#define RL 47

#include <DHT.h> //  DHT.h library

#include <ESP8266WiFi.h> // ESP8266WiFi.h library

#define DHTPIN 0        //Not D0 Pin it is D3 Pin, Here 0 means GPIO 0.

#define DHTTYPE  DHT11

const char\* ssid     = "IOT";// replace subscribe with your WiFi SSID(Name)

const char\* password = "iot12345";//replace with Your Wifi Password name

const char\* host = "[api.thingspeak.com](http://api.thingspeak.com/)";

const char\* writeAPIKey = "AEI0BCY7A4HFS113"; //copy yout ThingSpeak channel API Key.

DHT dht(DHTPIN, DHTTYPE, 15);

char auth[] = "iLqX9pTBCtbZQejYp6uUSjF7DRcIcM64";

 #include <BlynkSimpleEsp8266.h>

void setup() {

  Blynk.begin(auth, ssid, password);

  pinMode(D0,OUTPUT);

// Initialize sensor

 Serial.begin(115200);

 dht.begin();

delay(1000);

 Serial.println("Connecting to ");

       Serial.println(ssid);

//  Connect to WiFi network

  WiFi.begin(ssid, password);

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

delay(500);

    Serial.print(".");

  }

   Serial.println("");

   Serial.println("WiFi connected");

}

void loop() {

  Blynk.run();

digitalWrite(D0,HIGH);

delay(350);

digitalWrite(D0,LOW);

delay(350);

float analog\_value;

  float VRL;

  float Rs;

  float Ro;

  for(int test\_cycle = 1 ; test\_cycle <= 500 ; test\_cycle++) //Read the analog output of the sensor for 200 times

  {

    analog\_value = analog\_value + analogRead(A0); //add the values for 200

  }

  analog\_value = analog\_value/500.0; //Take average

  VRL = analog\_value\*(5.0/1023.0); //Convert analog value to voltage

  //RS = ((Vc/VRL)-1)\*RL is the formulae we obtained from datasheet

  Rs = ((5.0/VRL)-1) \* RL;

  //RS/RO is 3.6 as we obtained from graph of datasheet

  Ro = Rs/3.6;

float humidity = dht.readHumidity();

float temperature = dht.readTemperature();

if(temperature > 25){

    Blynk.email("[abhiramivksj@gmail.com](mailto:abhiramivksj@gmail.com)", "ESP8266 Alert", "Fish Transportation Alert:High Tempearture");

    Blynk.notify("ALERT:High Tempearture");

  }

  if(humidity>80){

    Blynk.email("[abhiramivksj@gmail.com](mailto:abhiramivksj@gmail.com)", "ESP8266 Alert", "Fish Transportation Alert:High Humidity");

    Blynk.notify("ALERT:High Humidity");

  }

  if(Ro>350){

    Blynk.email("[abhiramivksj@gmail.com](mailto:abhiramivksj@gmail.com)", "ESP8266 Alert", "Fish Transportation Alert:High Ammonia");

    Blynk.notify("ALERT:High Ammonia");

  }

if (isnan(humidity) || isnan(temperature)) {

return;

  }

// make TCP connections

  WiFiClient client;

const int httpPort = 80;

if (!client.connect(host, httpPort)) {

return;

  }

  String url = "/update?key=";

  url+=writeAPIKey;

  url+="&field1=";

  url+=String(humidity);

  url+="&field2=";

  url+=String(temperature);

  url+="&field3=";

  url+=String(Ro);

  url+="\r\n";

// Request to the server

  client.print(String("GET ") + url + " HTTP/1.1\r\n" +

"Host: " + host + "\r\n" +

"Connection: close\r\n\r\n");

  Serial.print("Temperature:");

  Serial.print(temperature);

  Serial.print("\n");

  Serial.print("Humidity:");

  Serial.println(humidity);

    Serial.print("Ammonia PPM :");

  Serial.println(Ro);

  Serial.println("Send to ThingSpeak.\n");

client.stop();

  Serial.println("Wait for 15 sec to update next datapack in thingSpeak");

delay(1000);

}