Engineering Tripos Part IIB
4B25 : Coursework 6

Final Report : SleepFit

Jason Y J TOO

Thursday 17th January, 2019

1 Introduction

Today, there is an increasing trend of people migrating towards urban areas. The result is an ultra dense city where people are surrounded by many sources of light, noise and air pollution. The problem that the SleepFit aims to solve is to provide users with a quantitative insight into the environments they live in so that the appropriate actions can be taken. This is so that the necessary adjustments can be made to improve their quality of living. For example, using the light sensor, it would alert users when the room is to bright for sleep, or too dim for work. Using the other sensors, a similar suggestion could be made to ensure that the room is at it's optimum condition for life activities.

The *SensorTag 2650* has many on-board sensors (10 in total) whilst being in an compact and portable package as shown in *Figure 1*. Plus, as it possible to connect multiple sensors to *Raspberry Pi*, a number of SensorTags can be used all around a room/home to get precise measurements. With the rise of virtual home assistants like Amazon Alexa & Google Assistant, the SleepFit system could interact with existing smart home features and control lights, thermostats and even some cases, smart window, in ensuring optimum conditions at all times for a given activity.

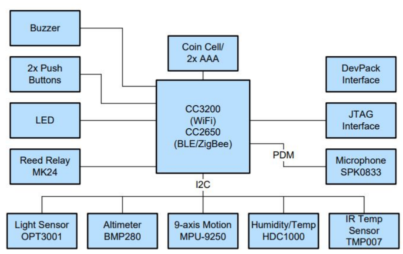


Figure 1: Block Diagram for SensorTag 2650

2 State of the Art

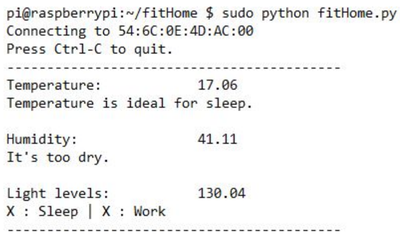
Indeed, the concept of have feedback in controlling an environment is not new. Thermostats have been doing the task of temperature regulation in past, and today we have devices such as the Nest which attempts to take it a step further. On top of just keeping temperature at home comfortable, the Nest Thermostat also has the objective of saving energy bill by learning individual user preferences & tolerances whilst dynamically adjusting the temperature based on the time of day. The SleepFit could be used to ensure that home appliances are working at their optimum level to reduce energy consumption. A necessary step towards reducing our carbon footprint and mitigating climate change.

Figure 2: Nest Smart Thermostat

3 Approach

The approach taken to build this system was to use the TI SensorTag 2650 to measure temperature, light and humidity and communicate those values over Bluetooth Low Energy (BLE) to a Raspberry Pi. To start a simple connection was made using *Bluez* to read the data held in temperature designated handle. A handle is a connection endpoint which makes communicating with the actual sensor itself much simpler as it will perform even lower level tasks on it own. Therefore, the only two lines code required was to enable the sensor and read from it. There is a document outlining this for each I^2C sensor in the project repository.

The next step was to implement the same for the light and humidity sensor using a Python script. The *bluepy* library was used to implement this as it already has code which would do just that. A Python script named, Sleep-Fit.py was written with wrapper functions that wrapped around lower level functions, as well as calling the main() function to run the code. Some other features such as saving the data in real time into a .csv file was added so that post-processing of the data could be implemented. A simple UI as shown in *Figure 3* was constructed to present the measurements as well as suggestions on appropriate activities & steps to improve the space's conditions.



```
pi@raspberrypi:~/fitHome $ sudo python fitHome.py
Connecting to 54:6C:0E:4D:AC:00
Press Ctrl-C to quit.
----------------------------------------
Temperature:            17.06
Temperature is ideal for sleep.

Humidity:               41.11
It's too dry.

Light levels:           130.04
X : Sleep | X : Work
----------------------------------------
```

Figure 3: User Interface

4 Results

Figure 4 shows the plots for the 3 measurements at dusk which was taken over a period of 30 minutes to produce 400 data points as the code was written such that it would have a refresh time of about 5 seconds. Though it might seems slow, one has to account for the fact that when the SensorTag is within it's protective casing, it would have a delayed response to the changes in environment as well. Machine Learning techniques could be applied to the data to learn patterns and gain insight (how draughty a room is, or the the direction the room is facing for example).

The measurement readings were compared with a simple Arduino setup with the following sensors, DHT11 (Humidity Sensor) & MPL3115A2 (Temperature Sensor) to assess its precision and accuracy. Indeed the values obtained were to within a tolerable range of each other and thus, the data from the SensorTag we being read and displayed correctly.

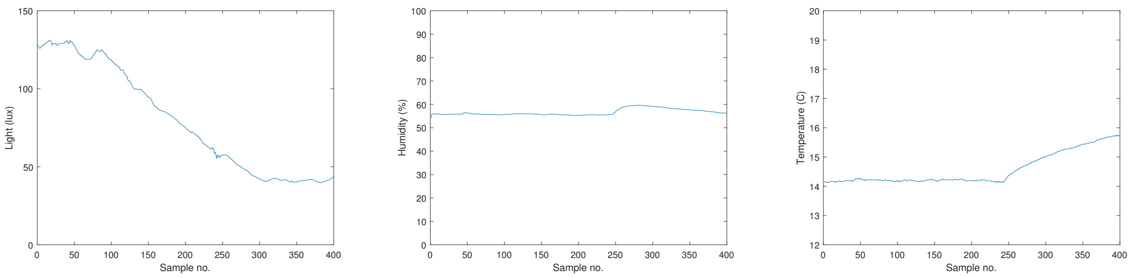


Figure 4: 400 samples of light(left), humidity(center) & temperature(right)