

**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»**

**Навчально-науковий інститут атомної та теплової енергетики  
Кафедра цифрових технологій в енергетиці**

## **ЗВІТ**

**з лабораторної роботи №7**

**з дисципліни «Розробка застосунків інтернету речей та  
сенсорних мереж»**

**Тема: «Вивчення використання протоколів IoT, зокрема  
MQTT»**

**Варіант №17**

**Виконав:**

**Студент групи ТР-12**

**Ковальов Олександр Олексійович**

**Дата здачі: 09.03.2025**

**Мета роботи.** Дослідити застосування тестових клієнтів для взаємодії через MQTT, налаштувати взаємодію Node-RED з іншими пристроями за допомогою MQTT, забезпечити спільну роботу MQTT-клієнта з мобільним телефоном, використати знання отримані в результаті дослідження для вирішення практичних завдань.

**Індивідуальне завдання:**

1. Провести експерименти з використанням тестових клієнтів для спілкування через MQTT;
2. Налаштувати взаємодію Node-RED з іншими пристроями через протокол MQTT;
3. Забезпечити взаємодію між MQTT-клієнтом та мобільним телефоном.

**Хід роботи.**

Було проведене підключення з тестового веб-клієнту hivemq до тестового брокера mosquitto. Але, не вдалося створити незашифроване підключення, через що довелося змінити порт на 8081 та увімкнути SSL.

**Connection** connected

Host: test.mosquitto.org Port: 8081 ClientID: clientId-SluAKkZFST Disconnect

Username: Password: Keep Alive: 60 SSL: ☒ Clean Session: ☒

Last-Will Topic: Last-Will QoS: 0 Last-Will Retain: ☐

Last-Will Message:

Було опубліковане повідомлення:

**Connection** connected

**Publish**

Topic: temp/random QoS: 1 Retain: ☐ Publish

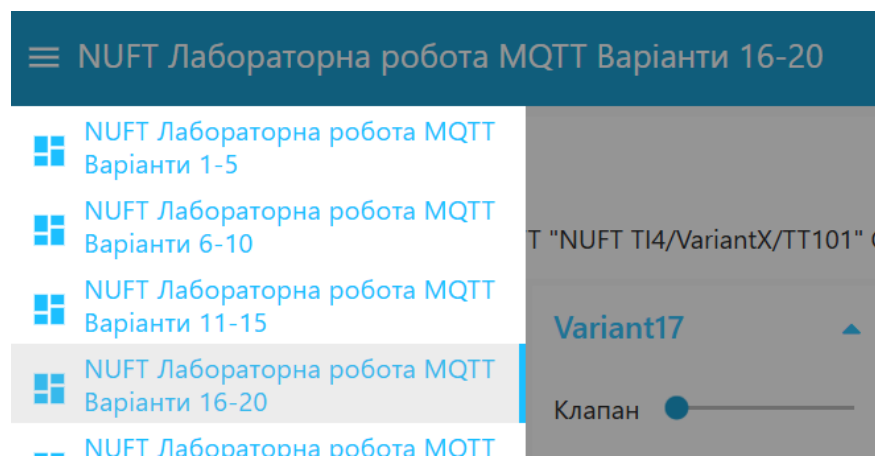
Message: 31.30

**Subscriptions** Add New Topic Subscription

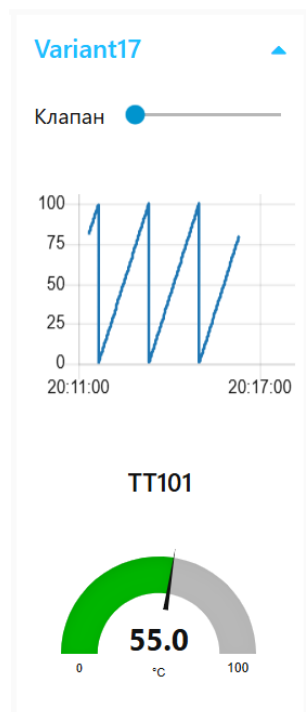
Отримане значення:



Після цього потрібно перейти на інший тестовий сервер, і обрати вкладку за своїм варіантом (17):



Панель для варіанту:

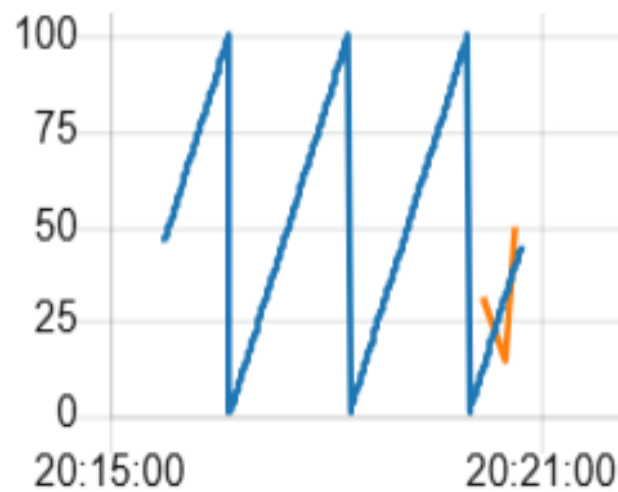


Були надіслані 3 значення з веб-клієнту, що видно на графіку:

## Variant17



Клапан ☒



Додамо підписку на клапан:

● connected



## Subscriptions



Add New Topic Subscription

Qos: 1



NUFT TI4/Variant17/...

Отримані повідомлення:

| Messages |                     |                                  |        | ⌵ |
|----------|---------------------|----------------------------------|--------|---|
|          | 2025-03-08 20:23:59 | Topic: NUFT TI4/Variant17/Klapan | Qos: 1 |   |
| 30       |                     |                                  |        |   |
|          | 2025-03-08 20:23:58 | Topic: NUFT TI4/Variant17/Klapan | Qos: 1 |   |
| 63       |                     |                                  |        |   |
|          | 2025-03-08 20:23:58 | Topic: NUFT TI4/Variant17/Klapan | Qos: 1 |   |
| 64       |                     |                                  |        |   |
|          | 2025-03-08 20:23:56 | Topic: NUFT TI4/Variant17/Klapan | Qos: 1 |   |
| 18       |                     |                                  |        |   |
|          | 2025-03-08 20:23:55 | Topic: NUFT TI4/Variant17/Klapan | Qos: 1 |   |
| 8        |                     |                                  |        |   |

Підписка на всі топіки:

● connected

⌵

Subscriptions

⌵

Add New Topic Subscription

Qos: 1

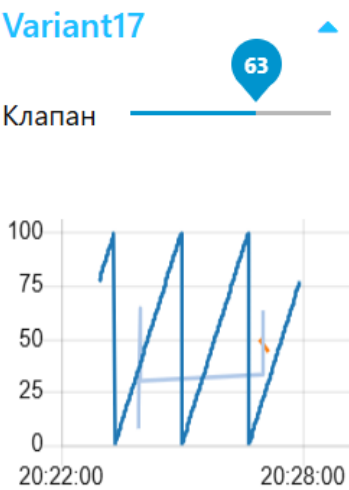
NUFT TI4/Variant17/#

X

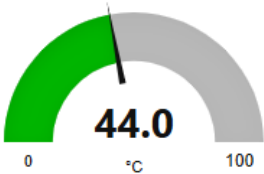
Повідомлення:

| Messages |                     |                                  |        | ⌵        |
|----------|---------------------|----------------------------------|--------|----------|
|          | 2025-03-08 20:27:10 | Topic: NUFT TI4/Variant17/TT101  | Qos: 1 |          |
| 44       |                     |                                  |        |          |
|          | 2025-03-08 20:27:01 | Topic: NUFT TI4/Variant17/Klapan | Qos: 1 |          |
| 63       |                     |                                  |        |          |
|          | 2025-03-08 20:27:01 | Topic: NUFT TI4/Variant17/Klapan | Qos: 1 |          |
| 35       |                     |                                  |        |          |
|          | 2025-03-08 20:27:00 | Topic: NUFT TI4/Variant17/Klapan | Qos: 1 |          |
| 33       |                     |                                  |        |          |
|          | 2025-03-08 20:26:56 | Topic: NUFT TI4/Variant17/TT101  | Qos: 1 |          |
| 50       |                     |                                  |        |          |
|          | 2025-03-08 20:26:30 | Topic: NUFT TI4/Variant17/TT101  | Qos: 0 | Retained |
| 55       |                     |                                  |        |          |

Панель:



ТТ101



Підписка на всі клапани, зміна значень:

### Connection

connected

### Publish

Topic: NUFT TI4/Variant17/TT101

QoS: 1

Retain: ☐

Message: 44

Publish

### Subscriptions

Add New Topic Subscription

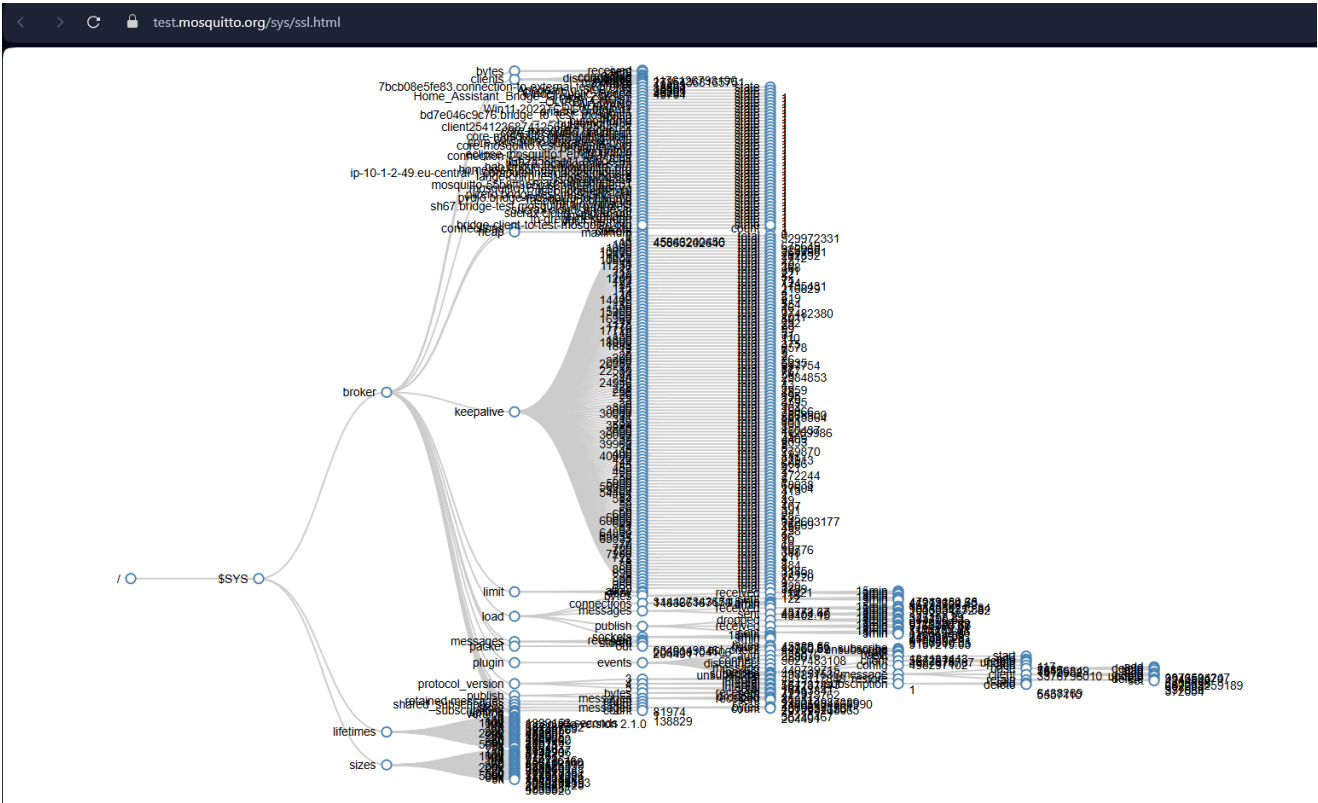
Qos: 1

NUFT TI4/+/Klapan

### Messages

|                     |                                  |        |    |
|---------------------|----------------------------------|--------|----|
| 2025-03-08 20:29:30 | Topic: NUFT TI4/Variant20/Klapan | Qos: 1 | 87 |
| 2025-03-08 20:29:29 | Topic: NUFT TI4/Variant20/Klapan | Qos: 1 | 48 |
| 2025-03-08 20:29:29 | Topic: NUFT TI4/Variant19/Klapan | Qos: 1 | 0  |
| 2025-03-08 20:29:28 | Topic: NUFT TI4/Variant19/Klapan | Qos: 1 | 44 |

Далі треба ознайомитись зі системними темами брокера mosquitto, але звичайний сайт не працює. Треба обов'язково під'єднуватись за допомогою TLS (<https://test.mosquitto.org/sys/ssl.html>):



Підписка на аптайм виконана:

Connection

connected

Publish

Topic

QoS

1

Retain

Publish

Message

Subscriptions

Add New Topic Subscription

Qos: 1

\$SYS/broker/uptime

X

Messages

2025-03-08 20:46:40

Topic: \$SYS/broker/uptime

Qos: 1

1339383 seconds

2025-03-08 20:46:31

Topic: \$SYS/broker/uptime

Qos: 1

Retained

1339373 seconds

Був встановлений node-red:

```
C:\Users\Alex>npm install -g --unsafe-perm node-red

added 311 packages in 14s

63 packages are looking for funding
  run `npm fund` for details

npm notice
npm notice New major version of npm available! 10.9.2 -> 11.2.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.2.0
npm notice To update run: npm install -g npm@11.2.0
npm notice
```

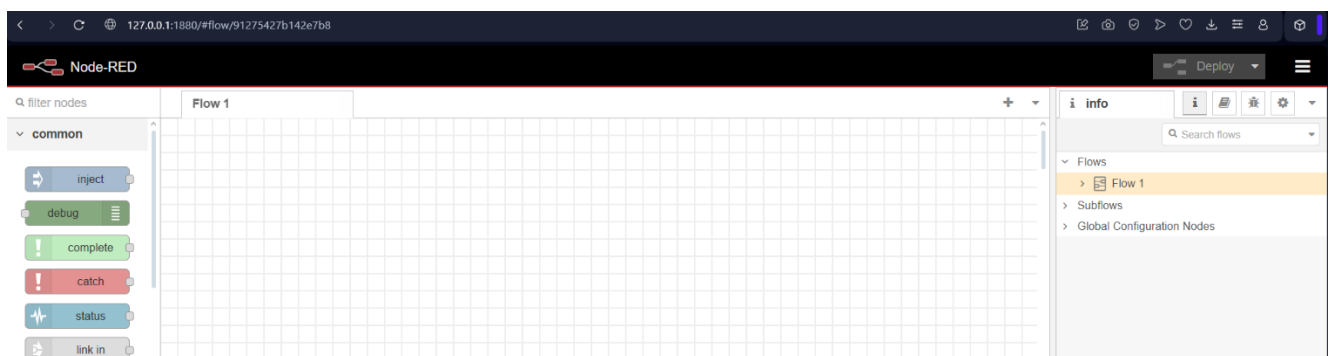
Запуск:

```
C:\Users\Alex>node-red
8 Mar 21:45:50 - [info]

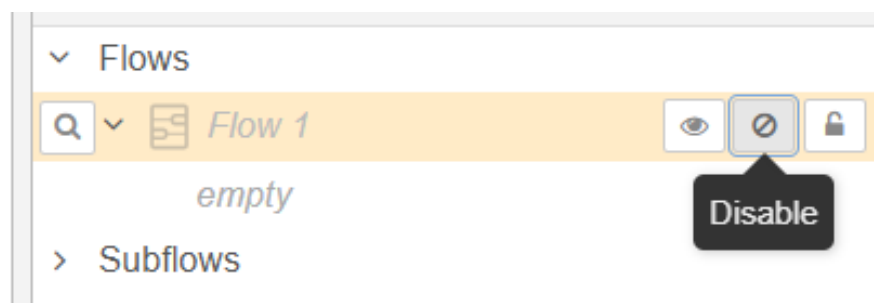
Welcome to Node-RED
=====

8 Mar 21:45:50 - [info] Node-RED version: v4.0.9
8 Mar 21:45:50 - [info] Node.js version: v22.13.0
8 Mar 21:45:50 - [info] Windows_NT 10.0.19045 x64 LE
8 Mar 21:45:51 - [info] Loading palette nodes
8 Mar 21:45:51 - [info] Settings file   : C:\Users\Alex\.node-red\settings.js
```

Результат:



Був відключений потік:





Налаштована нода mosquitto:

**Edit mqtt in node**

Delete

Cancel

Done

**Properties**

Server

mosquitto

Action

Subscribe to single topic

Topic

NUFT TI4/Variant17/#

QoS

1

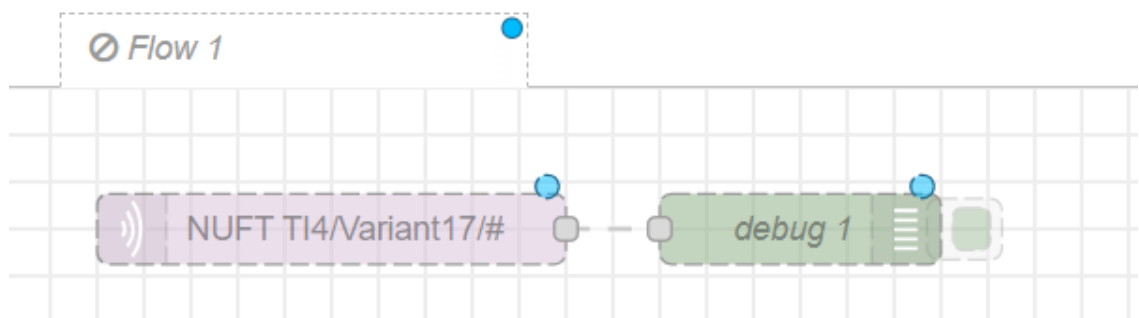
Output

auto-detect (parsed JSON object, string or buff

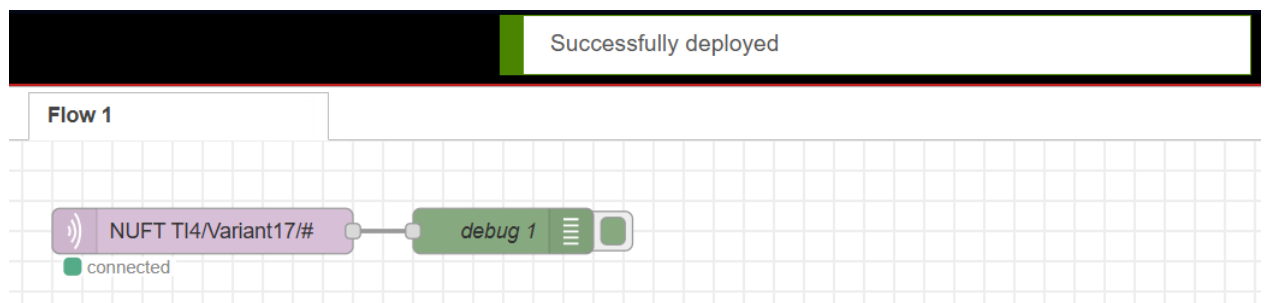
Name

Name

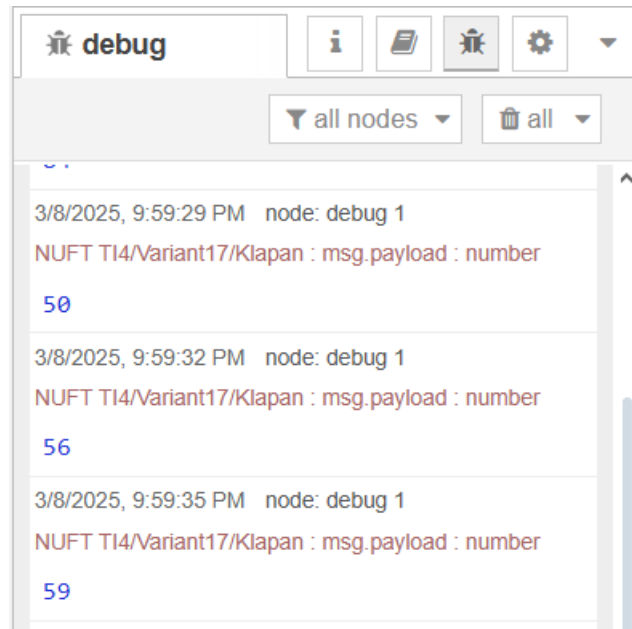
Додана нода «Debug»:



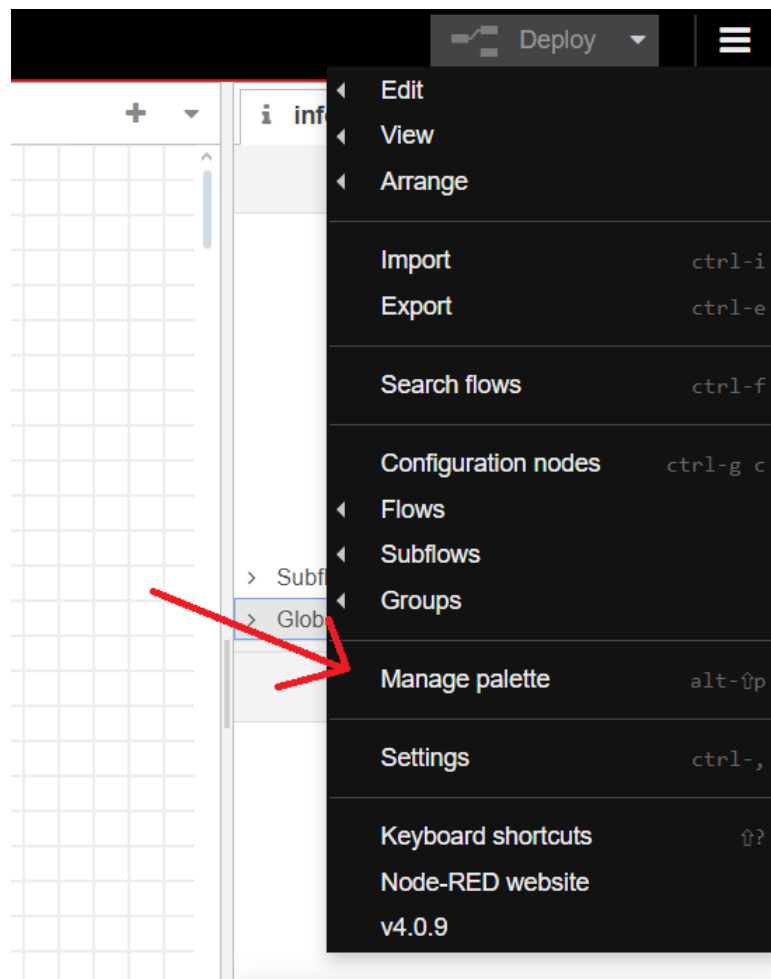
Деплой:



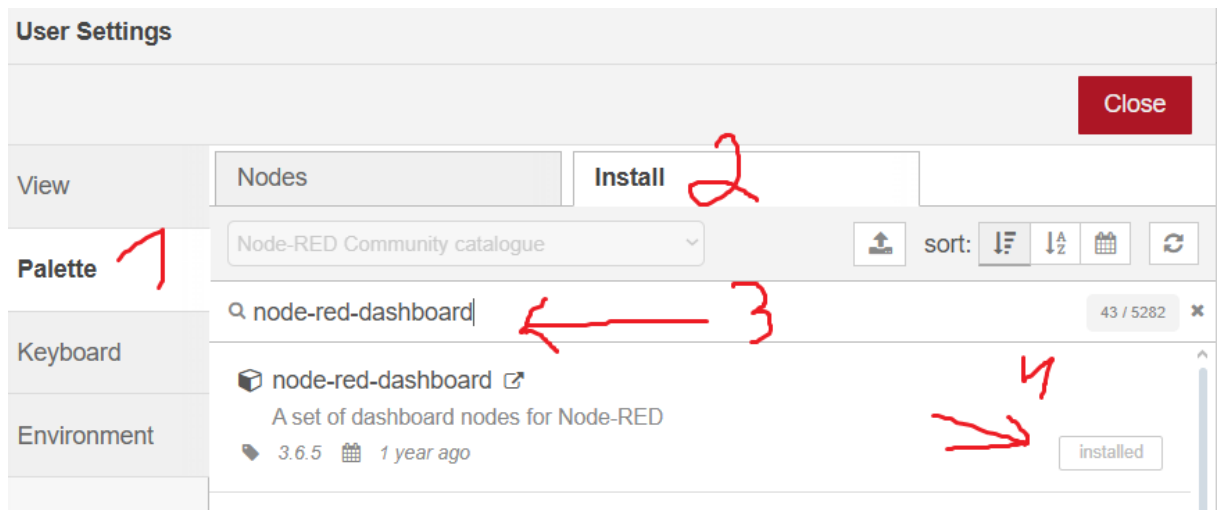
Відображення зміни значень:



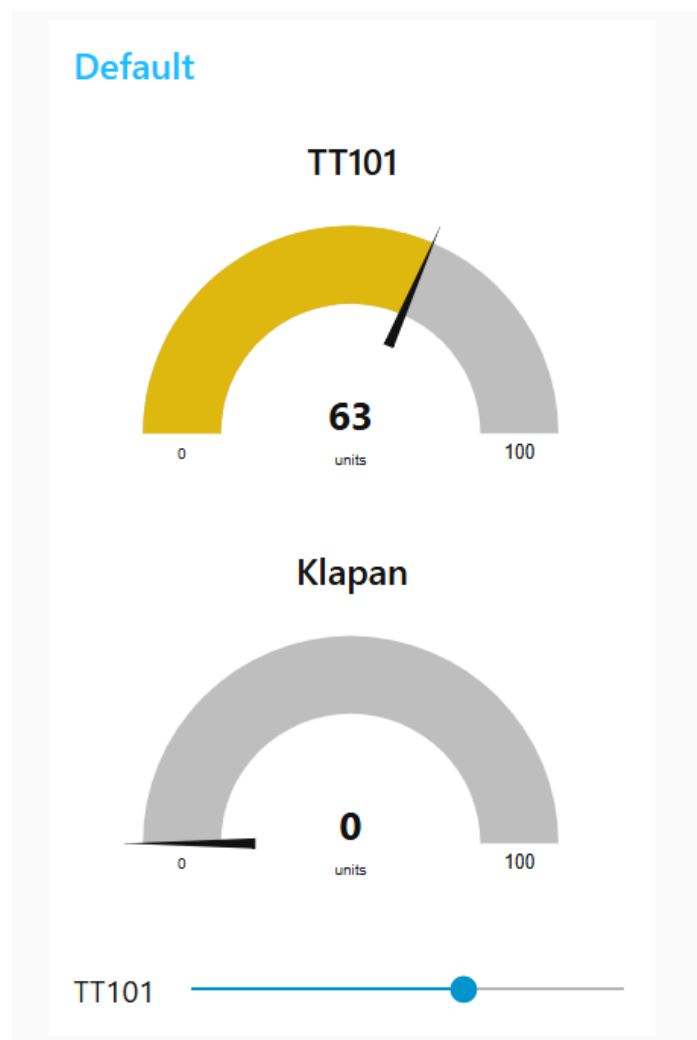
Після цього, були налаштовані ноди «датчик» та «слайдер». Але вони недоступні за замовчуванням, для цього треба встановити певне розширення. Це можна зробити, якщо натиснути на три полоски, потім «Manage palette»:



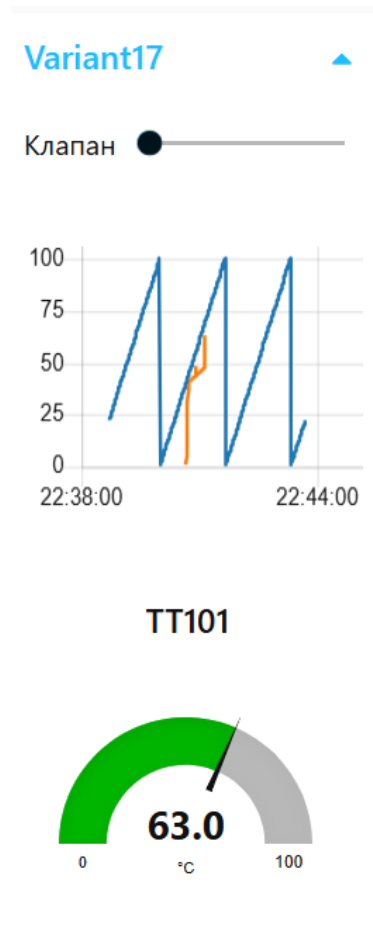
І після цього встановити розширення «node-red-dashboard»:



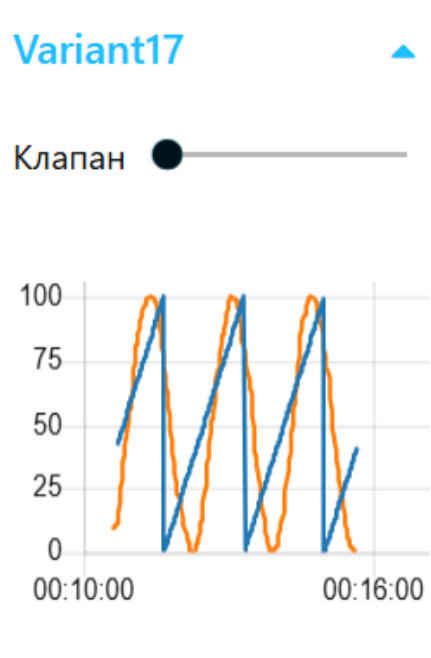
Результат можна побачити, якщо перейти за певним ендпоінтом під назвою ui.  
Посилання: <http://localhost:1880/ui/>. Порт в посиланні потрібно змінити на той, на якому заведений сервер локально.



Відповідно, при зміні значень слайдера, результат відображається й на тестовому сервері:



Далі був налаштований вибір – синусоїда чи косинусоїда. Можна обирати функцію для температури. Результат для косинусоїди:

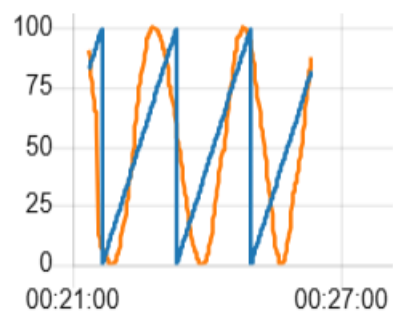


Результат для синусоїди:

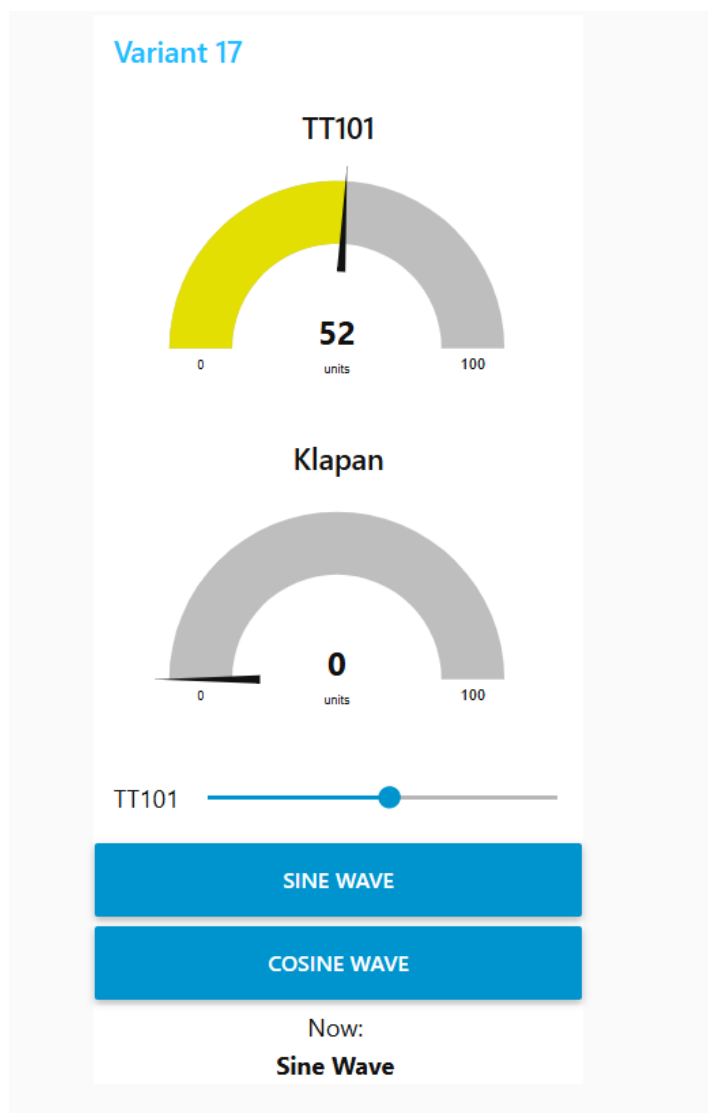
Variant17



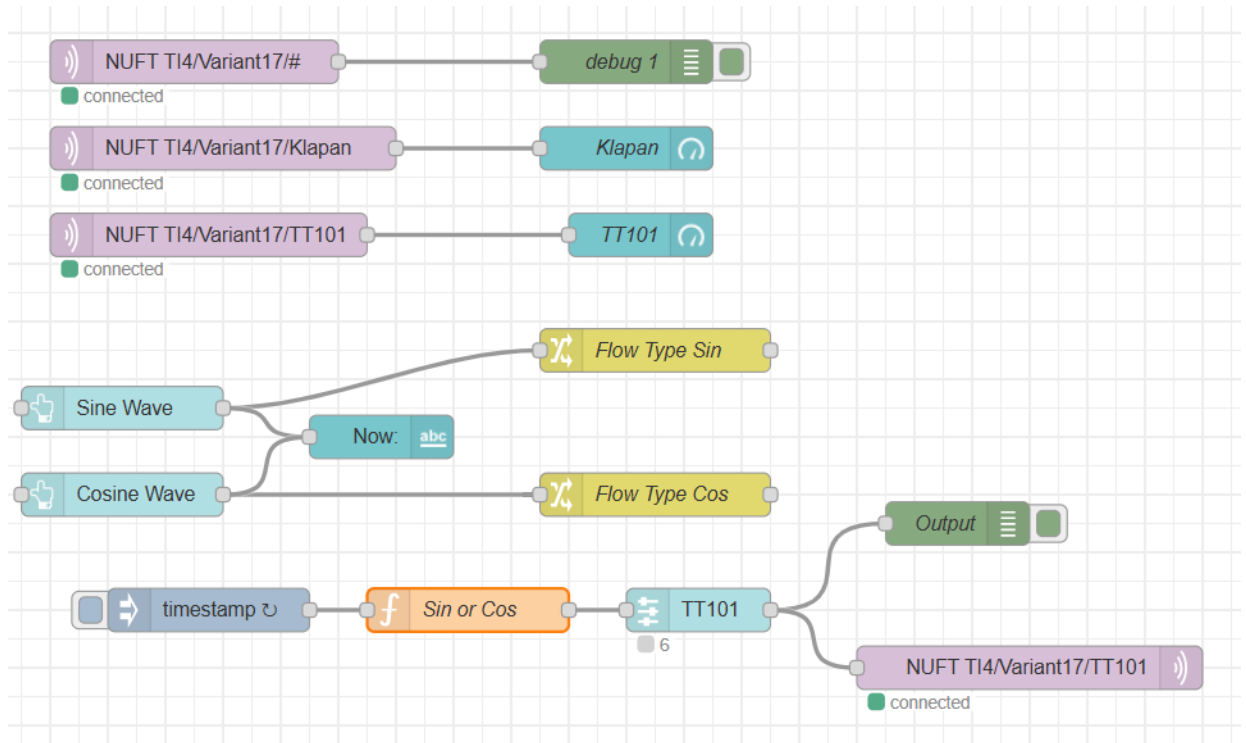
Клапан ☒



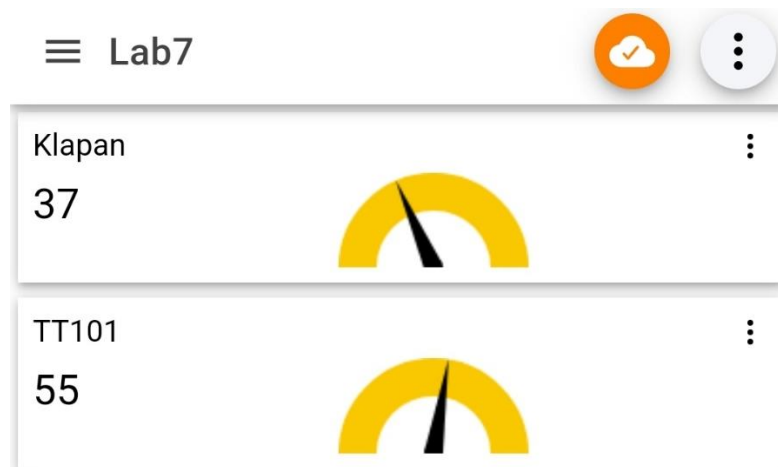
Інтерфейс:



## Налаштовані ноди:



І насамкінець, за допомогою додатку «IoT MQTT Panel» було здійснене підключення до тестового серверу. Значення були синхронізовані, з'єднання протестоване. Результат – на скріншоті.



**Висновок:** У процесі виконання роботи було досліджено застосування тестових клієнтів для взаємодії через MQTT, а також налаштування взаємодії Node-RED з іншими пристроями за допомогою цього протоколу. Зокрема, проведено експерименти, що дозволили глибше зрозуміти механізми обміну повідомленнями між пристроями, а також налаштування ефективної комунікації між MQTT-клієнтами та мобільними телефонами. У результаті виконаних завдань було здобуто практичні знання, які можна використовувати для вирішення різноманітних задач у сфері автоматизації та Інтернету речей, зокрема в контексті інтеграції різних пристроїв та платформ через MQTT.

## Контрольні питання:

1. Як IBM Websphere Message Queue допоміг вирішити проблеми, які виникали в незалежних та не конкуруючих розподілених системах?

IBM Websphere Message Queue допоміг вирішити проблеми в незалежних та не конкуруючих розподілених системах завдяки впровадженню надійного механізму передачі повідомлень. Це дозволило забезпечити гарантовану доставку даних між різними компонентами системи навіть за умов збоїв або непередбачених ситуацій, таких як відключення мережі чи тимчасова недоступність деяких пристроїв. В результаті система стала більш стійкою до помилок, а з'єднання між компонентами залишалося стабільним, що дозволяло знизити ризики втрати або пошкодження інформації.

2. На яких фундаментальних принципах базується MQTT, і чому він вважається оптимальним для застосувань в сфері Інтернету речей (IoT)?

MQTT базується на фундаментальних принципах публікації та підписки (publish/subscribe), що дозволяє здійснювати ефективний обмін даними між різними пристроями в мережі. Цей протокол є оптимальним для IoT, оскільки він дуже легкий, не вимагає великих ресурсів і дозволяє забезпечити ефективний обмін даними навіть при обмеженій пропускну здатності або нестабільних з'єднаннях. MQTT також підтримує надійне доставлення повідомлень, що важливо для пристроїв, які працюють в умовах низької доступності мережі.

3. Які інтернет-протоколи MQTT використовує для забезпечення організованого та двостороннього обміну даними?

MQTT використовує два основні інтернет-протоколи для забезпечення організованого та двостороннього обміну даними: TCP/IP для забезпечення надійного з'єднання між клієнтами та брокером, а також WebSocket для підтримки двостороннього зв'язку через HTTP. Це дозволяє пристроям передавати дані в реальному часі, забезпечуючи мінімальні затримки та підтримуючи стабільний зв'язок навіть за умов частих змін мережевих умов.

4. Які ключові компоненти складають систему MQTT і які функції виконують клієнти, брокери та шаблон публікації-підписки?

Система MQTT складається з трьох основних компонентів: клієнтів, брокерів і шаблону публікації-підписки. Клієнти є пристроями або програмами, які публікують повідомлення або підписуються на певні теми для отримання даних. Брокер – це сервер, який приймає повідомлення від публікуючих клієнтів і пересилає їх підписаним клієнтам. Шаблон публікації-підписки визначає, як клієнти взаємодіють між собою: один клієнт публікує дані на певний топик, а інші підписуються на цей топик, отримуючи оновлення.

5. Які існують рівні якості обслуговування (QoS) в MQTT і в чому полягають їх основні відмінності?

В MQTT існують три рівні якості обслуговування (QoS):

- QoS 0 – повідомлення доставляється один раз і не повторюється (найшвидший та найменш надійний рівень).
- QoS 1 – повідомлення доставляється принаймні один раз, гарантуючи, що воно буде отримано, але можуть бути дублікати.

- QoS 2 – повідомлення доставляється точно один раз, без дублювання, забезпечуючи найвищу надійність, але й з найбільшими витратами ресурсів.
6. *Які типові застосування використовують різні рівні якості обслуговування (QoS) у MQTT?*

Різні рівні якості обслуговування (QoS) в MQTT використовуються для різних застосувань.

- QoS 0 застосовується для передачі даних, де важлива швидкість, а дублікати повідомлень не є критичними, наприклад, для моніторингу стану датчиків.
  - QoS 1 використовується, коли необхідно гарантувати доставку даних, але дублікати можуть бути прийнятними, наприклад, для оновлення статусу пристроїв.
  - QoS 2 є найбільш підходящим для ситуацій, де важлива точність і доставлення повідомлень лише один раз, наприклад, для управлінських команд або відправлення важливих даних.
7. *Які переваги та особливості реалізації різних рівнів QoS в контексті IoT або M2M комунікацій?*

Переваги та особливості реалізації різних рівнів QoS в контексті IoT або M2M комунікацій включають в себе баланс між ефективністю і надійністю. QoS 0 дозволяє знизити навантаження на мережу та пристрої, що важливо в умовах обмежених ресурсів або великої кількості пристроїв. QoS 1 дає більш надійне доставлення без сильної втрати продуктивності, що корисно для збору даних. QoS 2 дає найвищу надійність, але може бути більш ресурсозатратним, що важливо для передачі критичних команд або даних, де дублікати неприпустимі.