#include <WiFi.h>

//#include "ThingSpeak.h"

#include <HTTPClient.h>

#include "DHT.h"

#define DHTPIN 4

#define DHTTYPE DHT11

const int pHpin = 34;  // ADC pin the sensor is connected to

const int turbidityPin = 36;

// Calibration points (update with your measured values)

const float voltageAtpH7 = 2.5;  // The voltage measured in pH 7 solution

const float voltageAtpH4 = 3.0;  // The voltage measured in pH 4 solution

DHT dht(DHTPIN, DHTTYPE);

const char\* ssid = "Atharva";

const char\* password = "12345678";

// Domain Name with full URL Path for HTTP POST Request

const char\* serverName = "http://api.thingspeak.com/update";

// write API Key provided by thingspeak

String apiKey = "NMD3AHAE6ZC8GPQO";

void setup() {

  Serial.begin(115200);

  WiFi.begin(ssid, password);

  dht.begin();

  Serial.println("Connecting");

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

    delay(500);

    Serial.print(".");

  }

  Serial.println("");

  Serial.print("Connected to WiFi network with IP Address: ");

  Serial.println(WiFi.localIP());

}

void loop() {

  if(WiFi.status()== WL\_CONNECTED){

      WiFiClient client;

      HTTPClient http;

      delay(10000); // wait for 10 seconds

     float h = dht.readHumidity();

      float t = dht.readTemperature();

      if (isnan(t)) {

       Serial.println(F("Failed to read from DHT sensor!"));

       return;

      }

      float pHvalue = readPH();

      float ntuval = turbidity();

      Serial.println(ntuval);

      // Your Domain name with URL path or IP address with path

      http.begin(client, serverName);

      // Specify content-type header

      http.addHeader("Content-Type", "application/x-www-form-urlencoded");

      // Data to send with HTTP POST

      String httpRequestData = "api\_key=" + apiKey + "&field1=" + String(t)+ "&field4=" + String(pHvalue) + "&field3=" + String(ntuval) + "&field2=" + String(h);

      // Send HTTP POST request

      int httpResponseCode = http.POST(httpRequestData);

      /\*

      // If you need an HTTP request with a content type: application/json, use the following:

      http.addHeader("Content-Type", "application/json");

      // JSON data to send with HTTP POST

      String httpRequestData = "{\"api\_key\":\"" + apiKey + "\",\"field1\":\"" + String(random(40)) + "\"}";

      // Send HTTP POST request

      int httpResponseCode = http.POST(httpRequestData);\*/

      Serial.print("HTTP Response code: ");

      Serial.println(httpResponseCode);

      http.end();

    }

    else {

      Serial.println("WiFi Disconnected");

    }

}

float readPH(){

    int sensorValue = analogRead(pHpin);  // Read the sensor's output

  float voltage = sensorValue \* (3.3 / 4095.0);  // Convert to voltage

  // Calculate the slope (m) and intercept (b) for the line y = mx + b

  float slope = (7.0 - 4.0) / (voltageAtpH7 - voltageAtpH4);

  float intercept = 7.0 - (slope \* voltageAtpH7);

  // Convert the voltage to pH value using the calibration data

  float pH = (voltage \* slope) + intercept;

  // Ensure pH value stays within the expected range

  pH = constrain(pH, 0, 14);

  delay(1000);  // Wait a second between readings

  return pH;

  }

float turbidity(){

  float ntu;

   int analogvalue =  analogRead(turbidityPin);

   Serial.println(analogvalue);

   float voltage = analogvalue \* (5.0 / 4095.0);

//   float  ntu = -1120.4\*sq(voltage)+5742.3\*voltage-4352.9;

   ntu = map(analogvalue, 753, 2010, 100, 0);

//  voltage = voltage/800;

//    voltage = round\_to\_dp(voltage,2);

//

//    if(voltage < 2.5){

//      ntu = 3000.00;

//    }else{

//      ntu = -1120.4\*sq(voltage)+5742.3\*voltage-4353.8;

//    }

    delay(1000);

    return ntu;

    }

float round\_to\_dp( float in\_value, int decimal\_place )

{

  float multiplier = powf( 10.0f, decimal\_place );

  in\_value = roundf( in\_value \* multiplier ) / multiplier;

  return in\_value;

}