ECE 591/592 791/792

**Internet of Things Architectures, Applications, and Implementation**

**Group #7:**

**Healthy Habits Tracker**

**04/26/2018**

Leonardo Falcon Izzi

Graham Flinchum

Sweta Rout

Individual contributions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component** | **Component weightage** | **Leonardo contribution** | **Graham contribution** | **Sweta contribution** |
| Sleeping pattern | 0.2 | 100% | 0 | 0 |
| Noise level | 0.2 | 0 | 100% | 0 |
| Cooking pattern | 0.2 | 0 | 0 | 100% |
| Analytics | 0.15 | 33.33% | 33.33% | 33.33% |
| Cloud and Web Page | 0.1 | 33.33% | 33.33% | 33.33% |
| Training Set and Testing Set | 0.05 | 33.33% | 33.33% | 33.33% |
| Report writing | 0.05 | 33.33% | 33.33% | 33.33% |
| Presentation and Demo | 0.05 | 33.33% | 33.33% | 33.33% |
| **Per student contribution** | 1 | 33.33% | 33.33% | 33.33% |

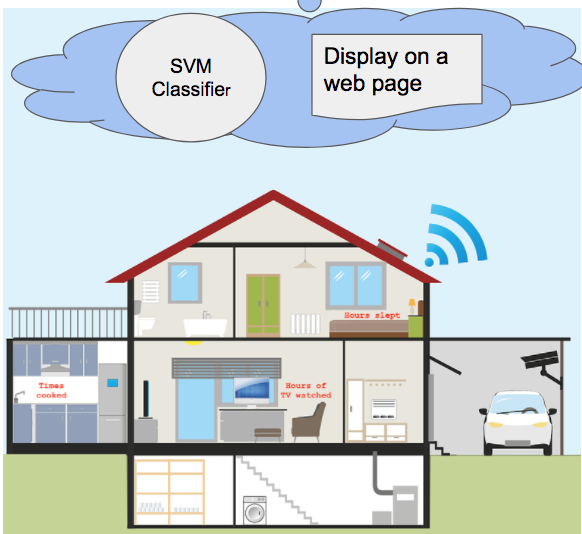
Introduction:

It is well known that improving health habits can diminish the risk of heart disease, stroke, high blood pressure, obesity, and it even improves mood. However, how can you tell if you’re practicing healthy habits?

The objective of our project was to design and implement a non-invasive monitoring system that would track the health habits of a resident. Having gathered this data, we would then feed it into a classifier that would determine the health of the resident. This classification would then be provided as feedback to resident, so as to adjust their habits. We determined, from reading health studies, that 5 significant drivers of health are eating healthy, drinking plenty of water, reducing stationary time, sleeping well, and exercising regularly.

We leveraged IoT technologies to automate sleeping, eating, and tv watching habits tracking in a smart home. We manually logged our habits and asked some of our peers to do the same. This served as training data for our Support Vector Machine (SVM) classifier. Test data results updates daily on a webpage for daily feedback to user.

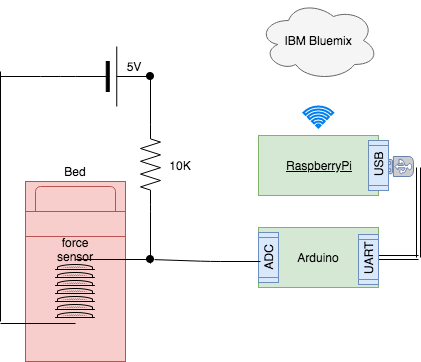
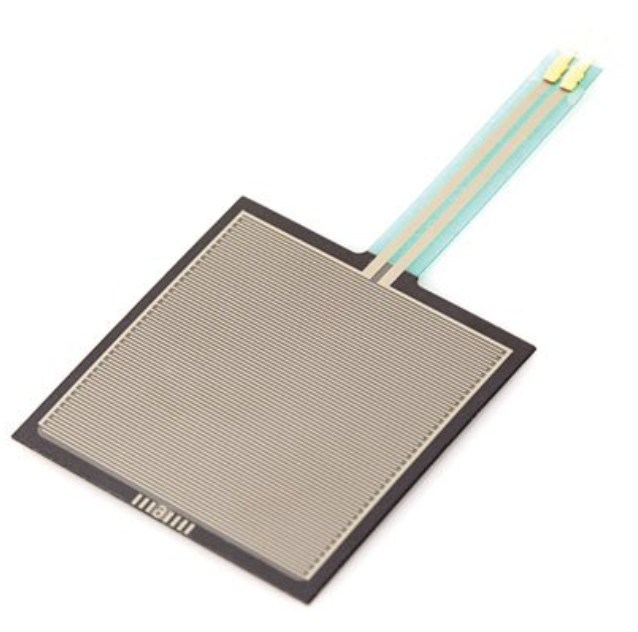
Design:



The figure above shows a graphical representation of the system implemented. Non-invasive sensors were designed in such a way that they would not interfere with the residents activities or what is being measured. These IOT devices broadcast data daily to IBM bluemix cloud hosting an SVM classifier and a webpage. The trained SVM classifier will then determine a health class for the resident, and post it for easy viewing on the web page.

**Sleep tracker:**

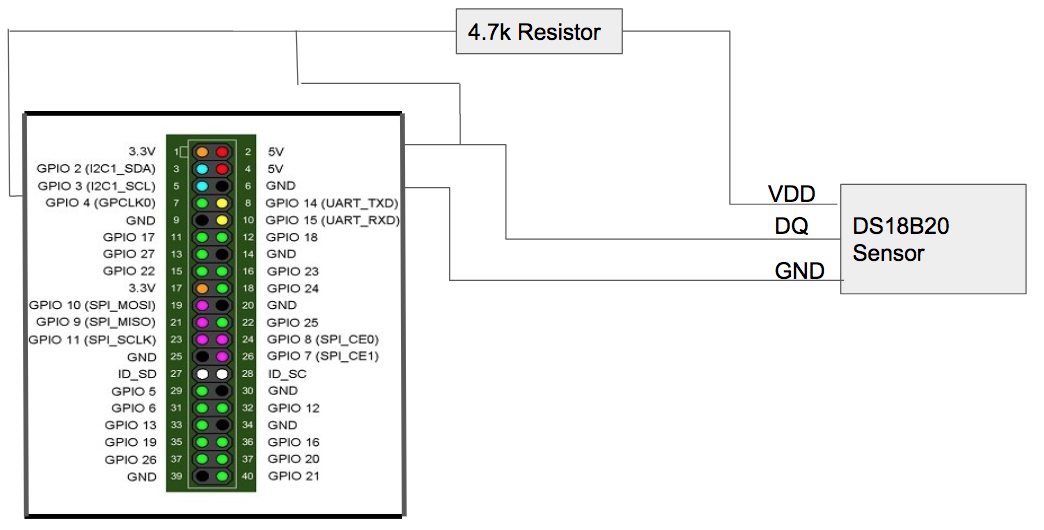
In order to track sleep a force sensor was used. The idea is to measure the presence of someone on a bed, as well as movement. These two factors could determine with reasonable accuracy whether a person is asleep, and can then accumulate time asleep. The sensor used is a force sensitive resistor manufactured by Pololu (pictured below).



In order to measure the resistance of the sensor, a voltage divider network was implemented (shown above). Whenever the resistance of the sensor decreases, the current through the sensor increases, and builds up a larger voltage. This voltage is captured by the analog to digital converter (ADC) of the Arduino Uno. The arduino uno samples the ADC every 500ms and serially pushes out the message to the Raspberry PI. The raspberry PI then runs it’s sleep detection algorithm (explained later) and accumulates hours of slept over the course of 24 hours. At the same time every day, it will publish a normalized number of time slept.

**Cooking counter:**

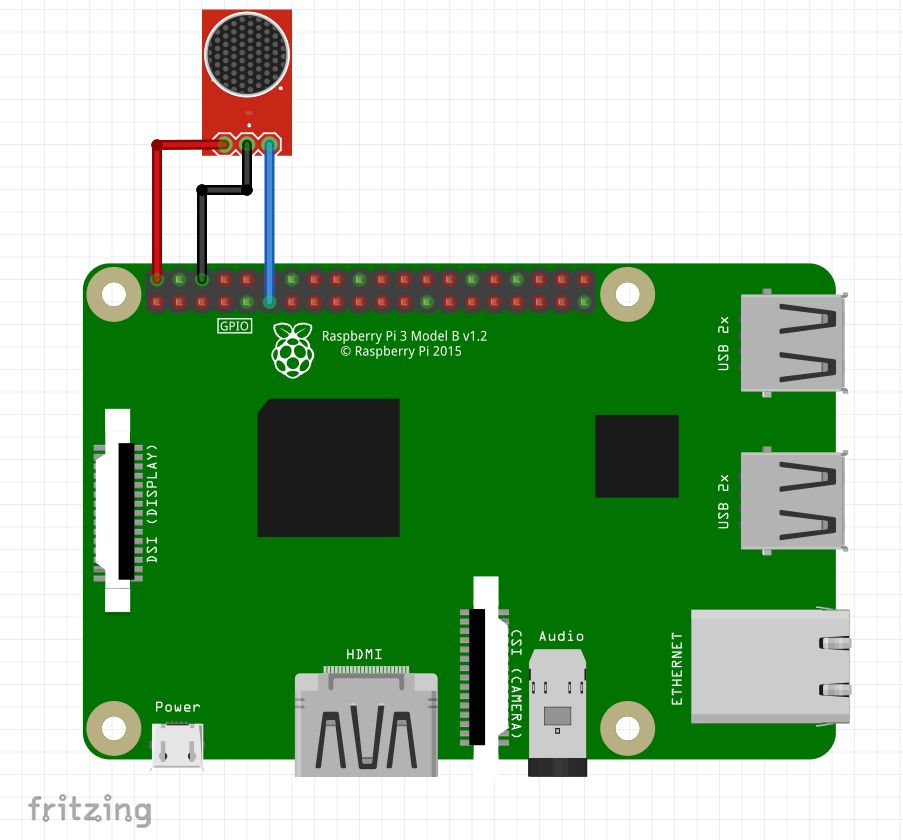
In order to keep track of a resident’s cooking habits. A DS18B20 temperature sensor was placed in proximity of the stove. Rapid increases in temperature would indicate that the resident is actively cooking. The temperature range that can be captured by this sensor is -55 to 125 in Deg celsius.

****

As shown above, the temperature sensor communicates to the Raspberry PI using “One-Wire” communication protocol, a proprietary serial communication protocol that uses only one wire to transmit the temperature readings to the microcontroller.

**TV watching tracker:**

In order to track time of TV watched, a sound sensor is placed in close proximity of a TV. It must be in such a way that the volume of normal conversation does not trigger the sensor. The sensor we used is a DOAKI microphone, and its ASIN is B00XT0PH10. It detects sounds but does not distinguish between different frequencies, has a physical dial that can be used to adjust its sensitivity, and outputs a digital 0 or 1, with 1 meaning that a sound is detected.

****

**Classifier:**

The classifier is hosted on the IBM bluemix cloud. It subscribes to the IOT clients and expects updates from them once a day.

Feature Selection :

We selected 5 features : Hours of sleep, Minutes of exercise, Cups of water, Hours of watching TV, Number of cooked meals. Based on these selected features, a person is classified to be Healthy, Average, Below Average or Not Healthy. All three members of the group guaranteed and agreed for the ground truths and class labels for each data sets in the train data. Every day, data was collected by the group members and friends for creating a training data set.

Training Model :

A fitted model is created based on the training set using SVM classification. Support vector machines (SVMs, also support vector networks) are **supervised** learning models with associated learning algorithms that analyze data used for classification and regression analysis.

Classification:

At the end of the day, data recorded from the sensors are sent to the IBM bluemix server. SVM classifier is run on the server to classify everydays’ activities and classify it. Manual data (cups of water and minutes of exercise) are given as input on the webpage.

Accuracy:

Similar to training set, test set is also created and given to the training model. After crossfold validation of 10 folds, it is found to have 80% accuracy. ‘Not Healthy’ test data is mostly found to be ‘Below Average’ as our training set didn’t have much data points classified as ‘Not Healthy’.

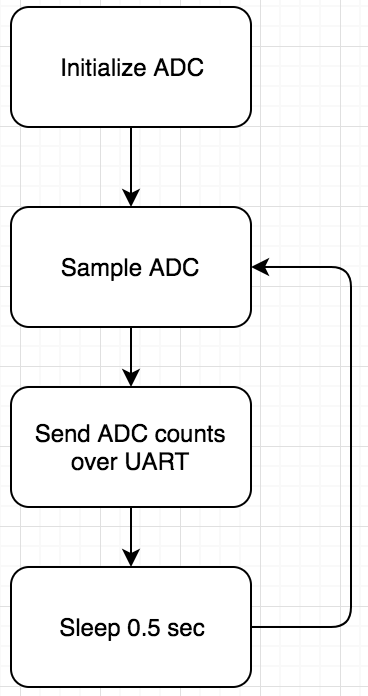
**Cloud and Web page:**

An IoTF service is created on the bluemix server. And, an app is created to deploy a web page to display results and take manual feature values as input. All the Raspberry Pis in the system are registered and devices are configured on the cloud. The devices are made to communicate with the bluemix app using the authentication tokens.

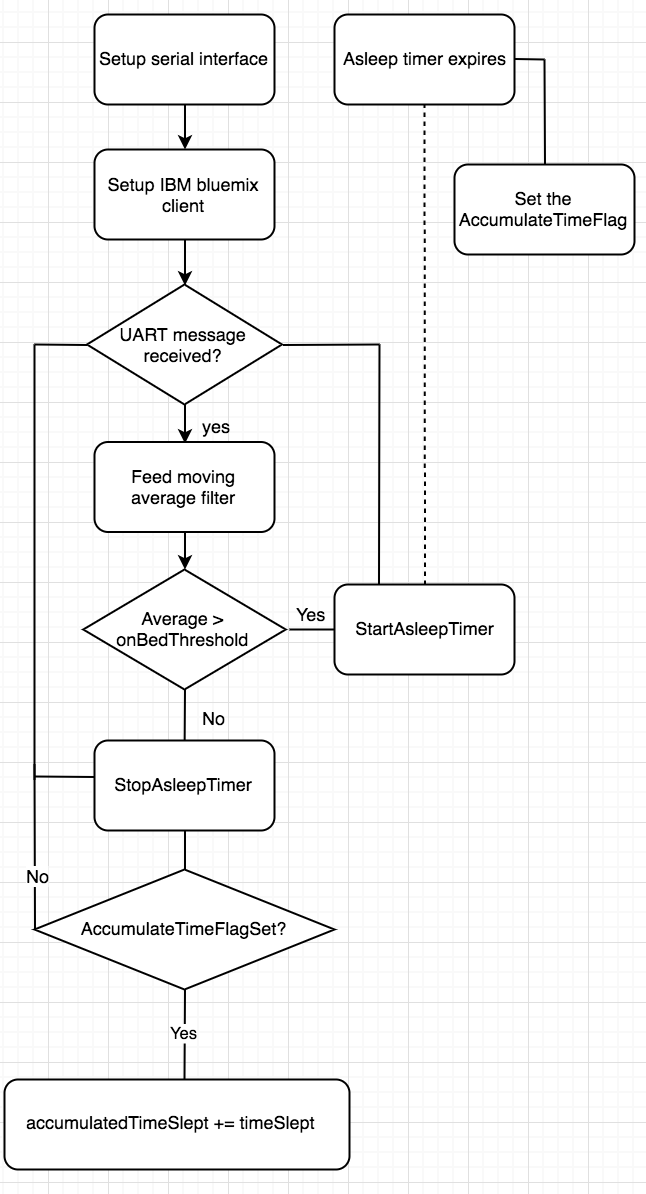
Implementation:

**Sleep tracker:**

The software on the Arduino was written in C, meanwhile the software on the Raspberry Pi was written in Python. The following flowcharts describes the flow of the sleep tracker software.



The flowchart on the left is a representation of the software running on the Arduino.



The flowchart above describes the sleep detection algorithm. It shows that the on-bed detection takes place when the sensor output surpasses a threshold, and the asleep detection takes place whenever the resident has been still in the bed for a certain amount of time. A moving average filter was used to smooth out the output from the sensor. When the resident gets off the bed, we add to the accumulated time slept the time slept during that particular session.

**Cooking counter:**

The programming language used for calculating number of meals is Python. And the interface used is modprobe. On setting w1–gpio and w1-therm, temperature is recorded by the sensor in the path /sys/bus/w1/devices/28-XXXXXXXXXXXX in celsius. Temperature is monitored from this

folder and accordingly number of meals is calculated. When the stove is ON, temperature recorded is within 130 deg Fahrenheit to 300 deg Fahrenheit. Accordingly number of meals are calculated per day. At the end of the day, this computed value is normalised in range of 0-1 and published to the cloud.

**TV watching tracker:**

The software for tracking the amount of TV watched was written in C and run on a Raspberry Pi that powered the sound sensor. When a sound is detected, a timer is started. If another sound is detected before the timer expires, then the timer is reset and waits for another sound. If no sound is detected and the timer expires, it adds the amount of time that it was detecting noise to the running total for the day, which will be published to the Pi once a day.

**Cloud and Web page:**

Programming language used to subscribe events from Raspberry Pi and deploy the webpage on bluemix.net is Python. Webapp deploys the web page using python library ‘Flask’. The web page is created in HTML, CSS, JavaScript, Jquery and AJAX. The web app subscribes to the events ‘sleep’, ‘exercise’ and ‘TV’ and records the values. When the user enters cups of water and minutes of exercise information on the web page and press enter key, it checks whether the sensor data is received by the cloud, If, not yet received, it displays ‘WAITING FOR DATA’ . Else, it normalizes cups of water and minutes of exercise and sends to the classifier along with the sensor data. Finally, a daily dashboard is displayed showing daily recorded values and the class label that these values are classified into.

Results and discussion:

The scope of the project was to automate the detection of sleep tracking, TV watching, and cooking times. However, during development we logged our sleeping, times cooked, TV watched, exercise, and cups of water drank. We also asked a few friends to help. The goal being having as much training data as possible for our classifier. Once acquired a sufficient number of samples, we classified each as “Healthy”, “Average”, “Below Average”, and “Unhealthy”. We did this based on research from multiple health studies. We gathered over 30 samples in total, which correspond to 30 days of healthy habits tracking from 6 different people.

In order to test our classifier we asked new friends to log their health habits for a day. Not surprisingly, most of the them fell in the “Average” category.

|  |  |  |
| --- | --- | --- |
| Feature | Max | Min |
| Sleep hours | 10 hrs | 3hrs |
| Water cups | 22 cups | 1 cups |
| Exercise minutes | 90 min | 0 min |
| Cooking times | 5 times | 0 times |
| Hours of TV watched | 5 hours | 0 hours |

The table above shows the normalization values chosen, based on the training data gathered.

Some areas of improvement identified were:

* Sleep sensor was very sensitive to the subject being right above the sensor; some shifting on the bed could drastically affect the output. In the future, we would add an array of multiple sensors. This would enable more consistent tracking of a subjects movements on the bed.
* Classification gives 80% accuracy and can be improved by oversampling of training data and more specific data points for all possible ranges.

Related work and references

* Sound Sensor (Raspberry Pi) <http://www.instructables.com/id/Sound-Sensor-Raspberry-Pi/>
* Exercise: How much should the average adult exercise every day? - Mayo Clinic <https://www.mayoclinic.org/healthy-lifestyle/fitness/expert-answers/exercise/faq-20057916>
* Water: How much should you drink every day? - Mayo Clinic <https://www.mayoclinic.org/healthy-lifestyle/nutrition-and-healthy-eating/in-depth/water/art-20044256>
* How many hours of sleep are enough? - Mayo Clinic <https://www.mayoclinic.org/healthy-lifestyle/adult-health/expert-answers/how-many-hours-of-sleep-are-enough/faq-20057898>
* 6 Bad Things That Happen When You Sleep Too Much: Your mood can suffer. | Prevention <https://www.prevention.com/health/6-bad-things-that-happen-when-you-sleep-too-much/slide/6>
* Limit TV watching to 2 hours to live longer, say scientists - Telegraph <https://www.telegraph.co.uk/news/health/news/9386569/Limit-TV-watching-to-2-hours-to-live-longer-say-scientists.html>
* Circuit basics for temperature sensor <http://www.circuitbasics.com/raspberry-pi-ds18b20-temperature-sensor-tutorial/>
* Deploying Web page https://www.ibm.com/blogs/bluemix/2015/03/simple-hello-world-python-app-using-flask/