SIT UP STRAIGHT-WEARABLE POSTURE CORRECTOR

Sanjana Kamath *sanjana@pdx.edu*
Deepika Punyamurtula *udeepika@pdx.edu*

Introduction

Stand. Sit. Walk. Read. We engage in these activities everyday, but how much do we know about ourselves and our bodies while we're in the midst of them? Are you standing up straight? Are you slouching at your desk while you are coding? Are you sitting incorrectly while watching TV?

Many people have a poor posture while they are sitting, and this is commonly found in the young and old. People with sedentary lifestyle often suffer from this problem and other related problems arising from this like back pain, fatigue and numerous doctor visits. Back problems and poor posture are correlated.

Most of the people have very little body awareness and no time to take posture correction classes/sessions by doctors. Thus, it is extremely important to have a good posture for a healthy mind and body. This is where our device comes into play.

Research Questions we wanted to address

1. How can one retain good posture?
2. How can we train a user to improve his posture gradually over time?
3. How can problems related to bad posture be reduced?
4. How can our device improve the overall lifestyle of a user?
5. What can we contribute to the wearable industry?

Humans nowadays spend most of their time sitting. This means that improving posture and encouraging more activity can have significant impact on peoples' health and well-being.

What we did to address the questions?

Wearable electronics has become very popular in the recent years and is continuing to advance more and more. An important use case of this technology would be in the field of health.

We therefore delved into the topic of a wearable for posture correction, by utilizing the Arduino microcontroller and other relevant sensors. When a bad posture is detected, the user is notified through sound output. His/her posture data is collected and can be viewed in our specially created Android application "*Sit up Straight*". Individuals who are afflicted with headaches, pain in the back, shoulder and neck can greatly benefit from this device. It aims at changing user's improper posture habits over time. It is designed to support back health and improve body awareness.

The motivation for this project came from the current popularity of wearable electronics, as well as the popularity of the Arduino microcontroller and the wide range of things that can be done with it. The devices available in market right now relating posture correction also are very less.

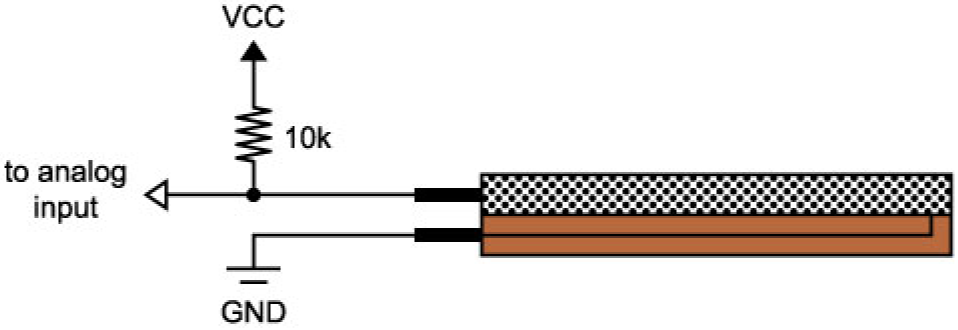
Description

Our design mainly employs the Arduino Uno and a flex sensor which helps detect the angle/degree of bend. The flex sensor should be in close proximity to the users body so that it can effectively sense a bend/flex in the users back.

How does the flex sensor work?

One side of the sensor is printed with a polymer ink that has conductive particles embedded in it. When the sensor is straight, the particles give the ink a resistance of about 30k Ohms. When the sensor is bent away from the ink, the conductive particles move further apart, increasing this resistance (to about 50k Ohms when the sensor is bent to 90º). When the sensor straightens out again, the resistance returns to the original value. By measuring the resistance, you can determine how much the sensor is being bent.
This flex sensor when connected to an arduino can help detect angles of bend.

The simplest way to incorporate this sensor is by using it in a voltage divider. Connect the flex sensor in the circuit described below:



The resistor and the flex sensor form a voltage divider, which divides VCC by a ratio determined by the two resistances. When the sensor is straight, the 10K resistor and the 30K flex sensor will cause the output voltage to be about 75 percent of VCC. When the sensor is bent, the voltage will increase to about 83 percent of VCC. If you're using 5V for VCC, you should see about 3.75V when the sensor is straight, and about 4.17V when the sensor is bent by 90º

How did we output audio/sound when incorrect posture is detected?

Once we had the setup of an Arduino Uno and a flex sensor in place, we decided to alert the user of an incorrect posture through audio output in the form of beeps. We used a Piezo buzzer element to achieve this. This buzzer makes a sound (a beep) when you apply voltage to it. Whenever the flex sensor detected an incorrect angle, we gave this buzzer a high voltage to produce a beeping sound. A Piezo is nothing but an electronic device that can both be used to output sounds. In our case we are plugging the Piezo on the pin number 9, that supports the functionality of writing a PWM signal to it.

How did we decide to log the data coming from the sensor?

Calibration of the flex sensor
Essentially, the voltage values we were getting from the arduino when is sensor is straight was 512 and when max bent was 614. Since such a range isn't the best, we used the *map()* function of arduino library to convert it to a large range like 0 to 90. So we got sensor readings from 0 to 90 which indicated angle of bend.

How did we include an RTC (real time clock) into the Uno?
We needed to log time and date information also since it was essential to know when and what time the incorrect posture was detected. Arduino Uno doesn't include an inbuilt RTC module. Time info needs to come from external sources when using the UNO.
To address this issue, we used battery-packed RTC clock. This is a separate chip which helps in timelogging. We had to make sure we buy a battery powered RTC since the clock had to be running at all times. The RTC we used is the DS1307. We used the DS1307 and time libraries to help log the time and date. The pins of the RTC we used are:

- **5V** is used to power to the RTC chip when you want to query it for the time. If there is no 5V signal, the chip goes to sleep using the coin cell for backup.
- **GND** is common ground and is required
- **SCL** is the I2C clock pin - it's required to talk to the RTC
- **SDA** is the I2C data pin - it's required to talk to the RTC

Handle data
We had large amounts of data coming from the sensor which we had to analyze and plot. We decided to use Arduino Uno SD card shield to help us log the data from the sensors. Essentially, we are collecting all the data into a micro SD card sitting inside the Arduino Uno shield. Once all the data is collected, the micro SD card can be plugged back into your Android phone and our Android app will then plot out all the relevant data to the users. The communication between the microcontroller and the SD card uses SPI (Serial Peripheral Interface library) , which takes place on digital pins 11, 12, and 13 of the Arduino.

How did we decide to represent this data to the user?
We decided to collect all the sensor info into the micro SD card in CSV file format. Once all the files are collected in the CSV files, we processed the data. We collected the following information:

1. Time
2. Date

3. Angle of bend
4. Did the buzzer beep?

   Based on the calibration of our flex sensor, we kept a bend of:
   - < 30 degrees detected by flex sensor as Normal
   - Between 30 and 40 as Mild
   - Between 40 and 50 as Severe
   - > 50 as Very Severe
5. Severity of back bend - How severe was the bad posture? [Mild, Severe, Very Severe]

Sample data:

Time(hh:mm:ss)	Date	Current Angle of the back (degrees)	Is_Beeped	Intensity
13:12:16	27/11/2013	12	0	Good
13:12:17	27/11/2013	31	1	Mild
13:12:18	27/11/2013	41	1	Severe
13:12:19	27/11/2013	39	1	Mild
13:12:20	27/11/2013	12	0	Good
13:12:21	27/11/2013	12	0	Good
13:12:22	27/11/2013	12	0	Good

For this, we developed an Android application called “Sit up Straight”. This application automatically reads the CSV file present in the phones micro SD card and loads up all the data into the SQLite database by creating a suitable table.

Once the data is loaded, we perform SQL queries to plot the data using Android’s AChartEngine graph library.

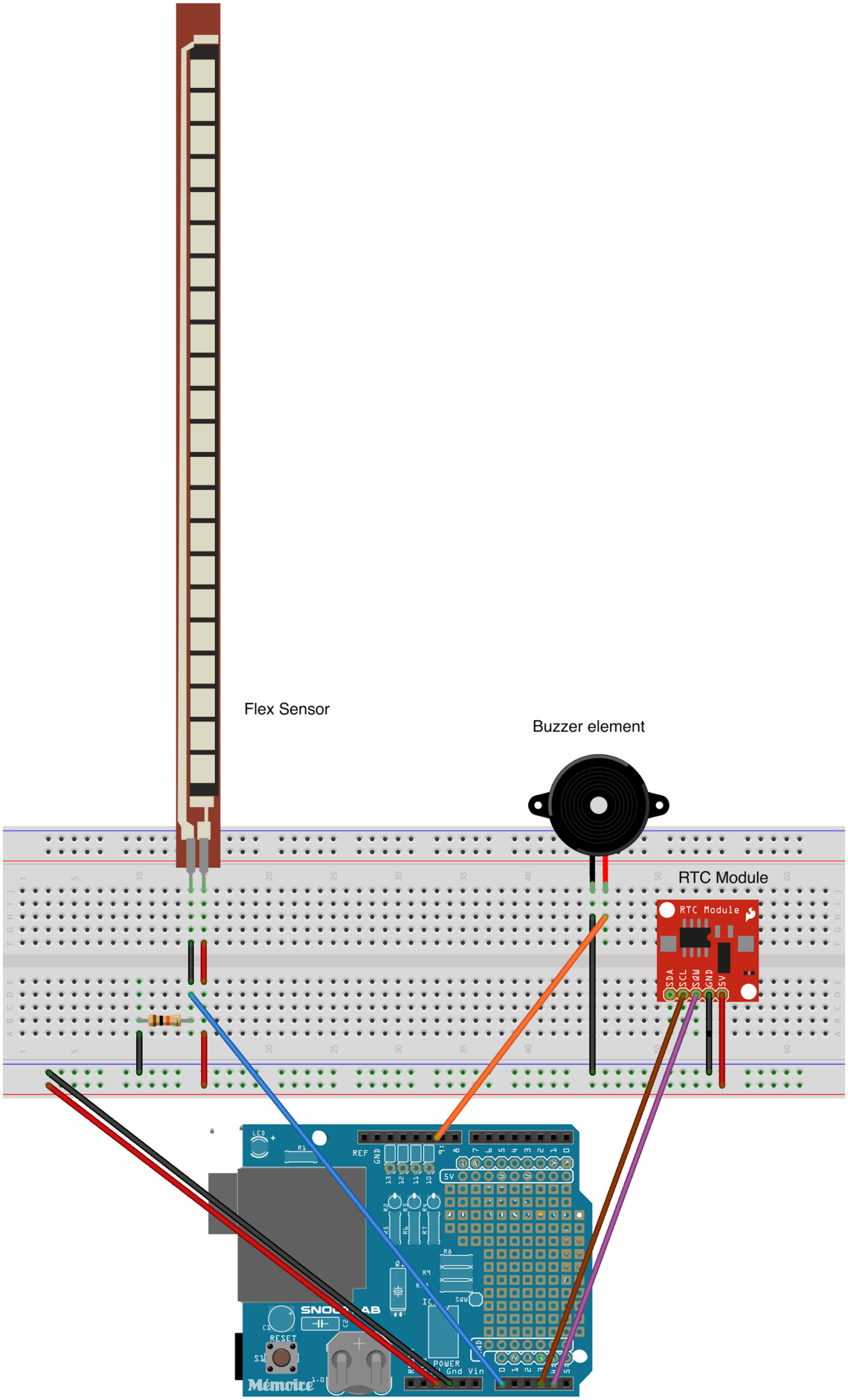
Arduino Schematic

Parts used:

1. Arduino UNO Rev 3
2. Flex Sensor 5’
3. Arduino UNO SD card shield
4. DS1307 RTC module
5. Peizo buzzer element
6. 10k ohms resistor
7. Jumper wires

8. Breadboard
9. Male to female Jumper wires
10. 9V Battery
11. Micro SD card

Please find the Arduino schematic in the next page.



Flex Sensor

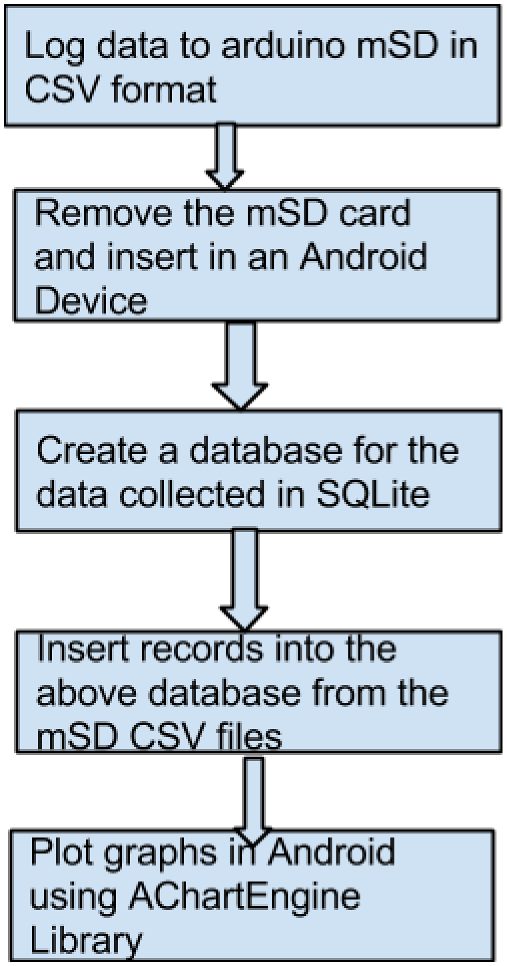
Buzzer element

RTC Module

MicroSD Card Shield for UNO
Arduini Uno Rev3 sitting below the SD card shield

Made with Fritzing.org

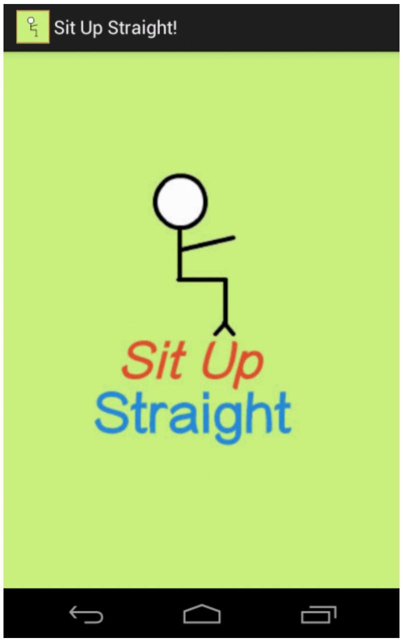
Work Flow


Log data to arduino mSD in CSV format
Remove the mSD card and insert in an Android Device
Create a database for the data collected in SQLite
Insert records into the above database from the mSD CSV files
Plot graphs in Android using AChartEngine Library

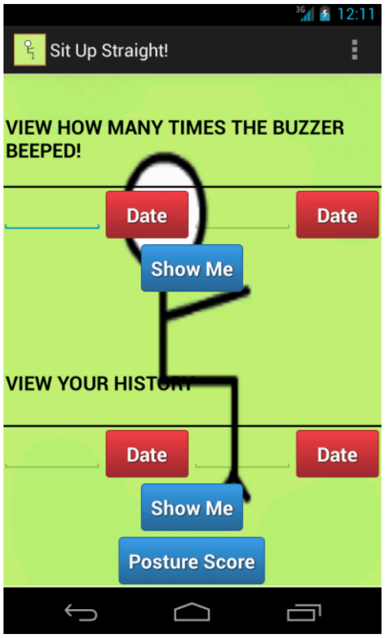
Android App

We have created an Android app called "*Sit Up Straight*" to help the user visualize the data collected from the Arduino setup.

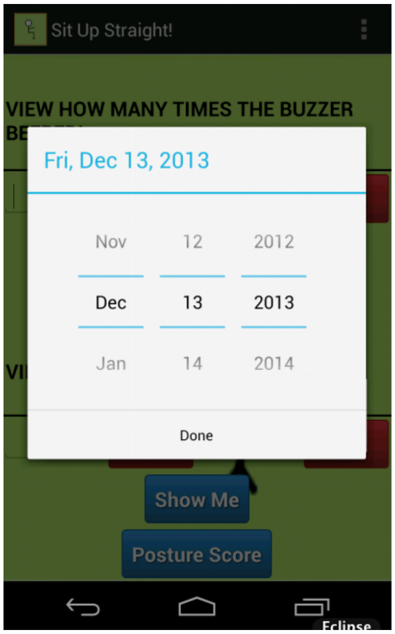
Screenshots of the UI



[Fig 1: Splash Screen]



[Fig 2: View Data Screen]



[Fig 3: Pick Date]

More about Fig 1: What is a splash screen? Why did we use it?

The first screen of the app is a Splash screen indicating the application is loading. It is also used to perform some preprocessing before the app loads.

The logo of our device is displayed in the splash screen while the database is initialized fully with sensor data from the CSV files stored in the microSD card of the phone.

Experiments carried out

The following were the things we focussed on while conducting the data collection.

1. How many times did the buzzer beep during the time when he was wearing the belt?
   We wanted to give the users the ability to view how the beeps of the buzzer changed (change could be an increase, decrease or no change at all) over time because of using the device regularly. It was to give the user the ability to view his progress in posture correction.
   Whenever the buzzer beeped, we logged that data into the mSD card.
   The log collected contains a Boolean field called 'Is_Beeped' which is set to true when the buzzer buzzes i.e. when the posture is bad. By summing over this value, the beep count for a day can be obtained.

2. View changes in the beep count over many days to find how the user's posture is being affected by the device?
   We tried to find out the variation in the beep count over days by plotting a bar graph of beep count over days. This helps us to determine how the user's posture is being

affected by the device.

3. How did the reaction time of the user improve once he started using the device?
   We were trying to find out how quickly the user was able to adjust his posture once he gets a buzzer beep. We observed that the user was becoming more reactive to the beeps with time. He becomes more and more aware of the device around his waist and begins to correct his posture faster when the buzzer beeped.

4. View the various positions of his back (angles of the back) while user was wearing the device.
   We kept a record of what major activity the user was doing as well as the position (bacj angle) he was in while we were collecting the data for that day. This gives the user a perspective of how his back positions were throughout the day. We can see how many times the user went above the "good posture angle (< 30 degrees flex)".

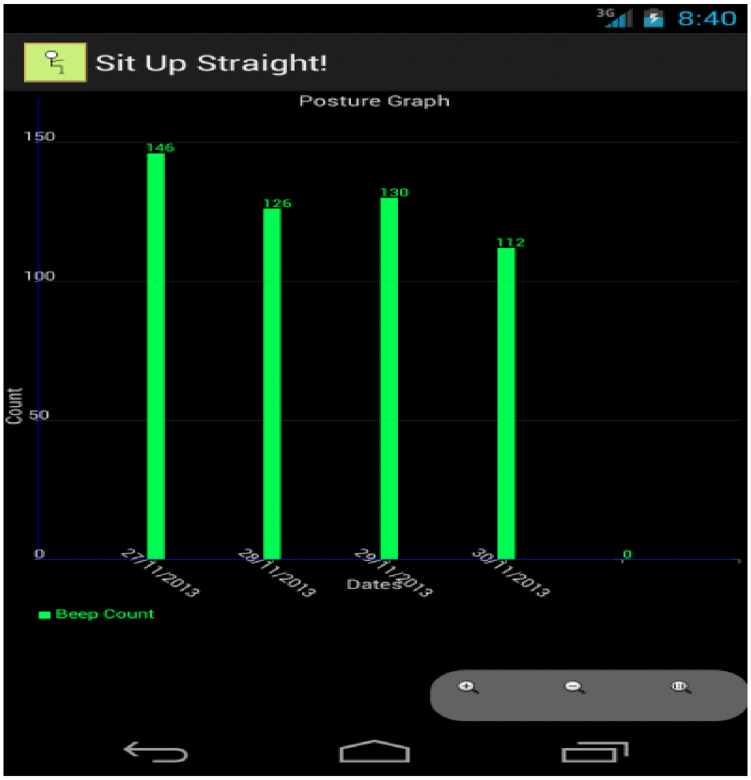
5. How the users back angles varied when there is no beeper attached to him versus improvements when he had an annoying beeper telling him to sit straight?
   We wanted to do this to show how the user corrects his posture as soon something tells him to do so versus when there is no monitoring. We removed the buzzer from the circuit to simulate this scenario. The results were very obvious, the users rectified his posture when the buzzer beeped. But when there was no buzzer, he continued to be in that position for a long time till he decided to change. When you have someone actively monitoring you, you tend to rectify something incorrect being done. We noticed the same thing.

6. Take a log of what activity the user was doing when he wearing the belt.
   We asked the user to log what activity he was doing during the time period when he was wearing the belt. An example of this log data has been pasted in Page 10. This helps analyze what activities make the user slouch. And compare how different activities affect the posture.

   We anlayzed this using the graphs we obtained. Few sample graphs are:

Visualization of Data

1. **Bar Graph**
   Used to represent the number of beep counts obtained each day



[Fig 4: Bar Graph 1]

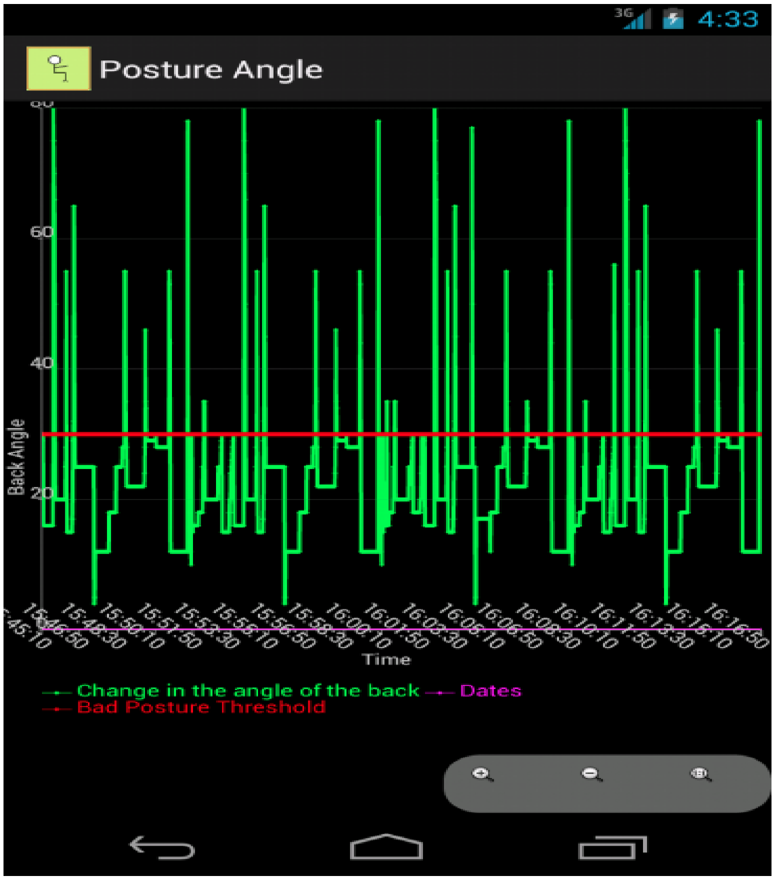
Major Activity log for the above days:

Date	**Major Activity**
27/11/2013	Watching TV on a couch
28/11/2013	Reading a book
29/11/2013	Using a Laptop
30/11/2013	Reading

2. Line Graph

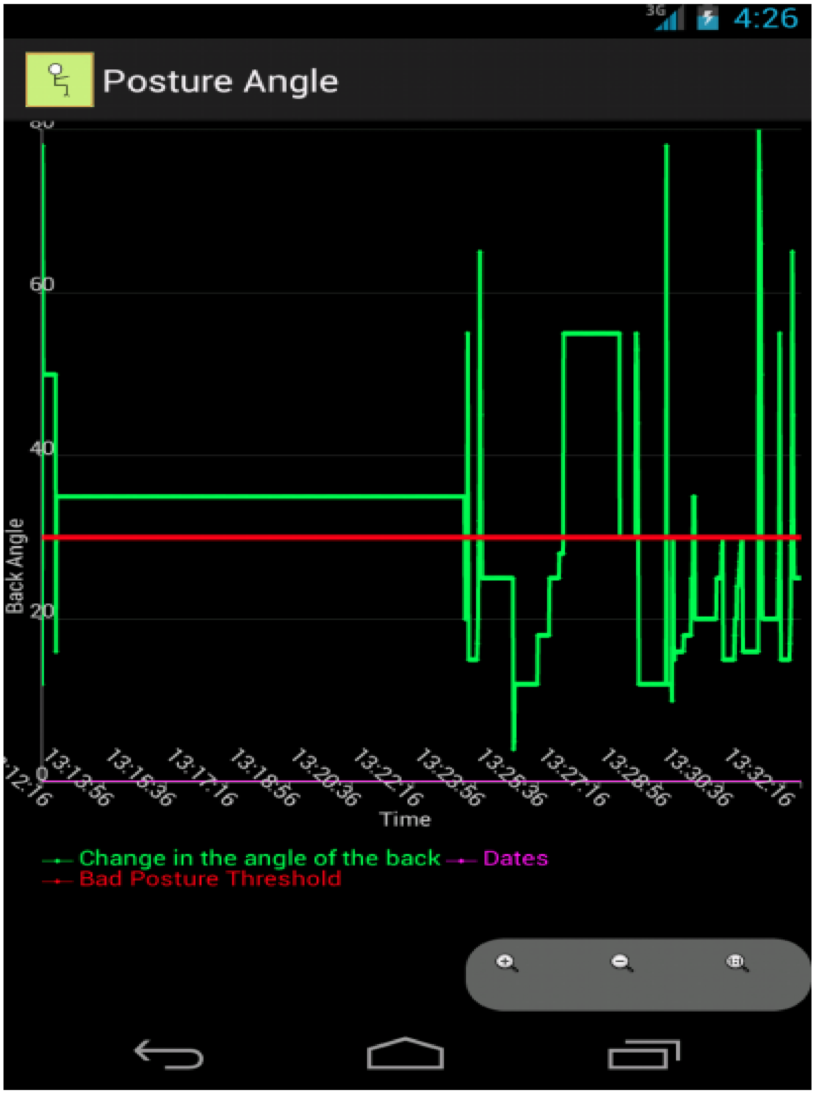
Plots the angle of the flex sensor over time as a continuous line graph. This graph also indicates a threshold value which is the ideal angle for the user’s posture (<30 degrees).

The below line graph indicates the sensor readings (angles of the back) **when the user is wearing the belt with the buzzer.**



[Fig 5: Line graph - Sensor reading with Buzzer]

The below line graph indicates the sensor readings (angles of the back) **when the user is wearing the belt without the buzzer.**

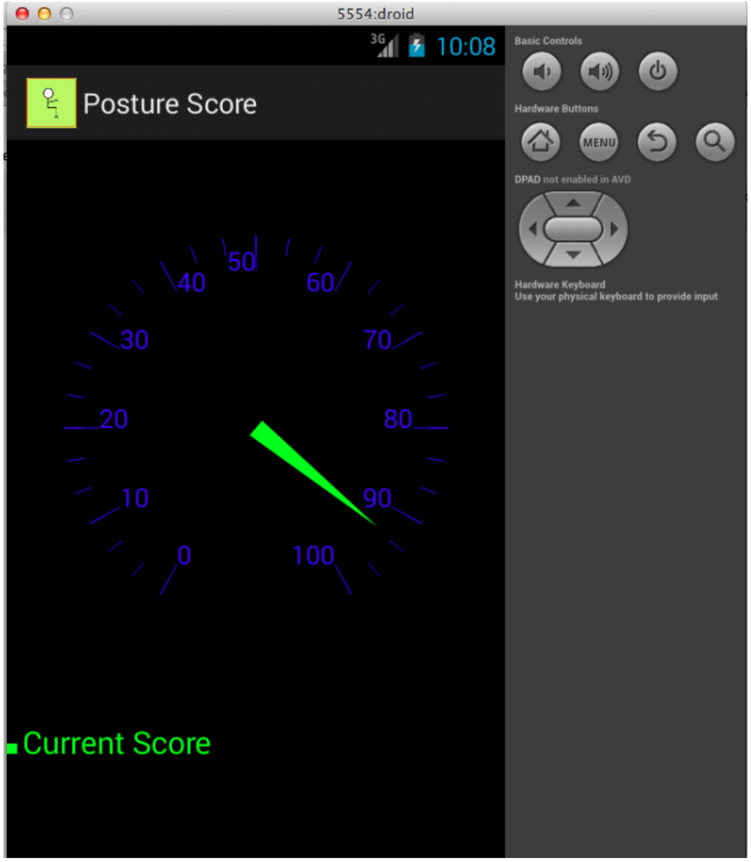


[Fig 5: Line graph - Sensor reading with no Buzzer]

3. <u>Posture Score</u>

The user is given a posture score for a given date which is obtained by calculating the fraction of the number of beeps over the total number of readings obtained for the day. This is represented in the form of a dial. The score ranges from 0 to 100.

A score of of 50 or more is considered good and is indicated in green while any score less than 50 is shown in red.



[Fig 6: Posture Score]

Results and Observations from the graphs

1. The total number of beeps showed a noticeable decrease indicating that the user’s posture was improving gradually. This inference was obtained from the bar graph plotted for the data. For e.g. in the Fig 4 we can see that the number of beeps for 30/11/2013 is lesser than the number of beeps for the days before it.
2. The reaction time or the time in which the user adjusts his posture back within the threshold value reduced over the days. This was inferred from the time in which the readings drop from severe to good.
3. The posture score is a motivation to the user to improve his posture more. The user is driven to get a better score and this is shown the beep count bar graph. The beeps of the buzzer reduced over time.
4. There is a stark difference in the back angle values when the buzzer monitors the user to change his posture to when there is no buzzer attached to the circuit. The user quickly changes his posture when the buzzer beeps

What is the impact of the results?

Looking at his/her own posture data and score the user can get a real time feedback about his posture. The Android App facilitates the understanding of the data collected by the user.

How does it relate to quantified life?

1. We mainly quantified human posture which helped us derive useful information to be presented by the user to improve his well being.
2. It involves quantifying oneself and collecting data about oneself which helps make informed decisions about one's daily behavior.
3. We collected data and tracking the users posture encouraging people to make small changes in their behavior leading to overall improvement of health.
4. We showed how the user's posture has changed over time due to using the device

What did we learn?

1. Quantification when integrated with devices can show better results in achieving the desired goals.
2. Quantifying aspects of our own life helps us to learn more about ourselves and finding out areas of personal improvements.
3. Proper real time feedback from machines and personal devices can greatly help in improving one's habits and lifestyle. The real time buzzer beep, the visual feedback through the android app are examples of such feedback.

Future Enhancements

1. Include different posture thresholds based on the activity the user is doing while wearing the device.
2. Provide profile centric modes which enable the user to mute the device but still collect data for future viewing. For example, when the user is in a meeting he would like it to be muted, but still can collect data.

Conclusion

Through this posture correction device we tried to bring together both wearable technology and quantification to help improve our posture. The data collected provides evidence that indeed the user's posture should positive difference.

References

[1] AchartEngine API *http://code.google.com/p/achartengine/*
[2] Flex Sensor and Arduino
http://www.adafruit.com/blog/2012/06/26/bad-posture-adafruit-waveshield-bend-sensor-arduino/
[3] Posture Corrector *http://www.sundh.com/blog/bad-posture/*