Adebanke Damola-Fashola

ITAI 3377- AI at the Edge and IIOT Environments

Professor Patricia McManus

February 11, 2025.

Reflective Journal: IIoT Protocols Project

1. **Introduction:** This project aimed to design and implement an IIoT sensor network using MQTT, CoAP, and OPC UA protocols. These protocols facilitate data transmission between industrial sensors and a central system for monitoring and visualization. The project involved setting up sensor simulations, developing corresponding servers, and implementing data visualization techniques.

My personal goals for this project were to:

- Gain hands-on experience with MQTT, CoAP, and OPC UA.
- Understand the strengths and weaknesses of each protocol in an IIoT environment.
- Develop troubleshooting skills by debugging connectivity and data transmission issues.
- Enhance my Python programming skills, particularly in async programming and real-time data visualization.
- Creating my first repository on Git Hub.

I expected to face challenges in protocol integration, coding, and handling real-time data processing, but I aimed to develop practical solutions to overcome them.

2. **Personal Contributions:** I worked alone on this project. I was responsible for implementing and troubleshooting the following components:

- **MQTT Sensor Simulation**: I modified the mqtt_sensor_simulation.py script to publish sensor data to an MQTT broker.
- **CoAP Sensor Simulation**: I modified the coap_sensor_simulation.py to send simulated temperature and humidity data to a CoAP server.
- **CoAP Server**: I developed coap_server.py to handle both GET and POST requests and ensure data retrieval for visualization.
- **OPC UA Sensor Simulation**: I modified the opcua_sensor_simulation.py to simulate an OPC UA client that writes sensor values.
- **Data Visualization**: I modified the data visualization Python codes to generate the data plotting for the sensors.

.

3. Learning Outcomes

MQTT:

- I learned how MQTT's publish-subscribe model works, and its lightweight nature makes it ideal for low-bandwidth applications.
- I understood how to configure an MQTT broker and manage topics and QoS levels for reliable data transfer.
- I discovered the importance of keeping the MQTT loop running to maintain active subscriptions.

CoAP:

- I gained experience with CoAP's request-response model, which is similar to HTTP but optimized for constrained devices.
- I learned how CoAP uses UDP, which makes it lightweight and susceptible to packet loss.

- I understood the need to handle retransmissions and proper server binding to avoid connection issues.

OPC UA:

- I explored OPC UA’s structured data model, which supports hierarchical nodes for industrial applications.
- I learned how to establish an OPC UA client-server connection and read/write sensor values.
- I understood the benefits of OPC UA’s security features (encryption, authentication) for industrial automation.

Hands-on Insights:

- I realized the importance of debugging network issues, like ensuring correct port binding and firewall configurations.
- I gained a deeper understanding of asynchronous programming to handle real-time data streaming.
- I improved my ability to visualize sensor data dynamically using Matplotlib.

Key Takeaways:

- I gained hands-on experience with implementing and troubleshooting MQTT, CoAP, and OPC UA protocols in an IIoT sensor network.
- I learned the strengths and limitations of each protocol, particularly in terms of reliability, efficiency, and data transmission.
- I developed a deeper understanding of asynchronous communication and event-driven architecture.
- My problem-solving skills improved by debugging network connectivity issues and optimizing data visualization.

- I recognized the importance of secure and efficient data exchange in industrial IoT applications.

4. Challenges and Solutions

CoAP Server Not Responding:

- Issue: The CoAP client was failing with a Network Error.
- Solution: I verified the server binding and changed it from 127.0.0.1 to 0.0.0.0 to accept external requests. I allowed UDP traffic on port 5683 in the firewall.
- Lesson Learned: Always verify that the server is correctly bound and using the correct IP/port.

MQTT Messages Not Being Processed:

- Issue: The client was not receiving messages after a few iterations.
- Solution: I implemented client.loop_start() to keep the MQTT event loop running in the background.
- Lesson Learned: MQTT requires continuous event loop handling to process incoming messages.

Data Visualization Disappearing:

- Issue: Matplotlib's plot would disappear immediately after execution.
- Solution: I used plt.pause() within a while True loop to keep the visualization running.
- Lesson Learned: Matplotlib requires periodic updates to maintain a real-time plot.

OPC UA Client Failing to Connect:

- Issue: The OPC UA client was unable to retrieve values from the server.

- Solution: I ensured that the correct Node IDs were used, and the server was running with the appropriate namespace configuration.
- Lesson Learned: OPC UA's node hierarchy must be carefully handled to correctly read/write values.

5. **Future Applications**

Applying Knowledge to Future Projects:

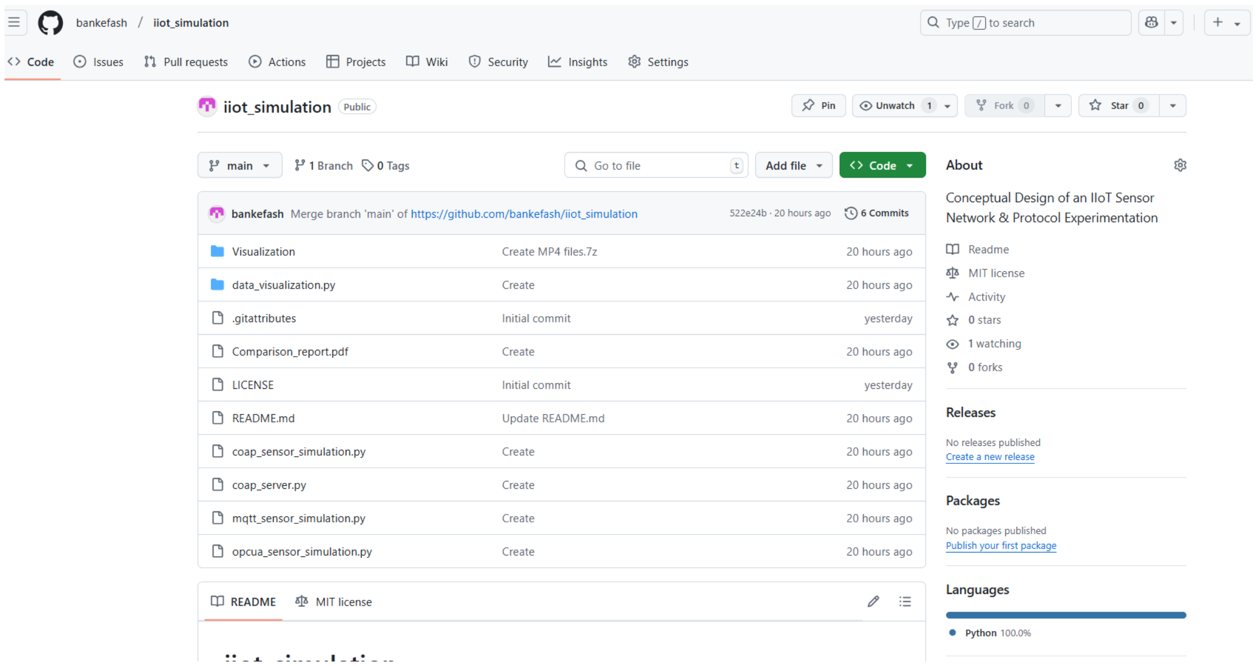
- This project provided real-world experience in implementing IIoT protocols widely used in industrial automation and smart manufacturing.
- The skills learned in troubleshooting network communications can be applied to other IoT and cloud-based systems.
- The ability to visualize real-time sensor data is useful for building monitoring dashboards in future projects.

Potential Improvements and Next Steps:

- Implement Advanced Security: Adding authentication and encryption to CoAP and MQTT connections.
- Optimize Data Processing: Using a time-series database (e.g., InfluxDB) to store and analyze historical sensor data.
- Scale the System: Deploying the system on a cloud platform to support multiple devices over the internet.
- Enhance Visualization: Integrating Grafana or a web-based dashboard instead of Matplotlib.

This project has been an enriching experience, equipping me with critical technical skills and problem-solving abilities that will be valuable in my future academic and professional endeavors.

6. GitHub Repository: https://github.com/bankefash/iiot_simulation



bankefash / iiot_simulation

Type / to search

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

iiot_simulation Public

Pin · Unwatch 1 · Fork 0 · Star 0

main · 1 Branch · 0 Tags · Go to file · Add file · Code

bankefash Merge branch 'main' of https://github.com/bankefash/iiot_simulation 522e24b · 20 hours ago · 6 Commits

Visualization	Create MP4 files.7z	20 hours ago
data_visualization.py	Create	20 hours ago
.gitattributes	Initial commit	yesterday
Comparison_report.pdf	Create	20 hours ago
LICENSE	Initial commit	yesterday
README.md	Update README.md	20 hours ago
coap_sensor_simulation.py	Create	20 hours ago
coap_server.py	Create	20 hours ago
mqtt_sensor_simulation.py	Create	20 hours ago
opcua_sensor_simulation.py	Create	20 hours ago

README · MIT license

About

Conceptual Design of an IIoT Sensor Network & Protocol Experimentation

Readme
MIT license
Activity
0 stars
1 watching
0 forks

Releases

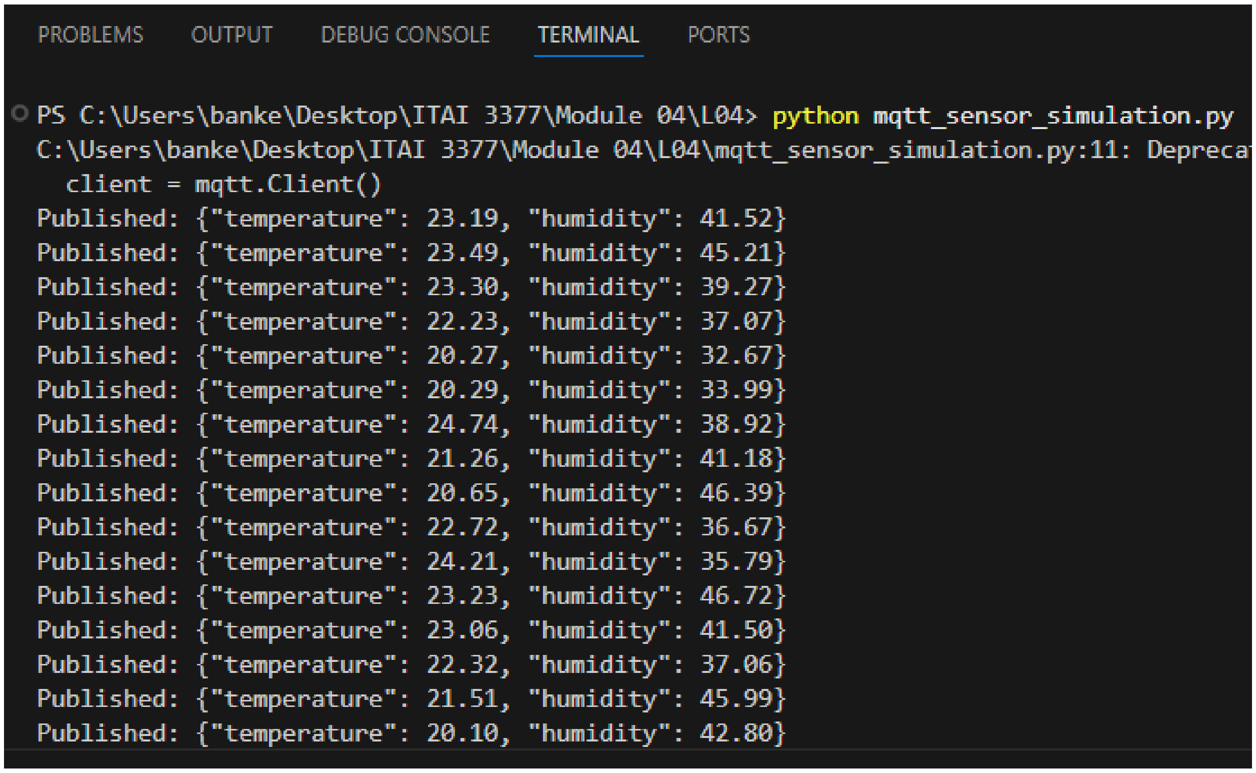
No releases published
Create a new release

Packages

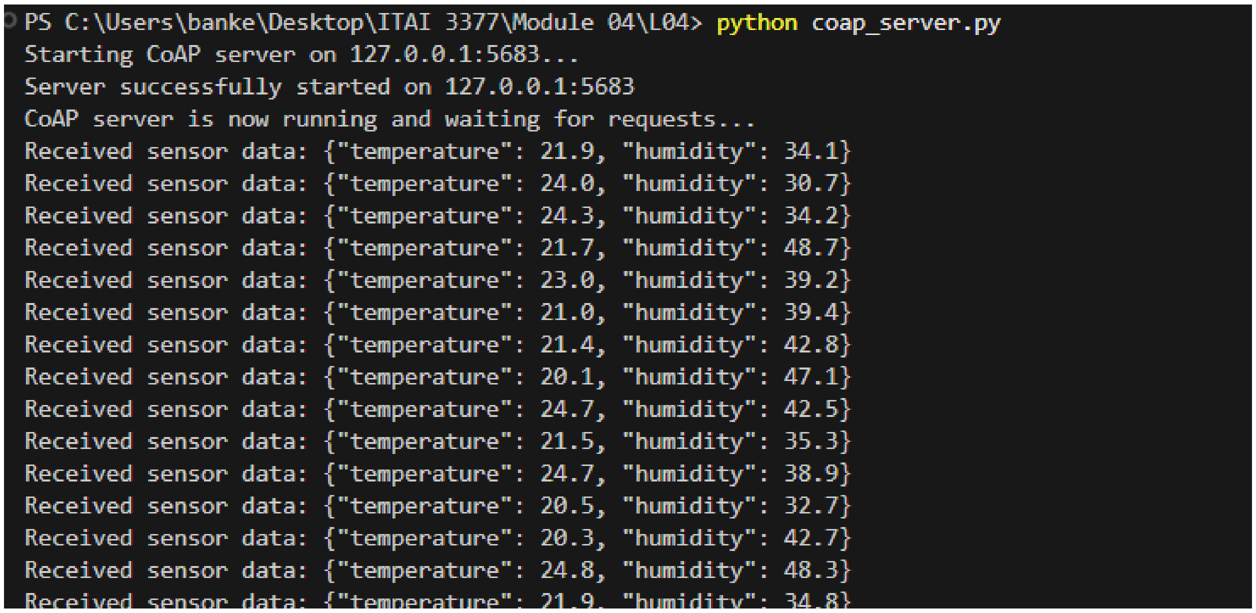
No packages published
Publish your first package

Languages

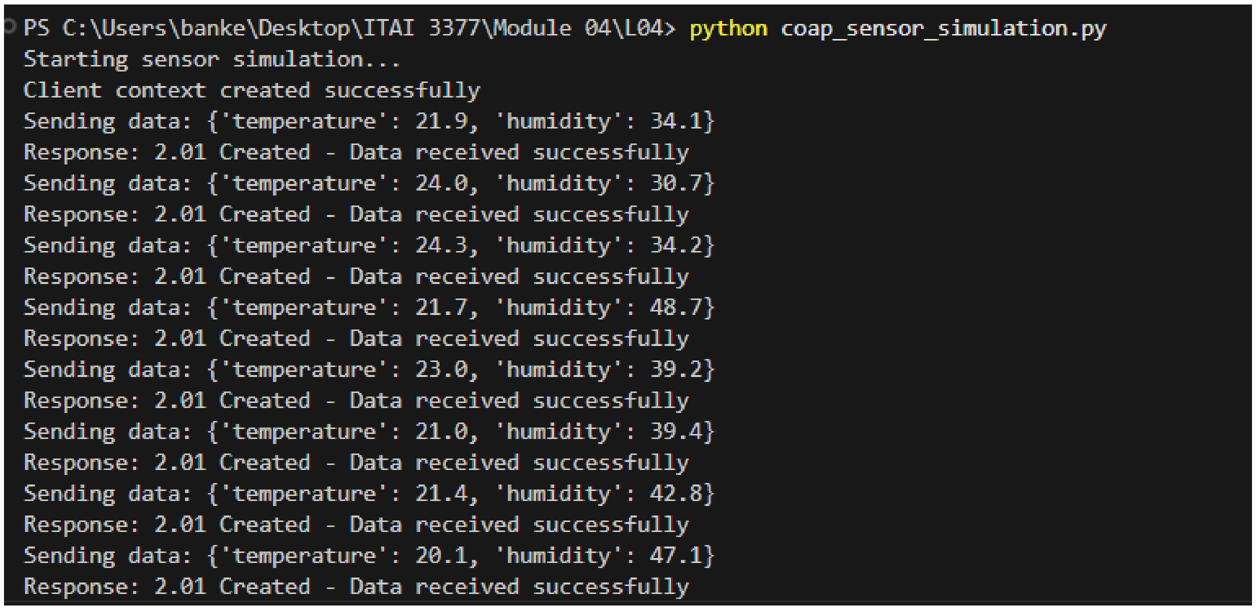
● Python 100.0%



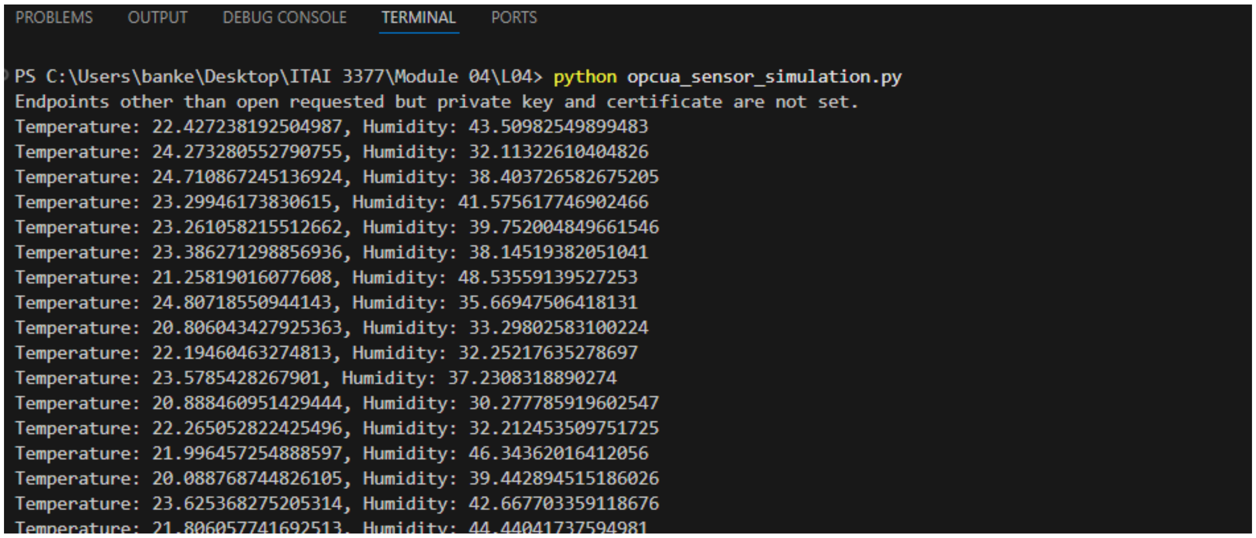
```
PS C:\Users\banke\Desktop\ITAI 3377\Module 04\L04> python mqtt_sensor_simulation.py
C:\Users\banke\Desktop\ITAI 3377\Module 04\L04\mqtt_sensor_simulation.py:11: Depreca
  client = mqtt.Client()
Published: {"temperature": 23.19, "humidity": 41.52}
Published: {"temperature": 23.49, "humidity": 45.21}
Published: {"temperature": 23.30, "humidity": 39.27}
Published: {"temperature": 22.23, "humidity": 37.07}
Published: {"temperature": 20.27, "humidity": 32.67}
Published: {"temperature": 20.29, "humidity": 33.99}
Published: {"temperature": 24.74, "humidity": 38.92}
Published: {"temperature": 21.26, "humidity": 41.18}
Published: {"temperature": 20.65, "humidity": 46.39}
Published: {"temperature": 22.72, "humidity": 36.67}
Published: {"temperature": 24.21, "humidity": 35.79}
Published: {"temperature": 23.23, "humidity": 46.72}
Published: {"temperature": 23.06, "humidity": 41.50}
Published: {"temperature": 22.32, "humidity": 37.06}
Published: {"temperature": 21.51, "humidity": 45.99}
Published: {"temperature": 20.10, "humidity": 42.80}
```



```
PS C:\Users\banke\Desktop\ITAI 3377\Module 04\L04> python coap_server.py
Starting CoAP server on 127.0.0.1:5683...
Server successfully started on 127.0.0.1:5683
CoAP server is now running and waiting for requests...
Received sensor data: {"temperature": 21.9, "humidity": 34.1}
Received sensor data: {"temperature": 24.0, "humidity": 30.7}
Received sensor data: {"temperature": 24.3, "humidity": 34.2}
Received sensor data: {"temperature": 21.7, "humidity": 48.7}
Received sensor data: {"temperature": 23.0, "humidity": 39.2}
Received sensor data: {"temperature": 21.0, "humidity": 39.4}
Received sensor data: {"temperature": 21.4, "humidity": 42.8}
Received sensor data: {"temperature": 20.1, "humidity": 47.1}
Received sensor data: {"temperature": 24.7, "humidity": 42.5}
Received sensor data: {"temperature": 21.5, "humidity": 35.3}
Received sensor data: {"temperature": 24.7, "humidity": 38.9}
Received sensor data: {"temperature": 20.5, "humidity": 32.7}
Received sensor data: {"temperature": 20.3, "humidity": 42.7}
Received sensor data: {"temperature": 24.8, "humidity": 48.3}
Received sensor data: {"temperature": 21.9, "humidity": 34.8}
```



```
PS C:\Users\banke\Desktop\ITAI 3377\Module 04\L04> python coap_sensor_simulation.py
Starting sensor simulation...
Client context created successfully
Sending data: {'temperature': 21.9, 'humidity': 34.1}
Response: 2.01 Created - Data received successfully
Sending data: {'temperature': 24.0, 'humidity': 30.7}
Response: 2.01 Created - Data received successfully
Sending data: {'temperature': 24.3, 'humidity': 34.2}
Response: 2.01 Created - Data received successfully
Sending data: {'temperature': 21.7, 'humidity': 48.7}
Response: 2.01 Created - Data received successfully
Sending data: {'temperature': 23.0, 'humidity': 39.2}
Response: 2.01 Created - Data received successfully
Sending data: {'temperature': 21.0, 'humidity': 39.4}
Response: 2.01 Created - Data received successfully
Sending data: {'temperature': 21.4, 'humidity': 42.8}
Response: 2.01 Created - Data received successfully
Sending data: {'temperature': 20.1, 'humidity': 47.1}
Response: 2.01 Created - Data received successfully
```



```
PS C:\Users\banke\Desktop\ITAI 3377\Module 04\L04> python opcua_sensor_simulation.py
Endpoints other than open requested but private key and certificate are not set.
Temperature: 22.427238192504987, Humidity: 43.50982549899483
Temperature: 24.273280552790755, Humidity: 32.11322610404826
Temperature: 24.710867245136924, Humidity: 38.403726582675205
Temperature: 23.29946173830615, Humidity: 41.575617746902466
Temperature: 23.261058215512662, Humidity: 39.752004849661546
Temperature: 23.386271298856936, Humidity: 38.14519382051041
Temperature: 21.25819016077608, Humidity: 48.53559139527253
Temperature: 24.80718550944143, Humidity: 35.66947506418131
Temperature: 20.806043427925363, Humidity: 33.29802583100224
Temperature: 22.19460463274813, Humidity: 32.25217635278697
Temperature: 23.5785428267901, Humidity: 37.2308318890274
Temperature: 20.888460951429444, Humidity: 30.277785919602547
Temperature: 22.265052822425496, Humidity: 32.212453509751725
Temperature: 21.996457254888597, Humidity: 46.34362016412056
Temperature: 20.088768744826105, Humidity: 39.442894515186026
Temperature: 23.625368275205314, Humidity: 42.667703359118676
Temperature: 21.806057741692513, Humidity: 44.44041737594981
```

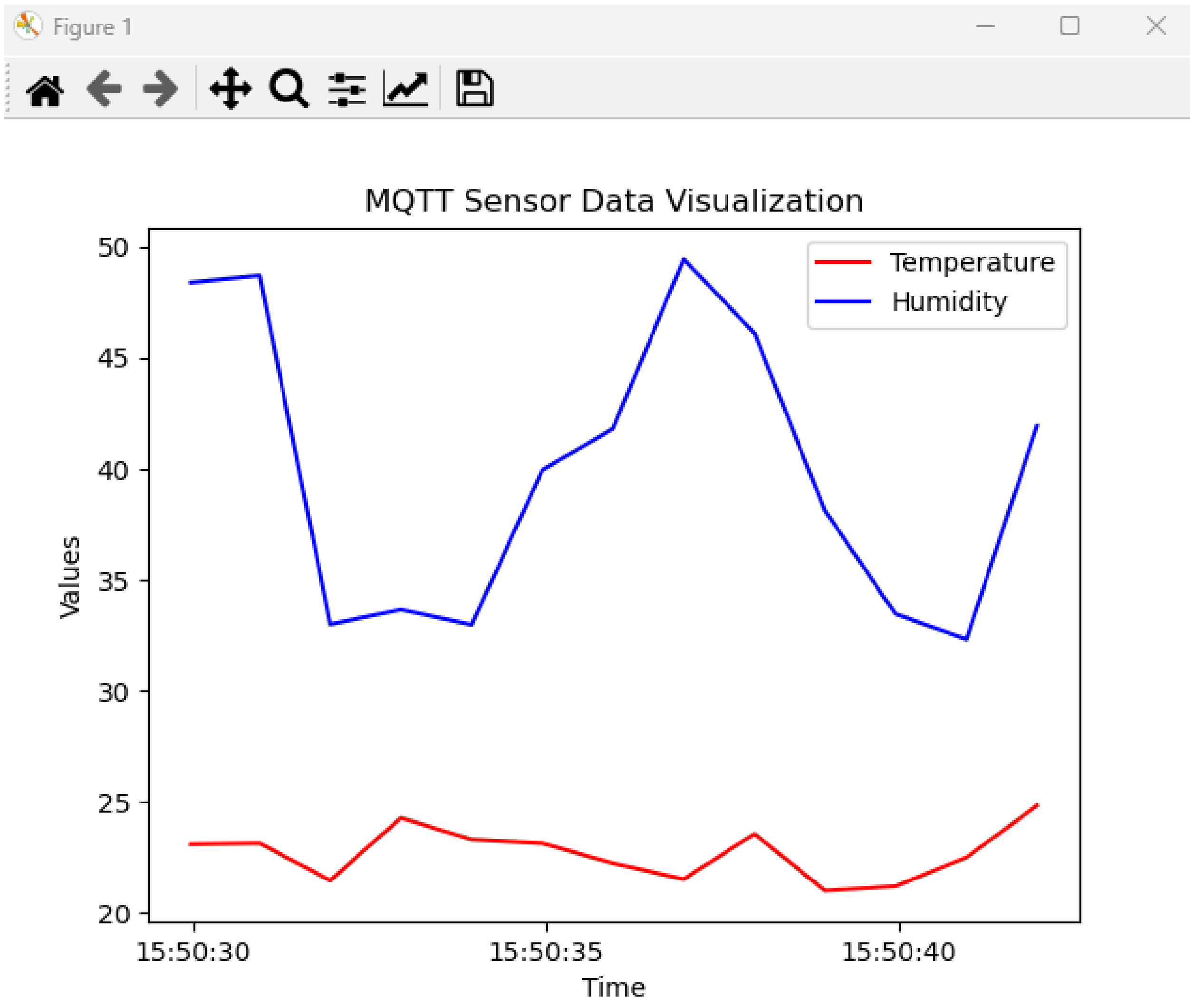


Figure 1
MQTT Sensor Data Visualization
Temperature
Humidity
Values
Time
50
45
40
35
30
25
20
15:50:30
15:50:35
15:50:40

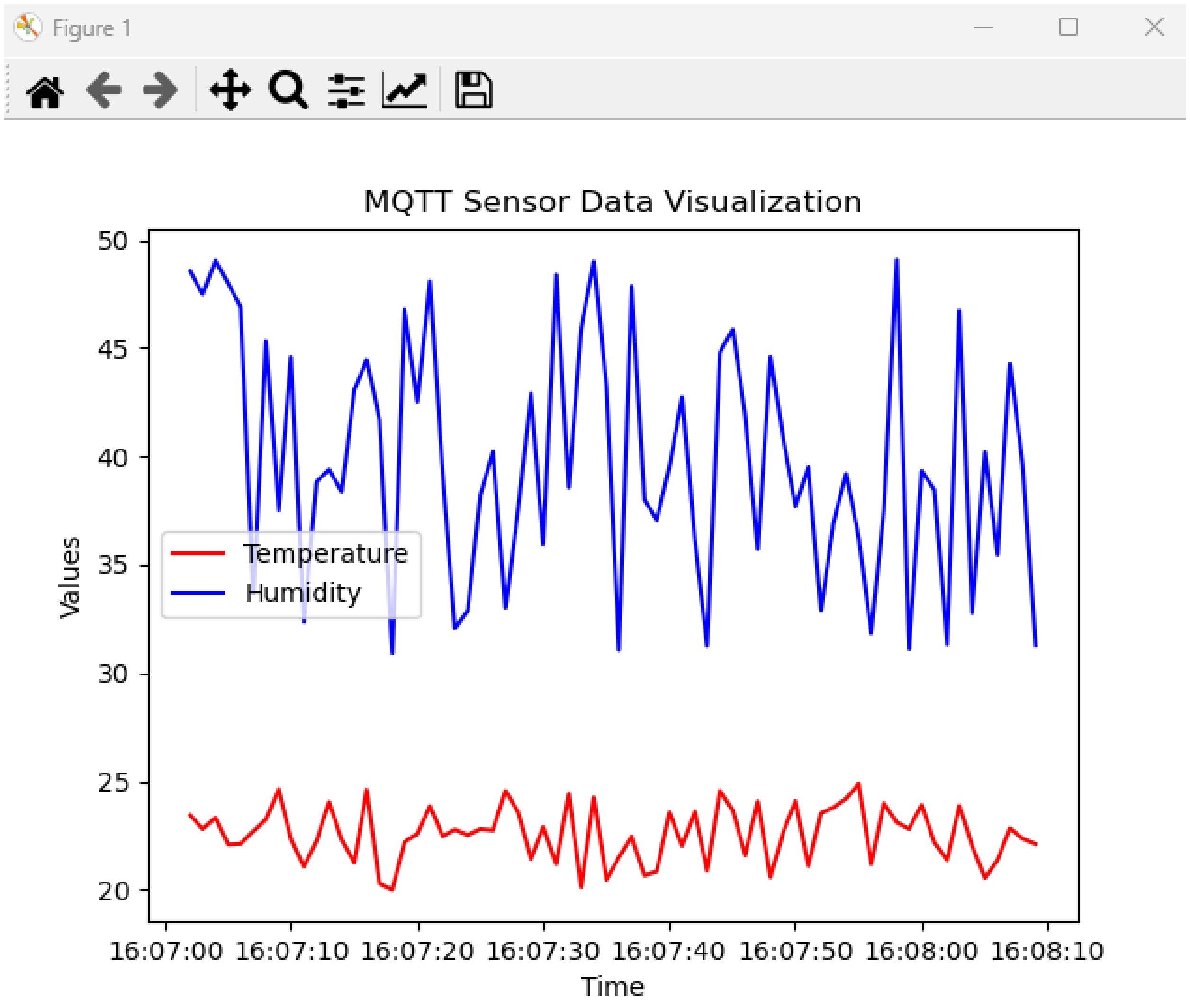


Figure 1
MQTT Sensor Data Visualization
50
45
40
35
30
25
20
Values
Temperature
Humidity
16:07:00
16:07:10
16:07:20
16:07:30
16:07:40
16:07:50
16:08:00
16:08:10
Time

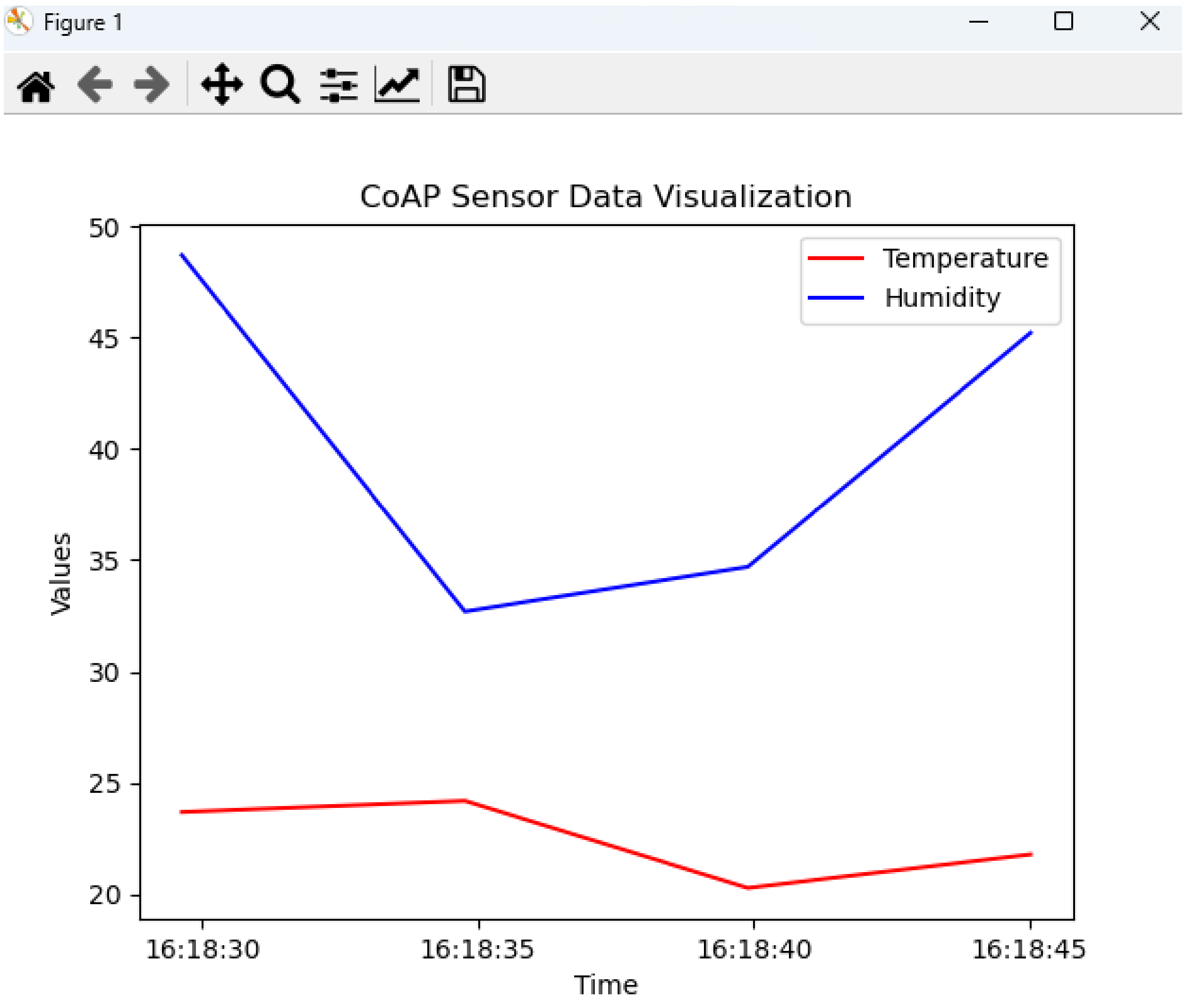


Figure 1
CoAP Sensor Data Visualization
Temperature
Humidity
Values
Time
50
45
40
35
30
25
20
16:18:30
16:18:35
16:18:40
16:18:45

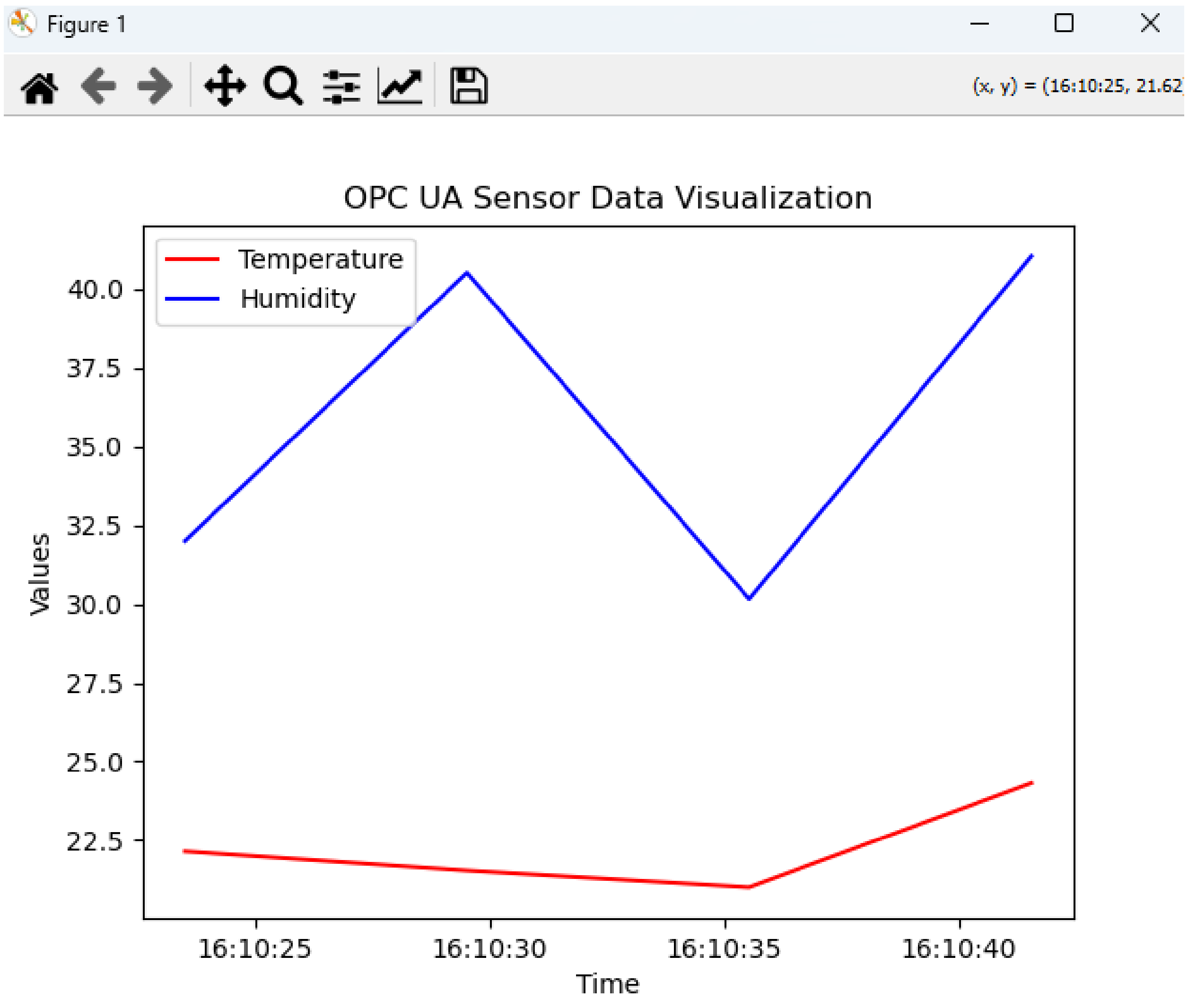


Figure 1
(x, y) = (16:10:25, 21.62
OPC UA Sensor Data Visualization
Temperature
Humidity
Values
40.0
37.5
35.0
32.5
30.0
27.5
25.0
22.5
16:10:25
16:10:30
16:10:35
16:10:40
Time

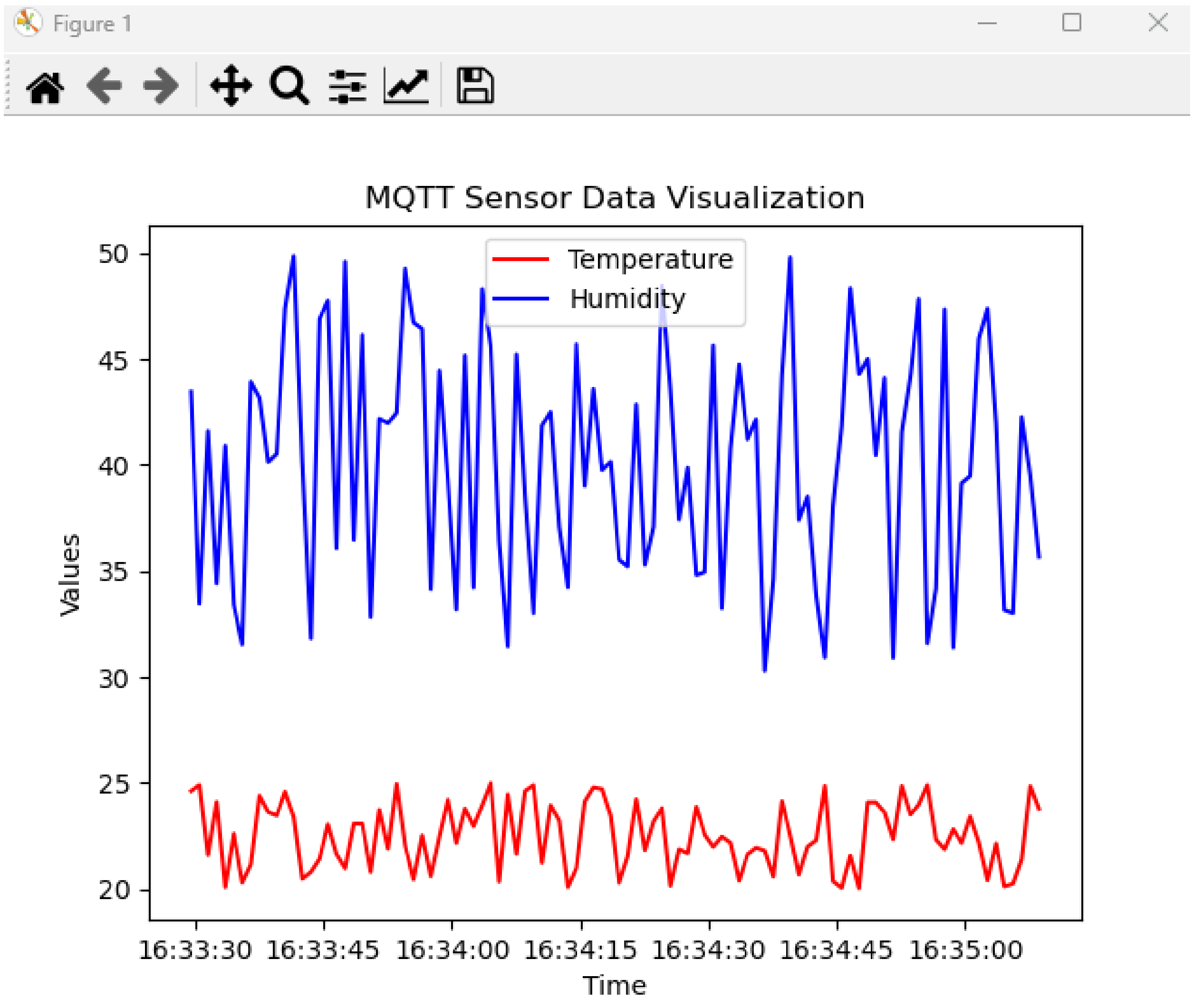


Figure 1
MQTT Sensor Data Visualization
Temperature
Humidity
Values
Time
50
45
40
35
30
25
20
16:33:30
16:33:45
16:34:00
16:34:15
16:34:30
16:34:45
16:35:00

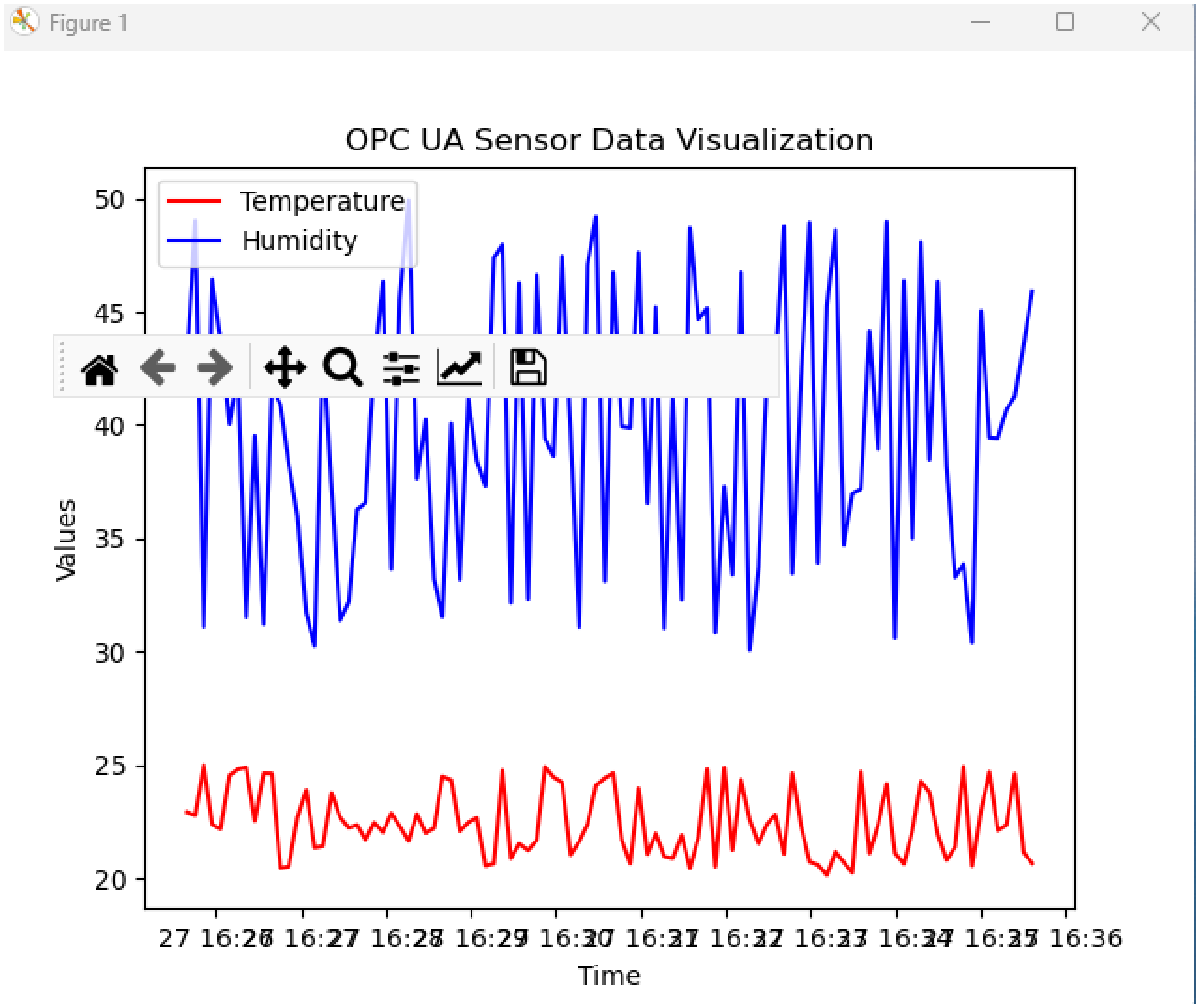


Figure 1
OPC UA Sensor Data Visualization
Temperature
Humidity
Values
Time
50
45
40
35
30
25
20
27 16:26
16:27
16:28
16:29
16:30
16:31
16:32
16:33
16:34
16:35
16:36

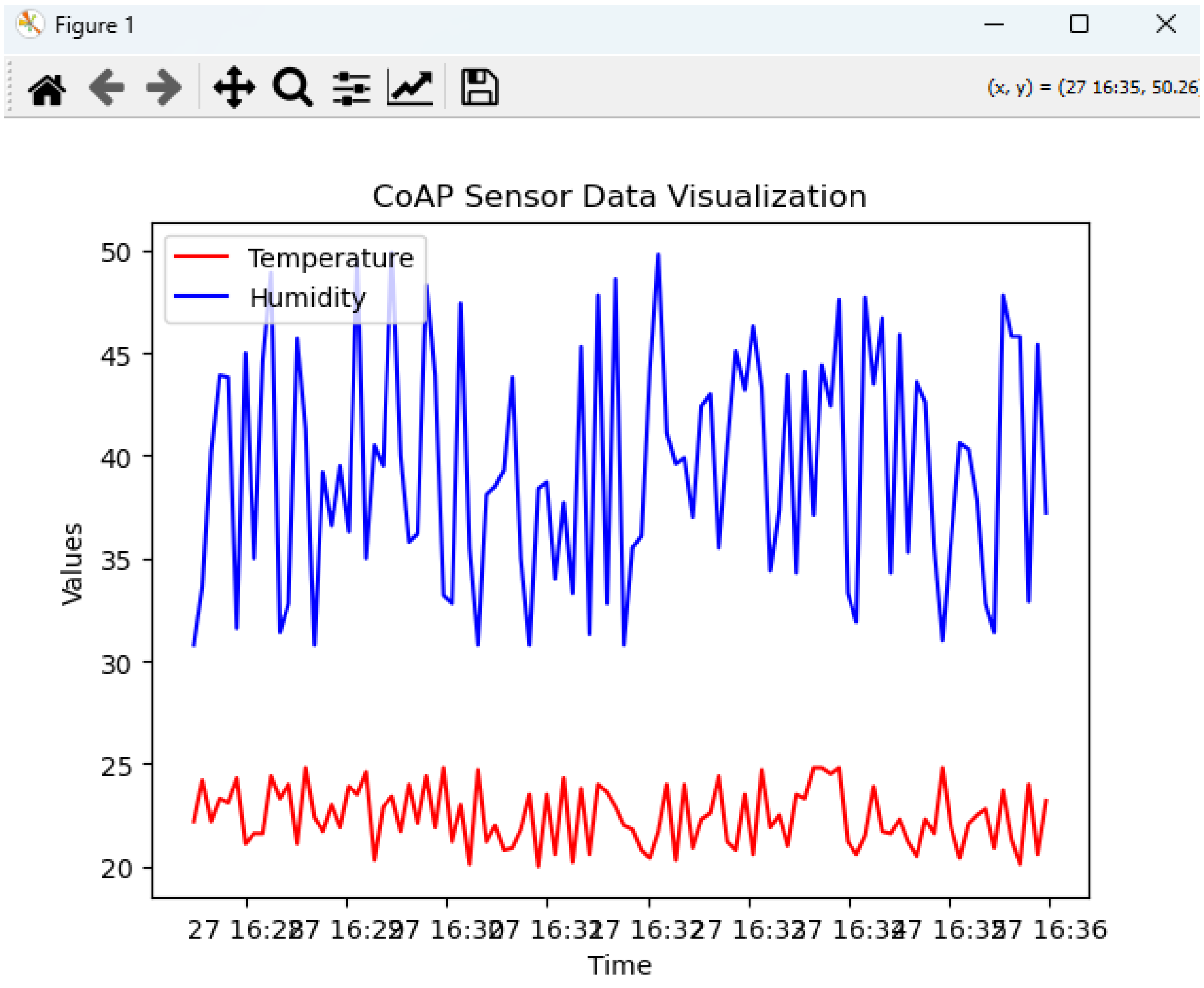


Figure 1
(x, y) = (27 16:35, 50.26)
CoAP Sensor Data Visualization
Temperature
Humidity
Values
50
45
40
35
30
25
20
27 16:28
27 16:29
27 16:30
27 16:31
27 16:32
27 16:33
27 16:34
27 16:35
27 16:36
Time