

미세먼지 센서 프로젝트

미세먼지가 사회적인 문제가 되고 있는 가운데 전문적인 장비가 아닌 저렴한 센서와 장비들로 비교적 간단하게 미세먼지를 측정하고 분석할 수 있다. 오차가 일정 부분 포함되어 있기 때문에 전문적인 용도로 사용하기에는 다소 부족하지만 충분한 수준의 정밀도가 필요하지 않은 분야에서 활용하기에는 충분하다.

1. 미세먼지 센서에 관해서

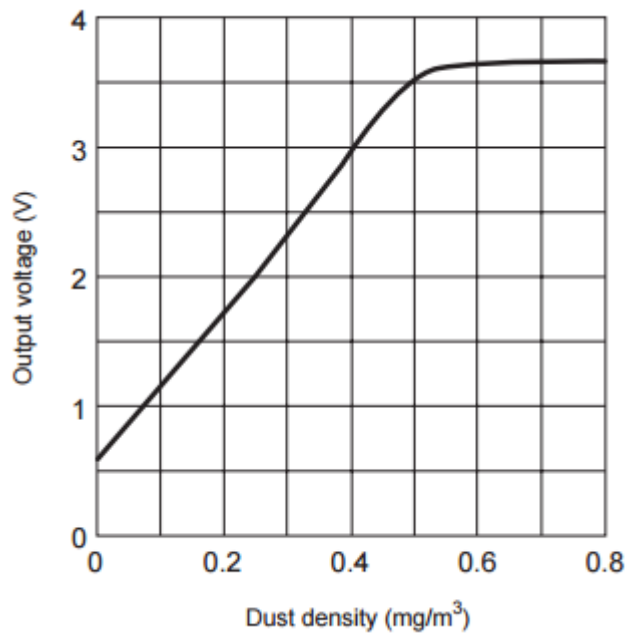
여기서는 샤프에서 출시한 GP2Y10 먼지 센서를 활용해서 먼지를 측정해 본다.



<그림1-1> GP2Y10 먼지센서

GP2Y10 은 담배 연기 등 미세 입자를 검출할 수 있는데 전류 소모량이 적고 (20mA max 11mA typical) DC 7V 까지의 전원으로 동작 할 수 있다. 출력은 아날로그 값으로 출력되는데 아두이노에서는 내장 되어 있는 ADC(Analog Digital Converter)를 사용해서 이 값을 바로 분석할 수 있다. 만약 라즈베리파이를 사용해서 분석하기 위해서는 별도로 아두이노를 사용하거나 혹은 외부에 ADC 를 추가해서 사용할 수 있다.

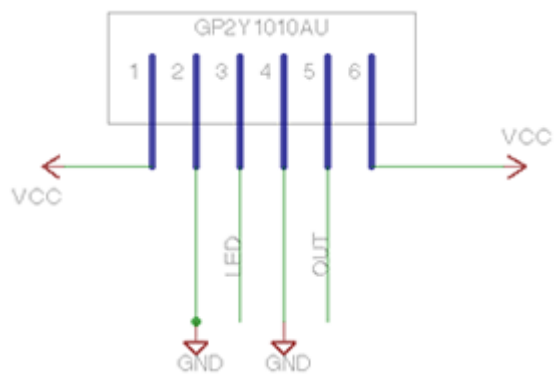
GP2Y10 은 $0.5V/0.1mg/m^3$ 의 감도를 가지고 있다. 출력값은 전압으로 출력되는데 아래 그래프에서 값을 참조할 수 있다.



<그림 1-2>GP2Y10 데이터 시트

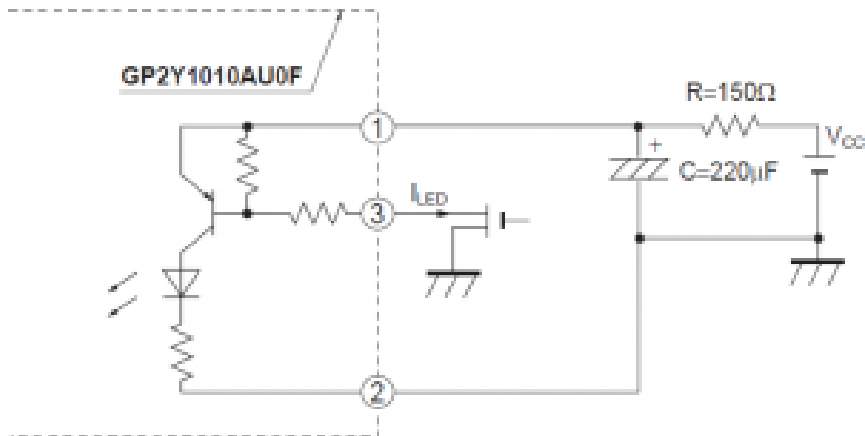
자세한 내용은 https://www.sparkfun.com/datasheets/Sensors/gp2y1010au_e.pdf 의 데이터 시트를 참조할 수 있다.

GP2Y10 은 적외선 LED 를 활용해서 먼지를 검출하는 광학식 센서 이므로 직사 광선이 바로 비치는 곳에서 사용하는 것은 좋지 않다.



<그림 1-3>GP2Y10 핀 배열

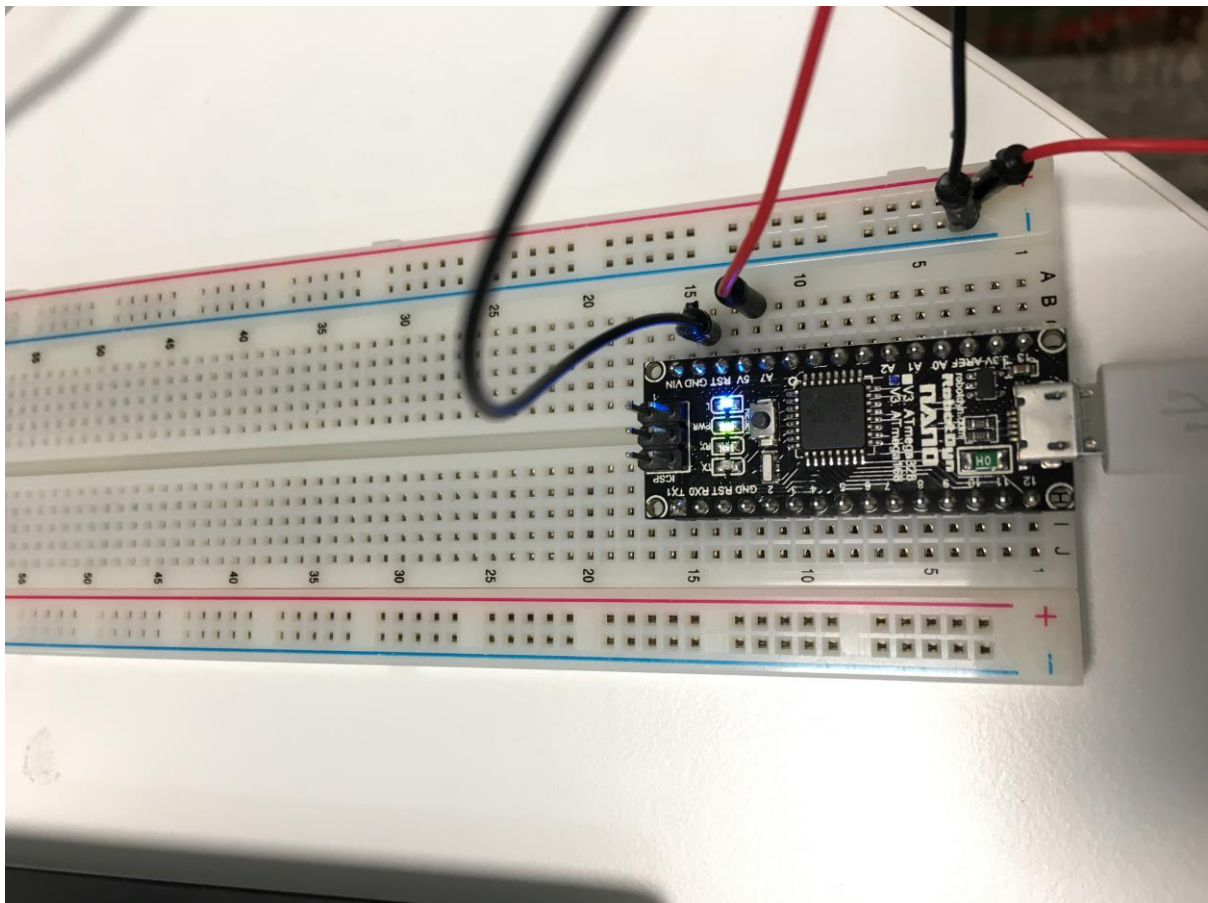
GP2Y10 은 6 개의 핀을 가지고 있으며 왼쪽 3 개의 핀은 내장된 적외선 LED 를 출력하기 위한 핀이다. 그리고 오른쪽 3 개의 핀은 출력 값을 받기 위한 핀들이다. 한 가지 주의 할 점은 1~3 핀을 연결 할 때에는 150 옴의 저항과 220uF 의 콘덴서가 함께 연결해야 한다.



<그림 1-4>GP2Y10의 내부 구조

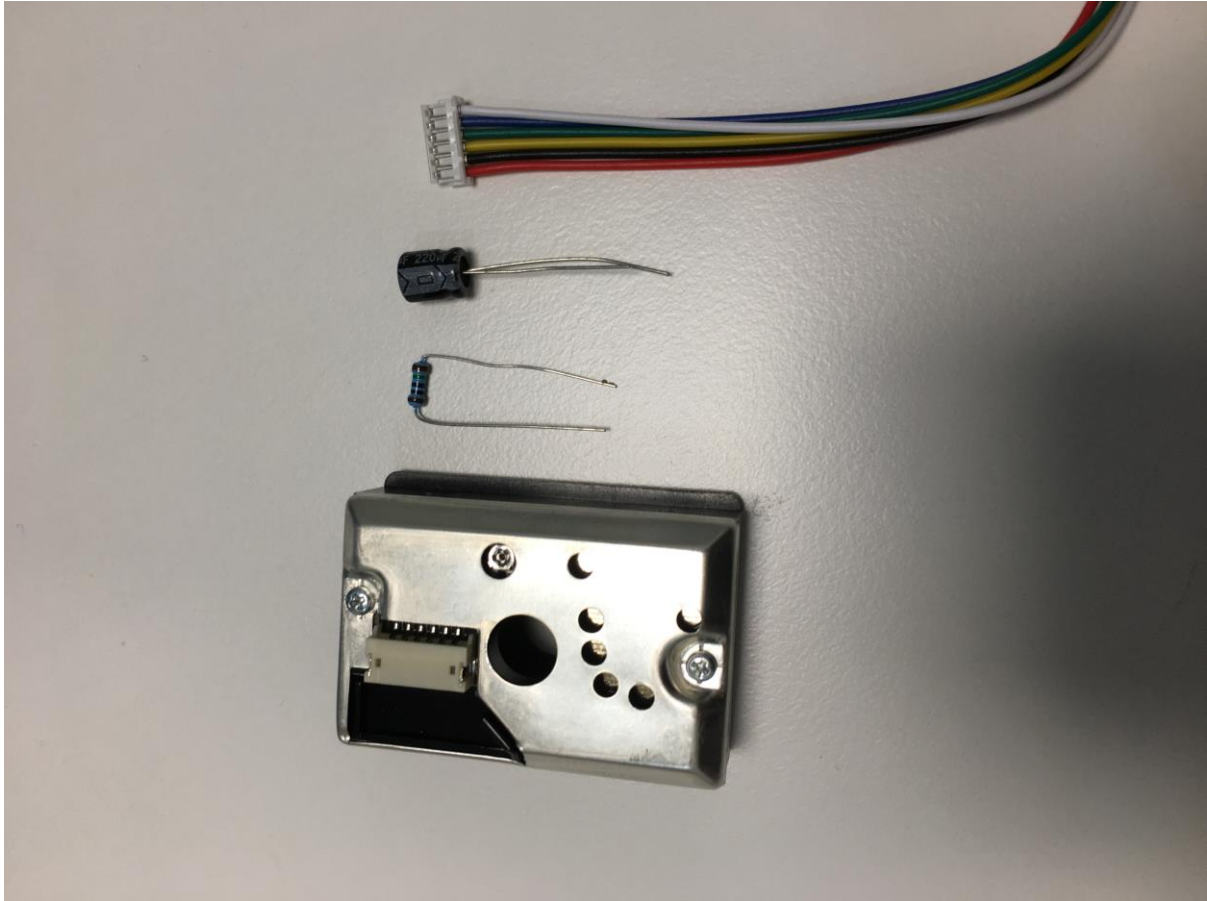
핀이 많아서 연결하기 어렵겠지만 천천히 하나씩 연결해 본다. 먼지센서의 적외선 LED 핀(3 번)은 아두이노의 디지털 2 번으로 연결하고 먼지센서의 출력값을 나타내는 6 번 은 아두이노의 아날로그 0 번 핀으로 연결한다.

2. 아두이노와 미세먼지 센서 회로 구성하기



<그림 2-1>Arduino 장착 및 전원선 노출

회로를 구성하기 위해서 Arduino 나노를 브레드보드에 장착한다. 또 앞으로 작업을 위해서 5V, GND 핀을 사용하기 좋게 빼놓는다.



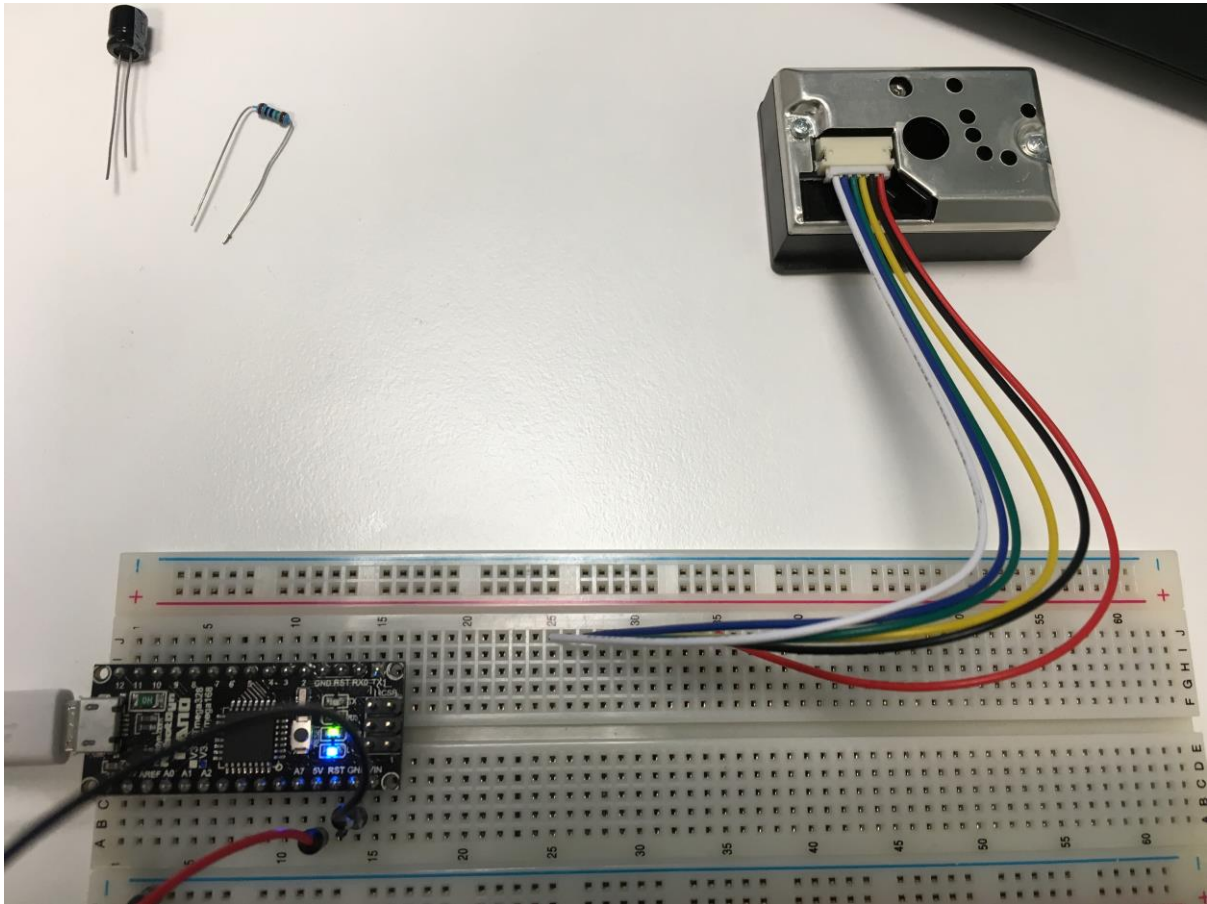
<그림 2-2>먼지센서와 부속품

먼지센서와 부속품을 확인한다. 회로를 구성하기 위해서는 저항과 콘덴서가 필요하다. 저항은 브레드 보드에 장착하기 용의 하게 구부려 놓는다.



<그림 2-3> 먼지 센서에 케이블을 연결한다.

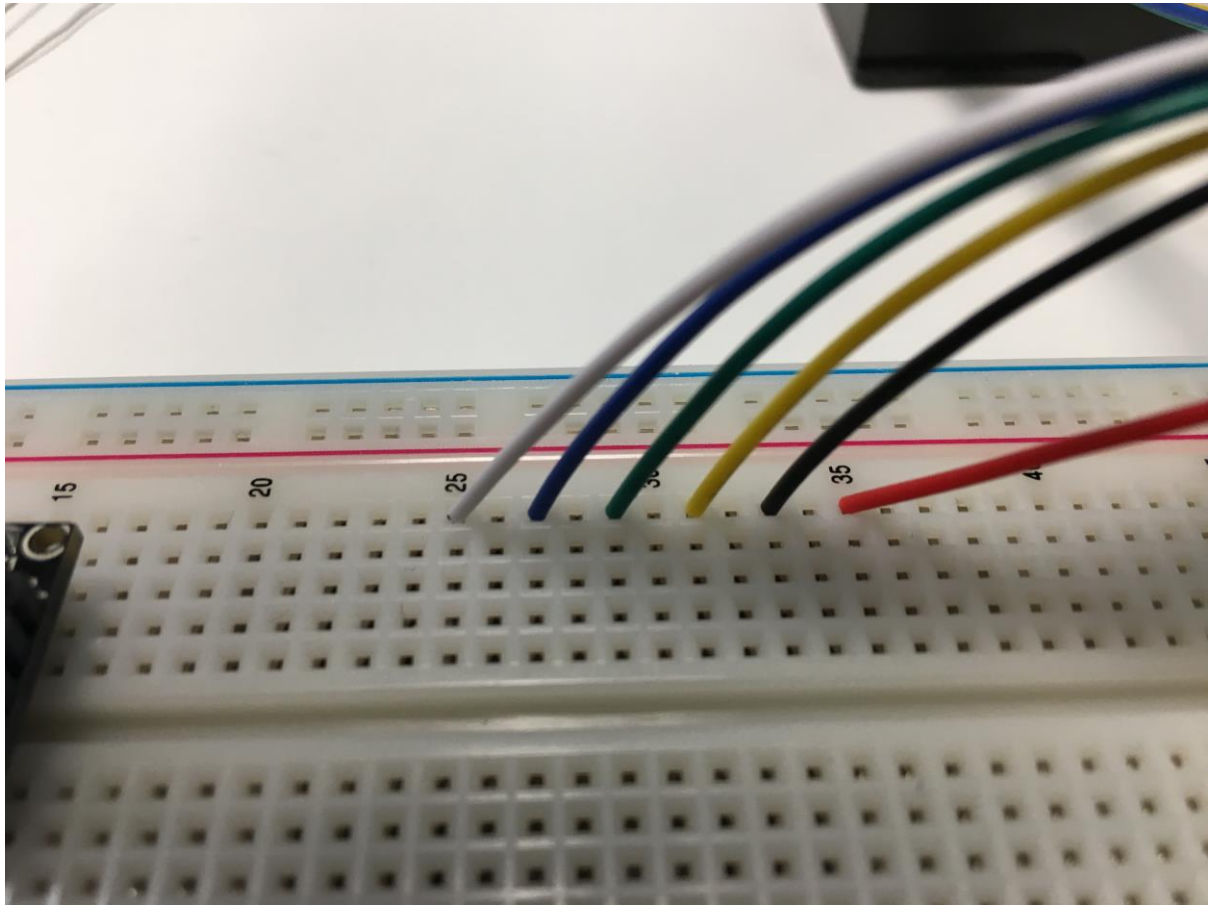
먼지 센서에 케이블을 연결한다. 케이블은 왼쪽이 흰색이 되도록 하며 왼쪽부터 1~6 번으로 번호가 부여되어 있다.



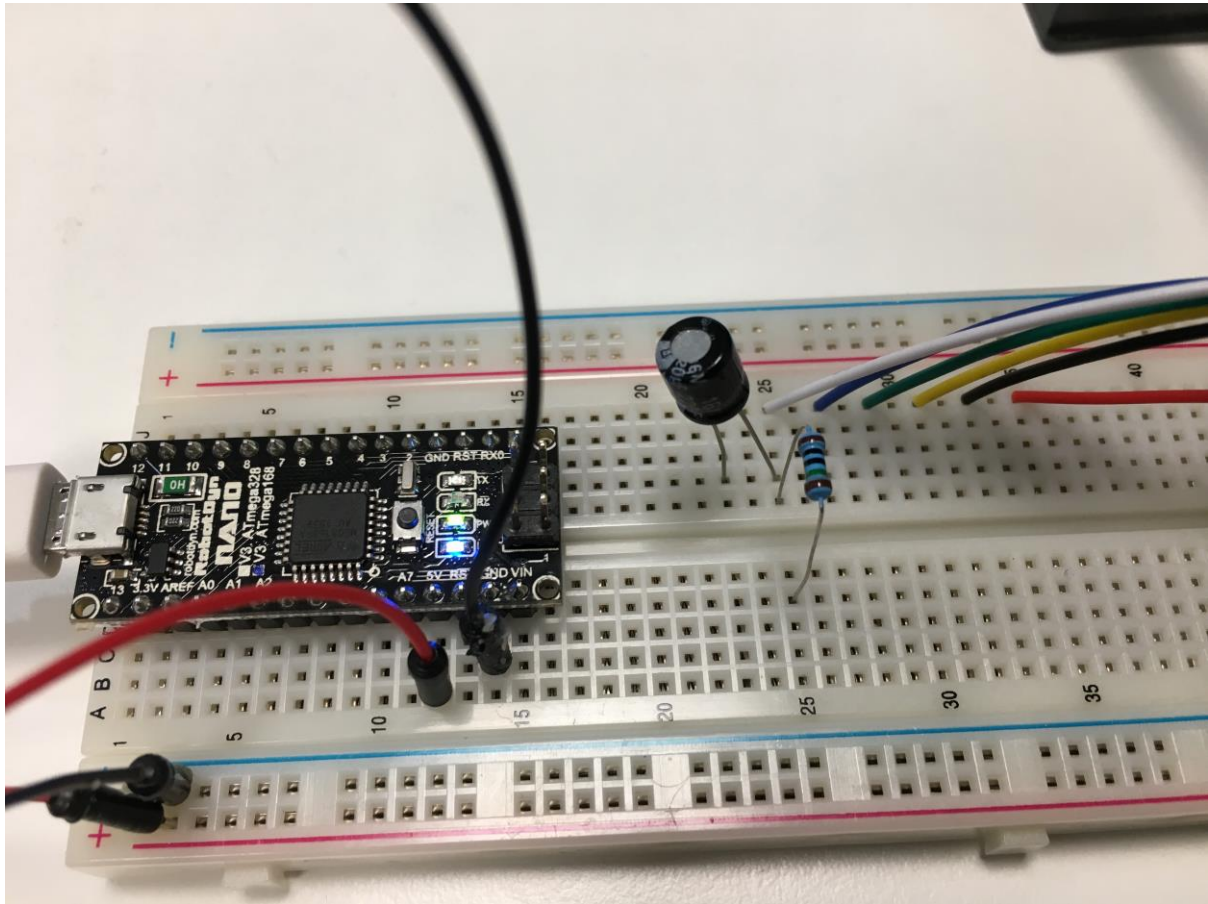
<그림 2-4> 먼지센서 연결

먼지센서와 브레드 보드를 연결한다. 원래는 브레드보드의 번호는 특별한 의미는 없지만 여기서는 설명을 위해서 아래와 같이 연결해 본다.

먼지 센서	브레드 보드
1	J, 25
2	J, 27
3	J, 29
4	J, 31
5	J, 33
6	J, 35



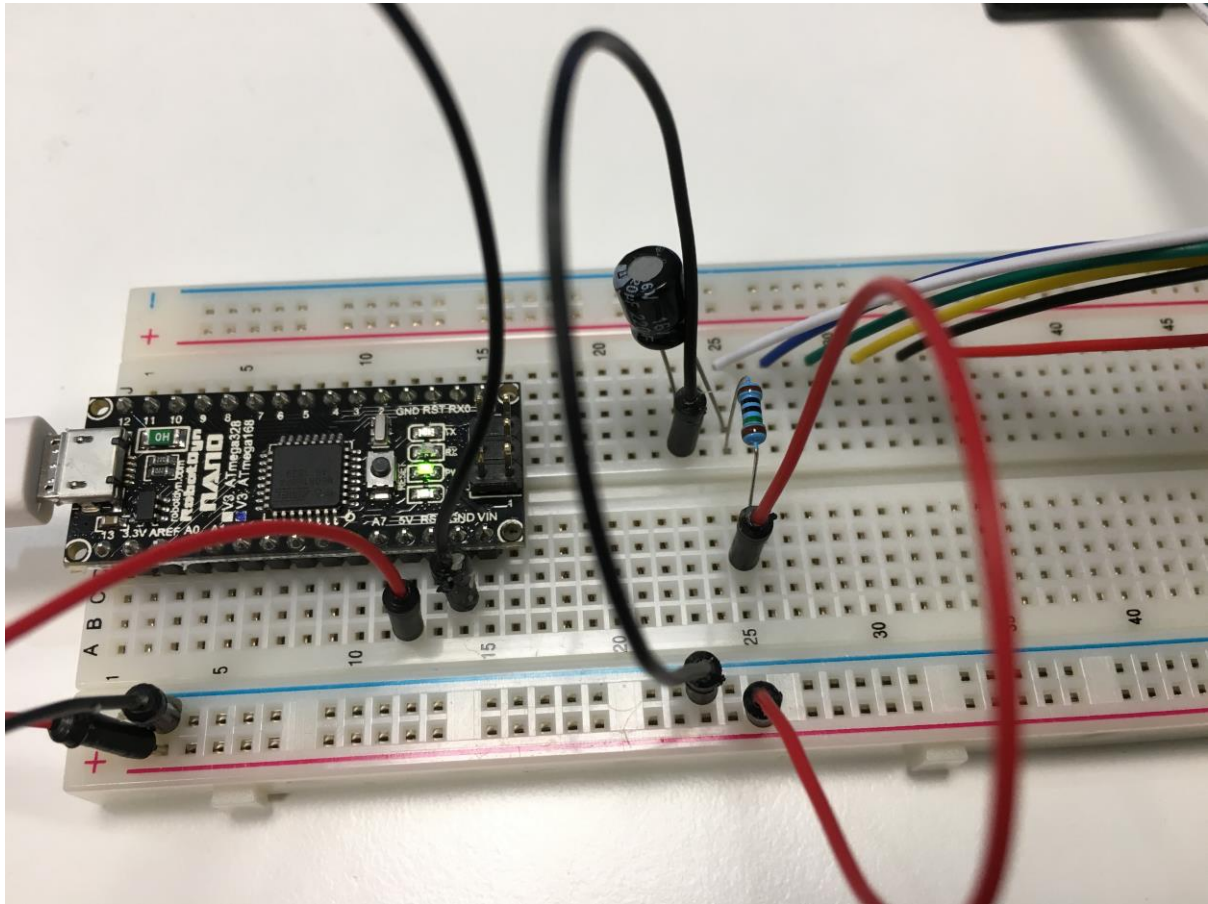
<그림 2-5> 먼지 센서가 연결된 모습



<그림 2-6>저항과 콘덴서의 연결

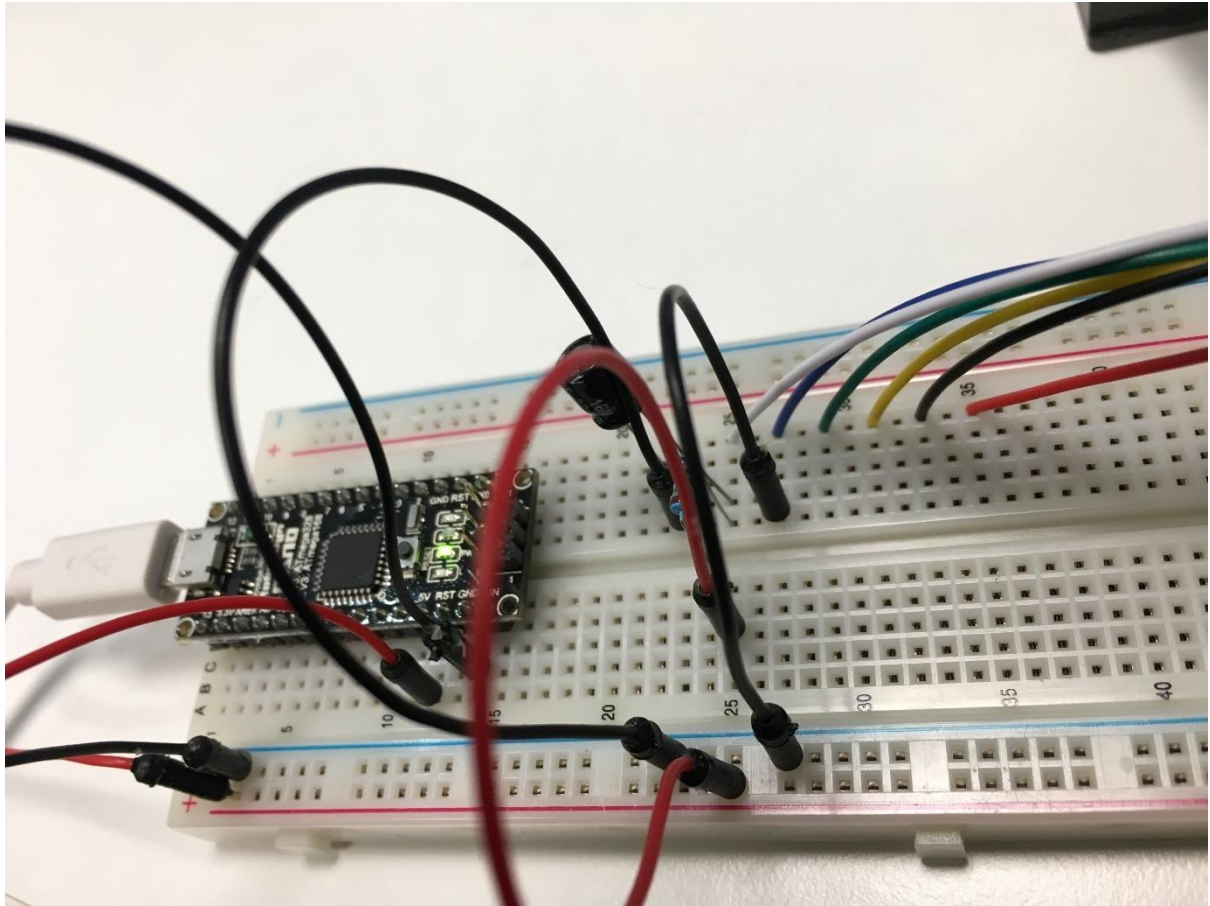
저항과 콘덴서를 연결한다.

저항	
F, 25	D, 25
콘덴서(+)	콘덴서(-)
G, 25	G, 23



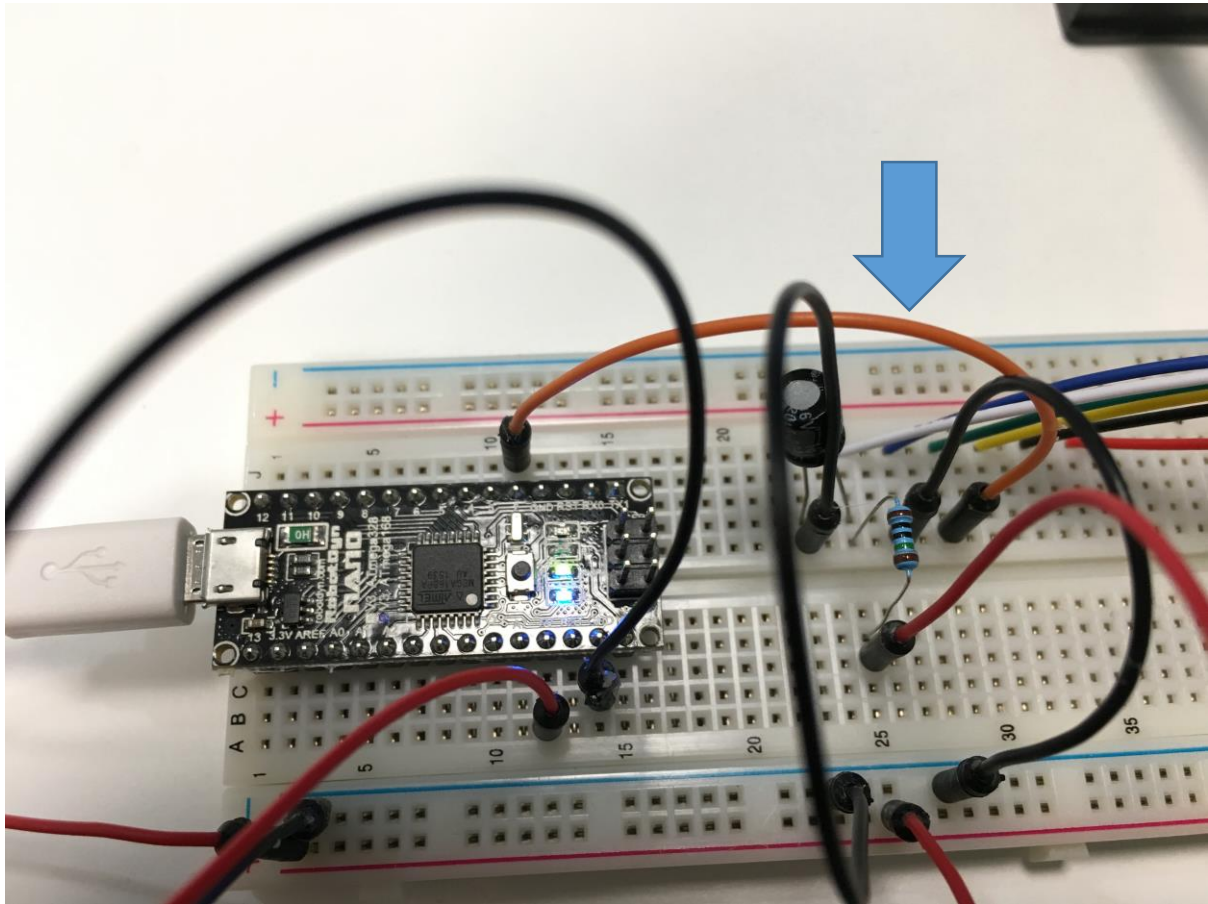
<그림 2-7>콘덴서와 저항에 전원 연결

5V 전원	저항
+, 25	C, 25
GND	콘덴서 (-)
-, 23	F, 23



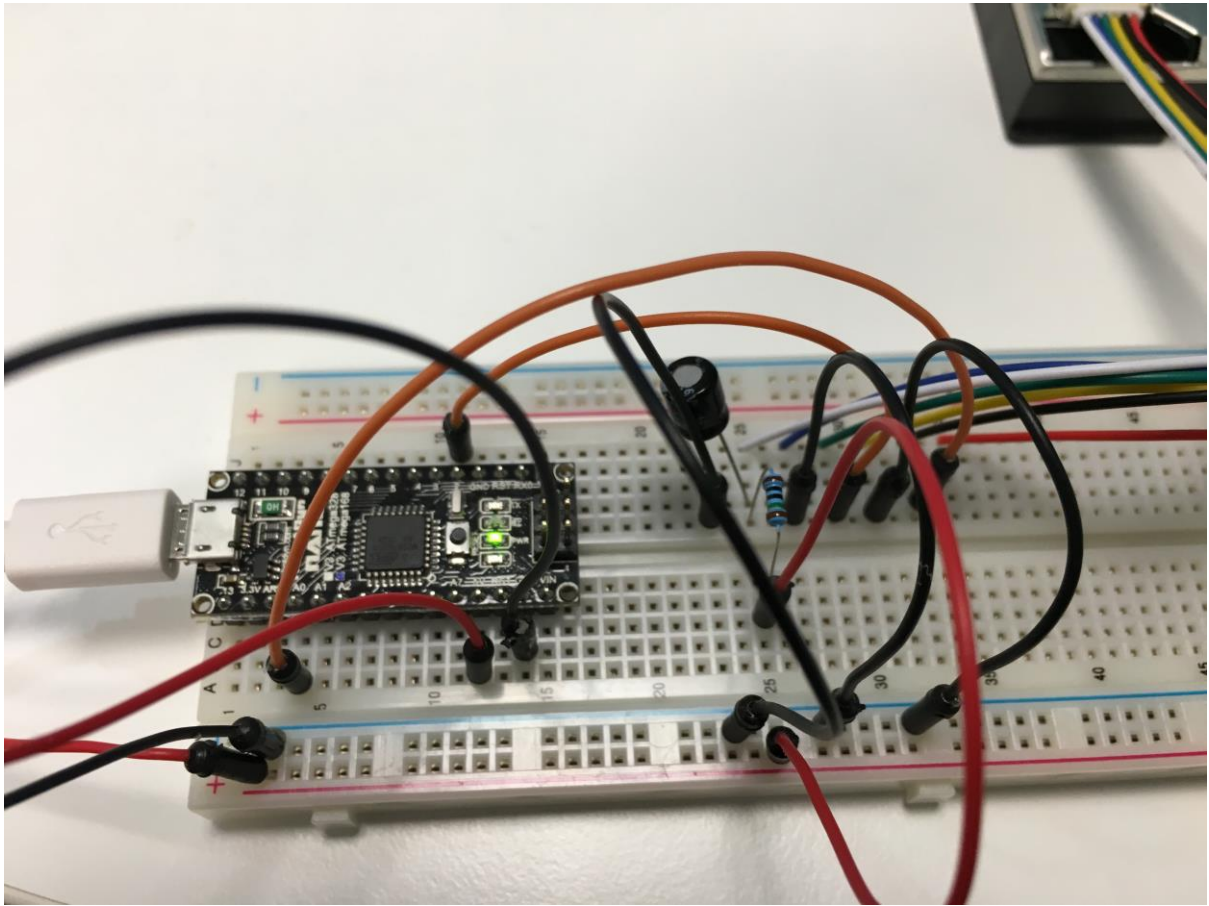
<그림 2-8> 먼지 센서의 2 번 핀과 GND 의 연결

GND	먼지센서(2 번)
-, 27	F, 27



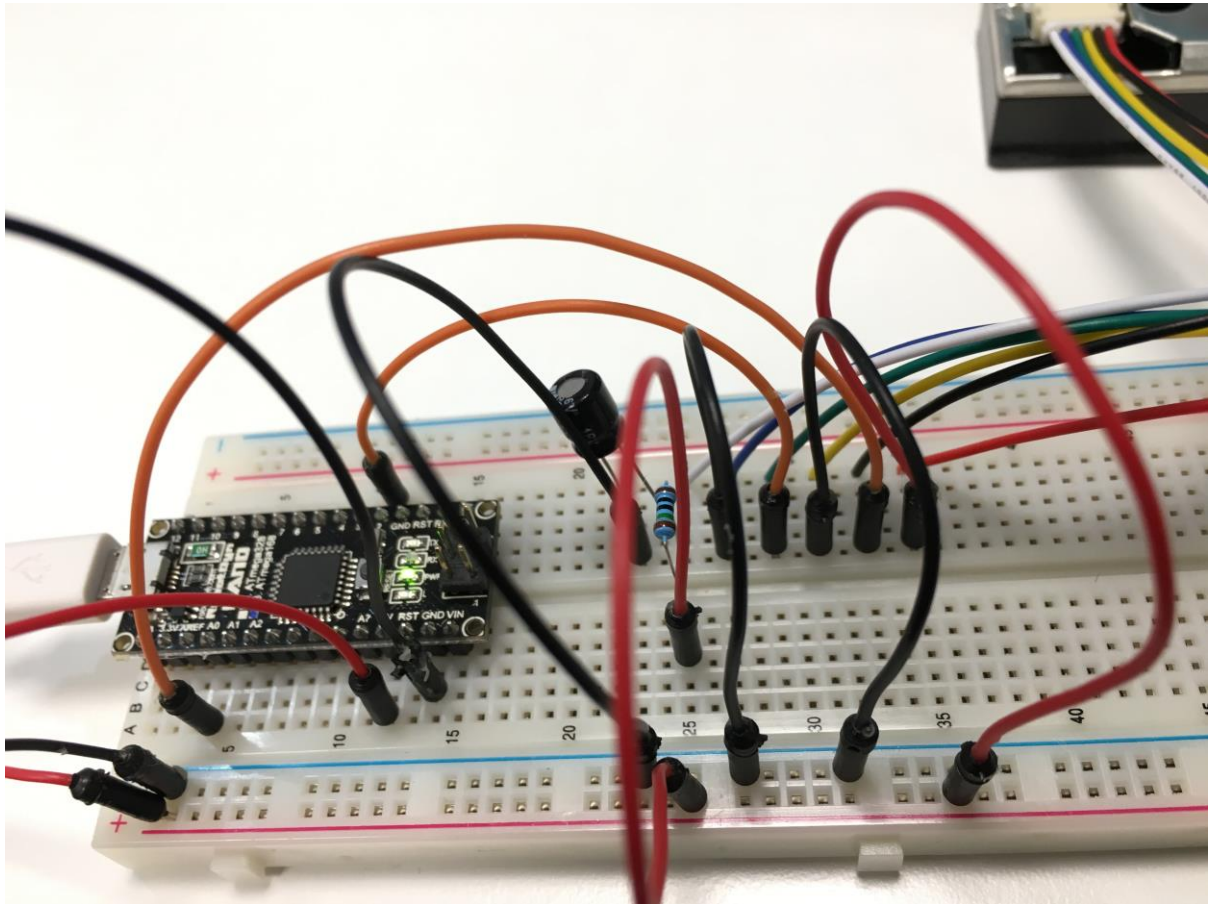
<그림 2-9>아두이노의 디지털 2 번 핀과 먼지센서의 3 번 핀을 연결한다.

ArduinoD2	먼지센서(3 번)
J, 11	F, 29



<그림 2-10> 먼지센서 4 번 핀과 GND 연결

GND	먼지센서(4 번)
-, 31	F, 31



<그림 2-11>출력선의 연결과 전원의 연결

이제 출력값과 전원을 연결하면서 작업을 마무리 할 차례이다.

ArduinoA0	먼지센서(5 번)
A, 4	F, 33
5V	먼지센서(6 번)
+, 35	F, 35

Arduino 코드 작성

전자 부품의 연결이 끝나고 나면 아래와 같이 코드를 입력한다.

```

int dustPin=0;
float dustVal=0;
float dustDensity = 0;

int ledPower=2;
int delayTime=280;
int delayTime2=40;
float offTime=9680;

void setup(){
    Serial.begin(9600);
    pinMode(ledPower,OUTPUT);
    pinMode(4, OUTPUT);
}

void loop(){
    // ledPower is any digital pin on the arduino connected to Pin 3 on the sensor
    digitalWrite(ledPower,LOW); // power on the LED
    delayMicroseconds(delayTime);

    dustVal=analogRead(dustPin); // read the dust value via pin 5 on the sensor
    delayMicroseconds(delayTime2);

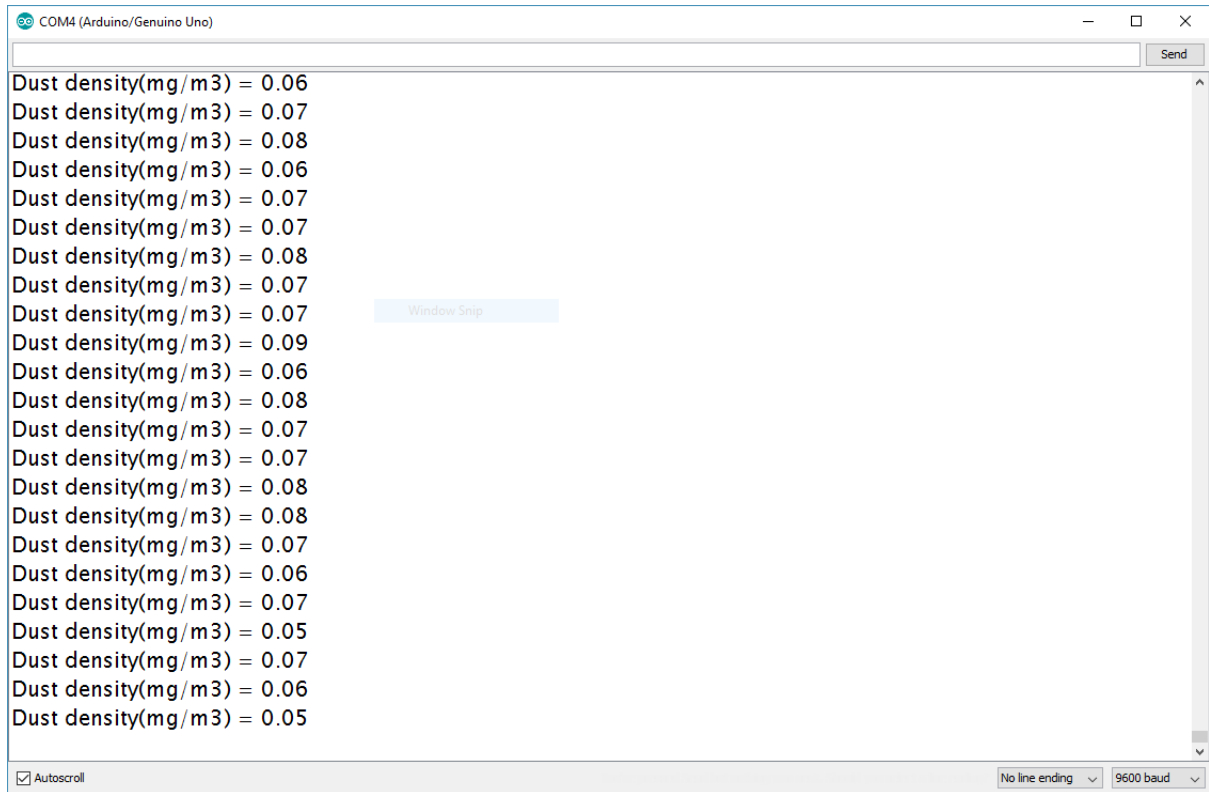
    digitalWrite(ledPower,HIGH); // turn the LED off
    delayMicroseconds(offTime);

    delay(3000);

    dustDensity = 0.17*(dustVal*0.0049)-0.1;
    Serial.print("Dust density(mg/m3) = ")
    Serial.println(dustDensity);
}

```

여기까지 잘 입력되었으면 실행하면 아래와 같이 결과 값이 출력된다.



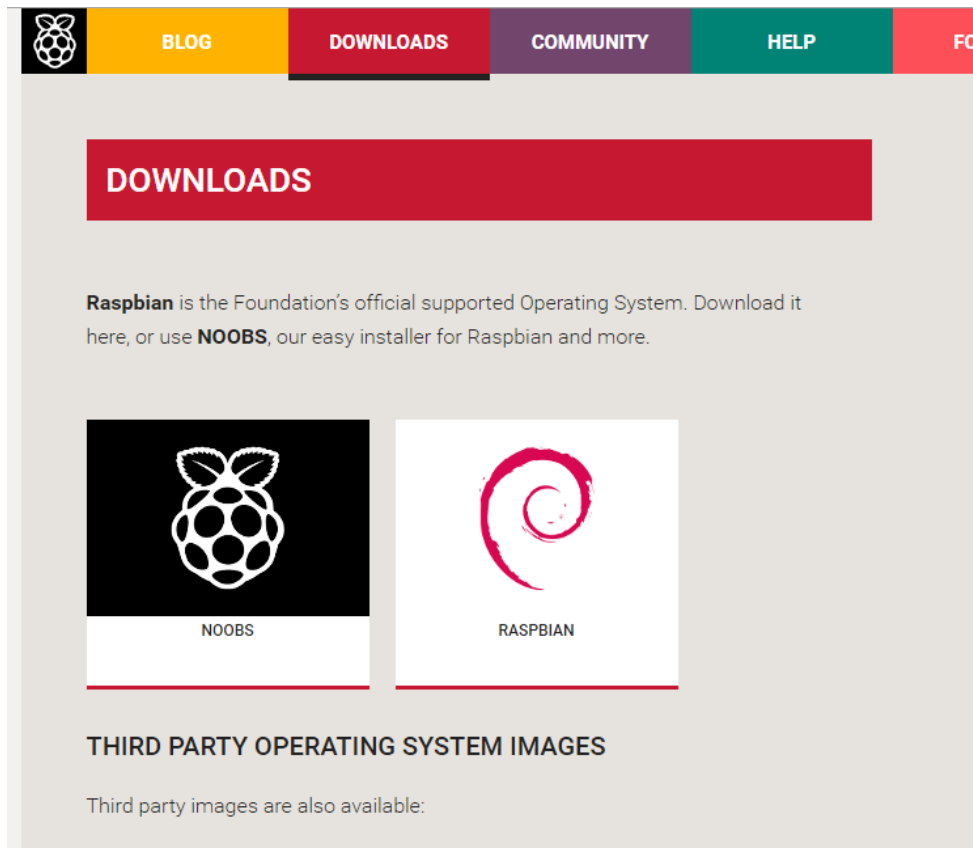
```
COM4 (Arduino/Genuino Uno)
Dust density(mg/m3) = 0.06
Dust density(mg/m3) = 0.07
Dust density(mg/m3) = 0.08
Dust density(mg/m3) = 0.06
Dust density(mg/m3) = 0.07
Dust density(mg/m3) = 0.07
Dust density(mg/m3) = 0.08
Dust density(mg/m3) = 0.07
Dust density(mg/m3) = 0.07
Dust density(mg/m3) = 0.09
Dust density(mg/m3) = 0.06
Dust density(mg/m3) = 0.08
Dust density(mg/m3) = 0.07
Dust density(mg/m3) = 0.07
Dust density(mg/m3) = 0.08
Dust density(mg/m3) = 0.08
Dust density(mg/m3) = 0.07
Dust density(mg/m3) = 0.06
Dust density(mg/m3) = 0.07
Dust density(mg/m3) = 0.05
Dust density(mg/m3) = 0.07
Dust density(mg/m3) = 0.06
Dust density(mg/m3) = 0.05
```

Raspberry Pi 에 Raspbian 설치하기

먼지센서를 연결하고 데이터를 가져오는 것은 아두이노가 담당하고 실제 데이터를 전송하는 역할은 Raspberry Pi 가 담당한다. 이런 역할 분담을 센서 노드(Sensor node) 그리고 게이트웨이(Gateway)로 분리하는 형태로 구성하는 게 일반적이다.

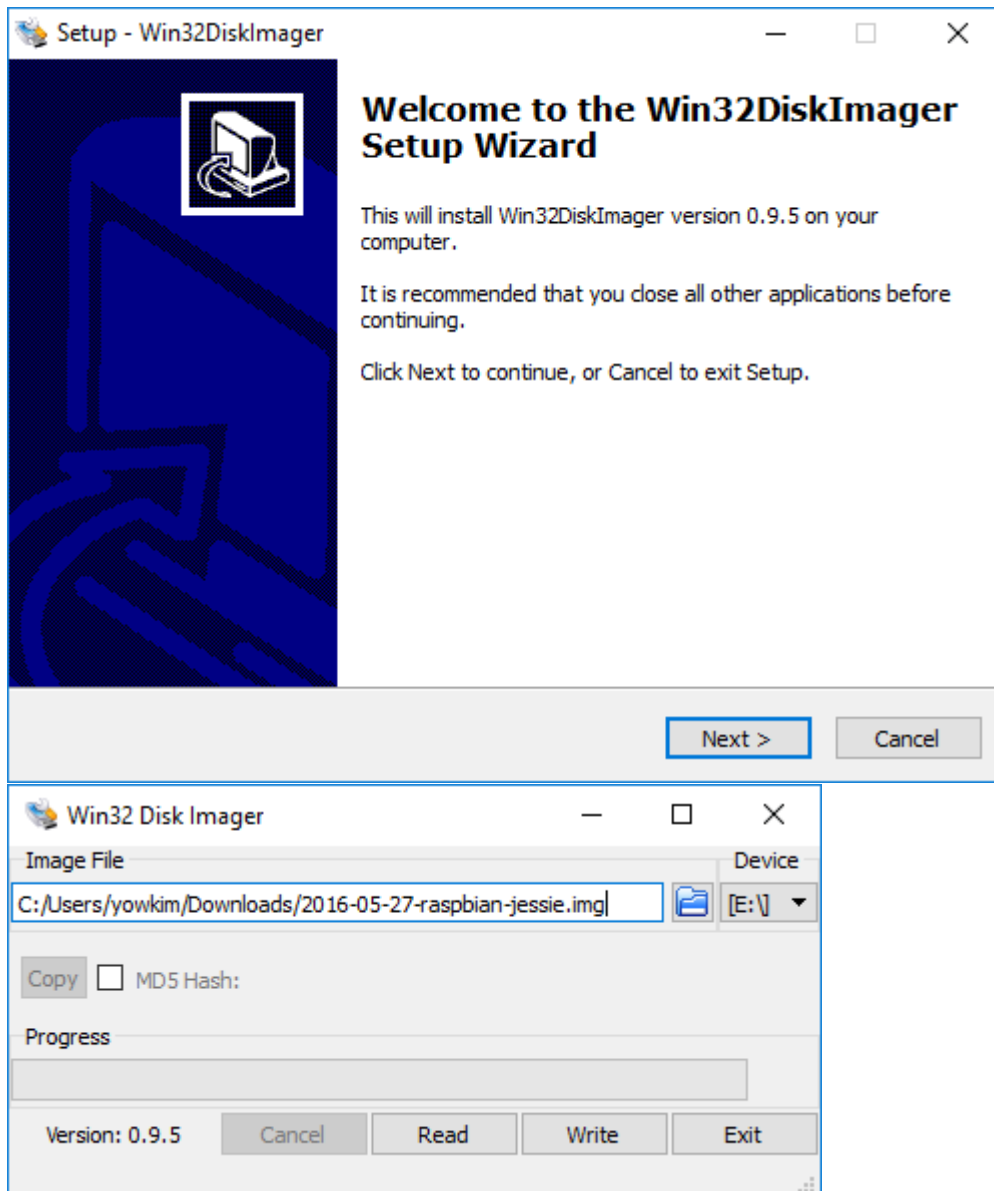
Raspberry Pi 는 하나의 완전한 컴퓨터의 구조를 가지고 있기 때문에 OS 를 설치해야 한다. Raspberry Pi 에 설치할 수 있는 OS 중에 가장 많이 알려진 OS 가 Linux 계열의 Raspbian 이다. Raspbian 은 Linux 배포판 중의 하나인 Debian(<https://en.wikipedia.org/wiki/Debian>)을 기반으로 하고 있다.

먼저 Raspbian 을 다운로드 한다. <https://www.raspberrypi.org/downloads/> 에서 Raspbian 을 다운로드 한다. 다운로드 받은 이후에는 zip 파일 형태로 압축되어 있는 파일의 압축을 풀어준다.



Raspbian 을 설치하기 위해서는 4 기가 이상의 MicroSD 메모리를 준비한다.
다운받은 Raspbian 은 이미지 형태로 바로 SD 메모리에 옮길 수 없다.
Win32DiskImager 를 사용해서 MicroSD 메모리에 복사한다. Win32DiskImager
다운로드

경로 https://sourceforge.net/projects/win32diskimager/?source=typ_redirect

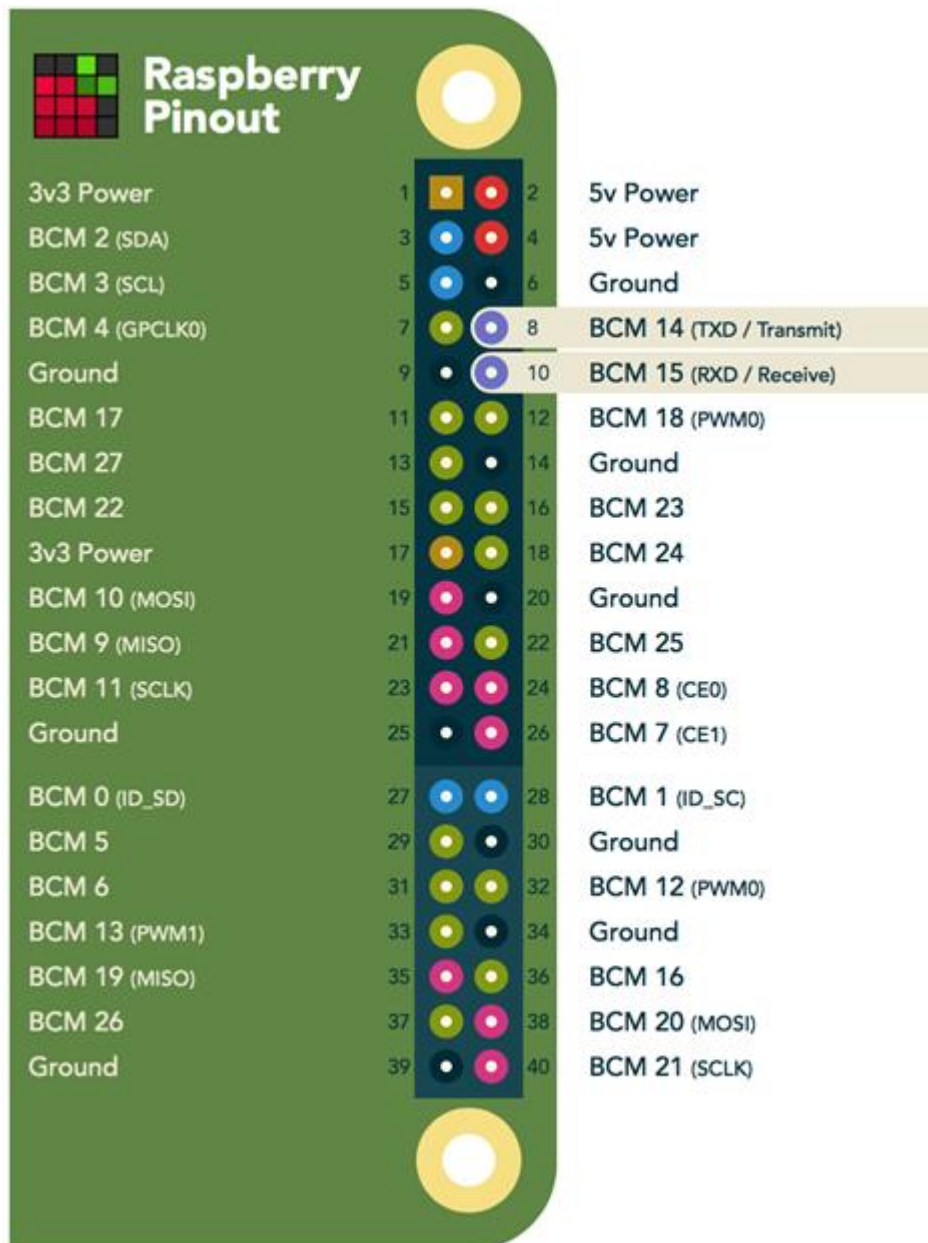


여기까지 잘 진행되었으면 MicroSD 메모리를 Raspberry Pi 에 설치해서 부팅할 수 있게 준비가 끝났다.

USB to Serial Cable 로 Raspberry Pi 연결하기

Raspberry Pi Console 로 Serial 로 바로 연결하는 방법이 USB to Serial 로 연결하는 방법이다. 좀 더 정확하게는 UART(Universal asynchronous receiver/transmitter)로 말 할 수 있는데 (UART 설명 : <https://ko.wikipedia.org/wiki/UART>) Raspaberry Pi GPIO(General Purpose Input/Output) 핀을 사용해서 직접 통신 할 수 있게 해준다.

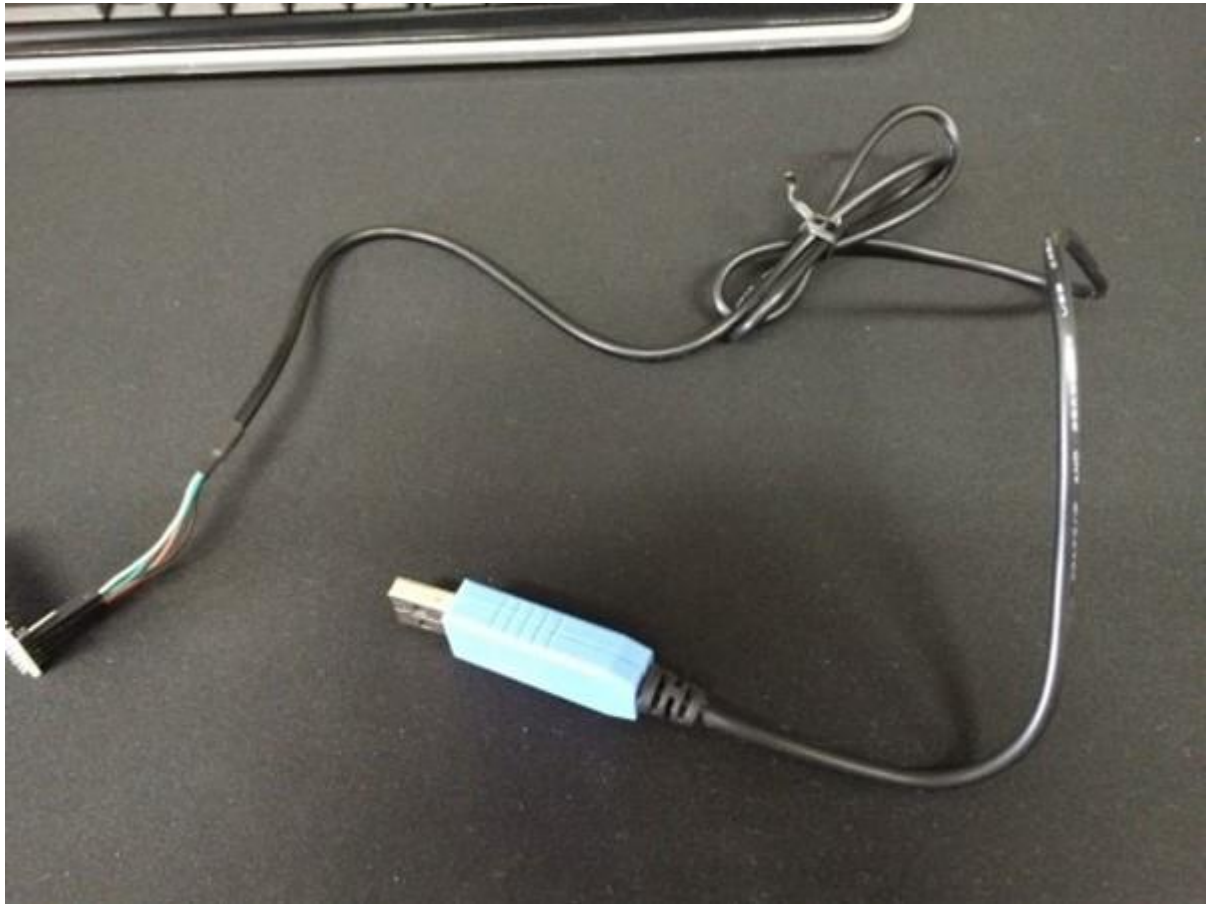
Raspberry Pi 의 UART 핀은 BCM 14, BCM 15 이다.



먼저 USB to Serial 제품을 준비한다. 보통 RS232 TTL(Transistor-Transistor Logic) 와 같이 표기 되어 있는 제품들을 사용할 수 있는데 윈도우에서 인식이 안되면 PL2303HX 드라이버 설치해야 한다. 대부분의 자동으로 설치하긴 하는데 간혹 안되는 제품이 있을 경우에는 아래 링크에서 설치해 주어야 한다.

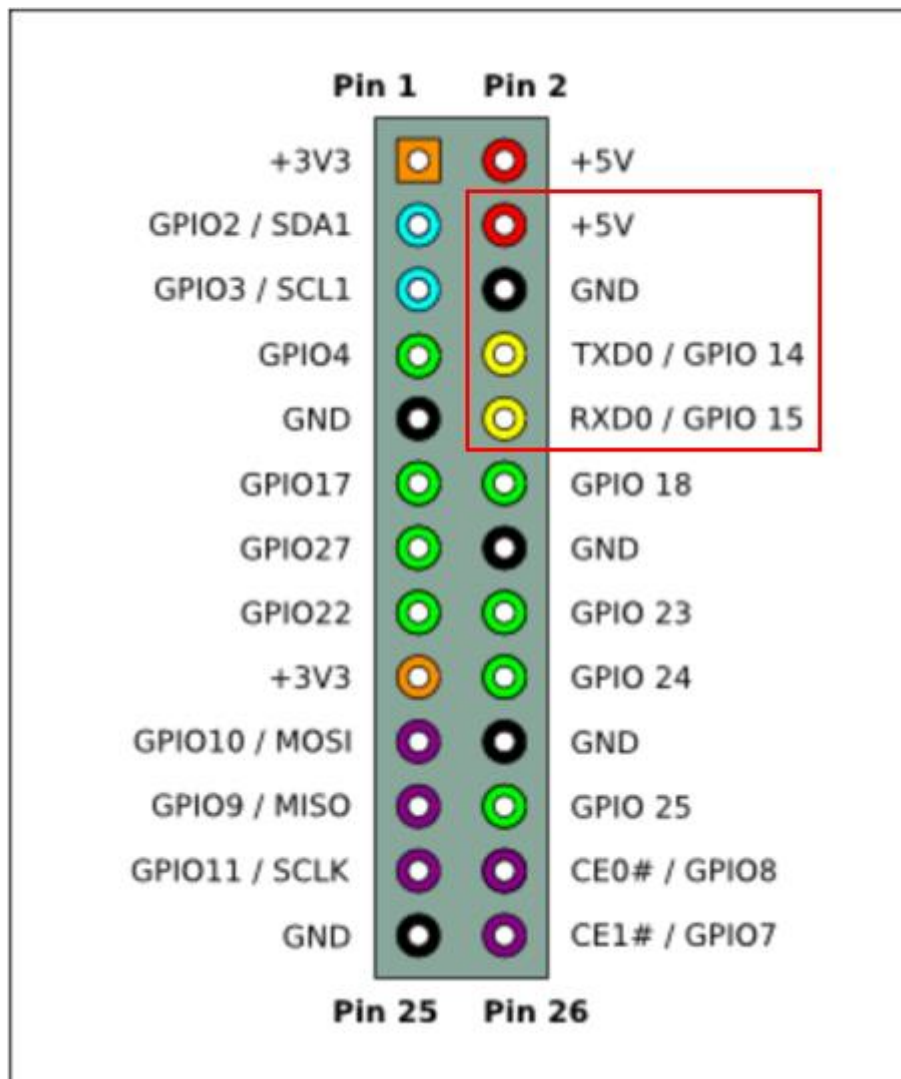
PL2303HX

드라이버 : http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=225&pcid=41

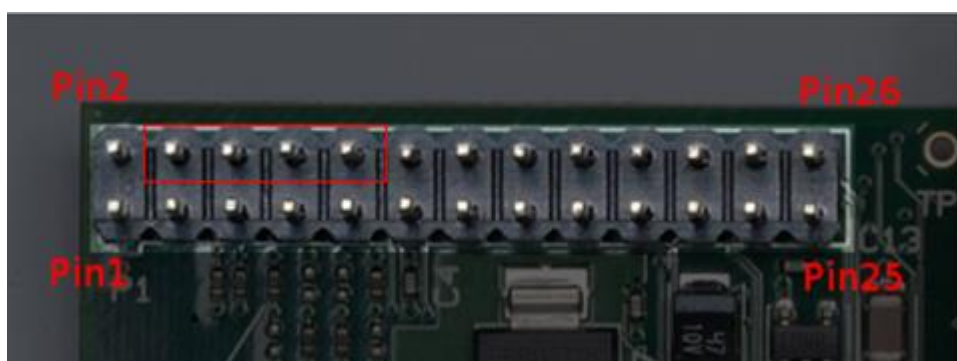


USB to Serial 은 3.3v, TXD, RXD, GND, 5v 로 구성 되어 있는데 4 개의 선으로 구성되어 있다.

빨간선	VCC(3.3 or 5V)
검은선	GND
녹색선	TXD
흰색선	RXD



출처 : http://elinux.org/RPi_Low-level_peripherals



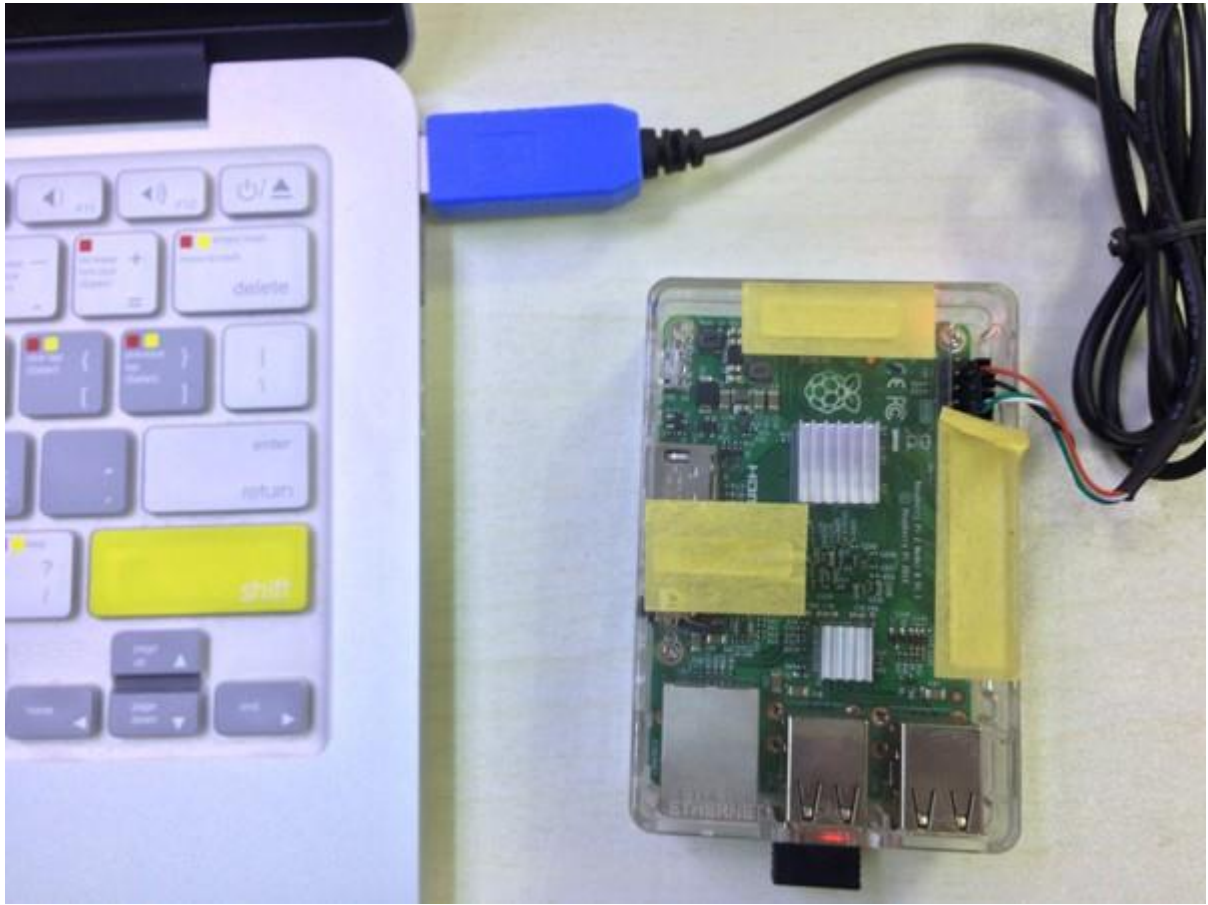
출처 : http://elinux.org/RPi_Low-level_peripherals

사용할 핀은 2 개인데(RX, TX) 5v, GND, TX, RX 를 전부 연결한다. (별도 전원연결시에는 5v 핀 제거)

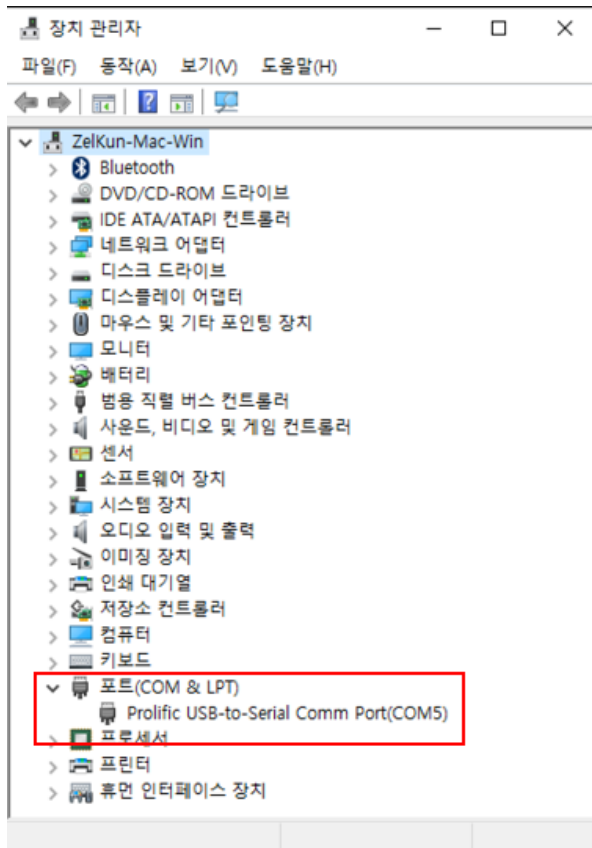
실제 연결할 때에는 아래 표와 같이 연결하면 된다.

PI	RS232
5V	5V (적색)
GND	GND (검정색)
TX	RX (녹색)
RX	TX (흰색)

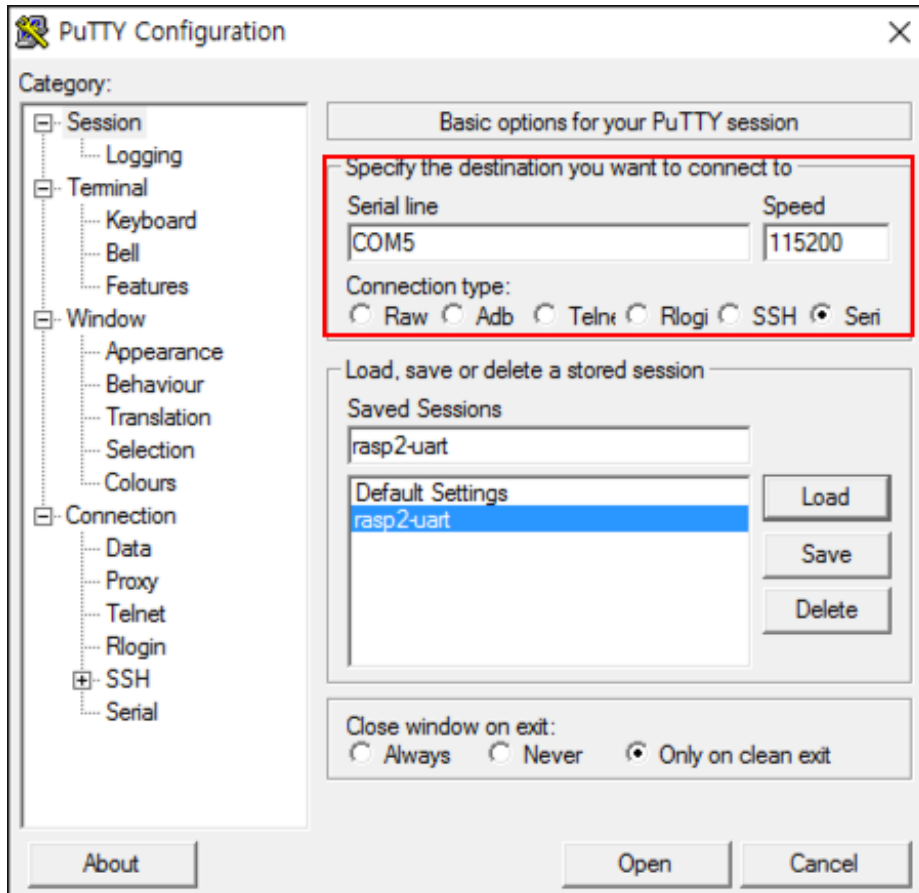




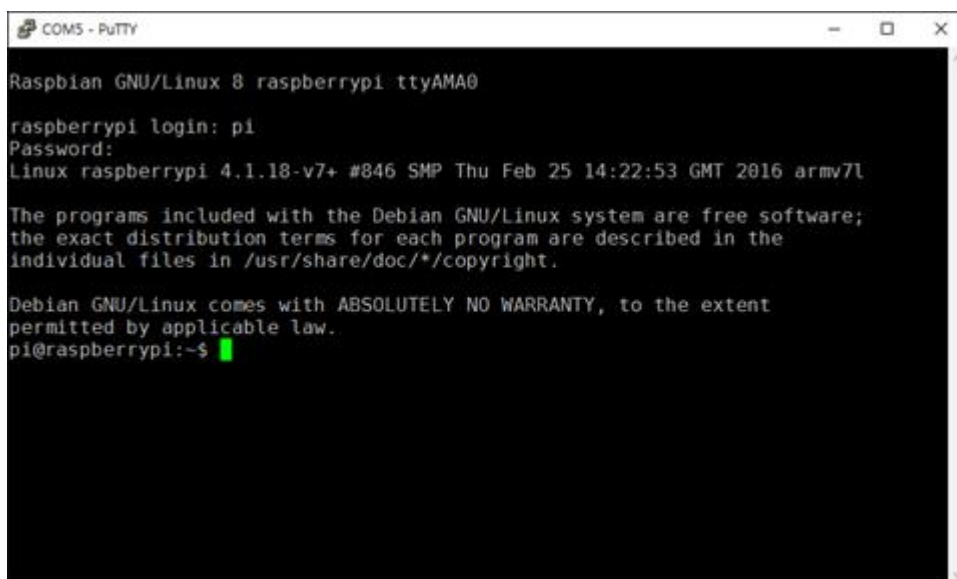
연결이 끝나면 접속을 위해서 관리 > 장치 관리자에서 연결된 Port 를 확인한다.



시리얼로 연결하기 위해서 Putty 를 설치하는데 Putty 는 무료로 다운로드 가능하다. Putty 는 <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html> 에서 다운로드 가능하다.



Serial Line 는 장치 관리자에서 확인한 포트 넘버로 설정하고 Speed 는 115200 으로 설정한다. 그리고 Connection Type 을 Serial 을 선택하면 연결 준비가 완료 된다. 자주 연결하기 위해서는 미리 Save 해두는 것도 좋다.



Raspberry Pi Network setting

아래와 같이

```
$ sudo /etc/network/interfaces  
  
...  
  
iface wlan0 inet dhcp  
  
wpa_ssid SSID-GOES-HERE  
  
wpa_psk WIFI-PASSWORD-GOES-HERE
```

재부팅을 하거나

```
$ sudo /etc/init.d/networking restart  
  
$ sudo ifup wlan0
```

인터넷이 되는지 확인합니다.

```
$ ping yahoo.com
```

여기까지 잘 설정이 되었다면 Raspberry Pi 의 설정이 잘 끝났다.

만약 위의 방식이 잘 되지 않는다면

```
network={  
    ssid="DCC Free Wifi"  
    key_mgmt=NONE  
}
```

와 같이 해결할 수도 있다.

이제 마지막으로 아두이노에서 넘어오는 수치를 받기 위해서 먼저 아두이노와 관련된 라이브러리를 설정한다.

```
sudo apt-get install arduino
```

이제 파이썬 코드를 작성한다. Nano test.py

```
import serial
```

```
import requests
```

```
port = '/dev/ttyUSB0'
```

```
device = 'device1'
```

```
serialFromArduino = serial.Serial(port, 9600)
```

```
serialFromArduino.flushInput()
```

```
while True:
```

```
    input_s = serialFromArduino.readline()
```

```
    print(input_s[0:4])
```

```
    dust = str(input_s[0:4], 'utf-8')
```

```
    requests.get('http://projectdust.azurewebsites.net?d=' + device + "&dust=" + dust )
```

이제 Python 코드를 실행 시켜서 잘 실행되는지 확인한다.

```
sudo python3 test.py
```