```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

const char* ssid = "abcd"; // user id for hotspot//
const char* password = "psswrd"; //psswrd of the hotspot //
const char* mqtt_server = "192.168.43.153";  // ip address of the host//
const char* clientID = "esp8266";
const char* outTopic = "output"; // here output means the string  that we give at the MQTT to obtain the result //

char ldr[20]; // variable to store the values //
float Temp;
float R1=10000,R2; // used the conversion formula //
float logR2 ;  // used the conversion formula //
float c1=1.009249522e-03, c2 = 2.378405444e-04, c3 = 2.019202697e-07;  // used the conversion formula //
int gate=D0;
volatile unsigned long duration=0;
unsigned char i[5];
unsigned int j[40];
unsigned char value=0;
unsigned answer=0;
int z=0;  // Declaration of variables //
int b=1; // Declaration of variables //


 WiFiClient espClient; // mqtt client //
PubSubClient client(espClient);
```

```
// logic to know whether our host has connected to the wifi or not //

void setup_wifi() {
 delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)  // loop works when the there is a problem with wifi connection and it shows "....."//
  {
   delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");   // if the wifi is connected ,loop condition fails and it executes from here. //
  Serial.println("IP address: ");  // string to show the id address of the host .//
  Serial.println(WiFi.localIP());  // localip() stores the ip address of the host .//
}

 // setup loop :we dercribe all the connected sensors and the pins //
void setup()
{
  Serial.begin(115200);  //baudrate //
  setup_wifi(); //here we are calling the wifi setup method .//
 client.setServer(mqtt_server, 1883); //port number os MQTT //
 pinMode(D2,OUTPUT);   //declaration of sensor connected pins in the circuit //
pinMode(D3,OUTPUT); //declaration of sensor connected pins in the circuit //
pinMode(D4,OUTPUT); //declaration of sensor connected pins in the circuit //
Serial.begin(9600);  //baudrate //

  // put your setup code here, to run once:
}
 //this method gets executed when host gets connected to the wifi but there is no MQTT client .//
void reconnect()
 {
  while (!client.connected())
 {
    Serial.print("Attempting MQTT connection...");
     if (client.connect(clientID)) {
      Serial.println("connected");
       client.publish(outTopic, clientID);

    }
 else {
      Serial.print("failed, rc="); // else part executes when the host doesn’t found any client even after reattempting for connectivity. //
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void loop()   // This loop helps to have the connection with the client continuously until the work gets done. //
{
  if (!client.connected())
  {
    reconnect();
  }

  publishldr();  //  calling  the function to print/send the ldr output values to MQTT//
  publishsensorvoltage();  //  calling  the  function to print/send the gas sensor detected voltages  to MQTT//
  publishTemp();//   calling  the function to print/send the thermistor sensed output values to MQTT//

}
// LDR FUNCTION.//
void publishldr()
{
  String d="";
  digitalWrite(D2,HIGH);
  digitalWrite(D3,LOW);
  digitalWrite(D4,LOW);
 int l = analogRead(A0);  //taking the output of mux to which sensors are connected .//
 d=d+"light intensity " +String(l);
 int z=d.length() + 1;
 d.toCharArray(ldr,z);
 Serial.print("");
  Serial.println(ldr);  // To display in serial monitor //
  client.publish(outTopic,ldr);  //to publish in MQTT client.//
  delay(1000);
}

void publishTemp()
```

```
{
  digitalWrite(D2,LOW);
  digitalWrite(D3,HIGH);

  digitalWrite(D4,LOW);
  String te = "";
  int Vo = analogRead(A0);  //taking the output of mux to which sensors are connected .//
  R2 = R1 * (1023.0 / (float)Vo - 1.0);  // function of a thermistor to convert the sensed value .//
  logR2 = log(R2);
  Temp = (1.0 / (c1 + c2*logR2 + c3*logR2*logR2*logR2));
Temp = Temp - 273.15;
  Temp = ((Temp * 5.0)/ 9.0 - 70.0);
  te=te+"temperature " + String (Temp);  //concatenating the obtained values to a string  to drive the point home effectively.//
  char charBuf[50];  // a character array //

 float y=te.length() +1;  // finding the length of the string.//
  String(te).toCharArray(charBuf,y); //converting it into character array because we used a  ‘’output’’ as a constant char to store the values. //
    Serial.print("Temperature= ");// To display in serial monitor //
  Serial.println(Temp); // To display in serial monitor //
  client.publish (outTopic,charBuf); );  //to publish in MQTT client.//
}



  void publishsensorvoltage()
  {
    digitalWrite(D2,LOW);
  digitalWrite(D3,LOW);
  digitalWrite(D4,HIGH);
  float sensorValue;

  sensorValue = analogRead(A0);    //taking the output of mux to which sensors are connected .//
float   sensorVoltage = sensorValue/1024*5.0;   //conversion of the sensed gas values to the corresponding voltage vlaues. //
  String se="";
  se=se+"pollutant gas  equivalent voltage values "+ String ( sensorVoltage);
  float x= se.length()+1;
  char charbuf1[50]; // a character array //
  String(se).toCharArray(charbuf1,x);   //string to character array conversion.//
  Serial.print("sensor voltage = "); ");// To display in serial monitor //
  Serial.print(sensorVoltage);
  Serial.println(" V");
  delay(1000);
  client.publish (outTopic,charbuf1); //converting it into character array because we used a  ‘’output’’ as a constant char to store the values. //
  }
```