#include <Wire.h>

#include <WiFi.h>

#include <PubSubClient.h>

// Update these with values suitable for your network.

#define ssid "\*\*\*\*\*\*\*\*"//put your wifi ssid here.

#define password "\*\*\*\*\*\*\*\*\*\*"//put your wifi password here.

#define mqtt\_server "\*\*\*\*\*\*\*\*\*\*"

// Defining Constant MQTT Topics

#define topic\_MQ2 "sensor/MQ-2/SMOKE"

#define topic\_MQ135 "sensor/MQ-135/Carbon"

#define topic\_MQ4 "sensor/MQ-4/CH4"

#define topic\_MQ7 "sensor/MQ-7/CO"

#include <LiquidCrystal\_I2C.h>

#define I2C\_SDA 21

#define I2C\_SCL 22

LiquidCrystal\_I2C lcd(0x27,20,4);

#define mq2Pin 33

#define MQ7 34

#define MQ135 32

#define MQ4 35

#define Buzzer 26

void callback(String topic, byte\* message, unsigned int length);

WiFiClient espClient;

PubSubClient client(espClient);

void setup() {

Serial.begin(115200);

lcd.begin();

lcd.backlight();

pinMode(Buzzer, *OUTPUT*);

pinMode(mq2Pin, INPUT);

pinMode(MQ4, INPUT);

pinMode(MQ7, INPUT);

pinMode(MQ135, INPUT);

setup\_wifi();

client.setServer(mqtt\_server, 1883);

client.setCallback(callback);

}

void setup\_wifi() {

delay(100);

// We start by connecting to a WiFi network

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

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

{

delay(500);

Serial.print(".");

}

randomSeed(micros());

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

void callback(String topic, byte\* message, unsigned int length) {

Serial.print("Message arrived on topic [");

Serial.print(topic);

Serial.print(". message:");

String messageTemp;

Serial.print("] ");

for (int i = 0; i < length; i++) {

Serial.print ((char)message[i]);

messageTemp += (char)message[i];

}

Serial.println();

}

void reconnect() {

// Loop until we're reconnected

while (!client.connected())

{

Serial.print("Attempting MQTT connection...");

// Attempt to connect

//if you MQTT broker has clientID,username and password

//please change following line to if (client.connect(clientId,userName,passWord))

if (client.connect("ESP32Client"))

{

Serial.println("connected");

//once connected to MQTT broker, subscribe command if any

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" try again in 5 seconds");

// Wait 6 seconds before retrying

delay(500);

}

}

} //end reconnect()

void loop() {

if (!client.connected()) {

reconnect();

}

client.loop();

lcd.setCursor(0,0);

lcd.print("GAS value: ");

lcd.setCursor(0,1);

lcd.print("C02 value: ");

lcd.setCursor(0,2);

lcd.print("CO value: ");

lcd.setCursor(0,3);

lcd.print("CH4 value: ");

int four = analogRead(MQ4);

Serial.print("CH4 Concentration: ");

Serial.print(four);//prints the methane value

Serial.print(" ");

Serial.println("ppm ");

lcd.setCursor(10,3);

lcd.print(four);

lcd.print(" ");

lcd.setCursor(14,3);

lcd.print("ppm ");

int othreefive = analogRead(MQ135); //

Serial.print("C02 value: ");

Serial.print(othreefive);

Serial.print(" ");

Serial.println("ppm ");

lcd.setCursor(10,0);

lcd.print(othreefive);

lcd.print(" ");

lcd.setCursor(14,0);

lcd.print("ppm ");

int two = analogRead(mq2Pin);

Serial.print("Gas Value :");

Serial.print(two);

Serial.print(" ");

Serial.println("ppm ");

lcd.setCursor(10,1);

lcd.print(two);

lcd.print(" ");

lcd.setCursor(14,1);

lcd.print("ppm ");

int val7 = analogRead(MQ7);

float voltage = val7 \* (5.0 / 1023.0);

int seven = (voltage - 0.1) / 0.1;

Serial.print("CO value: ");

Serial.print(seven);

Serial.print(" ");

Serial.println("ppm ");

lcd.setCursor(10,2);

lcd.print(seven);

lcd.print(" ");

lcd.setCursor(14,2);

lcd.print("ppm ");

Serial.println(" ");

Serial.println(" ");

if (two > 1250) {

lcd.clear();

digitalWrite(Buzzer, HIGH);

Serial.println("DANGEROUS !!! ");

Serial.println("GAS/SMOKE ALREADY OFF LIMIT");

lcd.setCursor(4,1);

lcd.print("DANGEROUS !!! ");

lcd.setCursor(0,2);

lcd.print("GAS HAS BEEN LEAKING");

delay (3000);

lcd.clear();

lcd.backlight();

} else {

digitalWrite(Buzzer, LOW);

}

if (seven > 100) {

lcd.clear();

digitalWrite(Buzzer, HIGH);

Serial.println("DANGEROUS !!! ");

Serial.println("CO ALREADY OFF LIMIT");

lcd.setCursor(4,1);

lcd.print("DANGEROUS !!! ");

lcd.setCursor(0,2);

lcd.print("CO HAS BEEN LEAKING");

delay (3000);

lcd.clear();

lcd.backlight();

} else {

digitalWrite(Buzzer, LOW);

}

if (othreefive > 2000) {

lcd.clear();

digitalWrite(Buzzer, HIGH);

Serial.println("DANGEROUS !!! ");

Serial.println("CO2 ALREADY OFF LIMIT");

lcd.setCursor(4,1);

lcd.print("DANGEROUS !!! ");

lcd.setCursor(0,2);

lcd.print("CO2 HAS BEEN LEAKING");

delay (3000);

lcd.clear();

lcd.backlight();

}

else{

digitalWrite(Buzzer, LOW);

}

if (four > 1000) {

lcd.clear();

digitalWrite(Buzzer, HIGH);

Serial.println("DANGEROUS !!! ");

Serial.println("Methane ALREADY OFF LIMIT");

lcd.setCursor(4,1);

lcd.print("DANGEROUS !!! ");

lcd.setCursor(0,2);

lcd.print("CH4 HAS BEEN LEAKING");

delay (3000);

lcd.clear();

lcd.backlight();

}

else{

digitalWrite(Buzzer, LOW);

}

delay(500);

char message[80];

snprintf(message, sizeof(message), "%d", two);

client.publish(topic\_MQ2, message);

snprintf(message, sizeof(message), "%d", four);

client.publish(topic\_MQ4, message);

snprintf(message, sizeof(message), "%d", seven);

client.publish(topic\_MQ7, message);

snprintf(message, sizeof(message), "%d", othreefive);

client.publish(topic\_MQ135, message);

}

#include <LiquidCrystal\_I2C.h>

#define I2C\_SDA 21

#define I2C\_SCL 22

LiquidCrystal\_I2C lcd(0x27,20,4); // set the LCD address to 0x27 for a 16 chars and 2 line display

#define Buzzer 26

const int MQ2 = 33;

const int AMQ7pin=34 ;//the AOUT pin of the CO sensor goes into analog pin A0 of the arduino

const int AMQ135pin=32;

const int AMQ4pin=35; // Connect AO of MQ4 with Analog channel 0 pin (A0) of Arduino

int threshold\_value; // A variable to store digital *output* of MQ4

int val7;

float voltage;

int ValueMQ4; // stores analog *output* of MQ4 sensor

int Valuemq7;

int ValueMQ135;

void setup() {

Serial.begin(115200);

lcd.begin(); // initialize the lcd

// Print a message to the LCD.

lcd.backlight();

pinMode(AMQ4pin, INPUT);

pinMode(AMQ7pin, INPUT);

pinMode(AMQ135pin, INPUT);

pinMode(Buzzer, *OUTPUT*);

}

void loop() {

lcd.setCursor(0,0);

lcd.print("GAS value: ");

lcd.setCursor(0,1);

lcd.print("C02 value: ");

lcd.setCursor(0,2);

lcd.print("CO value: ");

lcd.setCursor(0,3);

lcd.print("CH4 value: ");

ValueMQ135 = analogRead(AMQ135pin); // read analog input pin 0

Serial.print("C02 value: ");

Serial.print(ValueMQ135);

Serial.print(" ");

Serial.println("ppm ");

lcd.setCursor(10,0);

lcd.print(ValueMQ135);

lcd.print(" ");

lcd.setCursor(14,0);

lcd.print("ppm ");

int valuemq2 = analogRead(MQ2);

Serial.print("Gas Value :");

Serial.print(valuemq2);

Serial.print(" ");

Serial.println("ppm ");

lcd.setCursor(10,1);

lcd.print(valuemq2);

lcd.print(" ");

lcd.setCursor(14,1);

lcd.print("ppm ");

int val7 = analogRead(AMQ7pin);

float voltage = val7 \* (5.0 / 1023.0);

int Valuemq7 = (voltage - 0.1) / 0.1;//reads the analaog value from the CO sensor's AOUT pin

Serial.print("CO value: ");

Serial.print(Valuemq7);

Serial.print(" ");

Serial.println("ppm ");

lcd.setCursor(10,2);

lcd.print(Valuemq7);

lcd.print(" ");

lcd.setCursor(14,2);

lcd.print("ppm ");

ValueMQ4= analogRead(AMQ4pin); // Take Analog *output* measurement sample from AO pin of MQ4 sensor

Serial.print("CH4 Conentration: ");

Serial.print(ValueMQ4);//prints the methane value

Serial.print(" ");

Serial.println("ppm ");

lcd.setCursor(10,3);

lcd.print(ValueMQ4);

lcd.print(" ");

lcd.setCursor(14,3);

lcd.print("ppm ");

Serial.println(" ");

Serial.println(" ");

//prints the CO value

if (valuemq2 > 1250) {

lcd.clear();

digitalWrite(Buzzer, HIGH);

Serial.println("DANGEROUS !!! ");

Serial.println("GAS/SMOKE ALREADY OFF LIMIT");

lcd.setCursor(4,1);

lcd.print("DANGEROUS !!! ");

lcd.setCursor(0,2);

lcd.print("GAS HAS BEEN LEAKING");

delay (3000);

lcd.clear();

lcd.backlight();

} else {

digitalWrite(Buzzer, LOW);

}

delay(1000);

if (Valuemq7 > 1200) {

lcd.clear();

digitalWrite(Buzzer, HIGH);

Serial.println("DANGEROUS !!! ");

Serial.println("CO ALREADY OFF LIMIT");

lcd.setCursor(4,1);

lcd.print("DANGEROUS !!! ");

lcd.setCursor(0,2);

lcd.print("CO HAS BEEN LEAKING");

delay (3000);

lcd.clear();

lcd.backlight();

} else {

digitalWrite(Buzzer, LOW);

}

delay(1000);

if (ValueMQ135 > 2000) {

lcd.clear();

digitalWrite(Buzzer, HIGH);

Serial.println("DANGEROUS !!! ");

Serial.println("CO2 ALREADY OFF LIMIT");

lcd.setCursor(4,1);

lcd.print("DANGEROUS !!! ");

lcd.setCursor(0,2);

lcd.print("CO2 HAS BEEN LEAKING");

delay (3000);

lcd.clear();

lcd.backlight();

}

else{

digitalWrite(Buzzer, LOW);

}

if (ValueMQ4 > 1000) {

lcd.clear();

digitalWrite(Buzzer, HIGH);

Serial.println("DANGEROUS !!! ");

Serial.println("Methane ALREADY OFF LIMIT");

lcd.setCursor(4,1);

lcd.print("DANGEROUS !!! ");

lcd.setCursor(0,2);

lcd.print("CH4 HAS BEEN LEAKING");

delay (3000);

lcd.clear();

lcd.backlight();

}

else{

digitalWrite(Buzzer, LOW);

}

delay(1000);

}