CSC 591/791, ECE592/792
Spring 2022
Homework Assignment # 3

Instructions

- You must do this assignment in groups, as assigned.
- Submit your final codes and other documents (mentioned later in the description) via Moodle.
- The due date and time for this assignment: March 18, 2022, 11:45pm ET.

Objectives

1. To get you hands-on experience with MQTT.

Description

For this assignment, you will require three Raspberry Pis (I will call them Raspberry Pi A, Raspberry Pi B, and Raspberry Pi C) and two computers/laptops (or one computer/laptop and one smart phone) with WiFi interfaces, three LEDs, a light dependent resistor (LDR), a potentiometer, and some general resisters. If you choose, you can use any other prototype that is comparable to Raspberry Pi (i.e., has GPIO pins and runs some version of Linux), such as Beaglebone Black.

First of all, you have to install an MQTT broker on laptop # 1. You can use any MQTT broker you like (for example, look at the following links: http://mosquitto.org/blog/2013/01/mosquitto-debian-repository/, http://www.hivemq.com/try-out/). It is not mandatory to choose a particular broker because I want you to explore around and learn about features of various brokers. In your report, justify why you used the broker that you used. The only restriction is that for this assignment, you have to run the broker on laptop # 1, and not use some online broker (there are several such online brokers available as well for free).

This deployment has 1 MQTT broker (the laptop # 1) and 4 MQTT clients, described below:

1. Raspberry Pi A (this will be publisher as well as subscriber)
2. Raspberry Pi B (this will only be subscriber)
3. Raspberry Pi C (this will be publisher as well as subscriber) [given the inflated prices of Raspberry Pi these days, you can use a laptop/computer/smartphone to replace Raspberry Pi C]
4. Laptop # 2 OR the smart phone (this will only be subscriber)

Connect an LDR and a potentiometer to your Raspberry Pi A through an ADC (search online and you will see numerous projects describing how to connect an LDR and a potentiometer to Raspberry Pi). LDR is just a resistor whose resistance changes with the intensity of light. The potentiometer is basically a variable resister. There is a dial on the potentiometer, rotating which changes the resistance of the potentiometer. We will use the LDR to sense the amount of light, and turn an LED on and off based on the amount of light falling on the LDR. We will use the potentiometer to change the threshold at which the LED turns on.

The Raspberry Pi A will sample the values of both LDR and potentiometer every 100 milliseconds. Every time it samples the values, it compares the values of both LDR and potentiometer with the previous most recent values of the LDR and potentiometer, respectively. If the difference of either the LDR or the potentiometer is beyond a certain threshold (you determine the appropriate threshold values), it publishes the LDR and potentiometer values to broker running on laptop # 1. The Raspberry Pi A will publish the values of the LDR to the topic "lightSensor" and of the potentiometer to the topic "threshold". Before posting the values of the potentiometer to the topic, you may have to scale them. Take a hypothetical example. Suppose when you sample values from your ADC connected to LDR, you might observe that your ADC outputs a minimum value of 10 and maximum value of 100 based on no light to very bright light. Whereas, when you sample values from the potentiometer, you might observe that your ADC outputs values in the range 90 to 250 based on where the dial on the potentiometer is. In this case, you might want to bring the values of ADC for potentiometer in the same range as that for LDR. This can be done in various ways. For example, you can normalize the ADC values of both LDR and potentiometer to lie between 0 and 1 before publishing. You can decide whichever method you like to use. You will shortly see that these values will be used by Raspberry Pi C to decide when to turn on or turn off an LED based on how much light is being sensed by the LDR connected to Raspberry Pi A. With this potentiometer

we can basically control what intensity of the light should be there before the Raspberry Pi B turns off the LED.

Note that to compare the value of LDR or potentiometer with its previous most recent values that the Raspberry Pi A had published to the broker, use the following approach: Your Raspberry Pi A should subscribe to both these topics "lightSensor" and "threshold". Every time a message gets posted to these topics, the Raspberry Pi A receives it back. In case the Raspberry Pi A loses the connection and the previous values, it can reconnect to the broker and receive a "retained message" with the latest values. To accomplish this, every time Raspberry Pi A publishes a message to the broker, it must set the retain flag to make sure that the broker retains the values.

One more thing that Raspberry Pi A has to do is that in its connection message to the broker, it should include a lastwill message as a retained message with content "offline" to a topic "Status/RaspberryPiA". As soon as it connects to the broker, it should send a retained message to the topic "Status/RaspberryPiA" with content "online". This step will make sure that anyone who is subscribed to the topic "Status/RaspberryPiA " would know the status of the Raspberry Pi A, i.e., whether the Raspberry Pi A is online or offline. If the Raspberry Pi A does a graceful disconnect, it should still send a retained message to "Status/RaspberryPiA " with content "offline".

Raspberry Pi C is also connected to the broker and is subscribed to both topics "lightSensor" and "threshold". Every time it receives a message from the broker from either of the topics "lightSensor" and/or "threshold", it compares the LDR value with the threshold and generates a binary result: if "lightSensor" value >= "threshold" value, then the result is "TurnOn" otherwise "TurnOff". (Note that based on how you connected your LDR and potentiometer to the Raspberry Pi A, the results of the comparison could be opposite. The goal is that when there is more light, then the result of the comparison done by Raspberry Pi C should be "Turnoff", otherwise "TurnOn"). The Raspberry Pi C then compares the result with the previous decision it sent to the broker. If the decision has changed, it publishes the updated decision on the broker implemented on the Laptop # 1 to topic "LightStatus". Note that, just like for Raspberry Pi A, instead of remembering the decision Raspberry Pi C sent to the broker the last time, it can just subscribe to the topic "LightStatus" and set the retain flag every time it sends a new decision to this topic.

One more thing that Raspberry Pi C has to do (just like the Raspberry Pi A) is that in its connection message to the broker, it should include a lastwill message as retained message with content "offline" to the topic "Status/RaspberryPiC". As soon as it connects to the broker, it should send a retained message to the topic "Status/RaspberryPiC " with content "online". This step will make sure that anyone who is subscribed to the topic "Status/RaspberryPiC " should know the status of the Raspberry Pi C whether the Raspberry Pi C is online or offline. If the Raspberry Pi C does a graceful disconnect, it should still send a message to "Status/RaspberryPiC " with content "offline".

Same as RPi A

The Raspberry Pi B should also connect to the broker and subscribe to the topic "LightStatus", "Status/RaspberryPiA ", and "Status/RaspberryPiC". The Raspberry Pi B should also have three LEDs connected to it: LED1, LED2, and LED3. If the Raspberry Pi B receives the message "TurnOff" from topic "LightStatus", it should turn the LED1 off. If the Raspberry Pi B receives the message "TurnOn" from the topic "LightStatus", it should turn LED1 on. LED2 and LED3 will show the status of Raspberry Pi A and Raspberry Pi C. If Raspberry Pi B receives a message of "online" from topic "Status/RaspberryPiA ", it should turn the LED2 on. If Raspberry Pi B receives a message of "offline" from topic

"Status/RaspberryPiA", it should turn the LED2 off. If Raspberry Pi B receives a message of "online" from topic "Status/RaspberryPiC", it should turn the LED3 on and based on the most recent value it has received from "LightStatus", it should turn LED1 on or off. If the topic "LightStatus" does not yet have any value published on it, then Raspberry Pi B won't get any message from this topic. In that case the LED1 should stay off. If Raspberry Pi B receives a message of "offline" from topic "Status/RaspberryPiC", it should turn the LED1 and LED3 off.

Finally, your laptop # 2 or smartphone should be subscribed to all these topics: "lightSensor", threshold", "LightStatus", "Status/RaspberryPiA ", and "Status/RaspberryPiC " and should display the messages sent by the broker on these topics along with the timestamps. You will further keep a record on laptop # 2/smartphone when the LED1 was turned on and when it was turned off. For all messages, use QoS 2 (i.e., the highest possible quality of service).

You can use any programming language/environment you like.

Demo

The demos will be conducted in the week after the submission deadline. The TA will post time-slots for demo soon. Schedule a time-slot for the demo as soon as the TA posts them.

To do the demo, bring your two laptops and three Raspberry Pis. Before starting the demo, you will set up such that your laptop # 1 is running the broker. The Raspberry Pi B should be subscribed to all these topics. LED1, LED2, and LED3 should all be off. At this point, you will perform the following 7 steps as the TA asks:

1. The TA will ask you to connect Raspberry Pi A to broker. As soon as that happens, the LED2 connected to Raspberry Pi B should turn on. The TA will then ask you to send a graceful disconnect message to broker. As soon as that happens the LED2 should turn off. [**10**% of total grade]

2. The TA will ask you to connect Raspberry Pi A to broker again. As soon as that happens, the LED2 connected to Raspberry Pi B should turn on. The TA will then ask you to disconnect the Raspberry Pi A from the internet. This mimics an ungraceful disconnect. The broker should detect this ungraceful disconnect as soon as possible and publish the lastwill retained message (with content "offline") to topic "Status/RaspberryPiA ". This message will be received by Raspberry Pi B and it should turn the LED2 off again. [**15**% of total grade]

3. The TA will ask you to connect Raspberry Pi C to broker. As soon as that happens, the LED3 connected to Raspberry Pi B should turn on. The TA will then ask you to do an ungraceful disconnect by turning the Raspberry Pi C off. The broker should detect the ungraceful disconnect as soon as possible and publish the lastwill retained message (with content "offline") to topic "Status/RaspberryPiC". This message will be received by Raspberry Pi B and it should turn the LED3 off. [**10**% of total grade]

4. The TA will ask you to connect Raspberry Pi A and Raspberry Pi C to broker, and the LED2 and LED3 should turn on. Then he will ask you to put a finger on the LDR (to mimic a dark environment for the LDR). The LED1 connected to the Raspberry Pi B should turn on. The TA will then ask you to lift the finger off the LDR (to mimic bright environment) and the LED1 should turn off. [**15**% of total grade]

5. The TA will ask you to create a shadow over LDR until the LED1 turns on. He will then ask you to change the potentiometer value, while keeping your hand at the same position over at LDR where the LED1 had turned on. The change in potentiometer is essentially changing the threshold for the intensity of light at which the LED1 turns on. By changing the value of the potentiometer while keeping your hand at the same location, your should be able to turn the LED1 off. This will demonstrate that the potentiometer is publishing messages correctly and are being received by the Raspberry Pi C and Raspberry Pi B correctly. [**15**% of total grade]

6. The TA will look at the log file in laptop # 2/smartphone (you will share your screen on Zoom to demonstrate this). The log file should display all the messages sent by the broker in plain text. Note that we have implemented publishing logic such that duplicate messages never get published. If the TA sees a lot of duplicate messages in the log, that will result in negative grade. Note that you must NOT implement a code in laptop # 2/smart phone to remove duplicates. The duplicates should never be sent by the broker at the first place. Each set of 3 duplicates will result in 20% reduction in grade each for this step. As an example, if you have four duplicate messages, then it will count as one set of 3 duplicates, and result in 20% reduction of grade for this step. But if you have six duplicate messages, then it will count as two sets of 3 duplicates and result in 40%

reduction of grade for this step. If you think that a duplicate message is justified, then in the log file, your code should automatically mention a reason for getting that duplicate. [**15**% of total grade]

7. In a separate window, in laptop # 2/smart phone, display the timestamps when the LED1 was turned on and off ONLY during the demo. These time stamps should match the actual times when the LED1 was turned on and off. [**10**% of total grade]

What to submit

Please make only 1 submission per group as assigned by the instructor, i.e., only one student in the group should submit. In a single .zip file, upload the following to Moodle [**10**% of total grade]

1. Code for laptop 1 and a detailed readme file explaining how to execute the code
2. Code for laptop 2/smartphone and a detailed readme file explaining how to execute the code
3. Code for all three Raspberry Pis and a detailed readme file explaining how to execute each code
4. A document containing following information
   a) Names of each team member and percentage contribution of each team member. Please be fair and truthful. The grade of this assignment will be divided among team members according to their percentage contributions.
   b) What exactly did each team member do. Make a table. Assuming you have 5 members in your team, make a table with 6 columns. The first column should state various subtasks that your team undertook for this assignment and the next four columns should state the percentage contribution of the five team members for each subtask.
   c) Schematics diagram of connection of LEDs to Raspberry Pi B
   d) Schematics diagram of connection of LDR and potentiometer to Raspberry Pi A
   e) Describe all your design choices such as which MQTT broker did you implement along with a step by step instruction to install it on a laptop/computer, what frequency did you sample the ADC at, how did you scale values from potentiometer/LDR before posting to their corresponding topics so that they could be compared by Raspberry Pi C, what was the range of raw values (min and max) that your ADC got from the LDR, what are the range of raw values that your ADC got from potentiometer (min and max), what are the range of scaled values (min and max) that resulted after you scaled the values from the potentiometer and/or LDR, etc. The text for this part should be no more than 1 page, double column format, 10pt font size.

Food for Thought

Notice that if you were to implement such a system for every electrical/electronic device in your home, you could keep a track of when each device was turned on and off. Furthermore, for most electrical devices, we know their average power consumption. So this way, on your laptop or smart phone, you could do some data analysis and determine which device used how much power each day. This analysis, could help you identify what sources are contributing the most towards the power consumption in your house, and consequently, taking the most money out of your pocket. If we further extend it to a smart building, the power savings could be made significant. If we go a step further, we could apply some machine learning techniques and make the process of identifying the high power consuming devices automatic, and appropriately reducing their use without requiring any intervention from a human user. If you like, you can buy some wirelessly controllable power strips. You can implement a broker on a laptop in your home, or use an online service. Connect all your appliances to sockets in these smart power strips and program the sockets to send update messages to the broker, which can then keep a track of when different appliances were turned on and off. By putting in some effort, you can also manually record the power consumption of each device (it's usually written on each device in the units of kWh) and then calculate how much power each device used/is using in your home. The sockets on these power strips can be individually controlled using a smart phone app (that often comes with these smart power strips). Note that everything described in THIS paragraph is not a part of this assignment.