Air Quality Monitor (PM 2.5, Formaldehyde, Temperature & Humidity Sensor) SKU: SEN0233

Introduction

This is a multifunction air quality monitor that can be detected not only the concentration of particles and formaldehyde but also the value of temperature and humidity. As the particle concentration sensor it can be used to obtain the number of suspended particles in the air by laser. It also integrates in it an electrochemical probe for formaldehyde and an one- chip sensor for temperature and humidity. The sensor can be inserted into variable instruments related to the environment parameters in the air or other environmental improvement equipments to provide correct data in time.

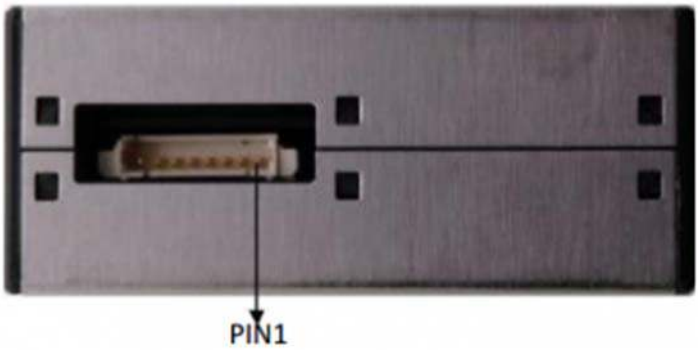
Features

- Zero false alarm rate
- Real-time response
- Correct data
- Minimum distinguishable particle diameter :0.3 micrometer
- High anti-interference performance because of the patent structure of six sides shielding
- Optional direction of air inlet and outlet in order to adapt the different design
- Concentration of Formaldehyde output
- Temperature and Humidity output

Specification

- Operating Voltage: 5.0VDC
- Active Current: 100 mA
- Standby Current: ≤200 uA
- Interface Level: L <0.8V@3.3V; H> 2.7V@3.3V
- Particle Range of measurement: 0.3 ~ 1.0μm; 1.0 ~ 2.5μm; 2.5 ~ 10μm
- Particle Counting Efficiency: 50%@0.3μm 98%@>=0.5μm
- Particle Effective Range (PM2.5 standard): 0 ~ 500μg/m
- Particle Maximum Range (PM2.5 standard)*: >0 ~ 2000ug / m³
- Particle Resolution: 1ug / m³
- Particle Maximum Consistency Error (PM2.5 standard data)*:
- ±10%@100~500μg/m³
- ±10μg/m³@0~100μg/m³
- Particle Standard Volume: 0.1 L
- Formaldehyde Effective Range: 0 ~ 1 mg / m³
- Formaldehyde Maximum Range: 0 ~ 2 mg / m³
- Formaldehyde Resolution: 0.001
- Formaldehyde Maximum Consistency Error: <± 5%
- Temperature Maximum Range: -10~50 ℃
- Temperature Resolution: 0.1 ℃
- Temperature Maximum Error: ± 0.5 ℃
- Humidity Maximum Range: 0 ~ 99%
- Humidity Resolution: 0.1%
- Humidity Maximum Error: ± 2%
- Default Baud rate: 9600bps
- Total Response time: ≤ 10s
- Operating Temperature Range: -10 ~ +60 ℃
- Working Humidity Range: 0~99%
- MTTF: ≥ 3 years
- Dimension: 50 * 38 * 21 mm/1.97*1.50*0.83 inches
- Weight: 50g

Board Overview



SEN0233_Interface

Num	Label	Description
PIN1	VCC	5V
PIN2	GND	GND
PIN3	SET	Set pin /TTL level@3.3V, high level or suspending is normal working status, while low level is sleeping mode.
PIN4	RXD	Serial port receiving pin/TTL level@3.3V
PIN5	TXD	Serial port sending pin/TTL level@3.3V
PIN6	RESET	Module reset signal /TTL level@3.3V, low reset.
PIN7	NC	
PIN8	NC	

Tutorial

In this section, we'll demonstrate two examples about how to use this sensor with different Mcirocontrollers.

Requirements

- **Hardware**
- DFRduino Leonardo (or similar) x 1
- Gravity IO Expansion Shield V7.1 x1
- Air Quality Monitor x1
- M-M/F-M/F-F Jumper wires

- **Software**
- Arduino IDE, Click to Download Arduino IDE from Arduino®

Connection Diagram

Sample Code1——Arduino Leonardo

- This code uses Leonardo HardwareSerial1 as communication port.

```
/*
   @file SEN0233.ino
   @brief Air Quality Monitor (PM 2.5, HCHO, Temperature & Humidity)
   @n Get the module here: https://www.dfrobot.com/product-1612.html
   @n This example is to detect formaldehyde, PM2.5, temperature and humidity
in the environment.

   @copyright  [DFRobot](http://www.dfrobot.com), 2017
   @copyright GNU Lesser General Public License

   @author [lijun](ju.li@dfrobot.com)
   @version  V1.0
   @date  2017-04-21
*/

char col;
unsigned int PMSa = 0,FMHDSa = 0,TPSa = 0,HDSa = 0,PMSb = 0,FMHDSb = 0,TPSb =
0,HDSb = 0;
unsigned int PMS = 0,FMHDS = 0,TPS = 0,HDS = 0,CR1 = 0,CR2 = 0;
unsigned char buffer_RTT[40]={};   //Serial buffer; Received Data
char tempStr[15];
void setup()
{
  Serial.begin(115200);
  Serial1.begin(9600);
}

void loop()
```

```
{
  while(!Serial1.available());
  while(Serial1.available()>0)   //Data check: weather there is any Data in S
erial1
  {
    for(int i=0;i<40;i++)
    {
      col =Serial1.read();
      buffer_RTT[i]=(char)col;
      delay(2);
    }

    Serial1.flush();

    CR1 =(buffer_RTT[38]<<8) + buffer_RTT[39];
    CR2 = 0;
    for(int i=0;i<38;i++)
      CR2 += buffer_RTT[i];
    if(CR1 == CR2)                 //Check
    {
      PMSa=buffer_RTT[12];          //Read PM2.5 High 8-bit
      PMSb=buffer_RTT[13];          //Read PM2.5 Low 8-bit
      PMS=(PMSa<<8)+PMSb;           //PM2.5 value
      FMHDSa=buffer_RTT[28];          //Read Formaldehyde High 8-bit
      FMHDSb=buffer_RTT[29];          //Read Formaldehyde Low 8-bit
      FMHDS=(FMHDSa<<8)+FMHDSb;     //Formaldehyde value
      TPSa=buffer_RTT[30];          //Read Temperature High 8-bit
      TPSb=buffer_RTT[31];          //Read Temperature Low 8-bit
      TPS=(TPSa<<8)+TPSb;         //Temperature value
      HDSa=buffer_RTT[32];          //Read Humidity High 8-bit
      HDSb=buffer_RTT[33];          //Read Humidity Low 8-bit
      HDS=(HDSa<<8)+HDSb;       //Humidity value
    }
    else
    {
```

```
        PMS = 0;
        FMHDS = 0;
        TPS = 0;
        HDS = 0;
      }
    }

    Serial.println("----------------------uart--------------------------");
    Serial.print("Temp : ");
    sprintf(tempStr,"%d%d.%d",TPS/100,(TPS/10)%10,TPS%10);
    Serial.print(tempStr);
    Serial.println("C");                //Serial pring Temperature
    Serial.print("RH   : ");
    sprintf(tempStr,"%d%d.%d",HDS/100,(HDS/10)%10,HDS%10);
    Serial.print(tempStr);             //Serial print humidity
    Serial.println(" %");                //"%"
    Serial.print("HCHO : ");
    Serial.print(FMHDS);
    Serial.println(" ug/m3");         // Serial print formaldehyde, unit: ug/m³
    Serial.print("PM2.5: ");
    Serial.print(PMS);
    Serial.println(" ug/m3");        // Serial print PM2.5, unit: ug/m³
    Serial.println();
}
```

Sample Code2——Arduino UNO (Software Serial)

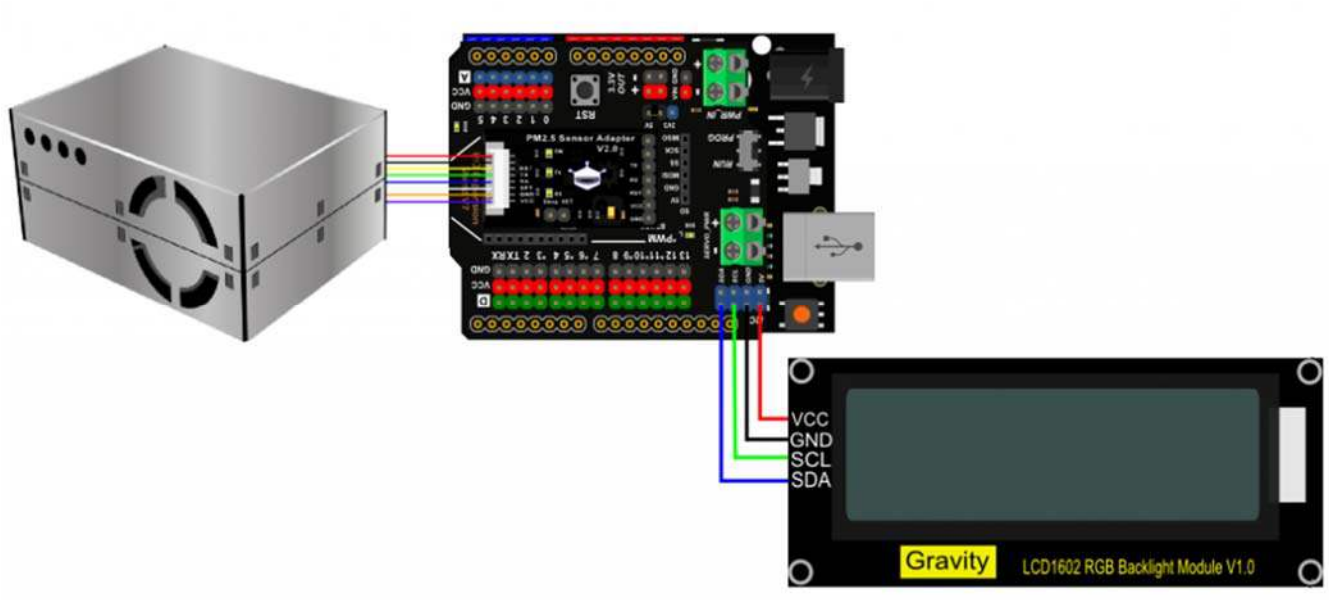
```
/*
   @file SEN0233.ino
   @brief Air Quality Monitor (PM 2.5, HCHO, Temperature & Humidity)
   @n Get the module here: https://www.dfrobot.com/product-1612.html
   @n This example is to detect formaldehyde, PM2.5, temperature and humidity
in the environment.
```

```
   @copyright  [DFRobot](http://www.dfrobot.com), 2017
   @copyright GNU Lesser General Public License

   @author [lijun](ju.li@dfrobot.com)
   @version  V1.0
   @date  2017-03-01
*/
#include <SoftwareSerial.h>
SoftwareSerial Serial1(10, 11); // Software RX, TX
char col;
unsigned int PMSa = 0,FMHDSa = 0,TPSa = 0,HDSa = 0,PMSb = 0,FMHDSb = 0,TPSb =
0,HDSb = 0;
unsigned int PMS = 0,FMHDS = 0,TPS = 0,HDS = 0,CR1 = 0,CR2 = 0;
unsigned char buffer_RTT[40]={};     //Serial buffer; Received Data
char tempStr[15];
void setup()
{
  Serial.begin(115200);
  Serial1.begin(9600);


}


void loop()
{
  while(!Serial1.available());
  while(Serial1.available()>0)        //Data check: weather there is any Data
in Serial1
  {
    for(int i=0;i<40;i++)
    {
      col =Serial1.read();
      buffer_RTT[i]=(char)col;
      delay(2);
    }
```

```
    Serial1.flush();

    CR1 =(buffer_RTT[38]<<8) + buffer_RTT[39];
    CR2 = 0;
    for(int i=0;i<38;i++)
      CR2 += buffer_RTT[i];
    if(CR1 == CR2)                          //Check
    {
PMSa=buffer_RTT[12];          //Read PM2.5 High 8-bit
      PMSb=buffer_RTT[13];          //Read PM2.5 Low 8-bit
      PMS=(PMSa<<8)+PMSb;           //PM2.5 value
      FMHDSa=buffer_RTT[28];          //Read Formaldehyde High 8-bit
      FMHDSb=buffer_RTT[29];          //Read Formaldehyde Low 8-bit
      FMHDS=(FMHDSa<<8)+FMHDSb;     //Formaldehyde value
      TPSa=buffer_RTT[30];          //Read Temperature High 8-bit
      TPSb=buffer_RTT[31];          //Read Temperature Low 8-bit
      TPS=(TPSa<<8)+TPSb;         //Temperature value
      HDSa=buffer_RTT[32];          //Read Humidity High 8-bit
      HDSb=buffer_RTT[33];          //Read Humidity Low 8-bit
      HDS=(HDSa<<8)+HDSb;       //Humidity value
    }
    else
    {
      PMS = 0;
      FMHDS = 0;
      TPS = 0;
      HDS = 0;
    }
  }

  Serial.println("-----------------------uart--------------------------");
  Serial.print("Temp : ");
  sprintf(tempStr,"%d%d.%d",TPS/100,(TPS/10)%10,TPS%10);
```

```
    Serial.print(tempStr);
    Serial.println(" C");                //Serial pring Temperature
    Serial.print("RH    : ");
    sprintf(tempStr,"%d%d.%d",HDS/100,(HDS/10)%10,HDS%10);
    Serial.print(tempStr);           //Serial print humidity
    Serial.println(" %");
    Serial.print("HCHO  : ");
    Serial.print(FMHDS);
    Serial.println(" ug/m3");        //Serial print formaldehyde, unit: ug/m³
    Serial.print("PM2.5: ");
    Serial.print(PMS);
    Serial.println(" ug/m3");        //Serial print PM2.5, unit: ug/m³
    Serial.println();
}
```

Expected Results



Application with LCD

Requirements

- **Hardware**
- DFRduino UNO (or similar) x 1
- Gravity IO Expansion Shield V7.1 x1
- Air Quality Monitor x1
- Gravity: I2C 16x2 Arduino LCD with RGB Backlight Display x1
- M-M/F-M/F-F Jumper wires

- **Software**
- Arduino IDE, Click to Download Arduino IDE from Arduino®

Connection Diagram

Sample Code

Please download the LCD library: **DFRobot RGB LCD**. How to install Libraries in Arduino IDE

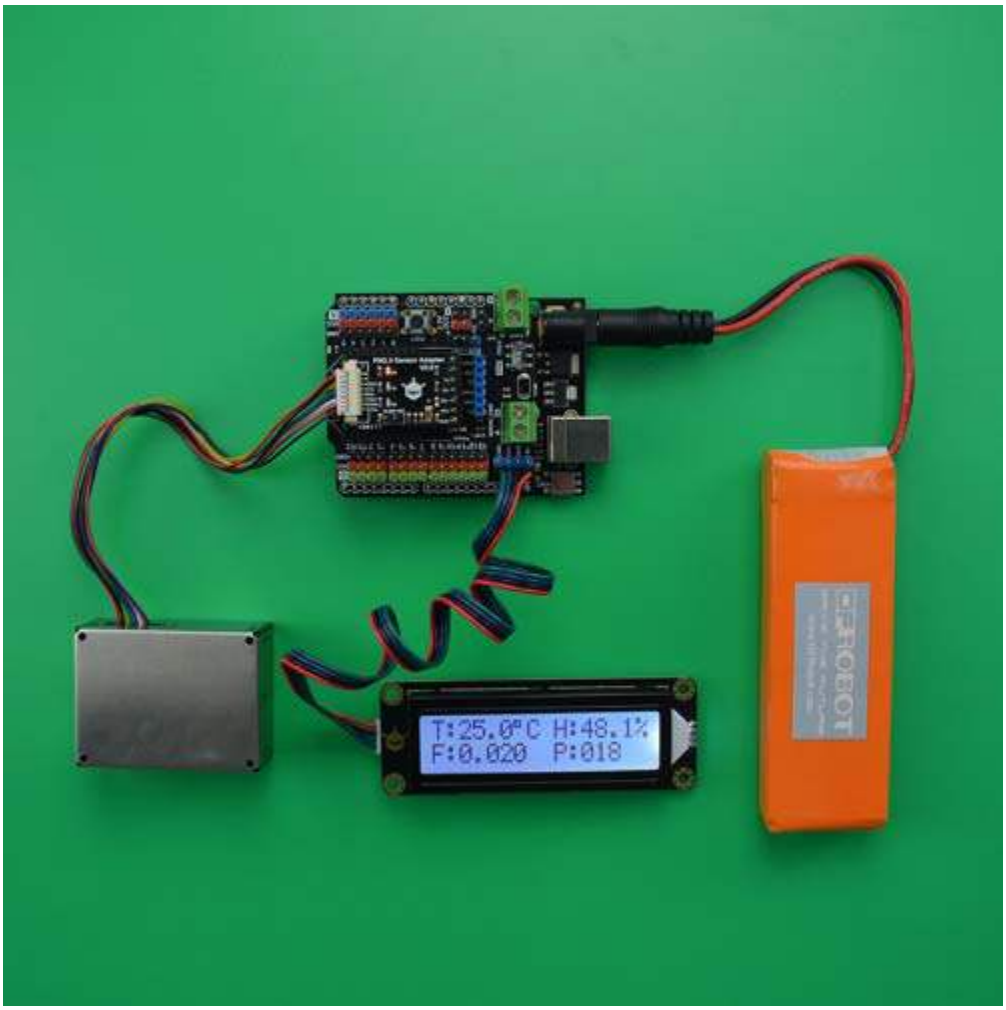
```
/*
   @file SEN0233.ino
   @brief Air Quality Monitor (PM 2.5, HCHO, Temperature & Humidity)
   @n Get the module here: https://www.dfrobot.com/product-1612.html
   @n This example is to detect formaldehyde, PM2.5, temperature and humidity
in the environment.

   @copyright  [DFRobot](http://www.dfrobot.com), 2017
   @copyright GNU Lesser General Public License

   @author [lijun](ju.li@dfrobot.com)
   @version  V1.0
   @date  2017-07-028
*/

#include <Wire.h>
#include "DFRobot_RGBLCD.h"
```

```
char col;
unsigned int PMSa = 0, FMHDSa = 0, TPSa = 0, HDSa = 0, PMSb = 0, FMHDSb = 0,
TPSb = 0,
             HDSb = 0,PMS = 0,TPS= 0,  HDS = 0, CR1 = 0, CR2 = 0,FMHDS = 0;
unsigned char buffer_RTT[40] = {};    //Serial buffer; Received Data
char tempStr[15];
DFRobot_RGBLCD lcd(16, 2);  //RGB address; 16 characters and 2 lines of show


void setup()
{
  Serial.begin(9600);
  lcd.init();                                    //Initialize LCD RGB Screen
  lcd.setRGB(0, 255, 0);               //Set backlight color
  lcd.setCursor(0, 0 );                   //Start from (0, 0)
  lcd.print("T:");
  lcd.setCursor(9, 0 );
  lcd.print("H:");
  lcd.setCursor(0, 1 );
  lcd.print("F:");
  lcd.setCursor(9, 1 );
  lcd.print("P:");
}


void loop()
{
  while (Serial.available() > 0)       //Data check: weather there is any Dat
a in Serial1
  {
    for (int i = 0; i < 40; i++)            //Read Serial port data
    {
      col = Serial.read();
      buffer_RTT[i] = (char)col;
      delay(2);
    }
```

```
    Serial.flush();
    CR1 = (buffer_RTT[38] << 8) + buffer_RTT[39];
    CR2 = 0;
    for (int i = 0; i < 38; i++)
      CR2 += buffer_RTT[i];
    if (CR1 == CR2)                        //Check
    {
      PMSa = buffer_RTT[12];       //Read PM2.5 High 8-bit
      PMSb = buffer_RTT[13];       //Read PM2.5 Low 8-bit
      PMS = (PMSa << 8) + PMSb;   //PM2.5 value
      FMHDSa = buffer_RTT[28];      //Read Formaldehyde High 8-bit
      FMHDSb = buffer_RTT[29];      //Read Formaldehyde Low 8-bit
      FMHDS = (FMHDSa << 8) + FMHDSb; //Formaldehyde value
      TPSa = buffer_RTT[30];       //Read Temperature High 8-bit
      TPSb = buffer_RTT[31];       ///Read Temperature Low 8-bit
      TPS = (TPSa << 8) + TPSb;    //Temperature value
      HDSa = buffer_RTT[32];     //Read Humidity High 8-bit
      HDSb = buffer_RTT[33];       //Read Humidity Low 8-bit
      HDS = (HDSa << 8) + HDSb;   //Humidity value
    }
    else
    {
      PMS = 0;
      FMHDS = 0;
      TPS = 0;
      HDS = 0;
    }
  }
  lcd.setCursor(2, 0 );
  sprintf(tempStr,"%d%d.%d",TPS/100,(TPS/10)%10,TPS%10);  //Temperature displ
ay
  lcd.print(tempStr);
  lcd.write(0xdf);               //°
  lcd.print('C');                  //C
  lcd.setCursor(11, 0 );
```

```
  sprintf(tempStr,"%d%d.%d",HDS/100,(HDS/10)%10,HDS%10);    //Humidity display
  lcd.print(tempStr);
  lcd.print('%');                //%
  lcd.setCursor(2, 1 );
  lcd.print((float)FMHDS/1000);  //print formaldehyde, unit: ug/m³
  lcd.print((int)FMHDS%10);
  lcd.setCursor(11, 1 );
  sprintf(tempStr,"%d%d%d",PMS/100,(PMS/10)%10,PMS%10);  //print PM2.5, unit:
ug/m³
  lcd.print(tempStr);
}
```

Expected Results



https://www.dfrobot.com/wiki/index.php/Air_Quality_Monitor_(PM_2.5,_Formaldehyde,_Temperature_%26_Humidity_Sensor)_SKU:_SEN02338-23-18