MQTT Commands

* Type the command to download Python client and install Pip

pip install paho-mqtt

* Create a directory and access it

mkdir mqtt

cd mqtt

* Create a file in the directory and paste the program in the file

sudo nano pahomqtt.py

import paho.mqtt.client as mqtt  
# 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("$SYS/#")  
  
# The callback for when a PUBLISH message is received from the server.  
def on\_message(client, userdata, msg):  
 print(msg.topic+" "+str(msg.payload))  
  
client = mqtt.Client()  
client.on\_connect = on\_connect  
client.on\_message = on\_message  
  
client.connect("iot.eclipse.org", 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()

* Compile and run the code

python mqtt.py

**To Subscribe**

import paho.mqtt.client as mqtt  
# The callback for when the client receives a CONNACK response from the server.  
def on\_connect(client, userdata, 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("topic")  
  
# The callback for when a PUBLISH message is received from the server.  
def on\_message(client, userdata, msg):  
 print(msg.topic+" "+str(msg.payload))  
  
client = mqtt.Client()  
client.on\_connect = on\_connect  
client.on\_message = on\_message  
  
client.connect("instance ip address", 1883, 60)  
  
client.loop\_forever()

* Compile and run the code

**To Publish**

import paho.mqtt.client as mqtt  
# The callback for when the client receives a CONNACK response from the server.  
def on\_connect(client, userdata, 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.publish(“topic”, “message” , 0)  
  
# The callback for when a PUBLISH message is received from the server.  
def on\_message(client, userdata, msg):  
 print(msg.topic+" "+str(msg.payload))  
  
client = mqtt.Client()  
client.on\_connect = on\_connect  
client.on\_message = on\_message  
  
client.connect("instance ip address", 1883, 60)  
  
client.loop\_forever()

* Compile and run the code

**Turning led on and off using Pi and MQTT**

* Create a file in the directory and paste the program in the file

sudo nano pahomqttled.py

Import time

Import RPi.GPIO as GPIO

import paho.mqtt.client as mqtt

GPIO.setmode(GPIO.BOARD)

GPIO.setup(11, GPIO.OUT)  
  
# 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("TOPIC")

#client.publish(“topic”, “message” , 0)  
  
# The callback for when a PUBLISH message is received from the server.  
def on\_message(client, userdata, msg):  
 print(msg.topic+" "+str(msg.payload))

if(str(msg.payload)==”1”):

print(“LED ON”):

GPIO.OUTPUT(11, True)

else:

print(“LED OFF”):

GPIO.OUTPUT(11, False)

client = mqtt.Client()  
client.on\_connect = on\_connect  
client.on\_message = on\_message  
  
client.connect("Your instance", 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()