

Grove - Laser PM2.5 Sensor (HM3301) SKU: 101020613



The Grove - Laser PM2.5 Sensor (HM3301) is a new generation of laser dust detection sensor, which is used for continuous and real-time detection of dust in the air.

Different from the pumping dust detection sensor, the HM-3301 innovatively uses fan blades to drive air, and the air flowing through the detection chamber is used as a test sample to perform real-time and continuous test on dust of different particle sizes in the air.

This module is suitable for dust detectors, intelligent air purifiers, intelligent air conditioners, intelligent ventilation fans, air quality testing, haze meters, environmental monitoring and relative products and applications

Features

- High sensitivity on dust particles of 0.3 μm or greater
- Real-time & continuous detection of dust concentration in the air
- Based on laser light scattering technology, readings are accurate, stable, and consistent
- Low noise
- Ultra-low power consumption

Specification

Item	Value
Operating Voltage	3.3V / 5V
Operating temperature	-10~60°C
Operating humidity	10%~90%RH (non-condensing)
Partical size	3 channels-- 2.5µm, 5µm, 10µm
Range (PM2.5 standard value)	1~500µg/ m ³ (Effective range) 1000 µg/ m ³ (Maximum range)
Resolution	Concentration: 1µg/ m ³ Counting concentration: 1s/0.1L
Stability time	30 seconds after power-on
Interface	I2C
I2C address	0x40

Typical applications

- Air purifier / air conditioner
- Air quality testing equipment
- Industrial PM value analysis
- Dust and smoke detection and analysis
- Real-time PM2.5, PM10, TSP detector
- Multichannel particle counter
- Environmental testing equipment

Hardware Overview

Pin Out



- 4 GND: connect this module to the system GND
- 3 VCC: you can use 5V or 3.3V for this module
- 2 SDA: I²C serial data
- 1 SCL: I²C serial clock

SET

Set the pin/TTL level @3.3V, high or floating for normal operation mode, low level for sleep mode

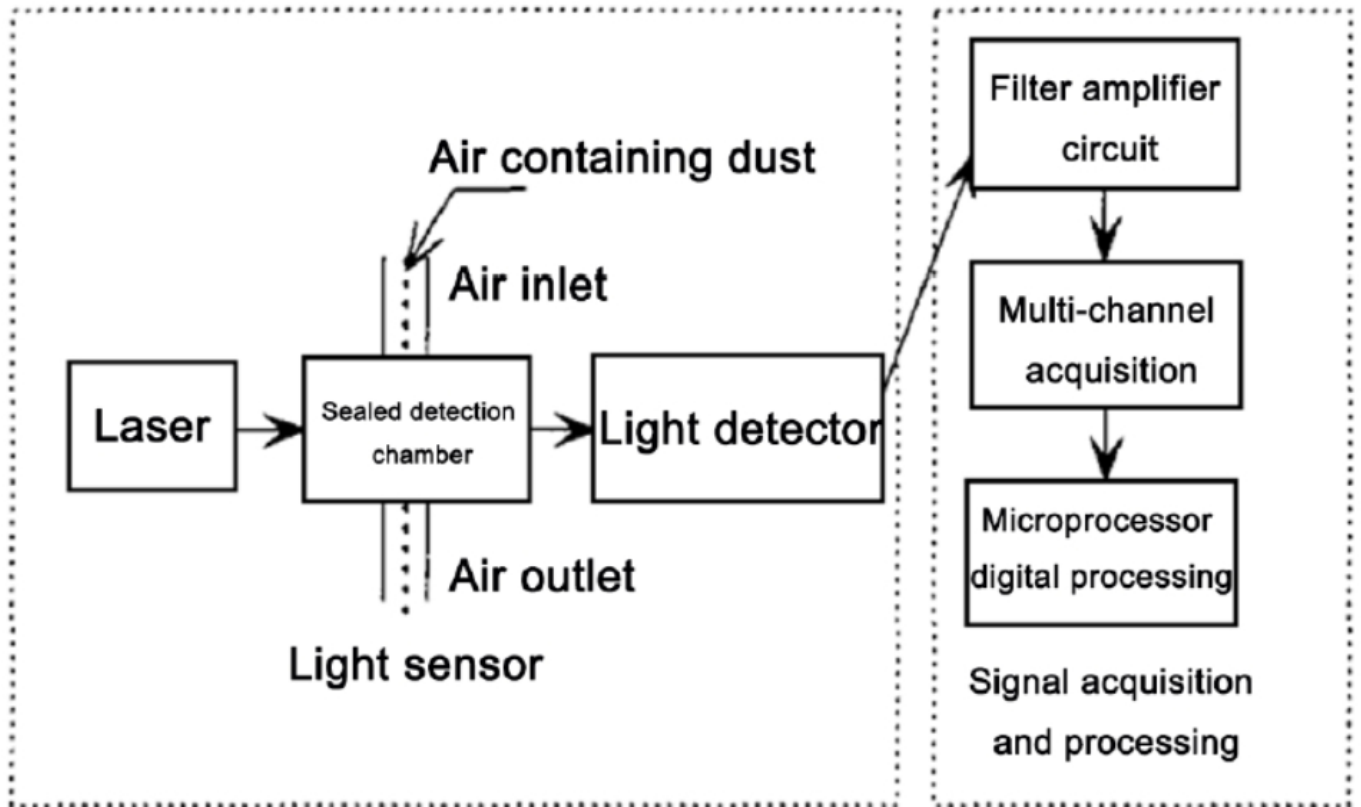
RESET

Module reset signal / TTL level @3.3V, low reset


Working Principle

The HM-3301 Dust Sensor is based on the advanced Mie scattering theory. When light passes through particles with quantity same as or larger than wavelength of the light, it will produce light scattering. The scattered light is concentrated to a highly sensitive photodiode, which is then amplified and analyzed by a circuit. With specific mathematical model and algorithm, the count concentration and mass concentration of the dust particles is obtained.

The HM-3301 dust sensor is composed of main components such as a fan, an infrared laser source, a condensing mirror, a photosensitive tube, a signal amplifying circuit and a signal sorting circuit.



Platforms Supported

Arduino	Raspberry Pi	BeagleBone	Wio	LinkIt ONE
				

!!!Caution The platforms mentioned above as supported is/are an indication of the module's software or theoretical compatibility. We only provide software library or code examples for Arduino platform in most cases. It is not possible to provide software library / demo code for all possible MCU platforms. Hence, users have to write their own software library.

Getting Started

Play With Arduino

Hardware

Materials required

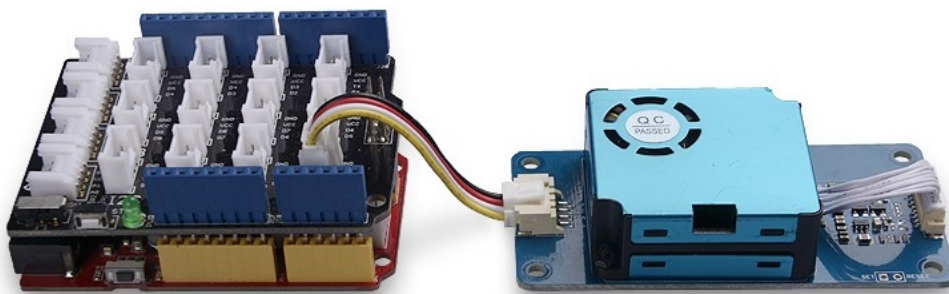
Seeeduino V4.2	Base Shield	Grove - Laser PM2.5 Sensor (HM3301)
----------------	-------------	-------------------------------------



!!!note **1** Please plug the USB cable gently, otherwise you may damage the port. Please use the USB cable with 4 wires inside, the 2 wires cable can't transfer data. If you are not sure about the wire you have, you can click [here](#) to buy

****2**** Each Grove module comes with a Grove cable when you buy. In case you lose the Grove cable, you can click [\[here\]\(https://www.seeedstudio.com/Grove-Universal-4-Pin-Buckled-20cm-Cable-%285-PCs-pack%29-p-936.html\)](https://www.seeedstudio.com/Grove-Universal-4-Pin-Buckled-20cm-Cable-%285-PCs-pack%29-p-936.html) to buy.

- **Step 1.** Connect the Grove - Laser PM2.5 Sensor (HM3301) to port **I²C** of Grove-Base Shield.
- **Step 2.** Plug Grove - Base Shield into Seeeduino.
- **Step 3.** Connect Seeeduino to PC via a USB cable.



!!!Note If we don't have Grove Base Shield, We also can directly connect this module to Seeeduino as below.

Seeeduino	Grove Cable	Grove - Laser PM2.5 Sensor (HM3301)
GND	Black	GND
5V or 3.3V	Red	VCC
SDA	White	SDA

Seeeduino Grove Cable Grove - Laser PM2.5 Sensor (HM3301)

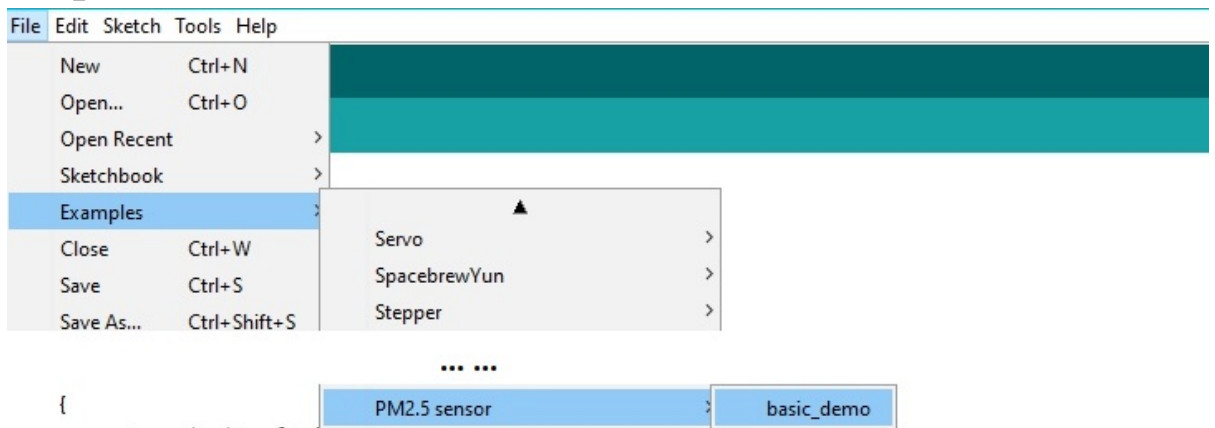
SCL Yellow SCL

Software

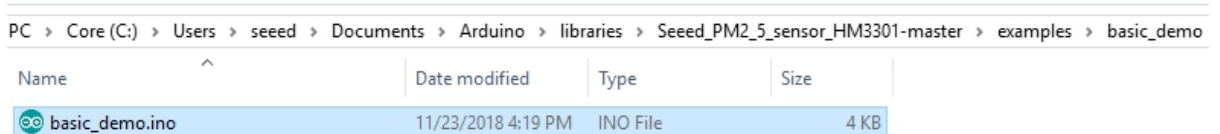
!!!Attention If this is the first time you work with Arduino, we strongly recommend you to see [Getting Started with Arduino](#) before the start.


- **Step 1.** Download the [Seeed_PM2_5_sensor_HM3301](#) Library from Github.
- **Step 2.** Refer to [How to install library](#) to install library for Arduino.
- **Step 3.** Restart the Arduino IDE. Open the example, you can open it in the following three ways :

1. Open it directly in the Arduino IDE via the path: **File --> Examples --> PM2.5 sensor --> basic_demo.**



2. Open it in your computer by click the **basic_demo.ino** which you can find in the folder **XXXX\Arduino\libraries\Seeed_PM2_5_sensor_HM3301-master\examples\basic_demo**, **XXXX** is the location you installed the Arduino IDE.



3. Or, you can just click the icon  in upper right corner of the code block to copy the following code into a new sketch in the Arduino IDE.

```
#include "Seeed_HM330X.h"

#ifdef ARDUINO_SAMD_VARIANT_COMPLIANCE
  #define SERIAL SerialUSB
#else
  #define SERIAL Serial
#endif
```

```

HM330X sensor;
u8 buf[30];

const char *str[]={ "sensor num: ", "PM1.0 concentration(CF=1,Standard particulate
matter,unit:ug/m3): ",
                    "PM2.5 concentration(CF=1,Standard particulate
matter,unit:ug/m3): ",
                    "PM10 concentration(CF=1,Standard particulate
matter,unit:ug/m3): ",
                    "PM1.0 concentration(Atmospheric environment,unit:ug/m3): ",
                    "PM2.5 concentration(Atmospheric environment,unit:ug/m3): ",
                    "PM10 concentration(Atmospheric environment,unit:ug/m3): ",
                    };

err_t print_result(const char* str,u16 value)
{
    if(NULL==str)
        return ERROR_PARAM;
    SERIAL.print(str);
    SERIAL.println(value);
    return NO_ERROR;
}

/*parse buf with 29 u8-data*/
err_t parse_result(u8 *data)
{
    u16 value=0;
    err_t NO_ERROR;
    if(NULL==data)
        return ERROR_PARAM;
    for(int i=1;i<8;i++)
    {
        value = (u16)data[i*2]<<8|data[i*2+1];
        print_result(str[i-1],value);
    }
}

err_t parse_result_value(u8 *data)
{
    if(NULL==data)
        return ERROR_PARAM;
    for(int i=0;i<28;i++)
    {
        SERIAL.print(data[i],HEX);
        SERIAL.print(" ");
        if((0==(i)%5)|| (0==i))
        {
            SERIAL.println(" ");
        }
    }
    u8 sum=0;
    for(int i=0;i<28;i++)

```

```

    {
        sum+=data[i];
    }
    if(sum!=data[28])
    {
        SERIAL.println("wrong checkSum!!!!");
    }
    SERIAL.println(" ");
    SERIAL.println(" ");
    return NO_ERROR;
}

/*30s*/
void setup()
{
    SERIAL.begin(115200);
    delay(100);
    SERIAL.println("Serial start");
    if(sensor.init())
    {
        SERIAL.println("HM330X init failed!!!");
        while(1);
    }
}

void loop()
{
    if(sensor.read_sensor_value(buf,29))
    {
        SERIAL.println("HM330X read result failed!!!");
    }
    parse_result_value(buf);
    parse_result(buf);
    SERIAL.println(" ");
    SERIAL.println(" ");
    SERIAL.println(" ");
    delay(5000);
}

```

!!!Attention The library file may be updated. This code may not be applicable to the updated library file, so we recommend that you use the first two methods.

- **Step 4.** Upload the demo. If you do not know how to upload the code, please check [How to upload code](#).
- **Step 5.** Open the **Serial Monitor** of Arduino IDE by click **Tool-> Serial Monitor**. Or tap the +ctrl+shift+m++ key at the same time. Set the baud rate to **115200**.

!!!Success If every thing goes well, when you open the Serial Monitor it may show as below:

```
Serial start
0
FF 0 0 0 2D
0 3F 0 45 0
22 0 32 0 3B
0 0 0 0 0
0 0 0 0 0
0 0

sensor num: 0
PM1.0 concentration(CF=1,Standard particulate matter,unit:ug/m3): 45
PM2.5 concentration(CF=1,Standard particulate matter,unit:ug/m3): 63
PM10 concentration(CF=1,Standard particulate matter,unit:ug/m3): 69
PM1.0 concentration(Atmospheric environment,unit:ug/m3): 34
PM2.5 concentration(Atmospheric environment,unit:ug/m3): 50
PM10 concentration(Atmospheric environment,unit:ug/m3): 59
```

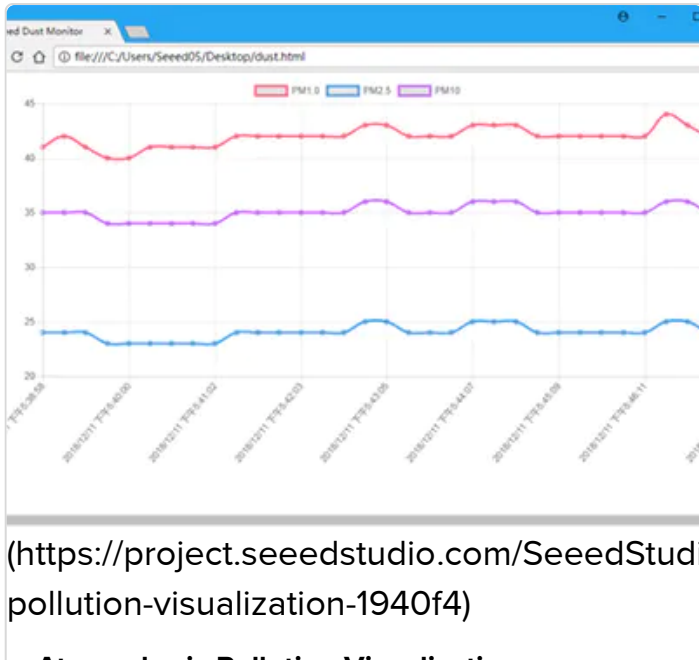
!!!Note The **standard particulate matter mass concentration value** refers to the mass concentration value obtained by density conversion of industrial metal particles as equivalent particles, and is suitable for use in industrial production workshops and the like. The **concentration of particulate matter in the atmospheric environment** is converted by the density of the main pollutants in the air as equivalent particles, and is suitable for ordinary indoor and outdoor atmospheric environments. So you can see that there are two sets of data above.

Resources

- [\[Zip\] Grove - Laser PM2.5 Sensor \(HM3301\) Eagle Files](#)
- [\[Zip\] Seeed_PM2_5_sensor_HM3301 Software Library](#)
- [\[PDF\] Datasheet HM3301](#)

Projects

Atmospheric Pollution Visualization: The air pollution problem attracts more and more attention. This time we tried to monitoring PM2.5 with Wio LTE and new Laser PM2.5 Sensor.



(<https://project.seeedstudio.com/SeedStudio/pollution-visualization-1940f4>)

Tech Support

Please do not hesitate to submit the issue into our [forum](#) or drop mail to techsupport@seeed.cc.