2022년 IoT기반 스마트 솔루션 개발자 양성과정

Embedded Application

3-Dust Sensor

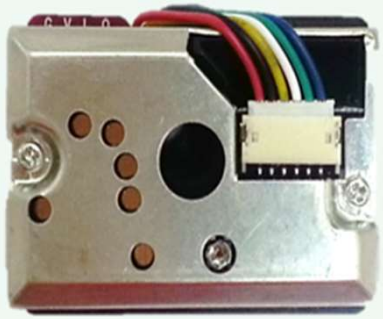
담당 교수 : 윤 종 이

010-9577-1696

ojo1696@naver.com

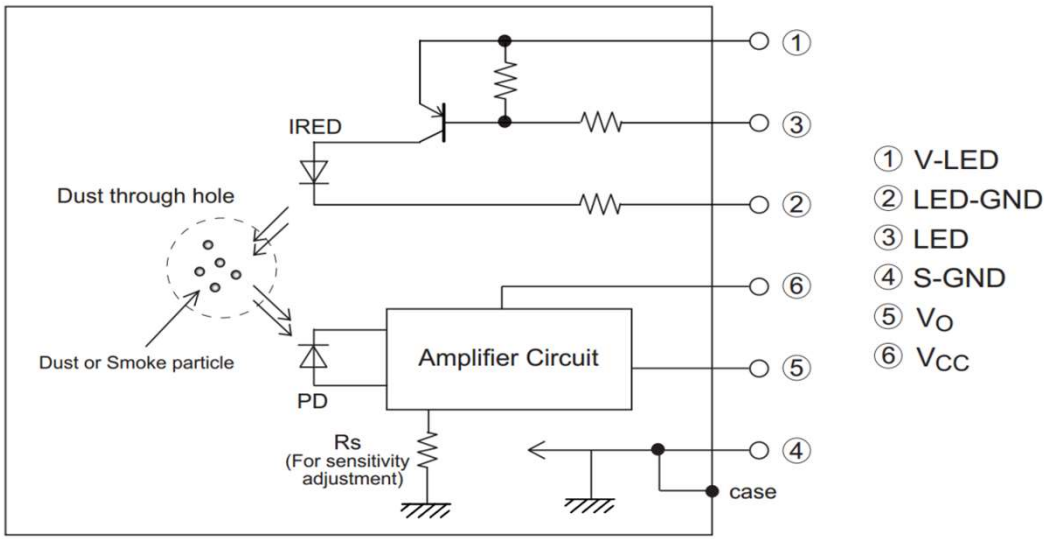
https://cafe.naver.com/yoons2022

GP2Y1010AU0F

센서 모듈 외형	모듈 항목	모듈 항목의 내용
	먼지 센서	GP2Y1010AU0F
	탐지 방법	Photometry
	연기 구별	Possible(with Pulse pattern of output)
	동작 전압	5V
	크기	45x48mm
담배연기와 도시의 미세 먼지 등을 검출하는 센서 모듈		

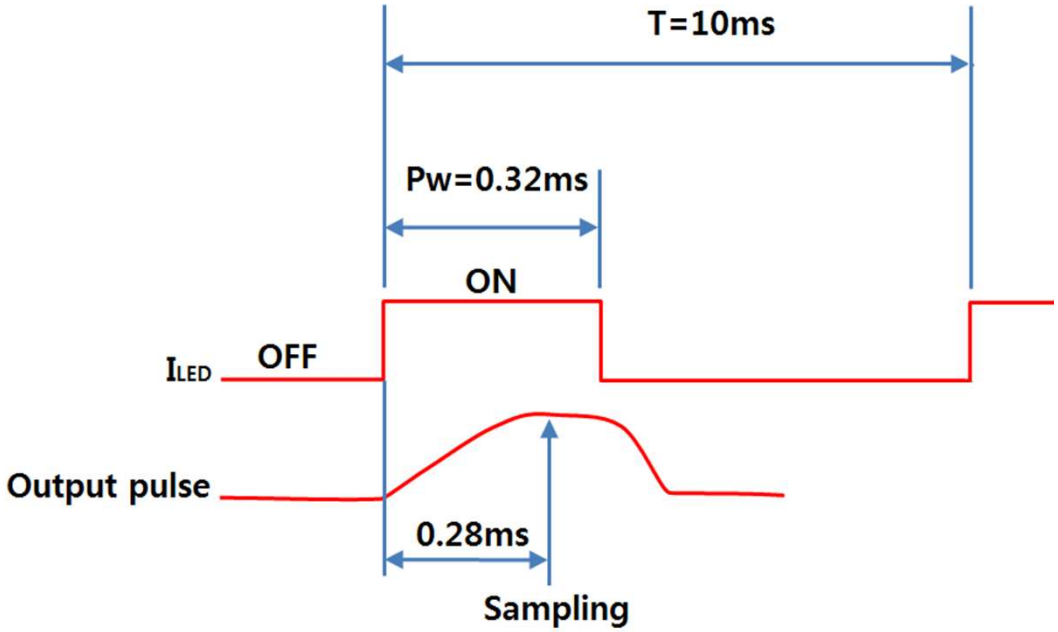
Dust Sensor

- Dust Sensor는 광학 먼지 센서로 공기 중의 먼지와 입자를 이용하여 공기 오염도를 측정해주는 센서
  - 위 부분에 있는 IRED(InfraRed Emitting Diode) 부분에서 적외선을 발광하고 발광된 적외선이 먼지 또는 연기 입자에 부딪치면 반사되어 PD(Photo Diode) 부분에 수광
  - 수광 되는 적외선의 양에 따라 사용자는 공기의 오염도 확인

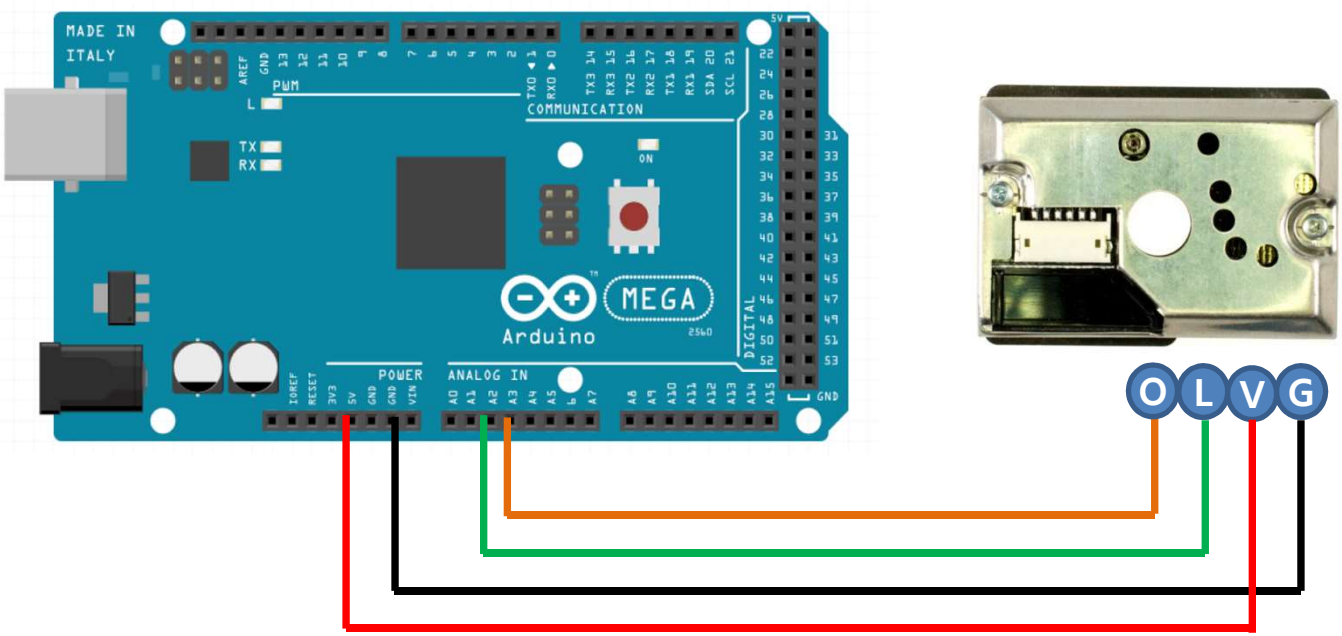


Sensor 특성

- Dust 센서의 측정 주기는 10ms
- LED의 ON 상태를 0.28ms간 유지시킨 후 Output pulse의 신호를 측정
- 0.04ms 후 LED를 OFF



Wiring

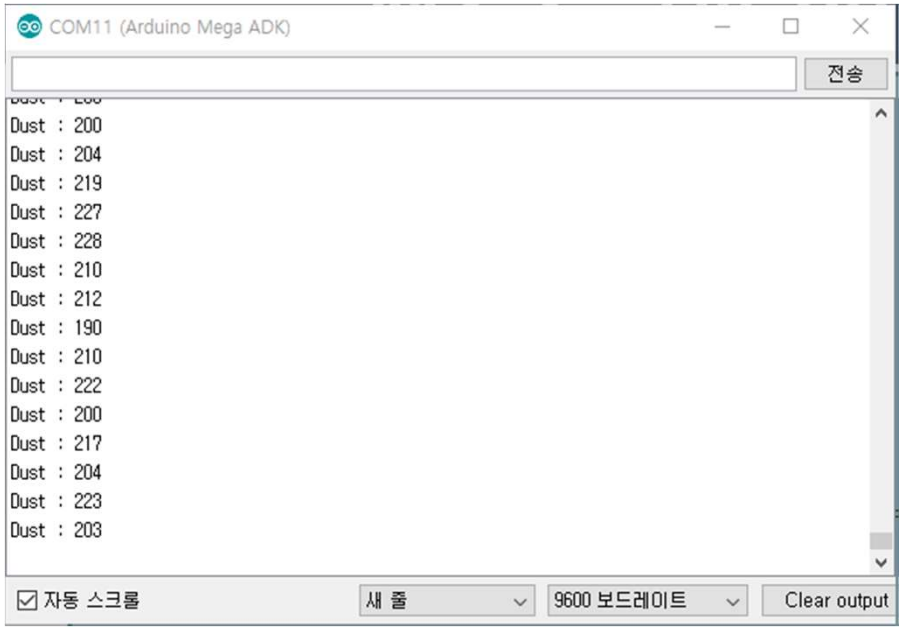

MADE IN
ITALY
PWM
COMMUNICATION
POWER
ANALOG IN
DIGITAL
MEGA
Arduino
O
L
V
G

M3-1 : Dust sensor

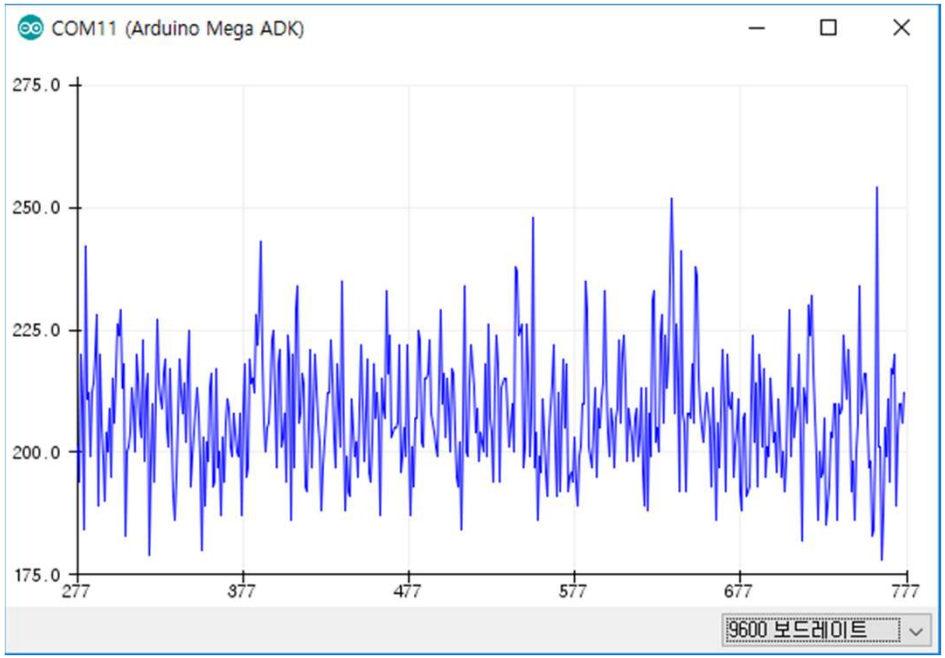
```
#define DUST_LED  A2
#define DUST_OUT  A3

void setup() {
  Serial.begin(9600);
  pinMode(DUST_LED, OUTPUT);
  pinMode(DUST_OUT, INPUT);
  digitalWrite(DUST_LED, HIGH);
}

void loop() {
  Serial.print("Dust : ");
  Serial.println(SensorRead());
  delay(200);
}
```

```
unsigned int SensorRead(void){
    unsigned int Sensor_data;

    digitalWrite(DUST_LED, LOW);
    delayMicroseconds(280);
    Sensor_data = analogRead(DUST_OUT);
    delayMicroseconds(40);
    digitalWrite(DUST_LED, HIGH);
    delayMicroseconds(9680);
    return Sensor_data;
}
```

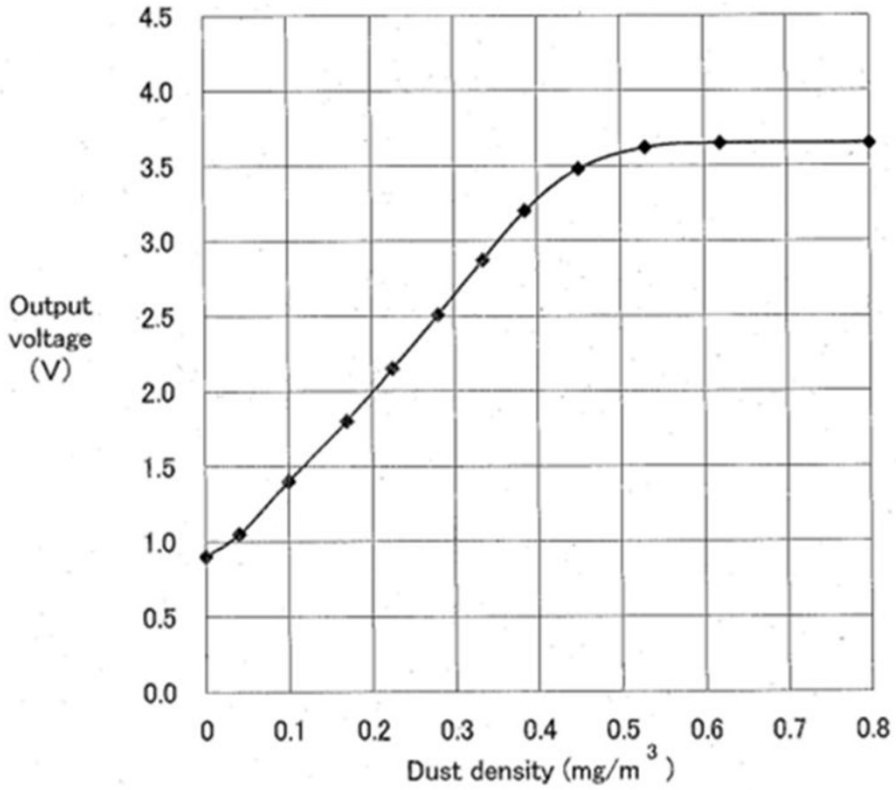
M3-1 : Serial Monitor



M3-1 : Serial Plotter

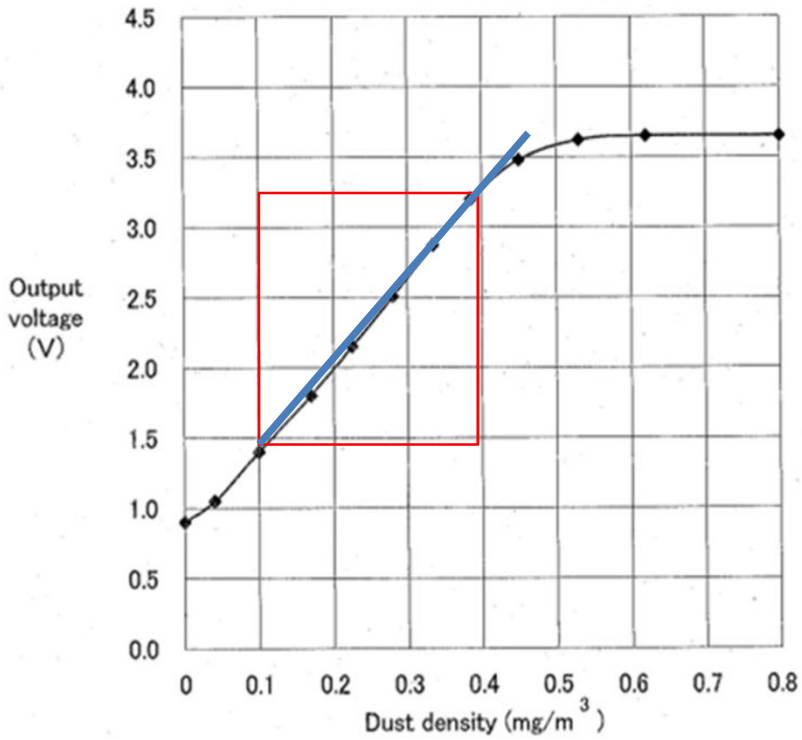


Dust density characteristics



Calibration

- 먼지 농도의 단위 : ug/m3
- 기울기 : y의 변화량/x의 변화량
  = (3.25-1.5) / (0.4-0.1) = 5.8
- mg/m3 = Vin/5.8 – 0.1
  = **ADC * (5/1023)/5.8 - 0.1**




```
float DustDensity_ugPm3(int Vin) {
  float Density=(float)Vin/5.8-0.1;
  return Density;
}
```

M3-2 : Calibration

```
float dustDensity=0;

void setup( ) {
  Serial.begin(9600);

  for (int k=0; k<500;k++){
    dustDensity=DustDensity_ugPm3(k);
    Serial.println(dustDensity);
  }
}

void loop( ) {
}

float DustDensity_ugPm3(int Vin) {
  float Density=(float)Vin/5.8-0.1;
  return Density;
}
```

M3-2 : Plotter

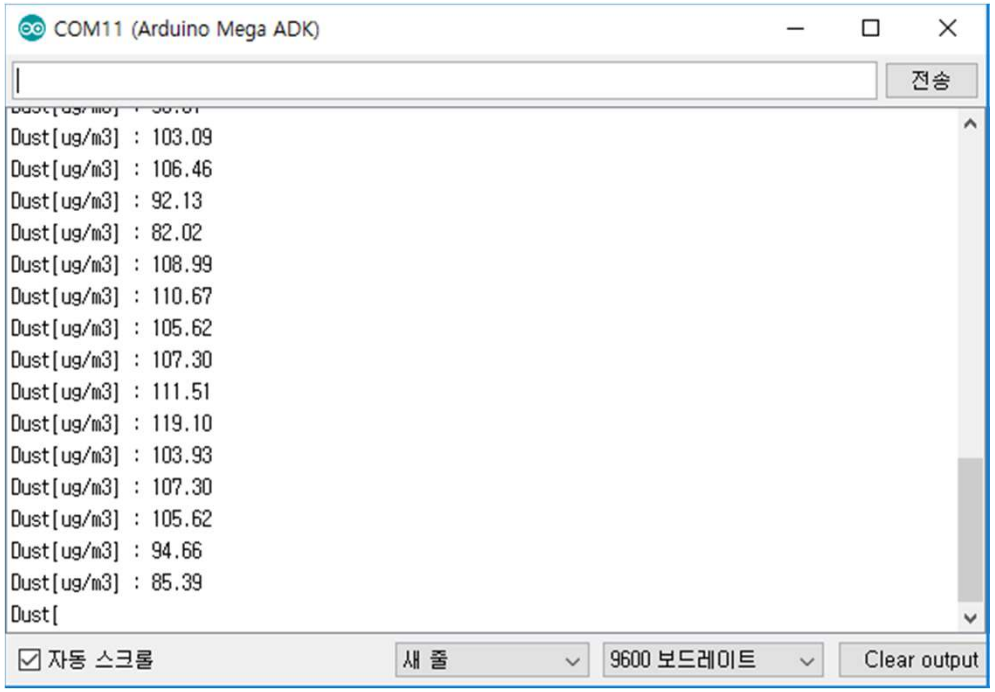

COM11 (Arduino Mega ADK)
90.0
60.0
30.0
0.0
-30.0
0
100
200
300
400
500
9600 보드레이트

M3-3 : Dust density(ug/m3)

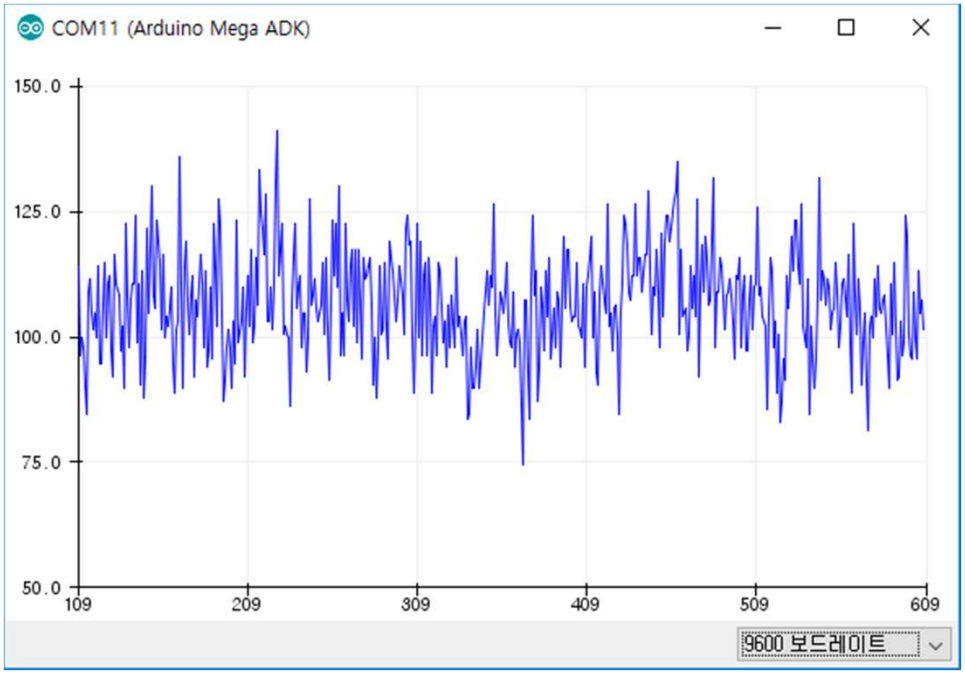
```
#define DUST_LED  A2
#define DUST_OUT  A3
int DustADC=0;
float dustDensity=0;

void setup() {
  Serial.begin(9600);
  pinMode(DUST_LED, OUTPUT);
  pinMode(DUST_OUT, INPUT);
  digitalWrite(DUST_LED, HIGH);
}
void loop() {
  DustADC=SensorRead( );
  Serial.print("Dust[ug/m3] : ");
  dustDensity = DustDensity_ugPm3(DustADC);
  Serial.println(dustDensity);
  delay(200);
}
```

```
unsigned int SensorRead(void){
  unsigned int Sensor_data;

  digitalWrite(DUST_LED, LOW);
  delayMicroseconds(280);
  Sensor_data = analogRead(DUST_OUT);
  delayMicroseconds(40);
  digitalWrite(DUST_LED, HIGH);
  delayMicroseconds(9680);
  return Sensor_data;
}

float DustDensity_ugPm3(int RawVal) {
  float Dust=(float)RawVal*(5.0/1023.0)/5.8-0.1;
  return Dust*1000;
}
```

M3-3 : Serial Monitor

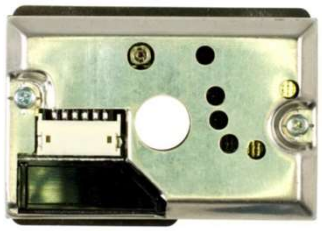


M3-3 : Serial Plotter



M3-4 : Bluetooth SPP

- 미세먼지 농도를 Bluetooth SPP로 전송하여 보자



Start	Command	Upper Data	Lower Data	End
'@'	'D'	0x--	0x--	₩n

M3-4 : Bluetooth SPP define

```
#include <SPP.h>
#include <SPI.h>

USB Usb;
BTD Btd(&Usb);
SPP SerialBT(&Btd, "YOONS-BT", "1234");

#define SPP_Packet_length 5
unsigned char SPP_data[SPP_Packet_length] = {'@', 'D', 0x03, 0xff, '₩n' };
unsigned char SPP_flag = 0;

#define DUST_LED  A2
#define DUST_OUT  A3

int DustADC=0;
float dustDensity=0;
```

M3-4 : Bluetooth SPP setup( )

```
void setup( ) {
  Serial.begin(115200);
  if (Usb.Init( ) == -1) {
    Serial.println(F("₩r₩nOSC did not start"));
    while (1);    //halt
  }
  Serial.println(F("₩r₩nSPP Bluetooth Library Started"));

  pinMode(DUST_LED, OUTPUT);
  pinMode(DUST_OUT, INPUT);
  digitalWrite(DUST_LED, HIGH);
}
```

M3-4 : Bluetooth SPP loop( )

```
void loop( ) {
  Usb.Task( );
  if (SerialBT.connected) {
    SPP_data[2]=(unsigned char)dustDensity/256;
    SPP_data[3]=(unsigned char)dustDensity%256;
    SerialBT.write(SPP_data, SPP_Packet_length);
    Serial.print("Dust[ug/m3] : ");
    Serial.println(dustDensity);
  }

  DustADC=SensorRead( );
  dustDensity = DustDensity_ugPm3(DustADC);
  delay(200);
}
```

M3-4 : Bluetooth SPP function( )

```
unsigned int SensorRead(void){
  unsigned int Sensor_data;

  digitalWrite(DUST_LED, LOW);
  delayMicroseconds(280);
  Sensor_data = analogRead(DUST_OUT);
  delayMicroseconds(40);
  digitalWrite(DUST_LED, HIGH);
  delayMicroseconds(9680);
  return Sensor_data;
}

float DustDensity_ugPm3(int RawVal) {
  float Dust=(float)RawVal*(5.0/1023.0) / 5.8 - 0.1;
  return Dust*1000;
}
```