Edge Computing Lab

Class: TY-AIEC

School of Computing, MIT Art Design Technology University

Academic Year: 2024-25

Extra Lab - 04

Title

Setting Up MQTT Broker on Raspberry Pi and Reading DHT11 / Air Quality Sensor Data

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
- 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.

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**

Conclusion: - Understood and **Implemented** Mosquitto MQTT broker on a Raspberry Pi, interface with a DHT11 temperature and humidity sensor

FAQ’s:

Name : Deepika Modgi

Batch : TY AIEC C

Roll no : 2223928

Subject : EC Lab.

Experiment 4: MQTT with Raspberry Pi.

Questions.

Q1] What is MQTT and why is it commonly used in IoT applications?

Ans. MQTT is Message Queing Telemetry transport a lightweight publish subscribe messaging protocol.

i] Lightweight : works on Low-bandwidth networks.

ii] Efficient : Uses minimal power, ideal for IoT devices.

iii] Reliable : Supports Qos (Quality of service) levels for message delivery.

iv] Scalable : Can handle multiple devices communicating in real life.

Q2] Explain the role of MQTT broker, publisher and subscriber in an IoT system.

Ans. i] MQTT broker:

Manages message distribution ~~and~~

ex. Eclipse mosquitto,

ii] Publisher (Data sender):

Sends message to an MQTT topic.

Ex. Raspberry Pi publishing sensor data

iii] Subscriber (Data Receiver):

Subscribes to a topic and receives message in real time.

Ex. A mobile app subscribing to sensor data.

Q3] How can you Install and set up the Mosquitto MQTT broker on Raspberry Pi?

Ans: i] Update Raspberry Pi:

```
sudo apt update && suda apt upgrade -y
```

ii] Install Mosquitto & client tools:

```
sudo apt Install mosquitto mosquito-clients -y
```

iii] Enable Mosquitto as a Background Service

```
sudo systemctl enable mosquitto.
sudo systemctl start mosquitto.
```

iv] Test MQTT Broker:

Open Terminal 1:

```
mosquitto_sub -h localhost -t test/topic.
```

Open Terminal 2:

```
mosquitto_pub -h localhost -t test/topic -m "Hello, MQTT!"
```

v] If set up correctly, terminal 1 will display:

Hello, MQTT!

Q4] Write a python Sript to publish temperature data from a DHT11 sensor to an MQTT topic.

Ans.

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

BROKER = "broker.hivemq.com"
TOPI C = "sensor / temperature"
DHT=SENSOR = Adafruit_DHT. DHT11
DHT_PIN = 7.

client = mqtt.client()
client.connect (BROKER, 1883, 60).

while True:
        humidity, temperature = Adafruit_DHT.
                    read (DHT_SENSOR. DHT_PIN).
        if humidity is not None and temperature
                is not None :
            payload = f"Temperature : {temperature}°C
                Humidity : {humidity}%"
            client.publish (TOPIC, payload)
              print (f"Published : {payload}")
            time.sleep(5).
```

Q5.] How can you subscribe to an MQTT topic from another device to receive real time sensor updates?

ii] Use this command on another device :

```
mosquitto_sub -h broker.hivemq.com
                 -t sensor/temperature.
```

iii] Python script:

```
import paho.mqtt.client as mqtt
 BROKER = "broker.hivemq.com"
  TOPIC = "sensor/temperature"

  def on_message (client, userdata, message):
          print (f"Received: {message.payload.
                                 decode()}")

  client=mqtt.Client()
  client.on_message = on_message.
  client.connect (BROKER, 1883, 60)
  client.subscribe (TOPIC).

     Print ("Waiting for messages....")
     client.loop forever()
```

Ans i] Use this command on another device:

```
mosquitto-sub -h broker.hivemq.com
              -t sensor/temperature,
```

ii] Python script:

```
import paho.mqtt.client as mqtt
BROKER = "broker.hivemq.com"
TOPIC = "sensor/temperature"

def on-message (client, userdata, message):
        print (f"Received: {message.payload.
                              decode()}")
client = mqtt.Client()
client.on_message = on_message.
client.connect (BROKER, 1883, 60)
client.subscribe (TOPIC).

print ("Waiting for messages....")
client.loop forever()
```