인하공업전문대학
INHA TECHNICAL COLLEGE

사물인터넷
3주차

인하공업전문대학 컴퓨터 정보과

김한결

사물인터넷 - 시계열데이터베이스

❖ InfluxDB란?

- 시계열(Time-Series) 데이터베이스(TSDB, Time-Series Database), 시간에 따라 변화하는 데이터를 효과적으로 저장하고 관리하는 데 최적화된 데이터베이스

☐ include secondary database models — 44 systems in ranking, March 2025

Rank			DBMS	Database Model	Score		
Mar 2025	Feb 2025	Mar 2024			Mar 2025	Feb 2025	Mar 2024
1.	1.	1.	InfluxDB	Time Series, Multi-model	21.50	-0.27	-5.39
2.	2.	↑ 3.	Kdb	Multi-model	7.10	+0.31	-0.59
3.	3.	↓ 2.	Prometheus	Time Series	6.38	-0.31	-1.62
4.	4.	↑ 5.	Graphite	Time Series	4.57	+0.20	-0.27
5.	5.	↓ 4.	TimescaleDB	Time Series, Multi-model	3.48	-0.14	-1.84
6.	6.	↑ 8.	QuestDB	Time Series, Multi-model	3.10	+0.00	+0.41
7.	7.	7.	Apache Druid	Multi-model	2.79	-0.07	-0.54
8.	8.	↓ 6.	DolphinDB	Multi-model	2.29	-0.09	-1.85
9.	9.	↑ 11.	GridDB	Time Series, Multi-model	1.98	+0.04	-0.02
10.	10.	↓ 9.	TDengine	Time Series, Multi-model	1.76	-0.06	-0.88
11.	11.	↓ 10.	RRDtool	Time Series	1.54	-0.05	-0.93
12.	12.	↑ 13.	Fauna	Multi-model	1.51	+0.02	-0.12
13.	↑ 14.	↑ 15.	Apache IoTDB	Time Series	1.44	+0.02	+0.25

출처 – https://db-engines.com/en/ranking/time+series+dbms

사물인터넷 - 시계열데이터베이스

❖ InfluxDB 주요 특징

1 시계열 데이터 저장

- 센서 데이터, 주식 가격, 서버 로그, 네트워크 모니터링 데이터 등 **시간과 함께 변화하는 데이터**를 저장하는 데 특화됨.
- `timestamp`(시간 정보)가 자동으로 추가됨.

2 고속 데이터 삽입 및 검색

- `INSERT` 속도가 빠르며, `SELECT` 시 시간 범위를 지정하여 빠르게 데이터를 검색할 수 있음.
- `"5분 동안의 평균 온도"` 같은 집계 연산 가능.

3 SQL과 비슷한 질의(Query) 언어 지원

- `InfluxQL`, `Flux` 같은 언어를 제공하여 **SQL과 유사한 방식**으로 데이터를 조회할 수 있음.
- 예제:

```sql
SELECT temperature FROM sensor_data WHERE time > now() - 1h;
```

→ 지난 1시간 동안의 온도 데이터 조회

4 태그(Tag) 기반 데이터 저장

- 센서나 기기의 데이터를 저장할 때 **태그(Tag)** 를 사용하면 검색 속도가 빨라짐.
- 예제 데이터 구조:

```css
measurement: "sensor_data"
tags: { "device": "arduino", "location": "room1" }
fields: { "temperature": 25.3 }
timestamp: 2025-03-17T12:00:00Z
```

사물인터넷 - 시계열데이터베이스

❖ InfluxDB 주요 개념

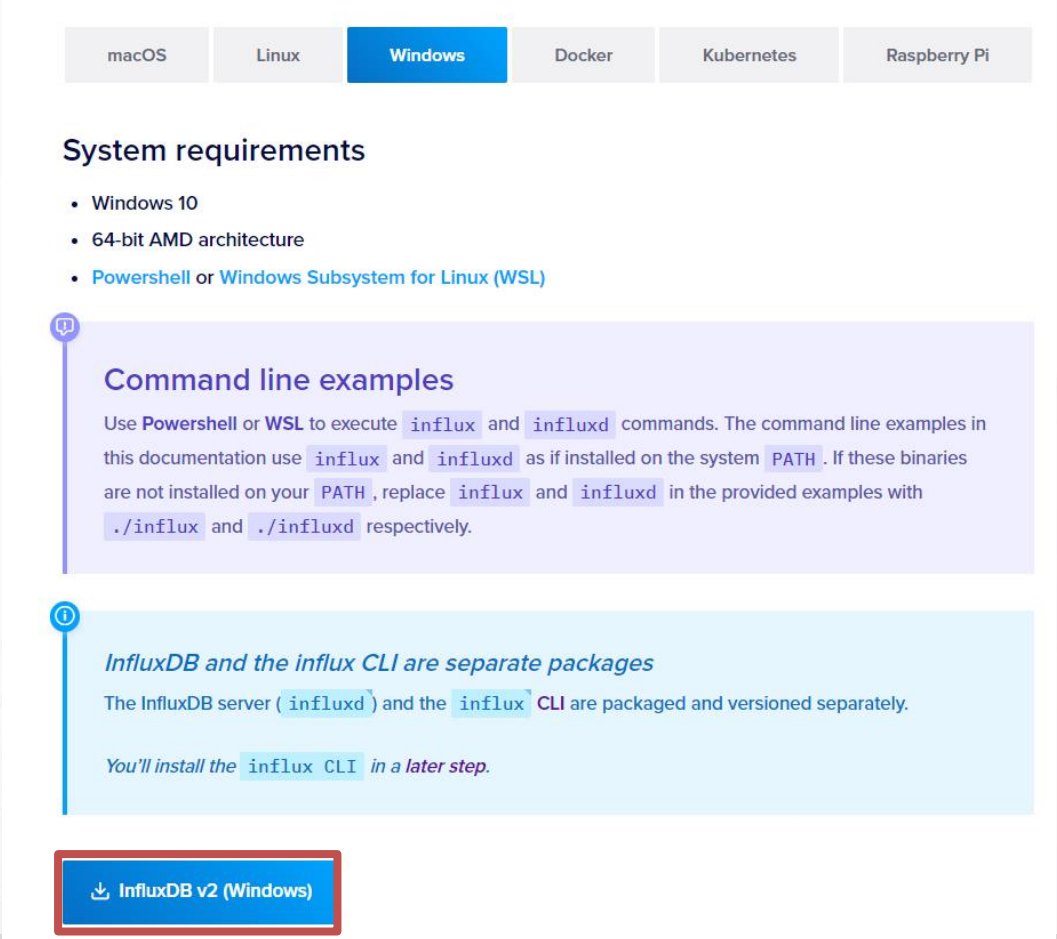
개념	설명
Measurement	테이블과 유사한 개념 (예: `"sensor_data"`)
Tag	빠른 검색을 위한 메타데이터 (예: `"device=arduino"`)
Field	저장할 데이터 값 (예: `"temperature=25.3"`)
Timestamp	데이터가 기록된 시간

🚀 InfluxDB vs 다른 데이터베이스 비교

특징	InfluxDB	MySQL	MongoDB
주요 목적	시계열 데이터	일반 데이터	문서 기반 데이터
속도	**빠름** (최적화됨)	보통	보통
저장 구조	시간 기반 (Timestamp 필수)	테이블 기반	JSON 문서 기반
적합한 용도	센서 데이터, 로그 분석	일반 데이터 저장	NoSQL 문서 저장

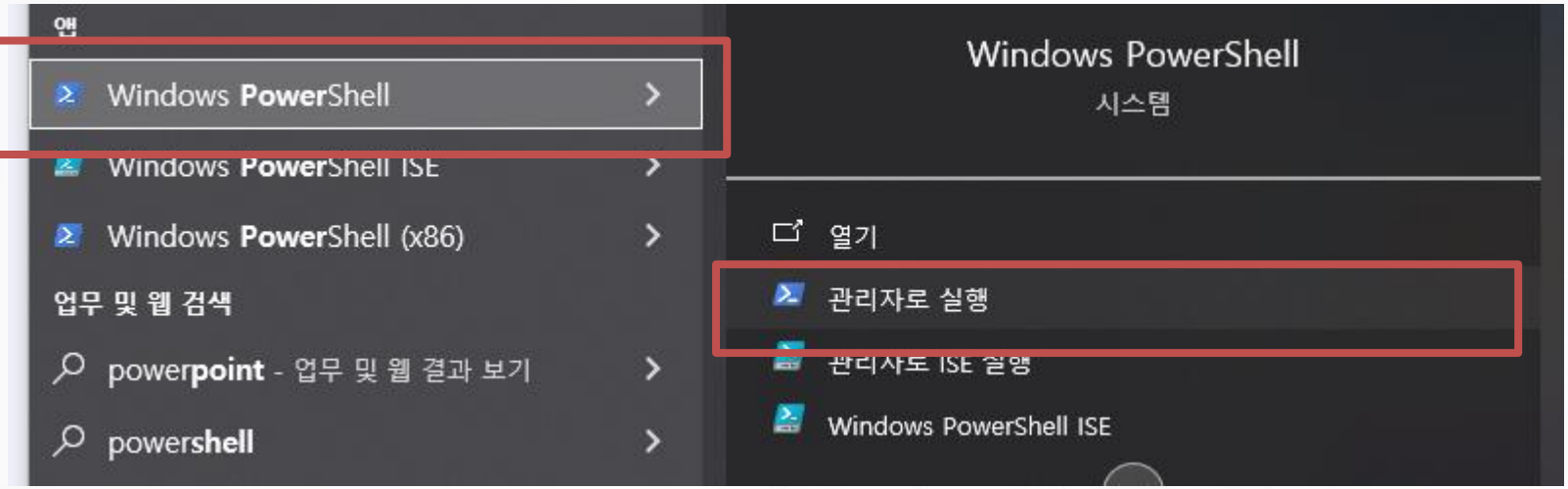
❖ InfluxDB 설치 (windows)

https://docs.influxdata.com/influxdb/v2/install/?t=Windows



사물인터넷 - 시계열데이터베이스

❖ **Windows PowerShell -> 관리자로 실행**



- 명령어 해석

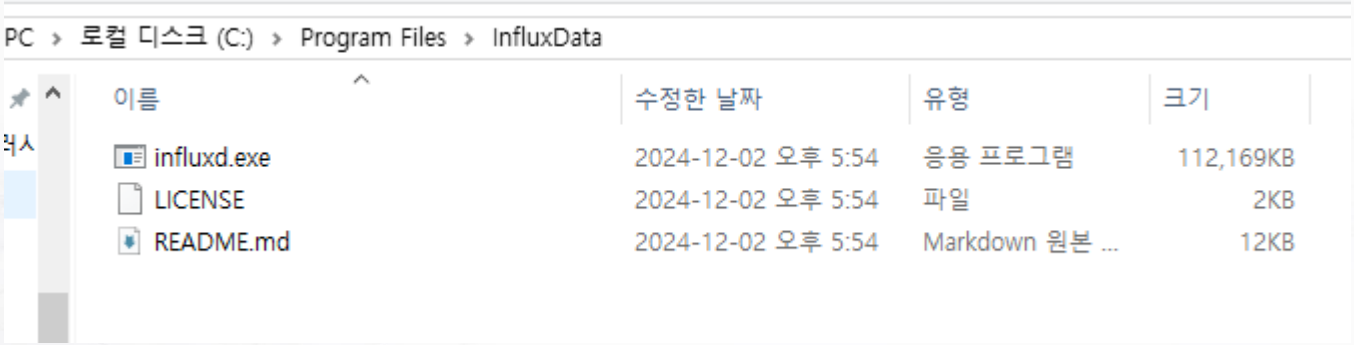
```
Expand-Archive .\influxdb2-2.7.11-windows.zip -DestinationPath 'C:\Program Files\InfluxData\'
mv 'C:\Program Files\InfluxData\influxdb2-2.7.11' 'C:\Program Files\InfluxData\influxdb'
```

사물인터넷 – 시계열데이터베이스(InfluxDB)

❖ Start InfluxDB

```
cd -Path 'C:\Program Files\InfluxData\influxdb'
./influxd
```

- 파일 경로 주의



사물인터넷 – 시계열데이터베이스(InfluxDB)

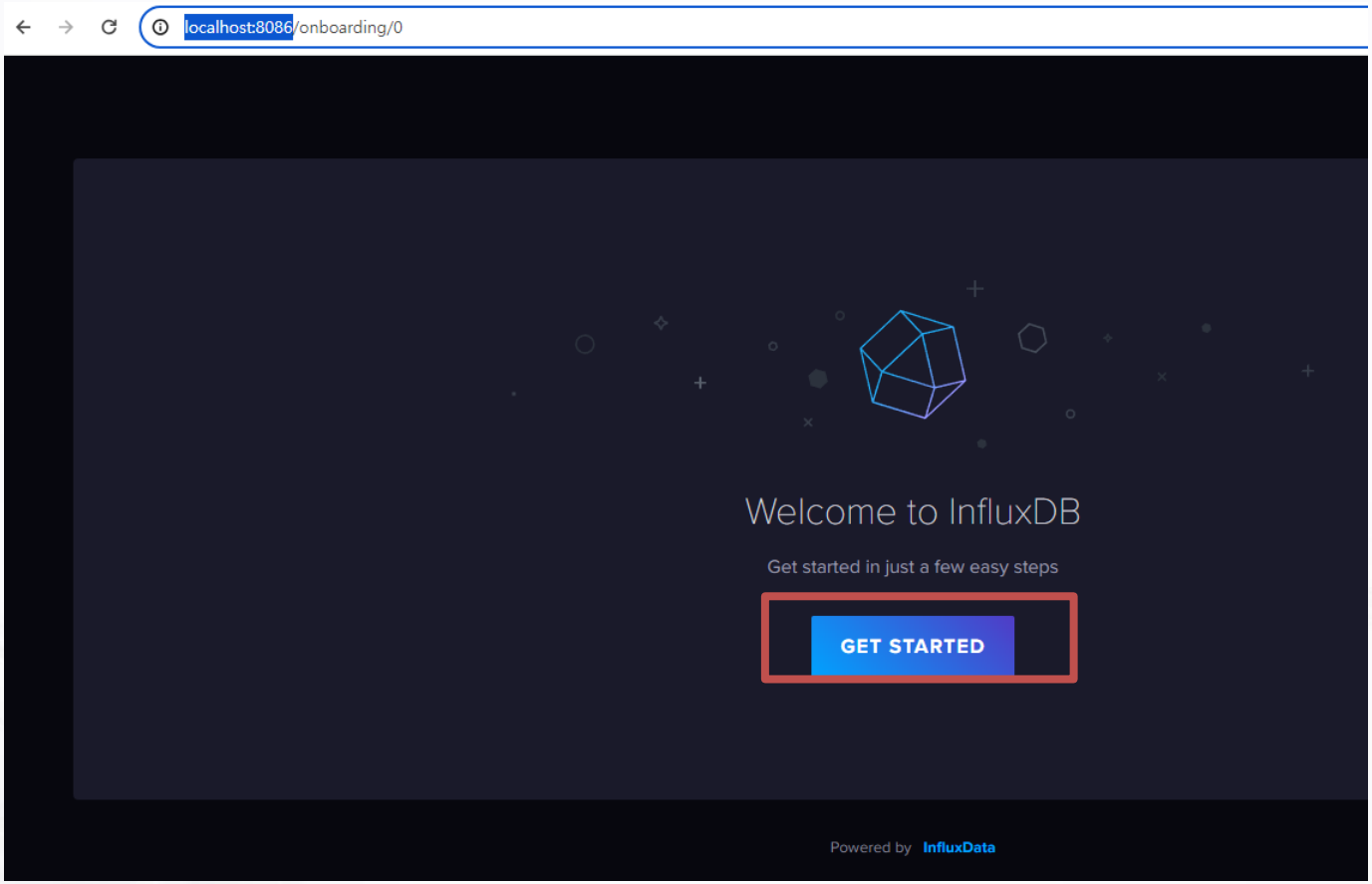
❖ 실행 화면

```
PS C:₩Users₩SUPER₩Downloads> cd -Path 'C:₩Program Files₩InfluxData₩'
PS C:₩Program Files₩InfluxData> ./influxd
2025-03-17T10:17:37.175226Z     info    Welcome to InfluxDB     {"log_id": "0vKua6a1000", "version": "v2.7.11", "commit
: "fbf5d4ab5e", "build_date": "2024-12-02T17:48:13Z", "log_level": "info"}
2025-03-17T10:17:37.181228Z     info    Resources opened        {"log_id": "0vKua6a1000", "service": "bolt", "path": "C
₩₩Users₩₩SUPER₩₩.influxdbv2₩₩influxd.bolt"}
2025-03-17T10:17:37.182216Z     info    Resources opened        {"log_id": "0vKua6a1000", "service": "sqlite", "path":
C:₩₩Users₩₩SUPER₩₩.influxdbv2₩₩influxd.sqlite"}
2025-03-17T10:17:37.184222Z     info    Bringing up metadata migrations {"log_id": "0vKua6a1000", "service": "KV migrat
ons", "migration_count": 20}
2025-03-17T10:17:37.259634Z     info    Bringing up metadata migrations {"log_id": "0vKua6a1000", "service": "SQL migra
ions", "migration_count": 8}
2025-03-17T10:17:37.302144Z     info    Using data dir  {"log_id": "0vKua6a1000", "service": "storage-engine", "service
: "store", "path": "C:₩₩Users₩₩SUPER₩₩.influxdbv2₩₩engine₩₩data"}
2025-03-17T10:17:37.303176Z     info    Compaction settings     {"log_id": "0vKua6a1000", "service": "storage-engine",
service": "store", "max_concurrent_compactions": 8, "throughput_bytes_per_second": 50331648, "throughput_bytes_per_seco
d_burst": 50331648}
2025-03-17T10:17:37.304210Z     info    Open store (start)      {"log_id": "0vKua6a1000", "service": "storage-engine",
service": "store", "op_name": "tsdb_open", "op_event": "start"}
2025-03-17T10:17:37.305834Z     info    Open store (end)        {"log_id": "0vKua6a1000", "service": "storage-engine",
service": "store", "op_name": "tsdb_open", "op_event": "end", "op_elapsed": "1.624ms"}
2025-03-17T10:17:37.306355Z     info    Starting retention policy enforcement service   {"log_id": "0vKua6a1000", "serv
ce": "retention", "check_interval": "30m"}
2025-03-17T10:17:37.307390Z     info    Starting precreation service    {"log_id": "0vKua6a1000", "service": "shard-pre
reation", "check_interval": "10m", "advance_period": "30m"}
2025-03-17T10:17:37.311092Z     info    Starting query controller       {"log_id": "0vKua6a1000", "service": "storage-r
ads", "concurrency_quota": 1024, "initial_memory_bytes_quota_per_query": 9223372036854775807, "memory_bytes_quota_per_q
ery": 9223372036854775807, "max_memory_bytes": 0, "queue_size": 1024}
2025-03-17T10:17:37.317330Z     info    Configuring InfluxQL statement executor (zeros indicate unlimited).     {"log_i
": "0vKua6a1000", "max_select_point": 0, "max_select_series": 0, "max_select_buckets": 0}
2025-03-17T10:17:37.324091Z     info    Starting        {"log_id": "0vKua6a1000", "service": "telemetry", "interval": "
h"}
2025-03-17T10:17:37.327611Z     info    Listening       {"log_id": "0vKua6a1000", "service": "tcp-listener", "transport
: "http", "addr": ":8086", "port": 8086}
```

사물인터넷 – 시계열데이터베이스(InfluxDB)

❖ 웹 GUI 실행 화면

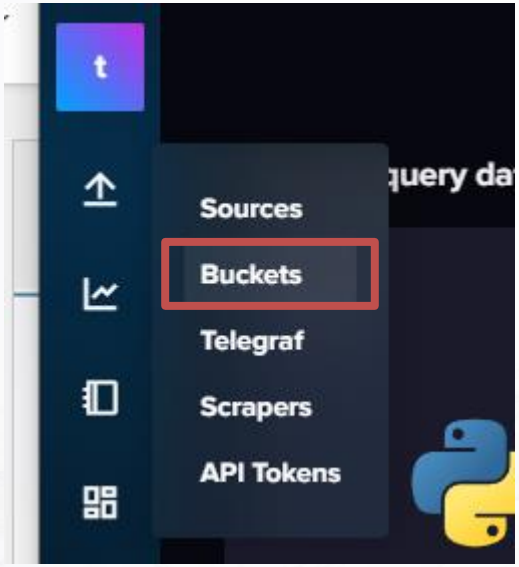
✓ http://localhost:8086/

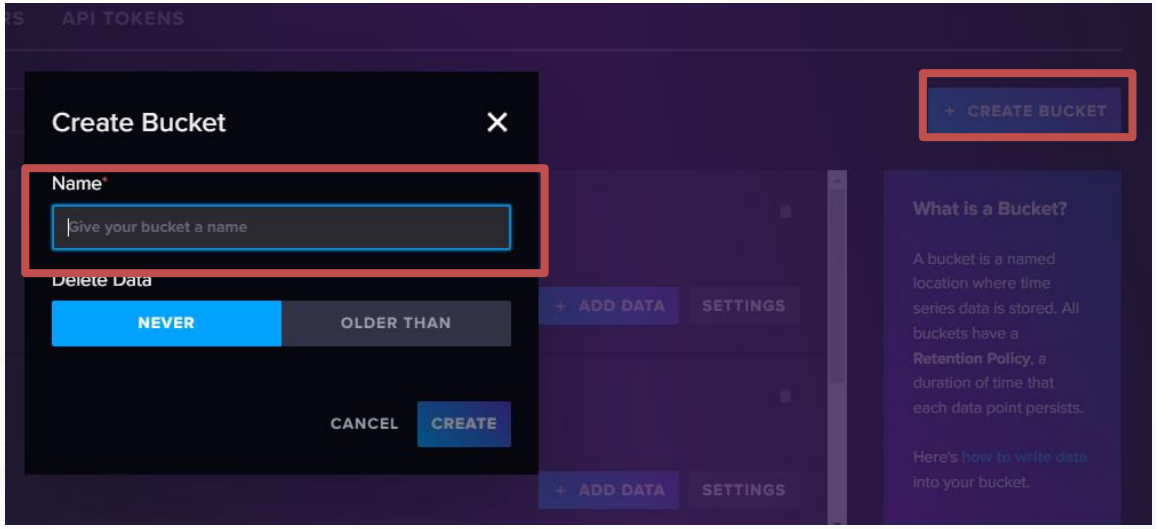


사물인터넷 – 시계열데이터베이스(InfluxDB)

❖ 웹 GUI 실행 화면

- ✓ http://localhost:8086/
- ✓ Buckets -> 데이터베이스 만들기(dust)

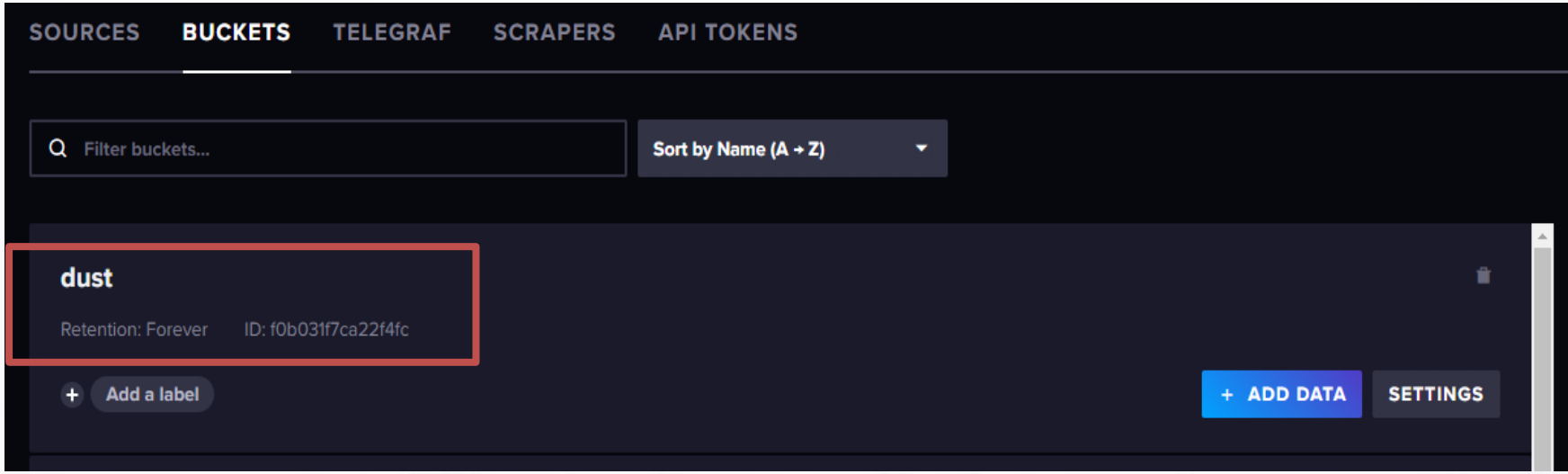




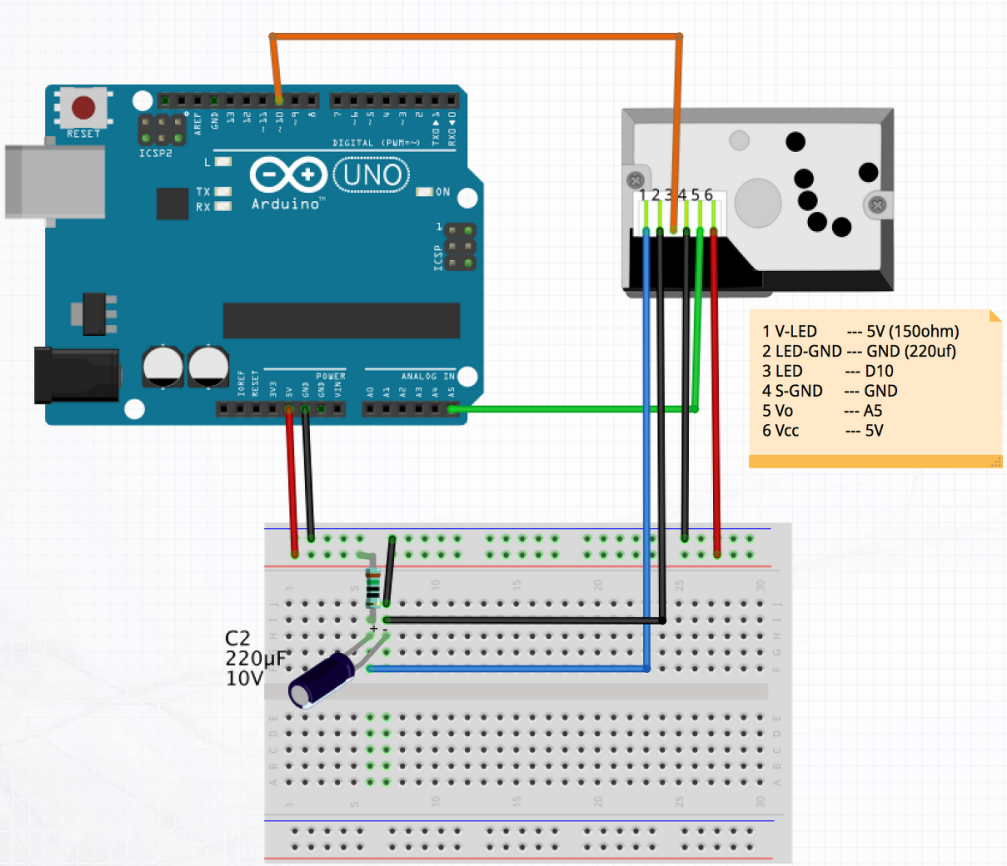
사물인터넷 – 시계열데이터베이스(InfluxDB)

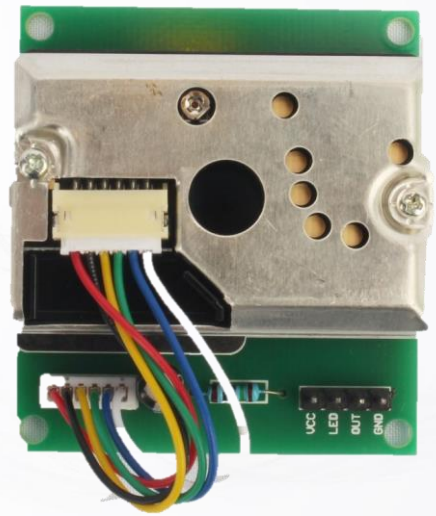
❖ 웹 GUI 실행 화면

✓ Buckets -> 데이터베이스 만들기(dust)



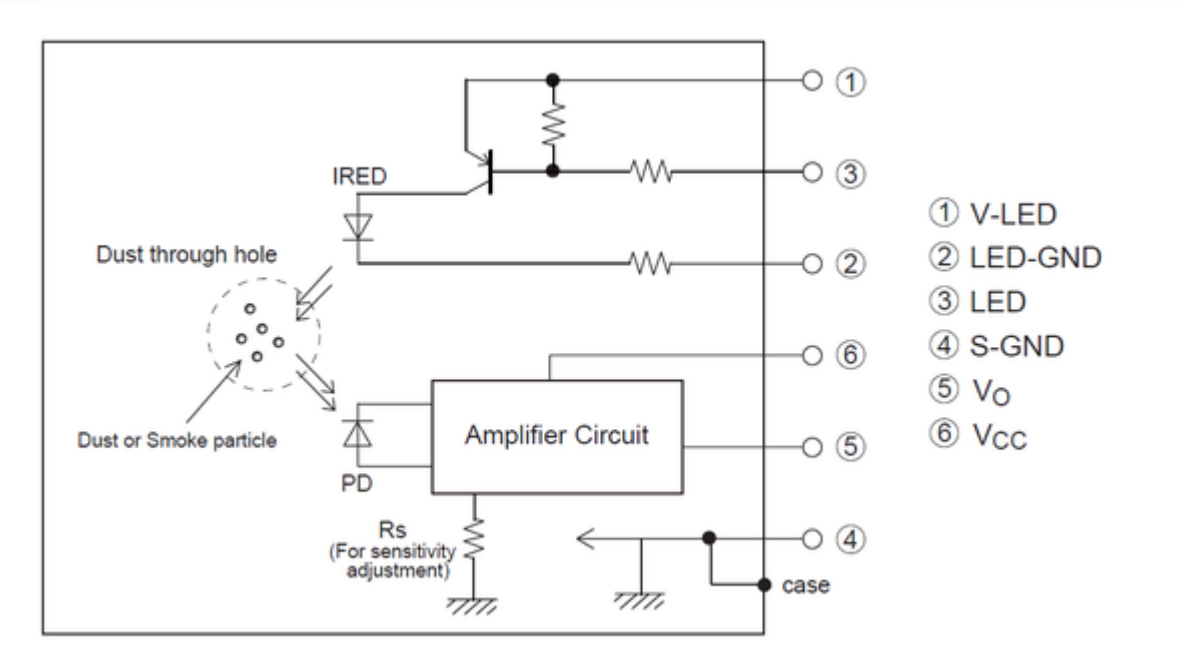
❖ 미세먼지센서(GP2Y1010AU0F)





❖ 미세먼지센서

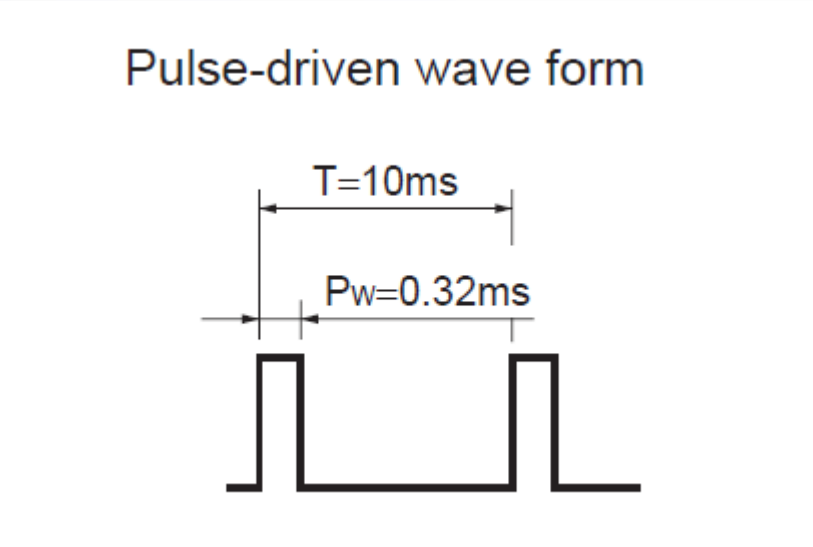
- DataSheet (GP2Y1010AU0F)

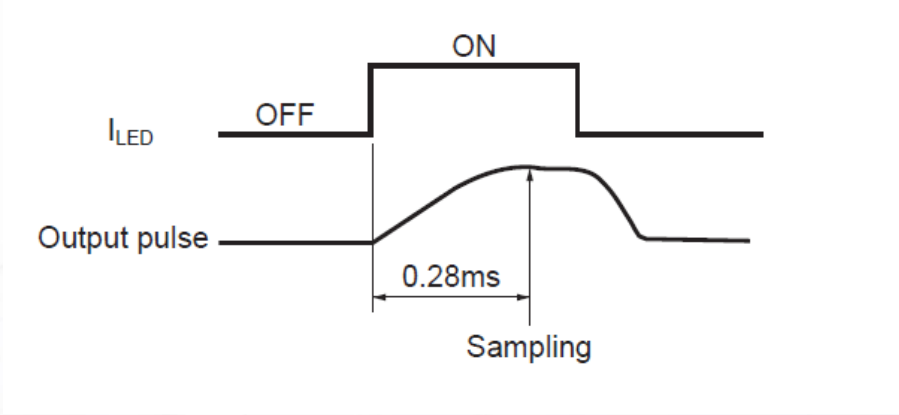


- 센서 중앙 홀을 통해 공기중에 먼지 량을 측정
- 원형 구멍 양옆으로 두개의 소자가 부착됨(적외선LED, 적외선 수신소자) ,

❖ 미세먼지센서 - 1

- DataSheet (GP2Y1010AU0F)





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

- 데이터 수신
- 적외선 LED 켠 후 0..28ms 흐르고 적외선 수신기를 작동시켜 값을 Read)

❖ 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
```

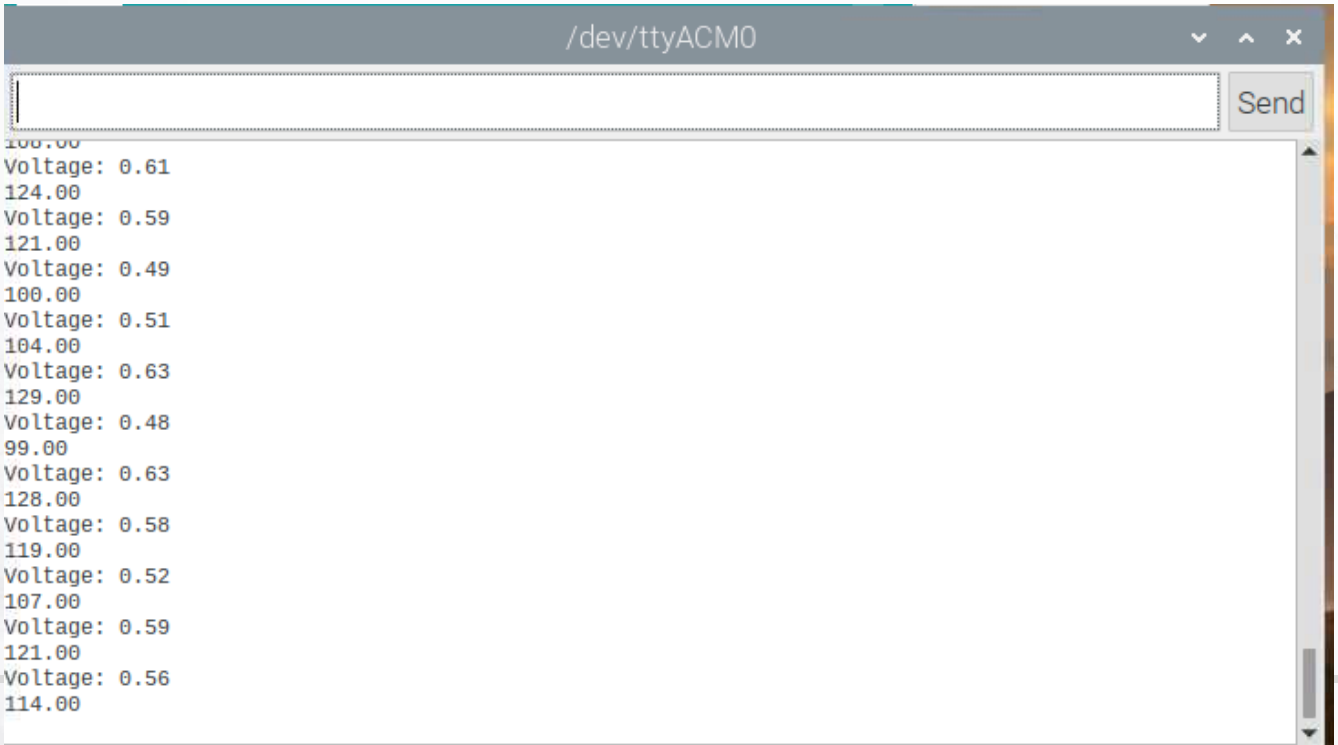
Autoscroll

사물인터넷 - 미세먼지센서 && 아두이노 연동 실습

❖ 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



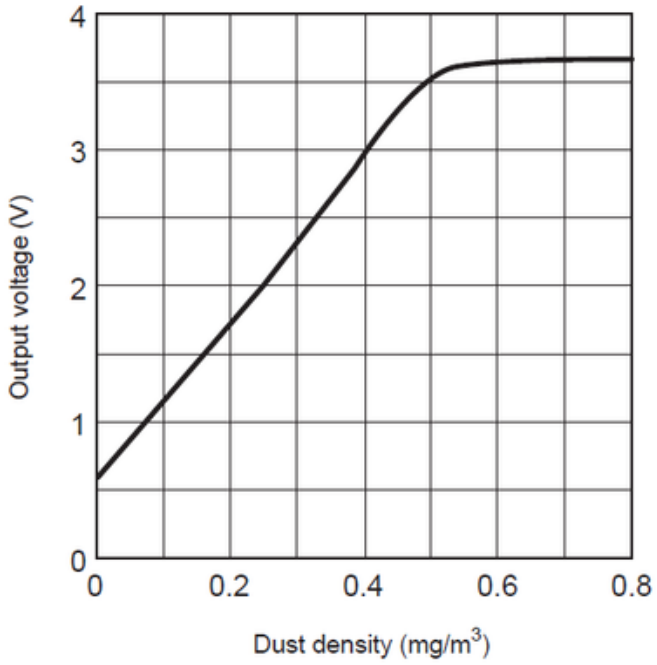
❖ 미세먼지센서 – 4

- 전압 값을 이용,
  미세먼지 양 측정

3-3 Electro-optical Characteristics (Ta=25℃, Vcc=5V)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Sensitivity	K	(*1)(*2)(*3)(*4)	0.425	0.5	0.575	V/ (100 μ g/m³)
Output voltage at no dust	Voc	(*2)(*3)(*4)	0.1	0.6	1.1	V
Output voltage range	VOH	RL=4.7kΩ (*2)(*3)(*4)	3.4	-	-	V
LED terminal current	ILED	LED terminal=0V (*2)(*3)	-	10	20	mA
Supply current	ICC	RL=∞ (*2)(*3)	-	11	20	mA

Fig. 3 Output Voltage vs.



사물인터넷 - 미세먼지센서 && 아두이노 연동 실습

❖ 미세먼지센서 – 5

- 아두이노 프로그램 코드

LED_test.ino

```
1   int Vo = A0;
2   int V_led = 12;
3
4   float Vo_value=0;
5   float Voltage = 0;
6   float dustDensity = 0;
7
8   void setup(){
9     Serial.begin(9600);
10    pinMode(V_led, OUTPUT);
11    pinMode(Vo, INPUT);
12  }
13  void loop(){
14    digitalWrite(V_led, LOW);
15    delayMicroseconds(280);
16    Vo_value = analogRead(Vo);
17    delayMicroseconds(40);
18    digitalWrite(V_led,HIGH);
19    delayMicroseconds(9680);
20
21    Voltage = Vo_value*5.0 / 1023.0;
22    dustDensity = (Voltage - 0.5)/0.005;
23
24    Serial.print("dust=" );
25    Serial.println(dustDensity);
26
27    delay(1000);
28  }
```

❖ Serial && influxdb 저장 미들웨어 예제 코드 (Python 사용)

```python
import serial
from influxdb_client import InfluxDBClient
import time


serial_port = 'COM4'
baud_rate = 9600
timeout = 2

# InfluxDB v2 설정
influxdb_url = "http://localhost:8086"
influxdb_token = "yb13lc8j_H7ANo9dv68YNVZdi3Yu3pZ7vwgVdWPxK1oOCtbznFZlWwzqRmRLZ9
influxdb_org = "test"  # InfluxDB organization
influxdb_bucket = "dust"  # 데이터가 저장될 bucket 이름

# InfluxDB 클라이언트 초기화
client = InfluxDBClient(url=influxdb_url, token=influxdb_token, org=influxdb_org
write_api = client.write_api()

# 시리얼 포트 열기
try:
    ser = serial.Serial(serial_port, baud_rate, timeout = timeout)
    print(f"Connected to {serial_port} at {baud_rate} baud")
except:
    print("Failed to connect to serial port")
    exit()
```

TODO

❖ Serial && influxdb 저장 미들웨어 예제 코드 (Python 사용) - 1

```
try:
    while True:
        if ser.in_waiting > 0:
            # 아두이노로부터 시리얼 데이터를 읽음
            line = ser.readline().decode('utf-8').strip()

            # 데이터가 유효한 경우 InfluxDB에 기록
            if "=" in line:
                key, value = line.split("=")
                try:
                    value = float(value)
                    data=f"sensor_data,device=arduino {key}={value}"
                    write_api.write(bucket=influxdb_bucket, record=data)
                    print(f"Data written to influxDB: {key}={value}")
                except ValueError:
                    print("Invalid data format")

        time.sleep(1)

except KeyboardInterrupt:
    print("프로그램이 종료되었습니다.")

finally:
    ser.close()
```

❖ Python module 설치 - 1

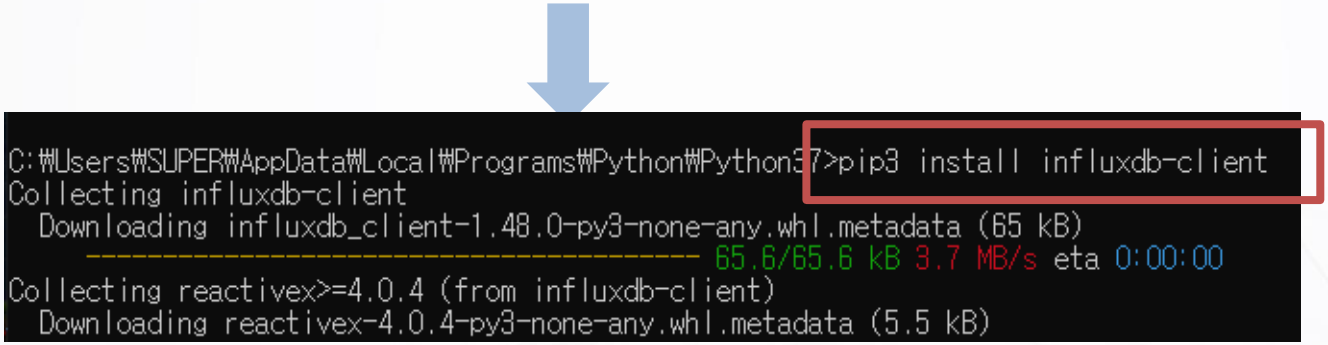
```
C:₩Users₩SUPER₩AppData₩Local₩Programs₩Python₩Python37>python dust.py
Traceback (most recent call last):
  File "dust.py", line 6, in <module>
    import serial
ModuleNotFoundError: No module named 'serial'

C:₩Users₩SUPER₩AppData₩Local₩Programs₩Python₩Python37>pip3 install serial
```

```
C:₩Users₩SUPER₩AppData₩Local₩Programs₩Python₩Python37>pip3 install pyserial
Collecting pyserial
  Downloading pyserial-3.5-py2.py3-none-any.whl.metadata (1.6 kB)
Downloading pyserial-3.5-py2.py3-none-any.whl (90 kB)
   ---------------------------------------- 90.6/90.6 kB 2.5 MB/s eta 0:00:00
Installing collected packages: pyserial
Successfully installed pyserial-3.5
```

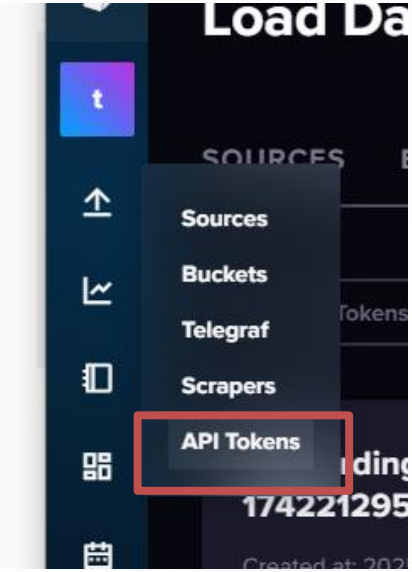
❖ Python module 설치 - 2

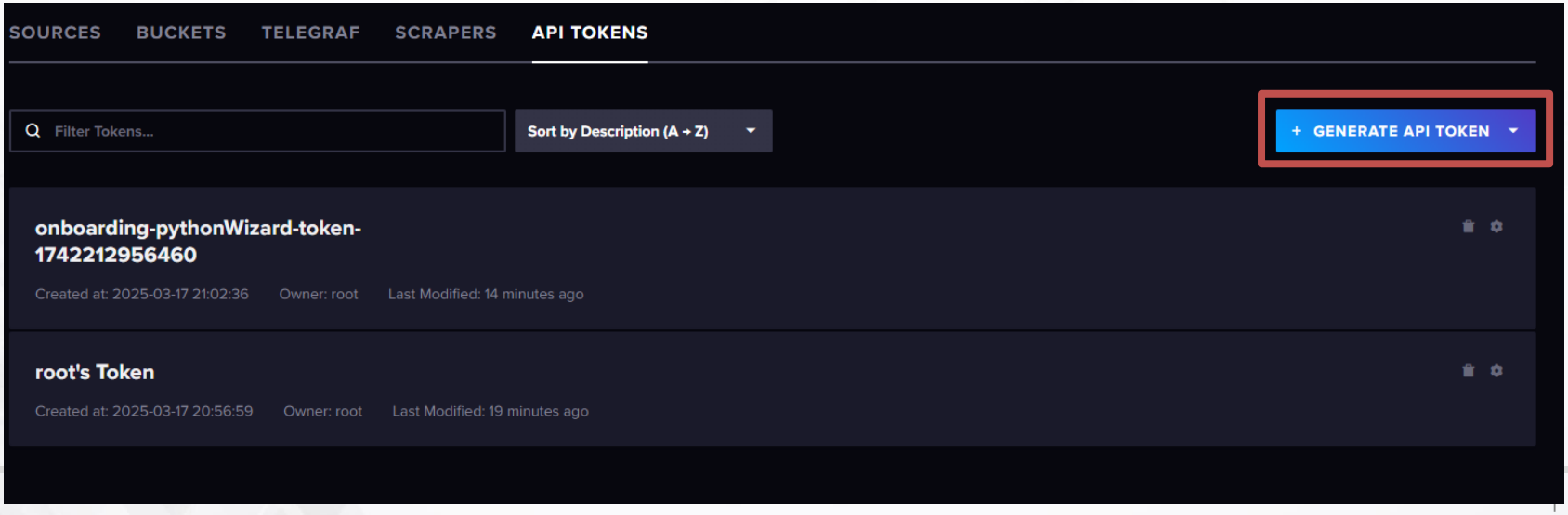
```
===== RESTART: C:/Users/PC/AppData/Local/Programs/Python/Python313/dust.py =====
Traceback (most recent call last):
  File "C:/Users/PC/AppData/Local/Programs/Python/Python313/dust.py", line 2, in
 <module>
    from influxdb_client import InfluxDBClient
ModuleNotFoundError: No module named 'influxdb_client'
```



사물인터넷 - 미세먼지센서 && 아두이노 연동 실습

❖ Influxdb token 생성

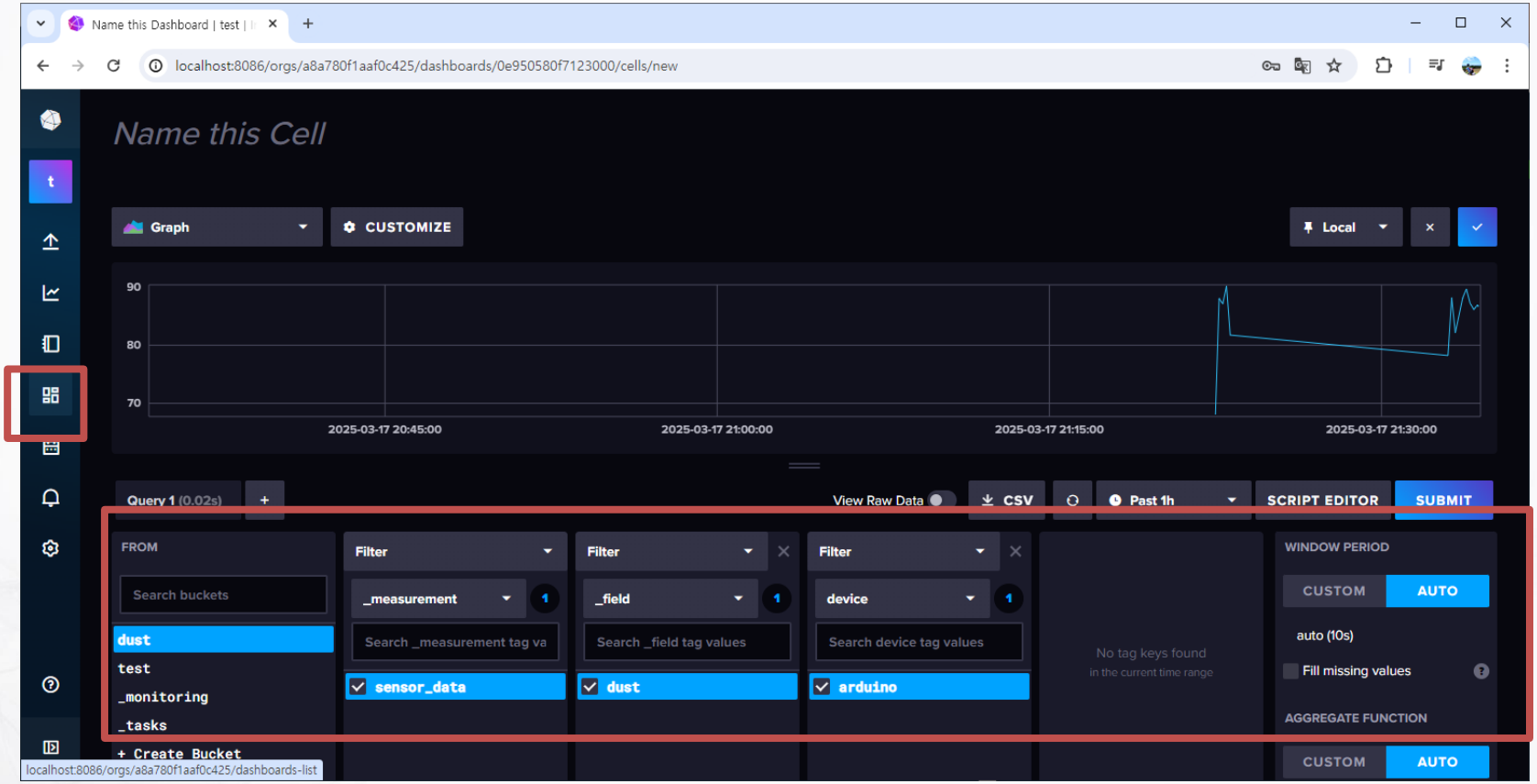




사물인터넷 - 미세먼지센서 && 아두이노 연동 실습

❖ Influxdb 데이터 조회

✓ Dashboard -> bucket -> measurement -> field -> tag



3주차 강의가 끝났습니다,

모두 고생하셨습니다.