**Ex .10. Setup the cloud platform to log the data**

**AIM:** To setup the cloud platform using rasbperry PI to log the data

**Steps:**

1. Open chrome tab and type “micropython mqtt weather logger in wokwi esp32”
2. Copy the following link from the program <http://www.hivemq.com/demos/websocket-client/>
3. Paste the above link in the new chrome tab. Click “connect” and the click “add New topic subscription”. In the Topic field, type "wokwi-weather" then click Subscribe"
4. Now click on the DHT22 sensor in the simulation, change the temperature/humidity, and you should see the message appear on the MQTT Broker, in the "Messages" pane.

**Program:**

import network

import time

from machine import Pin

import dht

import ujson

from umqtt.simple import MQTTClient

# MQTT Server Parameters

MQTT\_CLIENT\_ID = "micropython-weather-demo"

MQTT\_BROKER    = "broker.mqttdashboard.com"

MQTT\_USER      = "saco"

MQTT\_PASSWORD  = "saco@123"

MQTT\_TOPIC     = "wokwi-weather"

sensor = dht.DHT22(Pin(15))

print("Connecting to WiFi", end="")

sta\_if = network.WLAN(network.STA\_IF)

sta\_if.active(True)

sta\_if.connect('Wokwi-GUEST', '')

while not sta\_if.isconnected():

  print(".", end="")

  time.sleep(0.1)

print(" Connected!")

print("Connecting to MQTT server... ", end="")

client = MQTTClient(MQTT\_CLIENT\_ID, MQTT\_BROKER, user=MQTT\_USER, password=MQTT\_PASSWORD)

client.connect()

print("Connected!")

prev\_weather = ""

while True:

  print("Measuring weather conditions... ", end="")

  sensor.measure()

  message = ujson.dumps({

    "temp": sensor.temperature(),

    "humidity": sensor.humidity(),

  })

  if message != prev\_weather:

    print("Updated!")

    print("Reporting to MQTT topic {}: {}".format(MQTT\_TOPIC, message))

    client.publish(MQTT\_TOPIC, message)

    prev\_weather = message

  else:

    print("No change")

  time.sleep(1)

**RESULT:** Thus the program was executed and the output was verified successfully