SKU:SEN0460 (https://www.dfrobot.com/product-2439.html)

(https://www.dfrobot.com/product-2439.html)

Introduction

As the most basic part of life, air is attracting more and more attention, for example, the most common PM2.5, PM1.0, PM10, etc. Keeping an eye on the quality of the air we breath is becoming quite important. This time, DFRobot brings you this Air Quality Sensor that can measure particulate matter like

PM2.5, PM1.0, PM10. With easy-to-use Gravity interface, the air quality sensor works well with most main-controllers. And we provide you with relevant sample codes to help you quickly build up air quality monitoring projects.

Based on the principle of laser scattering, the PM2.5 air quality sensor employs a digital universal particle sensor that can continuously collect and calculate the number of suspended particles of different sizes in the air per unit volume, which is particle concentration distribution, and then convert to concentration and output via I2C interface. The sensor can be embedded in various instruments and meters or environmental improvement equipment related to the concentration of suspended particulate matter in the air to provide timely and accurate concentration data. It is typically suitable for smoking rooms, kitchens, hot springs, bathrooms and other environments.

How does the sensor work?

This sensor adopts the principle of laser scattering, that is, to make the laser irradiate the suspended particles in the air to produce scattering, and at the same time collect the scattered light at a certain angle, and obtain the curve of the intensity of the scattered light with time. Furthermore, the microprocessor, based on the MIE theory algorithm, obtains the equivalent particle size of the particles and the number of particles of different particle sizes per unit volume.

The block diagram of each functional part of the sensor is shown in Figure 1.

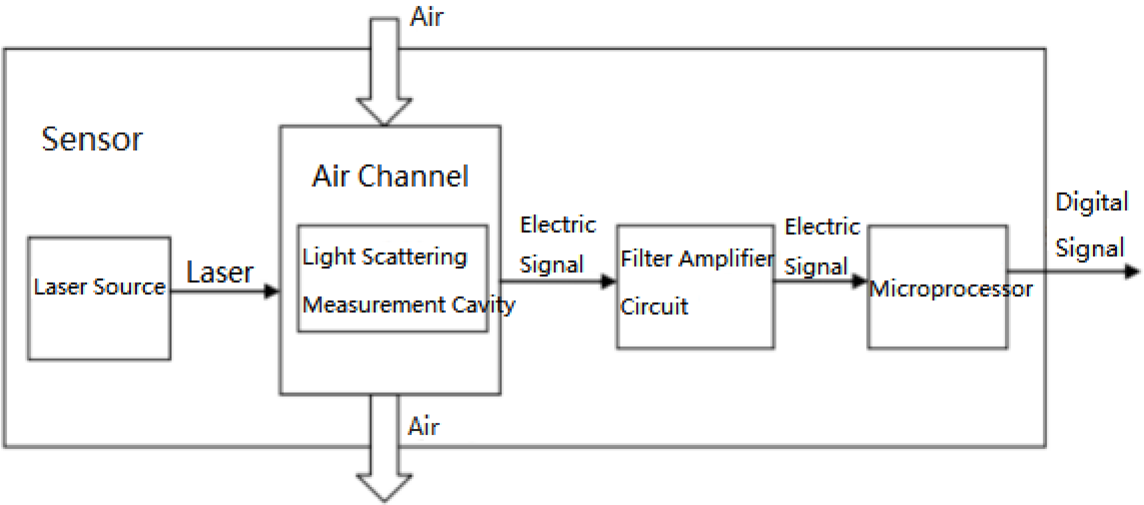


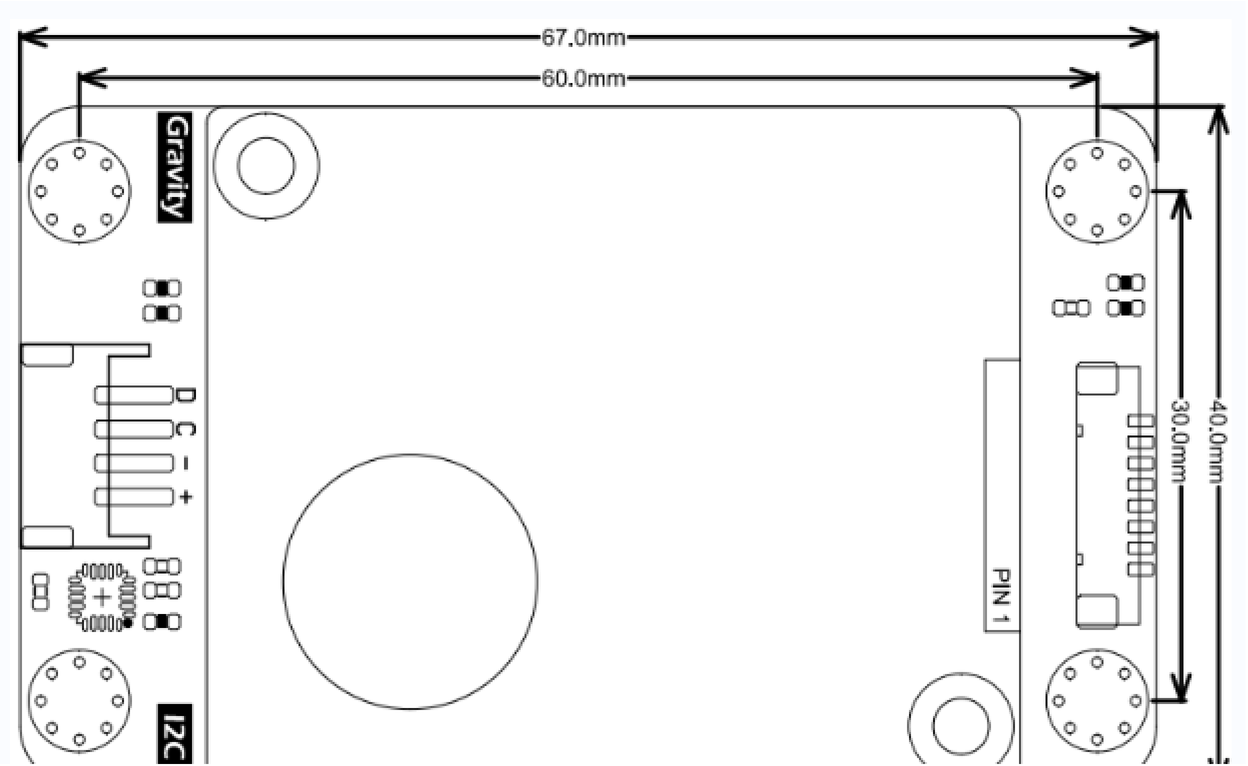
Figure 1 Sensor Functional Block Diagram

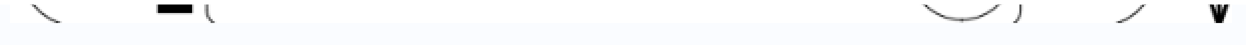
Feature

- Laser scattering principle, accurate measurement
- Zero false alarm rate
- Real-time response and support continuous acquisition

- Minimum recognizable particle size 0.3μm
- Six-sided all-round shielding, stronger anti-interference performance
- Ultra-thin design, only 12 mm, suitable for portable devices

Specification


67.0mm
60.0mm
40.0mm
30.0mm
Gravity
I2C
D
C
−
+
PIN 1



- Working Voltage: 5.0V
- Maximum Working Current: 100 mA
- Standby Current: ≤2 mA
- Data Interface Level: L <0.8V, H> 2.7V, the maximum voltage level is 3.3V
- Particle Measurement Range: 0.3 ~ 1.0 μm; 1.0 ~ 2.5μm; 2.5 ~ 10μm
- Particle Counting Efficiency: 50% @ 0.3μm 98% @ ≥0.5μm
- Effective Range of Particle Mass Concentration (PM2.5 standard value): 0~500ug/m^3
- Maximum Range of Particle Mass Concentration (PM2.5 standard value): ≥1000ug/m^3
- Resolution of Particle Mass Concentration: 1ug/m^3
- Consistency of Particle Mass Concentration (PM2.5 standard value): 1. ± 10% @100 ~ 500 ug/m^3; 2. ±10 ug/m^3@0 ~ 100 ug/m^3
- Standard Volume: 0.1 L
- Single Response Time: <1s
- Comprehensive Response Time: ≤10s
- Working Temperature Range: -10~+60℃
- Working Humidity Range: 0~95%% (non-condensing)

- Mean Time Between Failures: ≥5 years
- Product Size: 67 mm * 40 mm * 14 mm / 2.64* 1.57 * 0.55 inches

Precautions

Installation Note

1. The metal shell is connected to the internal power ground. Be careful not to short-circuit it with other external board circuit or chassis shell.
2. The best installation method is to connect the plane with air inlet and outlet holes with the user’s machine’s inner wall (where there are gas holes connect with the outside). If this method is impossible to use, make sure there is no obstacles within 2cm of the air outlets, and there should be a structure between the air inlet and the air outlet to isolate the airflow and avoid the airflow from the air outlet to the air inlet directly inside the user machine.
3. The vent opening on the inner wall of the user machine for the air inlet should not be smaller than the size of the air inlet.
4. When applied to purifier products, try to avoid placing the sensor directly in the air duct of the purifier. If it is unavoidable, set up a separate structural space and place the sensor in it to isolate it from the air duct of the purifier.

to isolate it from the air duct of the purifier.

5. When used with purifiers or fixed detection equipment, the sensor position should be more than 20cm above the ground. Otherwise, it may be contaminated by large dust particles or even flocs near the ground, causing the fan to wind and block rotation.

6. When the sensor is applied to outdoor fixed equipment, the protection against sandstorms, rain and snow, and willow catkins, etc. should be completed by the equipment.
7. The sensor should be an integral component, the user should not disassemble it, including the metal shielding shell, to prevent irreversible damage.
8. The sensor can be fixed with M3 tapping screws.

Others

1. This sensor data ensures the consistency between the factory individuals, and does not use third-party testing equipment or data as a comparison standard. If the user wants the final measurement result to be consistent with a third-party testing equipment, the user can perform data fitting based on the actual collected results.
2. This sensor is suitable for ordinary indoor environment. If the user equipment is used in the following actual environments, corresponding protective measures should be added to the sensor, otherwise the data consistency may decrease due to excessive dust, oil, and water ingress:

- Places where more than 50% of the time, the dust concentration is greater than 300mg/m^3 or more than 20% of the time, the dust concentration is greater than 500mg/m^3, for example a smoking room.
- Oily fume environment, such as kitchen
- High vapour environment, such as hot spring, bathroom
- Outdoor

PM2.5 air quality international standard table

Country/Organization	Annual average concentration （μg/m³)	24h average concentration （μg/m³)
WHO guideline value	10	25
WHO transitional goal 1	35	75
WHO transitional goal 2	25	50
WHO transitional goal 3	15	37.8

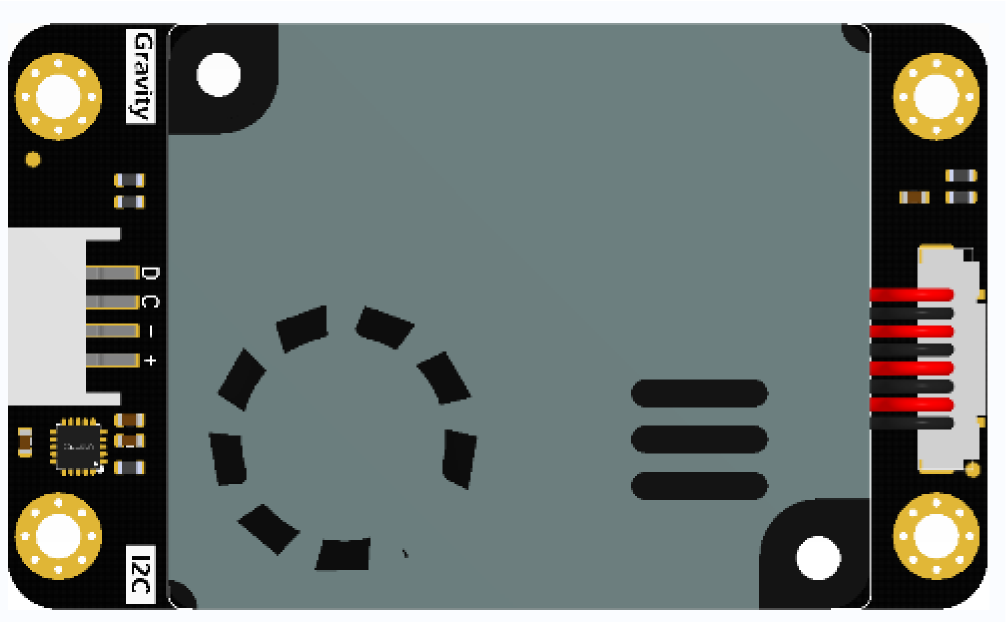
Country/Organization	Annual average concentration (μg/m³)	24h average concentration (μg/m³)
Australia	8	25
United States	15	35
Japan	15	35
European Union	25	None
China	35	75

Correspondence table of air pollution index (AQI) and PM1.0, PM2.5 pollutant concentration value

AQI index	Air quality level	Average PM1.0 concentration in 24 hours (μg/m³)	Average PM2.5 concentration in 24 hours (μg/m³)
0~50	Excellent	0~50	0~35

AQI index	Air quality level	Average PM1.0 concentration in 24 hours（μg/m³)	Average PM2.5 concentration in 24 hours（μg/m³)
51~100	Good	50~150	36~75
101~150	Light pollution	151~250	76~115
151~200	Moderate pollution	251~350	116~150
201~300	Heavy pollution	351~420	151~250
>300	Serious pollution	421~500	251~250

Overview



Serial number	Screen printing	Function description
1	VCC/+	Power positive

Serial number	Screen printing	Function description
1	VCC/+	Power positive
2	GND/-	Power negative
3	SCL/C	I2C clock line
4	SDA/D	I2C data line

Tutorial for Arduino

The product uses the Gravity standard I2C interface, which is relatively simple to use. Connect the sensor to the uno (or other motherboard) as shown in the wiring diagram.

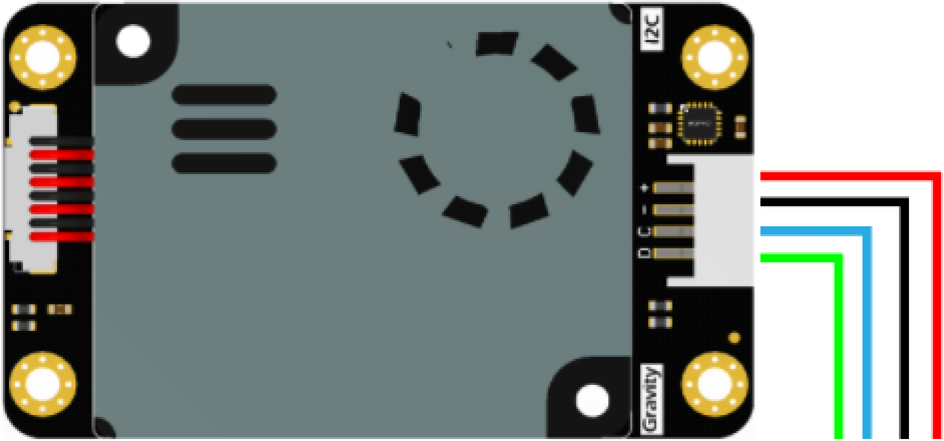
Preparation：

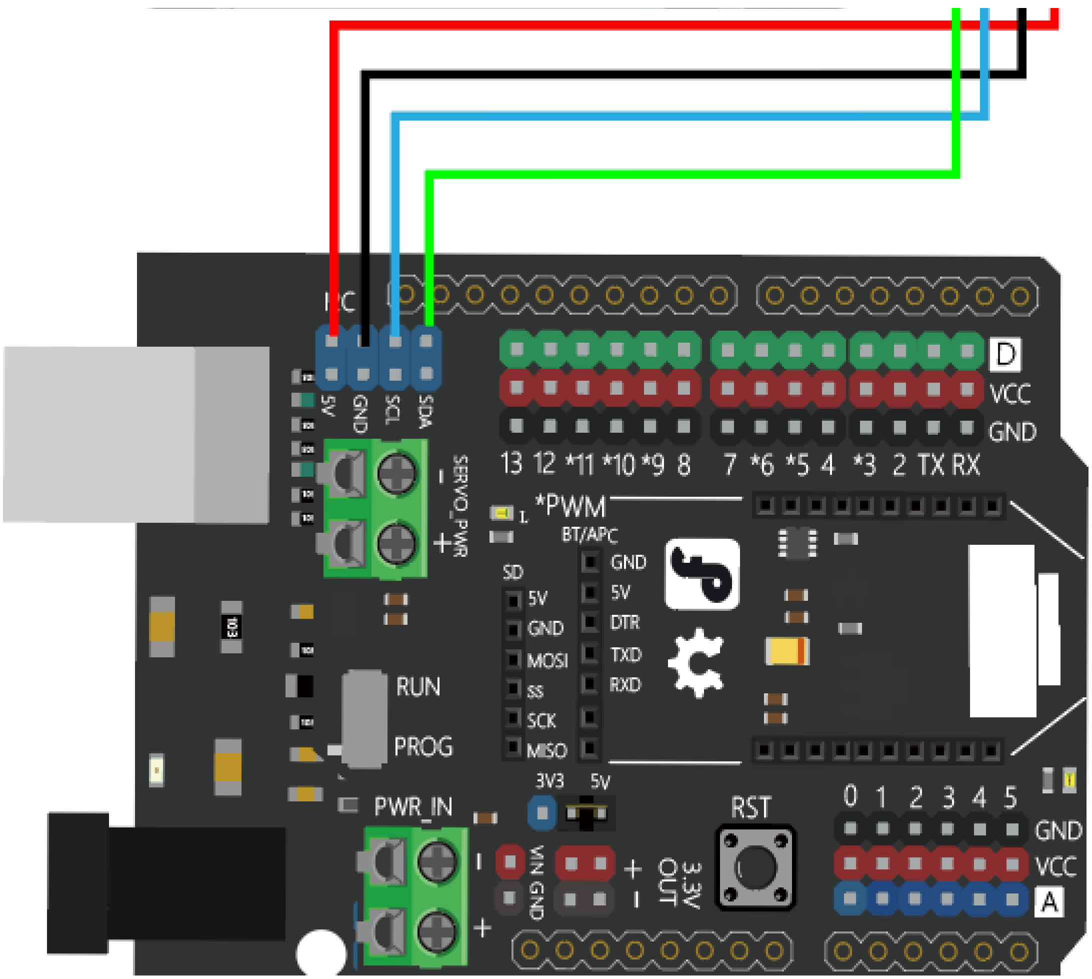
- **Hardware**
  - 1 x Arduino UNO control board
  - 1 x PM2.5 Air Quality Sensor
  - Several Dupont lines

- **Software**
  - Arduino IDE, click to download Arduino IDE (https://www.arduino.cc/en/software)

For how to install the library file, click the link (https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0)

- **Library file**: DFRobot_AirQualitySensor (https://github.com/cdjq/DFRobot_AirQualitySensor)
- **Connection diagram**




I²C
5V
GND
SCL
SDA
SERVO_PWR
-
+
D
VCC
GND
13 12 *11 *10 *9 8
7 *6 *5 4 *3 2 TX RX
*PWM
L
BT/APC
SD
5V
GND
MOSI
SS
SCK
MISO
GND
5V
DTR
TXD
RXD
RUN
PROG
PWR_IN
3V3
5V
VIN GND
+
-
3.3V OUT
RST
0 1 2 3 4 5
GND
VCC
A

- **Function list**

```
/*!
 *  @brief Get the concentration of the specified particle type
 *  @param type:PARTICLE_PM1_0_STANDARD  Particle concentration of PM1.0 in standard parti
                      PARTICLE_PM2_5_STANDARD  Particle concentration of PM2.5 in standard
                      PARTICLE_PM10_STANDARD   Particle concentration of PM10 in standard
                      PARTICLE_PM1_0_ATMOSPHERE Particulate matter concentration of PM1.0
                      PARTICLE_PM2_5_ATMOSPHERE Particulate matter concentration of PM2.5
                      PARTICLE_PM10_ATMOSPHERE  Particulate matter concentration of PM10 i
 *  @return concentration（ug/m3）
 */
    uint16_t gainParticleConcentration_ugm3(uint8_t type);
/*!
 *  @brief Get the number of particles per 0.1 liter of air
 *  @param type:PARTICLENUM_0_3_UM_EVERY0_1L_AIR
                      PARTICLENUM_0_5_UM_EVERY0_1L_AIR
                      PARTICLENUM_1_0_UM_EVERY0_1L_AIR
                      PARTICLENUM_2_5_UM_EVERY0_1L_AIR
                      PARTICLENUM_5_0_UM_EVERY0_1L_AIR
                      PARTICLENUM_10_UM_EVERY0_1L_AIR
                      PARTICLENUM_10_UM_EVERY0_1L_AIR
 *  @return number
 */
    uint16_t gainParticleNum_Every0_1L(uint8_t type);


/*!
 *  @brief Writes data to the specified register of the sensor
 *  @param NULL
 *  @return No return value
 */
    uint8_t gainVersion();
```

Sample code-standard particle concentration

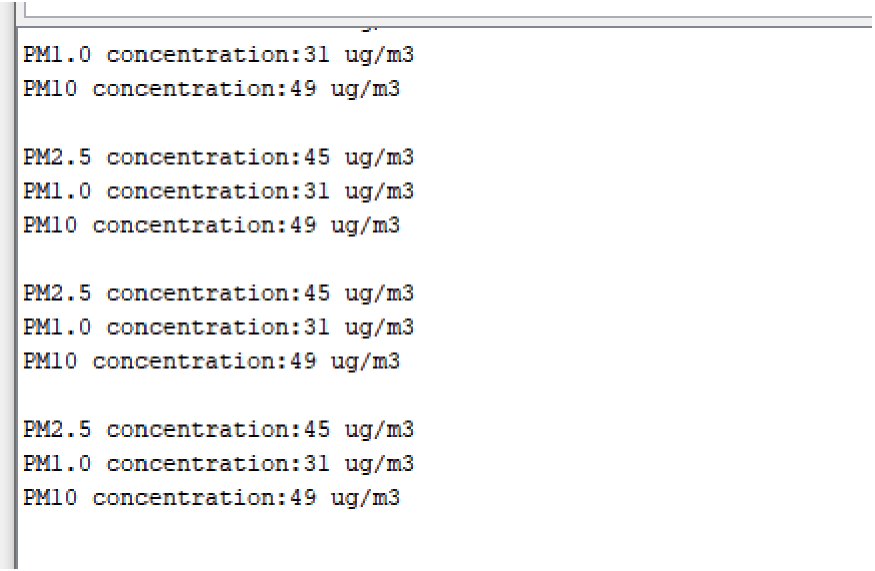
Function description: Get the standard particle concentration of PM2.5, PM1.0, PM10.

```
#include "DFRobot_AirQualitySensor.h"

#define I2C_ADDRESS    0x19
  DFRobot_AirQualitySensor_I2C particle(&Wire ,I2C_ADDRESS);

void setup() {
  Serial.begin(115200);
/**
  Sensor initialization is used to initialize IIC, which is determined by the communicatio
*/
  while(!particle.begin())
  {
    Serial.println("NO Deivces !");
    delay(1000);
  }
  Serial.println("sensor begin success!");
  delay(1000);
/**
  Get sensor version number
  Get sensor version number
*/
  uint8_t version = particle.gainVersion();
  Serial.print("version is : ");
  Serial.println(version);

  delay(1000);
}

void loop() {
/**
 *@brief : Get concentration of PM1.0 PM2.5 PM10
 *@param :PARTICLE_PM1_0_STANDARD  Standard particle
          PARTICLE_PM2_5_STANDARD  Standard particle
          PARTICLE_PM10_STANDARD   Standard particle
          PARTICLE_PM1_0_ATMOSPHERE  In atmospheric environment
          PARTICLE_PM2_5_ATMOSPHERE  In atmospheric environment
          PARTICLE_PM10_ATMOSPHERE   In atmospheric environment
*/
  uint16_t PM2_5 = particle.gainParticleConcentration_ugm3(PARTICLE_PM2_5_STANDARD);
  uint16_t PM1_0 = particle.gainParticleConcentration_ugm3(PARTICLE_PM1_0_STANDARD);
  uint16_t PM10 = particle.gainParticleConcentration_ugm3(PARTICLE_PM10_STANDARD);
  Serial.print("PM2.5 concentration:");
  Serial.print(PM2_5);
  Serial.println(" ug/m3");
  Serial.print("PM1.0 concentration:");
```

```
    Serial.print("PM1.0 concentration:");
    Serial.print(PM1_0);
    Serial.println(" ug/m3");
    Serial.print("PM10 concentration:");
    Serial.print(PM10);

    Serial.println(" ug/m3");
    Serial.println();
    delay(1000);
  }
```

Result

In the state of standard particulate matter, the concentration of PM2.5, PM1.0, PM10 read is: 45ug/m3, 31ug/m3, 49ug/m3.



```
PM1.0 concentration:31 ug/m3
PM10 concentration:49 ug/m3

PM2.5 concentration:45 ug/m3
PM1.0 concentration:31 ug/m3
PM10 concentration:49 ug/m3

PM2.5 concentration:45 ug/m3
PM1.0 concentration:31 ug/m3
PM10 concentration:49 ug/m3

PM2.5 concentration:45 ug/m3
PM1.0 concentration:31 ug/m3
PM10 concentration:49 ug/m3
```

Sample code-particulate matter concentration in atmospheric environment

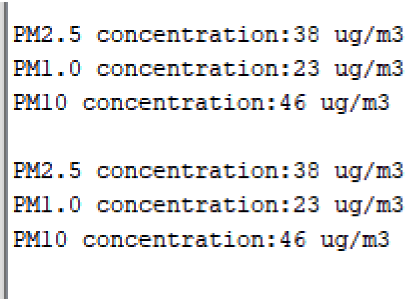
Function description: Obtain the concentration of particulate matter in the atmospheric environment of PM2.5, PM1.0, PM10.

```
#include "DFRobot_AirQualitySensor.h"

#define I2C_ADDRESS    0x19
  DFRobot_AirQualitySensor_I2C particle(&Wire ,I2C_ADDRESS);

void setup() {
  Serial.begin(115200);
/**
  Sensor initialization is used to initialize IIC, which is determined by the communicatio
*/
  while(!particle.begin())
  {
    Serial.println("NO Deivces !");
    delay(1000);
  }
  Serial.println("sensor begin success!");
  delay(1000);
/**
  Get sensor version number
  Get sensor version number
*/
  uint8_t version = particle.gainVersion();
  Serial.print("version is : ");
  Serial.println(version);

  delay(1000);
}

void loop() {
/**
 *@brief : Get concentration of PM1.0
 *@param :PARTICLE_PM1_0_STANDARD  Standard particle
          PARTICLE_PM2_5_STANDARD  Standard particle
          PARTICLE_PM10_STANDARD   Standard particle
          PARTICLE_PM1_0_ATMOSPHERE  In atmospheric environment
          PARTICLE_PM2_5_ATMOSPHERE  In atmospheric environment
          PARTICLE_PM10_ATMOSPHERE   In atmospheric environment
*/
  uint16_t PM2_5 = particle.gainParticleConcentration_ugm3(PARTICLE_PM2_5_ATMOSPHERE );
  uint16_t PM1_0 = particle.gainParticleConcentration_ugm3(PARTICLE_PM1_0_ATMOSPHERE );
  uint16_t PM10 = particle.gainParticleConcentration_ugm3(PARTICLE_PM10_ATMOSPHERE);
  Serial.print("PM2.5 concentration:");
  Serial.print(PM2_5);
  Serial.println(" ug/m3");
  Serial.print("PM1.0 concentration:");
```

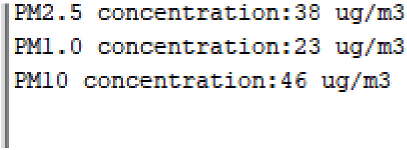
```
    Serial.print("PM1.0 concentration:");
    Serial.print(PM1_0);
    Serial.println(" ug/m3");
    Serial.print("PM10 concentration:");
    Serial.print(PM10);

    Serial.println(" ug/m3");
    Serial.println();
    delay(1000);
  }
```

Result

In the atmospheric environment, the particle concentration of PM2.5/PM1.0/PM10 is about: 38ug/m3, 23ug/m3, 46ug/m3.



```
PM2.5 concentration:38 ug/m3
PM1.0 concentration:23 ug/m3
PM10 concentration:46 ug/m3

PM2.5 concentration:38 ug/m3
PM1.0 concentration:23 ug/m3
PM10 concentration:46 ug/m3
```



```
PM2.5 concentration:38 ug/m3
PM1.0 concentration:23 ug/m3
PM10 concentration:46 ug/m3
```

Sample code-the number of particles per 0.1 liter of air

Function description: Read the number of particles above 0.3um/0.5um/1.0um/2.5um/5.0um/10um per 0.1 liter of air.

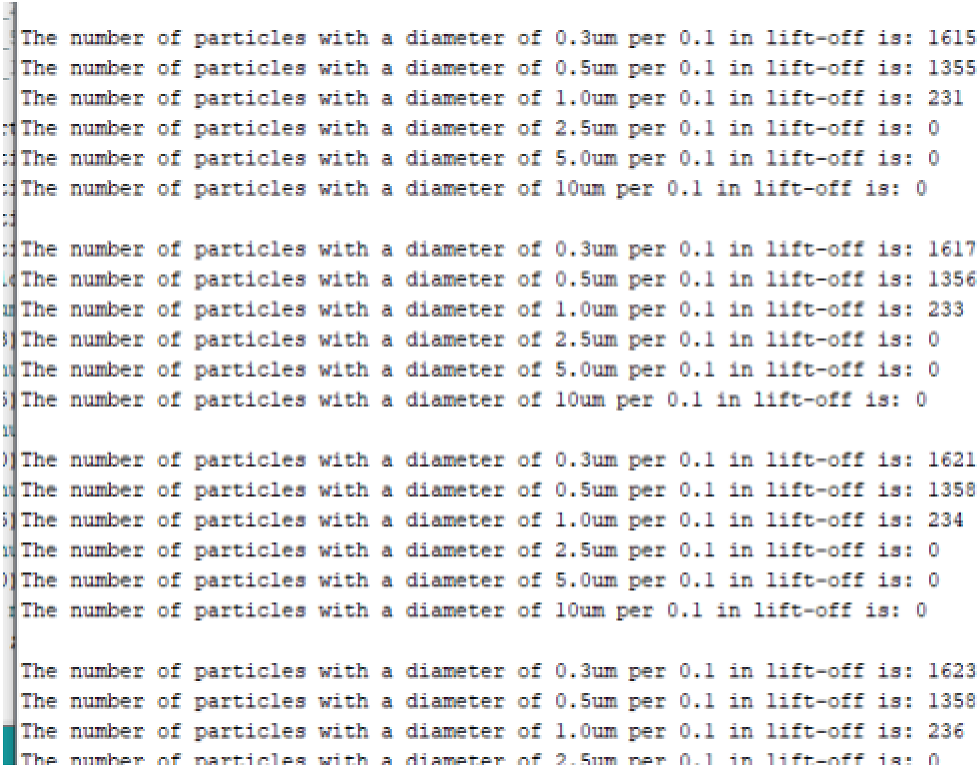
```
#include "DFRobot_AirQualitySensor.h"

#define I2C_ADDRESS    0x19
  DFRobot_AirQualitySensor_I2C particle(&Wire ,I2C_ADDRESS);

void setup() {
  Serial.begin(115200);
/**
  Sensor initialization is used to initialize IIC, which is determined by the communicatio
*/
  while(!particle.begin())
  {
    Serial.println("NO Deivces !");
    delay(1000);
  }
  Serial.println("sensor begin success!");
  delay(1000);
/**
```

```
/**
  Get sensor version number
*/
  uint8_t version = particle.gainVersion();
  Serial.print("version is : ");

  Serial.println(version);
  delay(1000);
}

void loop() {
/**
 *@brief : Get particle number of 0.3um/0.5um/1.0um/2.5um/5.0um/10um per 0.1L of air
 *@param :PARTICLENUM_0_3_UM_EVERY0_1L_AIR
          PARTICLENUM_0_5_UM_EVERY0_1L_AIR
          PARTICLENUM_1_0_UM_EVERY0_1L_AIR
          PARTICLENUM_2_5_UM_EVERY0_1L_AIR
          PARTICLENUM_5_0_UM_EVERY0_1L_AIR
          PARTICLENUM_10_UM_EVERY0_1L_AIR
*/
  uint16_t um0_3 = particle.gainParticleNum_Every0_1L(PARTICLENUM_0_3_UM_EVERY0_1L_AIR);
  uint16_t um0_5= particle.gainParticleNum_Every0_1L(PARTICLENUM_0_5_UM_EVERY0_1L_AIR);
  uint16_t um1_0= particle.gainParticleNum_Every0_1L(PARTICLENUM_1_0_UM_EVERY0_1L_AIR);
  uint16_t um2_5= particle.gainParticleNum_Every0_1L(PARTICLENUM_2_5_UM_EVERY0_1L_AIR);
  uint16_t um5_0= particle.gainParticleNum_Every0_1L(PARTICLENUM_5_0_UM_EVERY0_1L_AIR);
  uint16_t um10= particle.gainParticleNum_Every0_1L(PARTICLENUM_10_UM_EVERY0_1L_AIR);
```

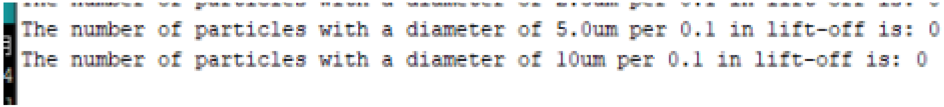
```
    uint16_t um10= particle.gainParticleNum_Every0_1L(PARTICLENUM_10_UM_EVERY0_1L_AIR);
    Serial.print("The number of particles with a diameter of 0.3um per 0.1 in lift-off is: "
    Serial.println(um0_3);
     Serial.print("The number of particles with a diameter of 0.5um per 0.1 in lift-off is:
    Serial.println(um0_5);

     Serial.print("The number of particles with a diameter of 1.0um per 0.1 in lift-off is:
    Serial.println(um1_0);
     Serial.print("The number of particles with a diameter of 2.5um per 0.1 in lift-off is:
    Serial.println(um2_5);
     Serial.print("The number of particles with a diameter of 5.0um per 0.1 in lift-off is:
    Serial.println(um5_0);
      Serial.print("The number of particles with a diameter of 10um per 0.1 in lift-off is:
    Serial.println(um10);
    Serial.println("");
    delay(1000);
  }
```

Result

It is read that the number of 0.3um/0.5um/1.0um/2.5um/5.0um/10um particles per 0.1 liter of air is approximately: 1615, 1356, 233, 0, 0, 0.



```
The number of particles with a diameter of 0.3um per 0.1 in lift-off is: 1615
The number of particles with a diameter of 0.5um per 0.1 in lift-off is: 1355
The number of particles with a diameter of 1.0um per 0.1 in lift-off is: 231
The number of particles with a diameter of 2.5um per 0.1 in lift-off is: 0
The number of particles with a diameter of 5.0um per 0.1 in lift-off is: 0
The number of particles with a diameter of 10um per 0.1 in lift-off is: 0

The number of particles with a diameter of 0.3um per 0.1 in lift-off is: 1617
The number of particles with a diameter of 0.5um per 0.1 in lift-off is: 1356
The number of particles with a diameter of 1.0um per 0.1 in lift-off is: 233
The number of particles with a diameter of 2.5um per 0.1 in lift-off is: 0
The number of particles with a diameter of 5.0um per 0.1 in lift-off is: 0
The number of particles with a diameter of 10um per 0.1 in lift-off is: 0

The number of particles with a diameter of 0.3um per 0.1 in lift-off is: 1621
The number of particles with a diameter of 0.5um per 0.1 in lift-off is: 1358
The number of particles with a diameter of 1.0um per 0.1 in lift-off is: 234
The number of particles with a diameter of 2.5um per 0.1 in lift-off is: 0
The number of particles with a diameter of 5.0um per 0.1 in lift-off is: 0
The number of particles with a diameter of 10um per 0.1 in lift-off is: 0

The number of particles with a diameter of 0.3um per 0.1 in lift-off is: 1623
The number of particles with a diameter of 0.5um per 0.1 in lift-off is: 1358
The number of particles with a diameter of 1.0um per 0.1 in lift-off is: 236
The number of particles with a diameter of 2.5um per 0.1 in lift-off is: 0
```



```
The number of particles with a diameter of 5.0um per 0.1 in lift-off is: 0
The number of particles with a diameter of 10um per 0.1 in lift-off is: 0
```

Tutorial for micro:bit - MakeCode

The product uses the Gravity standard I2C interface, which is relatively simple to use. Connect the sensor to the micro:bit as shown in the wiring diagram.

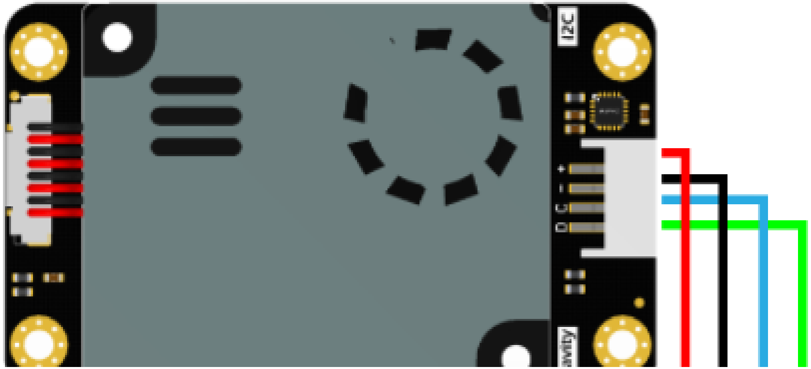
Preparation：

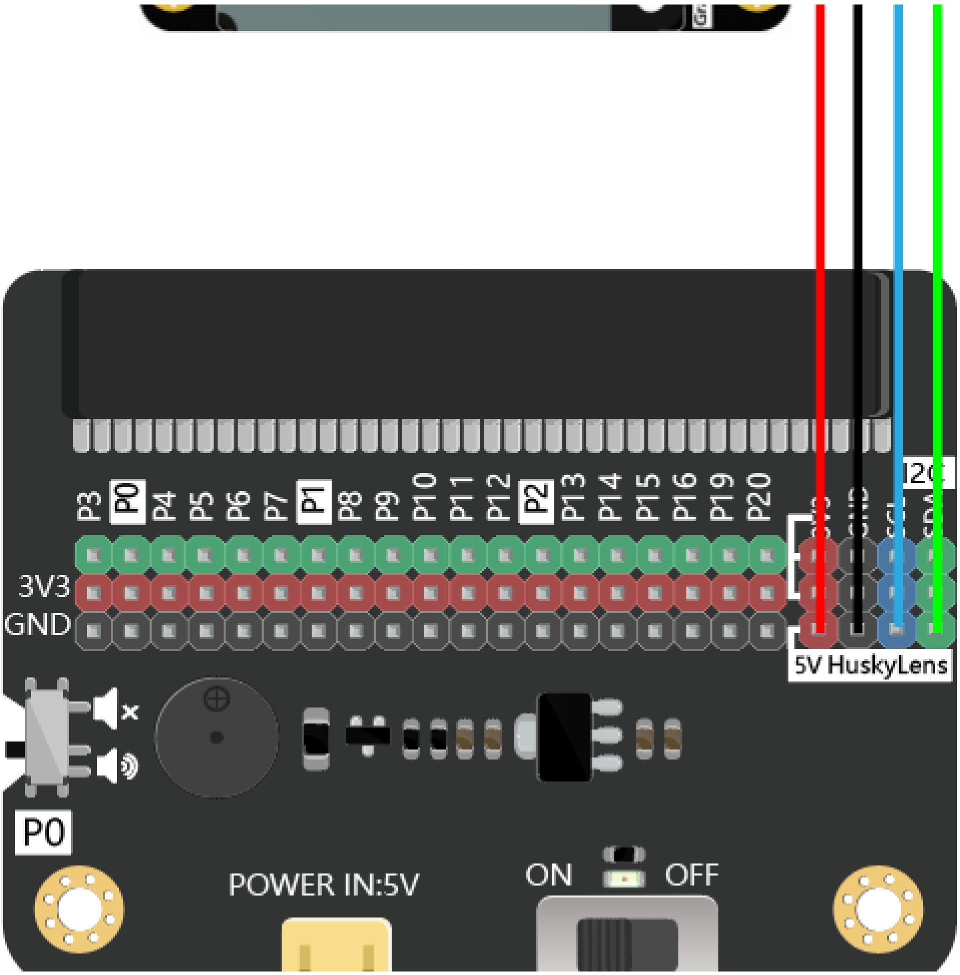
- **Hardware**
  - 1 x micro:bit control board
  - 1 x PM2.5 Air Quality Sensor
  - 1 × MBT0008 micro:IO Extend micro:bit control board IO expansion board (https://www.dfrobot.com/product-1867.html)
  - Several Dupont lines
- **Software**

makecode (https://makecode.microbit.org/#)

How to use makecode software? MakeCode online graphical programming basic operation tutorial (https://wiki.dfrobot.com/Makecode%20Get-started%20Tutorial) MakeCode basic operation tutorial (https://wiki.dfrobot.com/Makecode%20Get-started%20Tutorial)

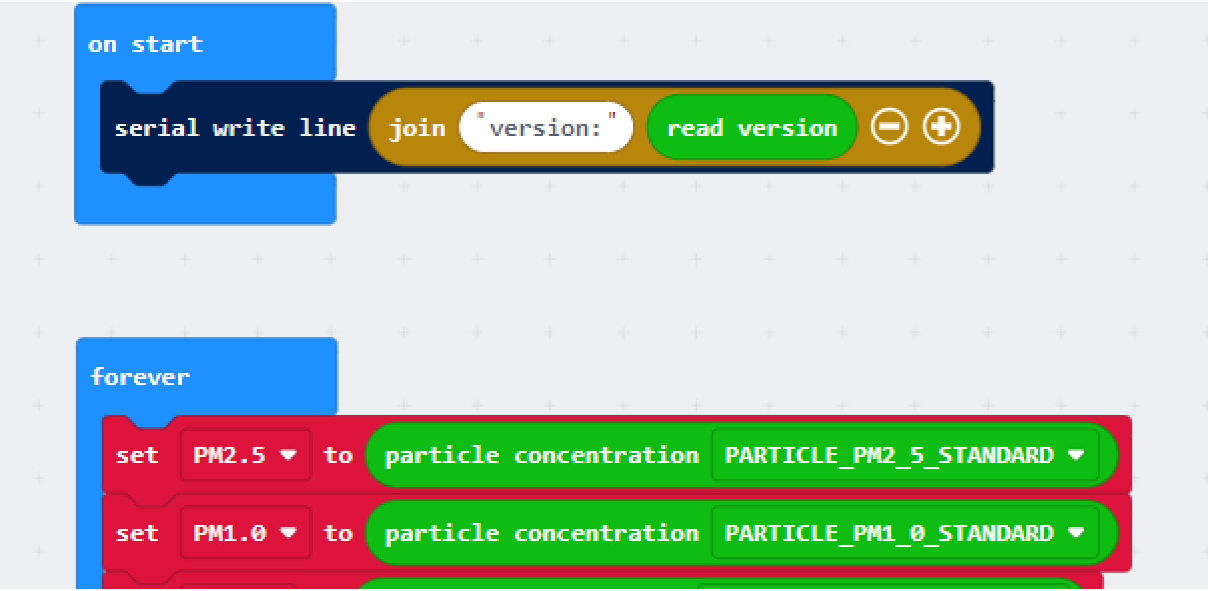
- **Library file:** https://github.com/tangjie133/pxt-DFRobot_PM25_Air_Quality_Sensor (https://github.com/tangjie133/pxt-DFRobot_PM25_Air_Quality_Sensor) （Temporary)
- **Connection diagram**



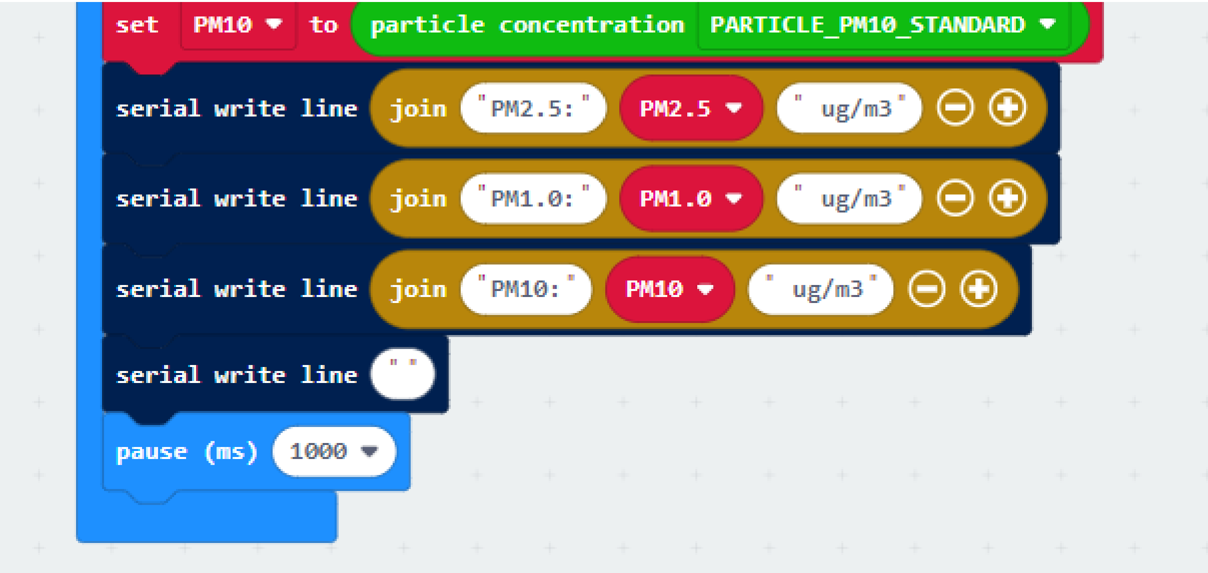

P3
P0
P4
P5
P6
P7
P1
P8
P9
P10
P11
P12
P2
P13
P14
P15
P16
P19
P20
I2C
GND
SCL
3V3
GND
5V HuskyLens
P0
POWER IN:5V
ON
OFF

Sample code-standard particle concentration

Function description: Get the standard particle concentration of PM2.5, PM1.0, PM10. Program link: https://makecode.microbit.org/_9HRJ7VTsDaM2 (https://makecode.microbit.org/_9HRJ7VTsDaM2)



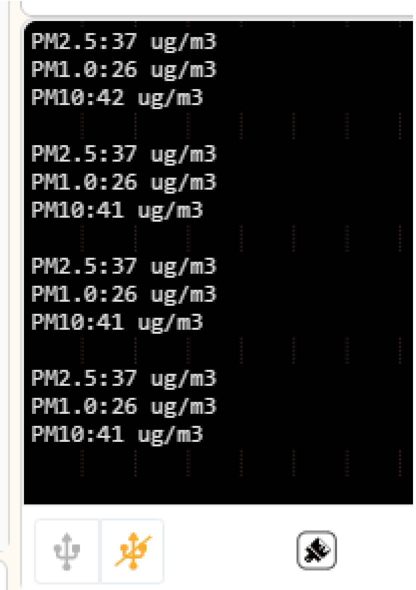
```
on start
    serial write line  join  "version: "  read version  ⊖ ⊕

forever
    set  PM2.5 ▾  to  particle concentration  PARTICLE_PM2_5_STANDARD ▾
    set  PM1.0 ▾  to  particle concentration  PARTICLE_PM1_0_STANDARD ▾
```



```
set PM10 to particle concentration PARTICLE_PM10_STANDARD
serial write line join "PM2.5:" PM2.5 " ug/m3"
serial write line join "PM1.0:" PM1.0 " ug/m3"
serial write line join "PM10:" PM10 " ug/m3"
serial write line " "
pause (ms) 1000
```

Result

In the state of standard particulate matter, the concentration of PM2.5, PM1.0, PM10 read is: 37ug/m3, 26ug/m3, 41ug/m3. The data printed by the serial port is as follows:

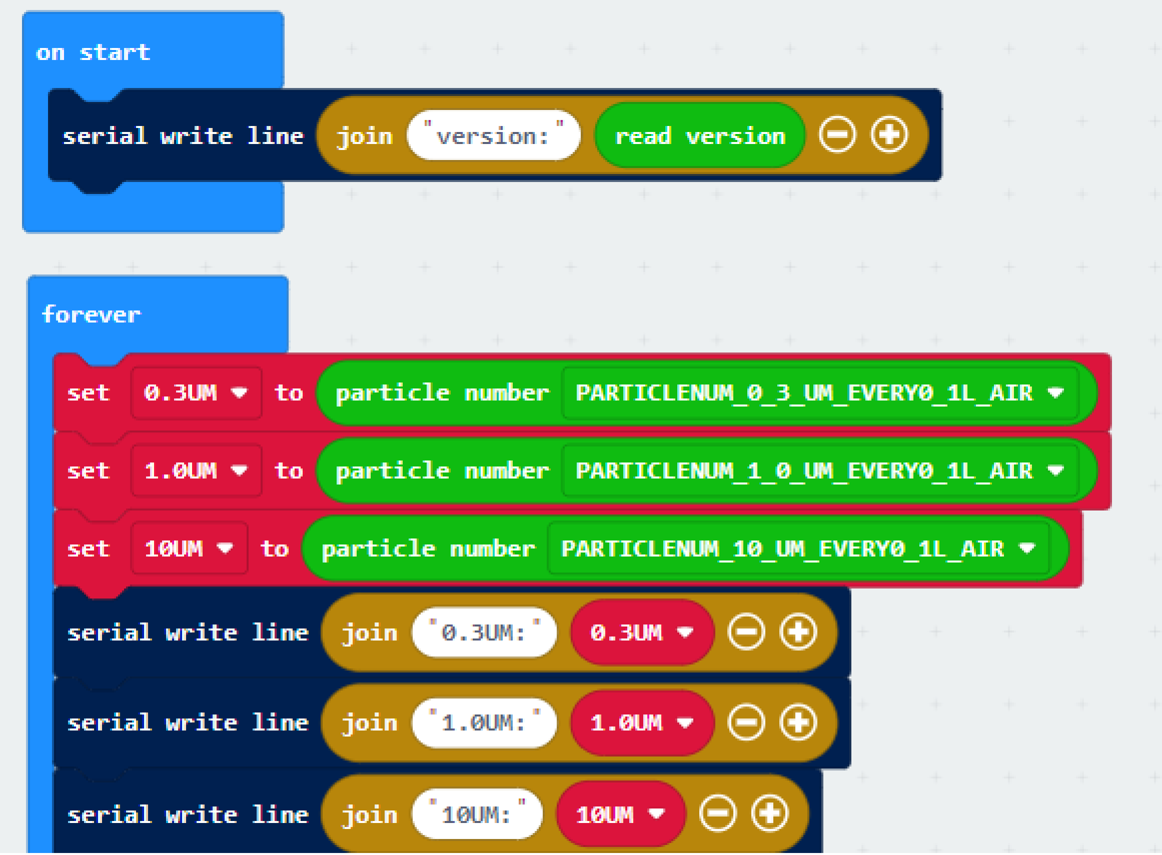


```
PM2.5:37 ug/m3
PM1.0:26 ug/m3
PM10:42 ug/m3

PM2.5:37 ug/m3
PM1.0:26 ug/m3
PM10:41 ug/m3

PM2.5:37 ug/m3
PM1.0:26 ug/m3
PM10:41 ug/m3

PM2.5:37 ug/m3
PM1.0:26 ug/m3
PM10:41 ug/m3
```

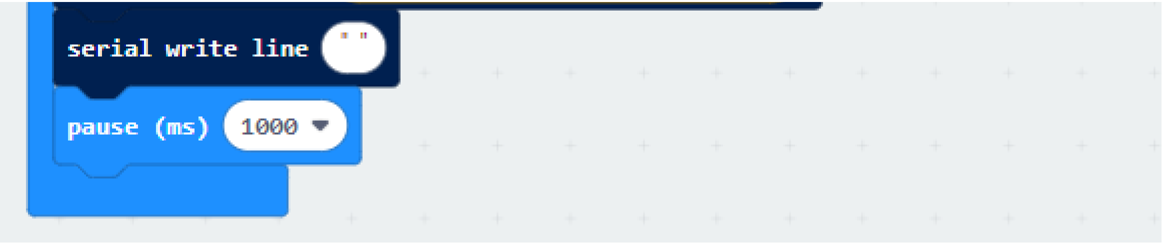
Sample code-the number of particles per 0.1 liter of air

Function description: Read the number of particles above 0.3um/1.0um/10um per 0.1 liter of air.

Program link: https://makecode.microbit.org/_JMxHs15fK29r (https://makecode.microbit.org/_JMxHs15fK29r)



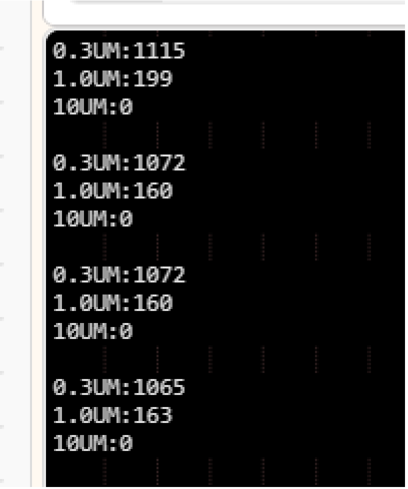
```
on start
    serial write line  join  "version:"  read version  ⊖ ⊕

forever
    set 0.3UM ▾ to particle number PARTICLENUM_0_3_UM_EVERY0_1L_AIR ▾
    set 1.0UM ▾ to particle number PARTICLENUM_1_0_UM_EVERY0_1L_AIR ▾
    set 10UM ▾ to particle number PARTICLENUM_10_UM_EVERY0_1L_AIR ▾
    serial write line  join  "0.3UM:"  0.3UM ▾  ⊖ ⊕
    serial write line  join  "1.0UM:"  1.0UM ▾  ⊖ ⊕
    serial write line  join  "10UM:"  10UM ▾  ⊖ ⊕
```



```
serial write line " "
pause (ms) 1000
```

Result

It is read that the number of 0.3um/1.0um/10um particles per 0.1 liter of air is about 1115, 160, and 0 respectively.



```
0.3UM:1115
1.0UM:199
10UM:0

0.3UM:1072
1.0UM:160
10UM:0

0.3UM:1072
1.0UM:160
10UM:0

0.3UM:1065
1.0UM:163
10UM:0
```

[redacted]

FAQ

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum** (https://www.dfrobot.com/forum/).

More Documents

DFshopping_car1.png Get **Gravity PM2.5 Air Quality Sensor** (https://www.dfrobot.com/product-2439.html) from DFRobot Store or **DFRobot Distributor**. (https://www.dfrobot.com/index.php?route=information/distributorslogo)

Turn to the Top