```
/*
CopyRights   2020/7/18
                机械工程学院
                机械电子工程专业
                17 级机械电子工程 2 班盛梓茂
                《基于 arduino 和 esp8266 的智能家居的设计》
                ESP8266 部分代码
AllRights Reserved
*/
#include <ESP8266WiFi.h>
#include<dht11.h>       //测温度和湿度
#define tempGet D4                                        //温湿度传感器接口
#define Aout A0                                            //有毒传感器接收口
struct ss                                                 //温湿度变量
{
   float temperature;
   float humidity;
}result;
dht11 DHT11;
float gasvalue;

const char* ssid         = "nova 4";                           //我的 WIFI 名称
const char* password = "dragonszm";                       //我的 WIFI 密码
const char* host = "dweet.io";
static float value;
void setup() {
   Serial.begin(9600);
   pinMode(tempGet,INPUT);
   pinMode(Aout,INPUT);
   delay(10);
   Serial.print("Connecting to ");
   Serial.println(ssid);
   WiFi.begin(ssid, password);
   while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
   }

   Serial.println("");
   Serial.println("WiFi connected");
   Serial.println("IP address: ");
   Serial.println(WiFi.localIP());
}
void loop() {
        gasvalue = gasValue();   //读取有毒气体电压
         result   = getTemh();    //读取温度和湿度值
      Serial.print("temp:");
      Serial.println(result.temperature);
      Serial.print("hum:");
      Serial.println(result.humidity);
      Serial.print("ppm:");
      Serial.println(gasvalue);

   Serial.print("connecting to");
   Serial.println(host);
   WiFiClient client;
   const int httpPort=80;
   if(!client.connect(host,httpPort))
   {
      Serial.println("connection failed");
      return;
      }
      Serial.print("Requesting URL:");
      client.print(String("GET /dweet/for/esp8266_mythingsss?ppm=")
                                                       + gasvalue +
                                                       "&humidity="              +
result.humidity +
                                                       "&temperature="           +
result.temperature +
                                                       " HTTP/1.1\r\n" +
                                                        "Host: " + host + "\r\n" +
                                                        "Connection: close\r\n\r\n");
      delay(10);
      unsigned long timeout=millis();
      while(client.available()==0)
      {
        if(millis()-timeout>10000)
        {
         Serial.println(">>>Client Timeout!");
         client.stop();
         return;
         }
        }
        while(client.available())
        {
         String line=client.readStringUntil('\r');
         Serial.print(line);
         }
        Serial.println();
        Serial.println("closing connection.");

}
float gasValue() //对有毒气体的模拟值进行分析运算返回影响
{
   float ppm      = analogRead(Aout);

    return ppm;}
struct ss getTemh()//测得温度并将温度反映
{
   Serial.println("\n");
   int chk = DHT11.read(tempGet);
   Serial.print("Read sensor:");
   switch(chk)
   {
      case DHTLIB_OK:                   Serial.println("OK!");break;
      case DHTLIB_ERROR_CHECKSUM:Serial.println("CHECKSUM ERROR!");break;
      case DHTLIB_ERROR_TIMEOUT: Serial.println("TIMEOUT ERROR!");break;
      default:                          Serial.println("UNKNOWN ERROR!");break;
   }
   struct ss result;
   result.temperature = DHT11.temperature;
   result.humidity = DHT11.humidity;
   return result;
   Serial.print("Temperature:(摄氏度)");
   Serial.println((float)DHT11.temperature,2);
   Serial.print("Humidity:(%)");
   Serial.println((float)DHT11.humidity,2);
    delay(2000);
}
```