



*Download the PDF for your future use*

# *Just 4 Simple Steps*

Build your own IoT device  
using

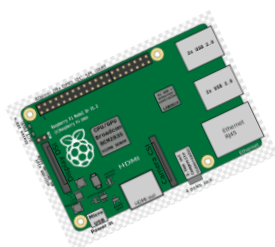
Raspberry Pi and MQTT

- Melvin Francis



**HIVEMQ**  
MQTT Broker

*IoT Device - Publisher*



*IoT Dashboard - Subscriber*



## Step - 1

Install the following dependencies:

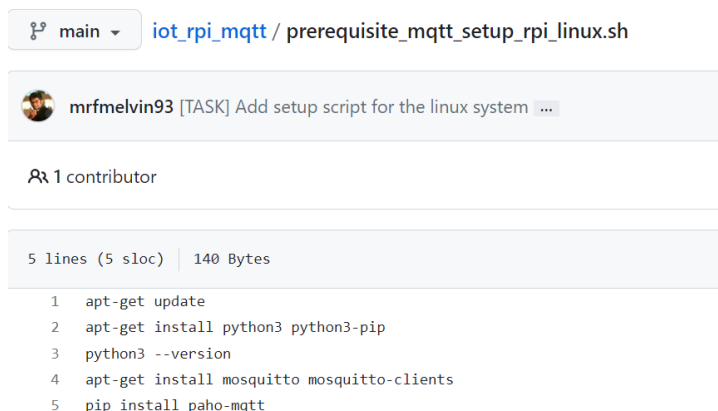
```
> sudo apt update

> sudo apt install python3 python3-pip

> python3 --version

> sudo apt-get install mosquitto mosquitto-clients

> pip install paho-mqtt
```



The screenshot shows a GitHub repository interface. At the top, there's a breadcrumb trail: "main" (with a dropdown arrow) followed by "iot\_rpi\_mqtt / prerequisite\_mqtt\_setup\_rpi\_linux.sh". Below this, the repository name "mrfmelvin93" is shown with a profile picture, followed by the commit message "[TASK] Add setup script for the linux system" and a truncated description "...". It indicates "1 contributor". The file details show "5 lines (5 sloc)" and "140 Bytes". The script content is displayed as a list of five lines of code:

```
1 apt-get update
2 apt-get install python3 python3-pip
3 python3 --version
4 apt-get install mosquitto mosquitto-clients
5 pip install paho-mqtt
```

[https://github.com/mrfmelvin93/iot\\_rpi\\_mqtt/blob/main/prerequisite\\_mqtt\\_setup\\_rpi\\_linux.sh](https://github.com/mrfmelvin93/iot_rpi_mqtt/blob/main/prerequisite_mqtt_setup_rpi_linux.sh)

## Step - 2

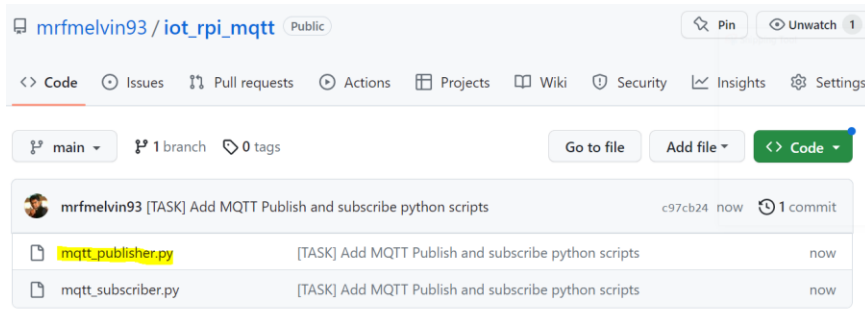
### Write the MQTT publisher python script for RPi

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

# create a client instance
client = mqtt.Client()

# connect to the broker
client.connect("broker.hivemq.com", 1883, 60)

# loop forever and publish sensor data
while True:
    temperature = random.randint(0, 100) # TODO: Replace it with your Sensor value
    humidity = random.randint(0, 100) # TODO: Replace it with your Sensor value
    client.publish("melvin/mqttExample/temperature", temperature)
    client.publish("melvin/mqttExample/humidity", humidity)
    print("Publishing-->melvin/mqttExample/temperature: ", temperature)
    print("Publishing-->melvin/mqttExamplehumidity: ", humidity)
    time.sleep(1)
```



[https://github.com/mrfrmelvin93/iot\\_rpi\\_mqtt/blob/main/mqtt\\_publisher.py](https://github.com/mrfrmelvin93/iot_rpi_mqtt/blob/main/mqtt_publisher.py)

## Step - 3

Write the MQTT subscriber python script to view the data

```
import paho.mqtt.client as mqtt

# create a client instance
client = mqtt.Client()


# define a callback for the message received event
def on_message(client, userdata, message):
    print("Subscribing-->" + message.topic, message.payload)


# connect to the broker
client.connect("broker.hivemq.com", 1883, 60)


# subscribe to the sensor data topics
client.subscribe("melvin/mqttExample/temperature")
client.subscribe("melvin/mqttExample/humidity")

# set the callback for the message received event
client.on_message = on_message

# loop forever and wait for messages
client.loop_forever()
```


 main ▾



 1 branch


 0 tags


Go to file

Add file ▾

 Code ▾

 **mrfmelvin93** [TASK] Add MQTT Publish and subscribe python scripts c97cb24 1 minute ago  1 commit

 mqtt\_publisher.py [TASK] Add MQTT Publish and subscribe python scripts 1 minute ago

 **mqtt\_subscriber.py** [TASK] Add MQTT Publish and subscribe python scripts 1 minute ago

[https://github.com/mrfmelvin93/iot\\_rpi\\_mqtt/blob/main/mqtt\\_subscriber.py](https://github.com/mrfmelvin93/iot_rpi_mqtt/blob/main/mqtt_subscriber.py)

## Step - 4

Run the Publisher on Rpi and the Subscriber py scripts

```
python3 mqtt_publisher.py
```

```
root@ceb20443c3:/mnt/c/Users/MelvinFrancis/OneDrive/Objects/Business/LinkedIn# python3 mqtt_publisher.py
Publishing-->melvin/mqttExample/temperature: 26
Publishing-->melvin/mqttExample/humidity: 30
Publishing-->melvin/mqttExample/temperature: 4
Publishing-->melvin/mqttExample/humidity: 82
Publishing-->melvin/mqttExample/temperature: 13
Publishing-->melvin/mqttExample/humidity: 37
Publishing-->melvin/mqttExample/temperature: 93
Publishing-->melvin/mqttExample/humidity: 41
Publishing-->melvin/mqttExample/temperature: 38
Publishing-->melvin/mqttExample/humidity: 25
Publishing-->melvin/mqttExample/temperature: 62
Publishing-->melvin/mqttExample/humidity: 23
Publishing-->melvin/mqttExample/temperature: 73
Publishing-->melvin/mqttExample/humidity: 52
Publishing-->melvin/mqttExample/temperature: 27
Publishing-->melvin/mqttExample/humidity: 49
Publishing-->melvin/mqttExample/temperature: 86
Publishing-->melvin/mqttExample/humidity: 45
Publishing-->melvin/mqttExample/temperature: 66
Publishing-->melvin/mqttExample/humidity: 86
Publishing-->melvin/mqttExample/temperature: 7
```

```
python3 mqtt_subscriber.py
```

```
Subscribing-->melvin/mqttExample/humidity b'30'
Subscribing-->melvin/mqttExample/temperature b'26'
Subscribing-->melvin/mqttExample/temperature b'4'
Subscribing-->melvin/mqttExample/humidity b'82'
Subscribing-->melvin/mqttExample/temperature b'13'
Subscribing-->melvin/mqttExample/humidity b'37'
Subscribing-->melvin/mqttExample/temperature b'93'
Subscribing-->melvin/mqttExample/humidity b'41'
Subscribing-->melvin/mqttExample/temperature b'38'
Subscribing-->melvin/mqttExample/humidity b'25'
Subscribing-->melvin/mqttExample/temperature b'62'
Subscribing-->melvin/mqttExample/humidity b'23'
Subscribing-->melvin/mqttExample/temperature b'73'
Subscribing-->melvin/mqttExample/humidity b'52'
Subscribing-->melvin/mqttExample/temperature b'27'
Subscribing-->melvin/mqttExample/humidity b'49'
Subscribing-->melvin/mqttExample/temperature b'86'
Subscribing-->melvin/mqttExample/humidity b'45'
Subscribing-->melvin/mqttExample/temperature b'66'
Subscribing-->melvin/mqttExample/humidity b'86'
Subscribing-->melvin/mqttExample/temperature b'7'
```



More tutorials on  
custom dashboard  
creation in my  
upcoming posts

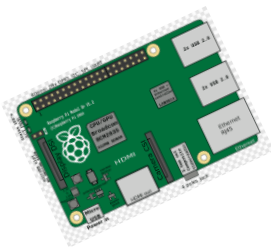


Don't forget to follow me on  
<https://www.linkedin.com/in/melvin-francis/>



**HIVEMQ**  
MQTT Broker

*IoT Device - Publisher*



*IoT Dashboard - Subscriber*

