//smoke sensor,gsm, lcd, bmp, dht11

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

int smokeA0=A0;

int buzzer =10;

float sensorValue;

#include <DFRobot\_DHT11.h>

DFRobot\_DHT11 DHT;

#define DHT11\_PIN 8

#include <SoftwareSerial.h>

SoftwareSerial mySerial(2,3);

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

LiquidCrystal\_I2C lcd2(0x26, 16,2);

#include <Wire.h>

#include "i2c.h"

#include "i2c\_BMP280.h"

BMP280 bmp280;

void setup()

{

   mySerial.begin(9600);

  Serial.begin(9600);

  lcd.init();                      // initialize the lcd

  lcd.backlight();

    lcd2.init();                      // initialize the lcd

  lcd2.backlight();

    pinMode(buzzer,OUTPUT);

  pinMode(smokeA0,INPUT);

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

  delay(2000);

  noTone(buzzer);

   // Setting the baud rate of GSM Module

       Serial.begin(9600);

    Serial.print("Probe BMP280: ");

    if (bmp280.initialize()) Serial.println("Sensor found");

    else

    {

        Serial.println("Sensor missing");

        while (1) {}

    }

    // onetime-measure:

    bmp280.setEnabled(0);

    bmp280.triggerMeasurement();

}

void loop()

{

  lcd.setCursor(0,0);

  lcd.backlight();

   sensorValue=analogRead(smokeA0);

  if(sensorValue > 300)

  {

    Serial.print(" | Smoke detected!\n");

    tone(buzzer,1000,200);

  }

  else

  {

     Serial.print(" | Smoke  not detected!\n");

    noTone(buzzer);

  }

    DHT.read(DHT11\_PIN);

  lcd.setCursor(0,0);

  lcd.backlight();

  lcd.print("TEMP:");

  lcd.setCursor(6,0);

  int y = DHT.temperature;

  lcd.print(y);

  lcd.setCursor(9,0);

  lcd.print("HUMI:");

   lcd.setCursor(14,0);

   int x =DHT.humidity;

  lcd.println(x);

  bmp280.awaitMeasurement();

    float pascal;

    bmp280.getPressure(pascal);

    bmp280.triggerMeasurement();

    Serial.print(" BP: ");

    Serial.print(pascal);

    Serial.print(" Pa");

    lcd.setCursor(0,1);

    lcd.print(" BP: ");

    lcd.setCursor(4,1);

    lcd.print(pascal);

    lcd.setCursor(11,1);

    lcd.print("Pa");

    if (Serial.available()>0)

      mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode

     delay(1000);  // Delay of 1 second

     mySerial.println("AT+CMGS=\"+919949823605\"\r"); // Replace x with mobile number

     delay(1000);

     mySerial.println("Tempurature: ");// The SMS text you want to send

     mySerial.println(y);

     mySerial.println("Humidity:");

     mySerial.println(x);

     mySerial.println("Pressure:");

     mySerial.println(pascal);

     delay(100);

     mySerial.println((char)26);// ASCII code of CTRL+Z for saying the end of sms to  the module

      delay(10000);

 if (mySerial.available()>0)

   Serial.write(mySerial.read());

   float sensorValue = analogRead(A1);

  Serial.print("Analog Value =");

  Serial.println(sensorValue);

  float voltage = (sensorValue / 1024) \* 5; //Arduino ADC resolution 0-1023

  Serial.print("Voltage =");

  Serial.print(voltage);

  Serial.println(" V");

  float wind\_speed = mapfloat(voltage, 0.4, 2, 0, 32.4);

  float speed\_mph = ((wind\_speed \*3600)/1609.344);

  Serial.print("Wind Speed =");

  Serial.print(wind\_speed);

  Serial.println("m/s");

  Serial.print(speed\_mph);

  Serial.println("mph");

  lcd2. setCursor (0, 0);

  lcd2.print ("Wind Speed");

//Here cursor is placed on second line

lcd2. setCursor (0, 1);

lcd2.print (wind\_speed);

lcd2.print ("m/s");

  Serial.println(" ");

  delay(300);

}

float mapfloat(float x, float in\_min, float in\_max, float out\_min, float out\_max)

{

  return (x - in\_min) \* (out\_max - out\_min) / (in\_max - in\_min) + out\_min;

}