



인하공업전문대학
INHA TECHNICAL COLLEGE

무선 네트워크 3주차

인하공업전문대학 컴퓨터 정보과
김한결

- 요약

- I. 아두이노 조도, 미세먼지, DHT11
- II. 아두이노 라즈베리파이 설치 및 프로그래밍
- III. 아두이노 라즈베리파이 Serial 통신 및 제어
- IV. InfluxDB 설치
- V. Grafana 설치 및 모니터링

❖ 아두이노

✓ 라즈베리파이 아두이노 IDE 설치

- `sudo apt-get install arduino`

✓ 아두이노 시뮬레이션 사이트

- <https://www.tinkercad.com/>

❖ 아두이노 && 조도센서



포토 레지스터
(PhotoResistor),
황화카드뮴(Cds)



광 포토셀 CdS 조도 센서 모듈 스펙

용도	빛이 많은지 적은 지 파악하기
동작 전압	5V
핀 구성	4핀 (GND / VCC / DO / AO)
특징	가변저항을 조절하여 감도 조정 가능

❖ 조도센서

- 조도센서는 주변의 밝기를 측정하는 센서로 극성이 없으나 빛의 양에 따라 전도율이 변하는 가변 저항 센서
- 조도 센서의 경우 사용시 저항을 연결해야함.
- 밤이 되면, 자동으로 켜지는 가로등, 자동차의 헤드라이트, 밝기에 따라 변하는 스마트폰의 화면 등에 사용

❖ 동작원리

$R = V/I$	R: 저항 V: 전압 I: 전류
[옴의 법칙]	

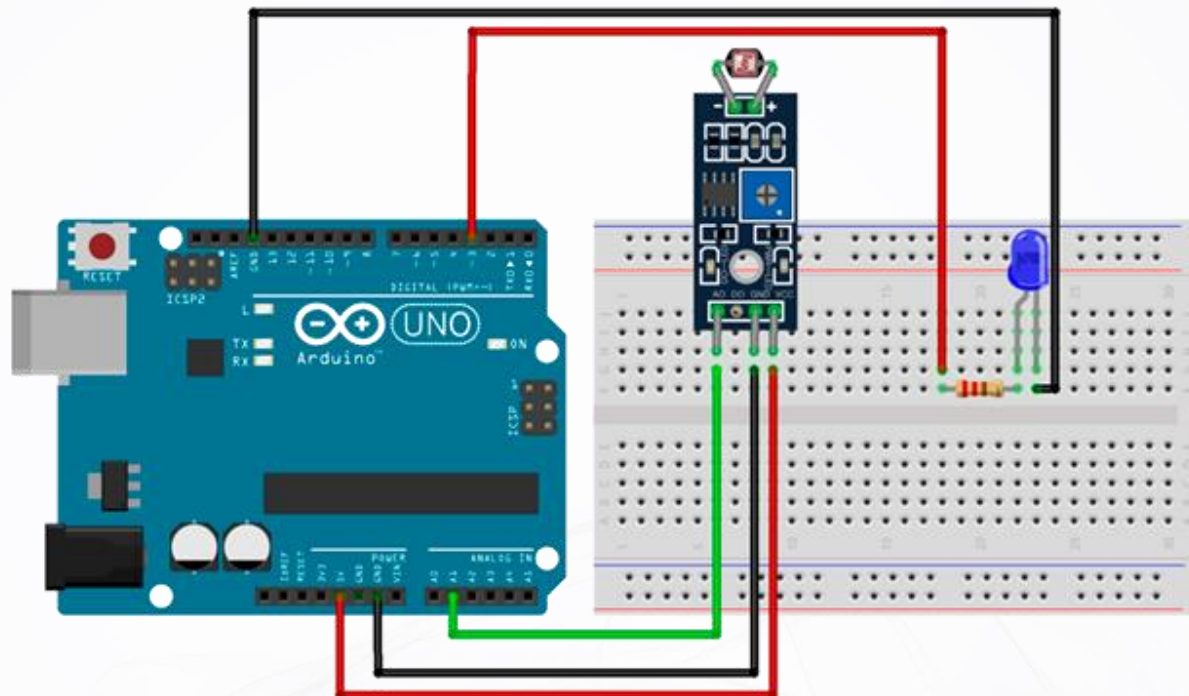
센서 내부의 CdS 광도전체가 빛의 양을 측정하여 빛의 양이 많아지면 전자를 운반하는 캐리어의 밀도가 높아지며 증가된 캐리어는 더 많은 전자를 운반하게 되어 전류가 증가합니다.

이때 생성된 전류는 전극을 타고 리드선으로 흐르게 됩니다.

옴의 법칙에 의해 빛이 강하면 조도 센서의 저항 값이 감소하고 빛이 약하면 증가합니다.

라즈베리파이 기반 실습 - 아두이노 연동

❖ 결선 방법



[결선회로도]

아두이노 우노보드	CdS 조도 센서 모듈
5V	VCC
GND	GND
A1	AO

[조도센서 모듈 결선표]

아두이노 우노보드	LED
D3	+ (긴 다리)
GND	- (짧은 다리)

[LED 결선표]

❖ 프로그램 코드

```
relay $
int Cds = 0;  // 조도 센서 데이터 수신 변수
int LED = 13;  // LED 연결한 핀

void setup() {
  Serial.begin(9600);
  pinMode(A1, INPUT); // 조도 센서를 입력 핀으로 설정
  pinMode(LED, OUTPUT); // LED를 출력 핀으로 설정
}

void loop() {
  Cds = analogRead(A1);  // 조도 센서의 측정 값을 Cds에 저장
  Serial.print("CDS_Sensor: ");
  Serial.println(Cds);    // 시리얼 모니터에 조도 센서의 측정 값 출력

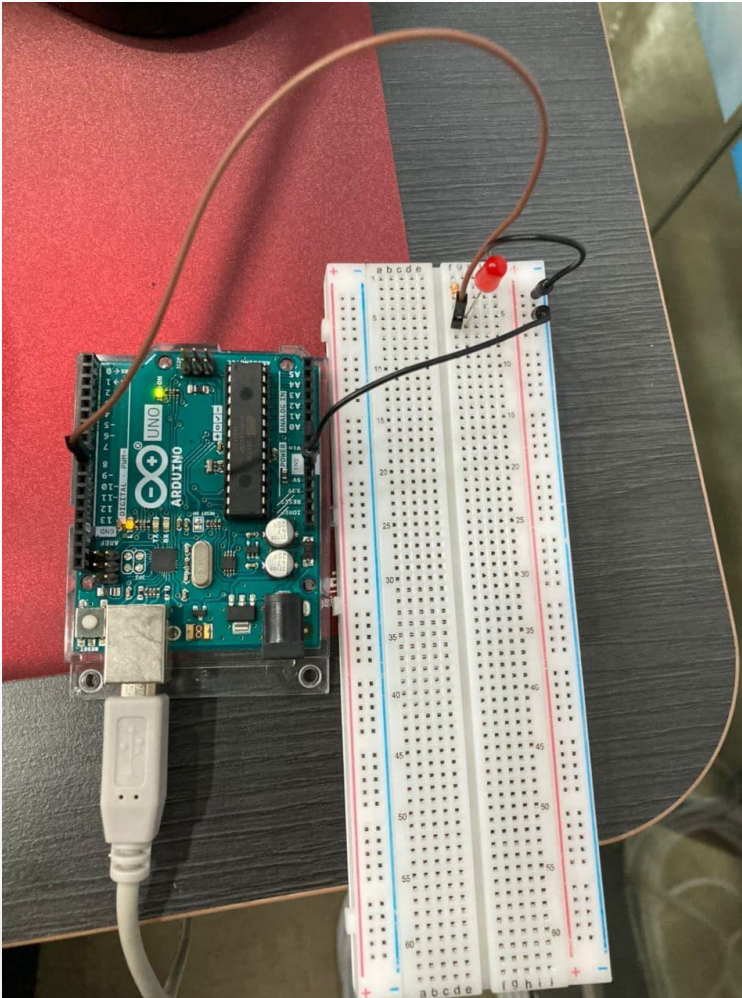
  if(Cds > 300) {          // 측정 값이 300 초과이면
    digitalWrite(LED, HIGH); // LED 켜기
    Serial.println("LED ON"); // 시리얼 모니터에 출력
  }

  else {                  // 측정 값이 300 이하이면
    digitalWrite(LED, LOW);  // LED 끄기
    Serial.println("LED OFF"); // 시리얼 모니터에 출력
  }

  delay(1000);            // 1초 쉬고 반복(1000ms = 1s)
}
```

라즈베리파이 기반 실습 - 아두이노 연동

❖ 아두이노 LED 제어



```
LED_simple | 아두이노 1.8.13
파일 편집 스케치 툴 도움말

LED_simple
int led = 8;
int val = 0;

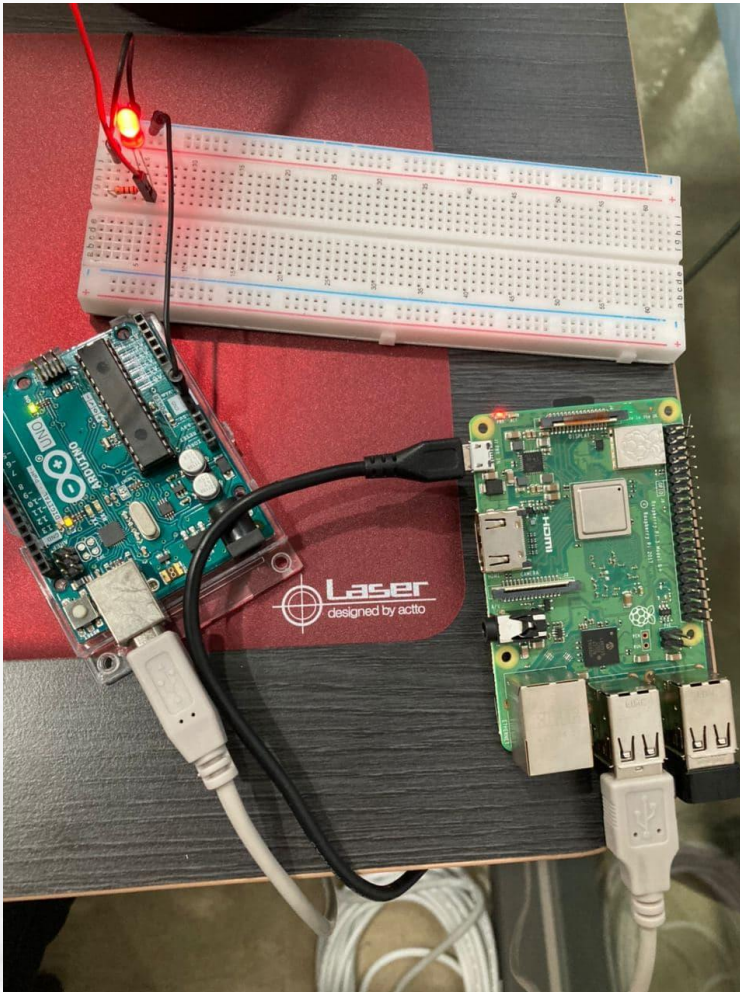
void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}

23 Arduino Uno on COM3
```


라즈베리파이 기반 실습 - 아두이노 연동

❖ 아두이노 LED 제어 - 1



```
LED_simple_1 | 아두이노 1.8.13
파일 편집 스케치 툴 도움말

LED_simple_1
int led = 9;
int val = 0;

void setup() {
  Serial.begin(9600);
  Serial.println("input 0~255");
  pinMode(led, OUTPUT);
}

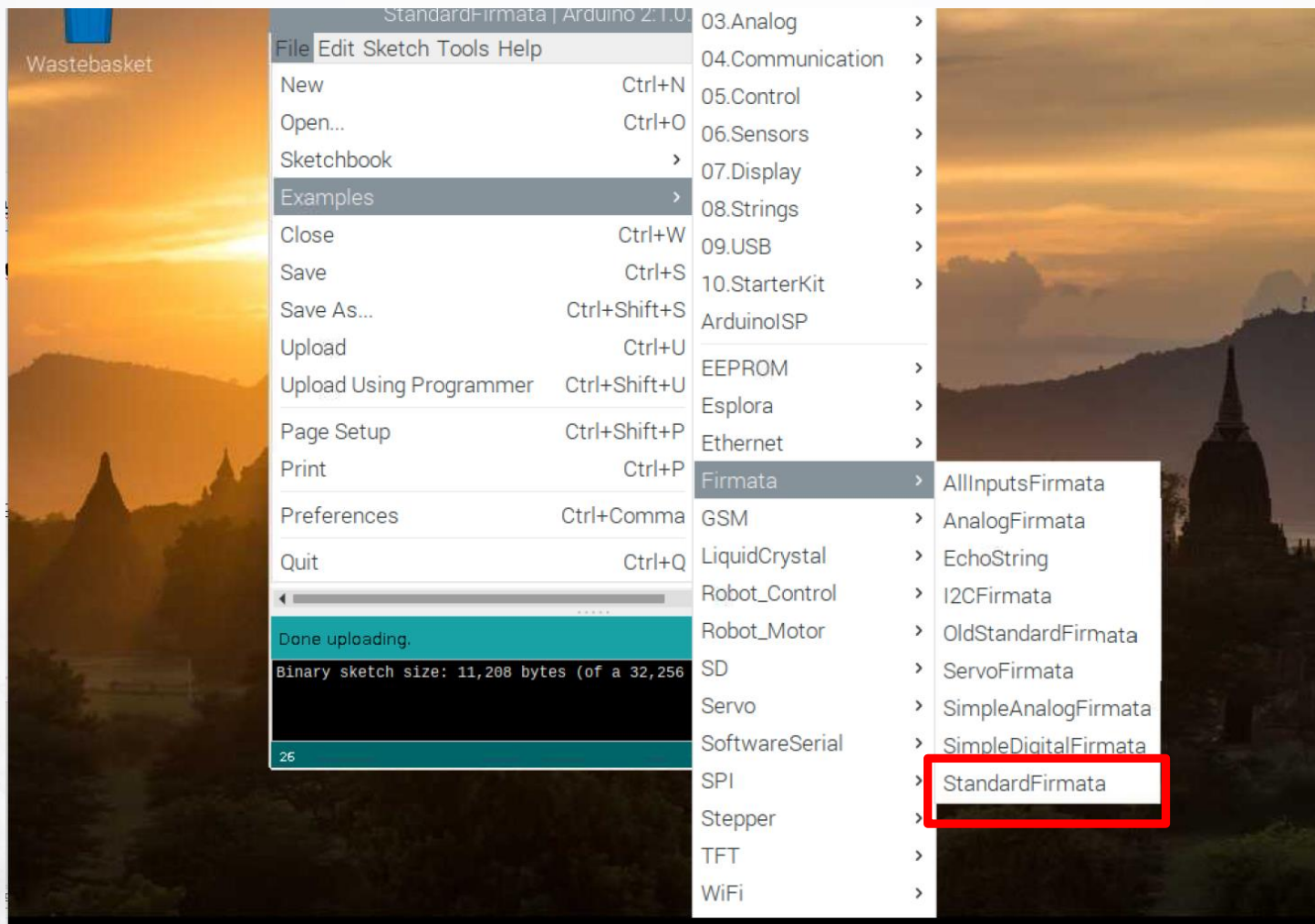
void loop() {
  if (Serial.available()) {
    val = Serial.parseInt();
    Serial.println(val);
    analogWrite(led, val);
  }
  delay(50);
}
```

저장 완료.

스케치는 프로그램 저장 공간 936 바이트(2%)를 사용. 최대 32256 바이트.
전역 변수는 동적 메모리 9바이트(0%)를 사용, 2039바이트의 지역변수가 남음. 최대는 2048 바이트.

25 Arduino Uno on COM3

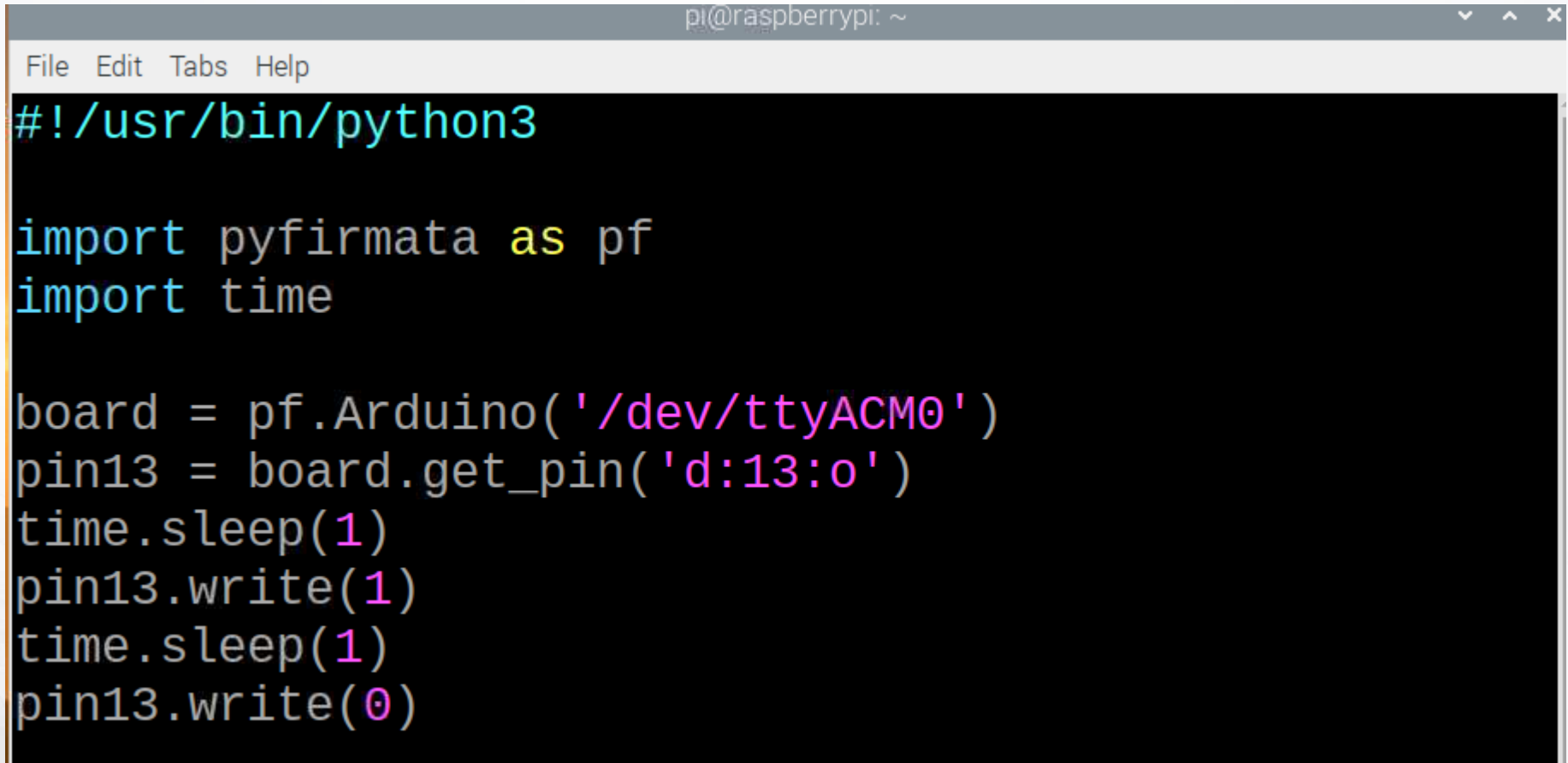
❖ 아두이노 LED 제어 - 1



라즈베리파이 기반 실습 - 아두이노 연동

❖ 라즈베리파이 python3 PyFirmate 라이브러리 설치

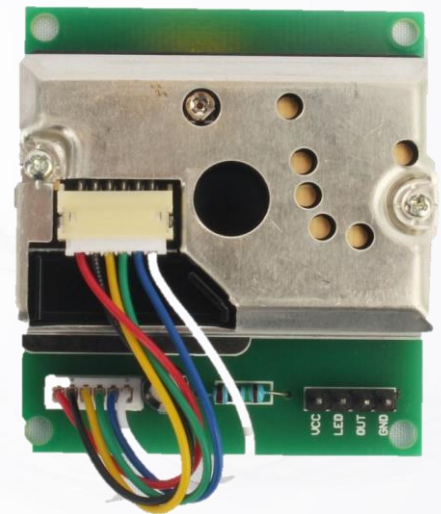
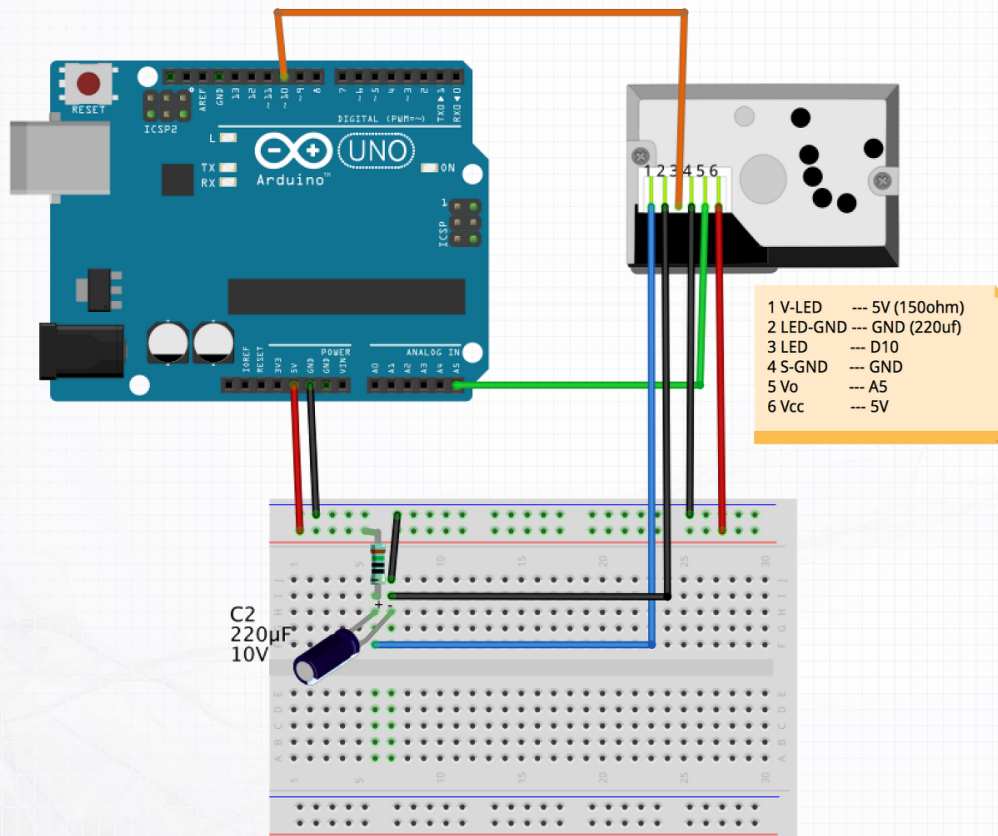
- pip3 install pyfirmata



```
pi@raspberrypi: ~  
File Edit Tabs Help  
#!/usr/bin/python3  
  
import pyfirmata as pf  
import time  
  
board = pf.Arduino('/dev/ttyACM0')  
pin13 = board.get_pin('d:13:o')  
time.sleep(1)  
pin13.write(1)  
time.sleep(1)  
pin13.write(0)
```

라즈베리파이 기반 실습 - 미세먼지센서 & 아두이노 연동

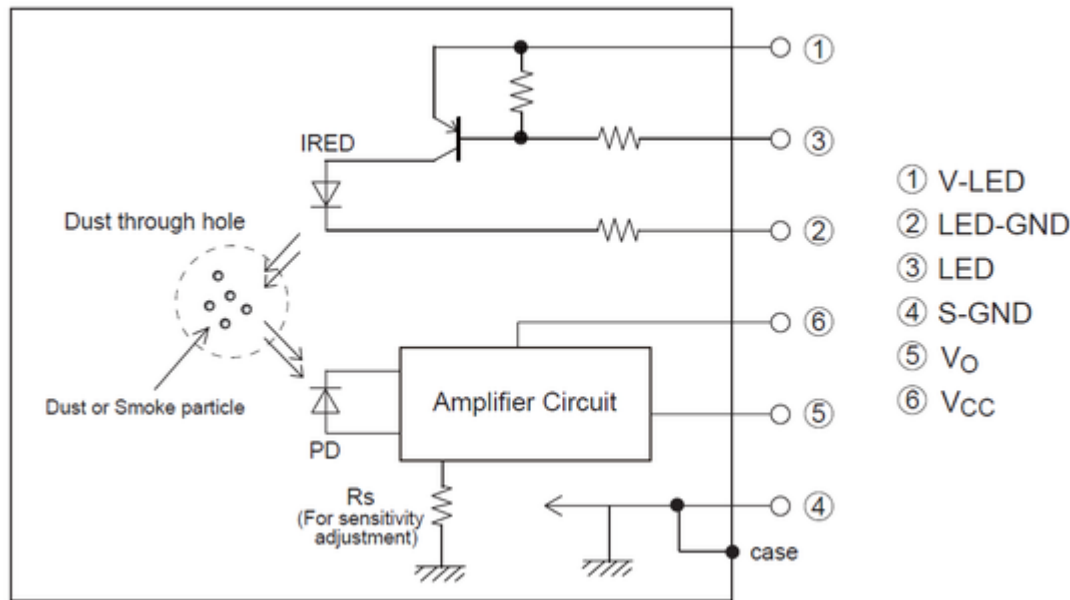
❖ 미세먼지센서(GP2Y1010AU0F)



라즈베리파이 기반 실습 - 미세먼지센서 && 아두이노 연동

❖ 미세먼지센서

- DataSheet (GP2Y1010AU0F)



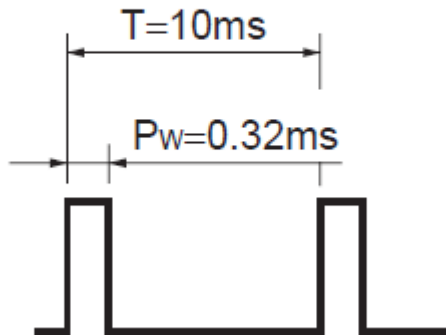
- 센서 중앙 홀을 통해 공기중에 먼지량을 측정
- 원형 구멍 양옆으로 두개의 소자가 부착됨(적외선LED, 적외선 수신소자),

라즈베리파이 기반 실습 - 미세먼지센서 && 아두이노 연동

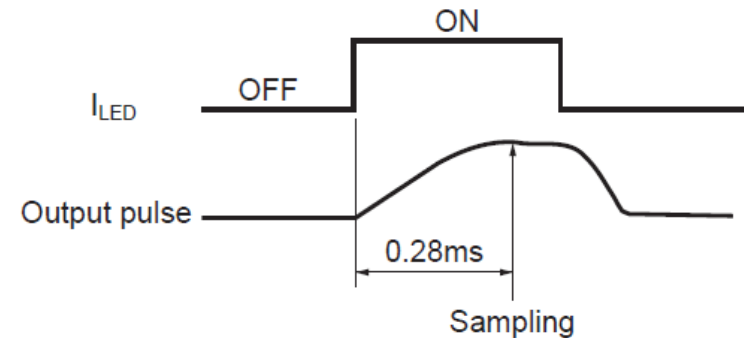
❖ 미세먼지센서 - 1

- DataSheet (GP2Y1010AU0F)

Pulse-driven wave form



- 적외선 LED 작동방법
- LED On/Off 총 10ms , (0.32ms LED ON, 9.68ms LED OFF) , 반복



- 데이터 수신
- 적외선 LED 켜 후 0.28ms 흐르고 적외선 수신기를 작동시켜 값을 Read)

라즈베리파이 기반 실습 - 미세먼지센서 && 아두이노 연동

❖ 미세먼지센서 - 2

- 아두이노 프로그램 코드

```
dust §  
  
int Vo = A0;  
int V_LED = 2;  
  
float Vo_value=0;  
  
void setup(){  
  Serial.begin(9600);  
  pinMode(V_LED, OUTPUT);  
  pinMode(Vo, INPUT);  
}  
  
void loop()  
{  
  digitalWrite(V_LED, LOW);  
  delayMicroseconds(280);  
  Vo_value = analogRead(Vo);  
  delayMicroseconds(40);  
  digitalWrite(V_LED, HIGH);  
  delayMicroseconds(9680);  
  
  Serial.println(Vo_value);  
  
  delay(1000);  
}
```

70.00
107.00
107.00
127.00
109.00
125.00
118.00
133.00
123.00
105.00
112.00

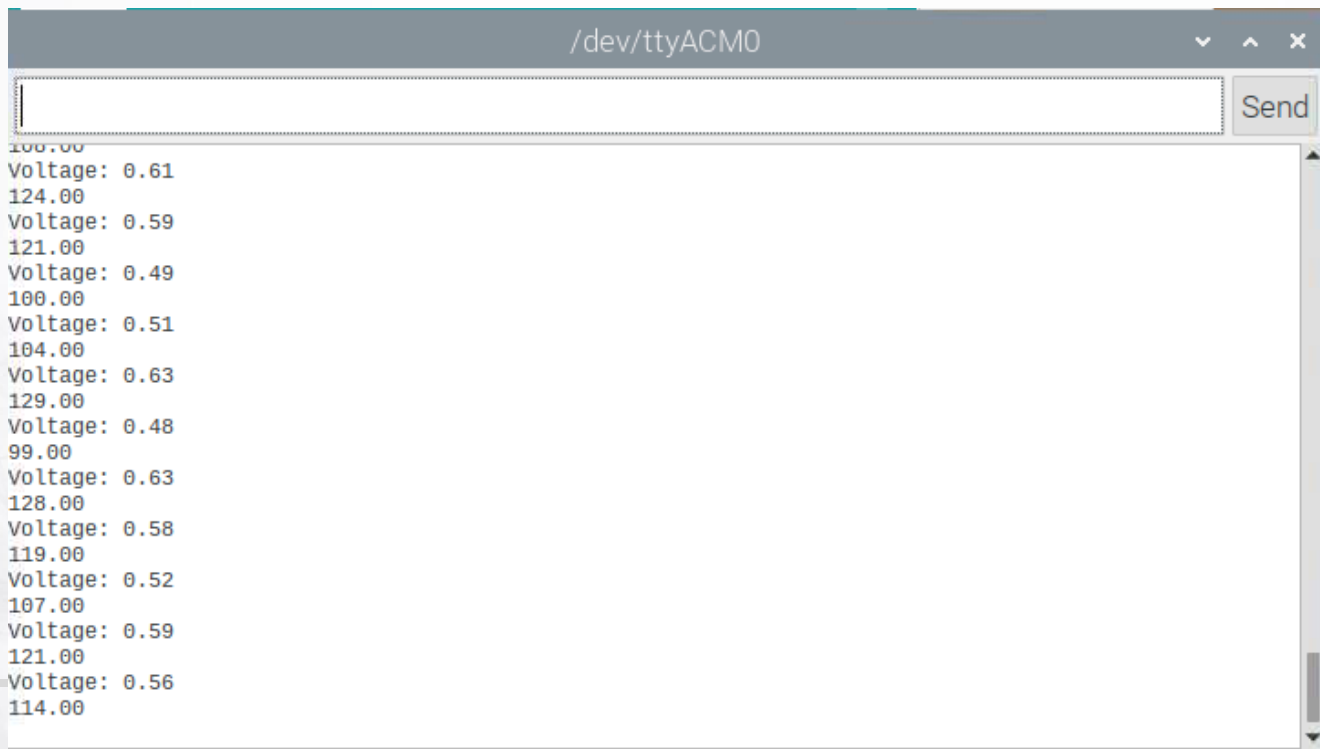
☒ Autoscroll

라즈베리파이 기반 실습 - 미세먼지센서 && 아두이노 연동

❖ 미세먼지센서 - 3

- 아날로그 데이터 : 전압을 0~1023로 표현
- 예) 5V 센서 사용시, 0~5V 값을 0 ~ 1023값으로 표현
- 전압 V : 아날로그 핀 값 x 5.0 / 1023.0 (원래전압)

❖ 출력 해보기



The screenshot shows a terminal window titled `/dev/ttyACM0`. The window contains a list of sensor readings. Each reading consists of a raw ADC value followed by a line indicating the corresponding voltage in Volts (V). The data is as follows:

Raw Value	Voltage (V)
100.00	0.61
124.00	0.59
121.00	0.49
100.00	0.51
104.00	0.63
129.00	0.48
99.00	0.63
128.00	0.58
119.00	0.52
107.00	0.59
121.00	0.56
114.00	

라즈베리파이 기반 실습 - 미세먼지센서 && 아두이노 연동

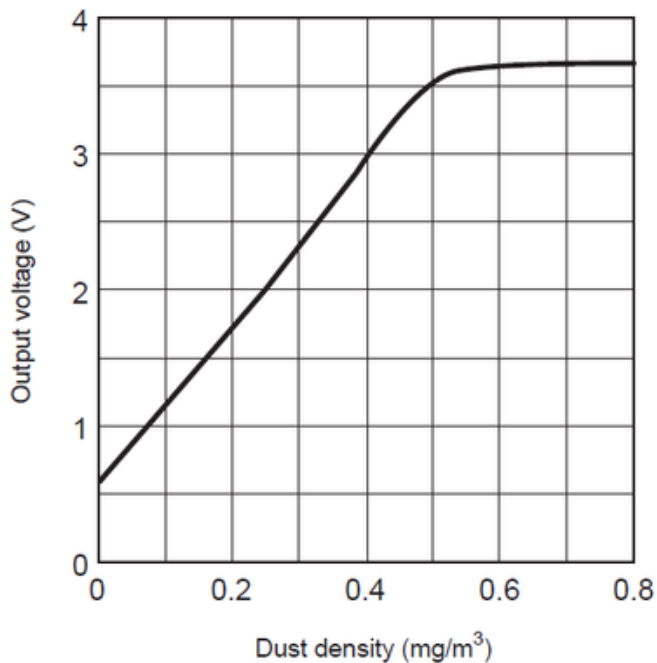
- ❖ 미세먼지센서 - 4
 - 전압 값을 이용,
미세먼지 양 측정

3-3 Electro-optical Characteristics

($T_a=25^{\circ}\text{C}$, $V_{CC}=5\text{V}$)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Sensitivity	K	(*1)(*2)(*3)(*4)	0.425	0.5	0.575	V/ ($100\mu\text{g}/\text{m}^3$)
Output voltage at no dust	Voc	(*2)(*3)(*4)	0.1	0.6	1.1	V
Output voltage range	VOH	$R_L=4.7\text{k}\Omega$ (*2)(*3)(*4)	3.4	-	-	V
LED terminal current	I _{LED}	LED terminal=0V (*2)(*3)	-	10	20	mA
Supply current	I _{CC}	$R_L=\infty$ (*2)(*3)	-	11	20	mA

Fig. 3 Output Voltage vs.



라즈베리파이 기반 실습 - 미세먼지센서 && 아두이노 연동

- ❖ 미세먼지센서 - 5
 - 아두이노 프로그램 코드

```
dust 5
int Vo = A0;
int V_LED = 2;

float Vo_value=0;
float Voltage = 0;
float dustDensity = 0;

void setup(){
  Serial.begin(9600);
  pinMode(V_LED, OUTPUT);
  pinMode(Vo, INPUT);
}

void loop()
{
  digitalWrite(V_LED, LOW);
  delayMicroseconds(280);
  Vo_value = analogRead(Vo);
  delayMicroseconds(40);
  digitalWrite(V_LED, HIGH);
  delayMicroseconds(9680);

  Voltage = Vo_value*5.0 / 1023.0;
  dustDensity = (Voltage - 0.3)/0.005;

  Serial.print("Voltage: ");
  Serial.println(Voltage);
  Serial.print("Dust Density: ");
  Serial.println(dustDensity);

  delay(1000);
}
```

```
Dust Density: 55.39
Voltage: 0.54
Dust Density: 47.53
Voltage: 0.60
Dust Density: 60.23
Voltage: 0.62
Dust Density: 63.17
Voltage: 0.57
Dust Density: 54.37
Voltage: 0.59
Dust Density: 57.30
Voltage: 0.62
Dust Density: 64.14
Voltage: 0.59
Dust Density: 57.30
Voltage: 0.44
Dust Density: 28.95
Voltage: 0.51
Dust Density: 42.64
Voltage: 0.57
Dust Density: 53.39
Voltage: 0.56
Dust Density: 52.41
```

☒ Autoscroll

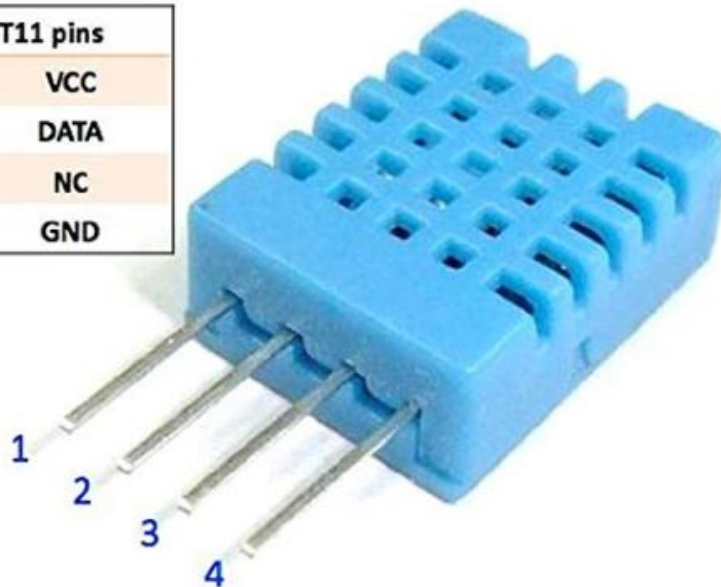
라즈베리파이 기반 실습 - DHT 11 && 아두이노 연동

❖ DHT11 아두이노 연동

#DHT11pin

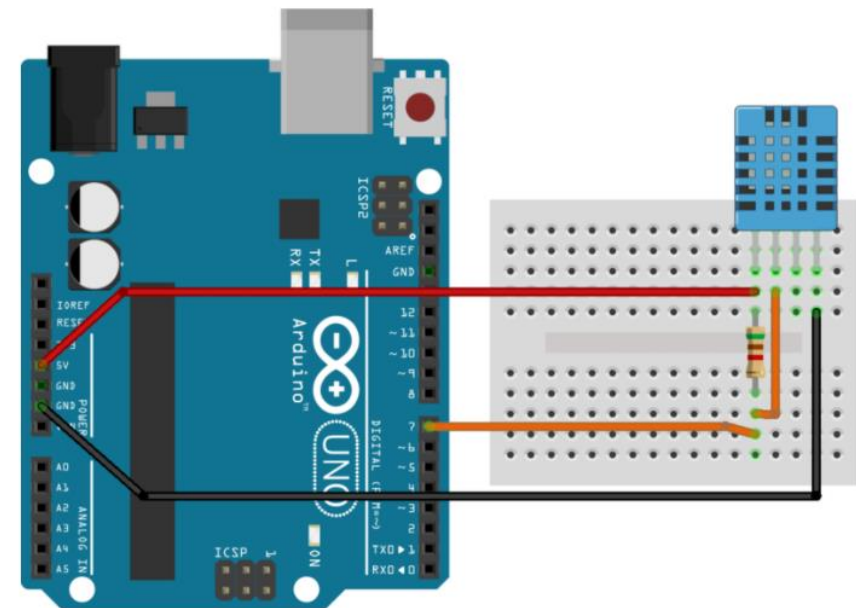
- 측정 습도 : 20 ~ 90 %
- 측정 온도 : 0 ~ 50℃

DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



#아두이노DHT11연결

아두이노	DHT11
5V	VCC
GND	GND
Digital 7번 핀	DATA(OUT)
저항 5.1K옴 권장	

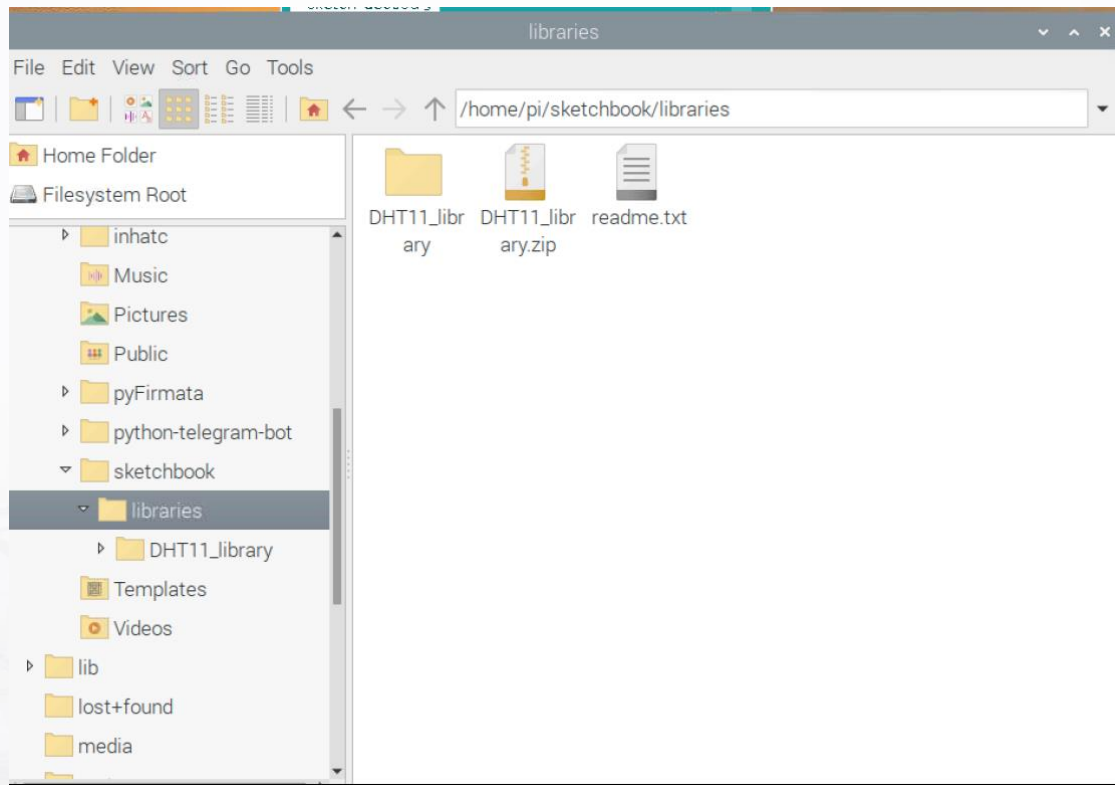


라즈베리파이 기반 실습 - DHT 11 && 아두이노 연동

❖ DHT11 아두이노 라이브러리 다운로드

✓ <https://github.com/sonnonet/inhatac>

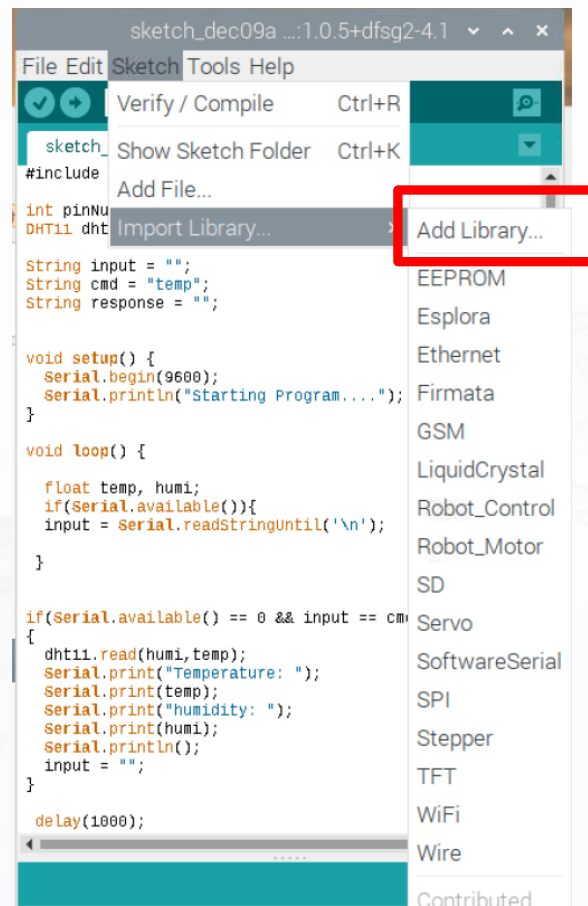
✓ 파일복사 DHT11_library.zip -> /home/pi/sketchbook/libraries



라즈베리파이 기반 실습 - DHT 11 && 아두이노 연동

❖ DHT11 라이브러리 추가

✓ Sketch -> import Library -> Add Library



❖ 아두이노 소스코드

- DHT11 라즈베리파이 연동 (Command)

```
#include <DHT11.h>

int pinNum = 7;
DHT11 dht11(pinNum);

String input = "";
String cmd = "temp";
String response = "";

void setup() {
  Serial.begin(9600);
  Serial.println("Starting Program...");
}

void loop() {

  float temp, humi;
  if(Serial.available()){
    input = Serial.readStringUntil('\n');

  }

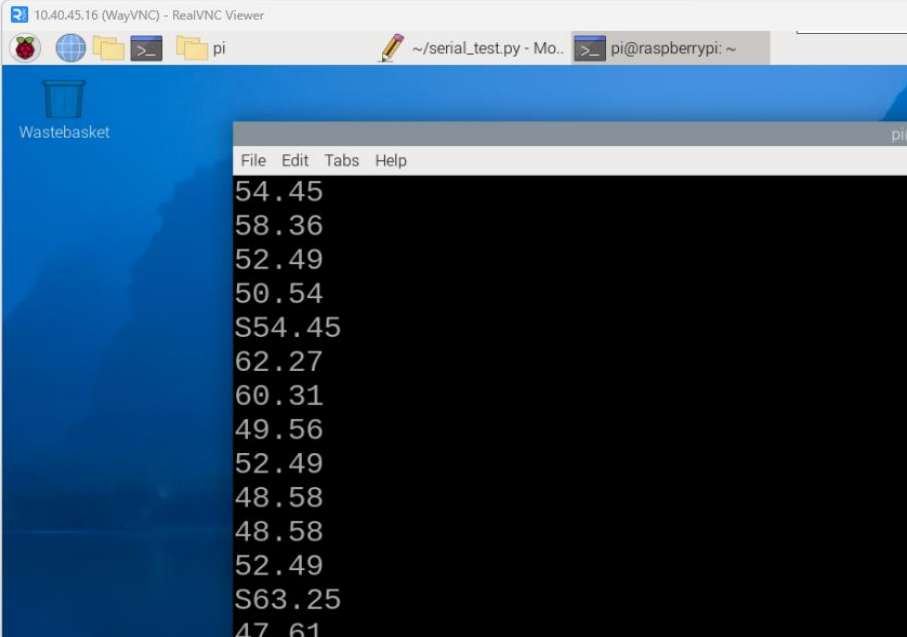
  if(Serial.available() == 0 && input == cmd)
  {
    dht11.read(humi,temp);
    Serial.print("Temperature: ");
    Serial.print(temp);
    Serial.print("humidity: ");
    Serial.print(humi);
    Serial.println();
    input = "";
  }

  delay(1000);
}
```

라즈베리파이 기반 실습 - 프로그래밍(Serial)

- ❖ Python3.11 소스코드 -> 라즈베리파이 Serial 통신
- ❖ vim serial_test.py

```
*untitled*
File Edit Format Run Options Window Help
1 import time
2 import serial
3
4 seri = serial.Serial('/dev/ttyACM0', baudrate = 9600, timeout = None)
5
6
7 while(True):
8     time.sleep(1)
9     if seri.in_waiting !=0:
10         content = seri.readline()
11         a = float(content.decode())
12         print(a)
13
```



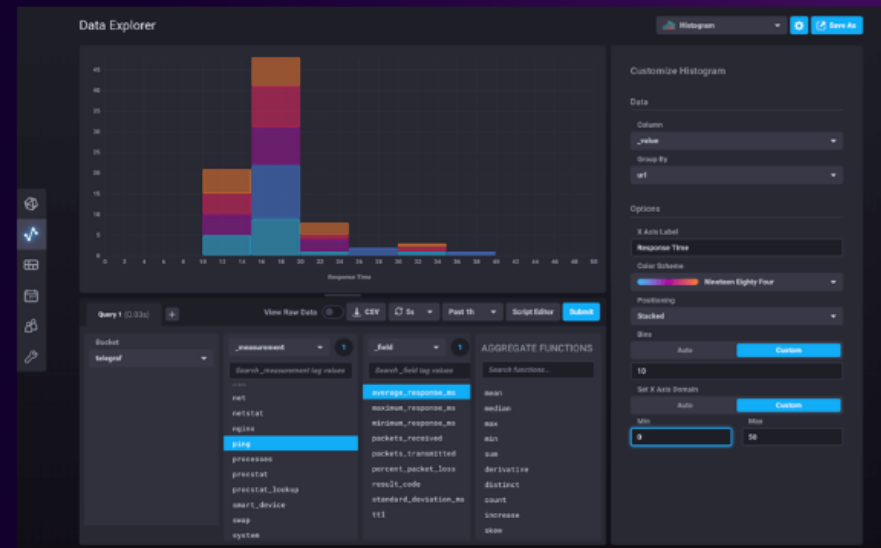
InfluxDB

<https://www.influxdata.com/>

InfluxDB

InfluxDB is a time series database designed to handle high write and query loads.

Get InfluxDB



InfluxDB 1.x

InfluxDB 1.x is the open source time series database component of the TICK Stack [Telegraf, InfluxDB, Chronograf, Kapacitor].

InfluxDB 2.0

Currently in beta, InfluxDB 2.0 incorporates everything you need in a time series platform into a single binary.

InfluxDB Cloud

InfluxDB Cloud is a fast, elastic, serverless time series platform as a service – easy to use with usage-based pricing

라즈베리파이 기반 실습 – InfluxDB 설치 및 설정

❖ 참고 사이트 주소 : https://github.com/sonnonet/2024_inhatc

🔗 InfluxDB 설치

- InfluxDB download key using wget

```
wget -q https://repos.influxdata.com/influxdata-archive_compat.key  
echo '393e8779c89ac8d958f81f942f9ad7fb82a25e133faddaf92e15b16e6ac9ce4c influxdata-archive_compat.key' |  
echo 'deb [signed-by=/etc/apt/trusted.gpg.d/influxdata-archive_compat.gpg] https://repos.influxdata.com'
```

- Packages are up to date && install Influxdb

```
sudo apt-get update && sudo apt-get install influxdb -y
```

- InfluxDB as a background service on startup

```
sudo service influxdb start
```

- InfluxDB is status (service)

```
sudo service influxdb status
```

InfluxDB 데이터베이스 만들기

```
$ influx
```

```
>create database <데이터베이스이름>
```

```
확인 : show databases
```

라즈베리파이 기반 실습 – InfluxDB 프로그래밍

- Python Serial to InfluxDB 프로그래밍
- vim serial_dust.py

```
1 import time
2 import requests, json
3 from influxdb import InfluxDBClient as influxdb
4 import serial
5
6 seri = serial.Serial('/dev/ttyACM0', baudrate = 9600, timeout = None)
7
8
9 while(True):
10     time.sleep(1)
11     if seri.in_waiting !=0:
12         content = seri.readline()
13         a = float(content.decode())
```

• Python Serial to InfluxDB 프로그래밍 - 1

```
14 data = [{
15     'measurement' : 'dust',
16     'tags':{
17         'InhaUni' : '2222',
18     },
19     'fields':{
20         'dust' : a,
21     }
22 }]
23 client = None
24 try:
25     client = influxdb('localhost',8086,'root','root','dust')
26 except Exception as e:
27     print ("Exception" + str(e))
28 if client is not None:
29     try:
30         client.write_points(data)
31     except Exception as e:
32         print("Exception write " + str(e))
33     finally:
34         client.close()
35 print("running influxdb OK")
```

라즈베리파이 기반 실습 - InfluxDB 설치 및 설정

- ❖ Python 프로그램 실행 시 influxdb 라이브러리 문제

```
pi@raspberrypi:~ $ python serial_dust.py
Traceback (most recent call last):
  File "/home/pi/serial_dust.py", line 3, in <module>
    from influxdb import InfluxDBClient as influxdb
ModuleNotFoundError: No module named 'influxdb'
```

```
pi@raspberrypi:~ $ pip install influxdb
error: externally-managed-environment
```

```
× This environment is externally managed
└> To install Python packages system-wide, try apt install
    python3-xyz, where xyz is the package you are trying to
    install.
```

If you wish to install a non-Debian-packaged Python package, create a virtual environment using `python3 -m venv path/to/venv`. Then use `path/to/venv/bin/python` and `path/to/venv/bin/pip`. Make sure you have `python3-full` installed.

For more information visit <http://rptl.io/venv>

note: If you believe this is a mistake, please contact your Python installation or OS distribution provider. You can override this, at the risk of breaking your Python installation or OS, by passing `--break-system-packages`.

hint: See PEP 668 for the detailed specification.

라즈베리파이 기반 실습 – InfluxDB 설치 및 설정

- ❖ Python 프로그램 실행 시 influxdb 라이브러리 문제 해결방법

```
pi@raspberrypi:~$ sudo rm /usr/lib/python3.11/EXTERNALLY-MANAGED
pi@raspberrypi:~$ pip install influxdb
Defaulting to user installation because normal site-packages is not writeable
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting influxdb
  Downloading https://www.piwheels.org/simple/influxdb/influxdb-5.3.2-py2.py3-no
ne-any.whl (79 kB)
    _____ 79.4/79.4 kB 247.9 kB/s eta 0:00:00
Collecting python-dateutil>=2.6.0
  Downloading https://www.piwheels.org/simple/python-dateutil/python_dateutil-2.
9.0.post0-py2.py3-none-any.whl (229 kB)
    _____ 229.9/229.9 kB 113.4 kB/s eta 0:00:00
Requirement already satisfied: pytz in /usr/lib/python3/dist-packages (from infl
uxdb) (2022.7.1)
Requirement already satisfied: requests>=2.17.0 in /usr/lib/python3/dist-package
s (from influxdb) (2.28.1)
```

라즈베리파이 기반 실습 – Grafana 설치 및 설정

❖ 참고 사이트 주소 : https://github.com/sonnonet/2024_inhatc

Grafana Installation

1. Install the prerequisite packages

```
sudo apt-get install -y apt-transport-https software-properties-common wget
```

2. Import the GPG key:

```
sudo mkdir -p /etc/apt/keyrings/  
wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee /etc/apt/keyrings/grafana.gpg >
```

3. To add a repository for stable releases, run the following command:

```
echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com stable main" | sudo tee -a
```

4. Run the following command to update the list of available packages:

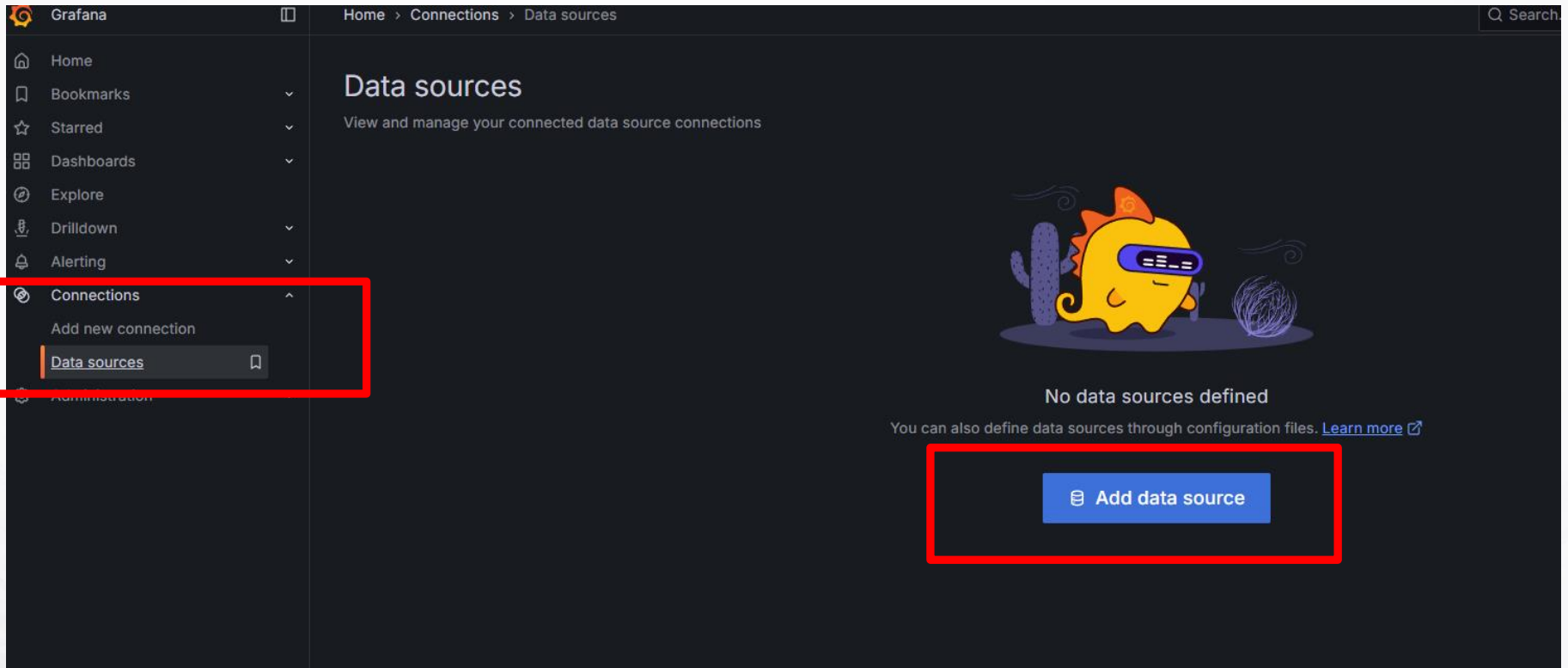
```
sudo apt-get update && sudo apt-get install grafana -y
```

5. Run the following command to server start

```
sudo systemctl start grafana-server
```

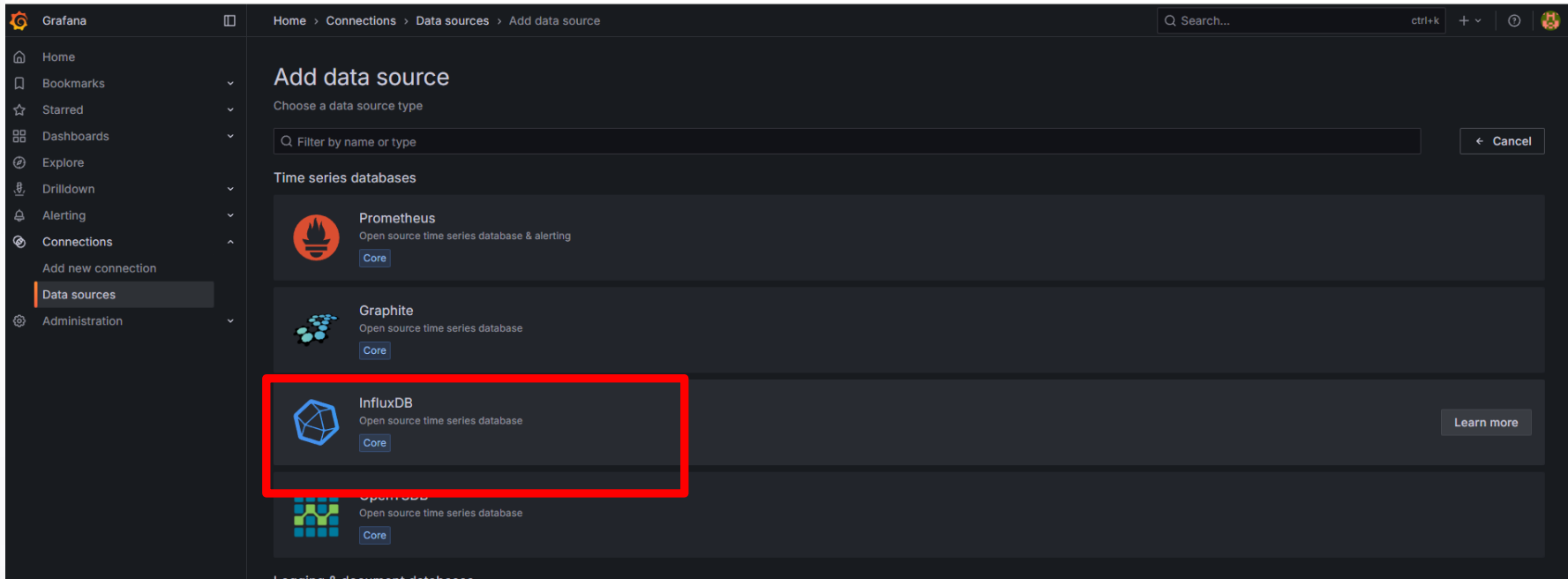
라즈베리파이 기반 실습 – Grafana 설치 및 설정

- ❖ Grafana 초기 로그인 (ID : admin, PW : admin) 변경 하지 말 것.
- ❖ Data sources (데이터 연결)



라즈베리파이 기반 실습 – Grafana 설치 및 설정

❖ Data sources (데이터 연결) -> influxdb



라즈베리파이 기반 실습 – Grafana 설치 및 설정

Type: InfluxDB

Settings

Name ⓘ influxdb Default ☒

Query language

InfluxQL

Please report any issues to:
<https://github.com/grafana/grafana/issues>

HTTP

URL

Your access method is **Server**, this means the URL needs to be accessible from the grafana backend/server.

http://localhost:8086

Invalid URL

Allowed cookies

Grafana proxy deletes forwarded cookies by default. Specify cookies by name that should be forwarded to the data source.

New tag (enter key to add) Add

Timeout

HTTP request timeout in seconds

Timeout in seconds

< - 별칭

< - 가져올 influxdb

ip주소 및 포트

라즈베리파이 기반 실습 – Grafana 설치 및 설정

The screenshot shows the Grafana configuration page for a new data source. A red box highlights the configuration fields, and another red box highlights the success message at the bottom.

Database	dust
User	root
Password	configured
HTTP Method	Choose
Min time interval	10s
Autocomplete range	12h
Max series	1000

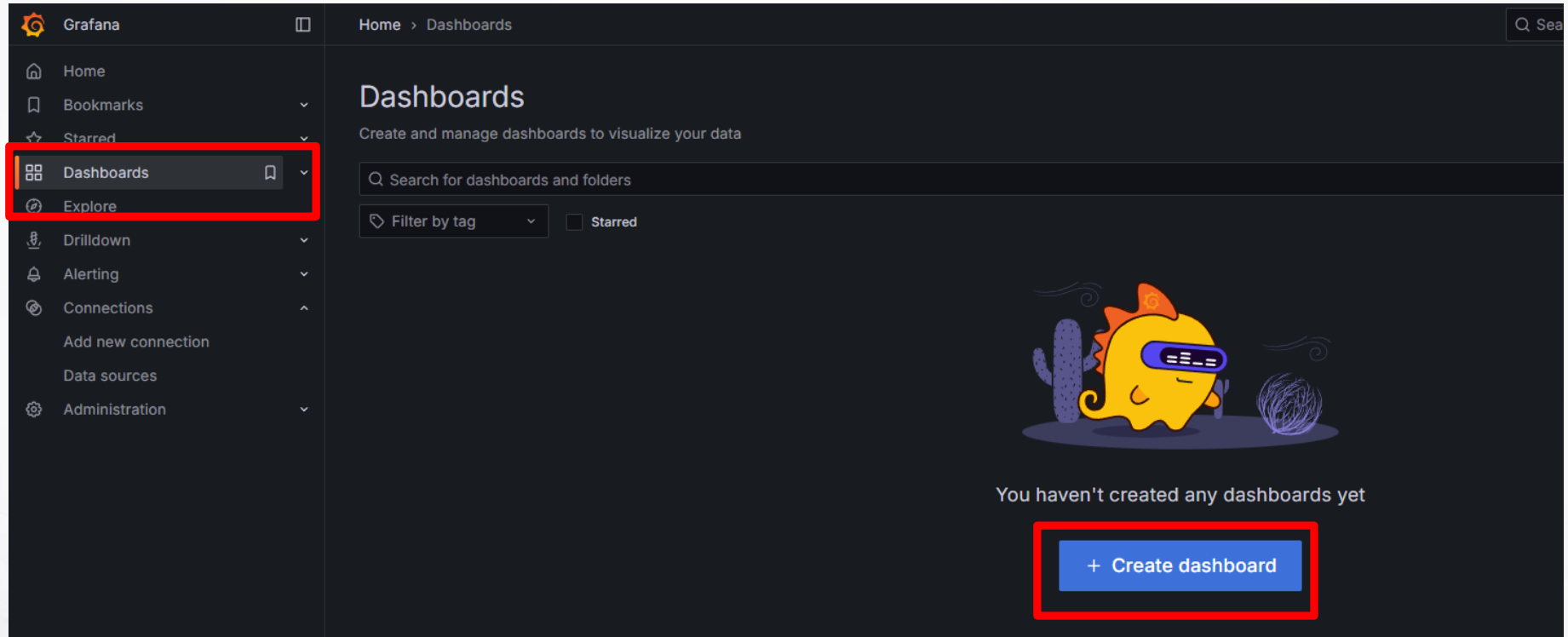
Reset

✓ datasource is working. 1 measurements found

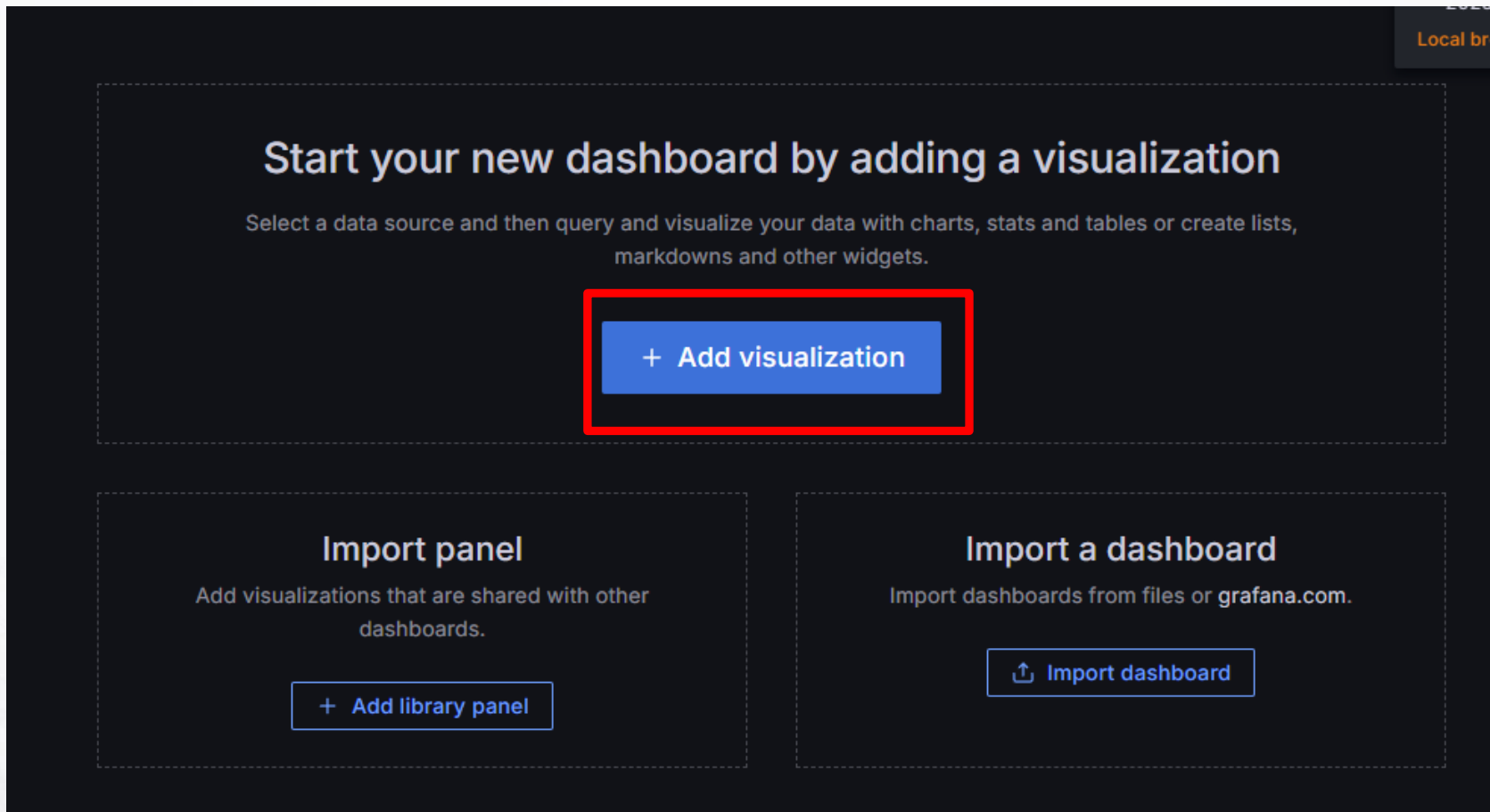
Next, you can start to visualize data by [building a dashboard](#), or by querying data in the [Explore view](#).

라즈베리파이 기반 실습 – Grafana 설치 및 설정

❖ Dashboards (데이터 표현) -> Create dashboard

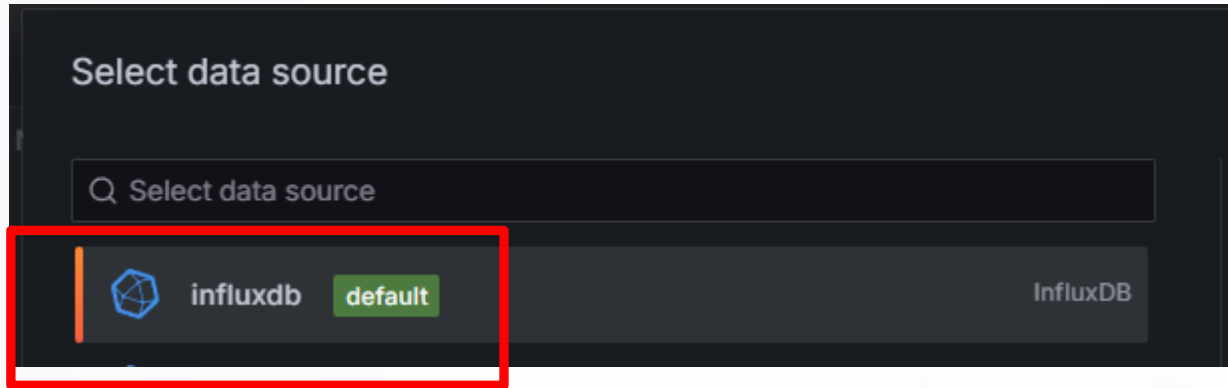


❖ Dashboards (데이터 표현) -> Add visualization



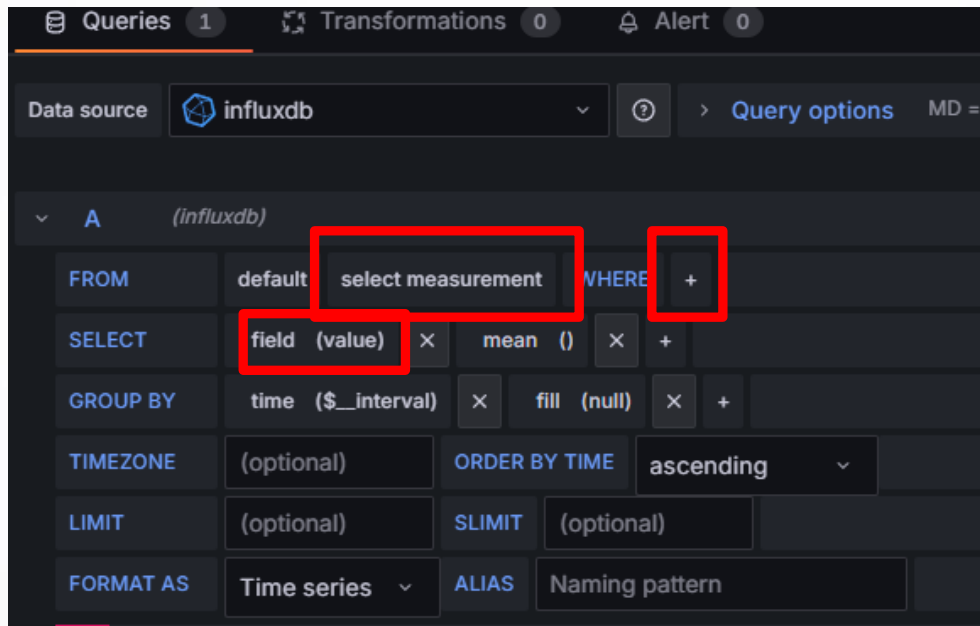
라즈베리파이 기반 실습 – Grafana 설치 및 설정

- ❖ Select data source (Data sources 설정한 별칭 선택)



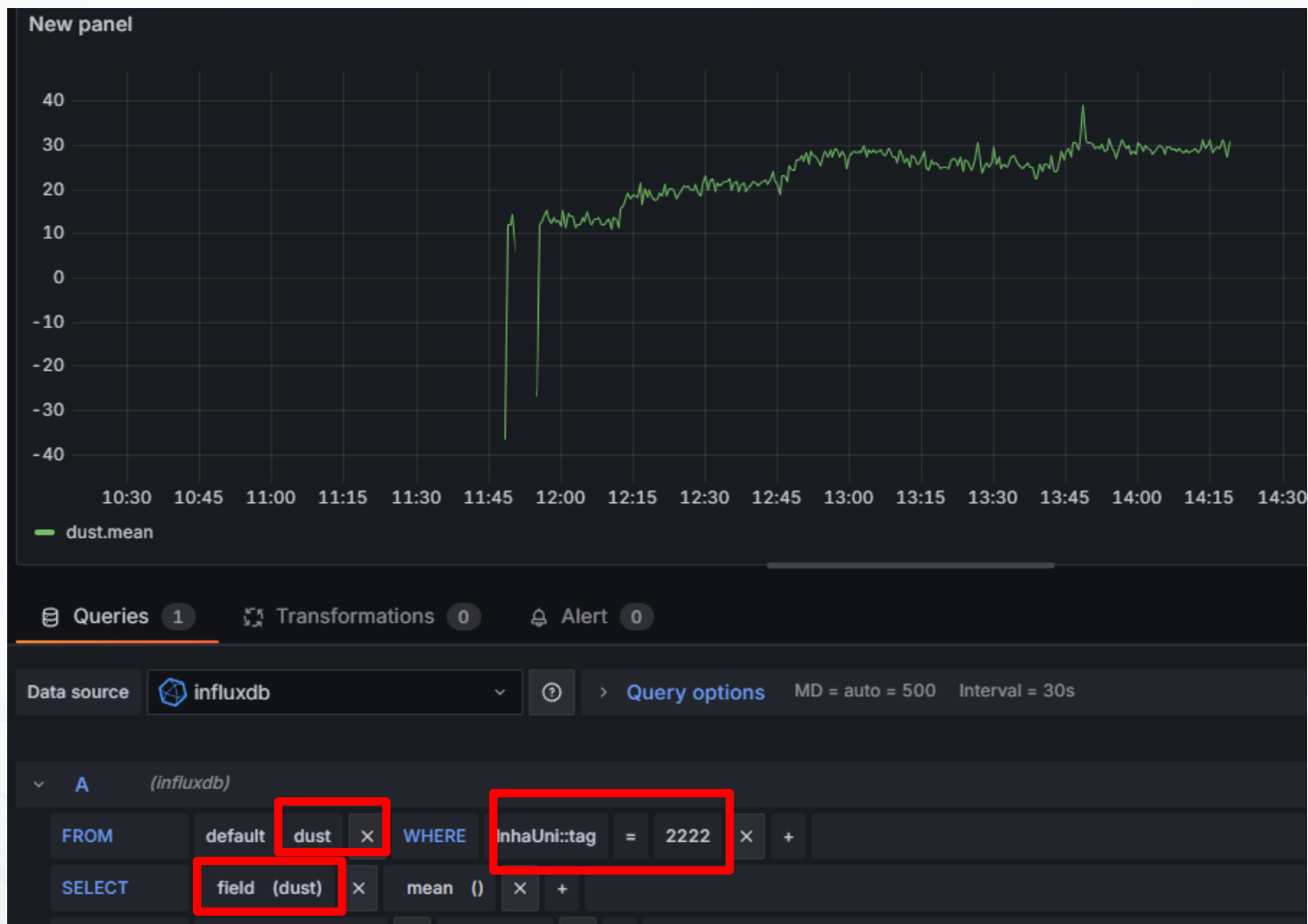
라즈베리파이 기반 실습 – Grafana 설치 및 설정

❖ Queries 데이터 조회



라즈베리파이 기반 실습 – Grafana 설치 및 설정

❖ Queries 데이터 조회 - 1



3주차 수업이 끝났습니다

고생하셨습니다.

