

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Освітній рівень

Мікроконтролерна підсистема керування Інтернетом речей на основі Node-RED

Назва теми

КвРКІ.2001127.20.01.01 ПЗ

Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

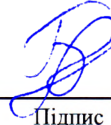
Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Виконав: студент III курсу, група КІ2с-20-1



Підпис

М.Г. Білецький

Ініціали, прізвище

Керівник



Підпис, дата

В.В. Яцків

Ініціали, прізвище

Нормоконтролер

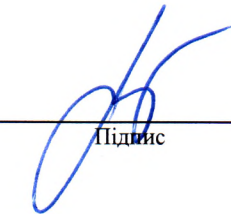


Підпис, дата

С.М. Лисенко

Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних
систем



Підпис

Т.О. Говорущенко

Ініціали, прізвище

« 7 » 06 2023 р.

Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ 11 ” 01 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Білецькому Максиму Геннадійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Мікроконтролерна підсистема керування Інтернетом речей на основі Node-RED

Керівник проекту (роботи) Яцків В.В., д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 1.03.2023 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2023 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Дослідження предметної області та постановка задачі

Моделювання та проектування програмно-технічного модуля керування Інтернетом речей на основі Node-RED

Апаратна реалізація програмно-технічного модуля керування Інтернетом речей на основі Node-RED





5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Структурна схема програмно-технічного модулю на основі Node-RED

Сценарій надсилання сповіщення електронною поштою технікам з обслуговування, коли параметр процесу досягне критичного рівня

Блок схема “Алгоритм надсилання сповіщення електронною поштою”

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КПС		
Антиплагіат	Нічепорук А.О., доцент кафедри КПС		

7. Дата видачі завдання « 11 » 01 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	20.02.2022	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.03.2023	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	10.03.2023	виконано
4	Робота над розділом 2 – моделювання та проектування програмно-технічного модуля керування Інтернетом речей на основі Node-RED	20.04.2023	виконано
5	Робота над розділом 3 – апаратна реалізація програмно-технічного модуля керування Інтернетом речей на основі Node-RED	30.04.2023	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2023	виконано
7	Попередній захист ВКР	26.05.2023	виконано
8	Захист ВКР на засіданні ЕК	Червень 2023 року	

Студент


Підпис

М. Г. Білецький

Ініціали, прізвище

Керівник проекту (роботи)


Підпис

В. В. Яцків

Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний модуль керування Інтернетом речей на основі Node-RED».

Автор роботи: Білецький Максим Геннадійович.

Керівник роботи: Яцків Василь Васильович

Пояснювальна записка: 57 с., 45 рис., 3 дод., 40 джерел.

Графічна частина: 3 креслення.

ПРОГРАМНО-ТЕХНІЧНИЙ, МОДУЛЬ, КЕРУВАННЯ, ІНТЕРНЕТ РЕЧЕЙ, NODE-RED.

Метою роботи є розробка програмно-технічного модуля, який дозволить керувати різноманітними об'єктами Інтернету речей за допомогою платформи Node-RED.

Об'єктом дослідження є процеси керування Інтернету речей (IoT) на основі Node-RED.

Предметом дослідження є програмно-технічний модуль керування Інтернетом речей на базі платформи Node-RED.

Практичне значення полягає в наявності можливості розробки інноваційних рішень в галузі Інтернету речей. Застосування даного модуля дозволить ефективно керувати та моніторити різні IoT-пристрої, забезпечуючи їх взаємодію та координацію з іншими системами.




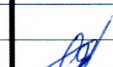
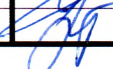

Підпис студента

06.06.2023

Дата

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	3
ВСТУП	4
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ..	6
1.1 Теоретичні аспекти Інтернету речей	6
1.2 Огляд існуючих програмних рішень для керування Інтернетом речей	11
1.3 Аналіз можливостей платформи Node-RED	13
1.4 Постановка задачі	21
1.5 Висновок до розділу	22
2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ	24
2.1 Аналіз функціональних вимог до програмно-технічного засобу	24
2.2 Аналіз нефункціональних вимог до програмно-технічного засобу	25
2.3 Опис архітектури програмно-технічного засобу	27
2.4 Висновок до розділу	36
3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ	38
3.1 Огляд технічних характеристик та вимог до апаратного забезпечення	38
3.2 Проєктування архітектури програмно-технічного засобу	38
3.3 Розробка програмно-технічного засобу на основі Node-RED	44
3.4 Тестування програмного засобу	55
3.5 Висновок до розділу	57
ВИСНОВКИ	59
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	61
ДОДАТОК А Копія креслення «Структурна схема програмно- технічного модулю на основі Node-RED»	
ДОДАТОК Б Копія креслення «Сценарій надсилання сповіщення електронною поштою»	
ДОДАТОК В Копія креслення «Блок схема “ Алоритм надсилання сповіщення електронною поштою”»	

					КвРКІ 2001127.20.01.01 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	Мікроконтролерна підсистема керування Інтернетом речей на основі Node-RED	Літера	Аркуш	Аркушів
Виконав		Білецький М.Г.				у		2
Перевір.		Яцків В.В.						
Н.контр.		Лисенко С.М.						
Затвер.		Говорушенко Т.О.		07.06				ХНУ КІ2с-20-1

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

IoT – Інтернет речей

АСУТП – автоматизовані системи управління технологічними процесами

HTTP – Hypertext Transfer Protocol

CoAP – Constrained Application Protocol

AMQP – Advanced Message Queuing Protocol

					КвРКІ 2001127.20.01.01 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

З кожним днем все більше і більше пристроїв підключається до Інтернету, і це створює величезну кількість можливостей для збору даних та контролю різних пристроїв за допомогою інтернету. Цей процес, відомий як Інтернет речей (IoT), став значним викликом для розробників програмного забезпечення, оскільки вони повинні розробляти програмне забезпечення для різноманітних пристроїв, що працюють на різних платформах.

Ця дипломна робота присвячена розробці програмно-технічного модуля для керування Інтернетом речей на основі Node-RED. Node-RED - це відкрите програмне забезпечення, яке може використовуватися для створення потокових діаграм, що дозволяє швидко та ефективно розробляти програмне забезпечення для Інтернету речей.

Метою цієї дипломної роботи є розробка програмно-технічного засобу для керування різними пристроями Інтернету речей (IoT) на основі платформи Node-RED. Даний засіб має на меті забезпечити зручний та ефективний спосіб керування пристроями IoT через графічний інтерфейс без потреби в глибокому знанні програмування. Крім того, метою є дослідження можливостей та потенціалу платформи Node-RED для розробки програмно-технічних засобів для Інтернету речей.

Завдання дипломної роботи включає наступні пункти:

1. дослідити предметну область Інтернету речей та існуючі рішення у сфері керування IoT;
2. вивчити Node-RED як інструмент для розробки програмного забезпечення у сфері IoT;
3. розробити програмно-технічний модуль для керування IoT на основі Node-RED;
4. реалізувати програмно-технічний модуль та провести його тестування на різних пристроях IoT;

					КвРКІ 2001127.20.01.01 ПЗ	Арк. 4
Зм.	Арк.	№ докум.	Підпис	Дата		

5. оцінити ефективність та можливості розробленого програмно-технічного модуля на основі проведених тестів;

6. розробити документацію до програмно-технічного модуля для подальшого використання в реальних умовах.

У результаті виконання даної дипломної роботи буде розроблений програмно-технічний засіб, що дозволить керувати Інтернетом речей за допомогою Node-RED. Даний засіб буде мати певні переваги порівняно з іншими засобами керування Інтернетом речей, зокрема, високу продуктивність, масштабованість та зручний інтерфейс користувача.

					КвРКІ 2001127.20.01.01 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Теоретичні аспекти Інтернету речей

Інтернет речей (IoT) – це мережа фізичних об'єктів, таких як датчики, пристрої збору даних, пристрої зчитування та інші пристрої, які забезпечують збір та обмін даними між собою та з центральними обчислювальними пристроями, що забезпечують управління цими пристроями. IoT використовується в різних галузях, таких як промисловість, транспорт, охорона здоров'я та побутові умови.

Для забезпечення взаємодії між пристроями Інтернету речей та обробки отриманих даних, необхідна певна програмна платформа. Однією з таких платформ є Node-RED – інструмент потокового програмування, розроблений для зручної розробки додатків Інтернету речей та збору даних. Node-RED забезпечує простоту розробки та розгортання програмних рішень, що використовують дані з IoT-пристроїв, завдяки своїй графічній інтерфейсу, побудованому на базі перетягування та з'єднання вузлів [9].

Архітектура Інтернету речей (IoT) є складною системою, що включає в себе різноманітні компоненти, що взаємодіють між собою. Основною метою архітектури IoT є забезпечення безперервної взаємодії між різноманітними пристроями та обробка великої кількості даних в режимі реального часу[4].

Архітектура IoT може бути представлена у вигляді кількох рівнів. Перший рівень – це фізичний рівень, що включає в себе датчики, актуатори та інші пристрої, що забезпечують збір даних та керування різними пристроями.

Другий рівень – це рівень мережі, що забезпечує передачу даних між пристроями та збір цих даних на центральній платформі. Цей рівень включає в себе різні протоколи зв'язку, такі як Wi-Fi, Bluetooth, Zigbee, і т.д.

					КвРКІ 2001127.20.01.01 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

Третій рівень – це рівень обробки даних, що включає в себе різні пристрої, що здатні обробляти велику кількість даних в режимі реального часу. Цей рівень включає в себе обчислювальні пристрої, такі як хмарні сервери, локальні сервери, і т.д.

Четвертий рівень - це рівень застосування, що включає в себе різноманітні програмні засоби, що забезпечують взаємодію між пристроями та обробку даних. Цей рівень може включати в себе різноманітні програмні засоби, такі як Node-RED, MQTT, і т.д. [20].

Архітектура IoT повинна бути забезпечена безпекою, зокрема забезпечувати конфіденційність та цілісність перед

Архітектура Інтернету речей відрізняється в залежності від реалізації. Тим не менше вона дещо схожа на архітектуру класичних систем АСУТП. Один із прикладів архітектури показаний на рисунку 1.1.

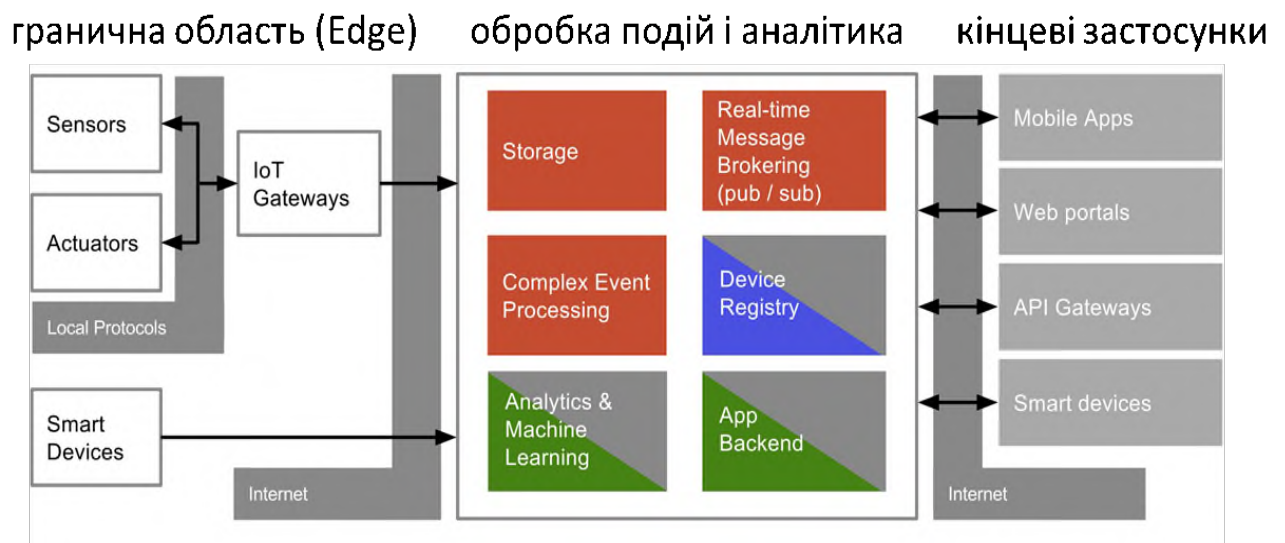


Рисунок 1.1 – Приклад архітектури Інтернету речей

Комунікація з пристроями Інтернету речей відбувається за допомогою датчиків і виконавчих механізмів, як у будь-якій АСУТП для керування об'єктом. Гранична область (Edge) складається з датчиків та інфраструктури для їх інтеграції з рівнем обробки подій через мережу Internet. На рівні обробки

подій події зберігаються, обробляються, перенаправляються додаткам і керуються адміністраторами та менеджерами пристроїв. Аналітичні сервіси та машинне навчання дозволяють зробити висновки про об'єкт. Цей рівень часто реалізується з використанням хмарних або туманних обчислень. Контроль, отримання результатів та адміністрування системи здійснюється через кінцеві застосунки з використанням Internet, аналогічно до НМІ в АСУТП[11].

На рисунку 1.2 показана подібна наведеній вище архітектура, однак у вигляді сервісів.

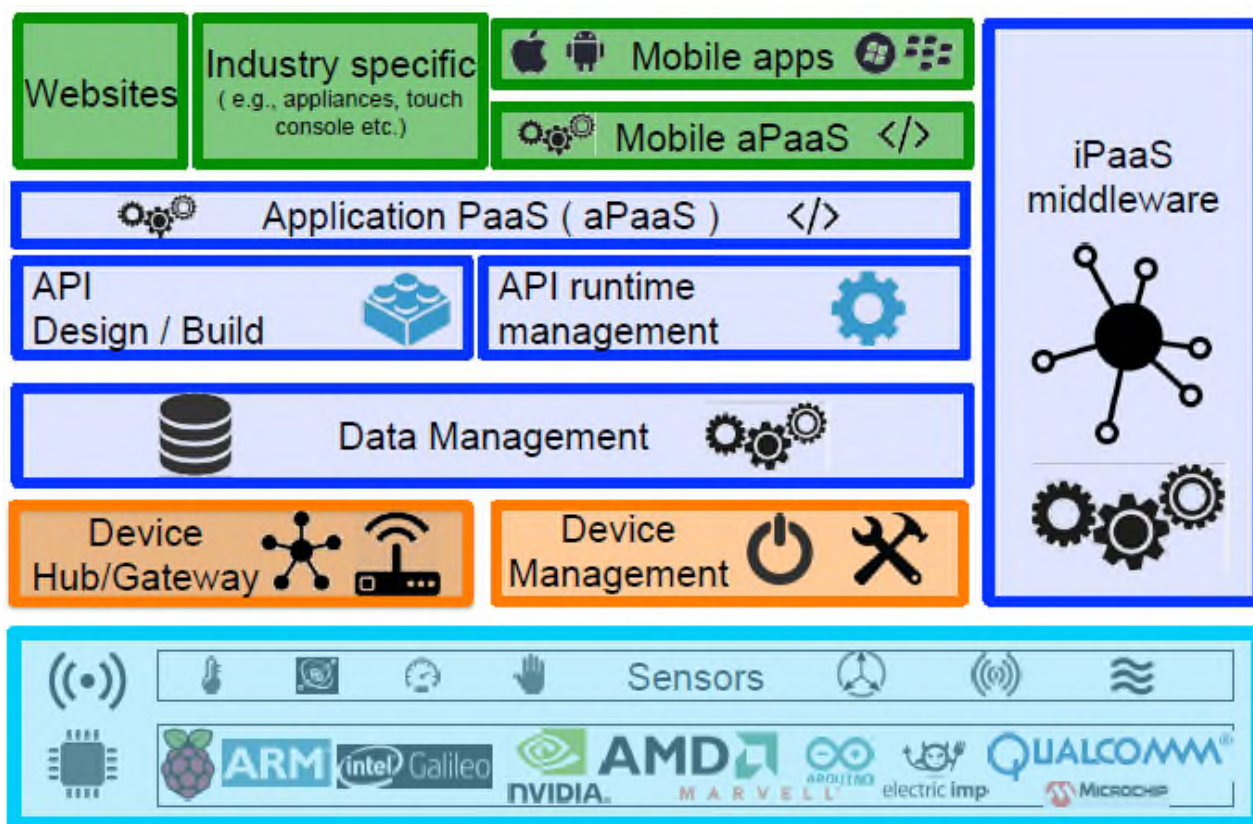


Рисунок 1.2 – Архітектура IoT у вигляді сервісів

На рівні Edge, границя системи Інтернету речей представлена датчиками (Sensors), Device Hub/Gateway (збір та маршрутизація даних) та Device Management (керування пристроями), які можуть виконуватися як на границі, так і на хмарних обчислювальних платформах. Функції збереження та первинної обробки подій (даних) зведені до Data Management. Усі інші функції

обробки, включаючи аналітику, реалізовані у вигляді додатків PaaS, які взаємодіють з сервісами керування даними через API (Application Program Interface) [16].

Системи збору та обробки даних в контексті Інтернету речей відіграють ключову роль в забезпеченні функціональності та ефективності цієї технології. Збір даних відбувається через датчики (sensors) та виконавчі механізми (actuators), які розміщені на різних об'єктах. Ці дані потім передаються на граничний рівень (Edge), де відбувається їх збір, обробка та передача на наступний рівень.

На граничному рівні зазвичай розташовані датчики та Device Hub/Gateway, які відповідають за збір та маршрутизацію даних, а також Device Management, яка забезпечує керування пристроями. Усі ці функції зведені до Data Management, що відповідає за збереження та первинну обробку даних.

На наступному рівні, який може бути реалізований з використанням хмарних або туманних обчислень, відбувається більш складна обробка даних, в тому числі аналітика, машинне навчання та інші додатки PaaS, що взаємодіють з сервісами керування даними через API (Application Program Interface).

Системи збору та обробки даних в Інтернеті речей мають великий потенціал у різних галузях, включаючи промисловість, транспорт, медицину та інші. Вони дозволяють збирати та аналізувати великі обсяги даних, що дозволяє забезпечити більш точні та ефективні рішення в різних сферах діяльності.

Передача даних в Інтернеті речей (IoT) відбувається з використанням різних протоколів та технологій залежно від конкретного застосування. Одним з найпоширеніших протоколів є MQTT (Message Queuing Telemetry Transport), який працює на основі публікації-підписки (publish-subscribe). У цьому протоколі пристрої публікують повідомлення в топіки (topics), а інші пристрої можуть підписатися на ці топіки та отримувати повідомлення. MQTT є легким та ефективним протоколом, що дозволяє передавати дані навіть при низьких швидкостях зв'язку.

Інші протоколи, що використовуються в IoT, включають HTTP (Hypertext Transfer Protocol), CoAP (Constrained Application Protocol), AMQP (Advanced Message Queuing Protocol) та інші. Кожен з цих протоколів має свої переваги та недоліки залежно від вимог до конкретної системи IoT.

Також для передачі даних можуть використовуватися різні мережі, включаючи Wi-Fi, Bluetooth, Zigbee, LoRaWAN та інші. Вибір мережі залежить від вимог до дальності передачі, швидкості, енергоспоживання та інших факторів[23].

Одним з викликів при передачі даних в IoT є захист інформації від несанкціонованого доступу та злому. Для цього використовуються різні методи, включаючи шифрування, аутентифікацію, авторизацію та інші заходи безпеки.

Інтернет речей (IoT) включає в себе розрізнення пристроїв, які можуть бути виробничого призначення, споживчого призначення, а також різні типи протоколів і мереж. В залежності від вимог до конкретної системи IoT, можуть використовуватись різні мови програмування та платформи розробки.

Першим елементом системи IoT є пристрій, який збирає дані. Загалом, це пристрої, підключені до Інтернету, тому кожен із них має свою IP-адресу. Їх складність варіюється від автономних мобільних роботів і вилкових навантажувачів, які переміщують продукти по фабриках і складах, до простих датчиків, які контролюють температуру або сканують витрки газу в будівлях.

Вони також включають персональні пристрої, такі як фітнес-трекери, які відстежують кількість кроків, які люди роблять щодня.

На наступному етапі процесу IoT зібрані дані передаються від пристроїв до точки збору. Переміщення даних може здійснюватися бездротовим способом за допомогою ряду технологій або через дротові мережі. Дані можна надсилати через Інтернет до центру обробки даних або хмари. Або передача може здійснюватися поетапно, за допомогою пристроїв-посередників, які збирають

					КвРКІ 2001127.20.01.01 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

дані, форматують їх, фільтрують, відкидають нерелевантні або повторювані дані, а потім надсилають важливі дані для подальшого аналізу.

Останній крок, обробка та аналітика даних, може відбуватися в центрах обробки даних або хмарі, але іноді це неможливий варіант. У випадку критичних пристроїв, таких як відключення в промислових умовах, затримка надсилання даних із пристрою до віддаленого центру обробки даних надто велика. Час зворотного зв'язку для надсилання даних, їх обробки, аналізу та повернення інструкцій (закрийте цей клапан, перш ніж труби лопнуть) може зайняти надто багато часу.

У таких випадках може вступити в гру периферійне обчислення, де розумний крайовий пристрій може агрегувати дані, аналізувати їх і, якщо необхідно, модифікувати відповіді, і все це на відносно близькій фізичній відстані, тим самим зменшуючи затримку. Пристрої Edge також мають висхідний зв'язок для надсилання даних для подальшої обробки та зберігання.

Зростаюча кількість варіантів використання периферійних обчислень, таких як автономні транспортні засоби, яким потрібно приймати рішення за частки секунди, прискорює розвиток периферійних технологій, які можуть обробляти та аналізувати дані негайно, не переходячи в хмару.

Оскільки кількість пристроїв IoT продовжує зростати, компанії продовжуватимуть удосконалювати функції безпеки та шукати швидші варіанти підключення, щоб забезпечити більше функцій для обробки та аналізу даних.

1.2 Огляд існуючих програмних рішень для керування Інтернетом речей

Найбільш популярні мови програмування для розробки додатків IoT:

1. C/C++ - ці мови є дуже швидкими та ефективними і підходять для написання драйверів та низькорівневого програмного забезпечення, що забезпечує зв'язок між пристроями IoT та мережами;

					КвРКІ 2001127.20.01.01 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

2. Python - це високорівнева мова програмування, що забезпечує багато можливостей для розробки додатків IoT. Вона дозволяє легко розробляти складні системи та швидко прототипувати рішення. Python також має багато бібліотек та фреймворків, які зроблюють розробку додатків для IoT ще простішою;

3. Java - це високорівнева мова програмування, яка є однією з найпопулярніших у світі. Java має багато бібліотек та фреймворків, які забезпечують можливість розробки додатків для IoT.

4. JavaScript - це мова програмування, яка використовується для розробки веб-додатків, але також може використовуватись для розробки додатків IoT, зокрема, для розробки мобільних додатків, які взаємодіють з IoT-пристроями через Bluetooth або Wi-Fi.

Огляд існуючих програмних рішень для керування Інтернетом речей (IoT) включає в собі різноманітні платформи, фреймворки та програмні засоби, що дозволяють створювати, керувати та моніторити IoT-рішення. Ось декілька з найбільш популярних програмних рішень для IoT:

1. Arduino: Arduino - це відкрита платформа, що дозволяє розробникам створювати прості та ефективні додатки IoT. Платформа має просту та зрозумілу інтерфейсну дошку та дозволяє використовувати більшість мов програмування;

2. Raspberry Pi: Raspberry Pi - це одноплатний комп'ютер, який можна використовувати як платформу для розробки додатків IoT. Він має дуже маленький розмір, що дозволяє використовувати його в обмеженому просторі;

3. Node-RED: Node-RED - це відкрита платформа, що дозволяє розробникам швидко та легко створювати взаємодіючі додатки IoT. Вона базується на візуальному програмуванні, що дозволяє розробникам легко створювати потоки даних, що працюють з різними датчиками та пристроями. Node-RED має велику кількість розширень та підтримує різні мови програмування;

4. AWS IoT: це сервіс платформи Amazon Web Services (AWS), який надає інструменти для збору, зберігання та обробки даних з IoT-пристроїв. AWS IoT також має інструменти для створення правил обробки даних та автоматичного реагування на події;

5. Microsoft Azure IoT: це платформа для підключення, моніторингу та керування IoT-пристроями. Azure IoT надає інструменти для збору та обробки даних, аналізу та візуалізації даних, а також інтеграції з іншими сервісами Azure.

6. Google Cloud IoT: це сервіс платформи Google Cloud, який дозволяє підключати, керувати та моніторити IoT-пристрої. Google Cloud IoT надає інструменти для збору, зберігання та обробки даних з IoT-пристроїв, а також для аналізу та візуалізації даних;

7. IBM Watson IoT Platform: це хмарна платформа, яка дозволяє збирати, аналізувати та використовувати дані з IoT-пристроїв. Вона включає в себе інструменти для розробки та впровадження додатків IoT, зокрема засоби збору даних, аналітики, керування даними та інтеграції;

8. Amazon Web Services (AWS) IoT: ця платформа надає інструменти для збору, зберігання та обробки даних з IoT-пристроїв. Вона також має інструменти для розробки та впровадження додатків IoT, зокрема інтеграцію з AWS Lambda, Amazon Kinesis та іншими.

1.3 Аналіз можливостей платформи Node-RED

Node-RED - це візуальна програмна платформа, побудована на платформі Node.js, яка дозволяє швидко та легко створювати IoT додатки. Основна перевага Node-RED полягає у тому, що вона надає можливість використовувати візуальні блоки для створення потоків даних, що дозволяє легко створювати та модифікувати програми без необхідності писати код з нуля.

					КвРКІ 2001127.20.01.01 ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

Основні можливості Node-RED:

1) візуальне програмування: Node-RED дозволяє використовувати готові блоки, які можна перетягувати та з'єднувати, щоб створювати потоки даних. Це дозволяє створювати додатки швидко та легко, не використовуючи велику кількість коду;

2) підтримка різних пристроїв: Node-RED підтримує різні протоколи та пристрої, що дозволяє працювати зі шлюзами IoT, сенсорами, контролерами, веб-службами та іншими пристроями;

3) розширюваність: Node-RED має велику кількість модулів, що дозволяє розширювати її функціональні можливості та додавати нові блоки;

4) підтримка управління даними: Node-RED підтримує збереження даних та може підключатись до різних баз даних, таких як MongoDB, MySQL та інші;

5) підтримка віддаленого доступу: Node-RED має вбудовану можливість підключення до віддалених пристроїв та збереження даних на серверах;

6) легка інтеграція з іншими системами: Node-RED може легко інтегруватися з іншими системами, такими як MQTT, Apache Kafka, Amazon Web Services та інші.

Node-RED є потужним інструментом для розробки та взаємодії з IoT-пристроями. Він має безліч корисних функцій та підтримує більшість популярних протоколів зв'язку та мережевих технологій, таких як MQTT, HTTP, TCP, UDP та інші.

Основні переваги платформи Node-RED:

1) легкість використання та швидкість розробки: Node-RED має інтуїтивний та простий інтерфейс, що дозволяє створювати потужні IoT-додатки за допомогою візуального програмування. Це дозволяє значно зменшити час розробки та спростити процес створення IoT-додатків;

2) розширюваність та наявність готових модулів: Node-RED має велику кількість готових модулів та плагінів, які дозволяють розширювати

функціональність платформи. Це дозволяє швидко додавати нові функції та можливості до вашого IoT-додатку без написання власного коду;

3) підтримка відкритих стандартів та протоколів: Node-RED підтримує більшість відкритих стандартів та протоколів для IoT, таких як MQTT, HTTP, TCP, UDP та інші. Це дозволяє взаємодіяти з різноманітними IoT-пристроями та системами без проблем зі сумісністю;

4) можливість розгортання на хмарних платформах: Node-RED можна використовувати на хмарних платформах, таких як IBM Cloud, AWS, Microsoft Azure та інші. Це дозволяє з легкістю розгорнути та масштабувати IoT-додатки на хмарних серверах;

5) підтримка Node.js: Node-RED побудований на Node.js, що дозволяє використовувати його в контексті Node.js-додатків.

Node-RED забезпечує можливість програмувати об'єднані апаратні пристрої, API та сервіси онлайн шляхом використання графічного редактора, інколи даже не потребуючи використання текстових мов програмування.

Редактор Node-RED заснований на браузері, що дозволяє легко збирати вузли зі широкої палітри, які можуть бути розгорнуті для виконання з одного клацання миші. Браузерний редактор потоків даних може бути використаний для створення окремих вузлів з різним функціоналом за допомогою JavaScript-функцій. Node-RED має базові вузли, але також можна встановлювати додатковий функціонал, використовуючи репозиторій NPM, або створювати свої вузли з унікальним функціоналом (рис.1.3).

У нижньому колонтитулі розташовані кнопки, за допомогою яких можна збільшувати, зменшувати або відновлювати стандартний рівень масштабування.

Крім того, тут також розміщена кнопка перемикачя для навігатора (рис.1.4).

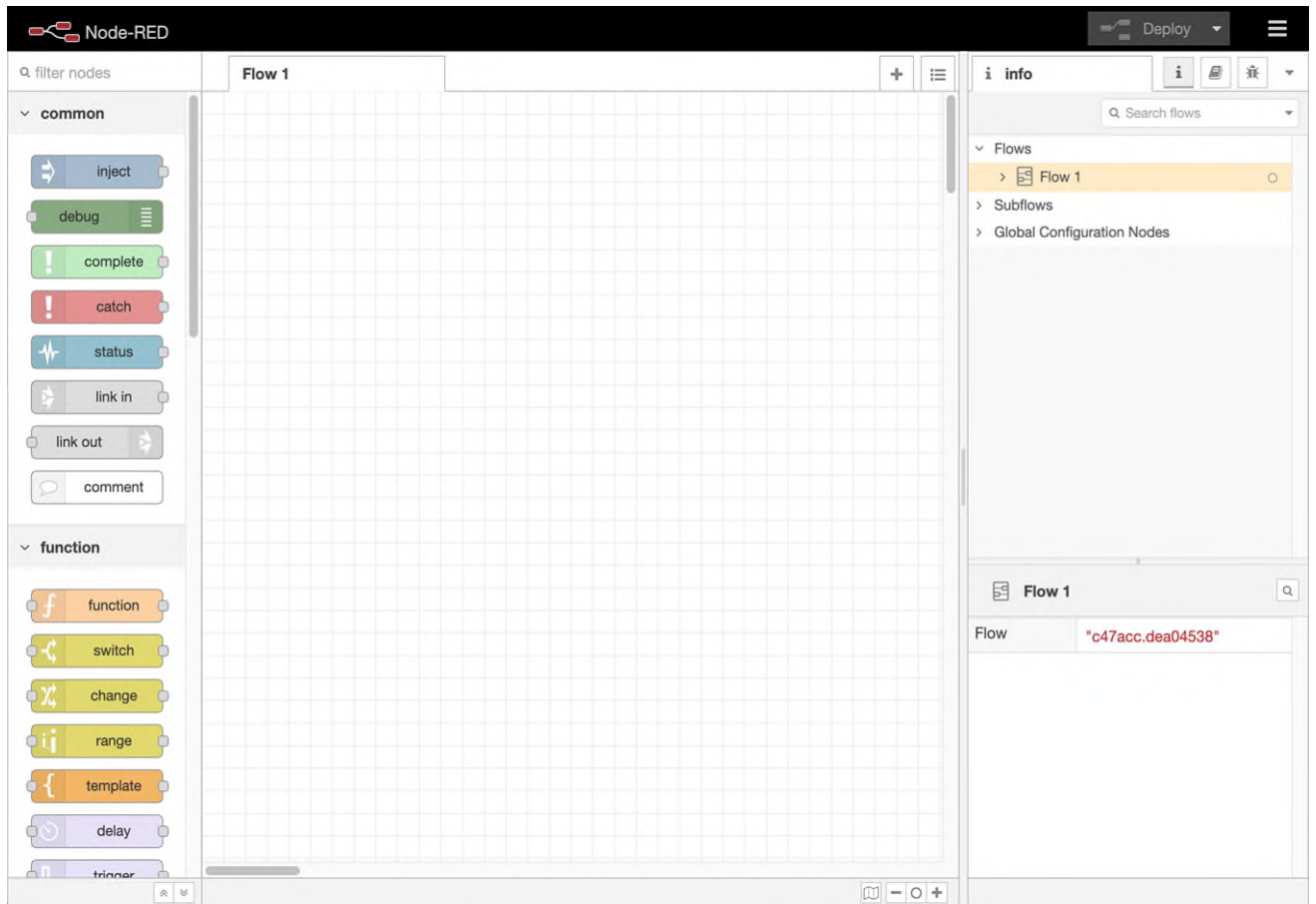


Рисунок 1.3 – Вигляд редактору Node-RED

Навігатор перегляду забезпечує звужений зображення всієї робочої області, яке виділяє потрібну зону для відображення. Цю зону можна перетягнути всередині навігатора, щоб швидко переміститись до інших частин робочої області.

Це також допомагає виявляти вузли, які можуть бути "поза полем зору" на інших ділянках робочого середовища[17].

Зовнішній вигляд робочої області можна налаштувати через вкладку "View" у діалоговому вікні "User Settings" (рис.1.5).

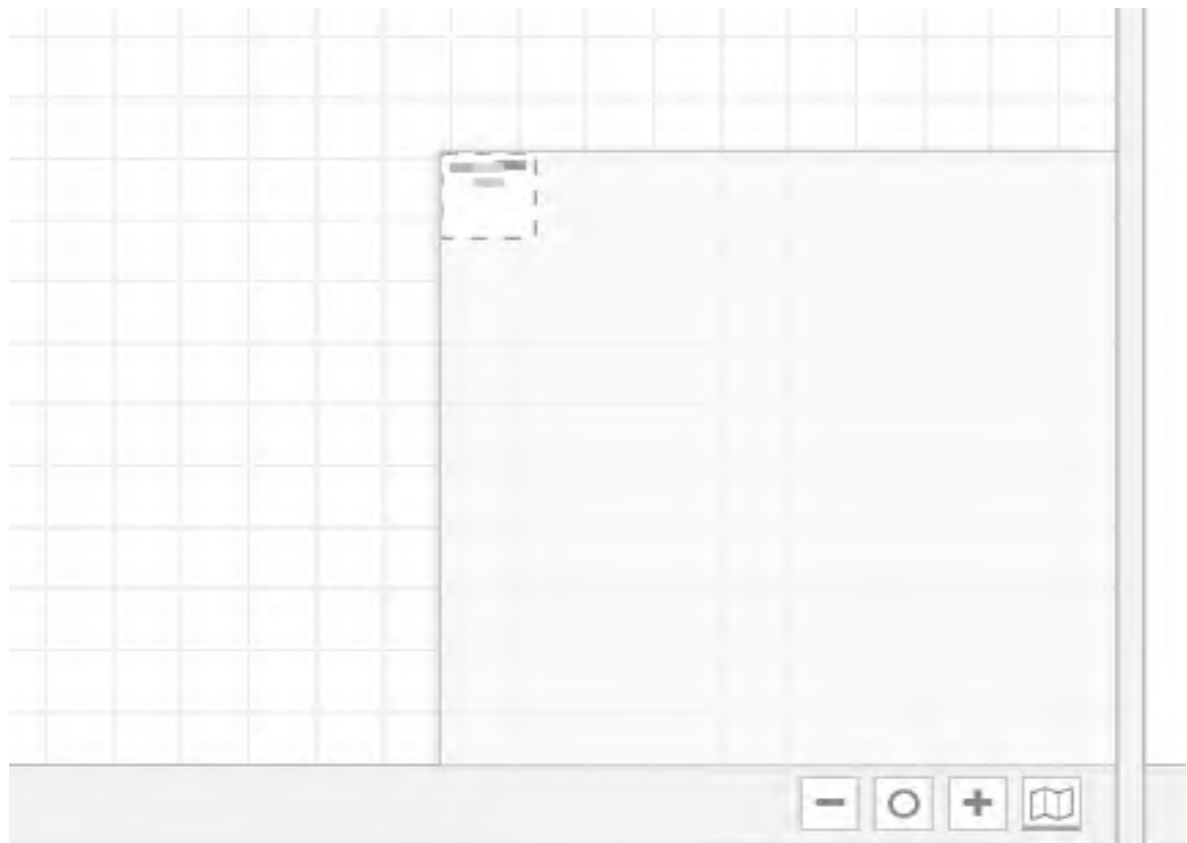


Рисунок 1.4 – Робоча область нижнього колонтитула з активним навігатором перегляду

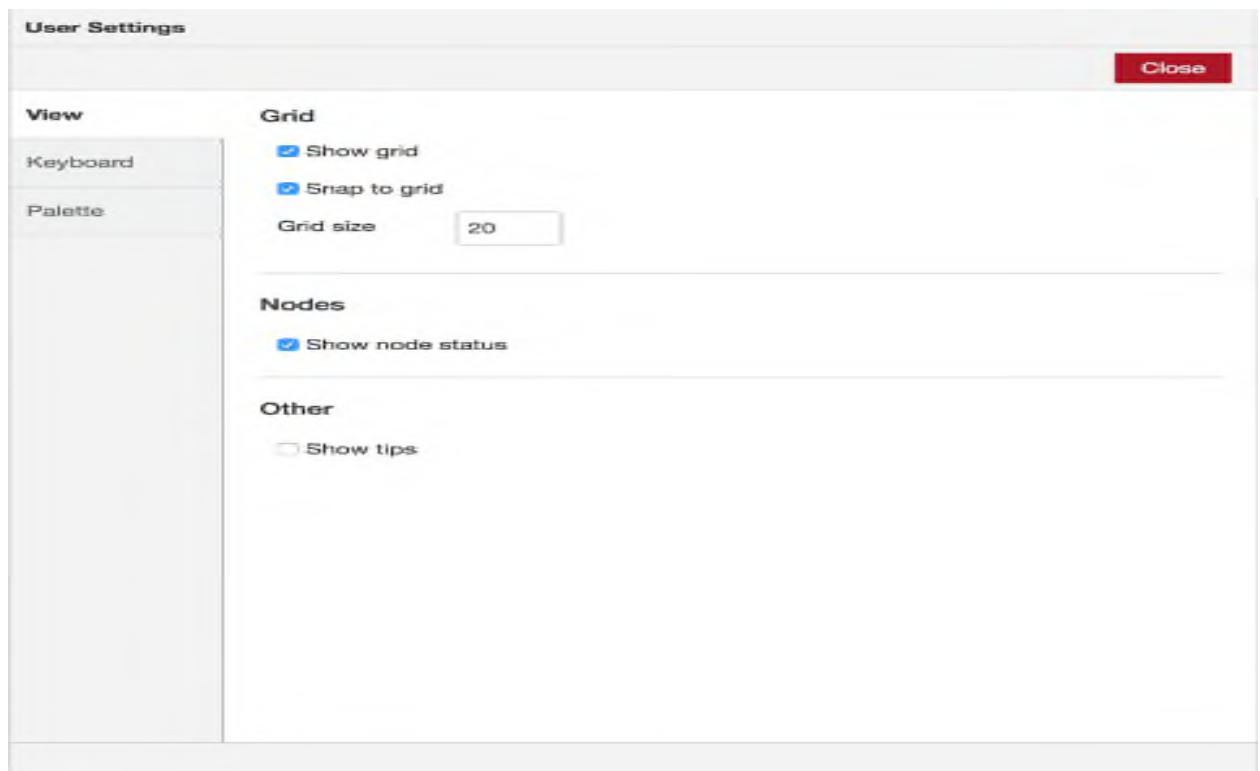


Рисунок 1.5 – Вкладка «User Settings»

У Node-RED програма складається з набору вузлів, які об'єднуються в потоки (flow) і розробляються в межах вкладок браузера з відповідною назвою. Вкладки, які містять потоки, також називаються потоками. Один потік може містити кілька наборів підключених вузлів, і тому термін "потік" часто використовується для опису цих наборів[29] (рис.1.6).



Рисунок 1.6 – Вкладки потоку

В Node-RED можна додавати, видаляти та деактивувати потоки, що не відображаються в розгорненому стані. До кожного потоку можна додавати опис у форматі Markdown, щоб легше розуміти його призначення. Також використовується неофіційний термін "потік" для опису одного набору підключених вузлів.

Тому потік (вкладка) може містити кілька потоків (наборів з'єднаних вузлів).

Для зміни властивостей потоку, потрібно двічі клацнути на його вкладці у верхній панелі. Таким чином, відкриється діалогове вікно "Flow Properties", де можливо налаштувати потрібні параметри (рис.1.7).

Вузли (Nodes) можуть бути додані до робочої області з використанням палітри вузлів або за допомогою їхніх імен. З'єднання між вузлами створюються через порти входу та виходу.

Вузол може мати:

- не більше одного вхідного порту;
- багато вихідних портів.

Мітки (labels) можуть бути додані до портів, щоб показати їх опис при наведенні курсору. Деякі вузли мають статусні повідомлення або піктограми,

щоб показати їх стан під час виконання. Синій кружок над вузлом відображається, якщо є зміни, які не були розгорнуті під час виконання. Червоний трикутник відображається, якщо в конфігурації виникли помилки. Вузли Inject та Debug є єдиними вузлами, які мають кнопки керування: вприскування (Inject) та відображення повідомлення (Debug).

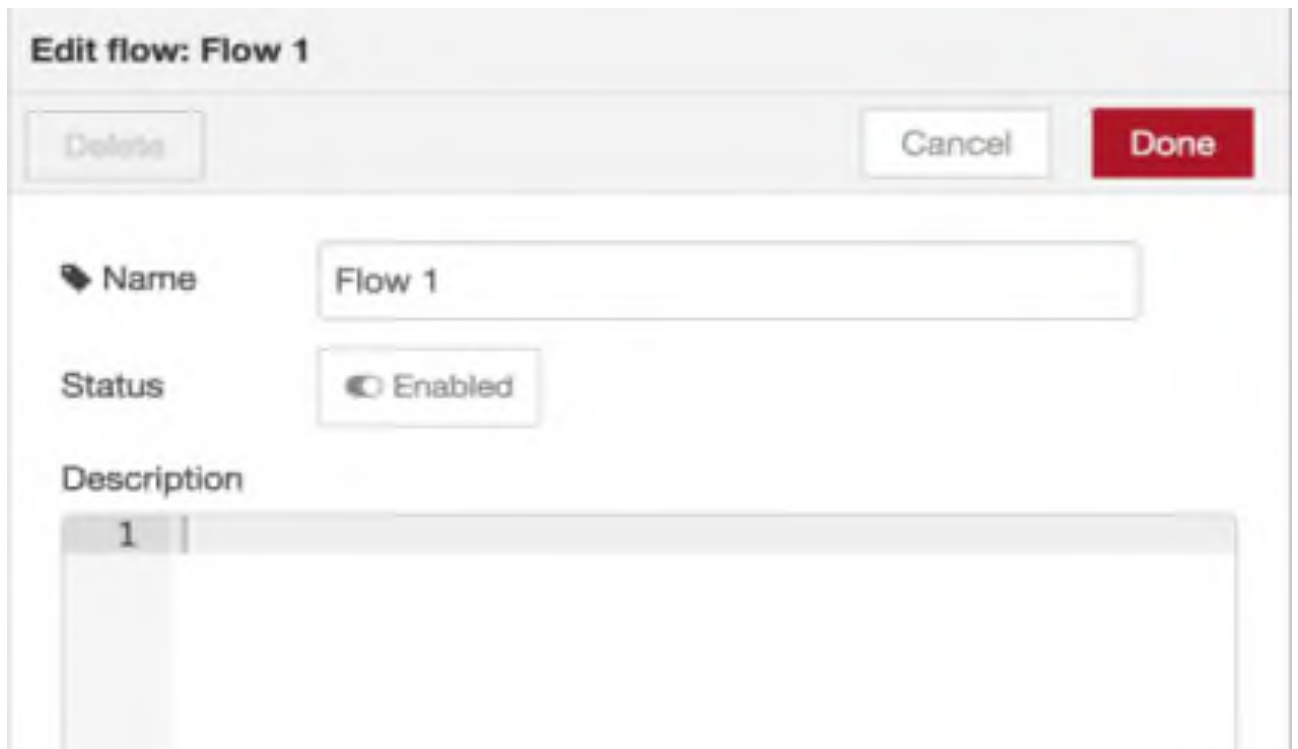


Рисунок 1.7 – Редагування властивостей потоку

Діалогове вікно редагування вузла має три окремих розділи: властивості (properties), опис (Description) та зовнішній вигляд (Appearance)

Конфігураційні вузли (config Node) - це особливий тип вузла, який містить налаштування для многоразового використання і може бути використаний іншими вузлами в потоці. Ці вузли доступні через бічну панель.

Вузли можуть бути об'єднані в групу (Group), утворюючи таким чином єдиний об'єкт, який може бути переміщений або скопійований в редакторі.

Огляд основних вузлів Node-RED v.1.1 включає наступні категорії вузлів[28] (рис.1.8):

- загальні (Common) - це найбільш загальні вузли, які використовуються для роботи з потоками;
- функціональні (Function) - ці вузли забезпечують функції перетворення повідомлень та керування потоком;
- послідовності (Sequence) - ці вузли використовуються для розбиття послідовностей повідомлень на підгрупи та збирання їх знову в одну послідовність;
- сховища (Storage) - ці вузли забезпечують роботу з читанням та записом файлів;
- мережні (Network) - ці вузли використовуються для роботи з WEB та IoT протоколами;
- парсери (Parser) - ці вузли забезпечують функції перетворення форматів повідомлень.

Під-потоки (subflow) – це група вузлів, яка згортається в один вузол у робочій області. Вони можуть бути використані для спрощення візуальної складності потоку або для об'єднання групи вузлів, яка використовується на різних етапах розробки.

Бічна панель надає користувачам наступні функції:

1. інформація (Information) - дозволяє переглянути докладну інформацію про вузли та отримати довідкову інформацію про їх функції;
2. Debug - дозволяє переглянути повідомлення, які передаються вузлам Debug.
3. конфігураційні вузли (Configuration Nodes) - дозволяє керувати налаштуваннями конфігураційних вузлів;
4. контекстні дані (Context data) - дозволяє переглянути вміст контекстів;
5. деякі вузли можуть мати власні панелі бічної панелі, наприклад, вузли node-red-dashboard.

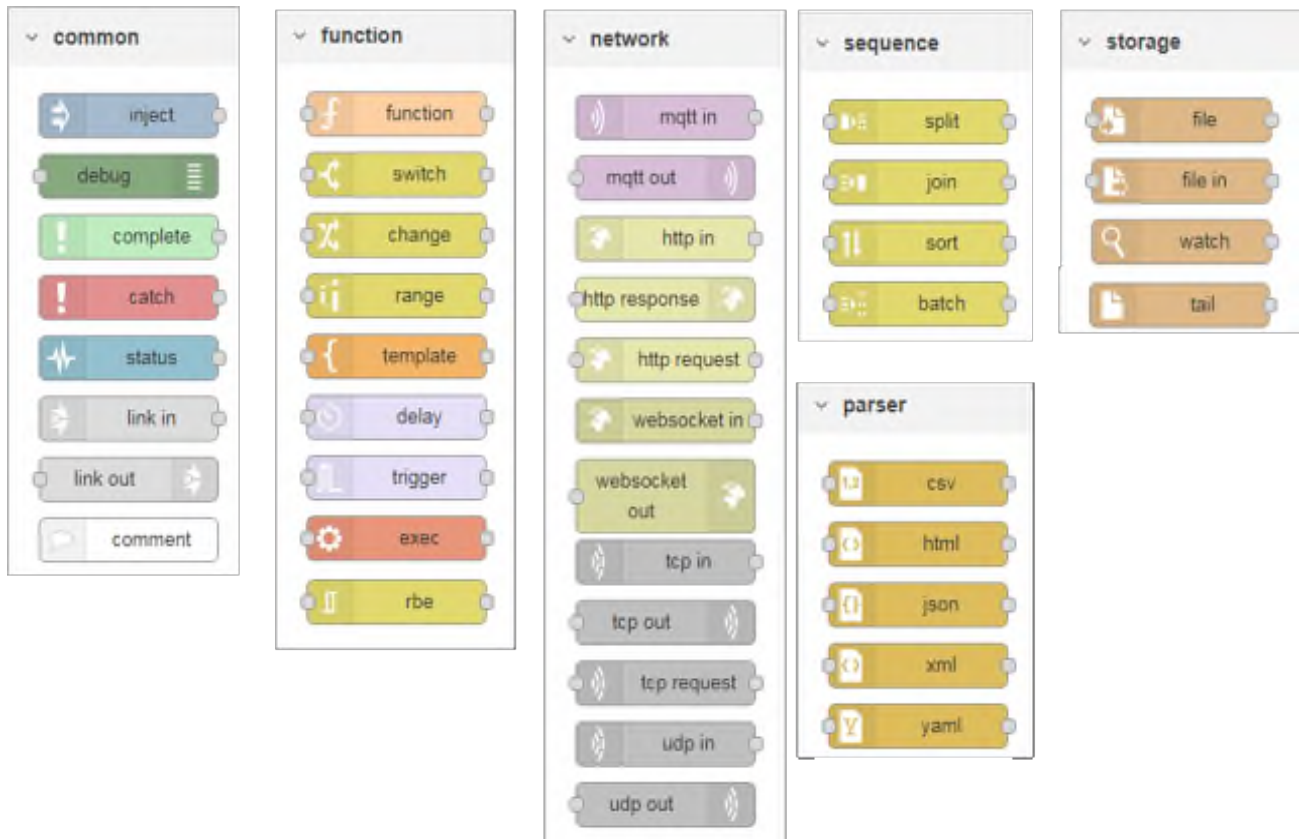


Рисунок 1.8 – Основні вузли Node-RED

Після ініціалізації програми, потік починає своє виконання через спрацьовування події, і кожен вузол у потоці виконує свою функціональність і передає отримані дані наступному вузлу у вигляді повідомлень

Подія, яка запускає потік це щось, що відбувається поза програмою, як-от зміна стану апаратного забезпечення, досягнення змінною заданого значення або певний проміжок часу. Розміщуючи та з'єднуючи вузли на полотні, хтось без глибоких знань програмування може створювати складні промислові програми IoT.

1.4 Постановка задачі

В данній кваліфікаційній роботі має бути розроблений програмно-технічний модуль керування Інтернетом речей на основі Node-RED. Основною

метою роботи є розробка зручного та ефективного інструменту керування різними пристроями Інтернету речей з використанням платформи Node-RED.

Для досягнення цієї мети потрібно виконати наступні завдання:

1. дослідити можливості платформи Node-RED для розробки програмного модуля керування Інтернетом речей та вивчити особливості роботи з нею;

2. розробити архітектуру програмного модуля та вибрати необхідні вузли для реалізації функціоналу;

3. реалізувати необхідний функціонал модуля, зокрема:

– розробити вузли для роботи з різними протоколами Інтернету речей (наприклад, MQTT, CoAP, HTTP);

– розробити вузли для зчитування та запису даних з пристроїв Інтернету речей;

– розробити вузли для обробки даних та прийняття рішень згідно з заданою логікою;

4. провести тестування та аналіз ефективності розробленого модуля на реальних пристроях Інтернету речей;

5. розробити документацію до програмного модуля, що містить опис архітектури, інструкції з використання, опис вузлів та приклади використання;

Результатом кваліфікаційної роботи має бути програмно-технічний модуль керування Інтернетом речей на основі Node-RED, який дозволяє зручно та ефективно керувати різними пристроями Інтернету речей з використанням графічного інтерфейсу.

1.5 Висновки до розділу

В рамках розділу було проведено дослідження і аналіз різних аспектів, пов'язаних з Інтернетом речей, а також існуючих програмних рішень для керування цією технологією. В результаті досліджень було сформульовано постановку задачі для розробки програмно-технічного модуля керування

					КвРКІ 2001127.20.01.01 ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

Інтернетом речей на основі Node-RED. Цей модуль має на меті забезпечити ефективно та гнучке управління підключеними пристроями та обмін даними в Інтернеті речей.

					КвРКІ 2001127.20.01.01 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ

2.1 Аналіз функціональних вимог до програмно-технічного засобу

Програмно-технічний модуль керування Інтернетом речей на основі Node-RED повинен виконувати наступні функціональні вимоги[14]:

1. підтримка протоколів IoT. Модуль повинен забезпечувати підтримку основних протоколів IoT, таких як MQTT, HTTP, WebSocket, CoAP та інших, для забезпечення взаємодії з IoT-пристроями;

2. керування та моніторинг IoT-пристроїв. Модуль повинен дозволяти користувачам керувати та моніторити IoT-пристрої за допомогою відповідних функцій. Керування може включати в себе встановлення параметрів, запуск та зупинку пристроїв, та інші дії, що залежать від можливостей конкретних пристроїв. Моніторинг повинен дозволяти переглядати дані, що отримуються від пристроїв, включаючи параметри, стани, інформацію про ресурси тощо;

3. підтримка візуального програмування. Модуль повинен забезпечувати можливість візуального програмування за допомогою Node-RED, що дозволяє створювати гнучкі та складні потоки обробки даних без необхідності в програмуванні на високорівневих мовах;

4. підтримка роботи з базами даних. Модуль повинен дозволяти зберігати та отримувати дані про IoT-пристрої в базі даних для подальшої обробки та аналізу.

5. розширюваність та конфігурування. Модуль повинен бути легко розширюваним та конфігуруватися залежно від потреб користувача. Повинні бути доступні інтерфейси для додавання нових протоколів, взаємодії з новими IoT-пристроями, розширення функціоналу та налаштування параметрів;

6. робота з базою даних. Модуль повинен підтримувати можливість зберігання та відображення даних в базі даних, включаючи часові мітки та ідентифікатори об'єктів;

					КвРКІ 2001127.20.01.01 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

7. робота з веб-сервером. Модуль повинен підтримувати можливість взаємодії з веб-сервером, зокрема, отримання даних з HTTP запитів та відправка HTTP відповідей;

8. відображення даних. Модуль повинен надавати можливість візуалізації даних, включаючи графіки, діаграми, таблиці та інші елементи.

9. робота з іншими протоколами та пристроями IoT. Модуль повинен підтримувати взаємодію з іншими пристроями IoT, що працюють на різних протоколах, таких як MQTT, CoAP, OPC UA, і т.д;

10. керування безпекою. Модуль повинен забезпечувати можливість забезпечення безпеки взаємодії з пристроями IoT та зберігання даних в базі даних.

11. підтримка масштабованості. Модуль повинен підтримувати можливість масштабування до великих систем з великою кількістю пристроїв та великим потоком даних;

12. надання документації. Модуль повинен надавати документацію про функціональні можливості, архітектуру, API, та інші деталі для легкої розуміння та використання користувачами;

13. підтримка розширення. Модуль повинен бути розширюваним та дозволяти додавати нові функції та взаємодіяти з іншими модулями для розширення можливостей системи.

2.2 Аналіз нефункціональних вимог до програмно-технічного засобу

1. ефективність та продуктивність:

– модуль повинен працювати ефективно та швидко обробляти великий потік даних з пристроїв IoT;

– потрібно забезпечити оптимізацію роботи модуля для забезпечення мінімального часу відповіді та максимальної продуктивності.

2. надійність та стабільність:

- модуль повинен бути стабільним та надійним у роботі;
- вимагається забезпечити відновлення роботи модуля в разі виникнення помилок або збоїв;
- потрібно регулярно здійснювати тестування модуля на стійкість та надійність.

3. сумісність та інтеграція:

- модуль повинен бути сумісним з різними пристроями IoT та підтримувати різні протоколи комунікації;
- важливо забезпечити можливість інтеграції модуля з іншими системами та платформами.

4. зручність використання та інтерфейс користувача:

- модуль повинен мати зручний та інтуїтивно зрозумілий інтерфейс для користувачів;
- важливо забезпечити простоту використання та легкість налаштування модуля;
- потрібно забезпечити зручну навігацію та взаємодію з вузлами та потоками в середовищі Node-RED.

5. безпека:

- модуль повинен забезпечувати безпеку зберігання та передачі даних, включаючи шифрування та аутентифікацію;
- важливо враховувати заходи безпеки для запобігання несанкціонованому доступу до системи та забезпечення конфіденційності даних.

6. Масштабованість:

- модуль повинен бути масштабованим для обробки зростаючого обсягу даних та забезпечення працездатності при великій кількості пристроїв IoT;
- потрібно враховувати можливість горизонтального та вертикального масштабування модуля;

7. Модульність та розширюваність:

- модуль повинен бути модульним, дозволяючи додавати нові функції та вузли без зміни основної структури модуля;
- важливо мати можливість розширювати функціональність модуля шляхом додавання власних вузлів та інших компонентів.

8. Документація та підтримка:

- потрібно надати докладну документацію, що пояснює використання та налаштування модуля;
- важливо забезпечити підтримку користувачів та відповіді на їхні запитання та проблеми щодо використання модуля;

9. Витрати та обмеження:

- модуль повинен бути ефективним з точки зору витрат ресурсів, таких як пам'ять та обчислювальна потужність;
- вимагається забезпечити підтримку різних платформ та апаратних засобів, враховуючи їхні обмеження;

10. Сумісність з екосистемою Node-RED:

- модуль повинен бути сумісним з екосистемою Node-RED та іншими компонентами, що використовуються у середовищі Node-RED;
- важливо дотримуватись стандартів та протоколів, що використовуються в Node-RED, для забезпечення сумісності та взаємодії з іншими модулями та вузлами.

2.3 Опис архітектури програмно-технічного засобу

Використовуючи u-create web, веб-середовище розробки для контролерів u-control, буде створенна програма, яка надсилатиме сповіщення електронною поштою технікам з обслуговування, коли параметр процесу досягне критичного рівня.

Під час розробки програми, буде продемонстровано процес збору, обробки та передачі даних з системи керування за допомогою Node-RED (рис.2.1).

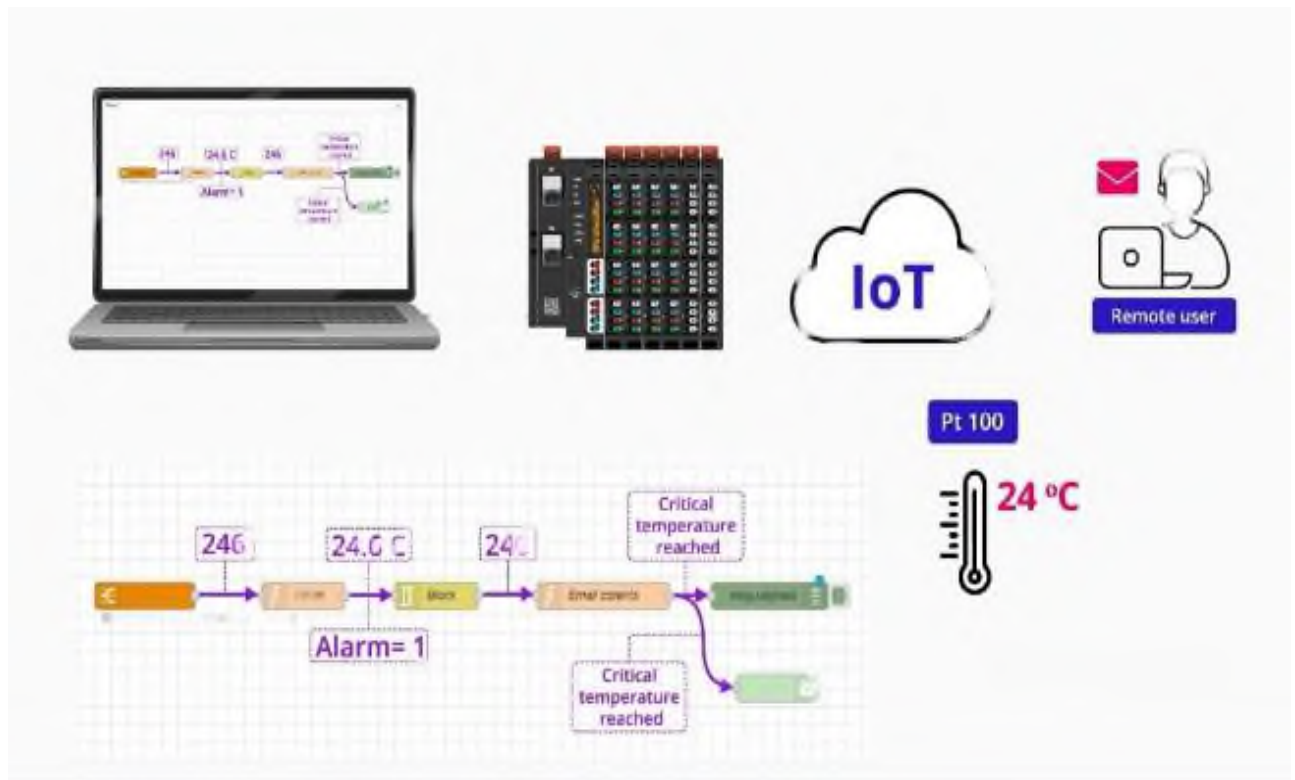


Рисунок 2.1 – Архітектура програмно-технічного засобу

Вхідні джерела даних.

Веб-середовище розробки. U-Create Web. U-Create Web – це середовище розробки, спеціально створене для розробки програмного забезпечення для контролерів U-Control. Воно надає зручний інтерфейс та набір інструментів для створення програмного коду та керування контролерами U-Control.

Основні особливості U-Create Web включають:

- візуальний інтерфейс: U-Create Web має інтуїтивний візуальний інтерфейс, який дозволяє легко створювати та керувати програмним кодом для контролерів U-Control. За допомогою графічного редактора можна встановлювати зв'язки між різними компонентами програми та налаштовувати їх параметри;
- бібліотека компонентів: U-Create Web має багатий набір готових компонентів, які можна використовувати при розробці програмного коду. Це

дозволяє прискорити процес розробки та забезпечити високу якість програмного забезпечення;

– інтеграція з контролерами U-Control: U-Create Web надає можливість зв'язуватися з контролерами U-Control та взаємодіяти з ними. За допомогою спеціальних інструментів та функцій можна завантажувати програмний код на контролер, керувати його режимами роботи та зчитувати дані з контролера;

– налаштування та налагодження: U-Create Web дозволяє налаштовувати різні параметри програми та контролера, виконувати налагодження програмного коду та аналізувати його роботу;

– підтримка різних мов програмування: U-Create Web підтримує різні мови програмування, такі як C, C++, Lua та інші, що дозволяє розробляти програмне забезпечення для контролерів U-Control з використанням обраної мови;

U-Create Web допомагає розробникам в створенні програмного забезпечення для контролерів U-Control, надаючи такі можливості:

– симуляція та тестування: U-Create Web дозволяє виконувати симуляцію роботи контролера та тестування програмного коду перед його завантаженням на реальний контролер. Це дозволяє виявити та виправити помилки до їх впливу на роботу контролера;

– відладка: U-Create Web надає засоби для відладки програмного коду, що дозволяє розробникам виявляти та виправляти помилки, аналізувати стан програми та зчитувати значення змінних під час виконання;

– управління версіями: U-Create Web може інтегруватися з системами контролю версій, такими як Git, що дозволяє розробникам керувати та відстежувати зміни в програмному коді, працювати в команді та спільно вдосконалювати програмне забезпечення;

– інтеграція з іншими сервісами: U-Create Web може взаємодіяти з іншими сервісами та інструментами, такими як хмарні платформи, веб-сервіси

або інші IoT-продукти. Це дає можливість розширити функціональність та інтегрувати контролер U-Control з іншими системами.

– документація та спільнота: U-Create Web забезпечує доступ до документації, прикладів та ресурсів, що допомагають розробникам у вивченні та використанні середовища розробки. Також існує активна спільнота розробників, де можна задавати питання, обмінюватись досвідом та знаходити рішення;

Загалом, U-Create Web є потужним середовищем розробки для контролерів U-Control надає інструменти, ресурси та функціональність, що сприяють швидкому та ефективному створенню програмного забезпечення для контролерів U-Control. Воно спрощує розробку, тестування, налагодження та управління версіями програмного коду, забезпечуючи зручний інтерфейс та готові компоненти для використання. Благодаря симуляції та можливості відладки, розробники можуть виявляти та виправляти помилки перед реальним використанням контролера. Крім того, інтеграція з іншими сервісами та інструментами дозволяє розширити функціональність та інтегрувати контролер U-Control з іншими системами. Загалом, U-Create Web є потужним інструментом, що допомагає в розробці програмного забезпечення для контролерів U-Control, забезпечуючи зручну та ефективну роботу з ними.

Контролер U-Control. Контролер U-Control є компактним пристроєм, призначеним для керування і моніторингу різних процесів і пристроїв. Він надає можливість збирати дані, керувати входами-виходами, здійснювати комунікацію з іншими пристроями та виконувати програми для автоматизації процесів. Зазвичай має різні входи-виходи, такі як аналогові та цифрові входи, аналогові та цифрові виходи, комунікаційні порти для підключення до зовнішніх пристроїв, таких як сенсори, актуатори, дисплеї та інші периферійні пристрої.

Контролер працює під управлінням певної операційної системи або платформи, яка забезпечує зручний інтерфейс для програмування, налаштування та моніторингу пристрою. Він може підтримувати різні мови програмування та

середовища розробки, такі як C/C++, Java, Python або спеціалізовані середовища, зокрема Node-RED.

Контролери U-Control широко використовуються у галузі автоматизації, індустріального управління, домашньої автоматизації та інших сферах, де потрібно керувати і контролювати процеси та пристрої.



Рисунок 2.2 - Контролер U-Control

Вузли та потоки.

iodata-in вузол (рис. 2.3). Вузол "iodata-in" в Node-RED є вузлом, який використовується для отримання вхідних даних з певного пристрою або джерела даних. Цей вузол дозволяє підключитися до різних зовнішніх пристроїв або сервісів, таких як датчики, контролери, бази даних або API, для отримання даних.

"iodata-in" може мати різні налаштування, які визначають, звідки і як отримувати дані. Це може бути адреса URL API, ідентифікатор пристрою або параметри підключення до конкретного джерела даних. Крім того, можуть бути

налаштовані функції обробки даних, такі як фільтрація, конвертація або агрегація[18].

При використанні вузла "iodata-in" потрібно налаштувати його параметри відповідно до джерела даних, після чого вузол буде приймати вхідні дані з цього джерела і передавати їх у наступні вузли потоку для подальшої обробки або виконання необхідних дій[26].



Рисунок 2.3 – Вузол iodata-in

Вузол Debug (рис. 2.4). Вузол "Debug" в Node-RED є вузлом, який використовується для відлагодження та відображення повідомлень під час розробки та виконання потоків. Цей вузол дозволяє переглядати дані, які передаються через потік у реальному часі, що допомагає відстежувати значення, відлагоджувати проблеми та перевіряти правильність роботи потоку.

Після розміщення вузла "Debug" у потоці ви можете налаштувати його для відображення конкретних властивостей повідомлення (рис. 2.5), таких як значення, тип, мітка часу та інші. Вузол може бути налаштований на відображення повідомлень, які проходять через нього, або тільки тих, що задовольняють певні умови.

Після розгортання програми вузол "Debug" буде виводити відповідні повідомлення в консолі налагодження Node-RED. Це дозволяє вам спостерігати за потоком даних, переконатися в правильності обробки та відлагоджувати будь-які проблеми, які можуть виникнути під час виконання потоку.

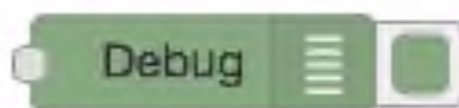


Рисунок 2.4 – Вузол Debug

Бічна панель надає структурований перегляд надісланих повідомлень, що спрощує їх аналіз. Кожне повідомлення на бічній панелі включає інформацію про час надходження та джерело, з якого було надіслано повідомлення. Ідентифікатор вихідного вузла можна натиснути, щоб відобразити цей вузол у робочій області. Кнопка, розташована на вузлі, дозволяє активувати або деактивувати його виходи.

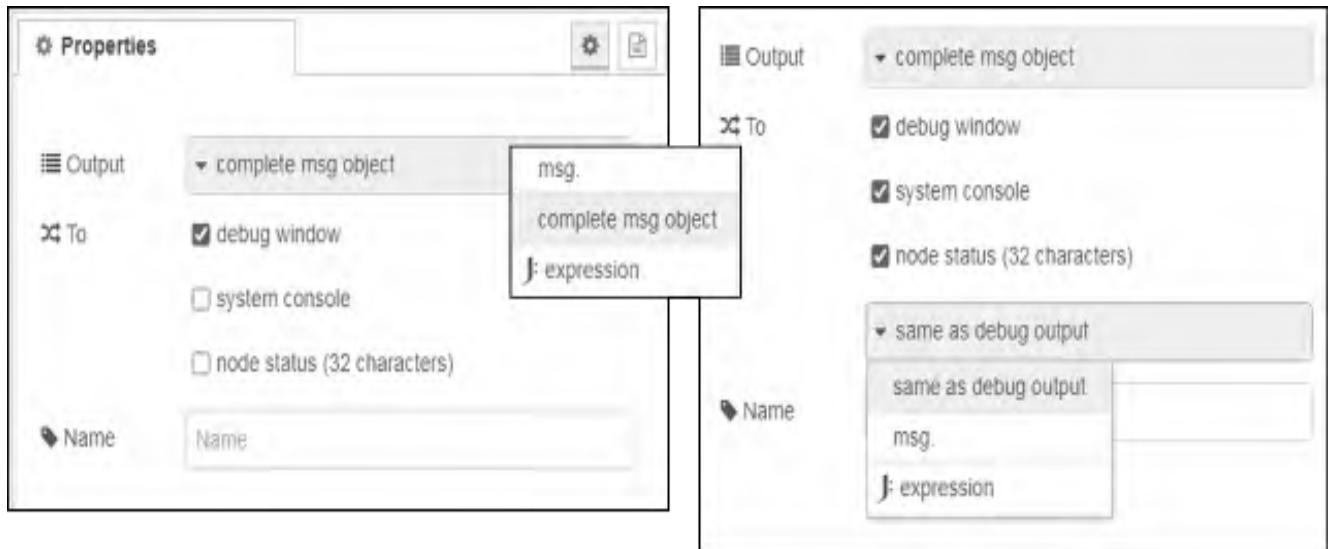


Рисунок 2.5 - Налаштування відображення властивостей повідомлення

Функціональні вузли.

Вузол Function (рис. 2.6). Вузол "Function" в Node-RED є потужним інструментом, який дозволяє вам виконувати власний JavaScript-код для маніпулювання даними потоку. Цей вузол дозволяє створювати власні функції та логіку обробки даних прямо всередині потоку.

Після розміщення вузла "Function" у потоці можливо відкрити його редактор, де можна ввести свій власний JavaScript-код. Ви можете доступатися до вхідних повідомлень та їх властивостей, змінювати дані, створювати нові властивості та визначати умови для пересилання повідомлень до наступних вузлів.

Вузол "Function" може бути використаний для розрахунків, перетворень даних, фільтрації, створення нових повідомлень, валідації та багатьох інших завдань, що вимагають власної логіки обробки даних.

Після розгортання програми вузол "Function" буде виконувати власний JavaScript-код для кожного вхідного повідомлення, дозволяючи вам впливати на дані, їх обробку та подальше поведінку потоку. Одним з корисних застосувань вузла "Function" є валідація та фільтрація даних. Ви можете перевіряти умови та фільтрувати повідомлення, що не задовольняють певні вимоги, перед тим як вони будуть передані до наступних вузлів.

Загалом, вузол "Function" дозволяє вам використовувати власний JavaScript-код для виразного та гнучкого управління обробкою даних в потоці Node-RED. Він відкриває безліч можливостей для створення складних потоків обробки даних з використанням ваших власних логік та функцій[27].



Рисункок 2.6 – Вузол Function

Вузол RBE (рис. 2.7). Вузол "RBE" (Report By Exception) в Node-RED є корисним інструментом для контролю повідомлень у потоці та блокування їх передачі, якщо значення не змінилося. Це дозволяє економити пропускну здатність мережі та ресурси системи, зменшуючи непотрібну передачу однакових даних.

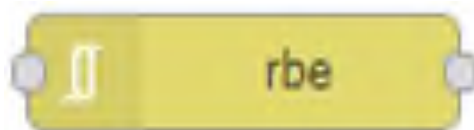


Рисунок 2.7 – Вузол RBE

Вузол працює на основі порівняння значень вхідних полів з попередніми значеннями. Якщо значення змінюється, вузол дозволяє йому пройти через потік і передає його на вихід. Якщо значення не змінюється, вузол блокує його та не передає наступним вузлам у потоці.

Властивості вузла "RBE" дозволяють налаштувати, які поля враховувати при порівнянні значень, чи враховувати тип даних, та чи зберігати останні значення для порівняння (рис.2.8)

Використання вузла допомагає покращити ефективність передачі даних у потоці Node-RED і зменшити непотрібну обробку повторюваних значень.

Вузол працює на основі msg.topic. Це означає, що один вузол rbe може обробляти декілька різних тем одночасно.

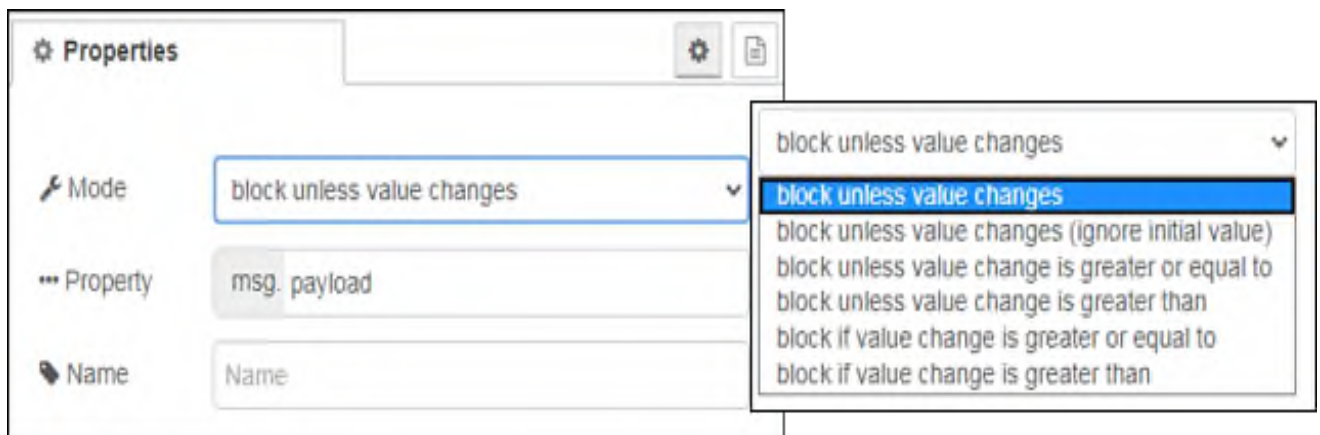


Рисунок 2.8 – Налаштування вузла RBE

Вузли зовнішніх сервісів:

Вузол Email (рис.2.9) Вузол "Email" в Node-RED є потоковим вузлом, який дозволяє надсилати електронні листи з потоку обробки даних. Він є корисним для автоматичної відправки електронних листів зі згенерованими даними або сповіщень.

Для налаштування вузла "Email" потрібно вказати параметри, такі як адреса електронної пошти відправника і отримувача, сервер електронної пошти,

порт, тип з'єднання, аутентифікаційні дані (якщо потрібно) та інші налаштування, які вимагаються для з'єднання з поштовим сервером.

Вузол "Email" може бути налаштований для надсилання текстових або HTML-повідомлень. Ви можете використовувати змінні контексту або значення полів повідомлення для динамічного форматування та персоналізації вмісту листа.

У випадку успішної надсилки електронного листа вхідне повідомлення буде передане на вихідний порт "sent". У випадку, якщо виникає помилка під час надсилання, повідомлення буде направлено на вихідний порт "error".

Вузол "Email" дозволяє автоматизувати процес надсилання електронних листів з потоку обробки даних, що робить його корисним інструментом для сповіщень, моніторингу та комунікації.



Рисунок 2.9 – Вузол Email

2.4 Висновок до розділу

У другому розділі було проведено детальний аналіз функціональних та нефункціональних вимог до програмно-технічного засобу. У першому підрозділі були ідентифіковані основні функції, які повинен виконувати модуль керування Інтернетом речей, включаючи збір та обробку даних, взаємодію з різними пристроями та сервісами, а також надання зручного інтерфейсу для користувача. В другому підрозділі були встановлені нефункціональні вимоги, такі як надійність, продуктивність, безпека та масштабованість програмно-технічного засобу.

Також, у третьому підрозділі було проведено детальний опис архітектури програмно-технічного засобу. В основі цієї архітектури лежить використання

Node-RED, який є потужним інструментом для розробки Інтернету речей. Було описано основні компоненти системи, такі як вузли, потоки та підключення між ними. Було детально розглянуто вузли, такі як "io-data-in", "Function", "RBE", "Email" та інші, які використовуються для збору, обробки та передачі даних в системі.

В результаті розділу "Проектування програмно-технічного засобу" було встановлено основні вимоги та описано архітектуру програмно-технічного засобу для керування Інтернетом речей на основі Node-RED. Цей розділ є важливим кроком у розробці програмно-технічного засобу і надає чітке розуміння його функціональності, властивостей та структури. Наступним кроком буде реалізація цих концепцій у практичному виконанні програмно-технічного засобу.

					КвРКІ 2001127.20.01.01 ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ

3.1 Огляд технічних характеристик та вимог до апаратного забезпечення

Вимоги до апаратного забезпечення наступні:

1. Процесор. Мінімум 1,0 ГГц процесор або більше.
2. Оперативна пам'ять. Мінімум 1 ГБ оперативної пам'яті (RAM) або більше.
3. Місце на жорсткому диску. Мінімум 100 МБ вільного місця на жорсткому диску.
4. Операційна система. Node-RED підтримує різні операційні системи, такі як Windows, macOS та Linux.
5. Версія Node.js. Node-RED підтримує версії Node.js від 10.x до останньої стабільної версії. Рекомендується використовувати останню стабільну версію Node.js для отримання найновіших функцій та виправлень помилок.
6. Пакети залежностей. Node-RED має свої власні пакети залежностей, які встановлюються разом з ним. Версії цих пакетів можуть бути зазначені у файлі package.json Node-RED. Для кращої сумісності та стабільності рекомендується використовувати версії пакетів, які поставляються з встановленою версією Node-RED.

3.2 Проектування архітектури програмно-технічного засобу

Для початку потрібно включити стартовий комплект u-control і підключити його до комп'ютера за допомогою USB-кабелю (рис.3.1).

Після встановлення з'єднання між контролером і комп'ютером необхідно відкрити веб-браузер і перейти до 192.168.10.202, IP-адреси USB-порту

контролера за умовчанням. Коли завантажується сторінка входу, можна увійти за допомогою облікових даних за замовчуванням (рис.3.2).



Рисунок 3.1 – З'єднання контролера з комп'ютером

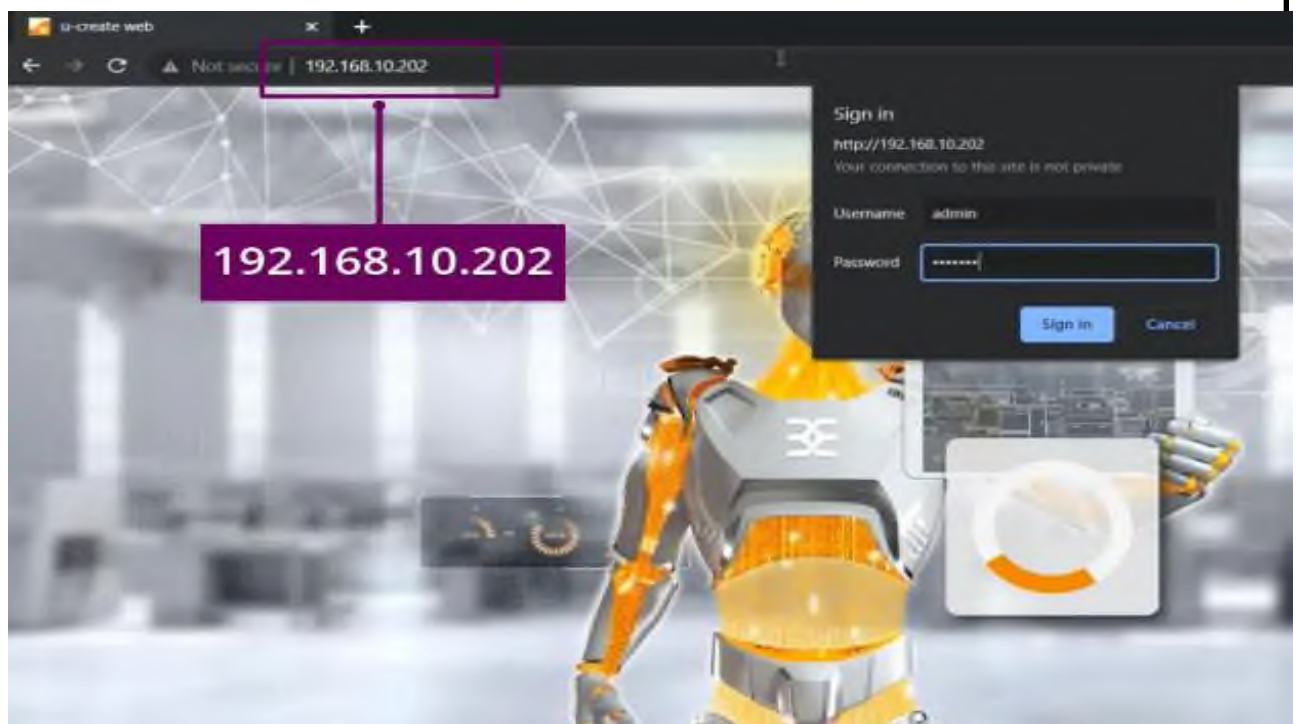


Рисунок 3.2 – Вхід до середовища розробки контролера u-control

На домашній веб-сторінці u-create потрібно натиснути «Редагувати», щоб почати змінювати програму реального часу на контролері. Відкривається редактор програми реального часу, і натиснути «Глобальні змінні», щоб змінити глобальні змінні програми реального часу (рис.3.3, рис3.4).

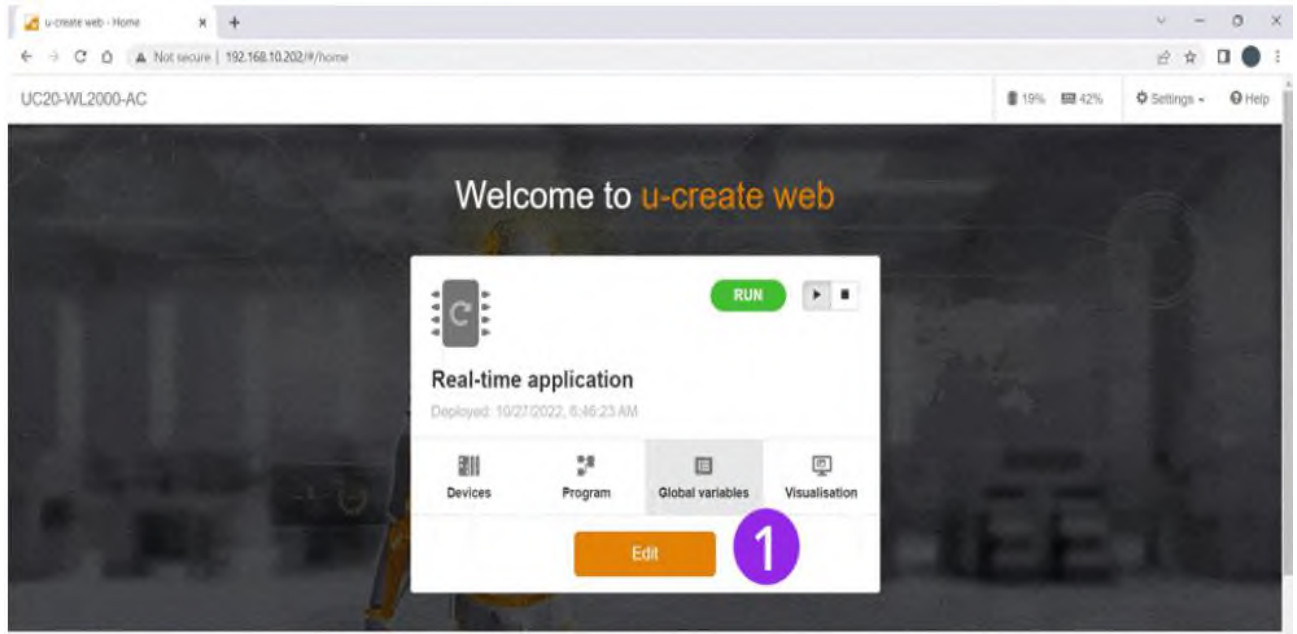


Рисунок 3.3 – Редактор додатків у реальному часі

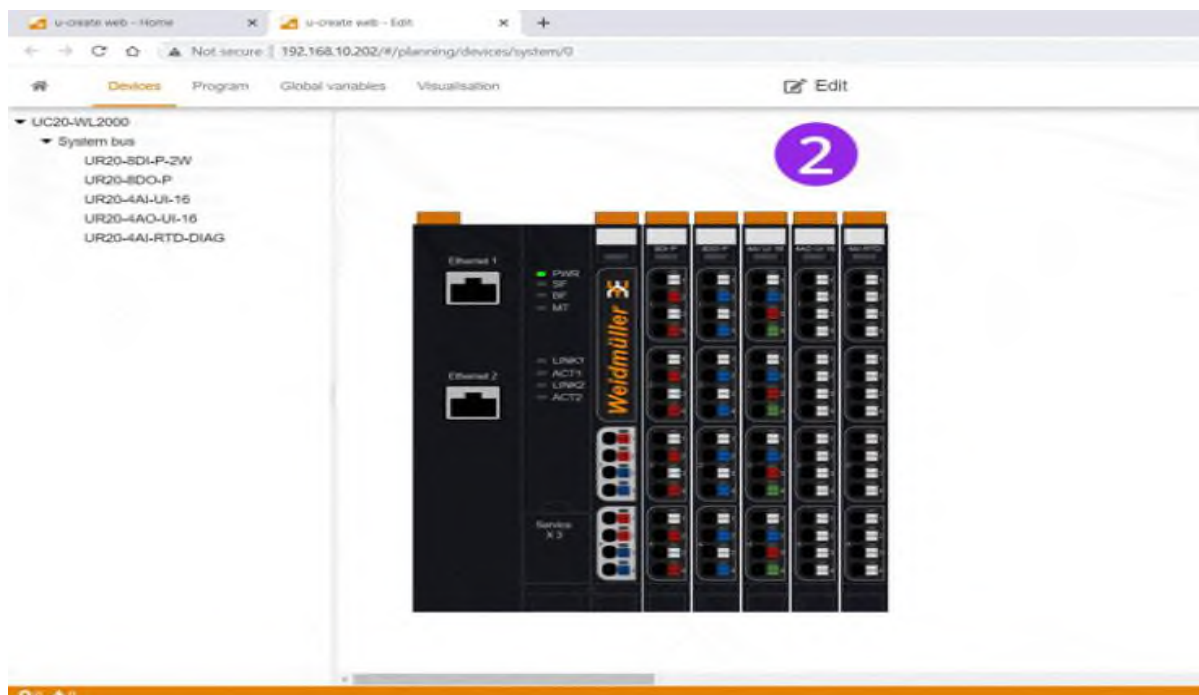


Рисунок 3.4 – Редактор додатків у реальному часі

В таблиці глобальних змінних, потрібно додати нову змінну, натиснувши на значок плюс у верхньому правому куті. Я обираю назву "Process Temperature" для нової змінної та встановлюю тип даних INT(рис.3.5).

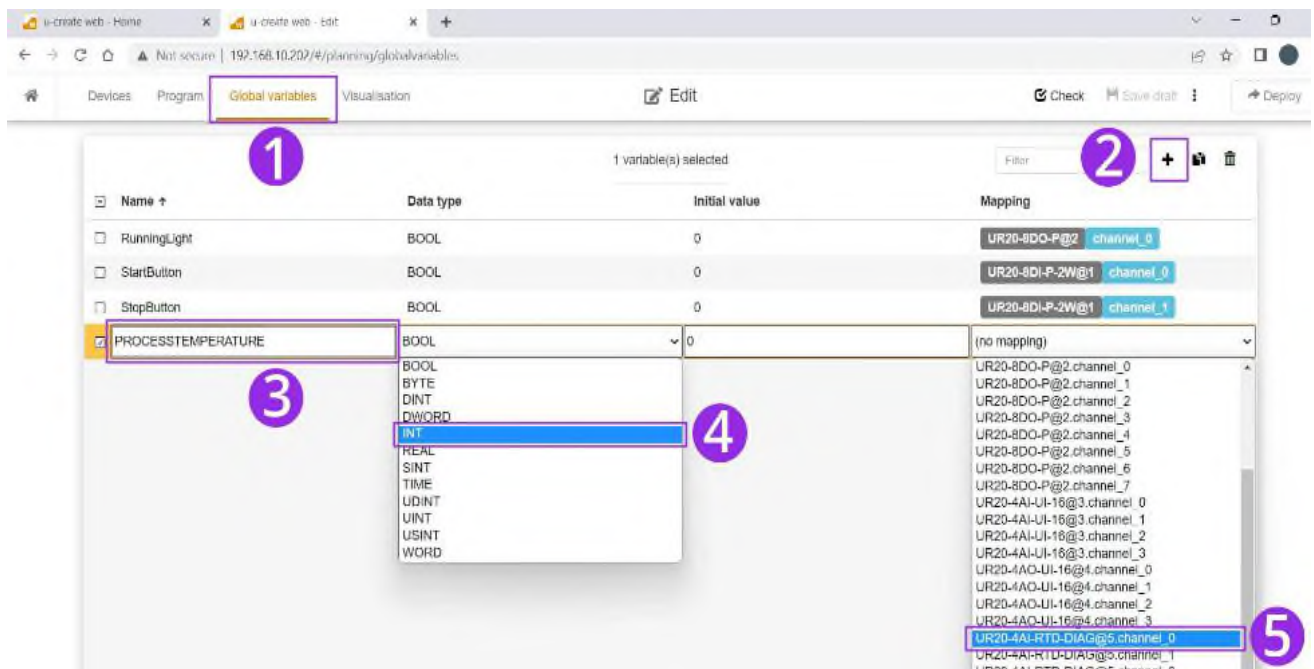


Рисунок 3.5 - Додавання глобальної змінної

Після цього потрібно пов'язати змінну з першим входом карти аналогового входу RTD Starter Kit. Цей вхід підключений до датчика температури Pt100 на стартовому наборі, який буде використовуватись протягом всієї демонстрації.

Живий перегляд:

Після налаштування змінних програми, потрібно натиснути кнопку "Розгорнути", щоб виконувати програму на контролері. Після короткої затримки програма успішно розгорнута, і я можу натиснути кнопку "Пуск", щоб активувати режим запуску контролера (рис.3.6, рис.3.7).

Після активації режиму запуску контролера, відбувається перехід до режиму реального часу, і після переключення на вкладку глобальних змінних я можу спостерігати за значеннями цих змінних і контролювати їх (рис.3.8, рис.3.9).

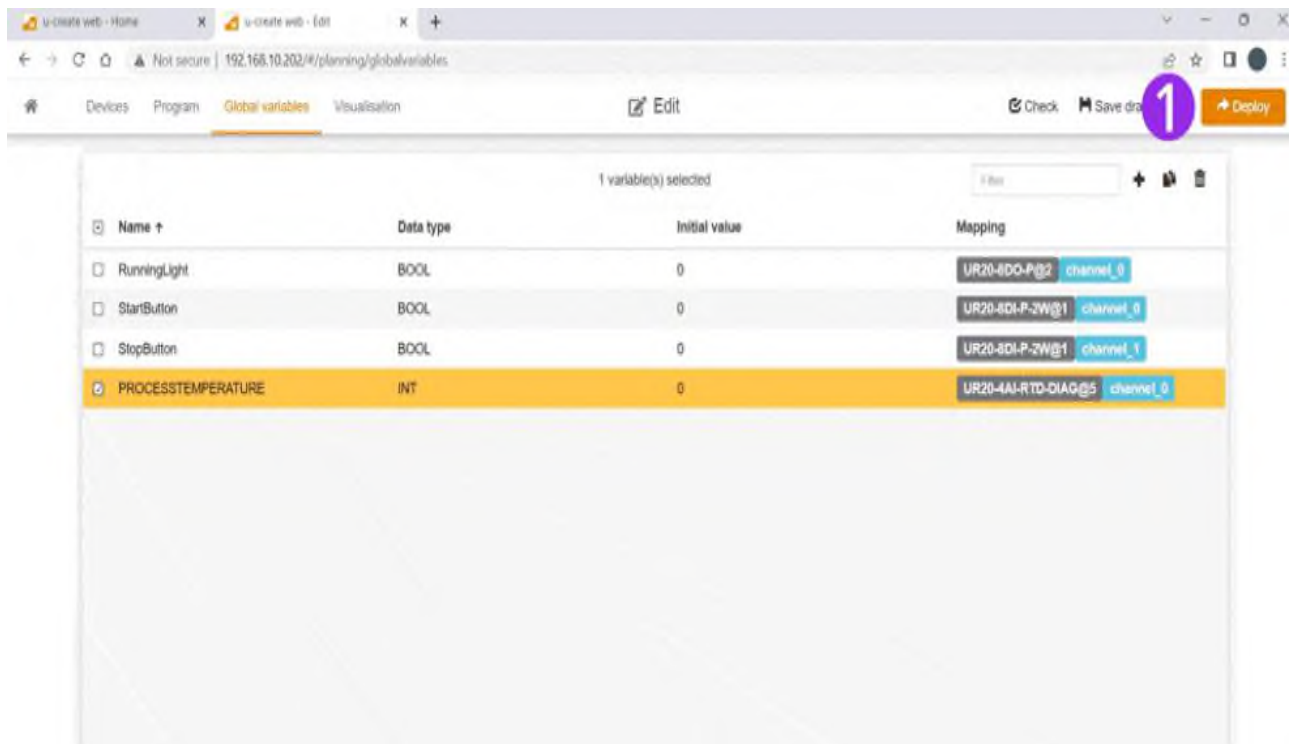


Рисунок 3.6 - Переведення контролера у режим запуску.

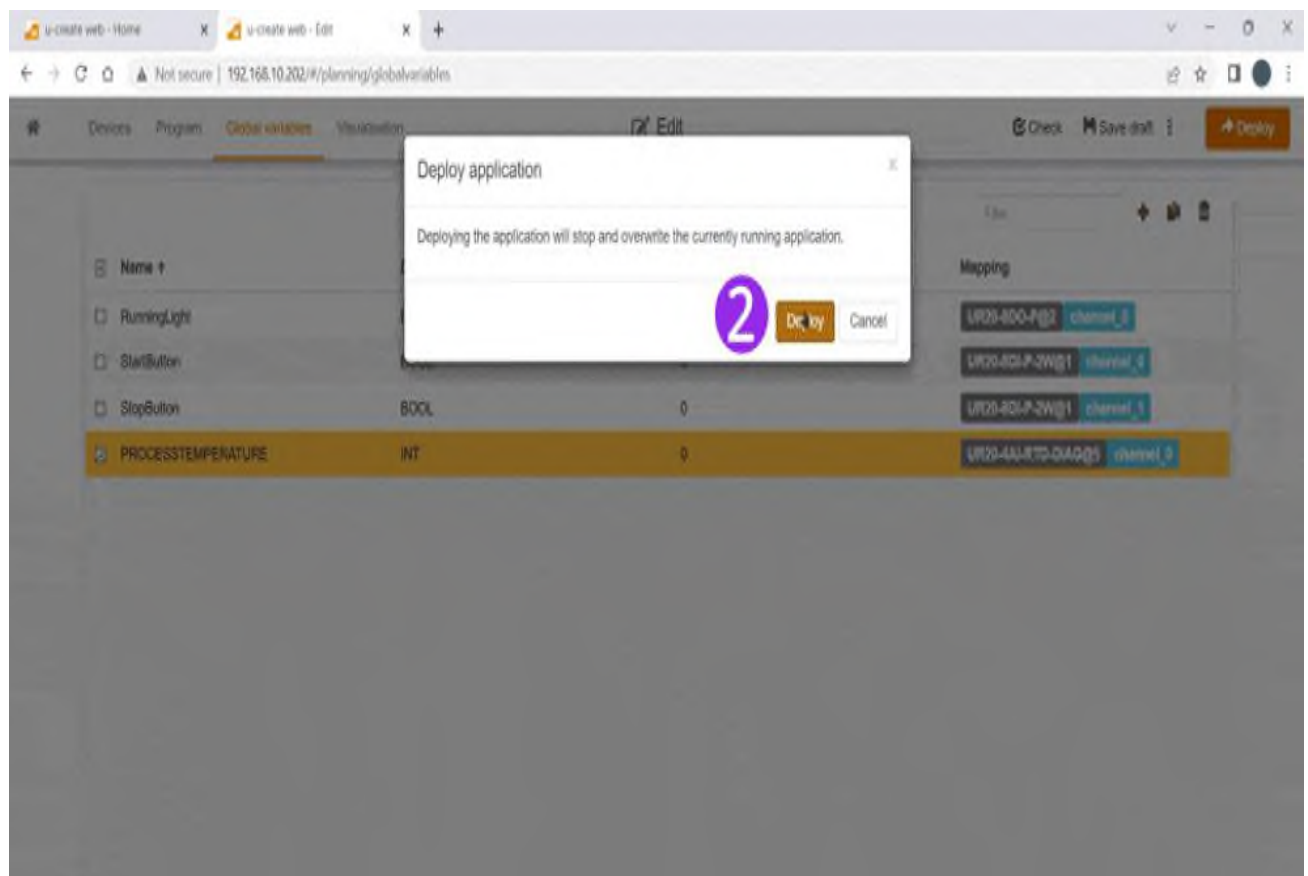


Рисунок 3.7 - Переведення контролера у режим запуску

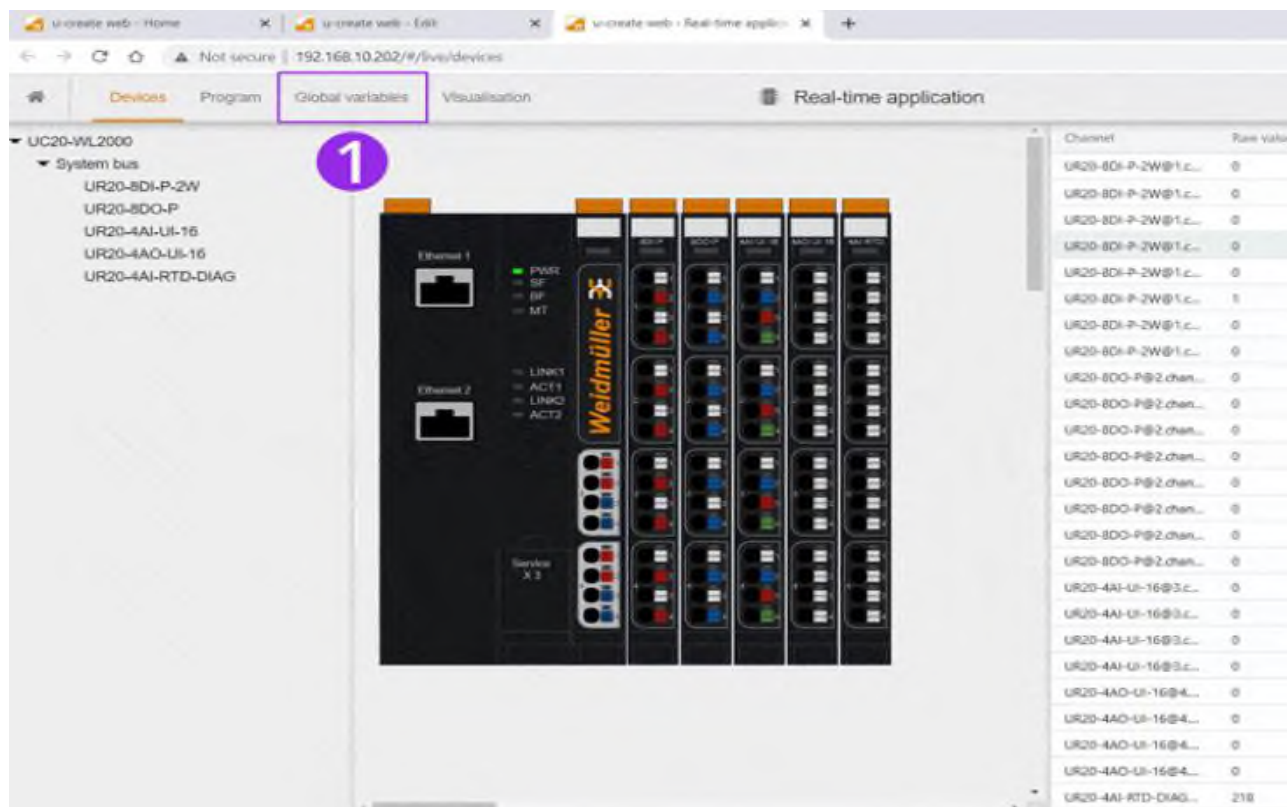


Рисунок 3.8 - Моніторинг значення глобальних змінних

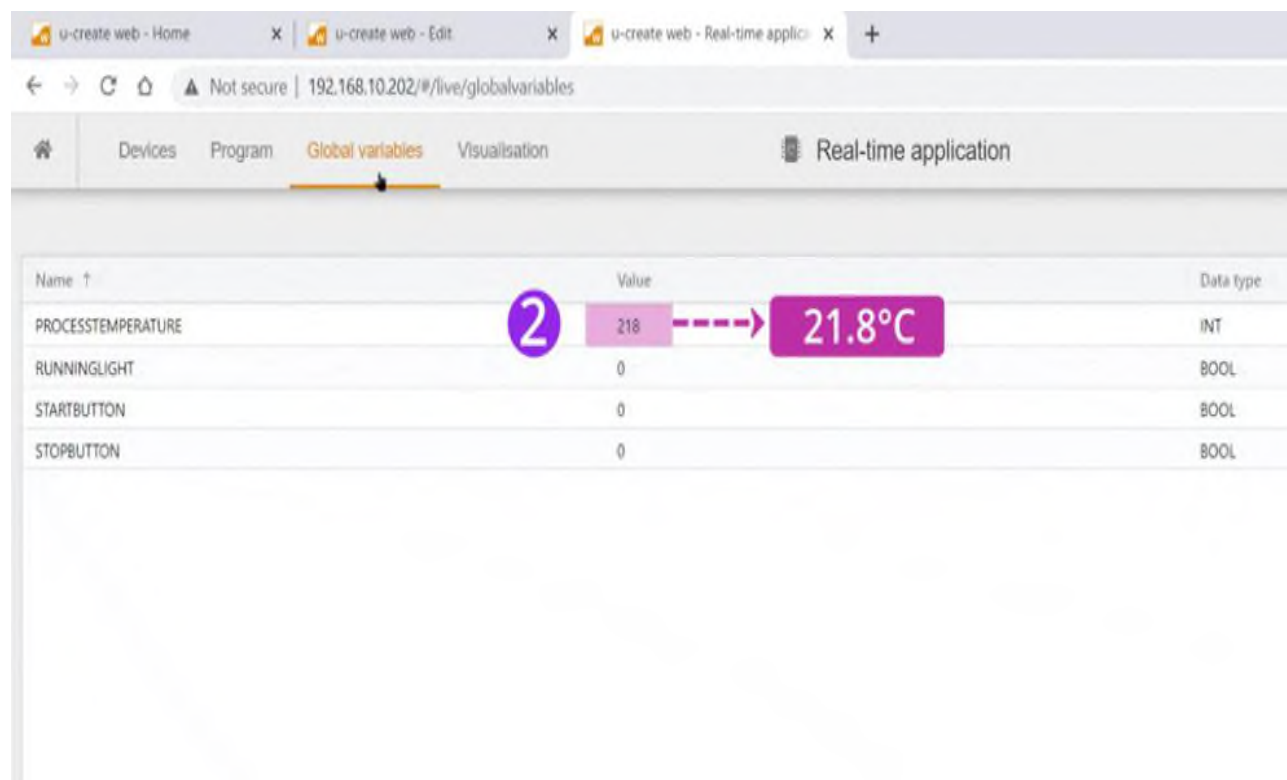


Рисунок 3.9 - Моніторинг значення глобальних змінних

3.3 Розробка програмно-технічного засобу на основі Node-RED

Тепер, коли змінні для програми налаштовані, потрібно створити програму Node-RED.

Повернувшись на домашню веб-сторінку u-create і перейшовши до розділу "Підключення, програми", можна легко отримати доступ до програми Node-RED.

Відкриється редактор Node-RED зі створеним потоком за замовчуванням під назвою Flow 1. Додаток Node-RED може мати кілька потоків.

На лівій панелі можна побачити палітру вузлів, які доступні для створення програми. Ця палітра включає вузли за замовчуванням, які постачаються разом з контролером u-control, а також можливість встановлення додаткових вузлів відповідно до потреб програми (рис.3.10, рис.3.11).

iodata-in вузол:

У даному випадку цей вузол служить для отримання значення глобальної змінної з контролера u-control, що є нашою основною метою. Перетягуємо цей вузол з палітри та опускаємо його на полотно (рис.3.12).

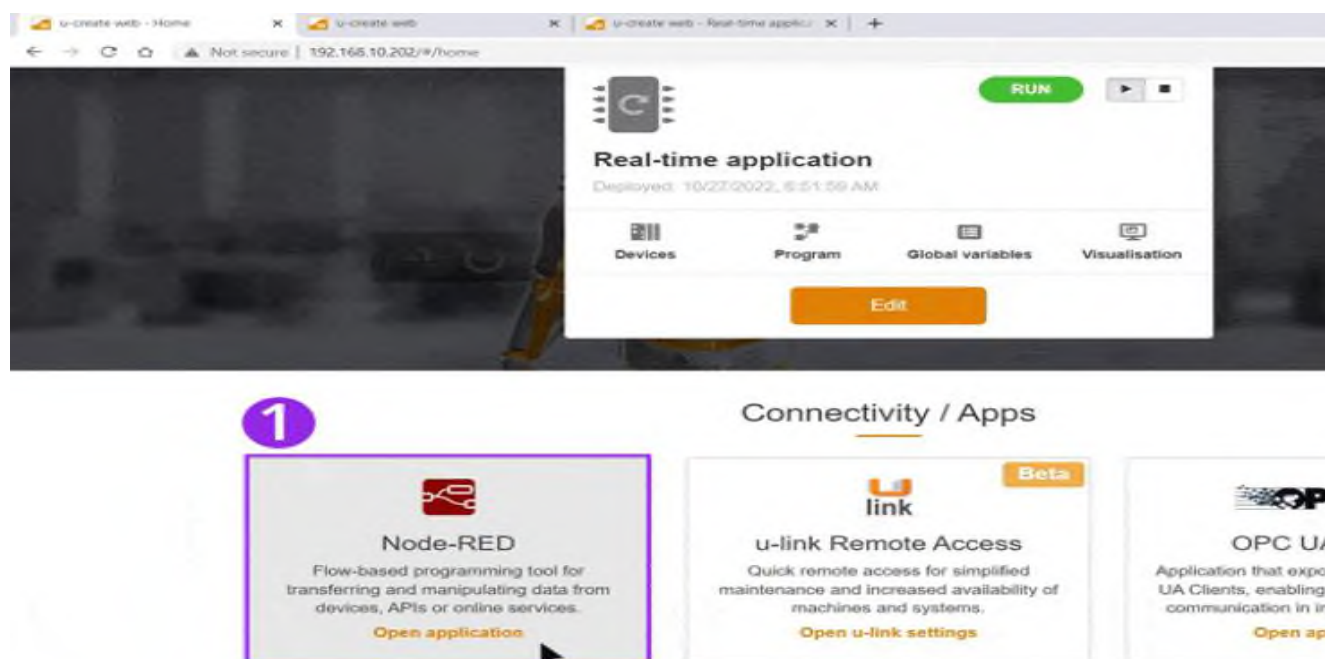


Рисунок 3.10 - Створення додатка Node-RED

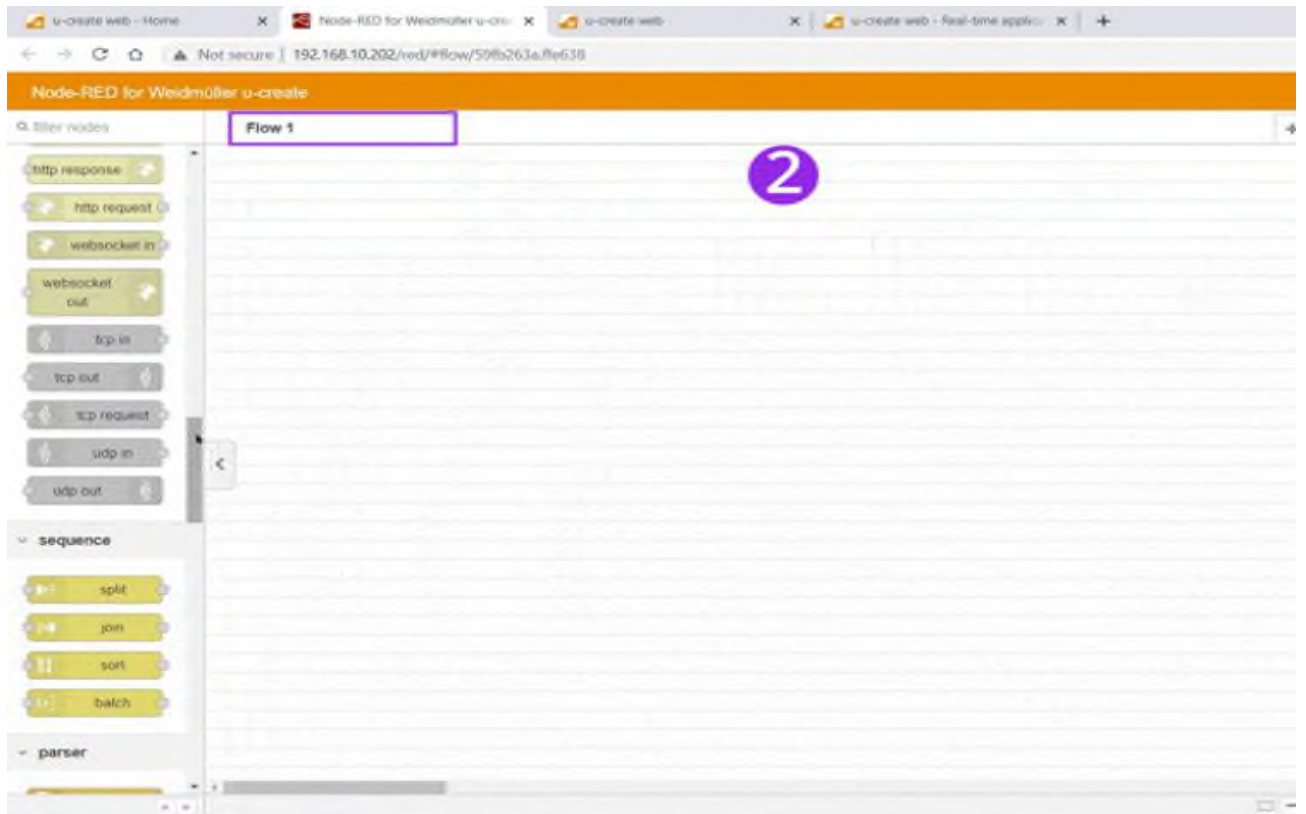


Рисунок 3.11 - Створення додатка Node-RED

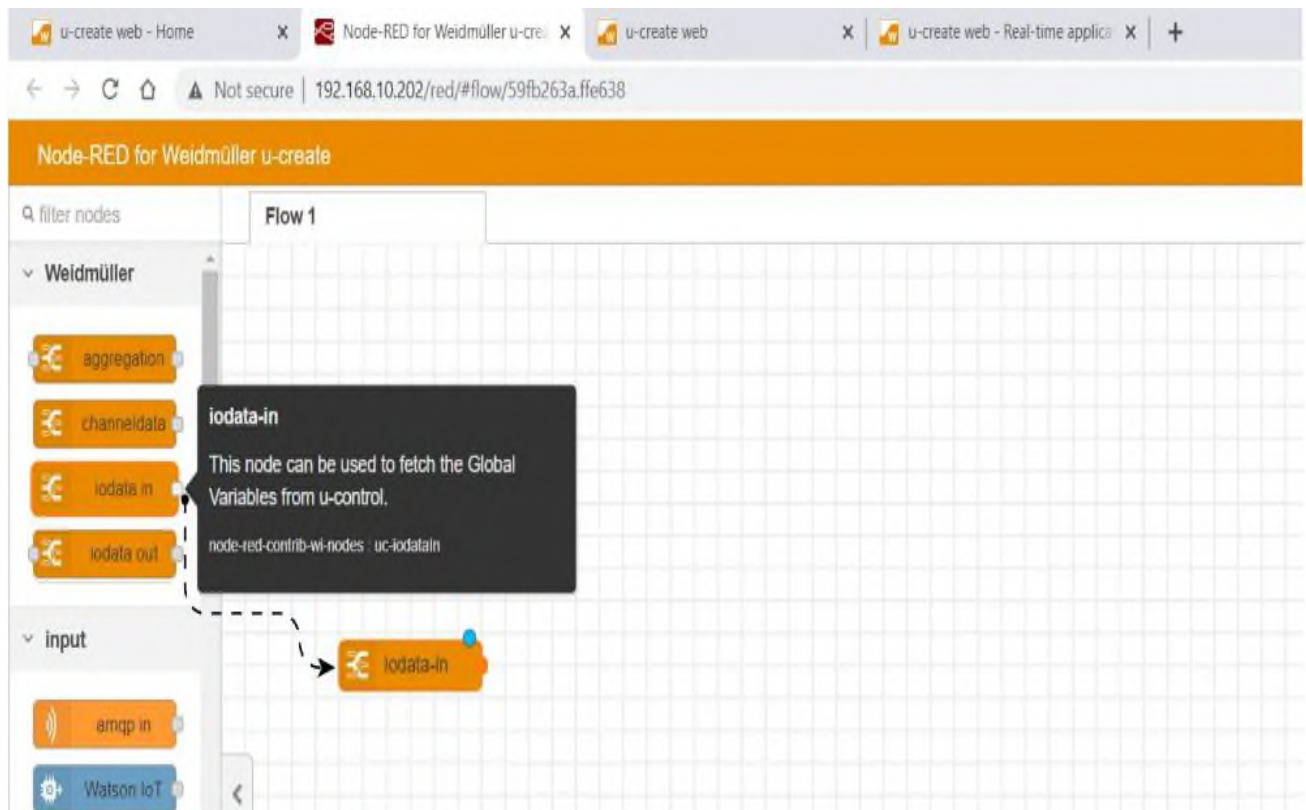


Рисунок 3.12 – iodata-in вузол

Можна відкрити властивості вузла, двічі клацнувши на вузлі. На панелі властивостей потрібно перевірити, що вузол отримує значення з правильної глобальної змінної та що час опитування становить 5 секунд (рис.3.13).

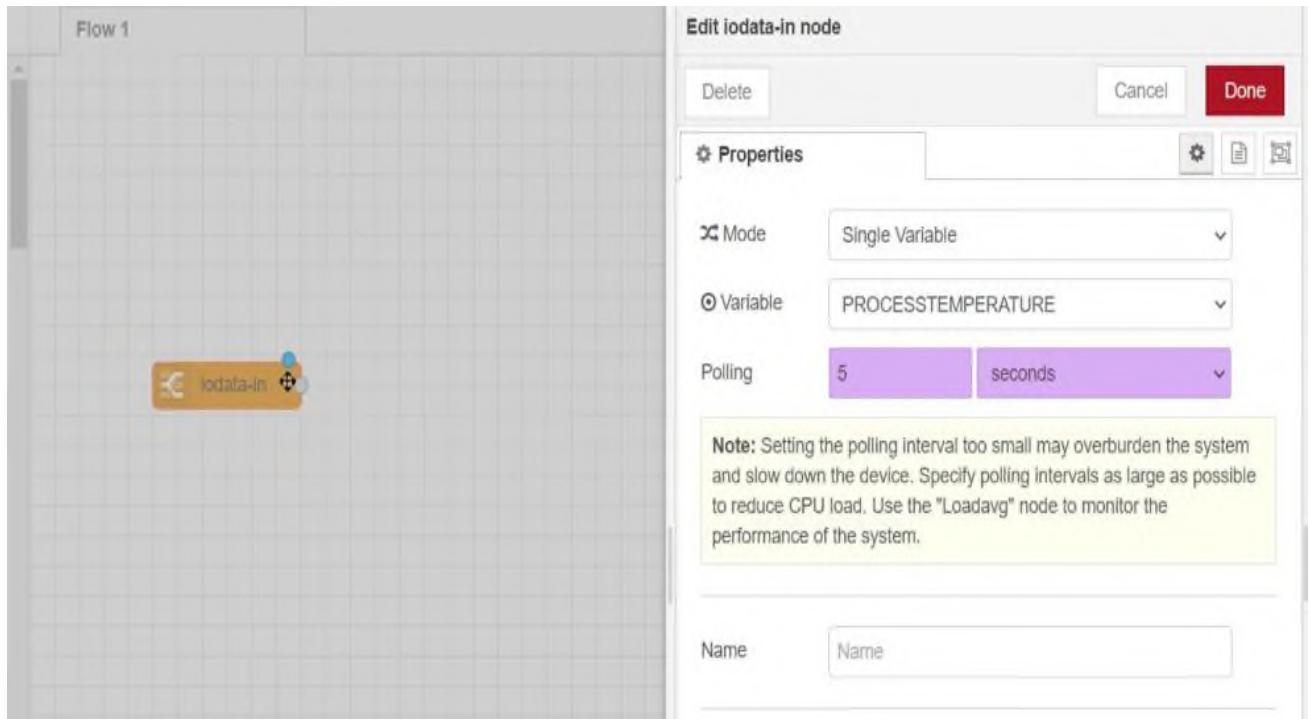


Рисунок 3.13 – Панель властивостей вузла

Це означає, що кожні 5 секунд цей вузол буде отримувати значення змінної ProcessTemperature і запускати потік. Варто зазначити, що хоча ми можемо встановити час опитування на рівні 50 мілісекунд, важливо розуміти, що програма Node-RED не є реальним часом і не призначена для керування в реальному часі. Зазвичай, програми Node-RED використовуються для збору даних протягом тривалих періодів часу.

Щоб оглянути структуру даних, що надходять з вузла, можливо використати вузол Debug. Для цього потрібно перейти до папки Common у палітрі вузлів і розміщую вузол Debug на робочому полотні. Щоб створити з'єднання між двома вузлами, потрібно перетягнути з маркера першого вузла та опустити на маркер другого вузла (рис.3.14).

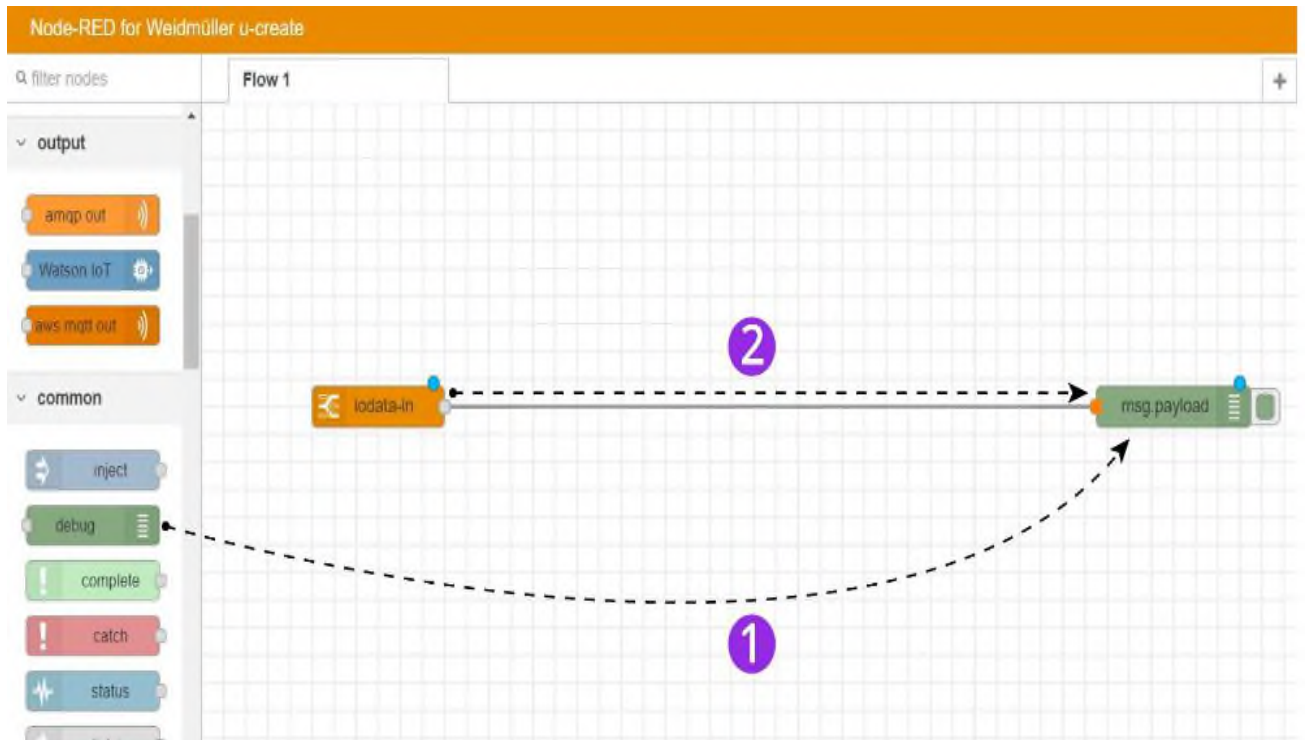


Рисунок 3.14 – Вузол налагодження

Синя крапка показує, що вузол ще не розгорнутий в межах поточної програми Node-RED.

Щоб розгорнути поточну програму, потрібно натиснути кнопку "Розгорнути".

Після розгортання програми я натискаю на символ помилки, щоб відкрити консоль налагодження. У цій консолі я можу переглядати повідомлення, які були зареєстровані вузлом Debug (рис.3.15, рис.3.16).

Можна помітити, що повідомлення, яке надходить з вузла iodata-in, представляє собою об'єкт з складною структурою даних.

Цей об'єкт містить тип даних глобальної змінної, її ім'я, значення та мітку часу отримання даних. В Node-RED повідомлення, які передаються між вузлами, завжди є об'єктами (рис.3.17).

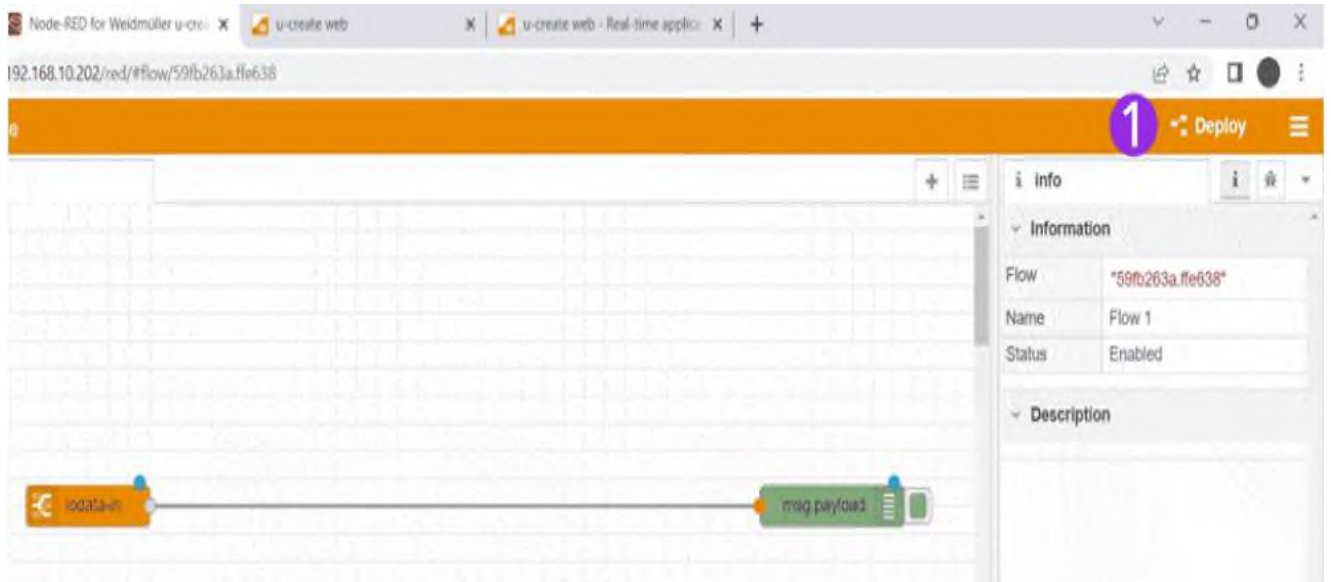


Рисунок 3.15 – Розгортання програми Node-RED

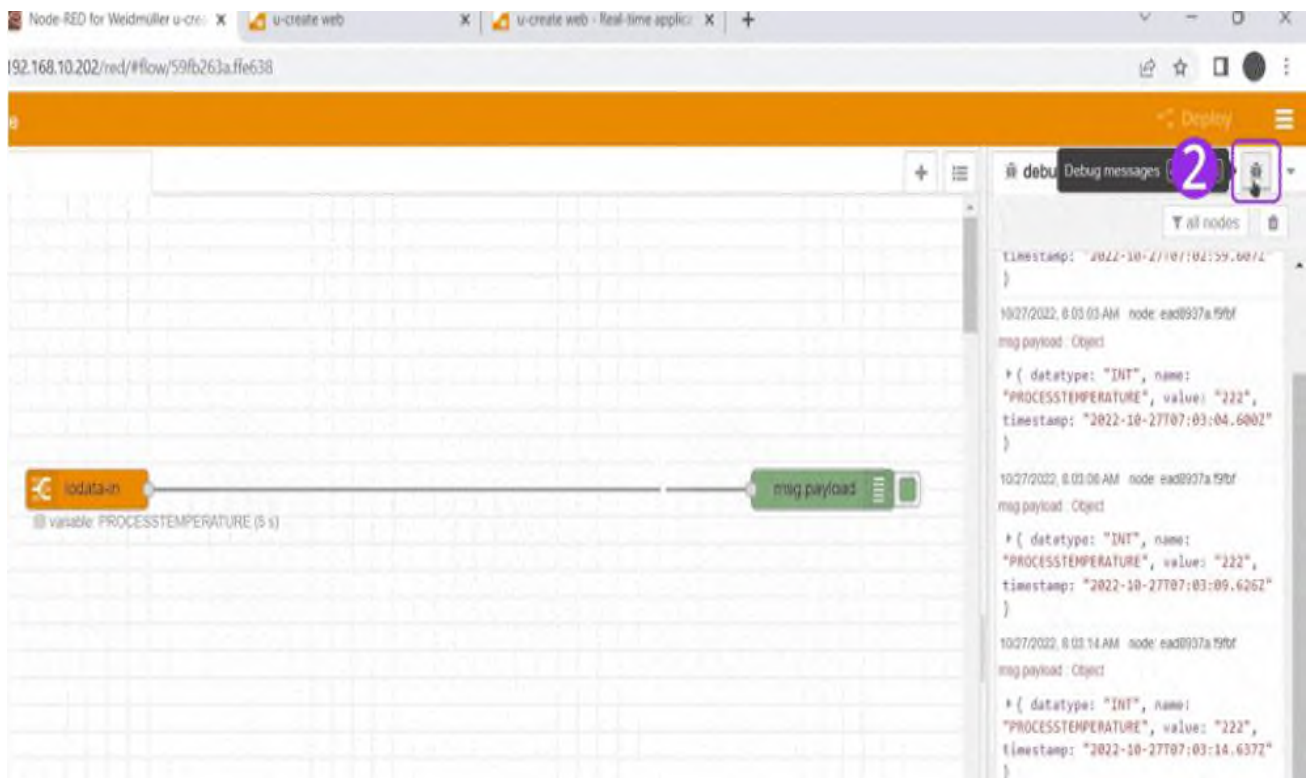


Рисунок 3.16 – Розгортання програми Node-RED

Функціональний вузол. Метою цієї програми є відправлення електронного листа на пошту технічному персоналу, коли значення процесу перетинає критичний діапазон.

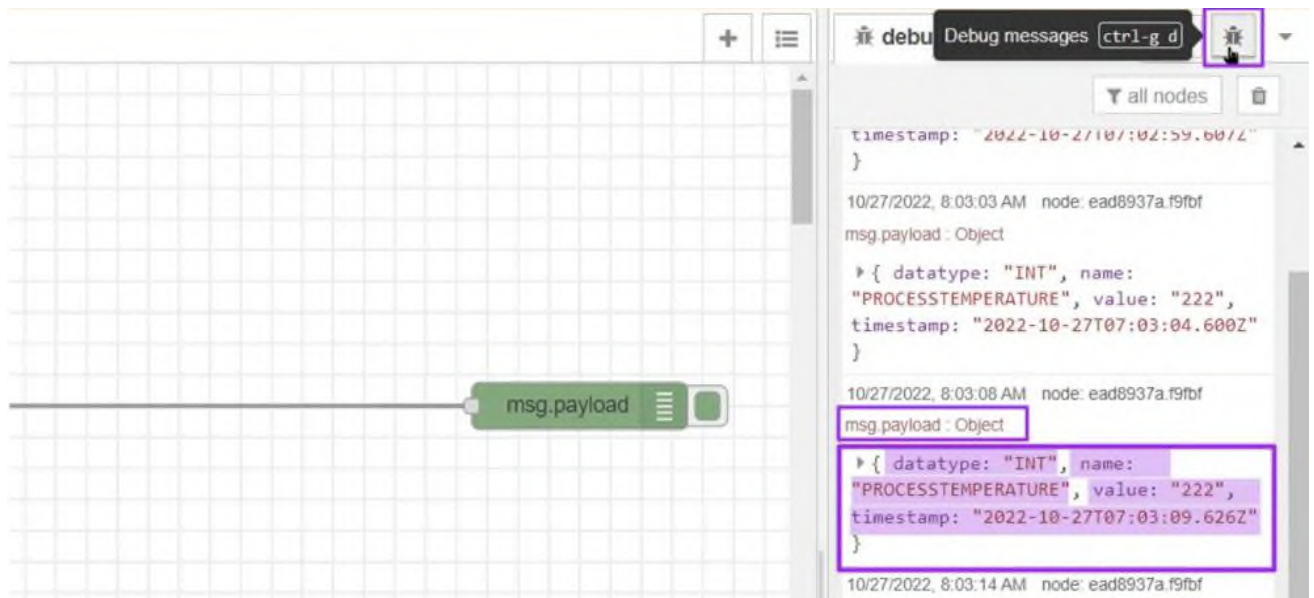


Рисунок 3.17 – Консоль налагодження

Для зручності визначення, чи досягло значення процесу критичного діапазону, можливо використати елемент тривоги, який буде включений до об'єкта повідомлення.

Для зміни вмісту повідомлення можна скористатись вузлом Function. Потрібно перетягнути вузол Функція з палітри та розмістити його на з'єднувальній лінії між існуючими вузлами. Вузол Функція додається між існуючими вузлами, а з'єднання між вузлами будуть створені автоматично.

За допомогою вузла Функція можна використовувати мову JavaScript для зміни вмісту об'єкта повідомлення (рис.3.18).

Для відкриття редактора JavaScript вузла потрібно подвійно клацнути на ньому.

У цьому прикладі ділиться необроблене значення від датчика на 10, щоб перетворити його в градуси Цельсія.

Після перетворення ми додаємо елемент нагадування до об'єкта повідомлення і встановлюємо значення цього елемента на 1, якщо значення процесу, яке ми контролюємо, перевищує 24, і 0, якщо воно менше 24 (рис.3.19).

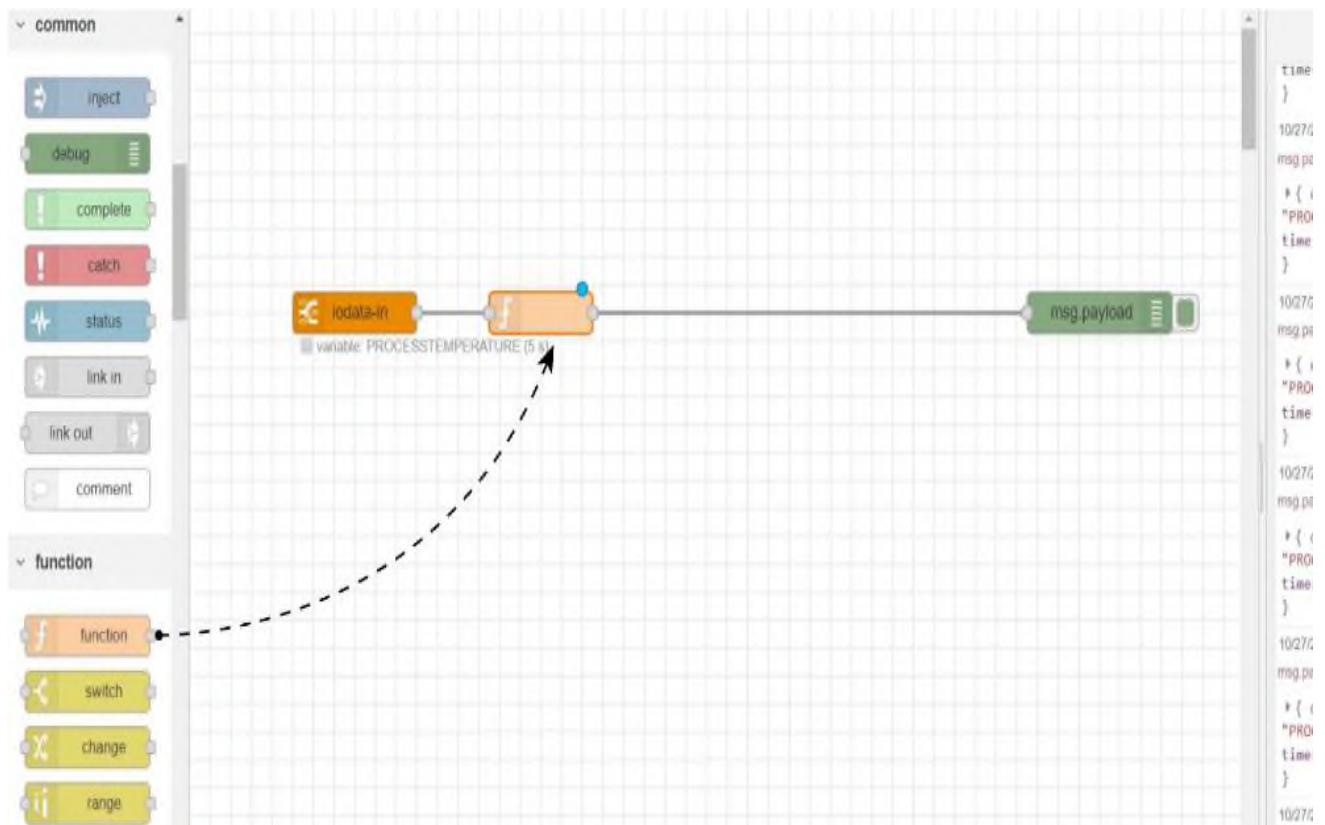


Рисунок 3.18 – Node-RED - вузол функції

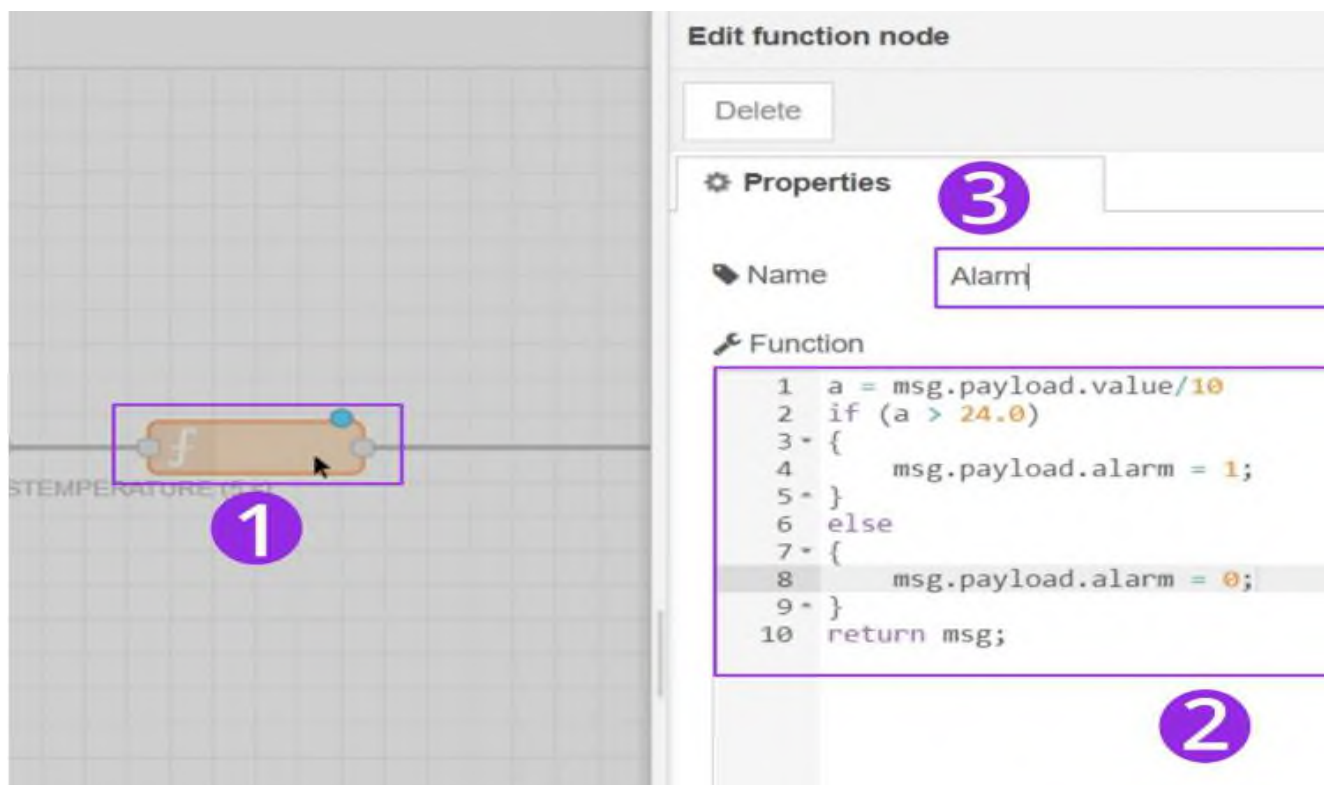


Рисунок 3.19 - Редактор JavaScript для вузла Function

Після розгортання додатку і закриття датчика температури, в вікні налагодження видно, що значення елемента сигналізації дорівнює 0, коли температура нижче 24 градусів, і 1, коли вона вище (рис.3.20).

Вузол RBE. Потрібно виконати надсилання електронної пошти лише тоді, коли значення елемента нагадування об'єкта повідомлення змінюється.

Для реалізації цієї функціональності потрібно використати вузол RBE, який блокує виконання потоку, якщо виконується певна умова. Потрібно перетягнути вузол RBE на полотно між вузлом Function і Debug, а потім двічі клацаю на вузол, щоб відкрити його властивості. У вікні властивостей потрібно налаштувати вузол RBE на блокування, якщо значення не змінюється (рис.3.21).

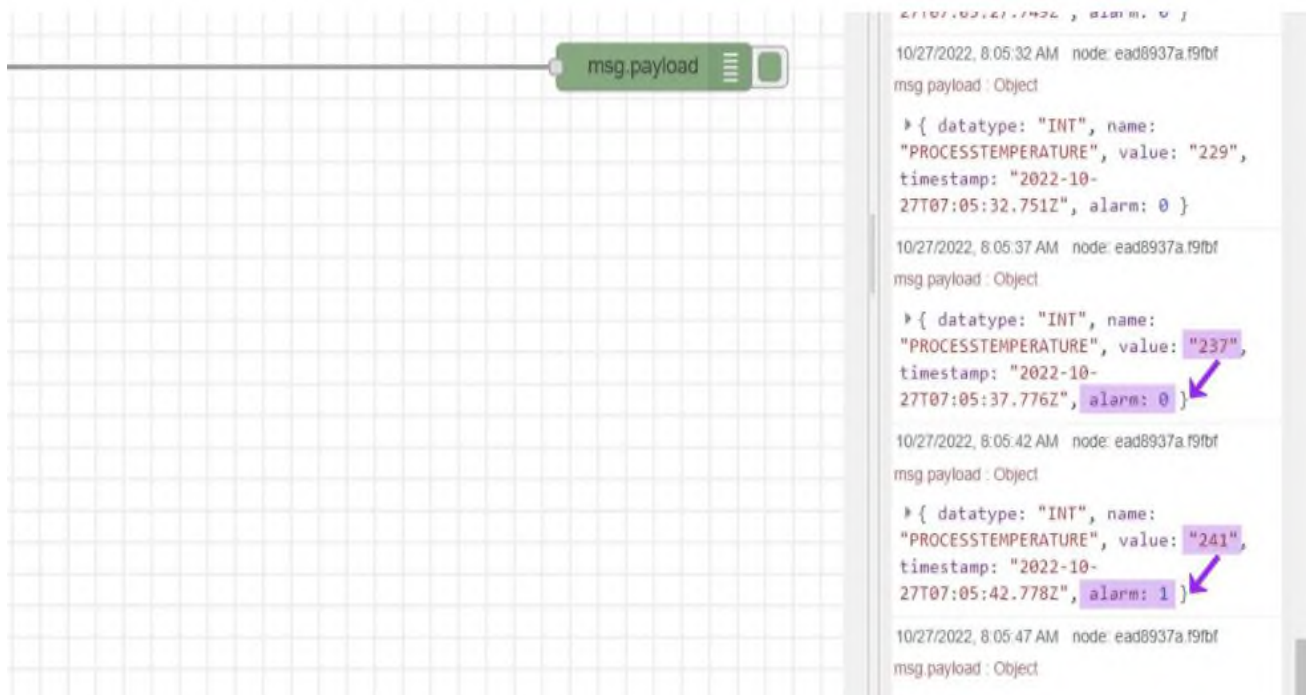


Рисунок 3.20 – Значення елемента сигналізації

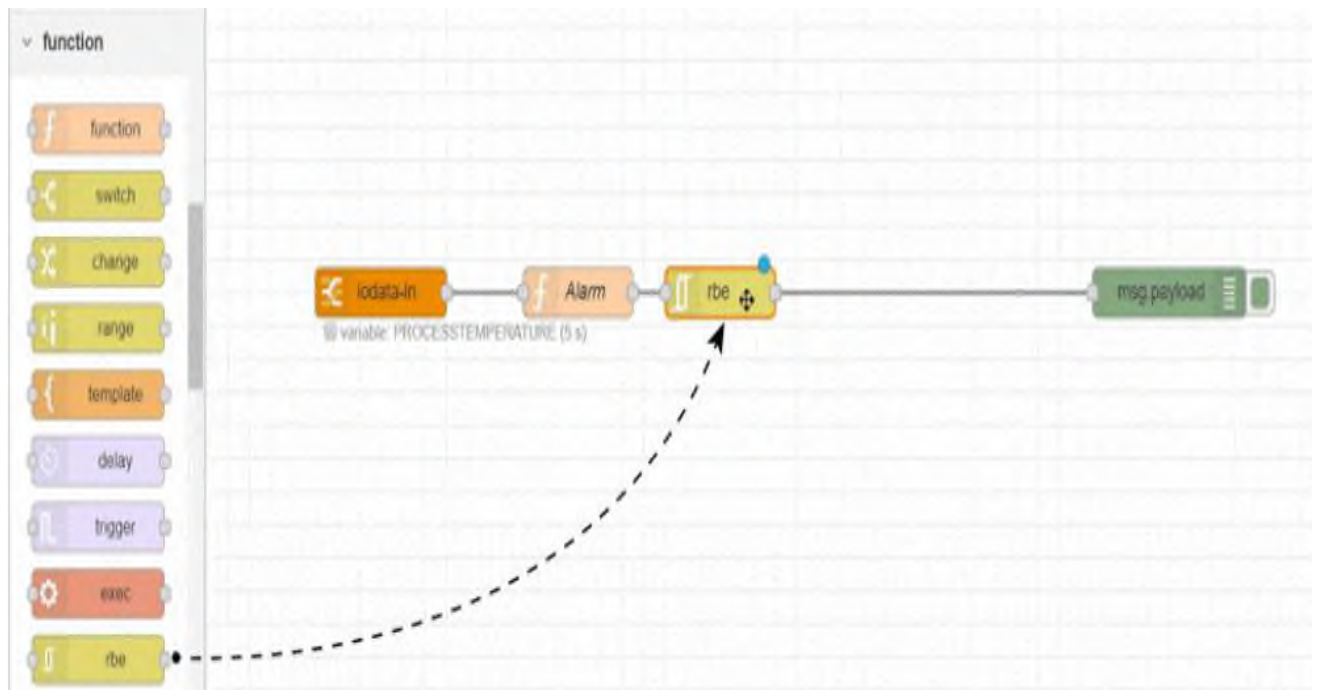


Рисунок 3.21 – Вузол RBE

Тепер, після розгортання програми видно, що вузол RBE зупиняє виконання потоку, доки не відбувається зміна значення елемента нагадування.

У вікні консолі налагодження я бачу, що реєструються повідомлення, коли значення елемента сигналізації стає 1 або 0.

Тепер наш потік виконується на основі події, а не залежно від пройденого часу (рис.3.22).

Вміст електронної пошти. Перш ніж надсилати електронний лист, потрібно налаштувати його вміст. Для цього необхідно використати інший вузол Function.

Знову потрібно перетягнути вузол функції з палітри, розміщую його на полотні між вузлом RBE і вузл Debug, і двічі клацаю на вузлі, щоб відкрити його властивості.

Потрібно додати фрагмент коду JavaScript тут, щоб оновити вміст об'єкта повідомлення залежно від активності сигналу тривоги (рис.3.23, рис.3.24).



Рисунок 3.22 – Вузол RBE - блокує, якщо значення не зміниться

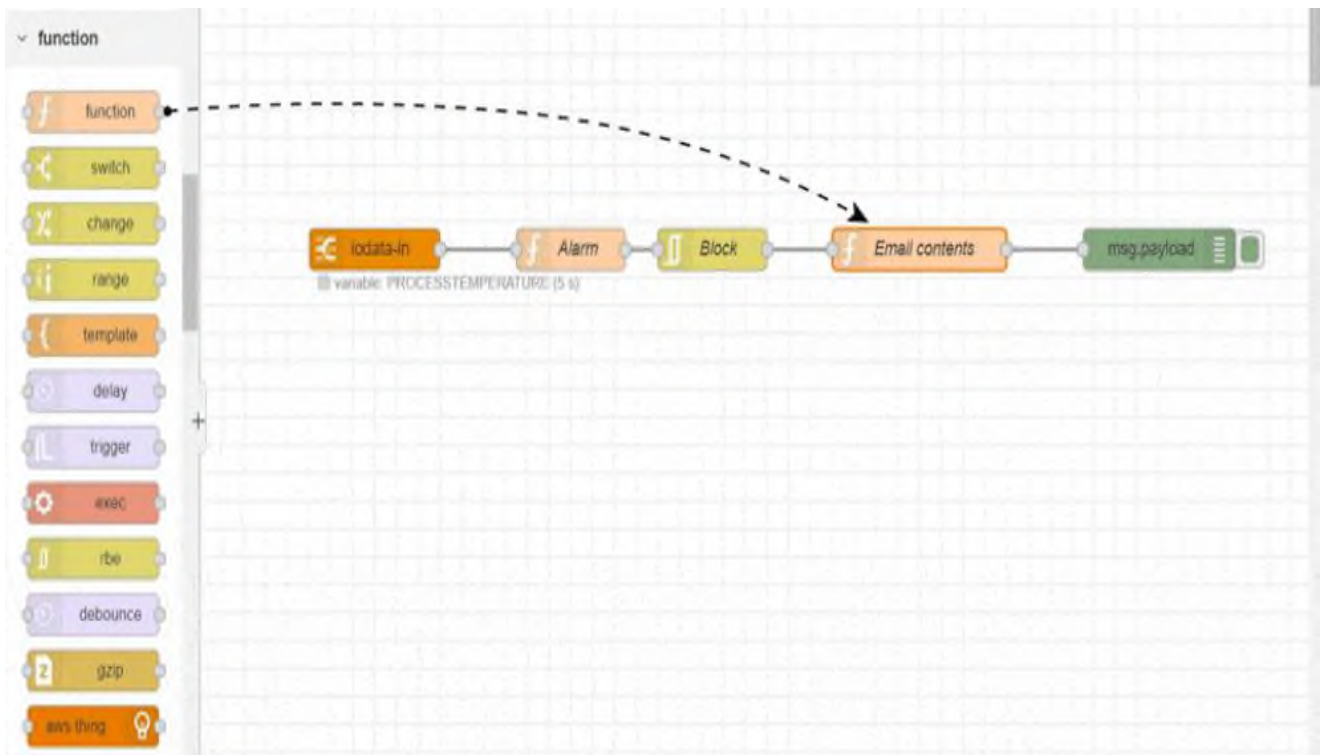


Рисунок 3.23 – Вміст електронної пошти

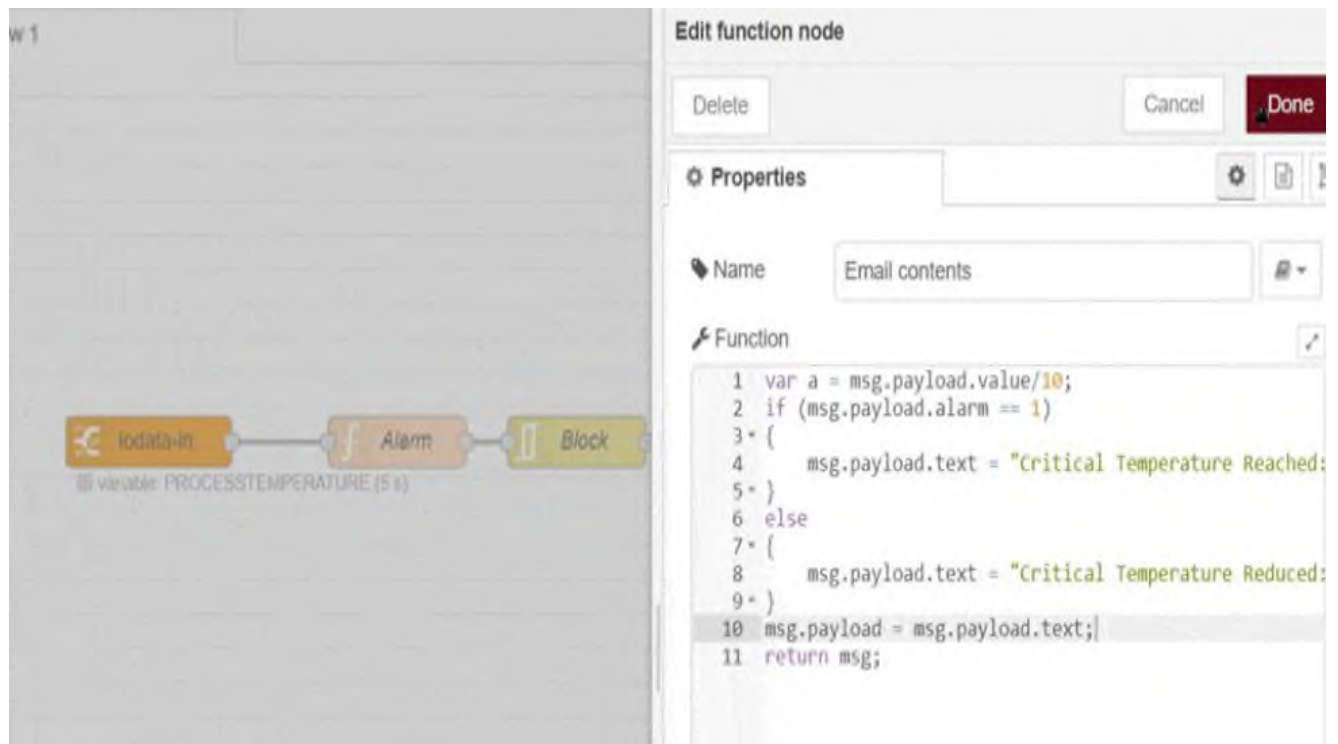


Рисунок 3.24 – Вміст електронної пошти

Після завершення налаштування функції можна закрити цей вузол і знову розгорнути програму.

У консолі налагодження видно, що вміст об'єкта повідомлення відповідає очікуванням, коли температура досягає критичного значення та повертається з критичного значення (рис.3.25).

Вузол електронної пошти. Далі необхідно підключити свій контролер u-control до Інтернету та додати вузол електронної пошти, щоб надсилати повідомлення електронною поштою.

Оскільки це останній вузол у потоці, не можливо додати його між вузлом Function і Debug.

Натомість, можна розмістити його на полотні та додаю додаткове з'єднання від вузла Function до вузла Email (рис.3.26).

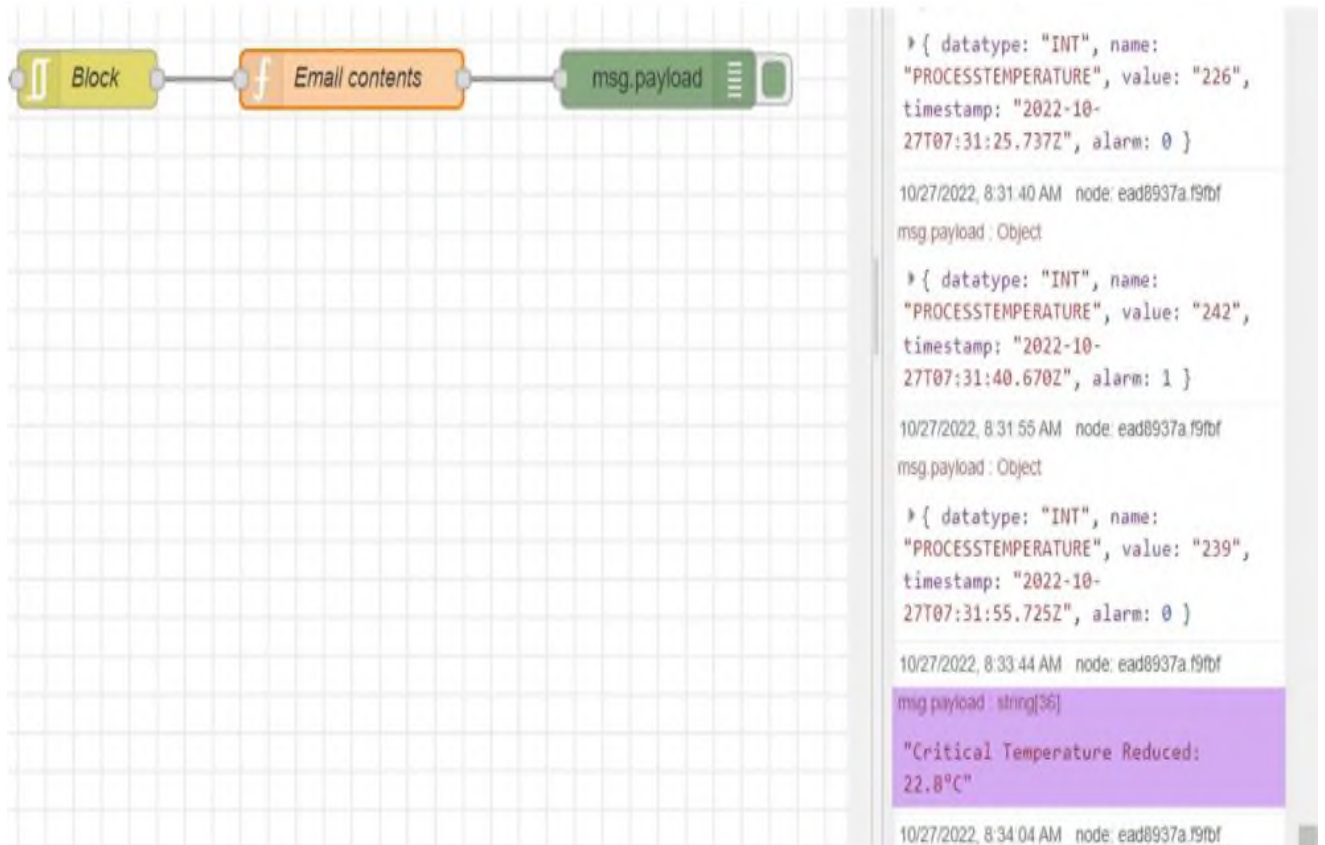


Рисунок 3.25 – Вміст об'єкта повідомлення

Потік можна розділити, щоб надсилати повідомлення від одного вузла до кількох вузлів протягом усього потоку.

Потрібно налаштувати вузол електронної пошти, використовуючи параметри, які показані тут (рис.3.27).

3.4 Тестування програмного засобу

Тепер можна закрити датчик температури, і при досягненні критичного значення температури, надходить повідомлення електронної пошти, яке реєструється на консолі налагодження (рис.3,28).

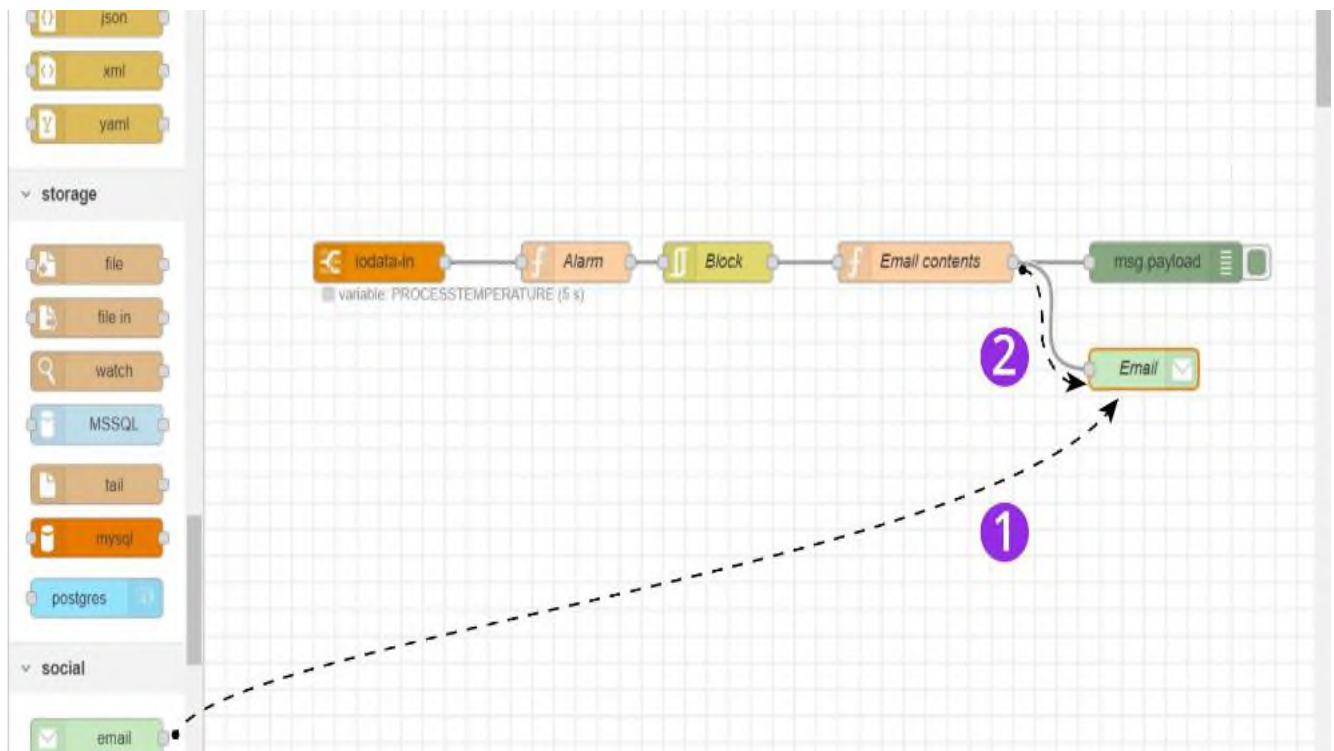


Рисунок 3.26 – Вузол електронної пошти

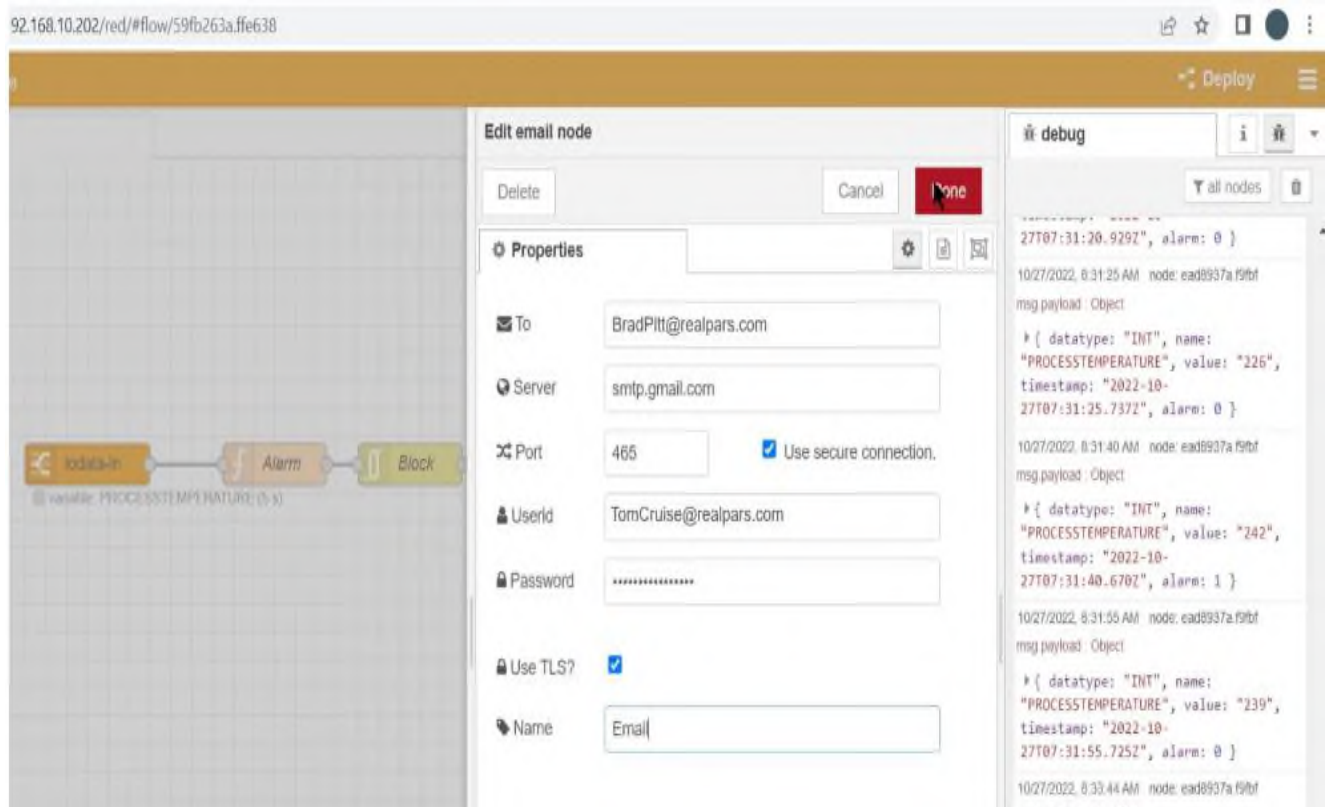


Рисунок 3.27 – Налаштувати вузол електронної пошти

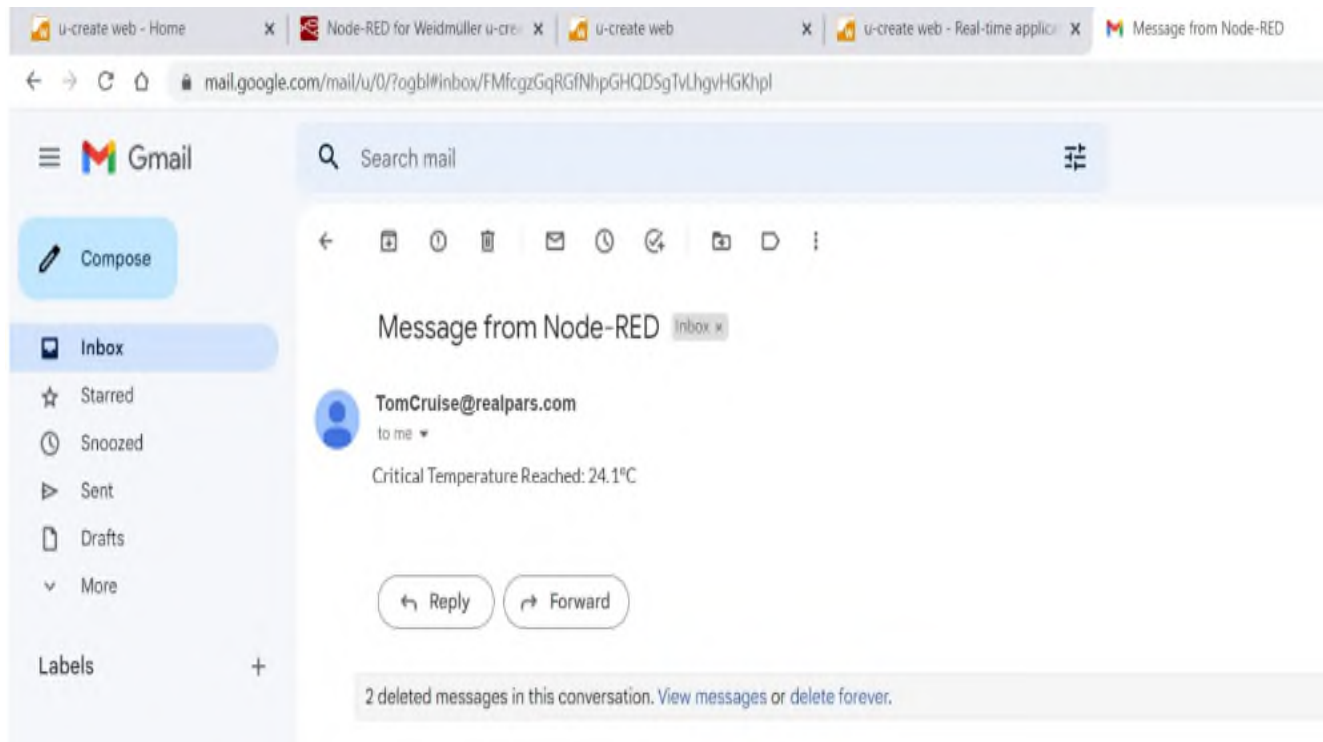


Рисунок 3.28 – Електронна пошта Node-RED

Коли перейти до папки вхідних повідомлень електронної пошти, видно, що надходить електронний лист від Node-RED з повідомленням про досягнення критичного значення температури процесу.

3.5 Висновок до розділу

У третьому розділі було проведено розробку та тестування програмно-технічного засобу, використовуючи u-create web, веб-середовище розробки для контролерів u-control.

У першому підрозділі були вивчені технічні характеристики та вимоги до апаратного забезпечення, необхідного для роботи програмно-технічного засобу. Було встановлено мінімальні вимоги до комп'ютера або сервера, на якому буде виконуватись Node-RED, а також перелік підтримуваних версій Node.js та пакетів залежностей.

У другому підрозділі було розроблено архітектуру програмно-технічного засобу, з урахуванням функціональних та нефункціональних вимог. Були визначені основні компоненти, модулі та їх взаємодію, що забезпечують роботу з Інтернетом речей на основі Node-RED.

У третьому підрозділі було проведено розробку програмно-технічного засобу, використовуючи Node-RED. Були виконані необхідні конфігурації, підключення до зовнішніх пристроїв та сервісів, налаштування функцій обробки даних та інше. Розроблено програму, яка надсилає сповіщення електронною поштою технікам з обслуговування, коли параметр процесу досягає критичного рівня.

У четвертому підрозділі було проведено тестування програмного засобу для перевірки його функціональності, стійкості та відповідності вимогам. Були виконані різноманітні тестові сценарії та аналіз результатів. Загальна робота програмно-технічного засобу була перевірена та підтверджена.

Отже, розділ "Програмно-апаратна реалізація та тестування програмно-технічного засобу" детально описує процес розробки, налаштування та перевірки програмно-технічного засобу на основі Node-RED для керування Інтернетом речей. Результатом цього розділу є створення працездатного засобу, який відповідає вимогам та виконує поставлені завдання.

ВИСНОВКИ

Кваліфікаційна робота присвячена розробці розширеного модуля керування Інтернетом речей, який базується на платформі Node-RED. Метою роботи є створення потужного інструменту для забезпечення зручного та ефективного керування мережею підключених пристроїв, зокрема в контексті розумного будинку.

У роботі проведений аналіз та обґрунтування вибору платформи Node-RED як основи для розробки програмно-технічного модуля. Використання Node-RED дозволяє забезпечити візуальне програмування, гнучкість налаштувань та широкі можливості інтеграції з різноманітними пристроями та сервісами Інтернету речей.

Розроблений програмно-технічний модуль включає в себе набір готових вузлів, які дозволяють взаємодіяти з різними типами пристроїв, зчитувати та передавати дані, контролювати режими роботи, реалізовувати логіку автоматизації та забезпечувати комунікацію з іншими системами.

У процесі розробки модуля було вирішено такі завдання, як інтеграція з різними протоколами зв'язку, розробка спеціалізованих вузлів для обробки та аналізу даних, створення зручного інтерфейсу користувача для керування та моніторингу пристроїв.

Для перевірки ефективності та функціональності модуля були проведені тестування. Результати досліджень підтвердили правильність вибору платформи Node-RED і продемонстрували високу продуктивність та надійність розробленого модуля. Встановлення та конфігурація модуля на практиці були простими та зручними, що підкреслює його високий рівень доступності для розробників та користувачів.

Отримані результати дозволяють стверджувати, що програмно-технічний модуль керування Інтернетом речей на основі Node-RED є ефективним та надійним інструментом для розробки рішень у сфері Інтернету речей. Він

забезпечує широкі можливості для інтеграції та керування різноманітними пристроями, спрощує розробку складних автоматизованих систем та підвищує рівень зручності та ефективності управління.

Результати даної кваліфікаційної роботи можуть бути використані для подальшого розширення та вдосконалення модуля, а також впроваджені в реальних проектах з Інтернету речей. Високий потенціал Node-RED та його гнучкість дозволяють широкому колу розробників та спеціалістів у сфері Інтернету речей ефективно використовувати цей модуль для реалізації інноваційних та прогресивних рішень.

Загалом, кваліфікаційної робота засвідчує, що програмно-технічний модуль керування Інтернетом речей на основі Node-RED є важливим кроком у напрямку створення розумних та зв'язаних середовищ, що пропонують широкі можливості для забезпечення комфорту, безпеки та енергоефективності. Цей модуль сприяє розвитку Інтернету речей та відкриває нові перспективи для впровадження інноваційних технологій у життя людей.

					КвРКІ 2001127.20.01.01 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Sinclair B. IoT Inc: How Your Company Can Use the Internet of Things to Win in the Outcome Economy, 2018. 245 с.
2. Kranz M. Building the Internet of Things: Implement New Business Models, Disrupt Competitors, Transform Your Industry. 2019. 161-163 с.
3. Пупена О. Довідник Node-RED з елементами опису Java Script, JSON, JSONata, 2020, 166 с. URL: https://drive.google.com/file/d/1tbhv1j-tiUGpIIAO4kWIInCRXJh0ZIqf/view?fbclid=IwAR2yP3egoT_Eie6nvtTQbZZDSVUyID3o-nmGTGHfgICvN8QZ4BDITM9X97UNode-RED (дата звернення: 15.04.2023)
4. Jamthe S. IoT Disruptions: The Internet of Things Innovations and Jobs, 2019. 32 с.
5. Lea P., Edge J. IoT and Edge Computing for Architects: Implementing edge analytics and machine learning with Azure IoT Edge, 2020. 475 с.
6. Shovic J. Raspberry Pi IoT Projects: Prototyping Experiments for Makers, 2020. 75 с.
7. Oliva Ramos R. Internet of Things Programming with JavaScript: Build interactive and smart IoT projects with JavaScript and Node.js, 2021. 142 с.
8. Russell B., Van Duren D. Practical Internet of Things Security: Design, deploy and secure your IoT infrastructure, 2021. 273 с.
9. Lea P. Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security, 2021. 48 с.
10. Pulver T. Hands-On Internet of Things with MQTT: Build connected IoT devices with Arduino and MQ Telemetry Transport (MQTT), 2021. 261 с.
11. Mazzei D. Designing and Building an IoT Product: The ultimate guide to building smart devices and connected products from concept to mass production, 2022. 33 с.

					КвРКІ 2001127.20.01.01 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

12. Madiseti V., Bahga A. Internet of Things: A Hands-On Approach for Beginners, 2022. 201 с.
13. Довідник Node-RED українською мовою. URL: <https://pupenasan.github.io/NodeREDGuidUKR/> (дата звернення 02.04.2023)
14. Ravulavaru A. Practical Internet of Things with JavaScript: Build standalone exciting IoT projects with JavaScript, 2018. 162 с.
15. Delsing J. IoT Automation: Arrowhead Framework, 2018. 105 с.
16. Bhattacharjee S. Practical Industrial Internet of Things Security: A practitioner's guide to securing connected industries, 2019. 258 с.
17. Rao M. Internet of Things with Raspberry Pi 3: Leverage the power of Raspberry Pi 3 and JavaScript to build exciting IoT projects, 2018. 192 с.
18. Node RED Programming Guide. URL: <https://noderedguide.com/> (дата звернення 15.03.2023)
19. Sewtz M. Learning Node-RED, 2018. 273 с.
20. Veneri G. Hands-On Industrial Internet of Things: Create a powerful Industrial IoT infrastructure using Industry 4.0, 2019. 402 с.
21. Liddell J. Building the Web of Things with Node-RED, 2020. 227 с.
22. Najera A. Node-RED Development: Create your own sensor network, 2021. 149 с.
23. " Hernandez R. Node-RED Programming Guide: A Comprehensive Guide to Learning Node-RED Programming for Beginners, 2019. 312 с.
24. Офіційний веб-сайт Node-RED. URL: <https://nodered.org/> (дата звернення 09.03.2023)
25. Cookbook Node-RED. URL: <https://cookbook.nodered.org/> (дата звернення 17.03.2023)
26. Node-RED Library. URL: <https://flows.nodered.org/> (дата звернення 14.04.2023)
27. Node-RED: How it simplified my IoT project - and how YOU can rapidly prototype for the Internet of Things. URL: <https://www.bouvet.no/bouvet->

[deler/utbrudd/node-red-how-it-simplified-my-iot-project-and-how-you-can-rapidly-prototype-for-the-internet-of-things](#) (дата звернення 05.04.2023)

28. Virtual Internet of Things Laboratory Using Node-RED. URL: <https://www.intechopen.com/chapters/81242> (дата звернення 09.03.2023)

29. Introduction to the Internet of Things (IoT), Node Red, and the Raspberry Pi. URL: https://mlhale.github.io/nebraska-gencyber-modules/introduction_to_iot_node_red/README/ (дата звернення 15.04.2023)

30. Lea R. Node-RED: Lecture 7 – Dashboards and UI techniques for Node-RED. URL: <http://noderedguide.com/lecture-7-dashboards-and-ui-techniques-for-node-red/> (дата звернення 04.04.2023)

31. Gaurav K. Using Node-RED for IoT and Edge Computing. URL: <https://www.opensourceforu.com/2022/02/using-node-red-for-iot-and-edge-computing/>. (дата звернення 19.04.2023)

32. Node-RED Cookbook на GitHub. URL: <https://github.com/node-red/cookbook.nodered.org> (дата звернення 20.04.2023)

33. Node-RED Wiki. URL: <https://github.com/node-red/node-red/wiki> (дата звернення 18.03.2023)

34. Nagino T. Practical Node-RED Programming, 2021. 11-12 с.

35. Офіційна документація Node-RED. URL: <https://nodered.org/docs/> (Дата звернення 01.03.2023)

36. Process MQTT data with Node-RED. URL: <https://www.emqx.com/en/blog/using-node-red-to-process-mqtt-data> (дата звернення 03.04.2023)

37. Installing Node-Red on a Raspberry Pi. URL: <https://pimylifeup.com/node-red-raspberry-pi/> (дата звернення 13.04.2023)

38. Node-RED & Raspberry Tutorial: How to create an Automatic Counter App. URL: https://www.industrialshields.com/es_ES/blog/raspberry-pi-para-la-industria-26/post/ii-node-red-raspberry-tutorial-how-to-create-an-automatic-counter-app-384 (дата звернення 18.03.2023)

					КвПКІ 2001127.20.01.01 ПЗ	Арк. 63
Зм.	Арк.	№ докум.	Підпис	Дата		

39. Machine Learning with Node-RED+Arduino. URL: <https://sriparnaiot.wordpress.com/machine-learning-with-node-redarduino/> (дата звернення 12.03.2023)

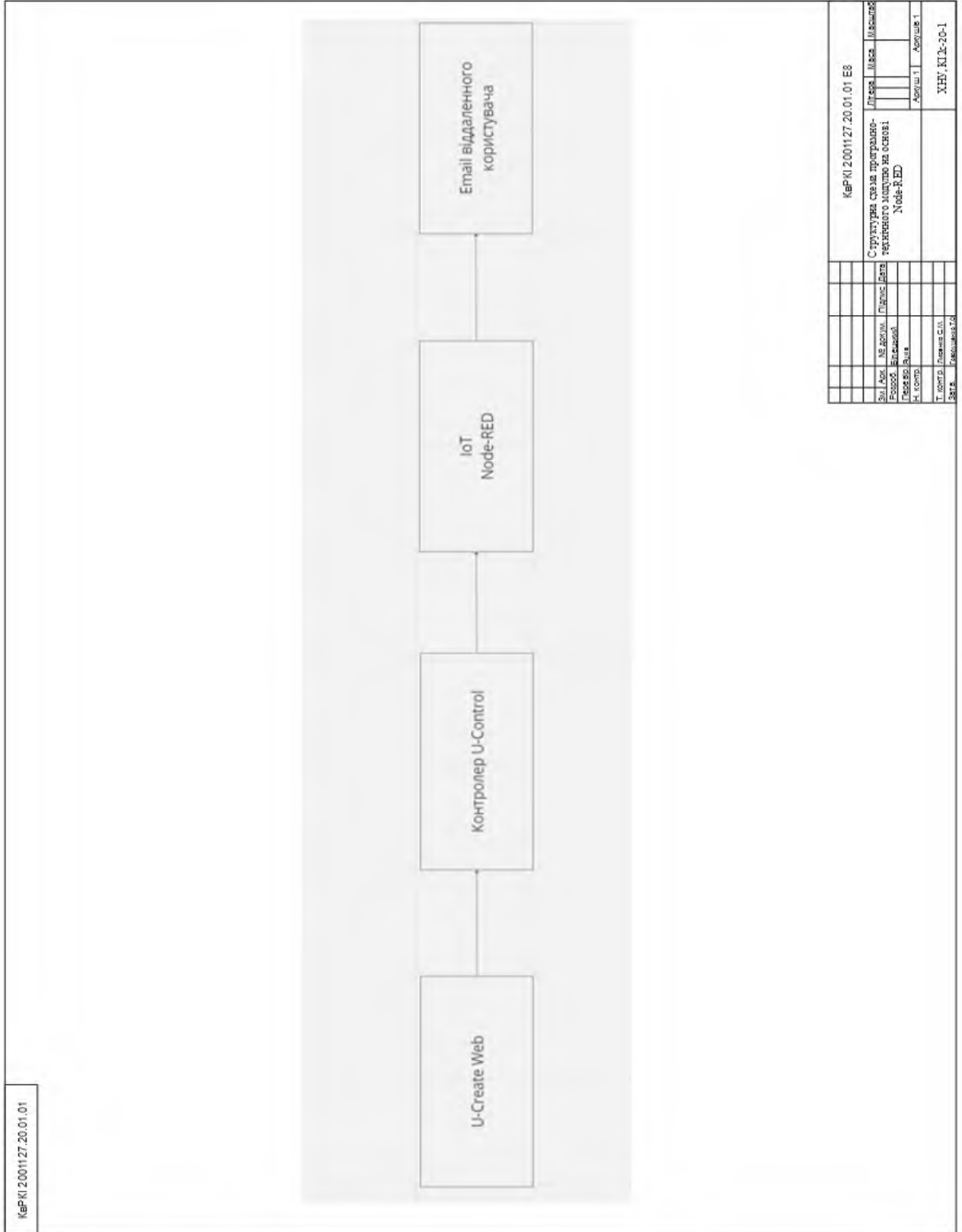
40. Getting Started with Node-RED Dashboard on Raspberry Pi. URL: <https://randomnerdtutorials.com/getting-started-node-red-dashboard/> (дата звернення 16.04.2023)

					КвРКІ 2001127.20.01.01 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

Додаток А

(обовязковий)

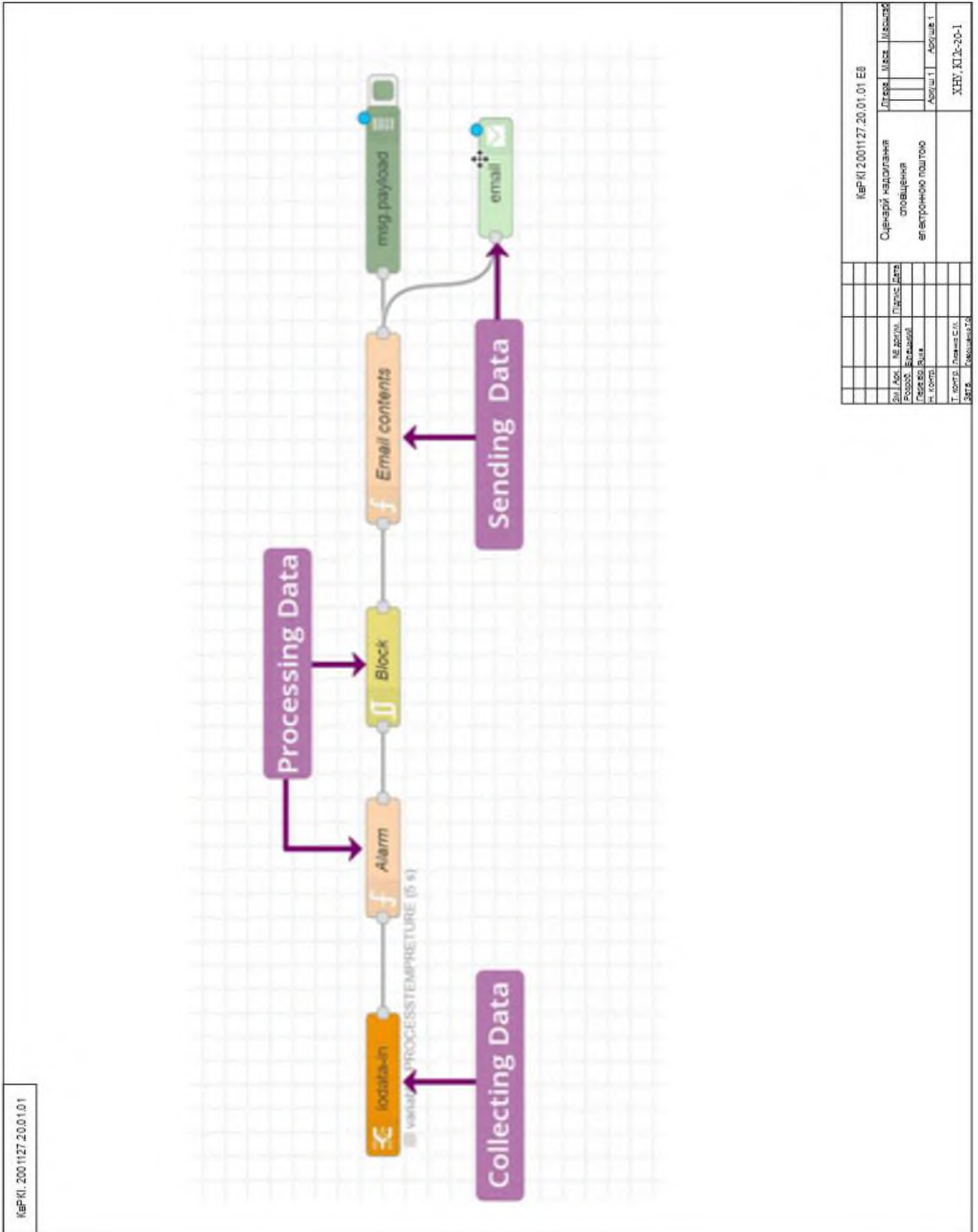
Копія креслення «Структурна схема програмно-технічного модулю на основі Node-RED»



Додаток Б

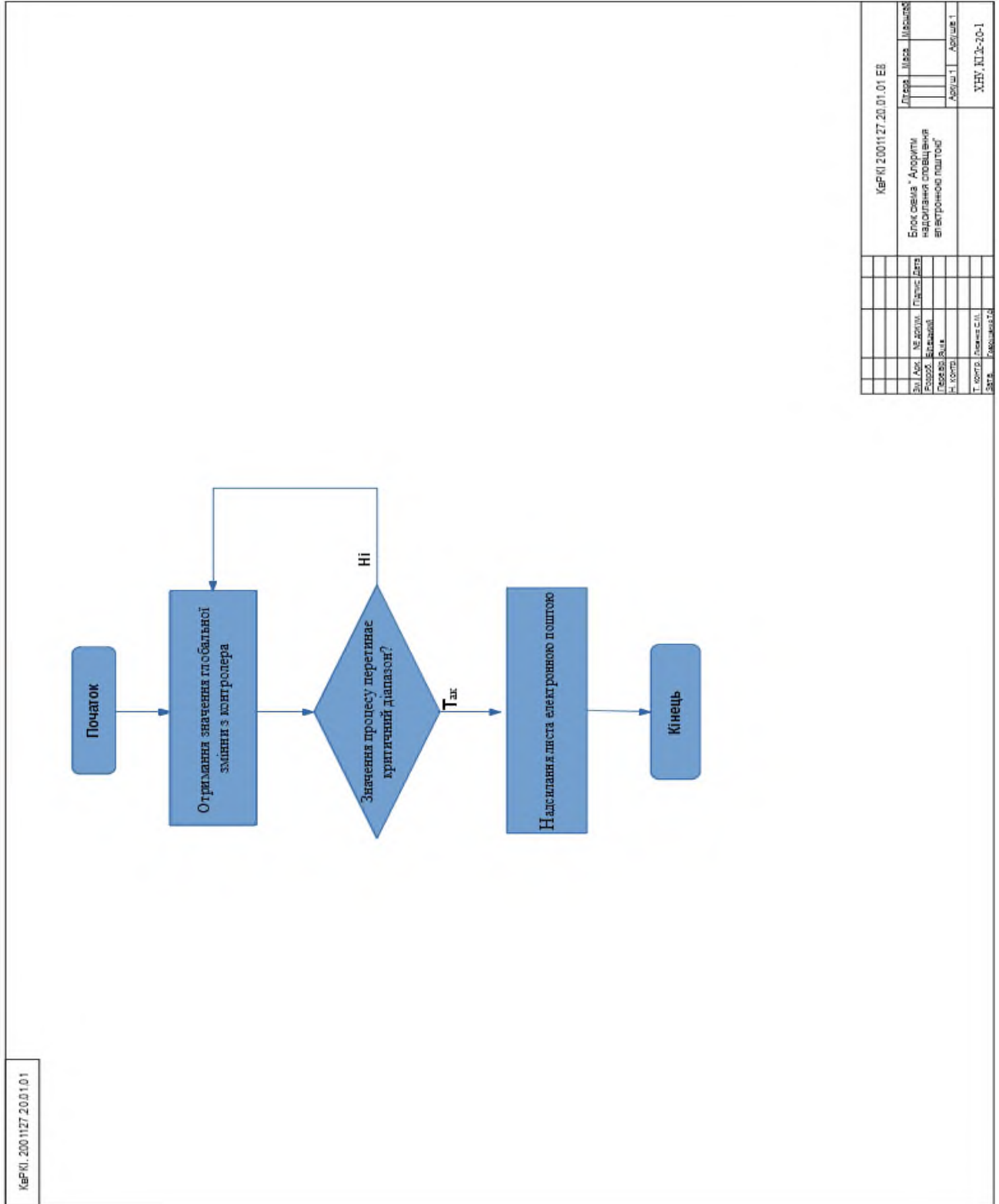
(обовязковий)

Копія креслення «Сценарій надсилання сповіщення електронною поштою»



Додаток В (обов'язковий)

Копія креслення «Блок схема “Алгоритм надсилання сповіщення електронною поштою”»



Ім'я користувача:
Кафедра КІ

ID перевірки:
1015447139

Дата перевірки:
06.06.2023 08:50:05 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
06.06.2023 08:56:20 EEST

ID користувача:
100005591

Назва документа: Білецький_Програмно-технічний модуль керування Інтернетом речей на основі Node-RED
Кількість сторінок: 65 Кількість слів: 8890 Кількість символів: 68205 Розмір файлу: 8.57 MB ID файлу: 1015107456

8.72% Схожість

Найбільша схожість: 3.28% з Інтернет-джерелом (<https://er.nau.edu.ua/bitstream/NAU/57125/1/%d0%a4%d0%9a%d0%...>)

8.37% Джерела з Інтернету	136	Сторінка 67
2.02% Джерела з Бібліотеки	131	Сторінка 68

0% Цитат

Цитати	1	Сторінка 69
Посилання	1	Сторінка 69

0% Вилучень

Немає вилучених джерел

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 8%**

ID: 114880 Назва: БКР Програмно-технічний модуль керування Інтернетом речей на основі Node-RED Додано в БД: 2023-06-06 Автора: М.Г. Білецький Керівники: В.В. Яцків Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	59879	496	1112 (2%)	15 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Білецький Максим Геннадійович

Тема: Програмно-технічний модуль керування Інтернетом речей на основі Node-RED

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 57

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розробка програмно-технічного модулю керування Інтернетом речей на основі Node-RED.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи проведено дослідження предметної області (проаналізовано теоретичні аспекти Інтернету речей, огляд існуючих програмних рішень для керування Інтернетом речей, аналіз можливостей платформи Node-RED) та виконано постановку задачі дослідження. В другому розділі кваліфікаційної роботи проведено моделювання та проєктування програмно-технічного модулю керування Інтернетом речей на основі Node-RED, а саме: виконано аналіз функціональних та нефункціональних вимог до програмно-технічного засобу, та проведено опис архітектури програмно технічного засобу на основі Node-RED. В третьому розділі кваліфікаційної роботи виконано апаратну та програмну реалізацію програмно-технічного модулю керування Інтернетом речей на основі Node-RED, а саме: використовуючи u-create web, веб-середовище розробки для контролерів u-control, був розроблений програмно-технічний засіб, програма якого надсилатиме сповіщення електронною поштою технікам з обслуговування, коли параметр процесу досягне критичного рівня.

4. Позитивні сторони роботи: Розроблено модуль керування Інтернетом речей з використанням середовища Node-RED

5. Негативні сторони роботи: В роботі не наведено порівняння запропонованого рішення із відомими аналогами

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на достатньому рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «задовільно» 3,2 (E)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Гурман
Іван Васильович доцент кафедри Інженерії програмного
забезпечення

“ 6 ” червня 2023 р.

ІВГ (підпис)

Завідувачу кафедри КІПС
д-р.техн.наук, проф. Говорущенко Т. О.

Білецький Максим Геннадійович
ПІБ здобувача вищої освіти

ФІТ, 3 курсу, групи КІ2с-20-1


ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

“05” червня 2023 р.

 (підпис)

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Програмно-технічний модуль керування Інтернетом речей на основі Node-RED

Автор: Білецький Максим Геннадійович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Яцків Василь Васильович, д.т.н, проф.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

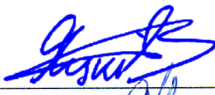
- 1) запозичення розміщені в розділі аналізу існуючих аналогів та відомих рішень, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;

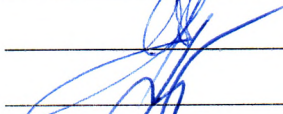
Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 8,72% і адресується до 40 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.


Керівник роботи

Гарант ОП

Завідувач кафедри КІС







В. В. Яцків

С. М. Лисенко

Т. О. Говорушенко