

강원혁신플랫폼

리눅스프로그래밍

MQTT 응용





Python에서 Flask 웹 프레임워크와 함께 사용되는 환경 변수로 Flask 애플리케이션이 포함된 Python 파일의 이름으로 설정하는 환경변수는 무엇인가요?

FLASK_APP






학습 내용

- 1 시뮬레이션 plot(온도, 습도, 조도)
- 2 시뮬레이션 looping(온도, 습도, 조도)
- 3 SensorData 테이블 변경
- 4 MQTT응용

학습 목표

 MQTT를 응용할 수 있다.



시뮬레이션 plot(온도, 습도, 조도)



```
import datetime
import random
import matplotlib.pyplot as plt

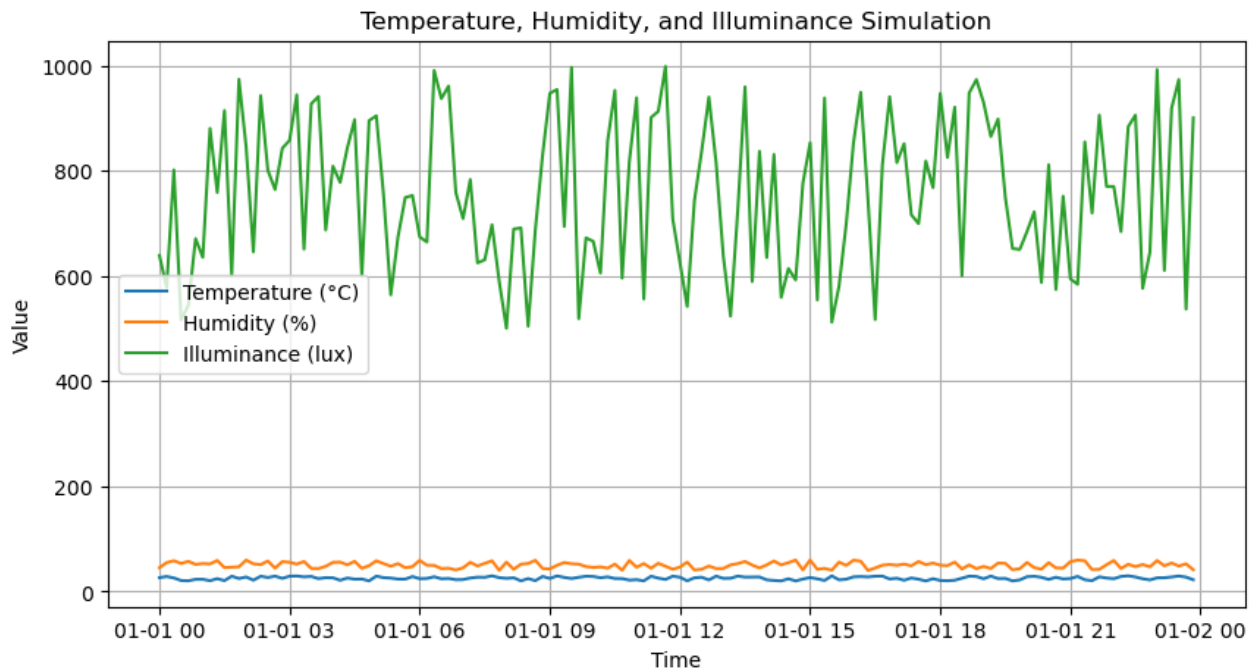
# Simulation parameters
start_date = datetime.datetime(2023, 1, 1, 0, 0, 0)
end_date = datetime.datetime(2023, 1, 2, 0, 0, 0)
time_delta = datetime.timedelta(minutes=10)
timestamps = [start_date + i*time_delta for i in range(int((end_date - start_date) /
time_delta))]
```

```
# Simulating temperature, humidity, and illuminance
temperature = [random.uniform(20, 30) for _ in range(len(timestamps))]
humidity = [random.uniform(40, 60) for _ in range(len(timestamps))]
illuminance = [random.uniform(500, 1000) for _ in range(len(timestamps))]
```



```
# Plotting the simulation results
plt.figure(figsize=(10, 5))
plt.plot(timestamps, temperature, label='Temperature (°C)')
plt.plot(timestamps, humidity, label='Humidity (%)')
plt.plot(timestamps, illuminance, label='Illuminance (lux)')
plt.xlabel('Time')
plt.ylabel('Value')
plt.title('Temperature, Humidity, and Illuminance Simulation')
plt.legend()
plt.grid(True)
plt.show()
```

시뮬레이션 plot(온도, 습도, 조도)



강원혁신플랫폼

리눅스프로그래밍



시뮬레이션 looping(온도, 습도, 조도)





```
import datetime
```

```
import random
```

```
import time
```

```
# Simulating temperature, humidity, and illuminance in real-time (infinite loop)
```

```
while True:
```

```
    current_time = datetime.datetime.now()
```

```
    temperature = random.uniform(20, 30)
```

```
    humidity = random.uniform(40, 60)
```

```
    illuminance = random.uniform(500, 1000)
```



시뮬레이션 looping(온도, 습도, 조도)

```
print(f"Time: {current_time} – Temperature: {temperature:.2f}°C, Humidity:  
{humidity:.2f}%, Illuminance: {illuminance:.2f} lux")
```

```
time.sleep(1) # Delay for real-time effect
```





시뮬레이션 looping(온도, 습도, 조도)

Time: 2023-01-01 00:00:00 - Temperature: 25.73°C, Humidity: 45.30%,
Illuminance: 915.96 lux

Time: 2023-01-01 00:10:00 - Temperature: 25.66°C, Humidity: 52.23%,
Illuminance: 612.27 lux

Time: 2023-01-01 00:20:00 - Temperature: 28.20°C, Humidity: 56.63%,
Illuminance: 928.46 lux

Time: 2023-01-01 00:30:00 - Temperature: 25.00°C, Humidity: 48.64%,
Illuminance: 735.56 lux

Time: 2023-01-01 00:40:00 - Temperature: 25.93°C, Humidity: 51.59%,
Illuminance: 710.53 lux

Time: 2023-01-01 00:50:00 - Temperature: 23.92°C, Humidity: 50.91%,
Illuminance: 873.47 lux





SensorData 테이블 변경





```
MariaDB [mydb]> desc SensorData;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
sensor_id	int(11)	YES	MUL	NULL	
reading	float	YES		NULL	
timestamp	datetime	YES		NULL	



```
MariaDB [mydb]> alter TABLE SensorData add column temperature float;
```

```
MariaDB [mydb]> alter TABLE SensorData add column humidity float;
```

```
MariaDB [mydb]> alter TABLE SensorData add column illuminance float;
```

```
MariaDB [mydb]>
```

```
MariaDB [mydb]> desc SensorData;
```

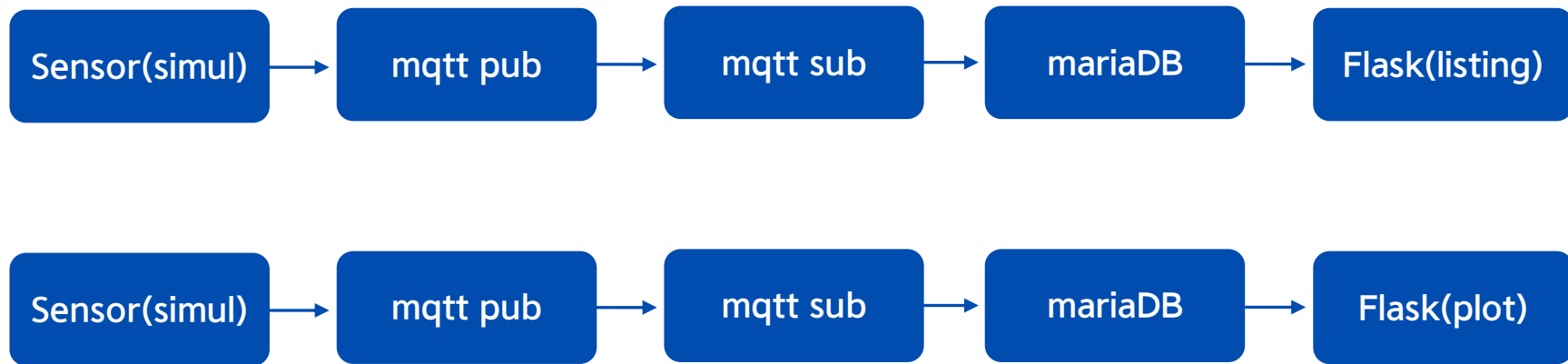


```
MariaDB [mydb]> desc SensorData;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
sensor_id	int(11)	YES	MUL	NULL	
reading	float	YES		NULL	
timestamp	datetime	YES		NULL	
temperature	float	YES		NULL	
humidity	float	YES		NULL	
illuminance	float	YES		NULL	

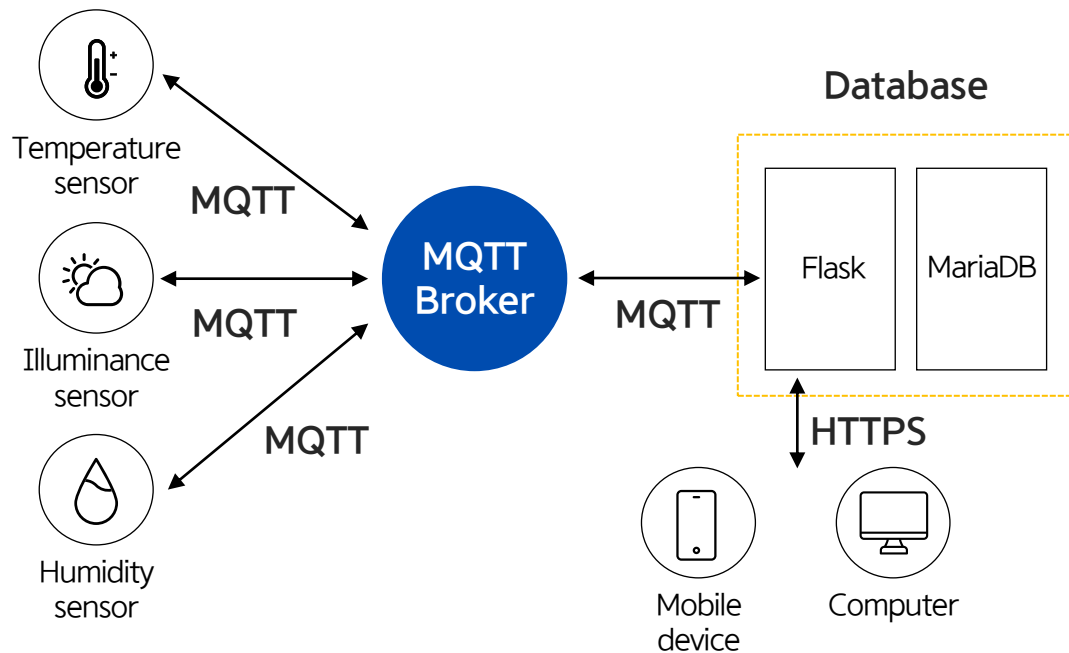


동작 구조





동작 구조



강원혁신플랫폼

리눅스프로그래밍



MQTT응용





```
(.venv) $ cat simul_sensors_mqtt_db.py
```

```
import datetime
```

```
import random
```

```
import time
```

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

```
import pymysql
```

```
# MQTT Broker configuration
```

```
broker_address = 'broker.hivemq.com'
```

```
broker_port = 1883
```

```
publish_topic = 'sensors'
```

```
subscribe_topic = 'sensors'
```



```
# MariaDB database configuration  
db_host = 'localhost'  
db_user = 'scott'  
db_password = 'tiger'  
db_name = 'mydb'  
table_name = 'SensorData'
```



```
# MQTT client setup and connection
```

```
mqtt_client = mqtt.Client()
```

```
mqtt_client.connect(broker_address, broker_port)
```

```
# MQTT client subscription and loop
```

```
mqtt_client.subscribe(subscribe_topic)
```

```
mqtt_client.on_message = on_message
```

```
mqtt_client.loop_start()
```

```
# Start the simulation and publishing process
```

```
simulate_and_send(mqtt_client)
```



```
# MQTT message callback
```

```
def on_message(client, userdata, message):
```

```
    try:
```

```
        payload = message.payload.decode()
```

```
        data = eval(payload) # Safely convert the string payload to a dictionary
```

```
    # Connect to the MariaDB database
```

```
        db_connection = pymysql.connect(host=db_host, user=db_user,  
password=db_password, database=db_name)
```

```
        cursor = db_connection.cursor()
```



Extract values from the received data

```
timestamp = datetime.datetime.strptime(data['time'], '%Y-%m-%d %H:%M:%S')
```

```
sensor_id = data['sensor_id']
```

```
reading = data['reading']
```

```
temperature = data['temperature']
```

```
humidity = data['humidity']
```

```
illuminance = data['illuminance']
```




```
query = f"INSERT INTO {table_name} (sensor_id, reading, timestamp, temperature,
humidity, illuminance) VALUES (%s, %s, %s, %s, %s, %s)"
cursor.execute(query, (sensor_id, reading, timestamp, temperature, humidity,
illuminance))
db_connection.commit()

# Close the database connection
db_connection.close()
print(f"Received and stored: Time: {timestamp}, SID: {sensor_id}, Reading:
{reading}, Temp: {temperature}°C, Humi: {humidity}%, Illuminance: {illuminance} lux")
except Exception as e:
    print(f"Error: {e}")
```

Function to simulate and send values in real-time

```
def simulate_and_send(client):
```

```
    while True:
```

```
        current_time = datetime.datetime.now()
```

```
        sensor_id = random.uniform(1,4)
```

```
        reading = random.uniform(20, 30)
```

```
        temperature = random.uniform(20, 30)
```

```
        humidity = random.uniform(40, 60)
```

```
        illuminance = random.uniform(500, 1000)
```

```
data = {  
    'time': current_time.strftime('%Y-%m-%d %H:%M:%S'),  
    'sensor_id': sensor_id,  
    'reading': reading,  
    'temperature': temperature,  
    'humidity': humidity,  
    'illuminance': illuminance  
}  
client.publish(publish_topic, str(data)) # Publish data to MQTT broker
```



```
print(f"Time: {current_time} – SID: {sensor_id:.2f}°C, Reading: {reading:.2f}°C, Temp:  
{temperature:.2f}°C, Humi: {humidity:.2f}%, Illuminance: {illuminance:.2f} lux")
```

```
time.sleep(1) # Delay for real-time effect
```



동작 구조





```
from flask import Flask, render_template, json, request  
import pymysql.cursors
```

```
app = Flask(__name__)
```

```
# Connect to the database
```

```
connection = pymysql.connect(host='localhost',  
/* MariaDB 연결 정보 생략 */
```

```
# Route for displaying the sensor data
```

```
@app.route('/')
```

```
def display_sensor_data():
```



```
with connection.cursor() as cursor:
```

```
    cursor.execute('SELECT * FROM SensorData ORDER BY id DESC LIMIT 100')
```

```
    sensor_data_data = cursor.fetchall()
```

```
    print(sensor_data_data)
```

```
    return render_template('index_mqtt.html', sensor_data=sensor_data_data)
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```



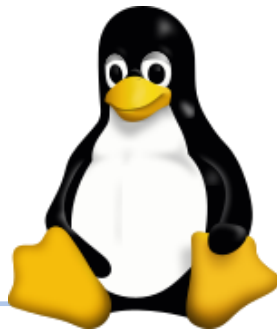
```
<h1>Sensor Data</h1>
<table border=1>
  <thead>
    <tr>
      <th>ID</th> <th>Sensor ID</th> <th>Reading</th> <th>Timestamp</th>
      <th>temperature</th>
      <th>humidity</th>
      <th>illuminance</th>
    </tr>
  </thead>
```




```
<tbody>
  {% for data in sensor_data %}
    <tr>
      {% for item, value in data.items(): %}
        <td align=center>{{ value }}</td>
      {% endfor %}
    </tr>
  {% endfor %}
</tbody>
</table>
```

```
MariaDB [mydb]> select * from SensorData;
```

id	sensor_id	reading	timestamp	temperature	humidity	illuminance
1	1	25.5	2023-07-14 13:55:11	NULL	NULL	NULL
2	2	30.2	2023-07-14 13:55:11	NULL	NULL	NULL
3	1	26.8	2023-07-14 13:55:11	NULL	NULL	NULL
4	3	18.9	2023-07-14 13:55:11	NULL	NULL	NULL
... 중간 생략						
91	2	22.8739	2023-07-14 20:11:28	21.9702	40.5939	568.771
92	2	20.0096	2023-07-14 20:11:29	21.9162	57.0926	988.962
94	3	23.0763	2023-07-14 20:11:31	26.3231	48.2287	582.964
95	3	20.7948	2023-07-14 20:11:32	25.8389	40.1456	521.737
97	2	26.242	2023-07-14 20:11:34	25.7664	51.936	590.64



FLASK_APP

Python에서 Flask 웹 프레임워크와 함께 사용되는 환경 변수
Flask 애플리케이션이 포함된 Python 파일의 이름으로 설정



Linux/macOS(터미널 사용)

```
$ export FLASK_APP=your_app.py
```

```
$ flask run
```

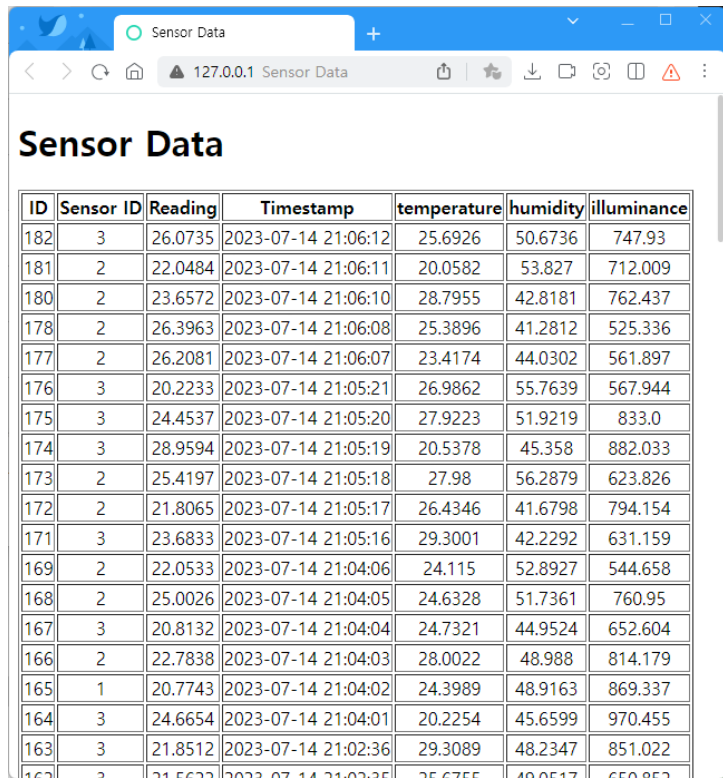


실행결과

```
(.venv) $ export FLASK_APP=helloflask_mqtt_db
```

```
(.venv) $ flask run
```

실행결과



The screenshot shows a web browser window with the title 'Sensor Data'. The address bar displays '127.0.0.1 Sensor Data'. The main content area is titled 'Sensor Data' and contains a table with 7 columns: ID, Sensor ID, Reading, Timestamp, temperature, humidity, and illuminance. The table lists 20 rows of sensor data, with IDs ranging from 163 to 182. The data includes timestamps from 2023-07-14 21:02:36 to 2023-07-14 21:06:12, and various readings for temperature, humidity, and illuminance.

ID	Sensor ID	Reading	Timestamp	temperature	humidity	illuminance
182	3	26.0735	2023-07-14 21:06:12	25.6926	50.6736	747.93
181	2	22.0484	2023-07-14 21:06:11	20.0582	53.827	712.009
180	2	23.6572	2023-07-14 21:06:10	28.7955	42.8181	762.437
178	2	26.3963	2023-07-14 21:06:08	25.3896	41.2812	525.336
177	2	26.2081	2023-07-14 21:06:07	23.4174	44.0302	561.897
176	3	20.2233	2023-07-14 21:05:21	26.9862	55.7639	567.944
175	3	24.4537	2023-07-14 21:05:20	27.9223	51.9219	833.0
174	3	28.9594	2023-07-14 21:05:19	20.5378	45.358	882.033
173	2	25.4197	2023-07-14 21:05:18	27.98	56.2879	623.826
172	2	21.8065	2023-07-14 21:05:17	26.4346	41.6798	794.154
171	3	23.6833	2023-07-14 21:05:16	29.3001	42.2292	631.159
169	2	22.0533	2023-07-14 21:04:06	24.115	52.8927	544.658
168	2	25.0026	2023-07-14 21:04:05	24.6328	51.7361	760.95
167	3	20.8132	2023-07-14 21:04:04	24.7321	44.9524	652.604
166	2	22.7838	2023-07-14 21:04:03	28.0022	48.988	814.179
165	1	20.7743	2023-07-14 21:04:02	24.3989	48.9163	869.337
164	3	24.6654	2023-07-14 21:04:01	20.2254	45.6599	970.455
163	3	21.8512	2023-07-14 21:02:36	29.3089	48.2347	851.022
162	2	21.5622	2023-07-14 21:02:35	25.6755	40.0517	650.852



01 • 시뮬레이션 plot(온도, 습도, 조도)

02 • 시뮬레이션 looping(온도, 습도, 조도)

03 • SensorData 테이블 변경

04 • MQTT응용