

Practical No. 9

Aim: Setup a Mosquitto MQTT server and client and write a Python script to communicate data between Pi's.

Step 1: Install MQTT server and clients packages

```
pi@raspberrypi:~$ sudo apt-get install -y mosquitto mosquitto-clients
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libev4 libmosquitto1 libwebsockets8
The following NEW packages will be installed:
  libev4 libmosquitto1 libwebsockets8 mosquitto mosquitto-clients
0 upgraded, 5 newly installed, 0 to remove and 205 not upgraded.
Need to get 387 kB of archives.
After this operation, 843 kB of additional disk space will be used.
Get:1 http://mirror.ossplanet.net/raspbian/raspbian buster/main armhf libev4 armhf 1:4.25-1 [34.5 kB]
Get:2 http://mirror.ossplanet.net/raspbian/raspbian buster/main armhf libmosquitto1 armhf 1.5.7-1 [57.4 kB]
Get:3 http://mirror.ossplanet.net/raspbian/raspbian buster/main armhf libwebsockets8 armhf 2.0.3-3 [85.6 kB]
Get:4 http://mirror.ossplanet.net/raspbian/raspbian buster/main armhf mosquitto armhf 1.5.7-1 [143 kB]
Get:5 http://mirror.ossplanet.net/raspbian/raspbian buster/main armhf mosquitto-clients armhf 1.5.7-1 [66.7 kB]
Fetched 387 kB in 4s (89.3 kB/s)
Selecting previously unselected package libev4:armhf.
(Reading database ... 152993 files and directories currently installed.)
Preparing to unpack .../libev4_1%3a4.25-1_armhf.deb ...
Unpacking libev4:armhf (1:4.25-1) ...
Selecting previously unselected package libmosquitto1:armhf.
Preparing to unpack .../libmosquitto1_1.5.7-1_armhf.deb ...
Unpacking libmosquitto1:armhf (1.5.7-1) ...
Selecting previously unselected package libwebsockets8:armhf.
Preparing to unpack .../libwebsockets8_2.0.3-3_armhf.deb ...
Unpacking libwebsockets8:armhf (2.0.3-3) ...
Selecting previously unselected package mosquitto.
Preparing to unpack .../mosquitto_1.5.7-1_armhf.deb ...
Unpacking mosquitto (1.5.7-1) ...
Selecting previously unselected package mosquitto-clients.
Preparing to unpack .../mosquitto-clients_1.5.7-1_armhf.deb ...
Unpacking mosquitto-clients (1.5.7-1) ...
Setting up libmosquitto1:armhf (1.5.7-1) ...
Setting up libev4:armhf (1:4.25-1) ...
Setting up mosquitto-clients (1.5.7-1) ...
Setting up libwebsockets8:armhf (2.0.3-3) ...
Setting up mosquitto (1.5.7-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mosquitto.service → /lib/systemd/system/mosquitto.service.
Processing triggers for systemd (241-5+rp11) ...
Processing triggers for man-db (2.8.5-2) ...
Processing triggers for libc-bin (2.28-10+rp11) ...
```

Step 2: Install paho package of MQTT

```
pi@raspberrypi:~/Downloads$ sudo pip install paho-mqtt
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting paho-mqtt
  Downloading https://files.pythonhosted.org/packages/25/63/db25e62979c2a716a74950c9ed658dce431b5cb0fde29eb8c9a9489a904/paho-mqtt-1.4.0.tar.gz (88kB)
    100% |#####| 92kB 1.0MB/s
Building wheels for collected packages: paho-mqtt
  Running setup.py bdist_wheel for paho-mqtt ... done
  Stored in directory: /root/.cache/pip/wheels/82/e5/de/d90d0f397648a1b58ffea1b5742ac8c77f1f43b550fa5a5
Successfully built paho-mqtt
Installing collected packages: paho-mqtt
Successfully installed paho-mqtt-1.4.0
pi@raspberrypi:~/Downloads$
```

Step 3: set the server IP address in the python program of server and client

```
mqtt_publisher.py - I:\pract9\mqtt_publisher.py (3.7.3)
File Edit Format Run Options Window Help
import paho.mqtt.publish as publish
MQTT_SERVER = "192.168.0.73"
MQTT_PATH = "test_channel"
publish.single(MQTT_PATH, "pradnyesh", hostname=MQTT_SERVER)
```

```
mqtt_subscriber.py - I:\pract9\mqtt_subscriber.py (3.7.3)
File Edit Format Run Options Window Help
import paho.mqtt.client as mqtt

MQTT_SERVER = "192.168.0.73"
MQTT_PATH = "test_channel"

# The callback for when the client receives a CONNACK response from the server.
def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))

    # Subscribing in on_connect() means that if we lose the connection and
    # reconnect then subscriptions will be renewed.
    client.subscribe(MQTT_PATH)

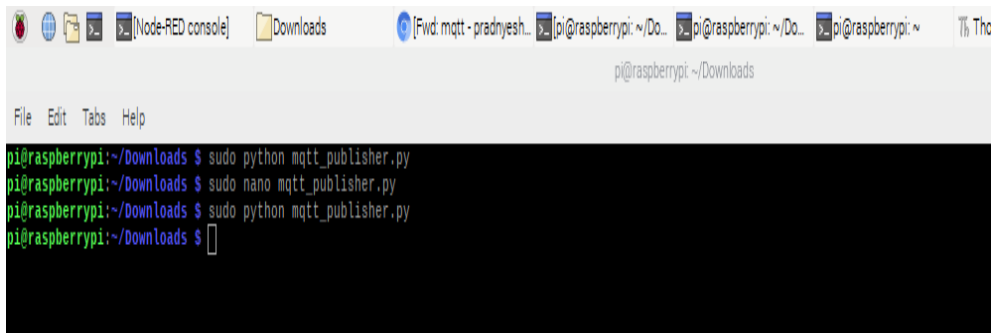
# The callback for when a PUBLISH message is received from the server.
def on_message(client, userdata, msg):
    print(msg.topic+" "+str(msg.payload))
    # more callbacks, etc

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

client.connect(MQTT_SERVER, 1883, 60)

# Blocking call that processes network traffic, dispatches callbacks and
# handles reconnecting.
# Other loop*() functions are available that give a threaded interface and a
# manual interface.
client.loop_forever()
```

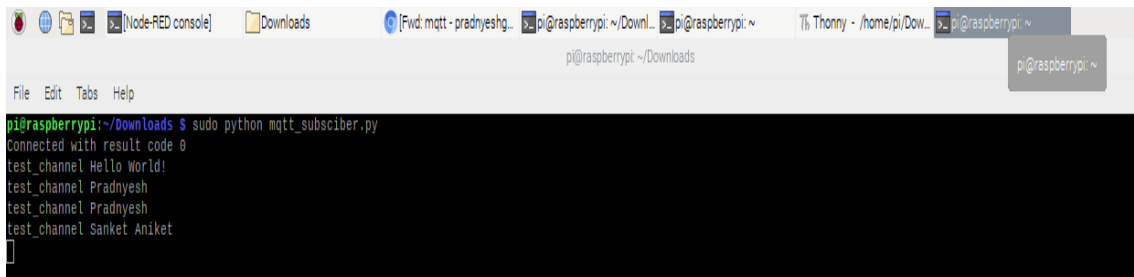
Step 4: Run the server program.



A terminal window on a Raspberry Pi showing the execution of the mqtt_publisher.py script. The window title is 'pi@raspberrypi: ~/Downloads'. The command prompt is 'pi@raspberrypi:~/Downloads \$'. The user enters 'sudo python mqtt_publisher.py', followed by 'sudo nano mqtt_publisher.py', and then 'sudo python mqtt_publisher.py' again. The prompt returns to '\$'.

```
pi@raspberrypi:~/Downloads $ sudo python mqtt_publisher.py
pi@raspberrypi:~/Downloads $ sudo nano mqtt_publisher.py
pi@raspberrypi:~/Downloads $ sudo python mqtt_publisher.py
pi@raspberrypi:~/Downloads $
```

Step 5: Run client script and send message to server



A terminal window on a Raspberry Pi showing the execution of the mqtt_subscriber.py script. The window title is 'pi@raspberrypi: ~/Downloads'. The command prompt is 'pi@raspberrypi:~/Downloads \$'. The user enters 'sudo python mqtt_subscriber.py'. The output shows 'Connected with result code 0', followed by three messages from 'test_channel': 'Hello World!', 'Pradnyesh', and 'Sanket Aniket'. The prompt returns to '\$'.

```
pi@raspberrypi:~/Downloads $ sudo python mqtt_subscriber.py
Connected with result code 0
test_channel Hello World!
test_channel Pradnyesh
test_channel Pradnyesh
test_channel Sanket Aniket
pi@raspberrypi:~/Downloads $
```