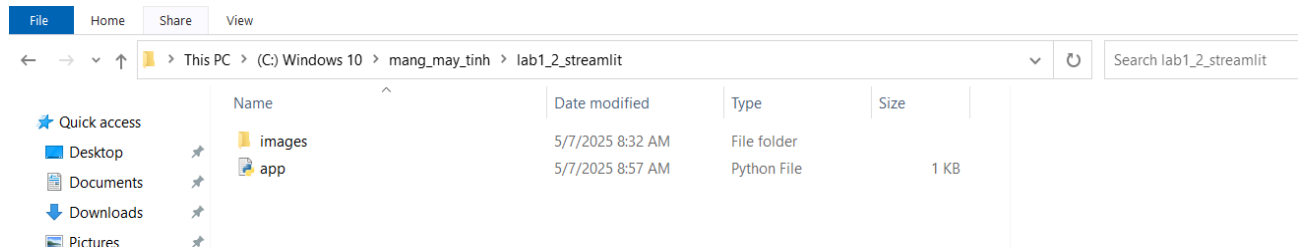


BÀI LAB 1.2 PHIÊN BẢN MÔ PHỎNG ĐIỀU KHIỂN LED QUA HTTP BẰNG STREAMLIT

Tạo thư mục lab1_2_streamlit gồm [app.py](#) và images:



Sau khi chạy đoạn code [app.py](#) ta được kết quả :

The screenshot shows a VS Code editor with the file `app.py` open. The code is as follows:

```
1 import streamlit as st
2 from PIL import Image
3 # Trạng thái LED mô phỏng
4 if "led_on" not in st.session_state:
5     st.session_state.led_on = False
6 # Xử lý sự kiện nút trước khi render
7 col1, col2 = st.columns(2)
8 with col1:
9     if st.button("BẬT LED"):
10         st.session_state.led_on = True
12 with col2:
13     if st.button("TẮT LED"):
14         st.session_state.led_on = False
15 # Sau khi cập nhật trạng thái → render ảnh
16 st.title("Điều khiển LED mô phỏng bằng Streamlit")
```

The terminal output shows the following messages:

```
2025-05-13 19:01:54.639 Thread 'MainThread': missing ScriptRunContext! This war
2025-05-13 19:01:54.638 Thread 'MainThread': missing ScriptRunContext! This war
ning can be ignored when running in bare mode.
2025-05-13 19:01:54.639 Thread 'MainThread': missing ScriptRunContext! This war
2025-05-13 19:01:54.639 Thread 'MainThread': missing ScriptRunContext! This war
```

Tiếp đó ta chạy trên cửa sổ CMD bằng lệnh ipconfig ta có kết quả sau:

```
Wireless LAN adapter Local Area Connection* 9:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 10:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . :
    IPv6 Address. . . . . : 2405:4802:17e0:a0a0:21b4:8387:dd13:79c
    Temporary IPv6 Address. . . . . : 2405:4802:17e0:a0a0:1540:2e5e:a7b3:b58a
    Link-local IPv6 Address . . . . . : fe80::21b4:8387:dd13:79c%12
    IPv4 Address. . . . . : 192.168.1.27
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::1%12
                                192.168.1.1

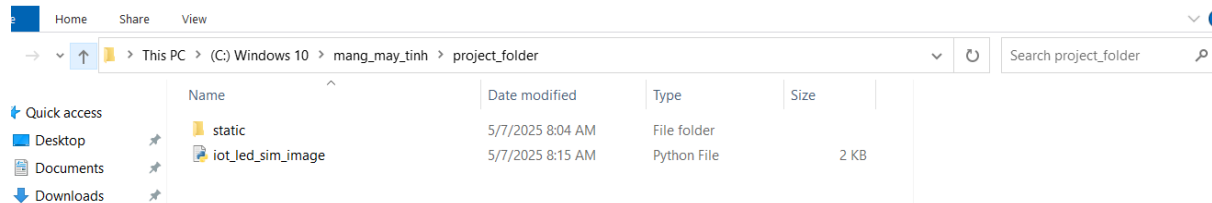
Ethernet adapter Bluetooth Network Connection 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :
```

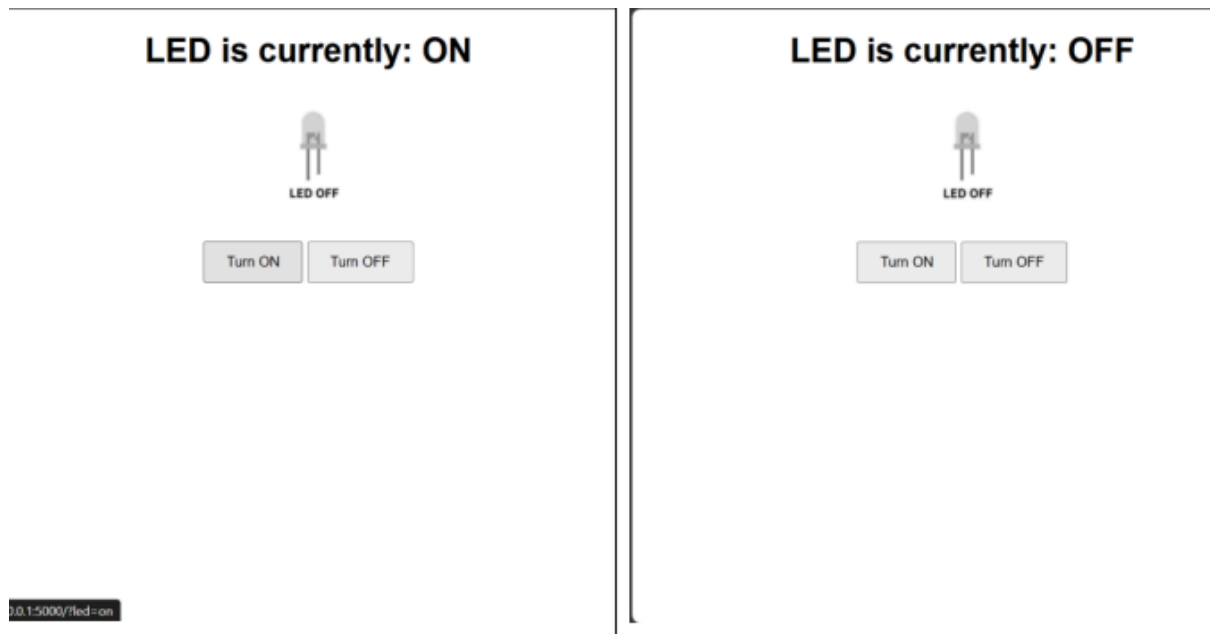
Ta có địa chỉ Ipv4 là: 192.168.1.1

BÀI LAB 1.2: Mô phỏng điều khiển LED qua HTTP bằng Framework Flask

Tạo thư mục project_folder gồm iot_led_sim_image.py và static:



Sau khi chạy đoạn code iot_led_sim_image.pyta được kết quả :



```

File Edit Selection View ... mang_may_tinh [Administrator]
SUBSCRIBE.py  IoT_data_logger.py  IoT_fake_publisher.py  IoT_data_logger_sqlite.py  IoT_led_sim_image.py project_folder
project_folder > IoT_led_sim_image.py > ...
1  from flask import Flask, request, render_template_string, url_for
2
3  app = Flask(__name__)
4
5  # Biến mô phỏng trạng thái LED
6  led_state = {"value": False}
7
8  # HTML giao diện có hình ảnh LED
9  HTML_TEMPLATE = """
10 <!DOCTYPE html>
11 <html>
12 <head>
13 ... <title>LED Control</title>
14 </head>
15 <body style="text-align:center; font-family:Arial">
16 ... <h1>LED is currently: {{ state }}</h1>
17 ...
18
19 . Use a production WSGI server instead.
20 * Running on all addresses (0.0.0.0)
21 * Running on http://127.0.0.1:5000
22 * Running on http://192.168.1.27:5000
23 Press CTRL+C to quit

```

Kết quả chạy trên cửa sổ CMD:

```

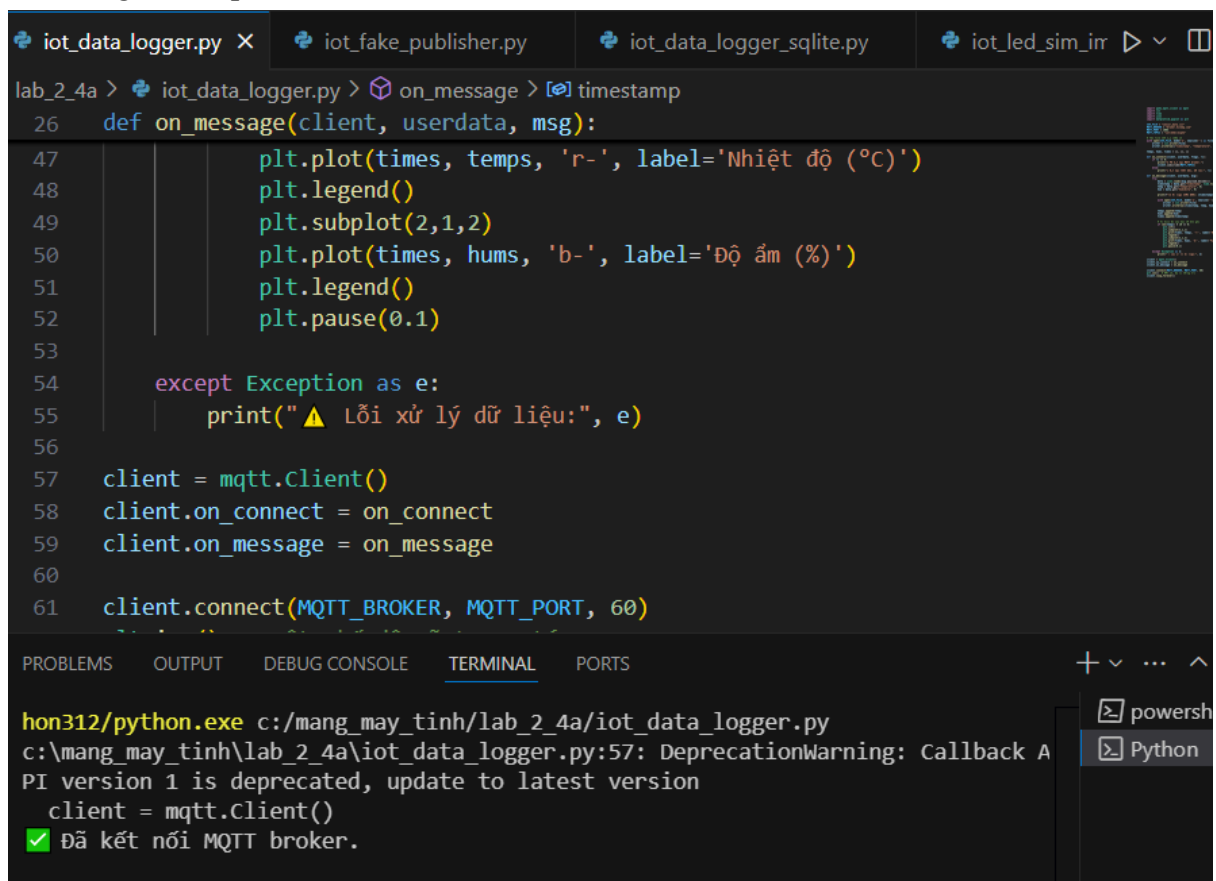
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.0.105:5000
Press CTRL+C to quit
192.168.0.105 - - [14/May/2025 08:33:52] "GET /?led=on HTTP/1.1" 200 -
192.168.0.105 - - [14/May/2025 08:33:52] "GET /static/led_off_white_120.png.png HTTP/1.1" 404 -
192.168.0.105 - - [14/May/2025 08:33:53] "GET /?led=off HTTP/1.1" 200 -
192.168.0.105 - - [14/May/2025 08:33:53] "GET /static/led_off_white_120.png.png HTTP/1.1" 404 -
192.168.0.105 - - [14/May/2025 08:33:53] "GET /?led=on HTTP/1.1" 200 -
192.168.0.105 - - [14/May/2025 08:33:53] "GET /static/led_off_white_120.png.png HTTP/1.1" 404 -
192.168.0.105 - - [14/May/2025 08:33:53] "GET /?led=off HTTP/1.1" 200 -
192.168.0.105 - - [14/May/2025 08:33:53] "GET /static/led_off_white_120.png.png HTTP/1.1" 404 -
192.168.0.105 - - [14/May/2025 08:33:54] "GET /?led=off HTTP/1.1" 200 -
192.168.0.105 - - [14/May/2025 08:33:54] "GET /static/led_off_white_120.png.png HTTP/1.1" 404 -
192.168.0.105 - - [14/May/2025 08:33:54] "GET /?led=on HTTP/1.1" 200 -
192.168.0.105 - - [14/May/2025 08:33:54] "GET /static/led_off_white_120.png.png HTTP/1.1" 404 -

```

Quan sát giao diện web hiển thị trạng thái LED và các nút điều khiển.

LAB 2.2a – GỬI DỮ LIỆU CẢM BIẾN LÊN MQTT BROKER

Dưới đây là ESP32 mô phỏng (trên Wokwi) đọc dữ liệu từ cảm biến DHT22, kết nối Wifi và gửi kết quả dữ liệu như hình sau:

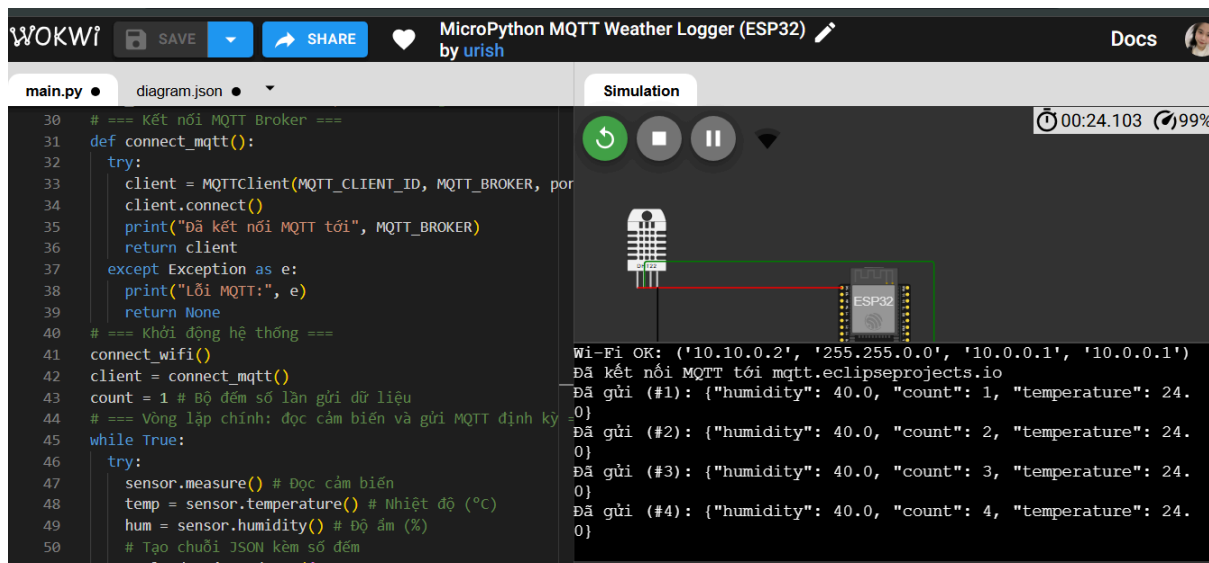


The screenshot displays the Wokwi IDE interface. The top panel shows four open files: `iot_data_logger.py`, `iot_fake_publisher.py`, `iot_data_logger_sqlite.py`, and `iot_led_sim_irr`. The main editor displays the `iot_data_logger.py` file with the following Python code:

```
lab_2_4a > iot_data_logger.py > on_message > [x] timestamp
26 def on_message(client, userdata, msg):
47     plt.plot(times, temps, 'r-', label='Nhiệt độ (°C)')
48     plt.legend()
49     plt.subplot(2,1,2)
50     plt.plot(times, hums, 'b-', label='Độ ẩm (%)')
51     plt.legend()
52     plt.pause(0.1)
53
54     except Exception as e:
55         print("⚠ Lỗi xử lý dữ liệu:", e)
56
57 client = mqtt.Client()
58 client.on_connect = on_connect
59 client.on_message = on_message
60
61 client.connect(MQTT_BROKER, MQTT_PORT, 60)
```

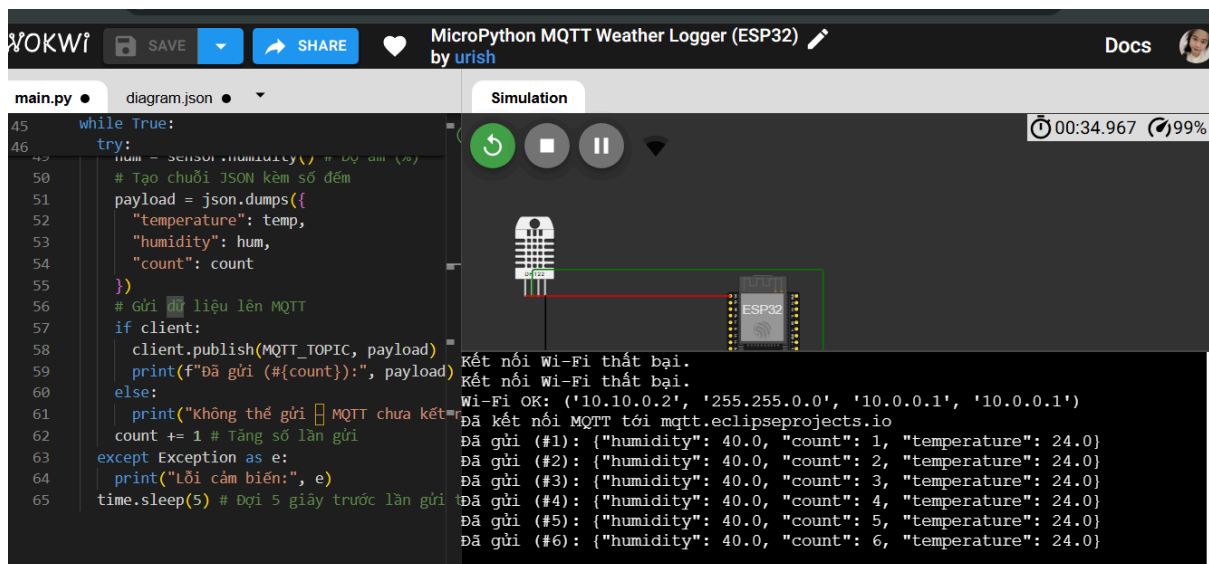
The bottom panel shows the terminal output for the command `hon312/python.exe c:/mang_may_tinh/lab_2_4a/iot_data_logger.py`. The output includes a deprecation warning and a confirmation message:

```
c:\mang_may_tinh\lab_2_4a\iot_data_logger.py:57: DeprecationWarning: Callback A
PI version 1 is deprecated, update to latest version
client = mqtt.Client()
✅ Đã kết nối MQTT broker.
```



LAB 2.2b – SUBSCRIBE DỮ LIỆU CẢM BIẾN TỪ MQTT BẰNG PYTHON TRÊN PC

Dưới đây là mô phỏng ESP32+DHT22 trong wokwi:



Tiếp theo, ta sẽ thấy dòng JSON xuất hiện liên tục mỗi 2 giây khi mở terminal hoặc CMD và chạy lệnh: python [subscriber.py](#)

```
Administrator: C:\Windows\system32\cmd.exe - python "C:\mang_may_tinh\LAB 2.2b\SUBSCRIBE.py"
Microsoft Windows [Version 10.0.19044.1263]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>python "C:\mang_may_tinh\LAB 2.2b\SUBSCRIBE.py"
C:\mang_may_tinh\LAB 2.2b\SUBSCRIBE.py:12: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
Kết nối MQTT thành công!
iot/khdl/esp32: {"temperature": 23.06, "humidity": 53.65, "timestamp": 1747142369.6418347}
iot/khdl/esp32: {"temperature": 23.84, "humidity": 35.16, "timestamp": 1747142371.6450522}
iot/khdl/esp32: {"temperature": 23.13, "humidity": 39.06, "timestamp": 1747142373.6496406}
iot/khdl/esp32: {"temperature": 24.85, "humidity": 41.74, "timestamp": 1747142375.6587064}
iot/khdl/esp32: {"temperature": 25.86, "humidity": 44.27, "timestamp": 1747142377.6678474}
iot/khdl/esp32: {"temperature": 26.94, "humidity": 52.72, "timestamp": 1747142379.6710627}
iot/khdl/esp32: {"temperature": 27.95, "humidity": 40.44, "timestamp": 1747142381.6778681}
iot/khdl/esp32: {"temperature": 26.18, "humidity": 46.07, "timestamp": 1747142383.6823885}
iot/khdl/esp32: {"temperature": 24.14, "humidity": 42.56, "timestamp": 1747142385.6848817}
iot/khdl/esp32: {"temperature": 26.91, "humidity": 44.24, "timestamp": 1747142387.687823}
```

LAB 2.4: LƯU TRỮ VÀ TRỰC QUAN HÓA DỮ LIỆU IoT

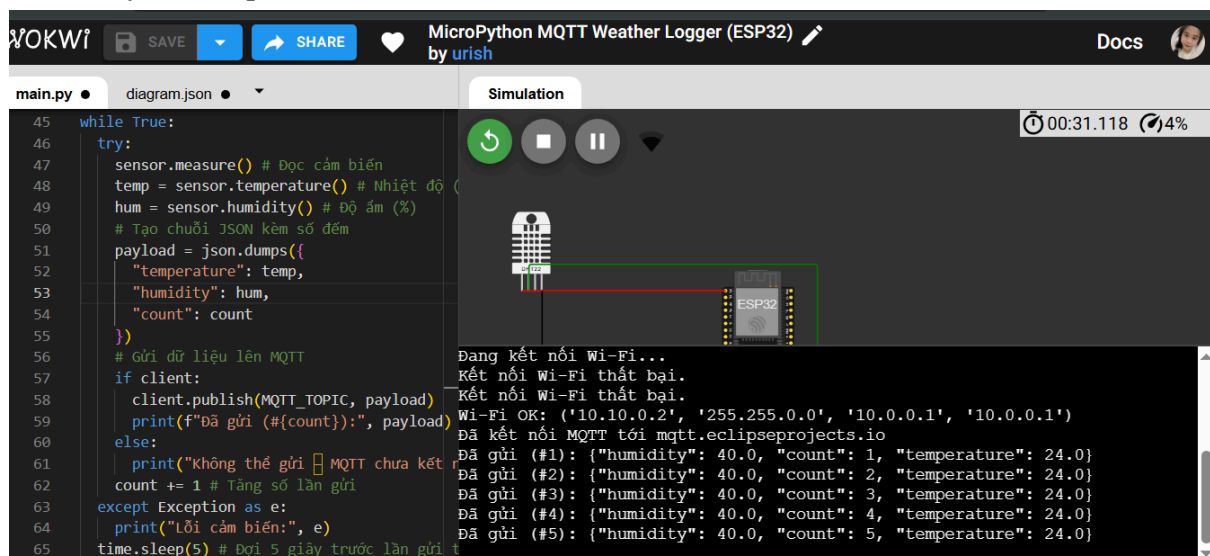
Lab 2.4a: Mô phỏng với wokwi (DHT22+ESP 32)

Kiểm thử Lab 2.4a

Bước 1: Khởi động Publisher Chạy mô phỏng trong Wokwi với ESP32 + DHT22 (main.py đang publish JSON lên iot/khdl/esp32)

Đảm bảo mỗi 2 giây có 1 bản tin JSON gửi lên.

Dưới đây là kết quả:



```
WOKWI SAVE SHARE MicroPython MQTT Weather Logger (ESP32) by urish Docs
main.py • diagram.json • Simulation 00:31.118 4%

45 while True:
46     try:
47         sensor.measure() # Đọc cảm biến
48         temp = sensor.temperature() # Nhiệt độ (°C)
49         hum = sensor.humidity() # Độ ẩm (%)
50         # Tạo chuỗi JSON kèm số đếm
51         payload = json.dumps({
52             "temperature": temp,
53             "humidity": hum,
54             "count": count
55         })
56         # Gửi dữ liệu lên MQTT
57         if client:
58             client.publish(MQTT_TOPIC, payload)
59             print(f"Đã gửi ({count}):", payload)
60         else:
61             print("Không thể gửi MQTT chưa kết nối")
62         count += 1 # Tăng số lần gửi
63     except Exception as e:
64         print("Lỗi cảm biến:", e)
65     time.sleep(2) # Đợi 2 giây trước lần gửi tiếp
```

Đang kết nối Wi-Fi...
Kết nối Wi-Fi thất bại.
Kết nối Wi-Fi thất bại.
Wi-Fi OK: ('10.10.0.2', '255.255.0.0', '10.0.0.1', '10.0.0.1')
Đã kết nối MQTT tới mqtt.eclipseprojects.io
Đã gửi (#1): {"humidity": 40.0, "count": 1, "temperature": 24.0}
Đã gửi (#2): {"humidity": 40.0, "count": 2, "temperature": 24.0}
Đã gửi (#3): {"humidity": 40.0, "count": 3, "temperature": 24.0}
Đã gửi (#4): {"humidity": 40.0, "count": 4, "temperature": 24.0}
Đã gửi (#5): {"humidity": 40.0, "count": 5, "temperature": 24.0}

Bước 2: Kiểm tra dữ liệu hiển thị trong terminal

Chạy `iot_data_logger.py` Vào thư mục `~\lab2.4\` có chứa file `iot_data_logger.py`

Gõ lệnh: `python iot_data_logger.py`

Quan sát log hiển thị: Dữ liệu nhận được: timestamp, nhiệt độ, độ ẩm

Nếu có lỗi, kiểm tra định dạng JSON hoặc kết nối MQTT

Dưới đây là kết quả:

```
client = mqtt.Client()
✅ Đã kết nối MQTT broker.
📡 Dữ liệu nhận được: 1747154067.894199, 24.81, 53.19
📡 Dữ liệu nhận được: 1747154069.896335, 25.22, 41.9
📡 Dữ liệu nhận được: 1747154071.8982978, 27.55, 51.02
📡 Dữ liệu nhận được: 1747154073.9004862, 24.43, 54.26
📡 Dữ liệu nhận được: 1747154075.9026675, 23.59, 45.11
📡 Dữ liệu nhận được: 1747154077.904151, 26.72, 35.77
📡 Dữ liệu nhận được: 1747154079.9056664, 25.91, 41.64
📡 Dữ liệu nhận được: 1747154081.907329, 27.41, 51.6
📡 Dữ liệu nhận được: 1747154081.907329, 27.41, 51.6
📡 Dữ liệu nhận được: 1747154083.9082084, 27.22, 54.6
📡 Dữ liệu nhận được: 1747154085.9105098, 23.62, 44.83
📡 Dữ liệu nhận được: 1747154087.9123468, 24.1, 53.84
📡 Dữ liệu nhận được: 1747154089.9142194, 26.64, 46.68
```

Bước 3: Kiểm tra file CSV

Mở file `sensor_data.csv` bằng Excel hoặc Notepad

Kiểm tra dữ liệu có được ghi dòng mới đúng định dạng không

Dưới đây là kết quả:

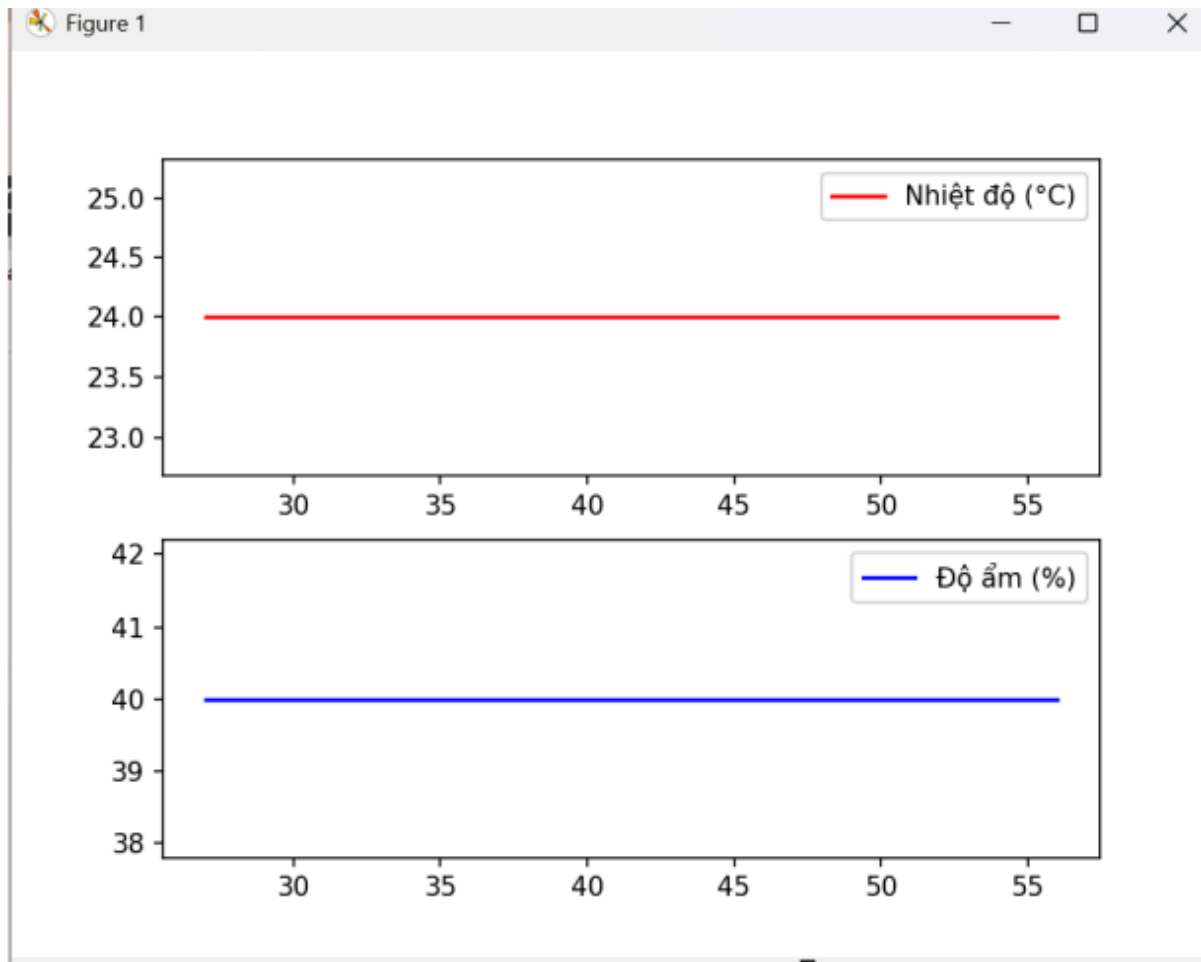
	A	B	C	D	E
1	timestamp	temperature	humidity		
2	1747154068	24.81	53.19		
3	1747154070	25.22	41.9		
4	1747154072	27.55	51.02		
5	1747154074	24.43	54.26		
6	1747154076	23.59	45.11		
7	1747154078	26.72	35.77		
8	1747154080	25.91	41.64		
9	1747154082	27.41	51.6		
10	1747154084	27.22	54.6		
11	1747154086	23.62	44.83		
12	1747154088	24.1	53.84		
13	1747154090	26.64	46.68		

Bước 4: Kiểm tra vẽ biểu đồ

Sau 10 bản tin: xuất hiện cửa sổ vẽ matplotlib

Biểu đồ gồm 2 phần: Trên: nhiệt độ theo thời gian Dưới: độ ẩm theo thời gian

Dưới đây là kết quả:



Bước 5: Dừng thử nghiệm

Nhấn Ctrl+C trong terminal để dừng script

Lab 2.4b: Mô phỏng dữ liệu ngẫu nhiên ngay trong publisher

Bước 1 Khởi động Publisher:

Thay vì chạy wokwi, chúng ta chạy một cửa sổ terminal: terminal 1: chạy: python iot_fake_publisher.py

Mở Terminal 2: chạy: python iot_data_logger.py

Dưới đây là kết quả:

```

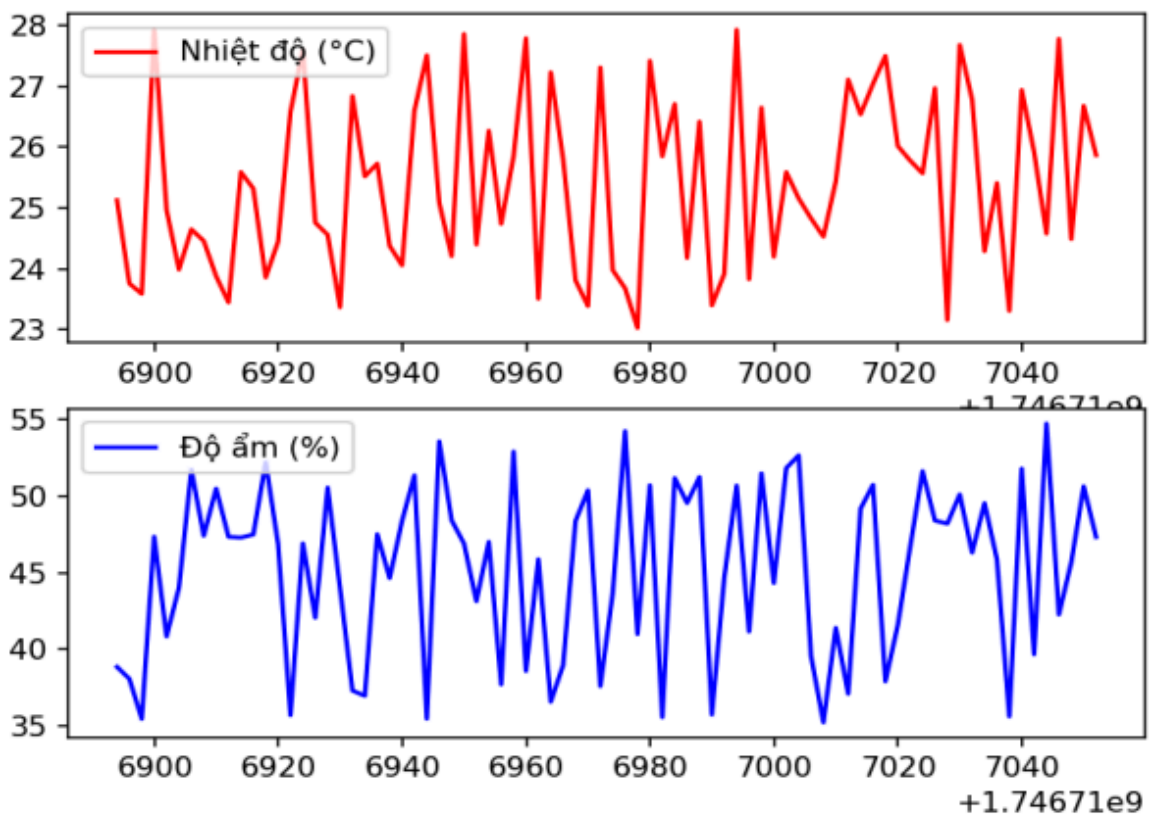
📶 Dữ liệu nhận được: 1747186722.9942584, 23.9, 37.5
📶 Dữ liệu nhận được: 1747186724.9951787, 25.12, 54.83
📶 Dữ liệu nhận được: 1747186726.9992342, 26.27, 49.87
📶 Dữ liệu nhận được: 1747186729.0029392, 23.49, 37.86
📶 Dữ liệu nhận được: 1747186731.0043576, 26.15, 41.03
📶 Dữ liệu nhận được: 1747186733.0060813, 24.28, 44.04
📶 Dữ liệu nhận được: 1747186735.011405, 24.29, 44.81
📶 Dữ liệu nhận được: 1747186737.015331, 26.2, 48.2
📶 Dữ liệu nhận được: 1747186739.016918, 26.28, 51.39

```

```

📶 Đã gửi: {'temperature': 25.44, 'humidity': 45.36, 'timestamp': 1747186649.9133084}
📶 Đã gửi: {'temperature': 25.33, 'humidity': 45.54, 'timestamp': 1747186651.9178987}
📶 Đã gửi: {'temperature': 27.79, 'humidity': 53.74, 'timestamp': 1747186653.920337}
📶 Đã gửi: {'temperature': 27.45, 'humidity': 49.65, 'timestamp': 1747186655.9224324}
📶 Đã gửi: {'temperature': 23.54, 'humidity': 50.39, 'timestamp': 1747186657.9245985}
📶 Đã gửi: {'temperature': 26.63, 'humidity': 52.8, 'timestamp': 1747186659.9298756}
📶 Đã gửi: {'temperature': 27.78, 'humidity': 39.81, 'timestamp': 1747186661.931985}
📶 Đã gửi: {'temperature': 24.81, 'humidity': 54.21, 'timestamp': 1747186663.934706}
📶 Đã gửi: {'temperature': 27.17, 'humidity': 36.77, 'timestamp': 1747186665.937413}

```



Lab 2.4c. Lưu dữ liệu cảm biến vào SQLite và vẽ biểu đồ realtime

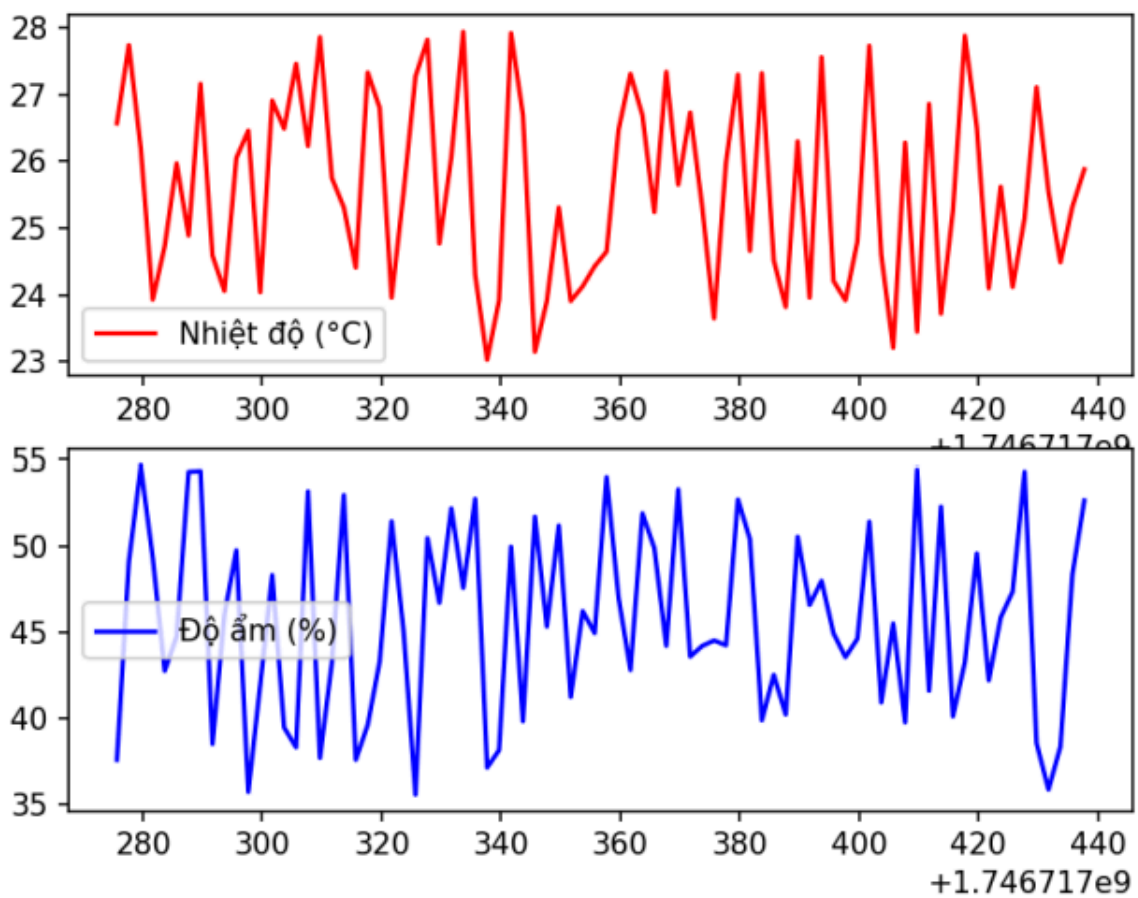
Bước 1: Terminal 1: python iot_fake_publisher.py

Terminal 2: python iot_data_logger_sqlite.py

Các bước còn lại thực hiện tương tự Lab 2.4a

```
C:\Users\Administrator>python "C:\mang_may_tinh\LAB_2_4B\iot_fake_publisher.py"
C:\mang_may_tinh\LAB_2_4B\iot_fake_publisher.py:8: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
    client = mqtt.Client()
[2] Đã gửi: {'temperature': 27.15, 'humidity': 52.13, 'timestamp': 1747187136.0384247}
[2] Đã gửi: {'temperature': 26.12, 'humidity': 37.46, 'timestamp': 1747187138.0399845}
[2] Đã gửi: {'temperature': 27.25, 'humidity': 53.39, 'timestamp': 1747187140.0438948}
[2] Đã gửi: {'temperature': 27.27, 'humidity': 44.13, 'timestamp': 1747187142.0478806}
[2] Đã gửi: {'temperature': 25.31, 'humidity': 54.04, 'timestamp': 1747187144.0500681}
[2] Đã gửi: {'temperature': 25.54, 'humidity': 46.85, 'timestamp': 1747187146.054705}
```

```
C:\Users\Administrator>python "C:\mang_may_tinh\lab_2_4c\iot_data_logger_sqlite.py"
C:\mang_may_tinh\lab_2_4c\iot_data_logger_sqlite.py:62: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
    client = mqtt.Client()
[2] Đã kết nối MQTT broker.
[2] Dữ liệu nhận: 1747187142.0478806, 27.27, 44.13
[2] Dữ liệu nhận: 1747187144.0500681, 25.31, 54.04
[2] Dữ liệu nhận: 1747187146.054705, 25.54, 46.85
[2] Dữ liệu nhận: 1747187148.0589404, 24.26, 36.38
```



Lab 2.4. d. Đọc dữ liệu từ file .csv

Giả sử file `sensor_data.csv` có nội dung như sau: 1 timestamp,temperature,h

1	timestamp,temperature,humidity
2	2023-07-15 10:30:00,25.5,60.0
3	2023-07-15 10:31:00,25.7,59.8

Sau khi chạy đoạn code `read_data_from_csv.py` ta có kết quả sau:

Figure 1

— □ ×

