    sensorView0.setText("You are in a healthy posture! Keep it!");

                int angle_diff = Math.abs(angle - last_angle);
                last_angle = angle;
                total_angle += angle_diff;

                int showtime = (int)time;
                sensorView2.setText("You have been in an unhealthy posture for " + showtime + " seconds!");

                sensorView3.setText("Today you have tilted your head for " + total_angle + " degrees!");

                recDataString.delete(0, recDataString.length());                    //clear all string data
               // strIncom =" ";
                dataInPrint = " ";
            }
        }
```

Figure 7

Appendix C Supplementary Figures

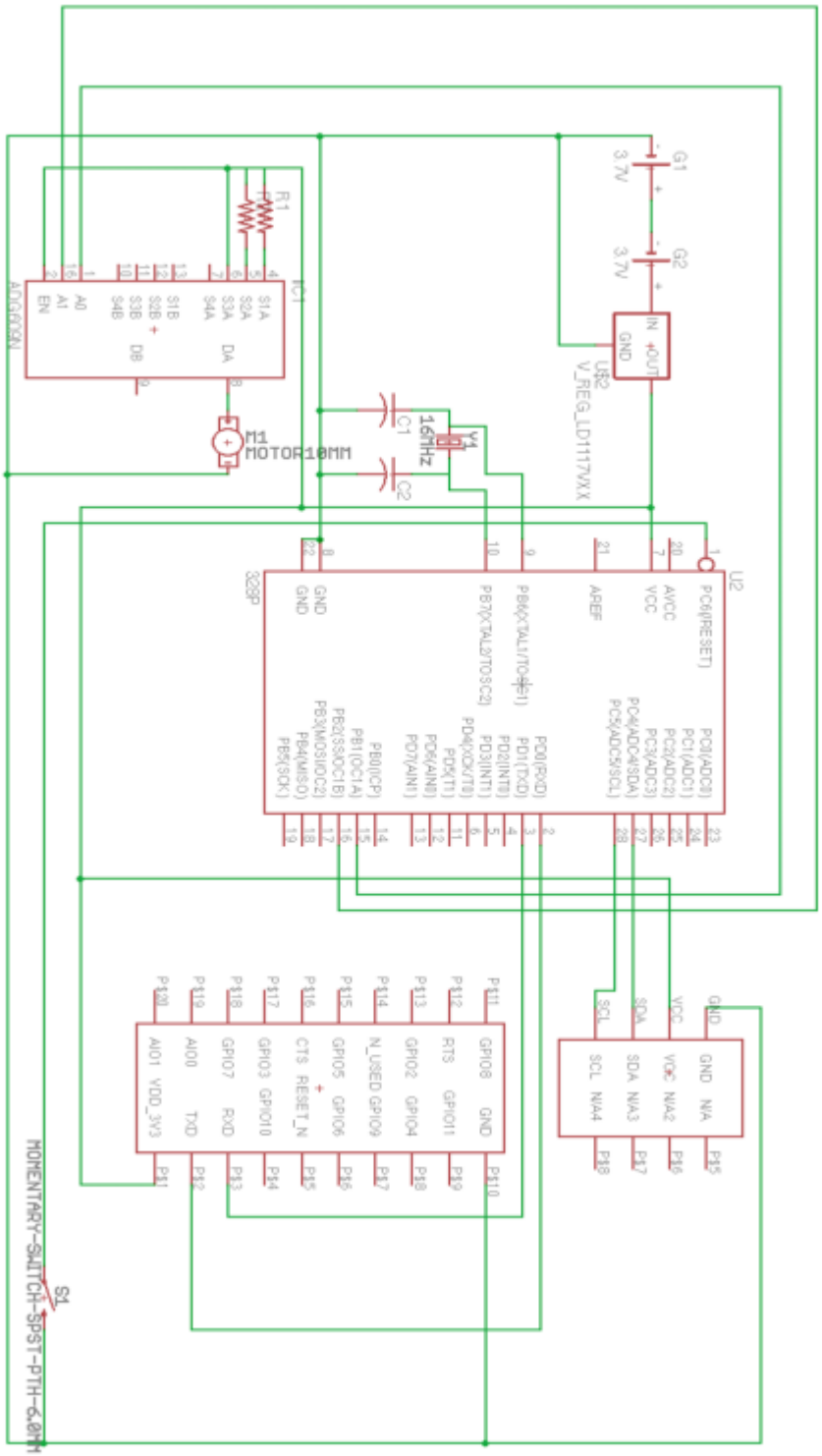

Figure 8. Schematic of overall circuit

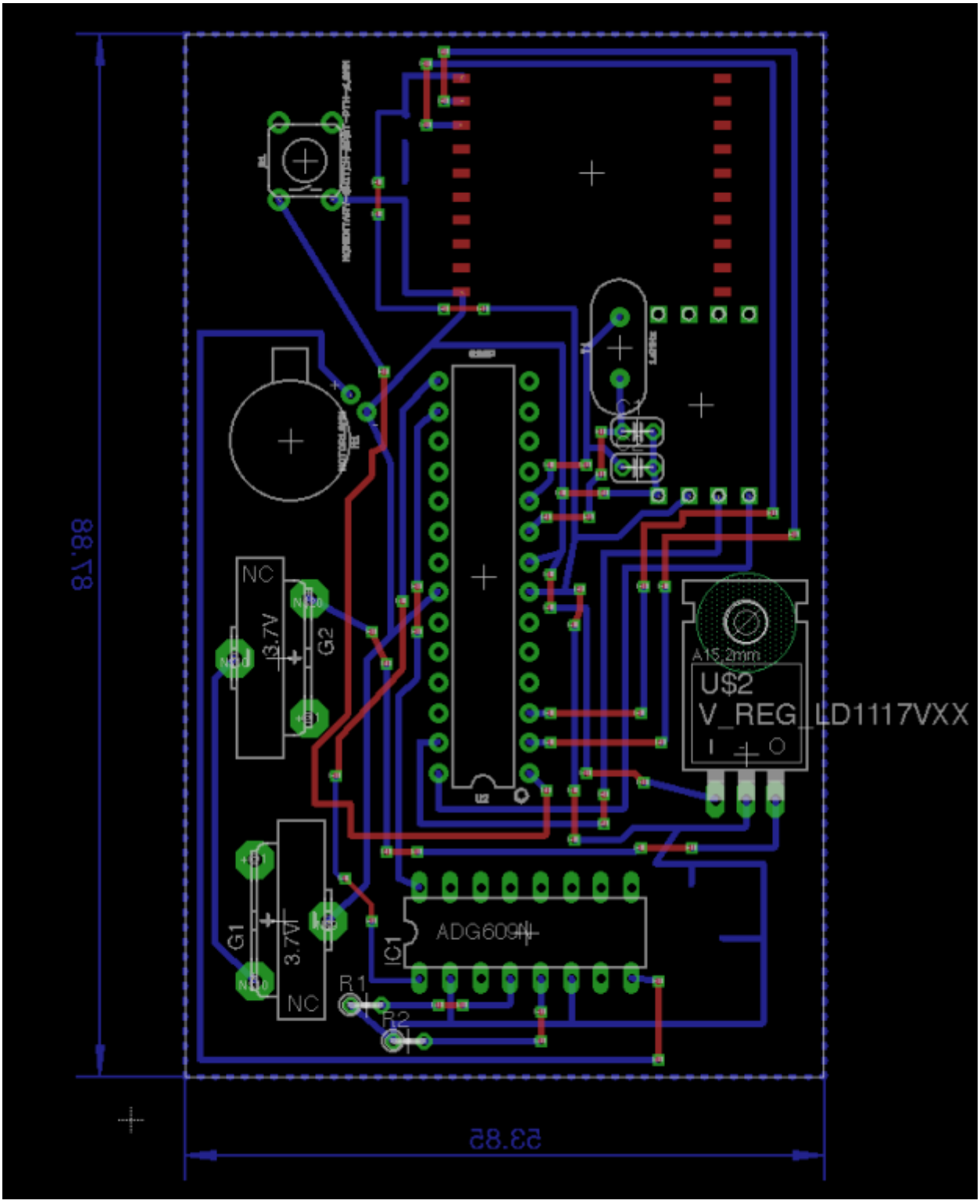
88.78
53.85
NC
3.7V
G2
G1
3.7V
NC
ADG609
IC1
R1
R2
C1
A15.2mm
U$2
V_REG_LD1117VXX

Figure 9. PCB

References

[1] A. Delgado & R. Nall. (2015). *Cervical Spondylosis* [online]. Available: http://www.healthline.com/health/cervical-spondylosis#Overview1

[2] K. Winer. (2016). *Arduino sketch for MPU-9250 9DoF with AHRS sensor fusion* [online]. Available: https://github.com/kriswiner/MPU-9250

[3] Precision Microdrives. (2016). AB-029 : VIBRATION MOTORS - VOLTAGE VS FREQUENCY VS AMPLITUDE [online]. Available: https://www.precisionmicrodrives.com/application-notes/ab-029-vibration-motors-voltage-vs-frequency-vs-amplitude

[4] Roving Networks. (2012). RN41XV & RN42XV Bluetooth Module [online]. Available: http://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/RN42XV.pdf

[5] Kansas Chiropractic Foundation. (2016). *Improving Kansas Health* [Online]. Available: http://www.kansaschirofoundation.org/goodposture-article.html

[6] Google, "BluetoothSocket," [Online]. Available: https://developer.android.com/reference/android/bluetooth/BluetoothSocket.html

[7] Google, "Communicating with the UI Thread," [Online]. Available: https://developer.android.com/training/multiple-threads/communicate-ui.html

[8] Ieee.org. (2017). *IEEE Code of Ethics* [online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html

[9] acm.org. (2017). *ACM Code of Ethics and Professional Conduct* [online]. Available: https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct