DEVELOPMENT PART-1

MQTT

MQTT stands for Message Queuing Telemetry Transport. It is an extremely simple and lightweight messaging protocol designed for limited devices and networks with high latency, low bandwidth or unreliable networks. Its design principles are designed to reduce the network bandwidth and resource requirements of devices and ensure security of supply.

With MQ Telemetry Transport, resource-constrained IoT devices can send or publish information on a specific topic to a server that acts as an MQTT message broker.

The MQTT broker is the center of every Publish / Subscribe protocol. Depending on the implementation, a broker can manage up to thousands of simultaneously connected MQTT clients. The broker is responsible for receiving all messages, filtering the messages, determining who subscribed to each message and sending the message to those subscribed clients. The Broker also holds the sessions of all persistent clients, including subscriptions and missed messages. Another task of the Broker is the authentication and authorization of clients

DHT11 Temperature sensor:

The DHT11 Temperature sensor is one of the low cost and small-sized sensors of its type. It is lab calibrated, stable and its signal output is digital.It is highly reliable when it comes to temperature & humidity sensing technology. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component and can connect to a high-performance 8-bit microcontroller. It's single-wire serial interface and 4-pin single row pin package makes system integration easy .The sensor converts the resistance measurement to humidity on the IC mounted to the back of the unit and transmits the readings directly to the Arduino.components that will be needed to connect the temperature sensor with tinkercad are TMP36, Arduino Uno, and Mini Breadboard. The 3 pin must be connected to the data (7), ground (GND), and voltage (5V).

Technical specifications:

Humidity Measurement Range: 20-90%RH

Temperature Measurement Range: 0-50℃
Humidity Accuracy: ±5%RH
Temperature Accuracy: ±2℃

The relative humidity is measured by the electrical resistance between two electrodes. The humidity sensing component of the DHT11 is a moisture holding substrate with the electrodes applied to the surface. The ions are released by the substrate as water vapor is absorbed by it, which in turn increases the conductivity between the electrodes. The change in resistance between the two electrodes is proportional to the relative humidity.

The sensor converts the resistance measurement to humidity on the IC mounted to the back of the unit and transmits the readings directly to the Arduino. The temperature readings from the DHT11 come from a surface mounted NTC temperature sensor built into the unit. The DHT11 uses one signal wire to transmit sensor readings to the Arduino digitally. The power comes from separate 5V and ground wires

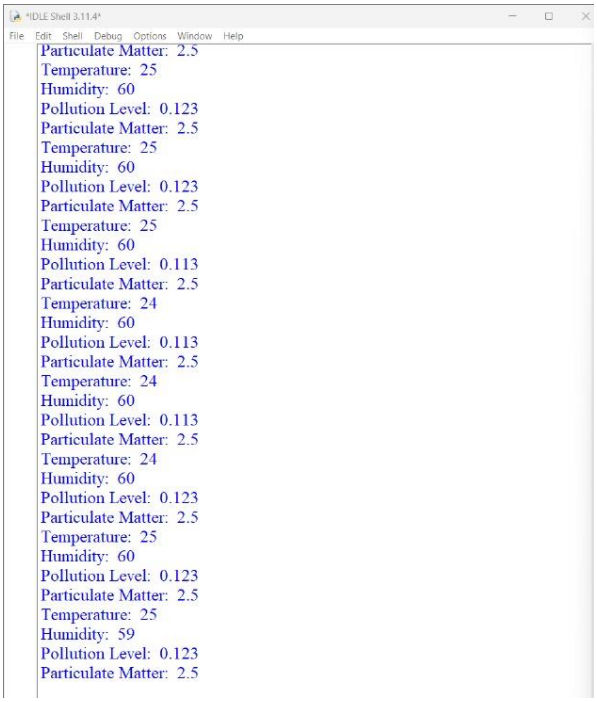
Setting up Arduino with DHT11:

DHT11 pin-1 is connected to 5v power supply pin on Arduino Uno
DHT11 pin-2 is connected to digital pin-A0 on Arduino Uno
DHT11 pin-3 is NC
DHT11 pin-4 is connected to GND (Ground) pin on Arduino Uno

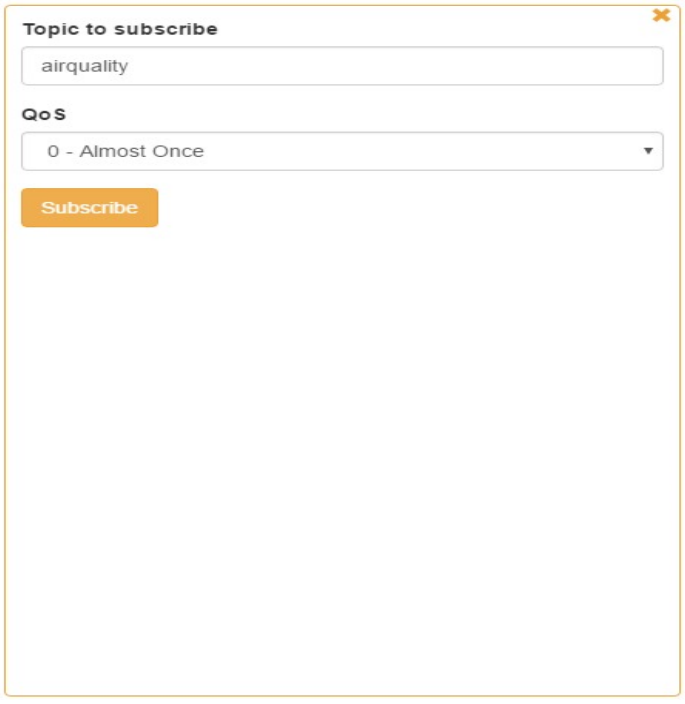
Python code:

```
import paho.mqtt.client as mqttClient
from threading import Thread
import json
import mysql.connector
import datetime

class Mqtt:
    def __init__(self):
        self.db = mysql.connector.connect(
            host="localhost",
            user="root",
            password="",
            database="demo")

        mqttclient = mqttClient.Client("52244535477668454")
        mqttclient.on_connect = self.on_connect
        mqttclient.on_message = self.on_message

        mqttclient.username_pw_set(username="", password="")

        mqttstatus = mqttclient.connect("broker.emqx.io", 1883, 60)
        mqttclient.subscribe("airquality", 2)
        mqttclient.loop_forever()

    def upload(self, msg):
        mqtt_msg = str(msg.payload.decode("utf-8"))
        mqtt_msg = mqtt_msg.replace("'", "\"")
        try:
            data = json.loads(mqtt_msg)

            temperature = data.get('temperature')
            humidity = data.get('humidity')
            pollution_level = data.get('pollution_level')
            particulate_matter = data.get('particulate_matter')

            print("Temperature: ", temperature)
            print("Humidity: ", humidity)
            print("Pollution Level: ", pollution_level)
            print("Particulate Matter: ", particulate_matter)

            cursor = self.db.cursor()
            insert_query = "INSERT INTO sensor_data (temperature, humidity, pollution_level,
particulate_matter, timestamp) VALUES (%s, %s, %s, %s, NOW())"
            data_to_insert = (temperature, humidity, pollution_level, particulate_matter)
            cursor.execute(insert_query, data_to_insert)
            self.db.commit()
            cursor.close()

        except json.JSONDecodeError as e:
            print("Error decoding JSON:", e)

    def on_connect(self, mqttclient, userdata, flags, rc):
```

```
        if rc == 0:
            print("Connected!")
        else:
            print("Connection failed")

    def on_message(self, mqttclient, userdata, msg):
        Thread(target=self.upload, args=(msg,)).start()

if __name__ == '__main__':
    Mqtt()
```



MQTT



airquality

{"temperature": 25, "humidity": 59, "pollution_level": 0.123, "particulate_matter": 2.5}

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : air quality, **messageId** : , **length** : 101, **Raw payload** : 123341161011091121011149711611711410134583250534432341041171091051001051161213458325357443234112111108108117116105111110951081011181011083458324846495051443234112971141161059911710897116101951099711611610111143458325046531251 0

{"temperature": 25, "humidity": 59, "pollution_level": 0.123, "particulate_matter": 2.5}

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : air quality, **messageId** : , **length** : 101, **Raw payload** : 123341161011091121011149711611711410134583250534432341041171091051001051161213458325357443234112111108108117116105111110951081011181011083458324846495051443234112971141161059911710897116101951099711611610111143458325046531251 0

DATABASE CONNECTION