**Practical - 5**

**Aim:**  Publish and Subscribe with MQTT

Control ESP8266 outputs and display sensor data from the ESP8266 on Node-RED (MQTT client).

**Components:** LED, Temperature and Humidity sensor DHT11, Jumper wires, ESP 8266 NodeMCU

**Procedure:**

* Install Mosquito Broker:
* The below link is helpful to follow full instruction on how to install Mosquito Broker

<http://www.steves-internet-guide.com/install-mosquitto-broker/>

* Library installation:

-Include the DHT library files (https://github.com/adafruit/DHT-sensor-library)

-Add "Adafruit\_Sensor.h" header file

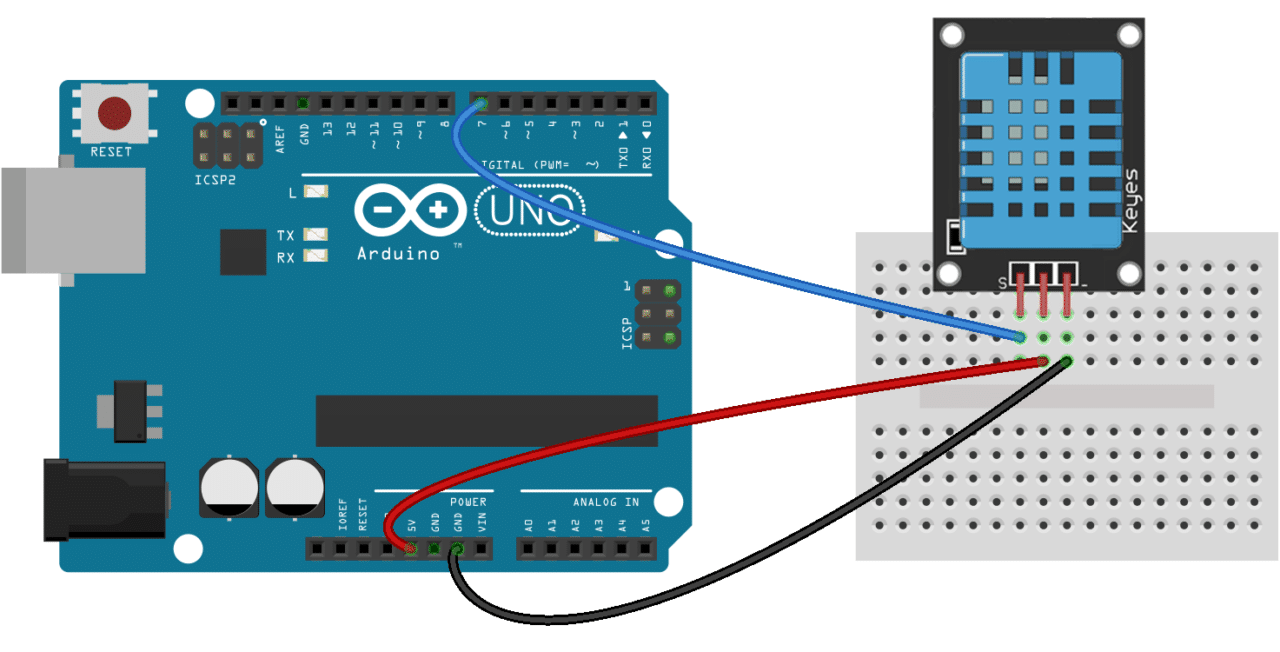
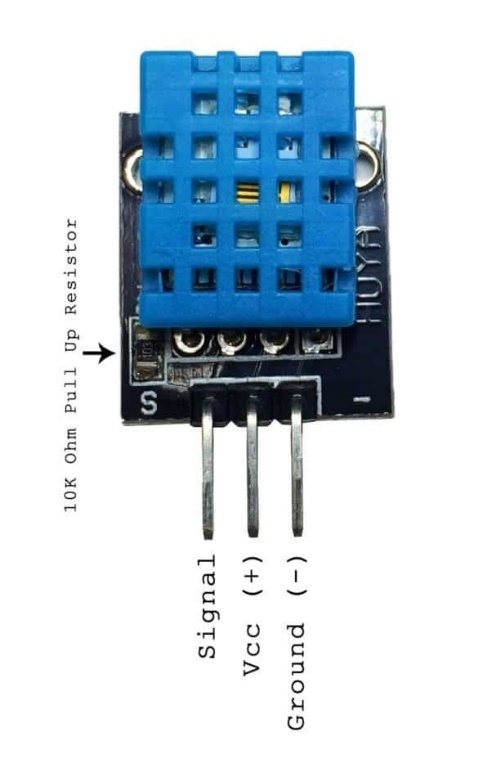
* Connections for ESP8266:

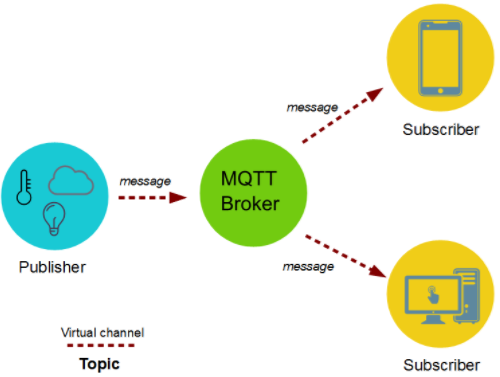
Sensor’s -Ve to GND

Sensor’s +Ve to 3v3

Sensor’s output to D7 (pin 13)

* Implementation in diagram





* Code:

#include <SimpleDHT.h>

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

SimpleDHT11 dht11;

#define DHT11\_PIN D7

const char\* ssid = "NU";

const char\* password = "12345678";

const char\* mqtt\_server = "192.168.43.169";

WiFiClient espClient;

PubSubClient client(espClient);

long lastMsg = 0;

char msg[50];

int value = 0;

void setup\_wifi() {

delay(100);

// We start by connecting to a WiFi network

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED)

{

delay(500);

Serial.print(".");

}

randomSeed(micros()) ;

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

void callback(char\* topic, byte\* payload, unsigned int length)

{

Serial.print("Command is : [");

Serial.print(topic);

int p =(char)payload[0]-'0';

byte temperature;

byte humidity;

int chk = dht11.read(DHT11\_PIN, &temperature, &humidity, NULL);

// if MQTT comes a 0 message, show humidity

if(p==0) {

Serial.println("to show humidity!]");

Serial.print(" Humidity is: " );

Serial.print(humidity);

Serial.println('%');

}

// if MQTT comes a 1 message, show temperature

if(p==1) {

// digitalWrite(BUILTIN\_LED, HIGH);

Serial.println(" is to show temperature!] ");

Serial.print(" Temp is: " );

Serial.print(temperature);

Serial.println(' C');

}

Serial.println();

}

void reconnect() {

// Loop until we're reconnected

while (!client.connected())

{

Serial.print("Attempting MQTT connection...");

// Create a random client ID

String clientId = "ESP8266Client-";

clientId += String(random(0xffff), HEX);

// Attempt to connect

//if you MQTT broker has clientID,username and password

//please change following line to if (client.connect(clientId,userName,passWord))

if (client.connect(clientId.c\_str()))

{

Serial.println("connected");

//once connected to MQTT broker, subscribe command if any

client.subscribe("OsoyooCommand");

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" try again in 5 seconds");

// Wait 6 seconds before retrying

delay(6000);

}

}

}

void setup() {

Serial.begin(9600);

setup\_wifi();

client.setServer(mqtt\_server, 1883);

client.setCallback(callback);

}

void loop() {

if (!client.connected()) {

reconnect();

}

client.loop();

long now = millis();

// read DHT11 sensor every 3 seconds

if (now - lastMsg > 3000) {

byte temperature;

byte humidity;

lastMsg = now;

int chk = dht11.read(DHT11\_PIN, &temperature, &humidity, NULL);

String msg="real time temperature: ";

msg = msg + temperature;

msg = msg+" C ;real time Humidity: " ;

msg = msg + humidity ;

msg = msg + "%";

char message[58];

msg.toCharArray(message,58);

Serial.println(message);

//publish sensor data to MQTT broker

client.publish("Data", message);

}

}