IoTSSC Project

Room Occupancy Monitoring

Paul Patras, School of Informatics, The University of Edinburgh

Issue date: 11 January 2019

1. Introduction

The aim of the IoTSSC project is to develop a full-stack IoT system that monitors the occupancy of a room, which is routinely shared among different users/groups. In this project, the experimental environment will be the lab space in room 4.12 in the Appleton Tower. Full-stack in this assignment means you will be developing:

1. Firmware for an embedded system,
2. A simple Android app that will enable forwarding the measurements collected by the embedded system to the cloud.
3. A cloud-based analytics pipeline including a visualisation dashboard.

You will be working as two-member teams. Each student will be provided with a Nordic NRF51-DK development board and sensors (accelerometer, gyroscope, magnetometer, Time of Flight distance sensor). Teams will be provided with a Motorola smartphone to be used for the purpose of this project. In addition, each student will be provided with credits to the Google Cloud Platform, which should be used to implement the cloud-side functionality.

Important: Please send me an email at paul.patras@ed.ac.uk with your name, UUN, and a valid GMail address. I will respond with a code that you can use to add credits to your Google Cloud Platform account. You will not need to create a separate account for this; just go directly to https://console.cloud.google.com and login using your GMail account.

Based on the prototypes developed and results obtained, you will write a report in workshop paper format. The report should appropriately introduce the problem domain and challenges, present the key building blocks of the designed prototype, and discuss the most important experimental results obtained.

The requirements specified in this handout provide you with guidance on the core functionality you are expected to achieve by the end of the project. Specific details are intentionally omitted, to stimulate your creativity, allowing you to make your own design choice and to add extra features that you find useful as you develop the prototype.

The project will be evaluated in two parts, as follows:

- **Part 1:** You are required to submit a short proposal document, outlining the planned IoT prototype. This assessment is formative (i.e. for feedback only).
- **Part 2:** The second part is summative and is marked on the workshop style paper that you documents your project and results obtained. This carries 55% of the marks.

2. Coursework Requirements

The project work will be split into two phases. The first phase will enable you to get started with embedded systems development, along with using the cloud platform; in this phase you should solely focus on gathering readings from the sensor(s) you choose to work with and relay these via the Android app to a cloud end-point. The second phase will focus on the practical implementation of the IoT system, that is making sense of the data that you collect and optimising the embedded system's operation. It is important that you complete Phase 1 first, otherwise you will not be able to make progress in Phase 2.

Phase 1 - Sensing and communication with the cloud
At the end of the first phase you should be familiar with the embedded device operation and its communication with the cloud end-point. This involves:

1. Developing an embedded application that will run on the IoT device and read from different sensors attached to it. This information can be time-of-flight sensor readings, the Bluetooth address of mobile devices sensed within the range of your system, or other sensor output that you identify as useful to achieve the room occupancy monitoring task. Your prototype will continuously/periodically scan the environment and record the sensor type, type of measurement collected (e.g. BLE advertisement and signal strength, PIR output, etc), the time of acquisition, and the value (BLE address, output level, distance, etc.).
2. Developing a simple Android app that will query the embedded system for measurements and retrieve these from the device via BLE communication.
3. Uploading the sensed information collected to the cloud for storage. Communication with the cloud must be performed using the Android app.

The documentation on the Google Core IoT core tutorial page on how to upload data to the cloud using the HTTP bridge will prove useful for point 3 above. You may use BigQuery to store the data that you collect. Alternatively, you can create a Google virtual machine (VM) and run an HTTP server within that to receive measurements and subsequently process these with whatever tool you find appropriate.

Important: Note that you will have a limited budget (credits amount) to work with the cloud. Hence you should be careful about how often you want to upload readings and you may want to compress payloads before making requests. Likewise, you may not want to leave a VM running all the time.

Phase 2 - Room occupancy monitoring

The second step is to build a room occupancy monitoring system using the sensors provided. This can be as simple as tracking whether there is someone in the room or not and for how long the room is busy/occupied, or how many occupants there are during precise time intervals. More sophisticated analysis is also encouraged and will be marked accordingly. This may include statistics about busiest times of the day for the monitored room, the distribution of the number of users throughout day/week, the most popular locations in the room, and so on. You can employ whatever sensor you wish and place the board in any location (fixed or mobile) you choose. If you use Bluetooth signals to detect the presence of users carrying mobile phones, be sure to estimate the distance to the user, so as to be confident that the user is within the room of interest. Note that you may need more than one embedded device if you take this route and employ some propagation model to estimate the distance to a mobile device based on the received signal strength (RSS).

The creative part comes in when integrating multiple sensors, since they can be affected by noise and other distortions. Therefore, working with imperfect data is a big plus to have in your occupancy estimation algorithm. Do not feel limited by the computation capabilities of the embedded device. If your algorithm requires more computation resources, the measurements collected by various sensors can be processed in the cloud, as long as they are appropriately time stamped.

You are encouraged to think of creative ways to collect ground-truth information (e.g. the exact location of a user in a room), which you can use for validating your implementation and evaluating the performance of the room occupancy algorithm you will implement - this will also bring you points for the rigour demonstrated in the evaluation of your solution.

In terms of data visualisation, you can be as creative as you wish and come up with interesting ways of visualising the data your device collects, how your algorithm performs, or how the occupancy of the environment sensed evolves across time and space. Think about what key insights a user can gain from your visualisations!

3. Resources Provided

IoT Device

You are provided with an Nordic nRF51 Development Kit with the following specifications:

- 32-bit ARM Cortex M0 CPU core
- 256kB/128kB flash + 32kB/16kB RAM
- Bluetooth LE support
- Arduino Uno shield compatible
- Configurable I/O mapping for analog and digital I/O
- 4 LEDs and 4 buttons, user-programmable.
- Flexible power management

The device datasheet is available here.

Sensors

A list of available sensors to connect to your board is given below:

- Adafruit VL53L0X is a Time of Flight distance sensor
- Parallax Passive Infra-Red (PIR) sensor.
- Adafruit LSM9DS1 Accelerometer + Gyro + Magnetometer
- Analog Devices TMP36GT9Z temperature sensor

You should take advantage of these sensors in creative ways to increase the accuracy of your room occupancy monitoring system.

Mobile Phone

You are provided with a Motorola Moto G (3rd generation) mobile phone running Android 5.1.1. Please do not perform software updates on these phones, as we are sharing them with the students taking the Secure Programming course.

The Cloud

You are provided with credits to use the Google Cloud Platform. Please contact the course organiser with a valid GMail account to receive these credits.

4. Deliverables, Deadlines, and Marking

The IoTSSC coursework assessment consists of two parts, as follows:

- **Part 1:** You will need to submit a document proposal outlining the planned prototype (firmware, communication, cloud integration, analytics), envisioned building blocks, algorithm(s) you plan to implement for achieving accurate room occupancy estimation, why you chose these, and a brief description of the evaluation methodology you intend to follow. This should not be longer than 2 pages. One proposal per team is sufficient, just make sure each team member submits the proposal in PDF format as follows:

  ```
  submit iotssc 1 proposal.pdf
  ```

  Deadline: Monday 28 January 2019, 16:00.

- **Part 2:** The second part of the assignment is summative and weights 55% of the final mark. For this part, you will need to submit:
  1. Device firmware for implementing the functionality developed during both phases.
  2. Mobile app implementing the communication with the embedded devices and respectively with the cloud.
  3. Any appropriate cloud code used for data processing.
  4. **Individual** reports that give a complete summary of your work.

The report should be prepared as a workshop style paper, formatted using the ACM LaTeX template available here, and should not exceed 7-pages. You can use as many pages as you wish for references.

You should use the following document header

```
\documentclass[sigconf]{acmart}
```

and format your bibliography using the `ACM-Reference-Format.bst` style.

The report should introduce the problem domain, and challenges facing room occupancy monitoring tasks and IoT prototyping in particular. You should describe your end-to-end prototype and include appropriate figures, discuss key design choices, and present the approach you took to implement the occupancy estimation process, motivating your choice. You should provide evidence of the prototype's performance, including graphs obtained with your visualisation tool(s) and/or tables documenting the performance of your system. Be creative and think about what matters beyond the accuracy of the tasks that you are implementing, e.g. code size, CPU/memory usage, energy footprint, and security/privacy guarantees of the system.

You should put all your code and report in a directory named *prototype* with the following structure (your firmware code should sit under a *firmware* sub-folder, the app code should reside in a *mobile-app* sub-folder, any cloud code you implemented for data processing should be placed in the *cloud* sub-folder:

```
prototype
    |
    □- firmware
    |
    □- mobile-app
    |
    □- cloud
    |
    □- report.pdf
```

Then submit your directory with:

```
submit iotssc 2 prototype
```

Deadline: Wednesday 20 March 2019, 16:00.

Deadlines are strict. Code sharing between different teams is strictly not permitted. You may use blocks of external code that you found online, and of course you are free to use standard frameworks, but please add an appropriate comment to acknowledge and explain such choices.

Marking
All teams will be ranked according to the quality of their implementation, the efficiency of their algorithm (occupancy estimation accuracy), and quality of the report.

To demonstrate the accuracy of the room occupancy monitoring functionality implemented, we will choose 3 occupation scenario/user movement patterns in the lab environment and each team will need to show how their solution estimates the occupancy of the room. This evaluation will be carried during a lab slot in week 9, that is on **Thursday 21 March 2019 from 14:10**, and it should take at most 5 minutes per team. Note that you are not expected to discuss any detail of the actual implementation during this quick evaluation, which is distinct from the presentation of your project.

Your report will be assessed on a number of criteria, as explained below.

Basic Criteria
1. Understanding of the problem
2. Completion of the project
3. Quality of the work
4. Quality of the report

Additional Criteria
1. Knowledge of existing solutions
2. Justification of design decisions
3. Solutions to any conceptual problems
4. Evaluation of solution
5. Amount of work

Exceptional Criteria
1. Evidence of originality
2. Publishable research

Additional notes:
- Excellent completion of the project means the submitted IoT end-to-end stack implements all the required functionality, i.e. embedded device application, communication with the cloud either via the mobile phone, data processing for occupancy estimation, data visualisation.
- Evidence of excellent quality of the work may include code optimisation for embedded devices.

- The evaluation will involve showing rigorous efforts were made to demonstrate that the produced room occupancy monitoring functionality is accurate and the performance of the implementation has been thoroughly assessed.
- The written report must be well written and organised, and provide a clear summary of the system design, algorithms implemented, and insights gained, which are backed by appropriate graphs.
- The amount of work will be measurable through the number of approaches explored to implement the occupancy monitoring functionality, including fusing data from different types of sensors to improve accuracy.

5. Project Presentations

Project presentations will be separate and carry additional **15 marks**. These will take place on **Friday 22 March from 14:10.** Each team is expected to present their prototype and results within 12 minutes, followed by 3 minutes of questions.

The presentations will be assessed in terms of:

1. How clearly you explain the context and formulate the problem you solved;
2. Description of technical challenges faced;
3. Description of solution design and methods adopted to solve the problems identified;
4. Evaluation and results;
5. Overall quality of presentation.