

MQTT Setup in Jetson Nano

OBJECTIVE

1. Setup the MQTT Broker and Client
2. Setup IoTMQTTPanel App
3. Create a python program for MQTT

OVERVIEW

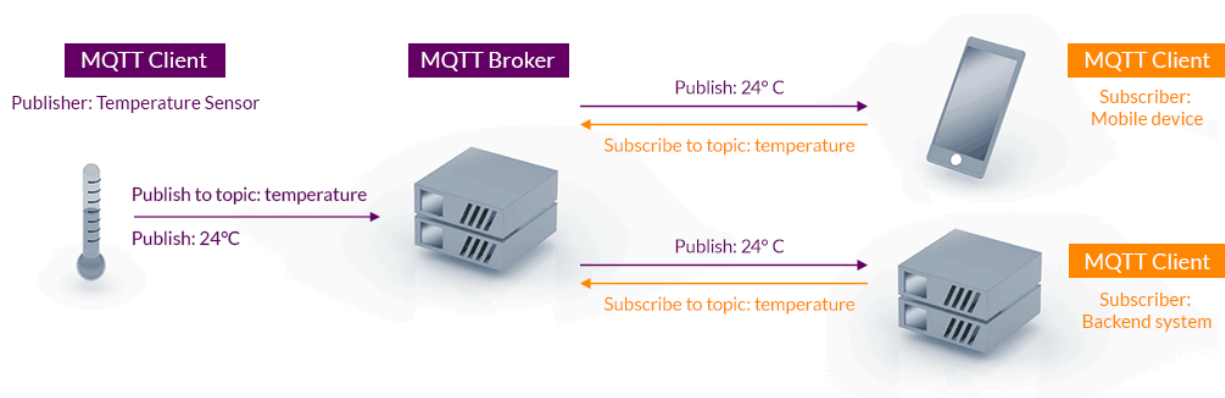
In this test, we are setting up the MQTT broker that is Mosquitto in the Jetson Nano. Also going to create python code to transfer data over MQTT.

Why MQTT?

MQTT is a standard message protocol for the Internet of Things(IoT). It is designed as extremely lightweight publish/subscribe message transport that connects remote devices

- Lightweight and Efficient
- Bi-directional Communications
- Scale to Millions of Things
- Reliable Message Delivery
- Security Enabled

MQTT Publish / Subscribe Architecture



REQUIREMENT

Software Required

1. VS Code

Android Application

2. IoTMQTTPanel

Setup the MQTT Broker and Client

OVERVIEW

Here try to install the Mosquitto MQTT Broker and sending data from one terminal to another with publisher and subscriber client.

PROCEDURE

Installation

Install Mosquitto Broker

```
sudo apt-get install mosquitto
```

Install Mosquitto Client

```
sudo apt-get install mosquitto-clients
```

Testing

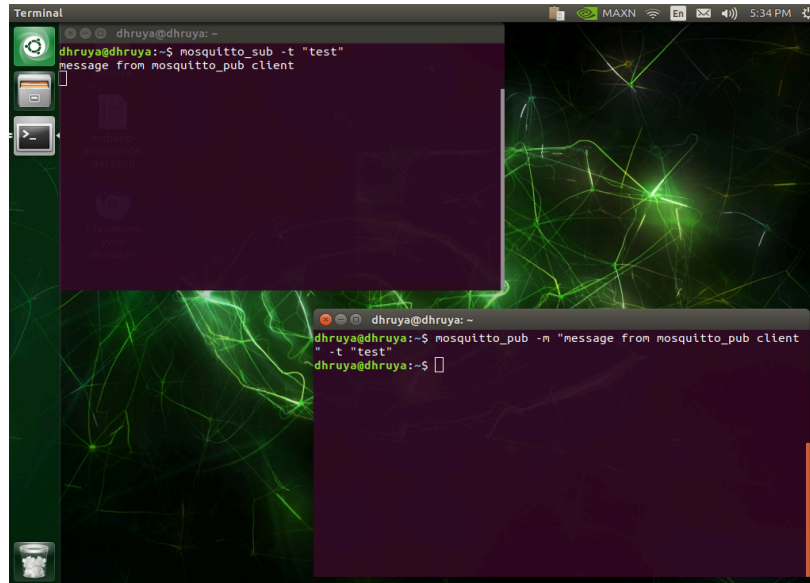
Sending message data from the publisher to subscribers which are connected to the same topic name "test".

Step1: Open the Terminal and start the subscriber

```
mosquitto_sub -t 'test'
```

Step2: Open another Terminal and publish the message

```
mosquitto_pub -m 'message from the moaquitto_pub client' -t 'test'
```

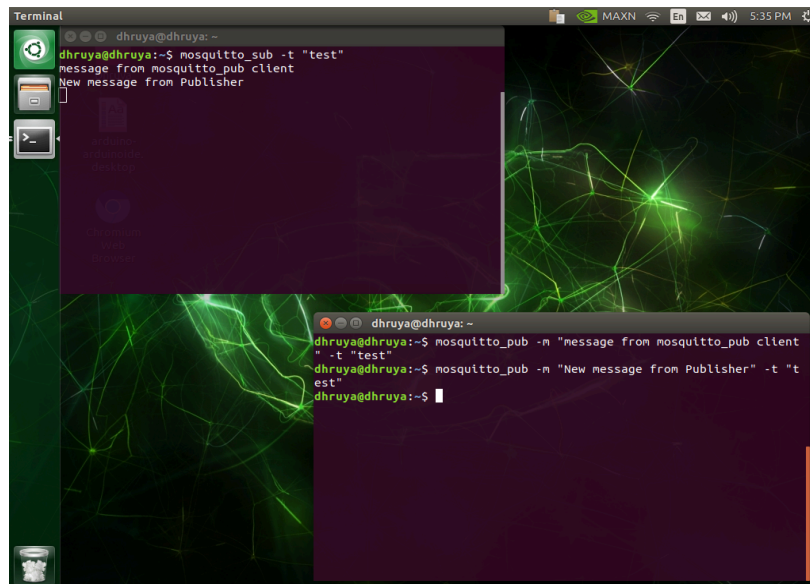


A terminal window titled "Terminal" with the prompt "dhruya@dhruya: ~". The user has run the command `dhruya@dhruya:~$ mosquitto_sub -t "test"`. The output shows "message from mosquitto_pub client". In the background, a smaller terminal window is visible with the command `dhruya@dhruya:~$ mosquitto_pub -n "message from mosquitto_pub client" -t "test"`.

```
dhruya@dhruya:~$ mosquitto_sub -t "test"
message from mosquitto_pub client

dhruya@dhruya:~$ mosquitto_pub -n "message from mosquitto_pub client" -t "test"
dhruya@dhruya:~$
```

With different message data on the same topic name.



A terminal window titled "Terminal" with the prompt "dhruya@dhruya: ~". The user has run the command `dhruya@dhruya:~$ mosquitto_sub -t "test"`. The output shows "message from mosquitto_pub client" followed by "New message from Publisher". In the background, a smaller terminal window is visible with two commands: `dhruya@dhruya:~$ mosquitto_pub -n "message from mosquitto_pub client" -t "test"` and `dhruya@dhruya:~$ mosquitto_pub -n "New message from Publisher" -t "test"`.

```
dhruya@dhruya:~$ mosquitto_sub -t "test"
message from mosquitto_pub client
New message from Publisher

dhruya@dhruya:~$ mosquitto_pub -n "message from mosquitto_pub client" -t "test"
dhruya@dhruya:~$ mosquitto_pub -n "New message from Publisher" -t "test"
dhruya@dhruya:~$
```

Setup the IoMQTTPanel APP

OVERVIEW

Earlier we were running both publisher and subscriber on the same machine, I want to send the data from another machine or device like my smartphone. For that, I connected both machines on the same gateway network and I installed the IoMQTTPanel application, where I can easily set up a connection. I created a dashboard in that application to simply send the text message on that topic name.

PROCEDURE

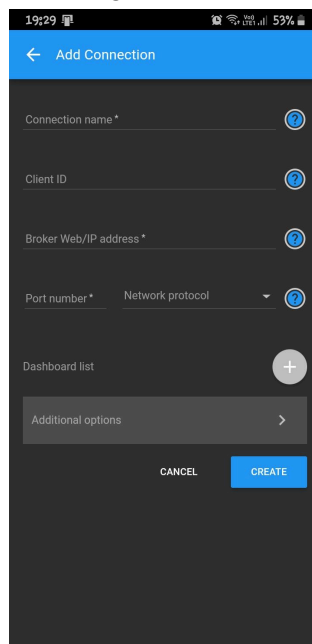
Android Phone Installation

IoMQTTPanel App

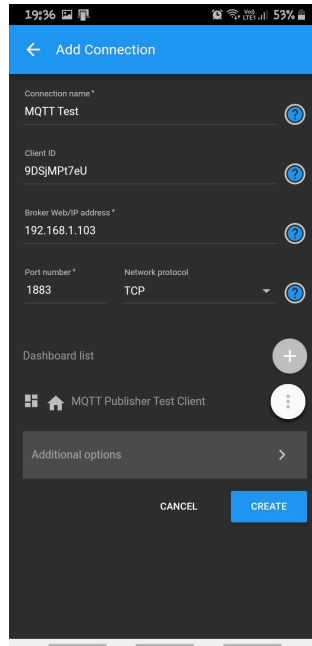
Download the IoMQTTPanel app from the Google play store

Add Connection

Step1: Open the App and Click on the plus sign button



Step2: Fill in the Connection name or click on the question mark to understand and click on create button

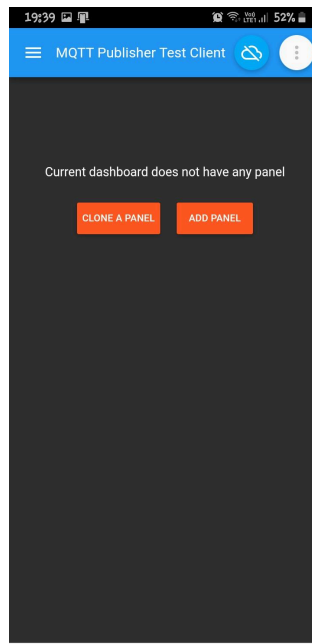


After that this connection will be added to our connection list.

Create Dashboard

To send the data from the Mobile client on the Subscriber client we have to create the dashboard or panel.

Step1: Click on the Connection we created



Step2: Click ADD PANEL

19:40 52%

← Add a Text Input panel

Panel name *

Topic *

☐ Show sent timestamp

☐ Confirm before publish

☐ Payload is JSON Data

☐ Clear text on publish

☐ Retain QoS

CANCEL CREATE

Step3: Add Panel name and add the topic name on which data to send and Create

19:41 52%

← Add a Text Input panel

Panel name *

Send Text

Topic *

dhruya

☒ Show sent timestamp

☒ Confirm before publish

Confirmation message *

Send

☐ Payload is JSON Data

☐ Clear text on publish

☐ Retain QoS

CANCEL CREATE

Test

Now we can send the data from the dashboard by simply add the text and click the send



Create Python Program for MQTT

Overview

For the Mosquitto MQTT broker, clients can be made with a python program using the paho-mqtt library. I have created MQTT publisher scripts that send the message data on the topic to which it is connected and the MQTT subscriber scripts which can receive that message data from the same topic.

Library Required

1. paho-mqtt

PROCEDURE

Installation

Install paho-mqtt Library

```
pip install paho-mqtt
```

Test 1

I have two scripts, one for publishing data and the other for receiving data that is connected to the same topic. In this program, I am using the online MQTT broker “mqtt.eclipseprojects.io”.

Publisher Node

```
import paho.mqtt.client as mqtt
from random import randrange, uniform
import time

mqttBroker = "mqtt.eclipseprojects.io"

client = mqtt.Client("Servo Data Publisher")
client.connect(mqttBroker)

while True:

    index = int(input("Enter Servo Position Index ID"))
    client.publish("SERVO", index)
    print("Just published " + str(index) + " to topic SERVO")
    time.sleep(1)
```

Subscriber Node

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

def on_message(client, userdata, message):
    print("received message:" ,str(message.payload.decode("utf-8")))

mqttBroker = "mqtt.eclipseprojects.io"

client = mqtt.Client("Servo Controller")
client.connect(mqttBroker)

client.loop_start()

client.subscribe("SERVO")
client.on_message=on_message

time.sleep(30)
client.loop_stop()
```

Problem

As I am using an online MQTT broker, it's taking time to receive the data on the subscriber side.

Solution

I already got an idea for this problem while reading articles on MQTT. I saw some of the codes where they used the IP address. I thought that I had to use the IP address of my machine, But that did not work for me. So I went deep into the documentation of the paho mqtt library where I got the keyword that is used to make a local machine as a broker and that is "localhost". So I simply use this keyword that makes my machine MQTT broker.

Test 2

Publisher Node

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

"""Publisher Client"""

# Import the Paho MQTT client
import paho.mqtt.client as mqtt

broker = "localhost"
port = 1883
topic = "topic/test"
msg = "Hello World!"

# Connecting client
client = mqtt.Client()
client.connect(broker, port, 60)

#client.publish("topic/test", "Hello world!")
result = client.publish(topic, msg)
# result: [0, 1]
status = result[0]
if status == 0:
    print(f"Send `{msg}` to topic `{topic}`")
else:
    print(f"Failed to send message to topic {topic}")

client.disconnect()
```

Subscriber Node

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

"""Subscriber Client"""

# Import the Paho MQTT client
import paho.mqtt.client as mqtt

broker = "localhost"
port = 1883
topic = "topic/test"

# Called after connecting the client
# 'rc' used to determine whether the client is connected successfully
def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    if rc == 0:
        print("Connected to MQTT Broker!")
    else:
        print("Failed to connect, return code %d\n", rc)
    client.subscribe(topic)

# Called after the client received messages from the MQTT Broker
# 'msg.topic' will give the name of subscribed topic
def on_message(client, userdata, msg):
    if msg.payload.decode() == "Hello world!":
        print("Yes!")
        client.disconnect()
    else:
        print(msg.payload.decode())
        client.disconnect()

# Connecting client
client = mqtt.Client()
client.connect(broker, port, 60)

client.on_connect = on_connect
client.on_message = on_message

client.loop_forever()
```

CONCLUSION

MQTT setup and installation process are very easy, without getting any problem I have done this setup in Jetson Nano. MQTT implementation with python using the paho mqtt library is easy.

RESULT

Using the localhost as a broker in the python code I am getting quick responses on the subscriber side.