

센서 데이터 모니터링 및 저장

Python을 활용하여 센서 데이터를 생성하고 IPC, MariaDB, MQTT를 통해 데이터를 저장, 전송, 시각화

변성일

목 차

1. 프로젝트 개요

목표

2. 기술 사항

사용 기술

사용 도구

3. 구성

센서 데이터

IPC

MySQL

MQTT

시각화

4. 구현

센서 데이터 생성

IPC

데이터 저장

MQTT 데이터 전송

데이터 시각화

5. 결과

프로젝트 목표

Python을 활용하여 센서 데이터를 생성하고 IPC, MariaDB, MQTT를 통해 데이터를 저장, 전송, 시각화

기술 사항

1

Python

2

MySQL

3

MQTT

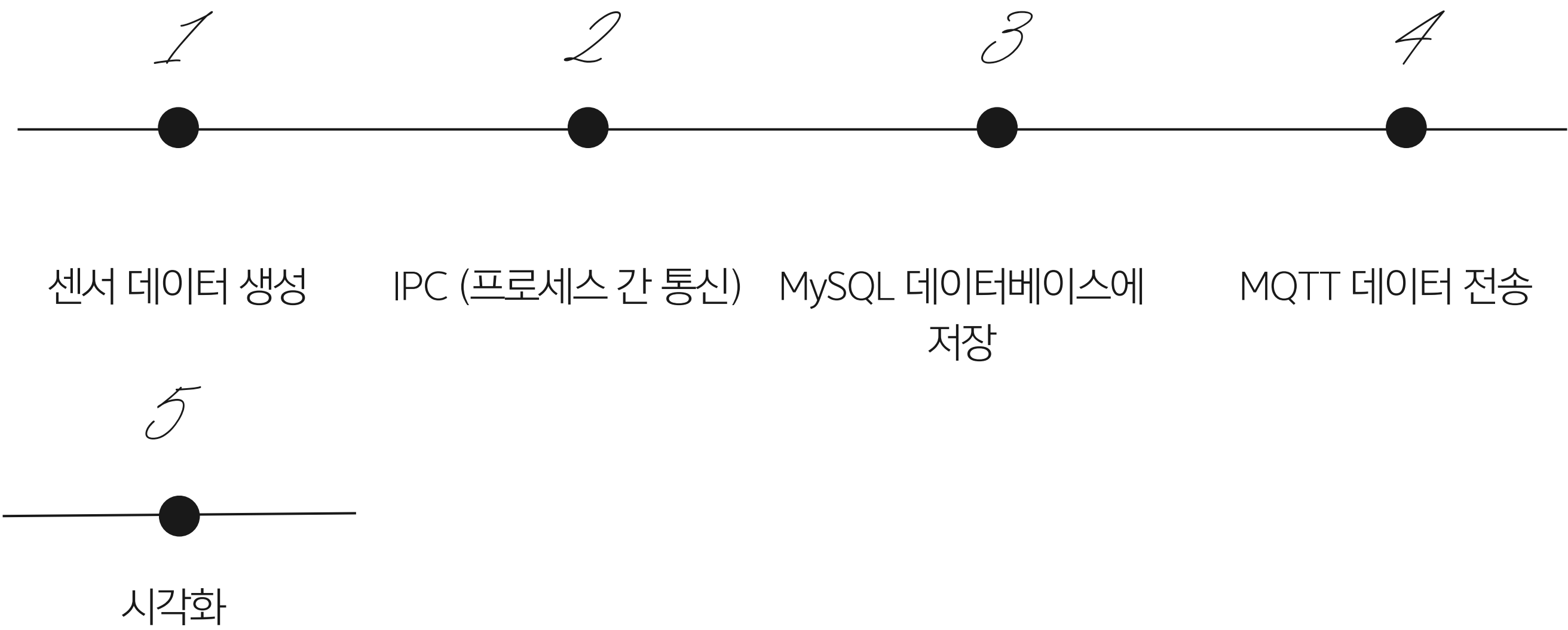
4

IPC

구성

진행계획

계획



1

-
- Python으로 가상의 센서 데이터 생성 (온도, 습도, 조도 등)
 - 1초 간격으로 데이터 생성

2

-
- IPC (프로세스 간 통신)
 - 파이프(FIFO)를 통해 데이터 전송

3

-
- MySQL 데이터베이스에 저장
 - 테이블 구조: id, timestamp, temperature, humidity, light_intensity

구성

구성

4



- MQTT 브로커를 통해 실시간 데이터 전송

구성

구성

5

-
- MQTT 데이터를 구독하여 실시간 플롯 생성

구현

① 센서 데이터 생성

```
import random
import time

def generate_sensor_data():
    while True:
        yield {
            "temperature": round(random.uniform(20.0, 30.0), 2),
            "humidity": round(random.uniform(40.0, 60.0), 2),
            "light_intensity": random.randint(300, 800),
        }
        time.sleep(1)
```

② IPC (프로세스 간 통신)

```
import os
import json

pipe_name = "/tmp/sensor_pipe"
if not os.path.exists(pipe_name):
    os.mkfifo(pipe_name)

def write_to_pipe(data):
    with open(pipe_name, "w") as pipe:
        pipe.write(json.dumps(data) + "\n")
        pipe.flush()

def read_from_pipe():
    with open(pipe_name, "r") as pipe:
        while True:
            yield json.loads(pipe.readline().strip())
```


③ 데이터 저장

```
import mysql.connector

def setup_database():
    conn = mysql.connector.connect(
        host="127.0.0.1",
        user="root",
        password="your_password", ←
        database="sensor_db"
    )
    cursor = conn.cursor()
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS sensor_data (
            id INT AUTO_INCREMENT PRIMARY KEY,
            timestamp DATETIME,
            temperature FLOAT,
            humidity FLOAT,
            light_intensity INT
        )
    """)
    conn.commit()
    conn.close()
```

④ MQTT 데이터 전송

```
import paho.mqtt.client as mqtt
import json

def publish_to_mqtt(data):
    client = mqtt.Client()
    client.connect("localhost")
    client.publish("sensor/data", json.dumps(data))
    client.disconnect()
```

⑤ 데이터 시각화

```
import matplotlib.pyplot as plt
from collections import deque

def monitor_data():
    temperatures, humidities, light_intensities = deque(maxlen=100), deque(maxlen=100)

    def on_message(client, userdata, message):
        data = json.loads(message.payload.decode())
        temperatures.append(data["temperature"])
        humidities.append(data["humidity"])
        light_intensities.append(data["light_intensity"])
        plt.plot(temperatures, label="Temperature")
        plt.legend()
        plt.pause(0.1)

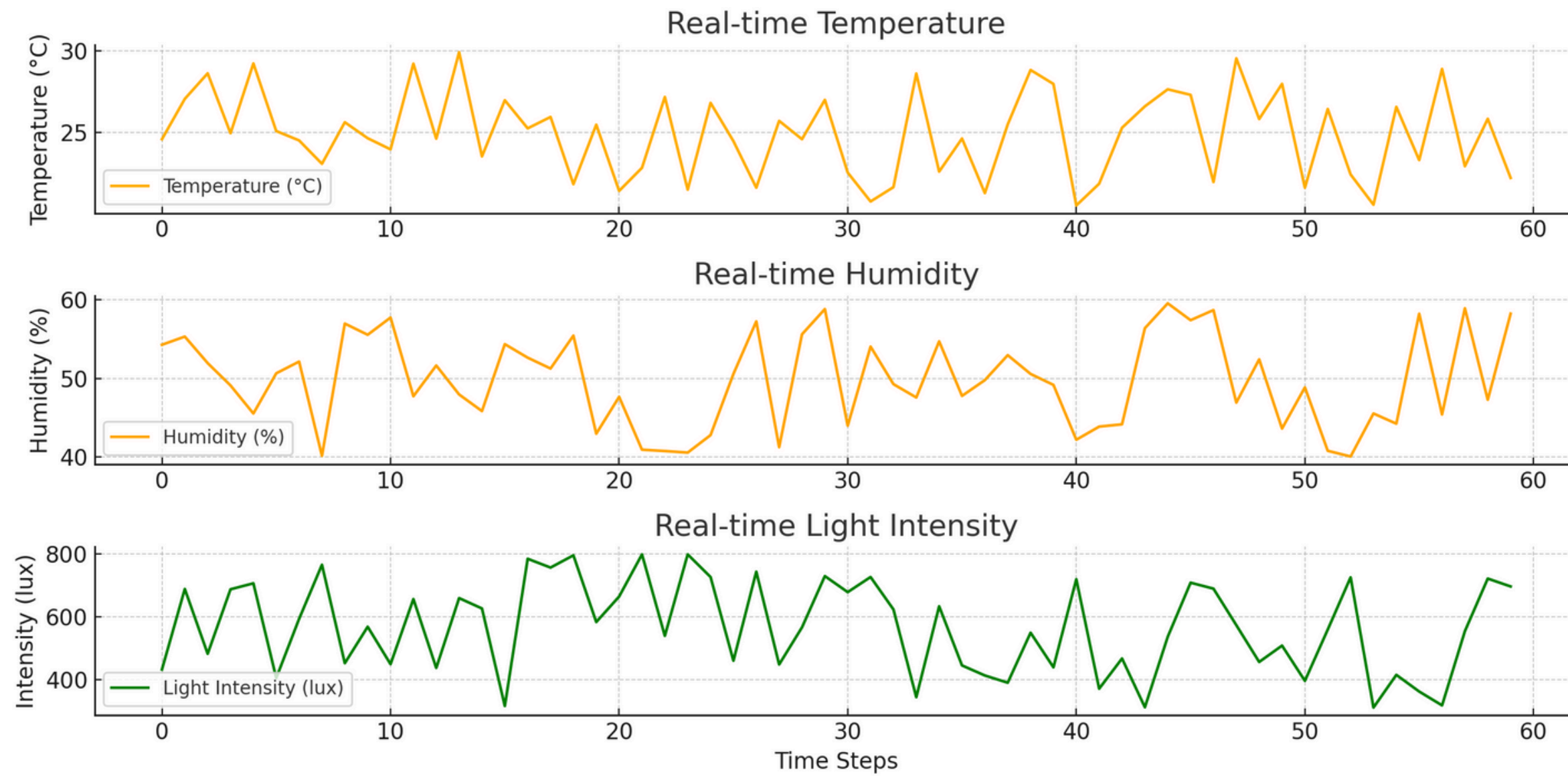
    client = mqtt.Client()
    client.on_message = on_message
    client.connect("localhost")
    client.subscribe("sensor/data")
    client.loop_start()
    plt.show()
```

결과



시각화

결과



Thank you!