Final report

Project: Posture detector using the TI SensorTag CC2650

Student: Miguel Antonio Chavez Tapia

Email: mac225@cam.ac.uk

Module: 4B25 - Embedded Systems for the Internet of Things

Module leader: Dr. Phillip Stanley-Marbell

Introduction

The following report covers the approach of developing a human posture detector using the TI SensorTag CC2650. Its utility is to solve the problem of an incorrect human posture by detecting it using a sensor and signalling to the user if an incorrect posture is detected. A correct posture prevents health issues and reduces stress.

The target users of such a device is regular people, as posture affects your health, social presentation and self-confidence. This interest rises in people who perform regular work in a specific position, as maintaining a correct posture during hours is more difficult, thus affecting health and stress, and consequently also performance. For people who works as presenters, for instance lecturers or speakers, posture is an important asset of the image they project to the audience.

State of the art

Current market for wearables that monitors posture includes some brands like Lumo Lift®, Prana, Upright Go™, Mevics, among others. These devices basically use accelerometers and gyroscopes to compute different readings, and one of them is posture. They work along with a smartphone app which gives a more interactive output to the user.

As a summary of the features of each device, a table is presented below. All these products use MEMS sensors and BLE to connect to smartphones.

Product	***Sensors***	***How to wear***	***Smartphone application***	***Features***
Lumo Lift	*Accelerometer, gyroscope, magnetometer*	*Magnetically holds to clothes*	*Yes*	*Posture tracking, steps taken, distance travelled, calories burned*
Prana	*Accelerometer*	*Clips to waistband*	*Yes*	*Posture tracking, breathe tracking*
Upright Go	*Accelerometer, stretch sensor*	*Adheres to skin in the back*	*Yes*	*Posture tracking*
Mevics	*Accelerometer, gyroscope, compass*	*Magnetically holds to clothes*	*Yes*	*Posture tracking, steps taken, distance travelled, calories burned*

Table 1. Current market products for posture detection. Source: see references.

Approach used on the project

In order to proceed with an estimation of the posture using the TI SensorTag, the start point was to analyse the sensors available. The movement sensor is the MPU9250 from InvenSense, which consists of a 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer. The solution is based on readings of this movement sensor. As a first approach, a reading of the gyroscope data was taken from the sensor registers. The readings were noisy and with a noticeable drift, specially in the Z axis, when the device is sitting still. As the usual readings of the angle of a human posture do not change quickly, another approach was considered.

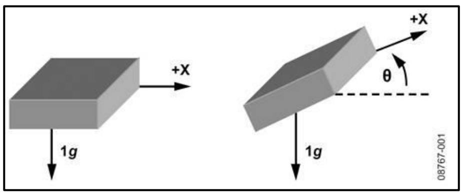

Figure 1. Effect of gravity on accelerometer and relation with angle of inclination. Source: Digikey

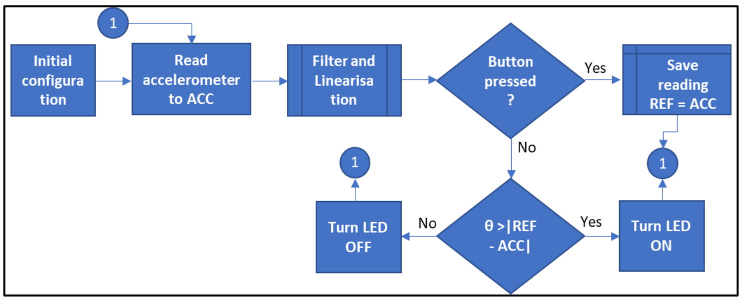

Figure 2. Flowchart of the main programme that runs on the TI SensorTag CC2650.

The readings of the accelerometer have a steadier state when the TI SensorTag stays still. The relying principle is the reading of the gravity's acceleration effect on the device. However, this effect is not linear but has rather a sinusoidal response. A safe approximation of the value of the angle in radians can be computed by using the relation $\sin\theta \cong \theta$, for small values of θ. Another factor that needs to be considered is that this method reads the absolute angle of the device. A relation between angle and accelerometer valuers is shown in figure 1.

The programming of the device needs to consider these points. The design of the program can be sum up in the steps depicted in figure 2. To address the considerations presented there are two main steps to compute accurate angle positions. First, as the relation between angle and accelerometer reading is not linear, the approach taken is to correct the readings by linearising them. The linearisation applied is explained in the annex section, but the main idea is to divide the sine in sections an apply a linear correction. Second, as the accelerometer reads the absolute angle of the device, there is a need to set a reference angle as one of the first steps. Therefore, the measure that is compared is the relative angle between the reference and the current accelerometer value. As the reference angle is unknown, it is not safe to assume that is very small, thus the need for linearisation.

Results

The system is implemented using the TI SensorTag CC2560 addressing 02 topics: improve battery lifetime and reduce complex calculations within the device. The starting point of the implementation is the TI SensorTag code, provided by Texas Instruments, for fully functionality of the SensorTag. To achieve the first goal, all the sensors of the device are not activated in the first initial configuration, except for the movement sensor MPU9250. Even for the movement sensor, only the accelerometer is configured as active. For the second goal, the process of linearisation is defined to work only with simple arithmetic operations and comparations to compensate for the non-linear response of the device. Similarly, the filter calculation is set to be an average filter. The details of the linearisation and the filter are explained in annex section.

The system implementation is depicted in figure 3, and the system usage in figure 4. For the system implementation the parameter set up for maximum angular difference between the reference and the current position angle is 15°. The basic step is to calculate the difference between the reference angle and the actual angle, after the readings were filtered and linearised. Details of the device's parameters are presented in the annex.

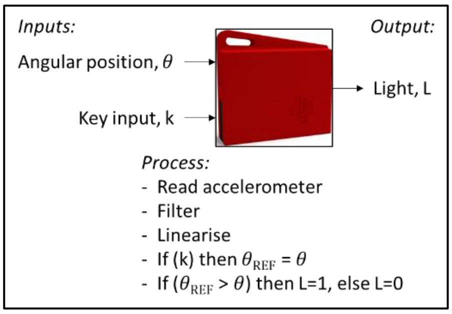

Figure 3. System overview.

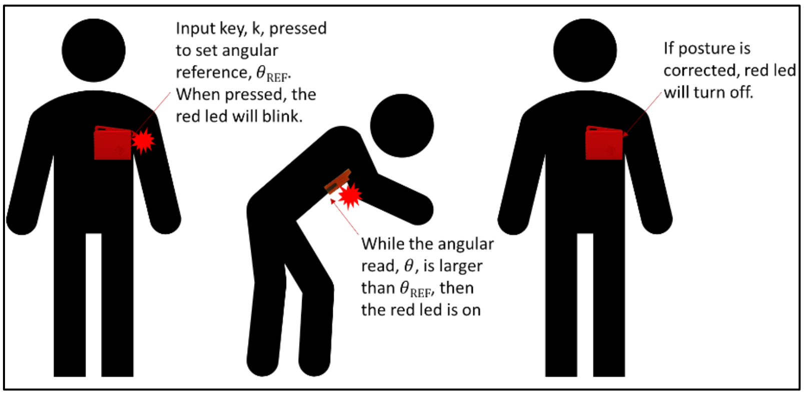

Figure 4. Setup device for usage.

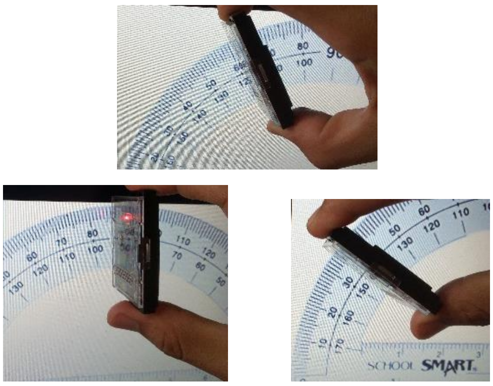

Figure 5. Measurement of inclination angle with reference 120°.

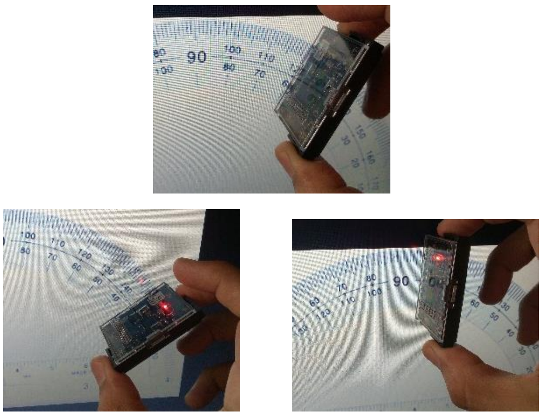

Figure 6. Measurement of inclination angle with reference -120°.

The implementation of the system's output is presented in figures 5 and 6. The references angles are set to 120° and 120° respectively. From the results it can be checked that the system works for both references angles. There is also an app developed using Evothings to read the accelerometer data and calculate more accurately, using trigonometric formulas, the angle difference and signal using a smartphone screen colour change.

Annex

Linearisation

The relation between the angle of inclination and the lecture of the accelerometer corresponds to a sine response, if we read the axis that is perpendicular to the magnetic field of the Earth. To compute a real value of the angle we would need to compute the $\sin^{-1}(\frac{accelerometer\_axis}{1G})$. Since this computation is not easily implemented on a microprocessor device (an approach seen in other devices include the use of lookup tables), the implementation for this project is to linearize the sine response. For small angles the approximation $\sin\theta \cong \theta$ is valid, but as the angle increases this relation is no longer valid. If we plot the response of a sine vs. the angle in radians, and the error of using this approximation for angles between -90° and 90°, we can see that the response gives an error of 5° for angles greater than 40°. Figure 7 and figure 8 shows the error of using the sine approximation.

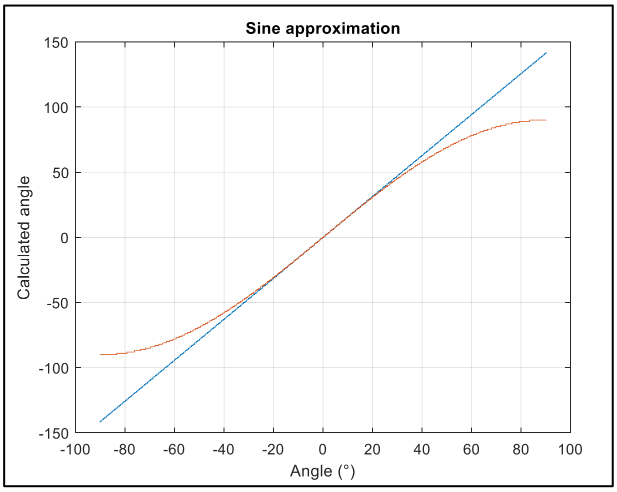

Figure 7. Sine response vs. linear response.

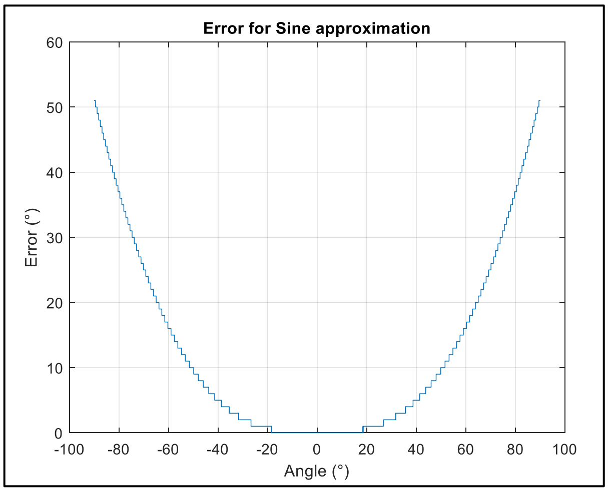

Figure 8. Error in angle between sinusoidal response and linear response.

In order to improve the response without the need to implement an arcsine, the approach is to linearise using only arithmetical operations. To achieve this goal, this simple equation for the angle is set by applying just additions and divisions in ranges for the readings of the accelerometer:

$$\alpha = \sin\alpha\,,$$

$$if\ \alpha > 30, then\ \alpha = \alpha + \frac{\alpha}{16}$$

$$if\ \alpha > 50, then\ \alpha = \alpha + \frac{\alpha}{32}$$

$$if\ \alpha > 70, then\ \alpha = \alpha + \frac{\alpha}{32} + \frac{\alpha}{64}$$

These calculations were found by doing approximations and trying to fit the divisions to power of 2 numbers, so the operations can be implemented by bitwise operations. Simulations were done using this linearisation and then plotted the function and its error. By implementing the linearisation the range of error less than 5° can be extended to 70°. The results are presented in figure 9 and 10.

Filter

The accelerometer sensor presents inherently noise. To overcome it, a filter was implemented. The filter is an average filter, which takes 8 readings of the accelerometer and average them. The factor 8 was selected in purpose for an easy implementation using bitwise operations. As the accelerometer reads 8 values, the accelerometer was configured to get around one reading after filtering each 0.96 second.

Accelerometer configuration

The accelerometer settings correspond for reading a maximum value of 2G on each axis and a sensing period of 120 milliseconds. As the filter takes 8 accelerometer readings to average them, it gives around 1 filter output per second. And as the sensor will not go for more than 1G for a static person, 2G is sufficient to cover the regular readings of environment.

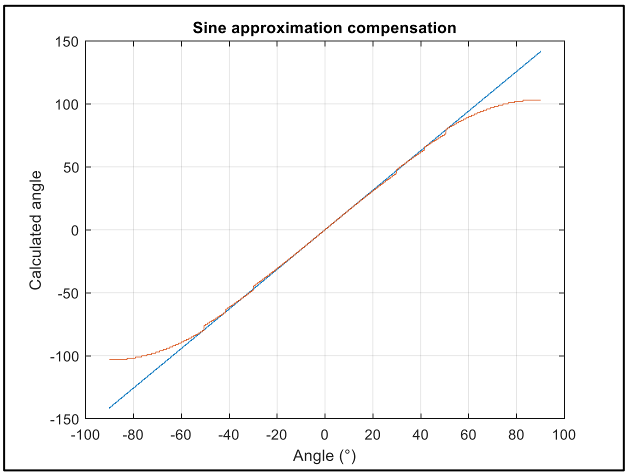

Figure 9. Linearisation function

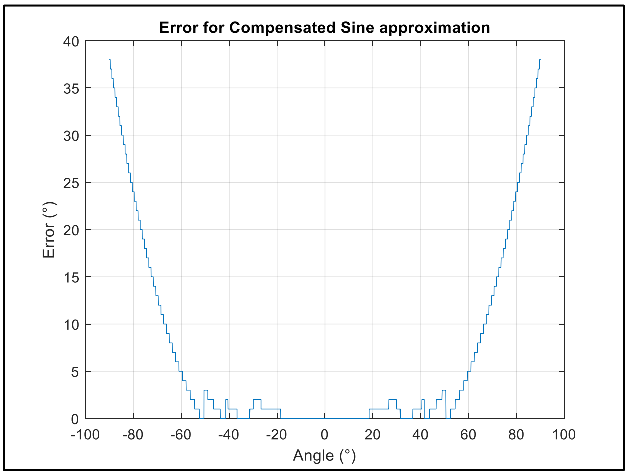

Figure 10. Error of linearisation

Taking the 2G range of the accelerometer and the fact that the response for expected for the angle from -90° and 90°, this represents an accelerometer reading of -1G to 1G, which is half of the full reading range of 16 bits, which is 15 bits. Therefore, the expected responsivity of the sensor would be $R = \frac{180°}{2^{15}}$. This gives $R \approx 0.0055°/LSB$. But the range in which the response is linear is when α is very small. If we say $\alpha = 30°$, then:

$$\alpha = \sin\alpha \times \left(\frac{2^{15}}{2g}\right) \approx 8170$$

As $\alpha = 30°$, the resolution can be approximated to $272\ bit/°$.

Evothings app

Using Evothings studio, an app was developed to compute the angle using trigonometric functions, 02 axis and a visual output when the calculated output exceeds a value. A smartphone with the Evothings app can use it to view the results while the SensorTag is advertising. Additionally, the app calculates the output in the same way that the SensorTag, thus can be used as a reference to check its accuracy.

References

Lumo - https://www.lumobodytech.com

Mevics - https://mevics.com

Trusted Reviews - http://www.trustedreviews.com/reviews/lumo-lift

Wearable - https://www.wareable.com/

Posture corrector reviews - http://www.posturesorted.com/