A Project Report on

Posture Senseii

Submitted in partial fulfillment of the requirement for the course

CS533 – INTERNET OF THINGS

SUBMITTED BY

Karthik Krishna 1760641

Basil George Poulose 1761002

Nikhil Niranjan 1761020

Department of Computer Science and Engineering

Faculty of Engineering,

CHRIST(Deemed to be University),
Kumbalgodu, Bangalore - 74.

September 2019.

A Project Report on

Posture Senseii

Submitted in partial fulfillment of the requirement for the course

CS533 – INTERNET OF THINGS

SUBMITTED BY

Karthik Krishna 1760641

Basil George Poulose 1761002

Nikhil Niranjan 1761020

Faculty-In-Charge Examiner Head of the Department

Department of Computer Science and Engineering

Faculty of Engineering,

CHRIST(Deemed to be University),
Kumbalgodu, Bangalore - 74.

September 2019.

Abstract:

In 21st Century, maximum percentage of our population suffers from back pain, injuries, neck pain and shoulder problems etc that leads to bad posture and interferes with our daily activities. Posture senseii is an IOT based posture detection and warning system that warns us about our incorrect posture by detecting changes occurring in our posture using an accelerometer sensor. The changes occur in different directions and are detected using the sensor, by calculating the angles according to the tiltation of a body. Indication of incorrect posture is provided to the user by the LED light, buzzer and through the Dabble app in our phone.

CONTENTS:

Introduction:

Sitting and standing put considerable pressure on the lower back; standing exerts five times more pressure than lying down, and sitting, surprisingly, is even more strenuous. Poor sitting posture is believed to be a major contributor to low-back pain and isn't good for your back either. In addition to helping prevent back and neck problems, good posture is important in positive ways. It improves your appearance and helps you project self-confidence and self-assurance. It can help you mentally and emotionally. And certainly it is worth achieving just for the aches and pains it may prevent.

As long as people aren't actually in pain, they tend to forget how delicately their backs are engineered. Poor posture can strain both muscles and ligaments, making you more vulnerable to injury—as well as complicating such everyday tasks as carrying groceries or even sitting at a desk.

An improperly aligned spine may narrow the space between vertebrae, thereby increasing the risk of compressed nerves. Posture is not simply what happens when you are sitting or standing still—it's also dynamic, and includes your posture when you move. Poor posture may include many elements—rounded shoulders, protruding buttocks and abdomen, overly arched lower back, and the head pushed forward into an exaggerated position.

Problems caused by bad posture:

Poor posture may be caused by many factors, including previous injuries, disease, poor muscle tone, and emotional stress. A sedentary lifestyle can reduce muscle tone and strength and lead to bad posture. Sore, aching feet have a negative effect on posture, too. Foot pain may mean simply that you're choosing the wrong shoes. Or you may need special supports—orthotic devices—in your shoes and an evaluation by a podiatrist. One very important factor is habit.

Contrary to what some people believe, straightening up now and then isn't enough: you need to be aware of—and to practice—other strategies to improve standing and sitting. In addition, fatigue can result in bad posture. In itself, poor posture isn't a health problem. But it won't improve without some effort on your part, and in the meantime it can have an adverse impact on your musculoskeletal system. If you don't take steps to improve your posture, you may eventually limit your lung expansion—which means less energy available to your body and brain—and develop chronic muscle aches, including headaches and back pains.

Outcome of the Project:

The intended outcome of this project to help resolve all the above mentioned problems and to help everyone have a healthy lifestyle. This projects presents us with a wearable device on your back and it's function is to detect changes in your posture and alerts the user accordingly using the the buzzer on the device and the Dabble app used in the phone which gives us a notification.

2) Design & Implementation:

Components:

- Breadboard
- Jumper wires
- HC-05 Bluetooth module
- Arduino Nano
- ADXL345 3-Axis Accelerometer
- A buzzer
- An LED light

Sensors:

Accelerometer:

- Movement in the body, and changes in posture will change angles measured by the accelerometer. Thus, body movement will act as input to the accelerometer .
- Measures the acceleration at the base of the curve of the spine and the top of the curve of the spine. Thus, calculate the angle by which the person is bending and check if threshold is crossed.
- These values are sent directly to the Arduino Nano which can be used to calculate the angle differential.

Arduino Nano:

- Receives data from the sensors and also the HC-05 Bluetooth module.
- Computes the angle of the back through the accelerometer data and also sees if the muscles are flexed. If any of these values cross the specified threshold values, then it alerts the user using external devices..
- Depending on the settings the Arduino Nano sends a signal to the buzzer and the Dabble application in the phone if the posture is not acceptable.

HC-05 Bluetooth Module:

- Takes the angles and muscle flex data from the microcontroller and relays them to a computer.

- If there is a change in the setting, relays back the new parameters.
- We will try to have a smart algorithm here that uses minimum energy to transfer data.
- The output can be seen on a computer in form of a graph maybe.

Power Supply:

- The only input will be a charger such the battery pack can be recharged. Given time, we would like to implement a charger module as well.
- The output would be controller current and voltage for which a regulator can be used.
- We can simply power the device using small portable 10W batteries (2 AA).

Architectural Diagram:

The architectural diagram is as follows:

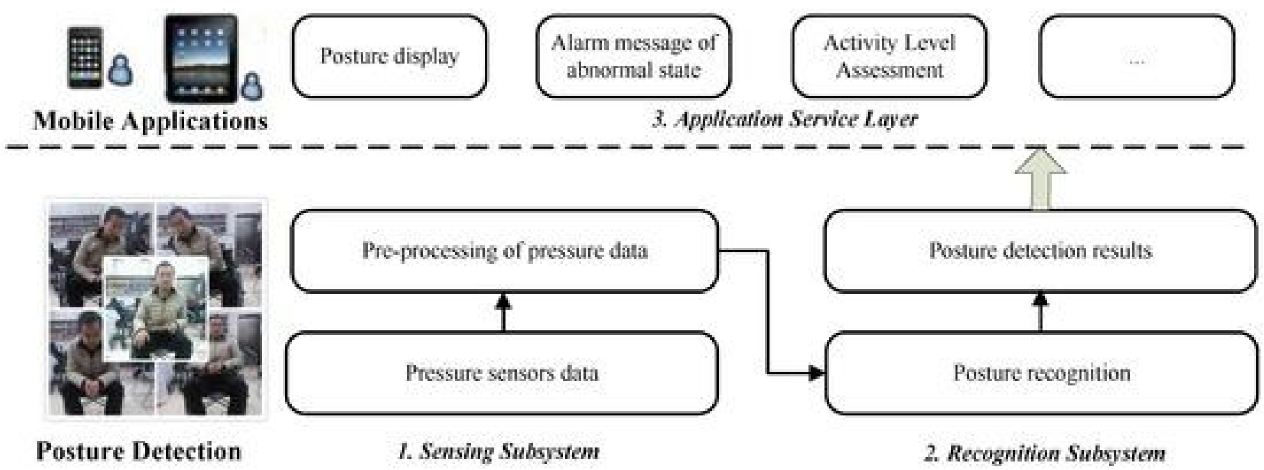


Fig 2.1 Architectural Diagram of Back Posture detector

Pin Diagram:

The pin diagram of Arduino Nano used in the project is as follows:

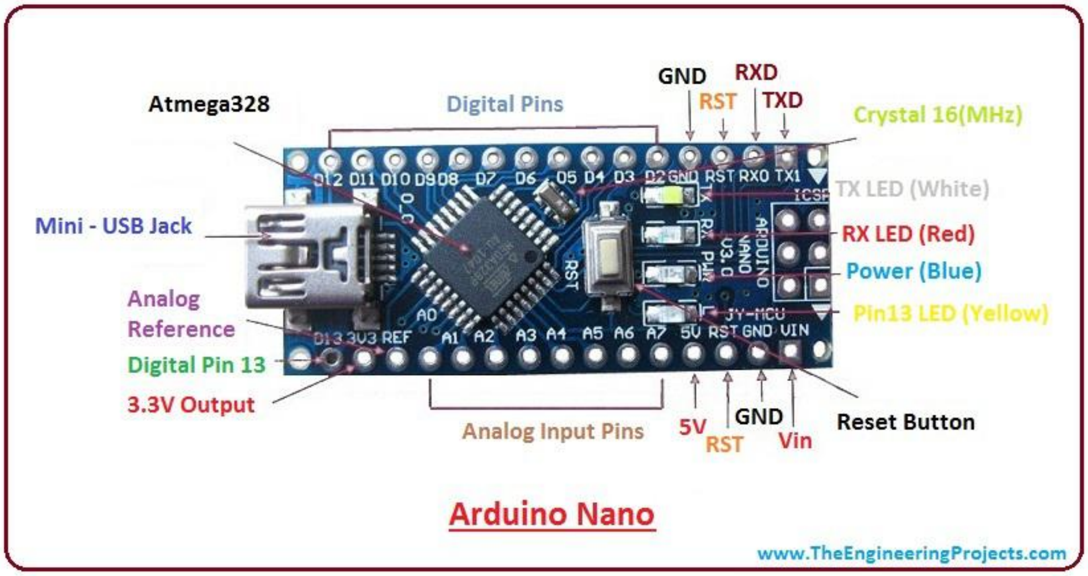


Fig 2.2 Pin Diagram of a Arduino Nano board

Overall Diagram:

Assemble the components by following this circuit diagram:

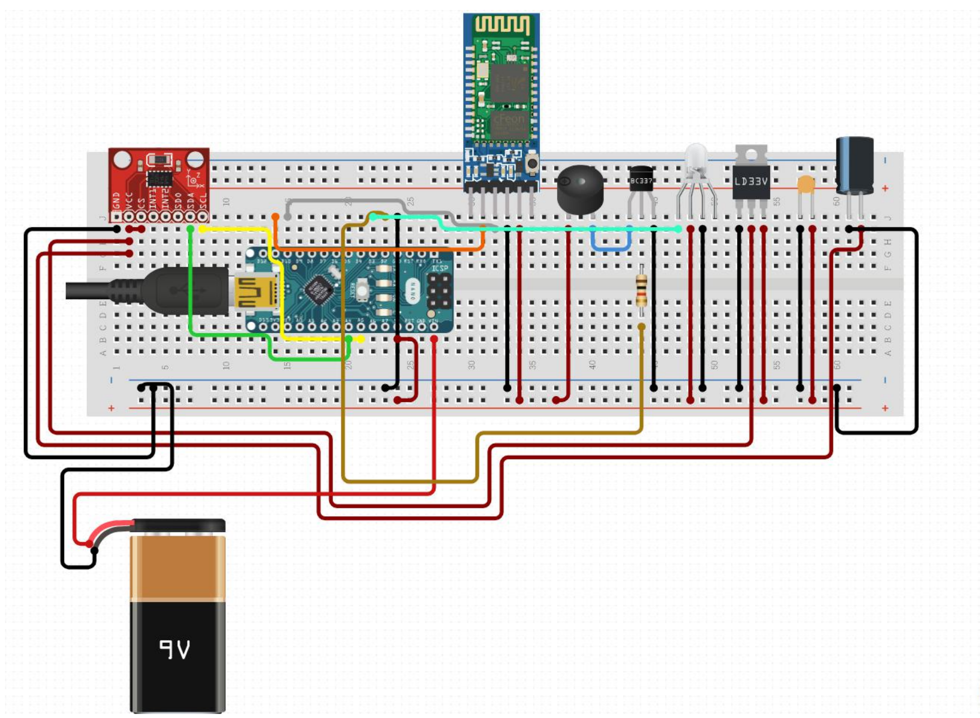


Figure 2.3 Final connections made for the device

Connections made with the Arduino Nano according to Fig 2.3 is as shown in the table 1 :

No.	ARDUINO NANO	ADXL 345	BLE HC-05	LED	BUZZER
1)	A4	SDA	-	-	-
2)	A5	SCL	-	-	-
3)	D2	-	TXD	-	-
4)	D3	-	RXD	-	-
5)	D13	-	-	Positive	-
6)	5V	VCC	VCC0)	-	-
7)	GND	GND	GND	-	Negative
8)	-	-	-	Negative	Positive

Table 1

3) **Testing, Outcome & Analysis:**

The system was worn by one person for continuously for sometime and was under examination the entire time. For simple functioning of the Posture Senseii, we assembled all the components onto the breadboard and stick it to the subject's back.

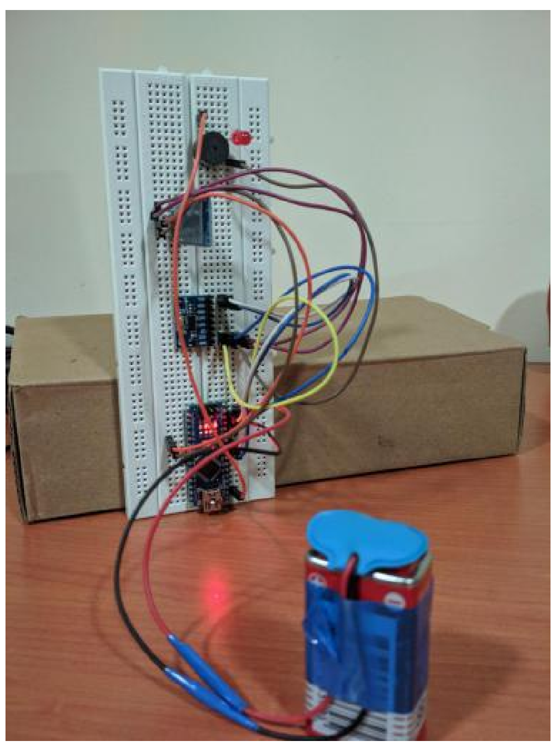


Figure 3.1 The accelerometer and Bluetooth module on the breadboard.

To begin the testing of the device, we first connect it to our mobile phones using the Bluetooth module present in it which we added. The entire device is powered by portable 10W batteries.

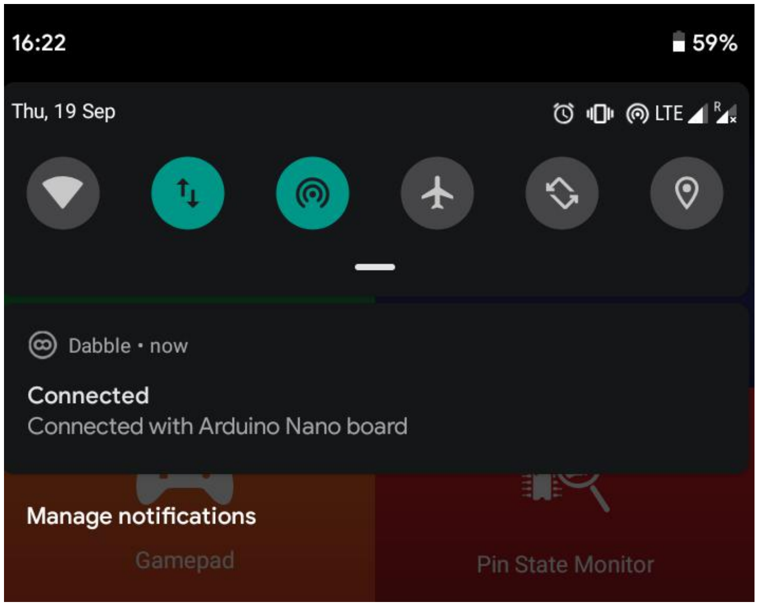


Figure 3.2 A connection between the Bluetooth module and the mobile phone has been established.

We then stick the entire breadboard onto the subject's back to get approximate data whenever the subject moves.

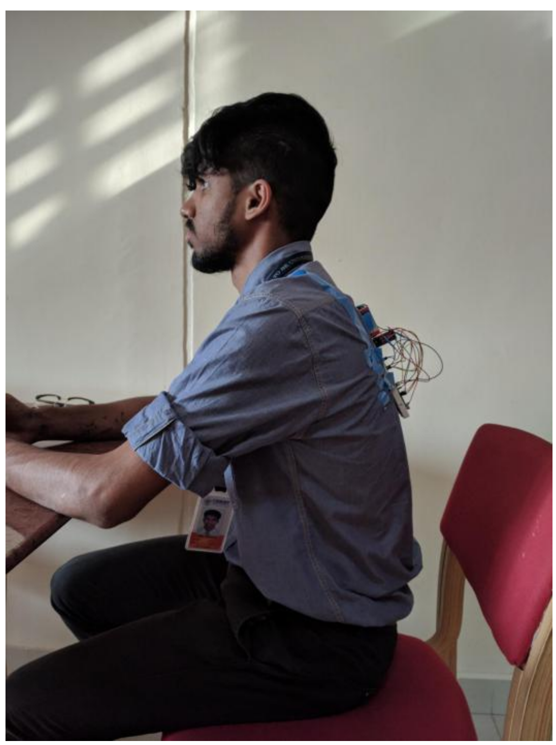


Figure 3.3 Breadboard on the back of the subject.

Live data is switched on all the time, so that every time the subject slouches or bends, the data is sent to us immediately and displayed in the app.

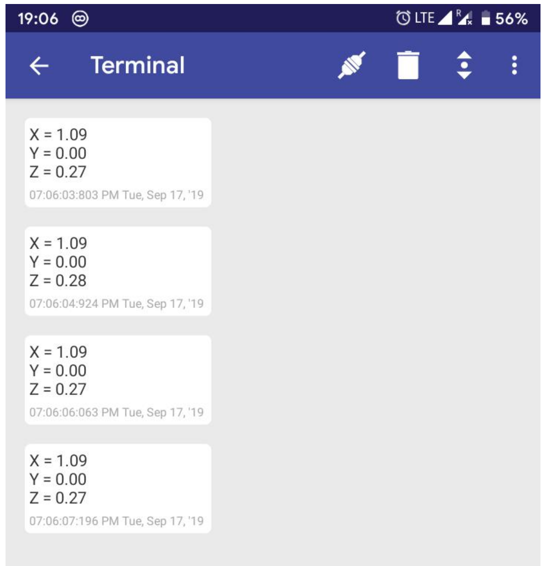


Figure 3.4 Dabble app on our phone is receiving data from the accelerometer every few milliseconds.

When the subject slouched to a certain degree which was bad for the subject's back, at which point the subject is alerted by the buzzer and a notification on the phone.

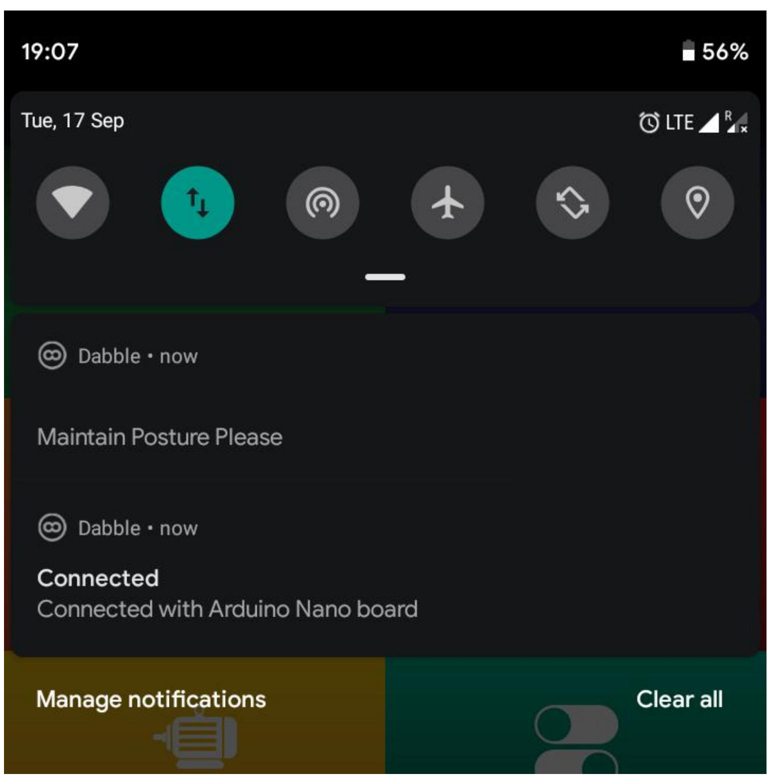


Figure 3.5 A notification is sent to your phone to alert the subject about his posture.

4) **Conclusion & Future Scope:**

Now, as the body ages, the human body posture changes inevitably and hence the values which are used by our device, Posture Senseii to notify the subject about their posture has to be adjusted according to the subject. Hence, in the future, several more testing phases need to be conducted on different ages, and gender groups for us to obtain a more reliable performance measure. And we can try and put the accelerometer on something comfortable so that the breadboard doesn't hurt the subject's back when they turn or try to move around and make sure it doesn't slip off and fall and get damaged.

5) Annexure I:

The code used to enable the accelerometer sensor to record accurate data and use Bluetooth module to connect to the mobile phone and send the information onto the Dabble app on the phone is displayed as follows:

Accelerometer code:

```
#include <Wire.h>  // Wire library - used for I2C communication


#define CUSTOM_SETTINGS

#define INCLUDE_TERMINAL_MODULE

#define INCLUDE_NOTIFICATION_MODULE

#define INCLUDE_MUSIC_MODULE

#include <Dabble.h>

String Serialdata = "";

bool dataflag = 0;

int ADXL345 = 0x53; // The ADXL345 sensor I2C address

float X_out, Y_out, Z_out;  // Outputs

int alert = 13;

#include <Wire.h>  // Wire library - used for I2C communication


void setup() {

  Serial.begin(9600); // Initiate serial communication for printing the results on the Serial monitor
```

```
  Dabble.begin(9600);
  Wire.begin();                 // Initiate the Wire library
                              // Set ADXL345 in measuring mode
  Wire.beginTransmission(ADXL345); // Start communicating with the device
  Wire.write(0x2D);             // Access talk to POWER_CTL Register - 0x2D
                              // Enable measurement
  Wire.write(8);                // (8dec -> 0000 1000 binary) Bit D3 High for measuring enable
  Wire.endTransmission();
```

Dabble App code:

```
//Dabble.waitForAppConnection();            //waiting for App to connect
  Notification.clear();
  Notification.setTitle("Posture Alert");
  pinMode(alert, OUTPUT);
}


void loop() {
  Dabble.processInput();
  // === Read acceleromter data === //
  Wire.beginTransmission(ADXL345);
  Wire.write(0x32); // Start with register 0x32 (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(ADXL345, 6, true); // Read 6 registers total, each axis value is stored in
2 registers


  X_out = ( Wire.read() | Wire.read() << 8); // X-axis value
  X_out = X_out / 256; //For a range of +-2g, we need to divide the raw values by 256,
according to the datasheet
```

```
  Y_out = ( Wire.read() | Wire.read() << 8); // Y-axis value
  Y_out = Y_out / 256;
  Z_out = ( Wire.read() | Wire.read() << 8); // Z-axis value
  Z_out = Z_out / 256;
  Serial.print("Xa= ");
  Serial.print(X_out);
  Serial.print("   Ya= ");
  Serial.print(Y_out);
  Serial.print("   Za= ");
  Serial.println(Z_out);
  Serialdata = ("X = " + String(X_out) + "\nY = " + String(Y_out) + "\nZ = " +
String(Z_out));
  Terminal.print(Serialdata);
   if (X_out <= 0.85) {
   Notification.notifyPhone(String("Maintain Posture Please "));
   digitalWrite(alert, HIGH);
   //Music.play("Po");
   } else {
   Notification.clear();
   digitalWrite(alert, LOW);
  }
  delay(1000);
}
```

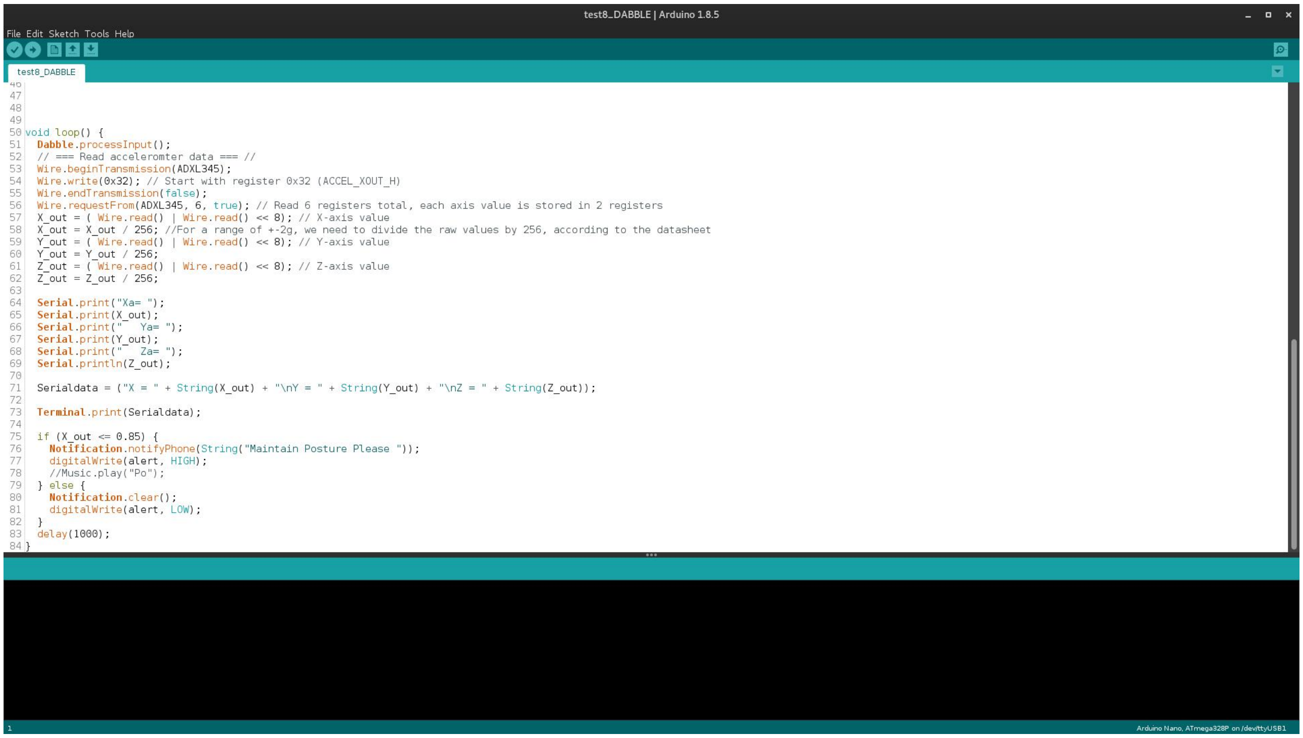


Fig 5.1 Screenshots of the Arduino IDE code.

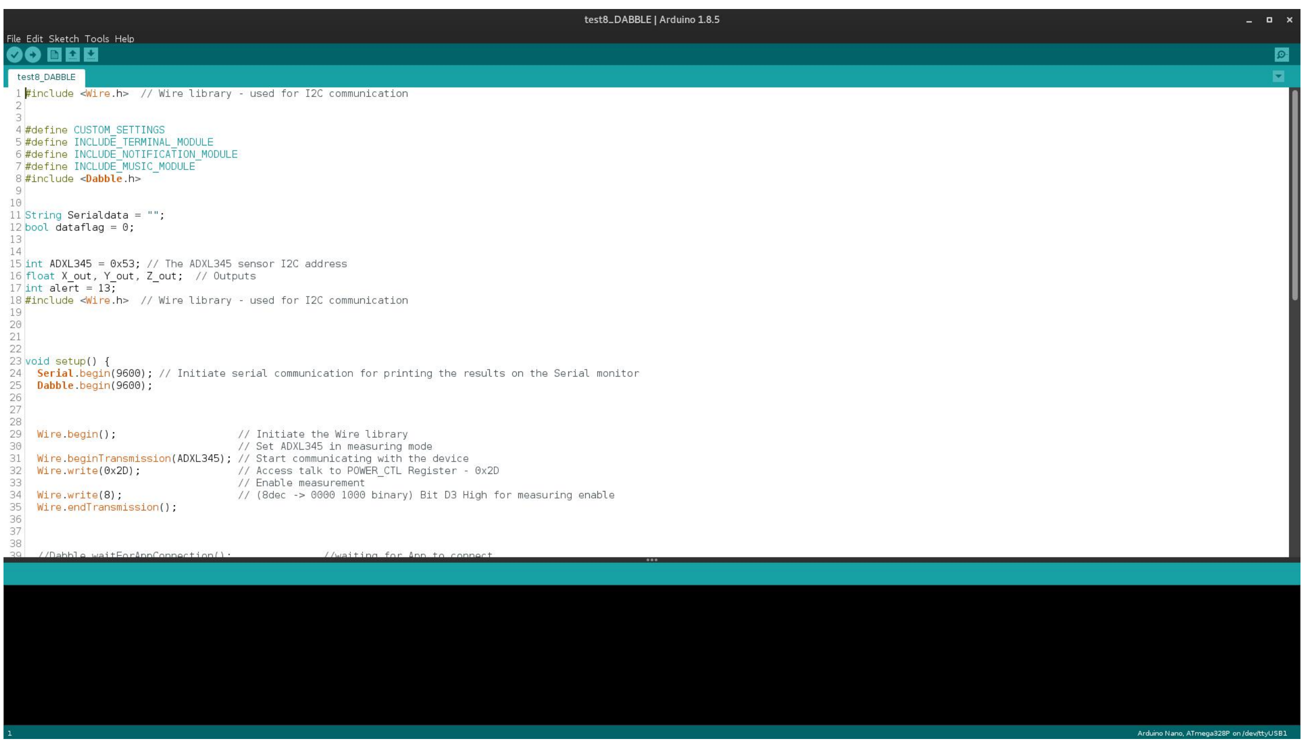
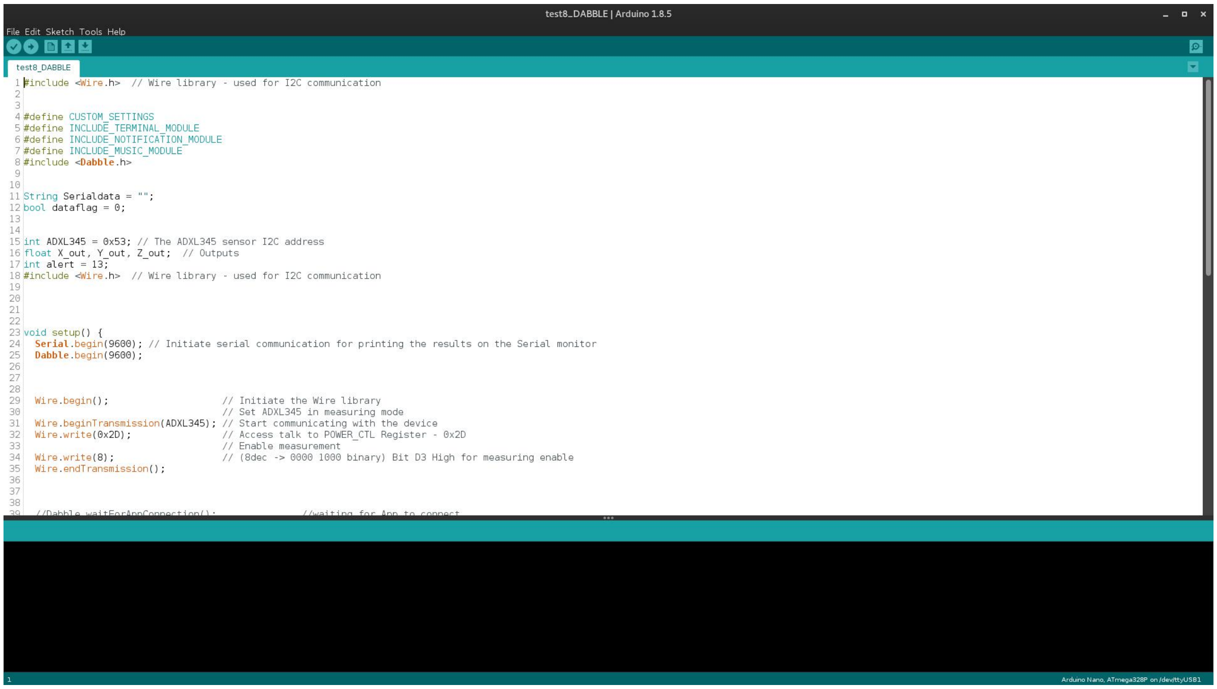


Fig 5.2 Screenshots of the Arduino IDE code.

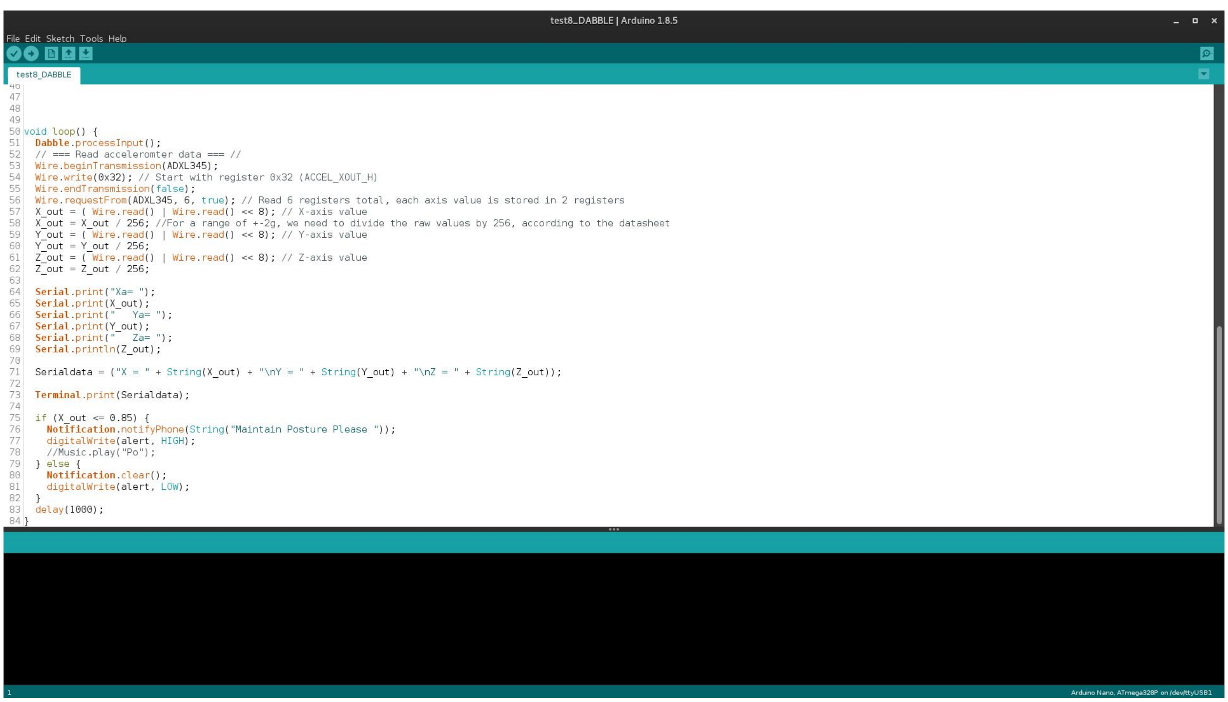
6) Annexure II:

These are the snapshots of our project in it’s entirety:




```
#include <Wire.h>  // Wire library - used for I2C communication


#define CUSTOM_SETTINGS
#define INCLUDE_TERMINAL_MODULE
#define INCLUDE_NOTIFICATION_MODULE
#define INCLUDE_MUSIC_MODULE
#include <Dabble.h>


String Serialdata = "";
bool dataflag = 0;


int ADXL345 = 0x53; // The ADXL345 sensor I2C address
float X_out, Y_out, Z_out;  // Outputs
int alert = 13;
#include <Wire.h>  // Wire library - used for I2C communication




void setup() {
  Serial.begin(9600); // Initiate serial communication for printing the results on the Serial monitor
  Dabble.begin(9600);



  Wire.begin();                    // Initiate the Wire library
                                   // Set ADXL345 in measuring mode
  Wire.beginTransmission(ADXL345); // Start communicating with the device
  Wire.write(0x2D);                // Access talk to POWER_CTL Register - 0x2D
                                   // Enable measurement
  Wire.write(8);                   // (8dec -> 0000 1000 binary) Bit D3 High for measuring enable
  Wire.endTransmission();
```

Fig 6.1 Screenshots of coding.




```
void loop() {
  Dabble.processInput();
  // === Read acceleromter data === //
  Wire.beginTransmission(ADXL345);
  Wire.write(0x32); // Start with register 0x32 (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(ADXL345, 6, true); // Read 6 registers total, each axis value is stored in 2 registers
  X_out = ( Wire.read() | Wire.read() << 8); // X-axis value
  X_out = X_out / 256; //For a range of +-2g, we need to divide the raw values by 256, according to the datasheet
  Y_out = ( Wire.read() | Wire.read() << 8); // Y-axis value
  Y_out = Y_out / 256;
  Z_out = ( Wire.read() | Wire.read() << 8); // Z-axis value
  Z_out = Z_out / 256;

  Serial.print("Xa= ");
  Serial.print(X_out);
  Serial.print("   Ya= ");
  Serial.print(Y_out);
  Serial.print("   Za= ");
  Serial.println(Z_out);

  Serialdata = ("X = " + String(X_out) + "\nY = " + String(Y_out) + "\nZ = " + String(Z_out));

  Terminal.print(Serialdata);

  if (X_out <= 0.85) {
    Notification.notifyPhone(String("Maintain Posture Please "));
    digitalWrite(alert, HIGH);
    //Music.play("Po");
  } else {
    Notification.clear();
    digitalWrite(alert, LOW);
  }
  delay(1000);
}
```

Fig 6.2 Screenshots of coding.

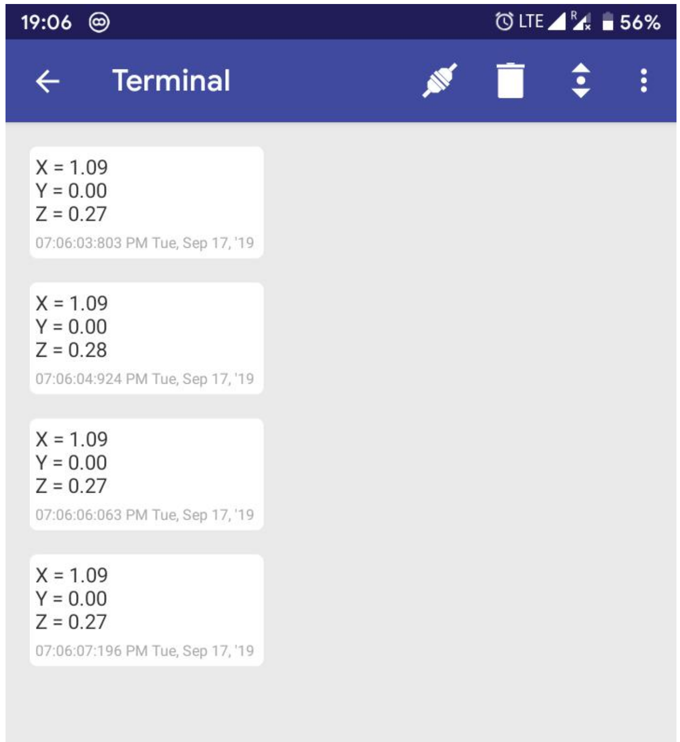


Fig 6.3 Screenshots of data sent from the accelerometer to the subject's mobile.

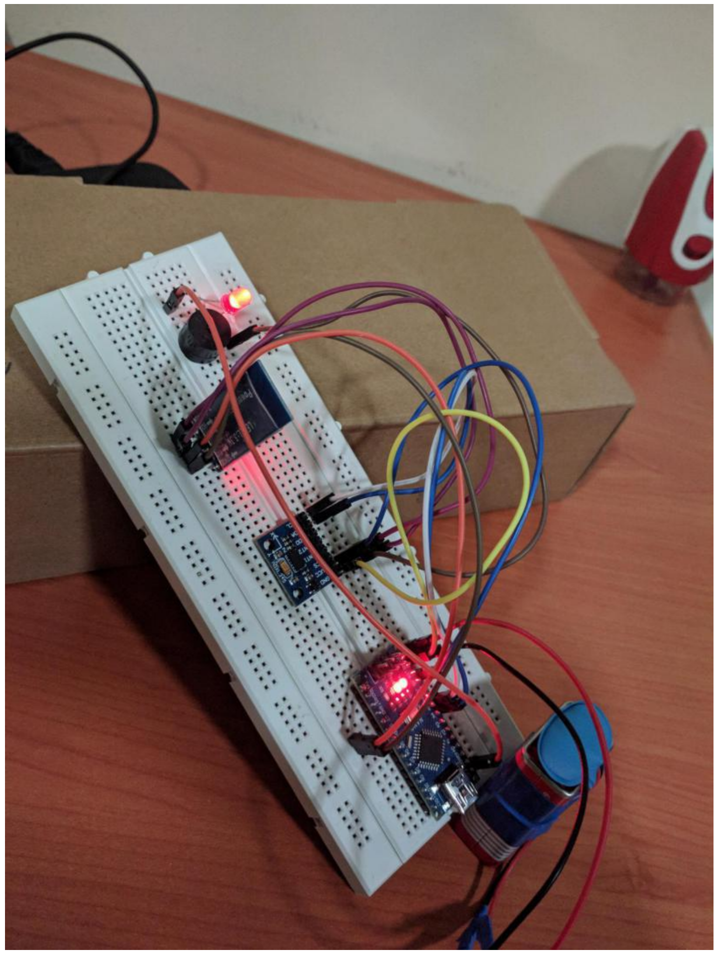


Fig 6.4 Snapshots of the back posture device.

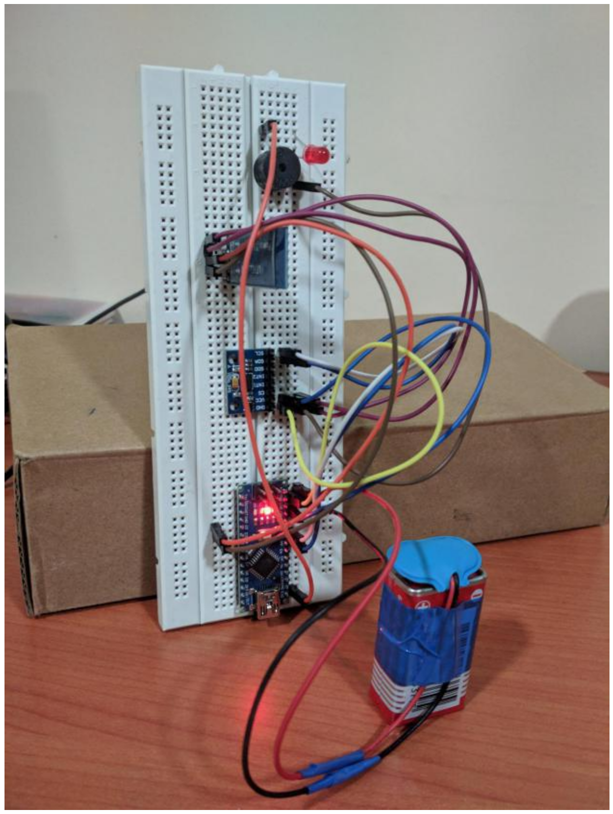


Fig 6.5 Snapshots of the back posture device.

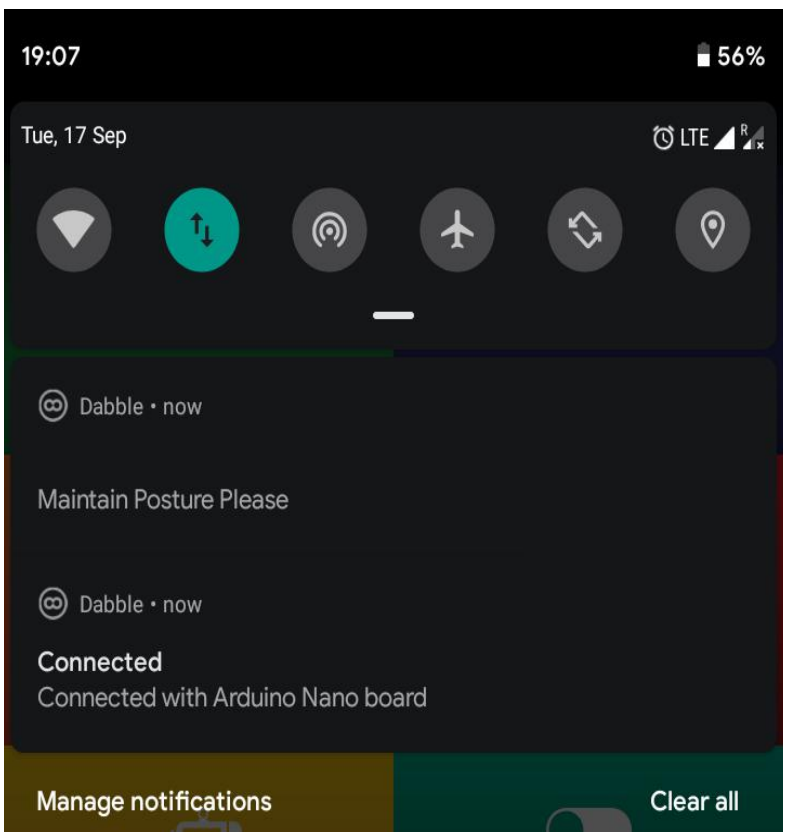


Fig 6.6 Snapshots of the back posture device.