Wowki code for DHT11 sensor with Arduino

Wowki link:
https://wokwi.com/projects/348637208603787858

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 12     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11
#define LED 2

DHT dht (12, DHT22);// creating the instance by passing pin and typr of dht connected

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "xey3re"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32_controller"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "BME280_sensor"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "BME280_sensor"      //Token
String data3;
float h, t;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by passing parameter like server id,portand wificredential
```

```
// Callback function
void callback(char* topic, byte* payload, unsigned int length) {
  // In order to republish this payload, a copy must be made
  // as the orignal payload buffer will be overwritten whilst
  // constructing the PUBLISH packet.

  // Allocate the correct amount of memory for the payload copy
  byte* p = (byte*)malloc(length);
  // Copy the payload to the new buffer
  memcpy(p,payload,length);
  client.publish("project.indhu.dht", p, length);
  // Free the memory
  free(p);
}


#define dht_dpin 12       //digital pin, that DHT's data line is connected
#define DHTTYPE DHT22     //When using DHT11, put here DHT11 instead of DHT22

//int temp;               //Use for DHT11 instead of float
float temp;               //Use float for showing decimals of the temperature
reading. I recommend using for DHT22, no point to use for DHT11

//int hum;                //Use for DHT11 instead of float
float hum;                //Use float for showing decimals of the humidity
reading. I recommend using for DHT22, no point to use for DHT11

void setup() {

Serial. begin(9600);      //Initiate serial monitor
 dht.begin();             //Initiate DHT sensor

}

void loop() {

//delay(1000);                          //wait a sec (recommended for DHT11)
delay(500);                           //wait a 0,5 sec (recommended for DHT22)

temp=dht.readTemperature(false);       //Read temperature of DHT and store it
in to variable (temp). FALSE reads in celsius, leave empty for farenheit
hum=dht.readHumidity();               //Read humidity of DHTand store it iin
variable (hum).

Serial.print("Temperature: ");        //Print text "Temperature: " in to
serial monitor
Serial.println(temp);                 //Print variable (temperature value) in
to serial port. ln for line break
```

```
Serial.print("Humidity: ");            //Print text "Humidity: " in to serial
monitor
Serial.println(hum);                  //Print variable (temperature value) in
to serial port. ln for line break
Serial.println(" ");                  //print empty line in to serial monitor

  delay(2500);                        //optional delay, not really any point
reading the sensor more than once every 3 seconds
}
```