Department of Electronic & Telecommunications Engineering

University of Moratuwa

EN3271 Internet of Things Systems Engineering

Course Project

2018 Batch **Semester 6**

Overview

This assignment includes 2 tasks as described below.

Task 1: To develop an MQTT Publisher using a Python script. The script will publish a set of virtual sensor information to an MQTT Broker. The set of sensors will represent a real IoT environment. The necessary background related to Python will be provided to you. **(Submission deadline: 19th August 2022 01:00 PM)**

Task 2: To obtain data pertaining to a given IoT environment from the MQTT Broker, (different from the one created by the group) and design and develop a NodeRED dashboard to visualize the data in the most effective manner. **(Submission deadline: 25th August 2022)**

Both tasks are group assignments. After Task 1 is completed, the submitted Python scripts will be evaluated and implemented to generate live data representing the relevant environment by the instructors. For Task 2, each group will be assigned a different environment to work with.

Task 1 – Virtual Sensor Environment

In this task you will design and emulate an IOT sensor network which will publish data to a Broker periodically or when an event occurs. Your group will be assigned one environment from the Table below for this task.

Environment	Sensors
Home Environment	Gas Sensors Smoke Sensors Security System Status Temperature sensor Humidity sensor Motion Sensors Power consumption
Manufacturing Environment	CO2 Sensors Smoke Sensors Pressure sensor Occupancy Detectors Production counting data Level sensors (Liquid) Generator Fuel capacity

Environment	Sensors
Solar Power Plant	Power Generation (kW) Temperature sensor Humidity sensor Wind Speed Wind Direction Illumination sensor (Detection of sunlight intensity) Battery Capacity

You must implement a Python script which will automatically carry out the following tasks.

- Connecting to an MQTT broker **(Server and other important information is provided in the Annex to this document)**
- Virtually creating 10 sensors for the given environment (Data generation must be randomized)
- Publishing virtually created data to relevant MQTT topics. The publishing interval for a sensor should **not be less than 10 seconds**
- Python script should be publishing values to the MQTT broker **continuously until the script is stopped**.

Virtually created sensors should be running on separate threads on Python. For 10 Sensors, 10 threads should be running simultaneously, and each sensor must publish data according to the sensor interval *T* defined by you. Sensors from the above list can be used multiple times, however, **each sensor should be used at least once. There should be at least one sensor that publishes data upon the occurrence of a particular event** (I.e., not periodically)

The data from each sensor should be in a practical value range (e.g., Temperature sensing in the range 20^{0}C – 40^{0}C). Python's random function can be used to generate data for each sensor. The following formula can be used to generate practical values for sensors without causing sudden, large changes in the dataset. (Given ratios can be changed according to the need)

Sensor value at time nT = 0.8 x Sensor value at time (n-1) T + 0.2 x random value at time nT

Sample MQTT topic for designed sensor network.

```
/<unique_group_id>/<SensorTypeXX>
```

- XX will be the sensor number in case multiple sensors are being used
- Group ID will be given to you.
- Sensor type should be abbreviated in a meaningful way (see example below)

Example:

Temperature sensor 1	**/Group12Y/temp01**
Humidity sensor 1	**/Group12Y/hum01**

Important points on Python Script

1. Scalability and reusability will be assessed. Script's parameter such as broker information, topic name, adding new sensors to the environment must be easily scalable.
2. Script should always publish sensor values from all the sensors according to the parameters set on initialization, until the script is killed or stopped. This should be verified running for few hours and analyzing data coming from the MQTT broker.
3. Sensors defined in the virtual environment must be publishing practical values as they would perform in real world environment. Unnecessary floating points values must be rounded off.
4. Sensor(s) which is/are publishing at random interval, and other sensor data publishing must be verified before the submission.

Submission for Task 1

You are required to submit the following to the Moodle:

1. Your fully tested Python script. **Please name this `group_ID_script.py`**
2. A document containing the following information about your IoT environment. **Please name the PDF document as `group_ID_legend.pdf`**

Sensor ID	Sensor Name/Type	Publishing Range (if defined)	Publishing Interval	Any other Remarks
Temp01	Temperature Sensor	20 – 40 (Celsius)	30 seconds	Assumed as Living room temperature sensor etc.

(See next Page)

Task 2 – IOT Dashboard

In this task, you will design and deploy a NodeRED dashboard for a live IoT Environment assigned to you. Your group will be given the relevant topic name where you should analyze the messages under it. First, you must identify the sensors under the given IoT environment and use those findings for the dashboard design. Your design should include the most appropriate gauges, graphs, or visualization tools for each sensor type defined under the environment.

Note: NodeRED should be deployed in a Raspberry Pi board.

Submission for Task 2

The NodeRED dashboard should be exported into JSON format and submitted to the Moodle **named a <group_ID_NodeRed.json>**

Evaluation

This will be evaluated by the submitted files and through a **demonstration & viva**.

Guidance

For any clarification regarding the assignment tasks or issues regarding the servers, please contact following instructors.

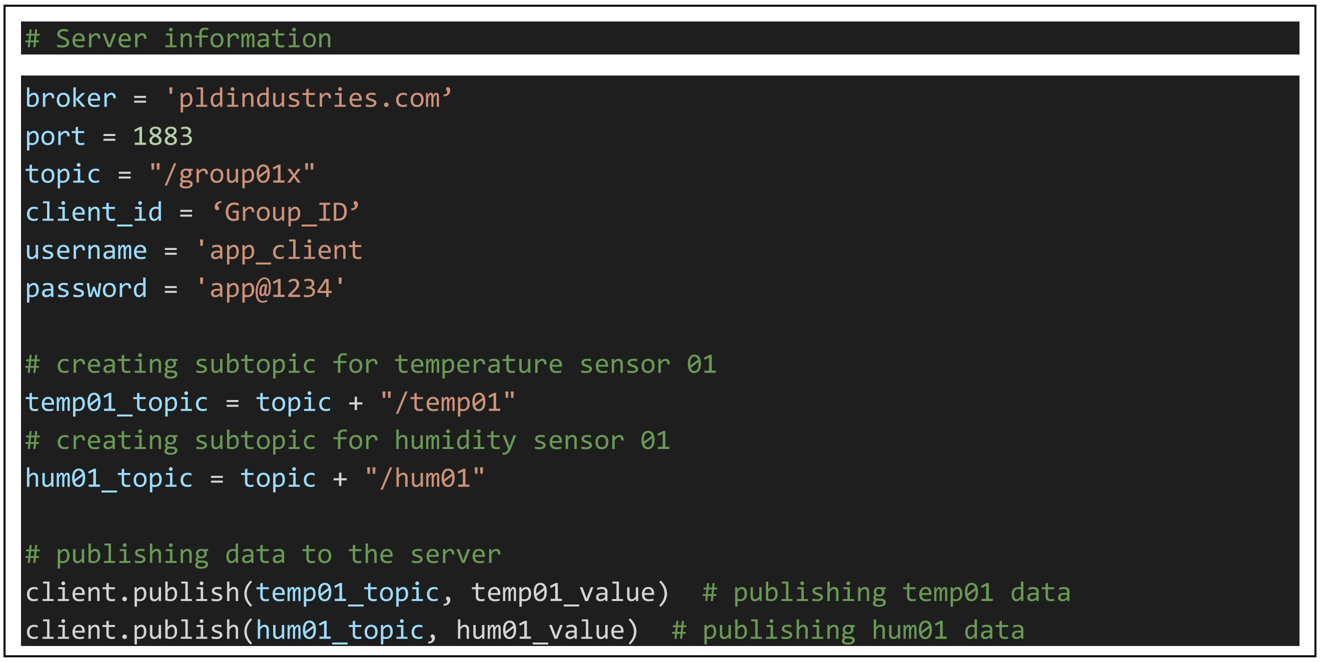
Ranush Wickramarathne (ranushw@uom.lk)
Pasan Dharmasiri (pasanl@uom.lk)

ANNEX

MQTT Server Information

Refer the code snippet available at the end of this document and re-use it as a template for the development of Publisher Script in the task 1. [Mandatory]

Broker	pldindustries.com
Port	1883
Client ID	<group_ID>
Username	app_client
Password	app@1234



```
# Server information

broker = 'pldindustries.com’
port = 1883
topic = "/group01x"
client_id = ‘Group_ID’
username = 'app_client
password = 'app@1234'

# creating subtopic for temperature sensor 01
temp01_topic = topic + "/temp01"
# creating subtopic for humidity sensor 01
hum01_topic = topic + "/hum01"

# publishing data to the server
client.publish(temp01_topic, temp01_value)  # publishing temp01 data
client.publish(hum01_topic, hum01_value)  # publishing hum01 data
```