CO838 Assessment 5:
Aggregation, Processing and Visualisation

Deadline: Mon 9th May 2016 23:55

Description

For this assessment you should work on your own to design and build a small project of your own design that demonstrates an application of the Internet of Things to a real-world application of your choice. In particular, your project should demonstrate at least two of the following, including at least one of either A or B, using MQTT as a way of communicating between them.

(A) Using a K64F board (or simple XBee-enabled sensors) to gather data

(B) Using a K64F board to output data or control some device.

(C) Displaying the data on the web or on a mobile app.

(D) Using a source of data from the Internet, a social network, or email/texts as an input to the system or/and as a means of output.

You should be careful to *design* the project before you start on the coding and hardware aspects of it. You can make use of code from previous assessments on this module in writing the code for this module, and/or the code demonstrated in lectures and seminars (please note such use with comments). If you use imported MBED libraries, keep these as imports (rather than copying+pasting the code into your own files). Be aware of the regulations concerning plagiarism and duplication of material (see the Moodle page for a link to the relevant FAQ).

You should think of an idea for your project by the end of week 12, and prepare a brief (1-2 minute) spoken presentation of your idea for the seminar session. The seminar session will be used to discuss the ideas, their feasibility and the appropriate technologies. Please email me (C.G.Johnson@kent.ac.uk) by 23:55 on the 8th April with one or two sentences describing your project idea.

This assessment counts for 40% of the module mark.

Examples

Here is a list of examples; I am happy for you to do one of these, but ideally you should be coming up with your own ideas:

- Gather temperature readings from the campus (or elsewhere) (A), and display the readings from them alongside the predicted local temperature from an open data weather forecast (D) in the form of thermometer readings on a web page (C).
- Gather temperature readings from a location (A), and display a graph of the temperature values over time, on a web page (C).
- Sound a buzzer (B) whenever there is an earthquake in the world (D). The volume of the buzzer depends on the distance of the earthquake from Canterbury.
- Estimate occupancy of a room on campuss using a switch on the door or a light sensor (A). Compare this against the online timetables for those rooms (D).
- Ring a bell (B) when the average temperature over the last half-hour gets below 10C (A).
- Display on an LCD display (A) the current most popular keywords on Twitter relating to the University of Kent (D).
- Build a fitness tracker (A) that counts steps, and posts to your Facebook account every thousand steps that you do (D).

- Adapt a dancing sunflower gadget (B) so that it moves more or less depending on how often its name is mentioned on Twitter (D).
- Build a device that reduces the volume on your computer (B) for a minute when the doorbell rings (A)—or when someone knocks on the door (A)!
- Make a "falling down" alert system: build something that posts on a web-page (C) or by email (D) which displays an alert whenever the accelerometer makes a downward movement within a short period of time (A).

You should think carefully about the design of your system before you start to build it. Make use of small prototypes of components of the system before plunging in to building the whole system.

Submission

For submission, you should create a single ZIP file that contains all of the source code for your solution. This is primarily so we can look at it to give you useful feedback. The top-level of the ZIP file should be structured with sub-directories for each device or program (e.g. K64F device code, host visualisation, control stuff). Code can be exported from the MBED environment easily enough — use the "ZIP archive" option (the resulting zipfile will have some FRDM-K64F specific binaries/objects, but that's okay). Please also give a link to the relevant MBED projects on the developer.mbed.org site.

You should also include a brief description/walkthrough (in the form of a pdf file) of the various parts and how they fit together, and a brief video (e.g. as a link to a YouTube/Vimeo video, or just include the video file in the zip file if it is small enough) demonstrating your system working and talking through it.

Sensors

Though the variety of on-board sensors are limited, it is relatively easy to wire up contact-switches or light sensors to the device (though that may preclude the use of the MBED application shield). The Shed can advise on and/or loan breadboards, wires, sensors. Keep it simple and don't over-commit yourself: better to have a simple system that works well, rather than a horrendously over-spec'd and only partially complete one.

The MQTT Server

Whatever is being sensed (door opening/closing or temperature, orientation) or being controlled (a bell sounding) should be communicated through an MQTT server. There is a University MQTT server available, or you can use a public server (see e.g. the week 11 seminar example). The University server is:

hostname: doughnut.kent.ac.uk (129.12.3.210) port: 1883

How exactly you talk to the MQTT server will depend on where your device is and how it is connected to the Internet. The host 'doughnut' is visible from some (but not all) campus networks, including 'raptor'. Thus, if you are off- campus or cannot see 'doughnut' directly, you'll need to *forward* connections over an SSH connection to raptor. On Linux (and perhaps MacOS X) this can be done with:

```
linux$ ssh -L 1883:doughnut.kent.ac.uk:1883 abc123@raptor.kent.ac.uk
```

replacing abc123 with your login. The Windows program 'putty' supports a similar mechanism of port forward- ing. In either case, connecting to your own PC/Mac on port 1883 should connect through to the MQTT server on 'doughnut'. This is an instance of the 'mosquitto' (http://mosquitto.org/) server, running *without* any authentication or encryption: thus, anyone can potentially post anything (some level of trust is assumed!); connections are logged.

Whether or not you implement the required MQTT protocol handling manually (i.e. on top of TCP sockets/streams) or use a convenient API is up to you — the choice may be dictated by the language and whether or not a suitable MQTT API is available.

MQTT Topics

The processed data from your sensor(s) should be published on the MQTT server using an appropriate topic (or multiple topics). As examples, the server on 'doughnut' has the following topics that are published to periodically:

unikent/building/cornwallis/room/S113/location	location as "lat:long" coordinates (5dp)
unikent/building/cornwallis/room/S113/ups/temperature	UPS internal temperature (Celsius)
unikent/building/cornwallis/room/S113/ups/linev	UPS line voltage
unikent/building/cornwallis/room/S113/ups/load	UPS load (percent)
unikent/building/cornwallis/room/S113/ups/statusstr	UPS status string

You can choose whatever topics are sensible/appropriate: if in doubt, emailme (C.G.Johnson@kent.ac.uk). For instance, if you were monitoring the temperature of your student house, suitable topics might be:

canterbury/hales-place/42/mysensor/temperature	temperature (Celsius)
canterbury/hales-place/42/location	location as "*lat:long*" coordinates (5dp)

Of course, you might want to be a little more anonymous in relation to where your sensor actually is:

unikent/users/abc123/mysensor/temperature	temperature (Celsius)
unikent/users/abc123/mysensor/pot1	value of potentiometer 1

Getting Help

Technical questions should be asked on the CO657 Moodle forum, but since these are publically visible (to the group) don't blindly paste in your whole program and simply ask "what's wrong?": frame questions in a generic way (and ideally that in a way that the response may be useful to other students on the module!). Queries by email (to C.G.Johnson@kent.ac.uk) are also fine.

Marking

The following criteria will be taken into account when marking the assessment (some areas may be irrelevant depending on the topic that you choose, e.g. you will not be penalized for choice of sensor if the project is not about sensing):

- The quality and complexity of the topic chosen.
- The design of the system.
- The choice of sensors and/or output.
- The coding around the sensor and/or output
- Handing the data, including appropriate processing of the data (outlier removal, scaling, keeping outputs within bounds).
- Quality of the web page or other communication.
- The quality of the MQTT communication code
- Robustness and security of the code.

Colin Johnson, School of Computing