UNIVERSITY OF LIVERPOOL

Department of Electrical Engineering and Electronics

ELEC423

Assignment 1 – MQTT Assignment

Module	ELEC423
Coursework name	Assignment 1 – MQTT Assignment
Component weight	15 % (of 15 credits)
Semester	1
HE Level	7
Lab location	PC lab 402 as timetabled, at other times for private study
Work	Individually
Timetabled time	6 hours
Suggested private study	16 hours (including report writing)
How much time did it take YOU?	Let us know anonymously via https://bit.ly/EEECARES
Assessment method	Summative
Submission format	Online via CANVAS
Submission deadline	Week 7 (Sunday 12th of November at 23:59 UK time)
Late submission	Standard University penalty applies
Plagiarism / collusion	Standard University penalties and procedures apply for plagiarism and collusion.
Resit opportunity	Yes, a resit exam will replace the Coursework Assessment (if the total module failed)
Marking policy	Marked and moderated independently
Anonymous marking	Yes
Feedback	Via comments on Canvas
Learning outcome	LO2: Demonstrate an ability to design and implement IoT based prototypes. LO3: Demonstrate a knowledge of the IoT architecture and technologies used to design and implement it. LO4: Demonstrate a knowledge of IoT embedded systems based on Linux.

Marking Criteria

Section	Marks available	Indicative characteristics	
		Adequate / pass (50%)	**Very good / Excellent**
Presentation and structure	15%	• The submission contains a document with a cover page (title, background, academic integrity declaration), the description contains focused screenshots, and an Appendix with full-screen screenshots (see below). • Submission contains original source code file(s). • Comprehensible language; punctuation, grammar and spelling are accurate. • Equations are legible, numbered and presented correctly where relevant.	• Appropriate use of technical, mathematic and academic terminology and conventions where relevant. • Word processed with consistent formatting. • Pages and equations are numbered; figures and tables are numbered/captioned. • Clear section headings and sub-headings where relevant • Correct cross-referencing (of figures, tables, equations) and citations where relevant.
Design and Method	50%	• Code is original and relates only to the assignment objectives. • Code is tidy, efficient and easy to follow / understand. • Code is clearly laid out and appropriately commented. • Code corresponds to design for each step. • Clear explanation of key procedures undertaken.	• Code is elegant, efficient and "DRY" (Don't Repeat Yourself). • Comments show excellent understanding of syntax and semantics. • Clear explanation of all aspects of your code. • Clear explanation of all the testing carried out. • Code is user-friendly where relevant.
Results	25%	• Execution results present for each block and well annotated. • Results presented for each step, including screenshots of the working solution. • Results for each step accompanied by a commentary. • Screenshots of program output, including the full desktop (date and time included) in the Appendix.	• Screenshots and code demonstrate successful, correct output for every task. • Tests indicating that there are no problems caused by wrong inputs. • The programs only satisfy the objectives of the Assignment, with no extra unnecessary functionality.
Introduction and Discussion	10%	• Problem background introduced clearly. • Discussion on what worked and what didn't. • Critical assessment of the design – strengths and weaknesses.	• Excellent understanding of the problem background is displayed. • Discussion on what could be improved or enhanced. • Discussion on how the program was fully tested.

IMPORTANT: Marking of all coursework is anonymous. Do not include your name, student ID number, email or any other personal information in your report or in the name of the file submitted via CANVAS.

DISCLAIMER: You can use the IoT kit or the IoT testbed to attempt the assessment. Therefore, if any issue arises with the IoT kit, this will not be accepted as an excuse.

DISCLAIMER: The IoT testbed is a testing environment, and you will be responsible for any activity carried out while using your account. Every activity is monitored, and you must comply with the IT Regulations and policies.

WARNING: If you are going to use the IoT testbed, please make sure that you have changed your account password before starting to work! This can be done using the `passwd` command, as underlined in the Lab 1 script.

WARNING: When marking the reports, I will be looking very closely for any signs of collusion, as this is unacceptable. I need to assess your own ability, not that of your friend or colleague. If I find any evidence of collusion, then the formal University rules will be followed, which normally results in all parties involved in the collusion being awarded 0 (i.e., if you do the original work and knowingly let somebody copy it, you may be awarded 0).

Instructions:

- Read this assessment script carefully before proceeding.
- Keep a record of everything you do.
- If you use data or work from other sources, be sure to reference them.
- Keep a record of all screenshots, results, answers, and comments made for the report. When submitting your work, make sure that all the results, screenshots, etc., are clear and readable; otherwise, you will lose marks.
- If you have any questions or feedback on this assessment, please send an email to V.Selis@liverpool.ac.uk

Pre-requisites

You should have the IoT kit with:

- A Wi-Fi configured interface to connect the RPi4 to the eduroam Wi-Fi network.
- An SSH server accepting SSH connections from a PC.
- A working Bash environment along with the "`nano`" text editor.
- A working DHT11 sensor module.
- Working mosquitto MQTT clients.
- A working MobaXterm or `ssh` application on a PC to SSH into the IoT kit.

Otherwise, you should access the IoT testbed (IP: 138.253.76.230) via an MWS PC on Campus or via the Remote Teaching Centre Service (RTCS) if not on Campus. Detailed information on how to use the RTCS is available on Canvas under "Supporting documents". You can access the IoT testbed using as a username your student ID by adding an "u" before it. For example, if your student ID is "123456789", your username will be "u123456789". By now, you should have changed your account password. If not, you need to use your student ID as a password by adding a "p" before it. For example, if your student ID is "123456789", your password will be "p123456789". **Change your password now!**

You should also have the following:

- Ability to use the DHT11 sensor module (Lecture 5, and Laboratories 3 and 4).
- Ability to use shell commands, and to create and execute shell scripts (Lectures 3, 4, 5, 6, 7 and 8, and Laboratories 1, 2, 3 and 4).
- Ability to collect information from the Linux networking system (Lecture 5 and Laboratory 4).
- Understanding how to apply the ETSI M2M reference architecture model to create IoT solutions (Lectures 4 and 6, and Laboratories 3 and 4).
- Understanding how the MQTT protocol works, including MQTT clients (publisher and subscriber) and MQTT broker (Lecture 6, and Laboratories 3 and 4).

If you are unsure of what to do, please contact me at V.Selis@liverpool.ac.uk before continuing.

Introduction

For this assessment, you will be using the IoT kit or IoT testbed with a Debian-based Embedded Linux (Raspberry Pi OS) to create an Internet of Things solution based on Machine-to-Machine (M2M) communications. This assesses your understanding of the M2M/IoT reference architecture model proposed by the European Telecommunications Standards Institute (ETSI) and how you approach implementing a solution based on this model. This solution will be based on the Message Queuing Telemetry Transport (MQTT) protocol, enabling an MQTT subscriber to autonomously receive messages from an MQTT publisher via an MQTT broker. You will be using an MQTT broker already configured inside the University's network. The details of the MQTT broker are:

- IP address: `138.253.77.122`
- TCP port: `8883`
- Username: `iot`
- Password: `ELEC423_2324.`

Please note that the last dot (`.`) is part of the password. During this assessment, you will be required to use a DHT11 temperature and humidity sensors module with the following connections:

- VCC is connected to 3.3V (pin #1) of the RPi4 board.
- DATA is connected to GPIO26 (pin #37) of the RPi4 board.
- GND is connected to the Ground (pin #25) of the RPi4 board.

You are required to have a working MobaXterm or `ssh` application to SSH into the RPi4 remotely.

Assignment Outline

You are required to create a shell script called "`launcher.sh`" that performs the following tasks:

1. Receive as its parameters the following options and arguments: **(8)**
   - `-h` : Show the help for the script and exit
   - `-s <IP>` : The IP address for the MQTT broker server.
   - `-t <PORT>` : The TCP port used by the MQTT broker server.
   - `-u <USERNAME>` : The username for establishing a secure connection.
   - `-p <PASSWORD>` : The password for establishing a secure connection.
   - `-v` : Enable verbose mode (store logs into a file called "`verbose.log`")

   parse them and check if these are correct.
2. Create a variable called "`UNIX_TS`", which contains the Unix time stamp. **(1)**
3. Manually calculate the digit sum of your student ID and assign this to a variable called "`DIGIT_SUM`". For example, if your student ID is "123456789", the digit sum will be $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 = 45$. **(1)**

4. Launch the mosquitto MQTT subscriber with the following details: **(22)**
   - Username and password.
   - The IP address and TCP port of the MQTT broker server.
   - Use "/sensor/*MAC_Ethernet*/*UNIX_TS*/" as a topic, e.g.:

     /sensor/001122334455/1635151544/

     Where *MAC_Ethernet* is the MAC address of the Ethernet interface (eth0) without the colons, and *UNIX_TS* is the content of the "`UNIX_TS`" variable.
   - Use "*MAC_WiFi*-*UNIX_TS*-sub" as a client ID, e.g.:

     001122334466-1635151544-sub

     Where *MAC_WiFi* is the MAC address of the Wi-Fi interface (wlan0) without the colons, and *UNIX_TS* is the content of the "`UNIX_TS`" variable.
   - Display the message(s) retrieved from the MQTT broker.
5. Wait 10 seconds. **(1)**
6. Launch the mosquitto MQTT publisher with the following details: **(38)**
   - Username and password.
   - The IP address and TCP port of the MQTT broker server.
   - Use "/sensor/*MAC_Ethernet*/*UNIX_TS*/" as a topic, e.g.:

     /sensor/001122334455/1635151544/

     Where *MAC_Ethernet* is the MAC address of the Ethernet interface (eth0) without the colons, and *UNIX_TS* is the content of the "`UNIX_TS`" variable.
   - Create a message containing the UNIX time stamp, and the temperature and humidity sensor values from the DHT11 module. This message will be sent to the MQTT broker. Please note that:
     - Sensor values may not always be available.
     - The temperature and humidity values must be divided by 10.
     - The message should be in the form:

     `UNIX_TS``, `***`TEMPERATURE_VALUE`***`, C, `***`HUMIDITY_VALUE`***`, %, `***`DIGIT_SUM`***

     Where "***`TEMPERATURE_VALUE`***" and "***`HUMIDITY_VALUE`***" are the temperature and humidity values, respectively, "***`UNIX_TS`***" is the UNIX time stamp, and "***`DIGIT_SUM`***" is the digit sum of your student ID.
   - Use "*MAC_WiFi*-*UNIX_TS*-pub" as a client ID, e.g.:

     001122334466-1635151544-pub

Where *MAC_WiFi* is the MAC address of the Wi-Fi interface (wlan0) without the colons, and *UNIX_TS* is the content of the "`UNIX_TS`" variable.

- The mosquitto MQTT publisher needs to publish the message every 20 seconds.

7. Terminate the mosquitto MQTT subscriber and publisher after 5 minutes. **(3)**
8. Terminate the script with an appropriate exit code. **(1)**

The usage for this script needs to be as follows:

```
./launcher.sh -s <IP> -t <PORT> -u <USERNAME> -p <PASSWORD> [-v] [-h]
```

For example:

```
./launcher.sh -s 138.253.77.122 -t 8883 -u iot -p ELEC423_2324. -v
```

Summary

That's the end of the assessment. You should now have created an Internet of Things solution based on a Machine-to-Machine communication protocol (MQTT).

What to Hand In

You should submit a Word or PDF document including a 1-page cover sheet (with title and academic integrity declaration that can be found on Canvas), a contents page, section headings and page numbers, etc. (see Marking Criteria.)
This document should also incorporate:

1. A note stating whether you used the IoT testbed or the IoT kit.
2. Introduction/background (no more than 500 words).
3. Description of each task must be documented with the following information (no more than 10 A4 pages using 12-point Times font):
    a. Written explanation of the functionality of your program.
    b. Evidence of the working steps showing the correct operation (e.g., result code, screenshots, etc.).
    c. Examples of the log file(s) created, if any.
4. Conclusion/discussion (no more than 500 words).
5. Include in the Appendix (not included in the 10-page limit above):
    a. The source code of the program you created with appropriate comments. **Please note that the source code must be included as text, not as a figure!**
    b. All the full-screen versions of your focused screenshots with relevant windows and the taskbar with date and time visible as evidence of original work. These do not have to be numbered.

Please note that the marks distributions above in brackets are an approximate guide and are subject to reasonable, slight adjustments at the point of grading. The bracketed marks add up to 75 and cover 'Design and Method' and 'Results' in the appropriate

proportions. 25 marks are left over for 'Presentation and Structure' and 'Introduction and Discussion' as described in the Marking Criteria. Note that the use of the English language will be taken into account when marking your report; this is worth 5% of the total mark, and part of the 'Presentation and Structure' criteria shown on Page 2.

Version History

Name	Date	Version
Dr V Selis	October 2023	Ver. 1.3
Dr V Selis	October 2022	Ver. 1.2
Dr V Selis	October 2021	Ver. 1.1
Dr V Selis	November 2020	Ver. 1.0