**MQ2 GAS SENSOR**

#define MQ2pin (0) // connected to A0 of arduino

#include <LiquidCrystal.h>

float sensorValue; //variable to store sensor value

const int buzzer = 8 ;

int MQ2 = A0;

int sensorThres = 150;

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup()

{

pinMode(buzzer, OUTPUT);

Serial.begin(9600); // sets the serial port to 9600

Serial.println("Gas sensor warming up!");

delay(20000); // allow the MQ-2 to warm up

**}**

**void loop()**

**{**

int analogSensor = analogRead(MQ2);

Serial.print("Pin A2: ");

Serial.println(analogSensor);

if (analogSensor > sensorThres)

{

tone(buzzer, 200, 100);

}

else

{

noTone(buzzer);

}

delay(100);

sensorValue = analogRead(MQ2pin); // read analog input pin 0

Serial.print("Sensor Value: ");

Serial.print(sensorValue);

if(sensorValue > 200)

{

Serial.print(" |Warning! AIR POLLUTED HIGH!");

lcd.print(" AIR POLLUTED HIGH!");

}

else

{

Serial.print("AIR POLLUTED LESS");

lcd.print(" AIR POLLUTED HIGH!");

}

Serial.println("");

delay(2000); // wait 2s for next reading

}

**MQ135 GAS SENSOR**

#include <LiquidCrystal.h>

int sensorValue;

const int buzzer = 8 ;

int MQ135 = A2;

int sensorThres = 800;

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup()

{

lcd.begin(16, 2);

Serial.begin(9600);

pinMode(MQ135, INPUT);

pinMode(buzzer, OUTPUT);

}

void loop()

{

int sensorValue = analogRead(MQ135);

Serial.print("Pin A2: ");

Serial.println(sensorValue);

if (sensorValue > sensorThres)

{

tone(buzzer, 1000, 200);

}

else

{

noTone(buzzer);

}

delay(100);

sensorValue = analogRead(2); // read analog input pin 0

Serial.print("Air Polluted HIGH! =");

Serial.print(sensorValue, DEC); // prints the value read

Serial.println(" PPM");

lcd.setCursor(0,0);

lcd.print("MQ135=");

lcd.print(sensorValue,DEC);

lcd.print(" PPM");

lcd.println(" ");

lcd.print(" ");

delay(2000); // wait 100ms for next reading

}

**MQ7 GAS SENSOR**

#include <LiquidCrystal.h>

int sensorValue;

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup()

{

// initialize serial communication at 9600 bits per second:

lcd.begin(16, 2);

Serial.begin(9600);

}

void loop() {

int sensorValue = analogRead(A1); // connected to A1 of arduino

Serial.print("MQ7 READINGS : ");

Serial.println(sensorValue);

lcd.setCursor(0,1);

lcd.print("mq7 = ");

lcd.print(sensorValue,DEC);

lcd.println(" ");

lcd.print(" ");

delay(100);

}

**DHT11 SENSOR**

#include <Adafruit\_Sensor.h>

#include <DHT.h>

#include <DHT\_U.h>

#define DHTTYPE DHT11 // DHT 11

#define DHTPIN 3

DHT\_Unified dht(DHTPIN, DHTTYPE);

uint32\_t delayMS;

void setup() {

Serial.begin(9600);

dht.begin();

sensor\_t sensor;

delayMS = sensor.min\_delay / 1000;

}

void loop() {

sensors\_event\_t event;

dht.temperature().getEvent(&event);

Serial.print(F("Temperature: "));

Serial.print(event.temperature);

Serial.println(F("°C"));

dht.humidity().getEvent(&event);

Serial.print(F("Humidity: "));

Serial.print(event.relative\_humidity);

Serial.println(F("%"));

delay(delayMS);

}

**ESP8266 WIFI MODULE**

#include <SoftwareSerial.h>

#define RX 7

#define TX 6

String AP = "Redmi 9A"; // AP NAME

String PASS = "1234567890"; // AP PASSWORD

String API = "XYMYJ9BEQHM5UDL4"; // Write API KEY

String HOST = "api.thingspeak.com";

String PORT = "30";

int countTrueCommand;

int countTimeCommand;

boolean found = false;

int valSensor = 1;

SoftwareSerial esp8266(RX,TX);

void setup()

{

Serial.begin(9600);

esp8266.begin(115200);

sendCommand("AT",5,"OK");

sendCommand("AT+CWMODE=1",5,"OK");

sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK");

}

void loop()

{

String getData = "GET /update?api\_key="+ API +"&field1="+gettemphumidValue()+"&field2="+getMQ135Value()+"&field3="+getMQ7Value()+"&field4="+getMQ2Value();

sendCommand("AT+CIPMUX=1",5,"OK");

sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK");

sendCommand("AT+CIPSEND=0," +String(getData.length()+4),4,">");

esp8266.println(getData);delay(1500);countTrueCommand++;

sendCommand("AT+CIPCLOSE=0",5,"OK");

}

String gettemphumidValue()

{

int temphumid;

temphumid=analogRead(A3);

return String(temphumid);

}

String getMQ135Value()

{

int MQ135;

MQ135=analogRead(A2);

return String(MQ135);

}

String getMQ7Value()

{

int MQ7;

MQ7=analogRead(A1);

return String(MQ7);

}

String getMQ2Value()

{

int MQ2;

MQ2=analogRead(A0);

return String(MQ2);

}

void sendCommand(String command, int maxTime, char readReplay[])

{

Serial.print(countTrueCommand);

Serial.print(". at command => ");

Serial.print(command);

Serial.print(" ");

while(countTimeCommand < (maxTime\*1))

{

esp8266.println(command);//at+cipsend

if(esp8266.find(readReplay))//ok

{

found = true;

break;

}

countTimeCommand++;

}

if(found == true)

{

Serial.println("OYI");

countTrueCommand++;

countTimeCommand = 0;

}

if(found == false)

{

Serial.println("Fail");

countTrueCommand = 0;

countTimeCommand = 0;

}

found = false;

}

**FULL CODE**

#include <Wire.h>

#include <DHT.h>

#include <LiquidCrystal.h>

#define DHTPIN A3

#define DHTTYPE DHT11

#define DELAY\_IN\_SEC 2000

DHT dht(DHTPIN, DHTTYPE);

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

const int mq2 = A0;

const int mq7 = A1;

const int mq135 = A2;

const int buzzer = 8;

float m = -0.6527;

float b = 1.30;

float R0 = 21.91;

double mq135Limit = 800;

double temperatureLimit = 99.5;

double humidityLimit = 50;

double mq7Limit = 400;

double mq2Limit = 10000;

double analogToPPM(int aValue)

{

float sensor\_volt;

float RS\_gas;

float ratio;

int sensorValue = aValue;

sensor\_volt = sensorValue \* (5.0 / 1023.0);

RS\_gas = ((5.0 \* 10.0) / sensor\_volt) - 10.0;

ratio = RS\_gas / R0;

double ppm\_log = (log10(ratio) - b) / m;

return ppm\_log;

}

void setup()

{

Serial.begin(9600);

lcd.begin(16, 2);

pinMode(mq135, INPUT);

pinMode(mq7, INPUT);

pinMode(mq2, INPUT);

pinMode(buzzer, OUTPUT);

dht.begin();

lcd.setCursor(0, 0);

lcd.print(" Atmosphere ");

lcd.setCursor(0, 3);

lcd.print("Gas Analyser");

delay (3000);

lcd.clear();

lcd.print("By");

lcd.setCursor(0, 3);

lcd.print("Kanaka");

delay (3000);

lcd.clear();

}

void loop()

{

//--------- DHT11 ---------//

float humidity = dht.readHumidity();

float temp = dht.readTemperature();

float f = dht.readTemperature(true);

if (isnan(humidity) || isnan(temp) || isnan(f)) {

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

return;

float hif = dht.computeHeatIndex(f, humidity);

float hic = dht.computeHeatIndex(temp, humidity, false);

}

//--------- DHT11 ---------//

double mq135\_value = analogToPPM(analogRead(mq135));

double mq7\_value = analogToPPM(analogRead(mq7));

double mq2\_value = analogToPPM(analogRead(mq2));

Serial.printl n("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Serial.print("MQ-135 Value: "); Serial.print(mq135\_value, DEC); Serial.println(" PPM");

Serial.print("MQ-7 Value: "); Serial.print(mq7\_value, DEC); Serial.println(" PPM");

Serial.print("MQ-2 Value: "); Serial.print(mq2\_value, DEC); Serial.println(" PPM");

Serial.print("DHT-Temp: "); Serial.print(temp); Serial.println(" c");

Serial.print("DHT-Humi: "); Serial.print(humidity); Serial.println(" %");

Serial.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

lcd.setCursor(0, 0);

lcd.print("mq135");

lcd.setCursor(0, 3);

lcd.println(mq135\_value);

lcd.setCursor(7, 7);

lcd.print("PPM");

if (mq135\_value > mq135Limit) {

digitalWrite(buzzer, HIGH);

}

delay(DELAY\_IN\_SEC);

digitalWrite(buzzer, LOW);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("mq7");

lcd.setCursor(0, 3);

lcd.println(mq7\_value);

lcd.setCursor(7, 7);

lcd.print("PPM");

if (mq7\_value > mq7Limit) {

digitalWrite(buzzer, HIGH);

}

delay(DELAY\_IN\_SEC);

digitalWrite(buzzer, LOW);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("mq2");

lcd.setCursor(0, 3);

lcd.println(mq2\_value);

lcd.setCursor(7, 7);

lcd.print("PPM");

if (mq135\_value > mq2Limit) {

digitalWrite(buzzer, HIGH);

}

delay(DELAY\_IN\_SEC);

digitalWrite(buzzer, LOW);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("DHT-11");

lcd.setCursor(0, 3);

lcd.print("Temp -");

lcd.setCursor(7, 4);

lcd.println(temp);

if (temp > temperatureLimit) {

digitalWrite(buzzer, HIGH);

}

delay(DELAY\_IN\_SEC);

digitalWrite(buzzer, LOW);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("DHT-11");

lcd.setCursor(0, 3);

lcd.print("Humi -");

lcd.setCursor(7, 4);

lcd.println(humidity);

if (temp > humidityLimit) {

digitalWrite(buzzer, HIGH);

}

delay(DELAY\_IN\_SEC);

digitalWrite(buzzer, LOW);

lcd.clear();

}