



Universidad
Zaragoza

ANEXOS

Trabajo Fin de Grado

DISEÑO DE SENSOR DE CO₂ CON ARDUINO

DESING OF A CO₂ SENSOR WITH ARDUINO

Autor

Javier Casas Ros

Directores

Raúl Igual Catalán
María Abián Vicén

Facultad / Escuela

Escuela Universitaria Politécnica de Teruel, Universidad de Zaragoza

Año
2021

ANEXOS

ANEXO I

En este anexo se recoge el código fuente del programa en C de la placa Arduino.

```
////////////////////////////////////
///JAVIER CASAS ROS 737385          //
/// ESCUELA UNIVERSITARIA POLITECNICA DE TERUEL, UNIVERSIDAD DE
ZARAGOZA  //
///TRABAJO FIN DE GRADO            //
///MEDIDOR DE CO2                  //
////////////////////////////////////
#include <LiquidCrystal.h>
#include <MHZ19.h>
#include <SoftwareSerial.h>

// Pin RX Arduino conectado al pin TX del MHZ19
#define RX_PIN 13
// Pin TX Arduino conectado al pin RX del MHZ19
#define TX_PIN 12
//LEDS
const int ledrojo=10;
const int ledverde=9;
const int ledamarillo=8;
// Objeto para sensor MHZ19
MHZ19 myMHZ19;
// Serial requerido por el MHZ19
SoftwareSerial mySerial(RX_PIN, TX_PIN);

// Pantalla LCD de 16x2
LiquidCrystal lcd(7,6,5,4,3,2);

// Contador para temporizar las mediciones
unsigned long timer = 0;

void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
  myMHZ19.begin(mySerial);
  // Turn auto calibration ON (OFF autoCalibration(false))
  myMHZ19.autoCalibration();
  lcd.begin(16,2);
  pinMode(ledrojo ,OUTPUT);
  pinMode(ledverde ,OUTPUT);
  pinMode(ledamarillo ,OUTPUT);
}

void loop() {
  // Mostrar el nivel de CO2 en el monitor serie
  int nivelCO2 = myMHZ19.getCO2();
```

```

//encender o apagar leds segun el nivel de ppm
if(nivelCO2 <= 800){
    digitalWrite(ledverde ,HIGH);
    digitalWrite(ledrojo ,LOW);
    digitalWrite(ledamarillo ,LOW);
    lcd.setCursor(0, 1);
    lcd.print("BUENO ");
}
    if(nivelCO2 > 800 && nivelCO2 < 1000){
        digitalWrite(ledrojo ,LOW);
        digitalWrite(ledverde ,LOW);
        digitalWrite(ledamarillo ,HIGH);
        lcd.setCursor(0, 1);
        lcd.print("REGULAR ");
    }
    if(nivelCO2 >= 1000){
        digitalWrite(ledrojo ,HIGH);
        digitalWrite(ledverde ,LOW);
        digitalWrite(ledamarillo ,LOW);
        lcd.setCursor(0, 1);
        lcd.print("MALO ");
    }

// Tomar mediciones cada 2 segundos

    if (millis() - timer >= 2000) {
// Obtener la medición de CO2 actual como ppm

        // Mostrar el nivel de CO2 en el monitor serie
        Serial.print("CO2 (ppm): ");
        Serial.println(nivelCO2);

// Mostrar el nivel de CO2 en la pantalla LCD
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("CO2 (ppm): ");
        lcd.setCursor(11, 0);
        lcd.print(nivelCO2);

        // Almacenar una referencia al momento actual para controlar tiempo transcurrido
        timer = millis();

    }
}

```

ANEXO II

En este anexo se recoge el código de referencia tomado como base para programar la placa Arduino en este proyecto.

```
#include<LiquidCrystal_I2C.h>
#include<MHZ19.h>
#include<SoftwareSerial.h>

// Pin RX Arduino conectado al pin TX del MHZ19
#define RX_PIN 7
// Pin TX Arduino conectado al pin RX del MHZ19
#define TX_PIN 6

// Objeto para sensor MHZ19
MHZ19 myMHZ19;
// Serial requerido por el MHZ19
SoftwareSerialmySerial(RX_PIN, TX_PIN);

// Pantalla LCD de 16x2
LiquidCrystal_I2Clcd(0x27,16,2);

// Contador para temporizar las mediciones
unsigned long timer =0;

voidsetup(){
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  mySerial.begin(9600);
  myMHZ19.begin(mySerial);
  // Turn auto calibration ON (OFF autoCalibration(false))
  myMHZ19.autoCalibration();
}
```

```

voidloop(){
// Tomar mediciones cada 2 segundos
if(millis()- timer >=2000){

// Obtener la medición de CO2 actual como ppm
int nivelCO2 = myMHZ19.getCO2();

// Mostrar el nivel de CO2 en el monitor serie
Serial.print("CO2 (ppm): ");
Serial.println(nivelCO2);

// Mostrar el nivel de CO2 en la pantalla LCD
lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("CO2 (ppm): ");
lcd.setCursor(11,0);
    lcd.print(nivelCO2);

// Obtener la temperatura actual en grados Celsius
int8_t temperatura = myMHZ19.getTemperature();

// Mostrar la temperatura en el monitor serie
Serial.print("Temperatura (C): ");
Serial.println(temperatura);

// Mostrar la temperatura en la pantalla LCD
    lcd.setCursor(0,1);
    lcd.print("Temp. (C): ");
    lcd.setCursor(11,1);
    lcd.print(temperatura);

// Almacenar una referencia al momento actual para controlar
tiempo transcurrido
    timer =millis();
}

```

}