

Edge Computing Lab
Class: TY-AIEC
School of Computing, MIT Art Design Technology University
Academic Year: 2024-25
Experiment No. 4

Title

Setting Up MQTT Broker on Raspberry Pi and Reading DHT11 Sensor Data and MQ9 Sensor

Objective

This lab aims to teach participants how to set up a Mosquitto MQTT broker on a Raspberry Pi, interface with a DHT11 temperature and humidity sensor, and publish the sensor data to an MQTT topic. Participants will also learn to subscribe to this topic using an MQTT client on an Android device.

Equipment and Materials

- Raspberry Pi (with Raspbian OS installed and internet connection)
- DHT11 Temperature and Humidity Sensor
- Jumper Wires
- Breadboard (optional)
- Android Device with MQTT Client App installed (e.g., MQTT Dash, MQTT Client)

Pre-lab Setup

Ensure the Raspberry Pi is set up with Raspbian OS, connected to the internet, and accessible via SSH or a monitor and keyboard. Install an MQTT client app on the Android device.

Lab Sections

Section 1: Setting Up the MQTT Broker

1. Update and Upgrade Raspberry Pi:
 - Open the terminal and execute ``sudo apt-get update`` and ``sudo apt-get upgrade``.
2. Install Mosquitto MQTT Broker:
 - Run ``sudo apt-get install -y mosquitto mosquitto-clients``.
3. Enable Mosquitto Service:
 - Use ``sudo systemctl enable mosquitto.service``.
4. Start Mosquitto Service:
 - Execute ``sudo systemctl start mosquitto.service``.
5. Test Installation:

- Subscribe to a test topic in one terminal using ``mosquitto_sub -h localhost -t test/topic``.
- Publish a message from another terminal with ``mosquitto_pub -h localhost -t test/topic -m "Hello MQTT"``.
- Confirm "Hello MQTT" message appears in the subscriber terminal.

Section 2: Interfacing with DHT11 Sensor

1. Connect DHT11 to Raspberry Pi:
 - Connect VCC to 5V, Data to GPIO4, and GND to Ground.
2. Install Python Libraries:
 - Install Adafruit_DHT and paho-mqtt using ``sudo pip3 install Adafruit_DHT paho-mqtt``.

Section 3: Publishing Sensor Data to MQTT

1. Write Python Script:
 - Create ``dht11_mqtt_publisher.py`` to read sensor data and publish it to ``home/sensor/dht11`` topic.
2. Run the Script:
 - Execute the script with ``python3 dht11_mqtt_publisher.py``.

Section 4: Subscribing with an Android MQTT Client

1. Configure MQTT Client App:
 - Set the Raspberry Pi's IP as the MQTT server and subscribe to ``home/sensor/dht11``.
2. Receive Data:
 - Observe the temperature and humidity data published from the Raspberry Pi on the Android device.

Section 5: Enhancements and Troubleshooting

- Discuss JSON formatting, security with TLS/SSL, error handling, and MQTT client configuration.
- Troubleshoot common issues related to sensor readings, MQTT connections, and data reception.

Lab Tasks

- Complete the setup and confirm successful data publication and subscription.
- Experiment with different topics and QoS levels.
- (Optional) Implement JSON formatting for sensor data.

Lab Questions

1. How does MQTT protocol ensure message delivery with different QoS levels?
2. What are the benefits of using JSON formatting for sensor data in MQTT messages?
3. How can TLS/SSL encryption enhance the security of MQTT communication?

Further Exploration

- Explore bridging Mosquitto to an external MQTT broker.
- Integrate sensor data into IoT platforms or home automation systems.

Submission Requirements

- Provide screenshots of the terminal showing successful Mosquitto installation, Python script execution, and MQTT messages on the Android client.
- Submit a short report discussing the setup process, encountered challenges, and how they were overcome.

References

- Mosquitto Documentation: <https://mosquitto.org/documentation/>
- Adafruit DHT Sensor Library: https://github.com/adafruit/Adafruit_Python_DHT
- Paho MQTT Client: <https://www.eclipse.org/paho/clients/python/>

- <https://www.youtube.com/watch?v=BFyPzC6No8k&t=1s>

1. **sudo apt update**
2. **sudo apt upgrade**
3. **sudo apt-get install mosquitto mosquitto-clients**
4. **sudo systemctl enable mosquitto.service**
5. **mosquitto_sub -d -t "binary/updates"**
6. **mosquitto_pub -d -t "binary/updates" -m "Hello MITADT" // On 2nd terminal**
7. **sudo nano /etc/mosquitto/mosquitto.conf**
 - a. add line over here as **listener 1883 allow_anonymous true**
8. **sudo boot**
9. **Open Terminal on Desktop PC as "MQTT Explorer"**

1. **sudo git clone https://github.com/binaryupdates/mqttclient.git**
2. **sudo pip3 install paho-mqtt**

Name : Sudip Satish Konde Class : TY (AIEE)
Roll No : 2223118 Sub : ECL

Assignment No 4.

① What is MQTT & why is it commonly used in IoT application.

MQTT is a lightweight message protocol designed for low bandwidth, high latency or unreliable networks, making it ideal for IoT application like smart homes, industrial automation & remote monitoring.

② Explain the role of MQTT broker, publisher & subscriber in an IoT system.

MQTT component :

- Broker : Central server that manage message transmission.
- Publisher : Sends data to a topic.
(e.g. a sensor publishing temperature)
- Subscriber : Receives messages from a topic.

③ How can you install & set up the mosquitto MQTT broker on Rasp Pi ?

Installing mosquitto on Rasp Pi

```
sudo apt update  
sudo apt install -y mosquitto mosquitto-clients  
sudo systemctl enable mosquitto  
sudo systemctl start mosquitto.
```

- ④ Write a python script to publish temp data from a DHT11 sensor to a MQTT topic?

```
import paho.mqtt.client as mqtt  
import Adafruit_DHT
```

```
sensor = Adafruit_DHT.DHT11  
pin = 4
```

```
client = mqtt.Client()  
client.connect("broker.hivemq.com", 1883, 60)
```

```
while True:  
    humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)  
    if temperature:  
        client.publish("iot/temperature",  
                        temperature).
```