

/\*optimizado para el Arduino UNO R3. Todos los datos recibidos son identificados con un caracter y se envian por el serial en formato JSON\*/

```
#include "Sseed_BME280.h"
#include "MutichannelGasSensor.h"
#include <Wire.h>
```

```
BME280 bme280;
```

```
int pin = 8;//entrada particulas
int ledpwm = 9;//salida pwm
int led = 4;
int ledestraccion = 5;
int alarma = 3;
int ledalarma = 2;
int option;
int value = 0;
unsigned long duration;
unsigned long starttime;
unsigned long sampletime_ms = 30000;//sampe 30s ;
unsigned long lowpulseoccupancy = 0;
float ratio = 0;
float concentration = 0;
String inData ;
```

```
void setup()
{
  Serial.begin(115200); // abrir el puerto
  //pinMode(ledpwm, PWM);
  gas.begin(0x04);//the default I2C address of the slave is 0x04
  gas.powerOn();
  //pwmWrite(ledpwm, 0);//inicializacion
  pinMode(ledestraccion, OUTPUT);
  pinMode(led, OUTPUT);
  pinMode(pin,INPUT);
  pinMode(alarma,INPUT);
  pinMode(ledalarma, OUTPUT);
  starttime = millis();//get the current time;
  if(!bme280.init()){
    Serial.println("Device error!");
  }
}
```

```
void loop()
{
  float tempc;
  float humidity;
  float pressure;
  float altitude;
  float no2;
  float co;
```

```

float nh3;
float ch4;
float c4h10;
float c3h8;
float h2;

duration = pulseIn(pin, LOW);
lowpulseoccupancy = lowpulseoccupancy+duration;

tempc = bme280.getTemperature();
humidity = bme280.getHumidity();
pressure = bme280.getPressure();
altitude = bme280.calcAltitude(pressure);

/*leemos el estado del pin alarma y enviamos en JSON por el port serial*/
value = digitalRead(alarma); //lectura digital de pin
if (value == HIGH) {
  digitalWrite(ledalarma, HIGH);
  //enviar alarma activada en JSON
  Serial.print(F("{\"alar\": \"}"));
  Serial.print(1);
  Serial.println(F("{}"));
}

/*leer datos de entrada por el port serial en formato caracter
y controlar pines y recibir datos*/
char inChar = Serial.read();
if (inChar == 'c'){
  digitalWrite(led, HIGH);
}
if (inChar == 'x'){
  digitalWrite(led, LOW);
}
if (inChar == 'a'){
  digitalWrite(ledestraccion, HIGH);
}
if (inChar == 'b'){
  digitalWrite(ledestraccion, LOW);
}
if (inChar == 'd'){
  digitalWrite(ledalarma, LOW);
}
if (inChar == 'y') {
  int data = Serial.parseInt();//recibe el valor del slider y hace el parser
  int data1 =data*40.95;//para adaptar el valor del slider al 100%
  //int data2 = map(data1, 0, 4095, 0, 65535);
  //pwmWrite(ledpwm, data2);
  analogWrite(ledpwm, data1);//en caso de no poder utilizar pwm
}

if ((millis()-starttime) > samplettime_ms)/*si el tiempo de muestreo es = 30s*/
{

```

```

ratio = lowpulseoccupancy/(sampletime_ms*10.0); // porcentaje entero
concentration = 1.1*pow(ratio,3)-3.8*pow(ratio,2)+520*ratio+0.62; /*concentracion
obtenida en base a los parametros de la grafica del humo*/
lowpulseoccupancy = 0;
starttime = millis();

nh3 = gas.measure_NH3(); //amoniac
co = gas.measure_CO(); //monoxido de carbono
no2 = gas.measure_NO2();//dioxido de nitrogeno
ch4 = gas.measure_CH4();//metano
c4h10 = gas.measure_C4H10();//iso-butano
c3h8 = gas.measure_C3H8();//propano
h2 = gas.measure_H2();//hidrogeno

//enviar datos de todos los sensores, en formato JSON por el port serial despues de 30s
Serial.print(F("{\"Temperatura\": "));
Serial.print(tempc);
Serial.print(F(", \"Humedad\": "));
Serial.print(humidity);
Serial.print(F(", \"Ratio\": "));
Serial.print(ratio);
Serial.print(F(", \"Concentracion\": "));
Serial.print(concentration);
Serial.print(F(", \"Pressure\": "));
Serial.print(pressure);
Serial.print(F(", \"NH3\": "));
Serial.print(nh3);
Serial.print(F(", \"CO\": "));
Serial.print(co);
Serial.print(F(", \"NO2\": "));
Serial.print(no2);
Serial.print(F(", \"CH4\": "));
Serial.print(ch4);
Serial.print(F(", \"C4H10\": "));
Serial.print(c4h10);
Serial.print(F(", \"C3H8\": "));
Serial.print(c3h8);
Serial.print(F(", \"H2\": "));
Serial.print(h2);
Serial.println(F("}"));
}
}

```