#include <LiquidCrystal\_I2C.h> // Include the LiquidCrystal\_I2C library

#include <Wire.h>

#include <DHT.h> // Include the DHT library

#include <WiFi.h> // Use the correct library for your WiFi module

#include<Arduino.h>

String apiKey = "KI20811DMHTANMA3"; // Enter your Write API key from ThingSpeak

const char \*ssid = "Oppo"; // replace with your wifi ssid and wpa2 key

const char \*pass = "369369369";

const char\* server = "api.thingspeak.com";

WiFiClient client;

// Define the pin for the gas sensor

const int gasSensorPin = A0;

// Define the pin for the relay

const int relayPin = 2; // Assuming the relay is connected to GPIO2

// Define the pin for the DHT22 sensor

const int DHTPIN = 4; // Assuming the DHT22 sensor is connected to GPIO4

// Define the type of DHT sensor (DHT22)

#define DHTTYPE DHT22

// Define threshold values for temperature and humidity

const float maxTemperature = 33.0; // Adjust as needed

const float minHumidity = 50.0; // Adjust as needed

// ADC parameters for ESP32

#define V\_REF 1100 // ADC reference voltage (mV)

#define ADC\_BITS 10 // ADC resolution (bits)

#define ADC\_MAX 1023 // Maximum ADC value

// Create an object for the LCD

LiquidCrystal\_I2C lcd(0x27, 16, 2); // I2C address 0x27, 16x2 display

// Create an object for the DHT sensor

DHT dht(DHTPIN, DHTTYPE);

void setup() {

Serial.begin(115200);

Serial.println("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, pass);

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

{

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

// Set relay pin as output

pinMode(relayPin, OUTPUT);

// Initialize the DHT sensor

dht.begin();

// Initialize the LCD

lcd.init();

lcd.backlight();

}

void loop() {

// Read the analog value from the gas sensor

int gasValue = analogRead(gasSensorPin);

// Read temperature and humidity from the DHT22 sensor

float temperature = dht.readTemperature();

float humidity = dht.readHumidity();

// Convert analog value to voltage

float voltage = gasValue \* (V\_REF / ADC\_MAX);

if (isnan(gasValue)|| isnan(temperature) || isnan(humidity) )

{

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

return;

}

if (client.connect(server, 80)) // "184.106.153.149" or api.thingspeak.com

{

String postStr = "field1=" + String(temperature) + "&field2=" + String(humidity) + "&field3=" + String(gasValue);

client.print("POST /update HTTP/1.1\n");

client.print("Host: api.thingspeak.com\n");

client.print("Connection: close\n");

client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");

client.print("Content-Type: application/x-www-form-urlencoded\n");

client.print("Content-Length: ");

client.print(postStr.length());

client.print("\n\n");

client.print(postStr);

Serial.print("Temperature: ");

Serial.println(temperature);

Serial.print("Humidity: ");

Serial.println(humidity);

Serial.print("Gas Level: ");

Serial.println(gasValue);

Serial.println("Data Sent to ThingSpeak");

}

// Print the gas sensor reading to serial monitor

Serial.print("Gas Value: ");

Serial.println(gasValue);

// Display the gas sensor reading on LCD

lcd.setCursor(0, 1); // Set cursor to second row

lcd.print("G:");

lcd.print(voltage, 2); // Display voltage with 2 decimal places

lcd.print(" H:");

lcd.print(humidity);

lcd.print("%");

// Display temperature and humidity readings on LCD

lcd.setCursor(0, 0);

lcd.print("T: ");

lcd.print(temperature);

lcd.print(" C");

// Check if gas value exceeds threshold to turn on relay

// if (voltage > 68) {

// // Turn on the relay (assuming it's active LOW)

// digitalWrite(relayPin, LOW);

// Serial.println("Relay ON");

// } else {

// // Turn off the relay

// digitalWrite(relayPin, HIGH);

// Serial.println("Relay OFF");

// }

// Check temperature and humidity to determine relay state

if (temperature > maxTemperature || humidity > minHumidity || voltage > 68) {

// Conditions met to turn on the relay

digitalWrite(relayPin, LOW); // Turn on the relay

Serial.println("Relay ON (Temperature or Humidity exceeded)");

} else {

// Conditions not met, turn off the relay

digitalWrite(relayPin, HIGH); // Turn off the relay

Serial.println("Relay OFF");

}

// Print temperature and humidity to serial monitor

Serial.print("Temperature: ");

Serial.print(temperature);

Serial.println(" \*C");

Serial.print("Humidity: ");

Serial.print(humidity);

Serial.println(" %");

// Delay before next reading

delay(1000); // Adjust delay time as needed

}