Experiment 5: Raspberry Pi MQTT with Sense HAT Emulator and LED Control

Internet of Things (IoT) – Architecture, Communication Technology, and Applications

---

Objective

This experiment aims to demonstrate the integration of **Raspberry Pi 4**, the **Sense HAT Emulator**, and **MQTT communication** via the **HiveMQ broker**.

Part 1: Publishing and Subscribing to Sensor Data

- Publishing **temperature and humidity data** from the Sense HAT Emulator to an MQTT broker.
- Subscribing to receive the published data using Mosquitto.
- Implementing MQTT communication using **C programming language**.

Part 2: Controlling an LED via MQTT

- Subscribing to an MQTT topic (`LED/control/yourname`) to receive **"on"** and **"off"** commands.
- Controlling an LED using MQTT messages from the **myMQTT** mobile app.

Materials and Requirements

Hardware

- Raspberry Pi 4 (or any Raspberry Pi with Internet access)
- MicroSD card with Raspberry Pi OS installed
- Keyboard, mouse, and display (or SSH access)
- LED + 330Ω resistor
- Breadboard and jumper wires

Software

- Raspberry Pi OS (Buster/Bullseye)
- **Programming Language:** C
- **MQTT Broker:** HiveMQ
- **MQTT Client:** myMQTT (Mobile App)
- **Required Libraries:**
  - `paho-mqtt` (for MQTT communication)
  - `sense-emu` (for Sense HAT Emulator)
  - `mosquitto-clients` (for testing MQTT communication)
  - `pigpio` (for LED control)

Part 1: Publishing and Subscribing to Sensor Data

Step 1: Install Sense HAT Emulator

Installation Commands

bash
CopyEdit

```
sudo apt update
sudo apt install -y sense-emu python3-sense-emu
```

Start the Sense HAT Emulator GUI

bash
CopyEdit

```
sense_emu_gui &
```

Step 2: Compile Paho MQTT C Library from Source

bash
CopyEdit

```
sudo apt install -y cmake gcc g++ make git
git clone https://github.com/eclipse/paho.mqtt.c.git
```

```
cd paho.mqtt.c
cmake -Bbuild -H.
sudo cmake --build build/ --target install
sudo ldconfig
```

Verify Installation

bash
CopyEdit

```
ls /usr/local/include/MQTTClient.h
```

Step 3: Install MQTT Dependencies for Testing

bash
CopyEdit

```
sudo apt install -y mosquitto-clients
```

Verify Installation

bash
CopyEdit

```
mosquitto_pub --help
mosquitto_sub --help
```

Step 4: Write a Python Script to Read Sense HAT Data

Create and Edit the Python Script

bash
CopyEdit

```
nano sensehat_reader.py
```

Python Code

python
CopyEdit

```
#!/usr/bin/env python3
```

```
from sense_emu import SenseHat

sense = SenseHat()
temp = sense.get_temperature()
humidity = sense.get_humidity()

print(f"{temp:.2f},{humidity:.2f}")  # Outputs "temperature,humidity"
```

Make the Script Executable

bash
CopyEdit

```
chmod +x sensehat_reader.py
```

Step 5: Write a C Program to Publish Sense HAT Data via MQTT

Create and Edit the C File

bash
CopyEdit

```
nano publisher.c
```

C Code

c
CopyEdit

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <MQTTClient.h>

#define ADDRESS "tcp://broker.hivemq.com:1883"
#define CLIENTID "RaspiPublisher"
#define QOS 1
#define TIMEOUT 10000
```

```
#define SENSE_TEMP_TOPIC "Sense/temperature/yourname"
#define SENSE_HUMID_TOPIC "Sense/humidity/yourname"

void get_sensehat_data(float *temperature, float *humidity) {
    FILE *fp;
    char buffer[50];

    fp = popen("python3 sensehat_reader.py", "r");
    if (fp == NULL) {
        fprintf(stderr, "Error: Could not run Sense HAT script\n");
        *temperature = -1;
        *humidity = -1;
        return;
    }

    if (fgets(buffer, sizeof(buffer), fp) != NULL) {
        sscanf(buffer, "%f,%f", temperature, humidity);
    }

    pclose(fp);
}

void publish(MQTTClient client, char *topic, char *payload) {
    MQTTClient_message pubmsg = MQTTClient_message_initializer;
    MQTTClient_deliveryToken token;

    pubmsg.payload = payload;
    pubmsg.payloadlen = strlen(payload);
    pubmsg.qos = QOS;
    pubmsg.retained = 0;

    MQTTClient_publishMessage(client, topic, &pubmsg, &token);
    MQTTClient_waitForCompletion(client, token, TIMEOUT);
}

int main() {
    MQTTClient client;
```

```
    MQTTClient_connectOptions conn_opts =
MQTTClient_connectOptions_initializer;

    MQTTClient_create(&client, ADDRESS, CLIENTID,
MQTTCLIENT_PERSISTENCE_NONE, NULL);
    MQTTClient_connect(client, &conn_opts);

    while (1) {
        float temp, humidity;
        char temp_str[10], humid_str[10];

        get_sensehat_data(&temp, &humidity);

        sprintf(temp_str, "%.2f", temp);
        sprintf(humid_str, "%.2f", humidity);

        publish(client, SENSE_TEMP_TOPIC, temp_str);
        publish(client, SENSE_HUMID_TOPIC, humid_str);

        sleep(5);
    }

    MQTTClient_disconnect(client, TIMEOUT);
    MQTTClient_destroy(&client);
    return 0;
}
```

Compile and Run the C Program

bash
CopyEdit

```
gcc -o publisher publisher.c -I/usr/local/include -L/usr/local/lib
-lpaho-mqtt3c
./publisher
```

Part 2: Controlling an LED via MQTT

Step 1: Circuit Setup

- **GPIO 4 → LED Anode**
- **GND → LED Cathode via 330Ω resistor**

Step 2: Write the C Program for LED Control

bash
CopyEdit

```
nano led_control.c
```

C Code

c
CopyEdit

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <pigpio.h>
#include <MQTTClient.h>

#define ADDRESS "tcp://broker.hivemq.com:1883"
#define CLIENTID "LedSubscriber"
#define LED_TOPIC "LED/control/yourname"
#define LED_PIN 4

void messageArrived(void *context, char *topicName, int topicLen,
MQTTClient_message *message) {
    char payload[message->payloadlen + 1];
    memcpy(payload, message->payload, message->payloadlen);
    payload[message->payloadlen] = '\0';

    if (strcmp(payload, "on") == 0) {
        gpioWrite(LED_PIN, 1);
        printf("LED Turned ON\n");
    } else if (strcmp(payload, "off") == 0) {
        gpioWrite(LED_PIN, 0);
        printf("LED Turned OFF\n");
    }
```

```
    MQTTClient_freeMessage(&message);
    MQTTClient_free(topicName);
}

int main() {
    gpioInitialise();
    gpioSetMode(LED_PIN, PI_OUTPUT);

    MQTTClient client;
    MQTTClient_create(&client, ADDRESS, CLIENTID,
MQTTCLIENT_PERSISTENCE_NONE, NULL);
    MQTTClient_setCallbacks(client, NULL, NULL, messageArrived, NULL);
    MQTTClient_connect(client,
&MQTTClient_connectOptions_initializer);
    MQTTClient_subscribe(client, LED_TOPIC, 1);

    while (1) {
        MQTTClient_yield();
    }
}
```

Compile and Run the LED Control Program

bash
CopyEdit

```
gcc -o led_control led_control.c -lpaho-mqtt3c -lpigpio
sudo ./led_control
```

Control the LED from myMQTT App

- **Topic:** `LED/control/yourname`
- **Messages:** `"on"` to turn on, `"off"` to turn off.

Conclusion

This experiment successfully demonstrates **MQTT communication**, **Sense HAT data publishing**, and **LED control** via a **Raspberry Pi and myMQTT app**.