

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
Науково-Навчальний інститут атомної та теплової енергетики
Кафедра автоматизації енергетичних процесів**

Магістерська дисертація

на тему: Комплексний моніторинг стану навколишнього
середовища із дистанційною системою збору інформації

Керівник: ст.викл. Штіфзон О.Й.

Допущено до захисту
«__» _____ 20__ р.

Захищено з оцінкою

Сторінок _____

Виконав
студент 11 курсу
групи ТА-11мп

Гузь Д.Р

Залікова книжка №

Слайдів _____

Креслень _____

Ілюстрацій _____

Київ – 2022 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ГОРЯ СІКОРСЬКОГО”

Науково-Навчальний інститут атомної та теплової енергетики
Кафедра автоматизації енергетичних процесів

«На правах рукопису»
УДК 004.735

«До захисту допущено»
Завідувач кафедри

_____ / **Володимир ВОЛОЩУК** /
“ _____ ” _____ 2022 р.

Магістерська дисертація
на здобуття ступеня магістра

за освітньо-професійною програмою
**“Автоматизація та комп’ютерно-інтегровані технології
кібер-енергетичних систем”**
зі спеціальності
**151 “Автоматизація та комп’ютерно-інтегровані
технології”**

на тему: _____ Комплексний моніторинг стану навколишнього середовища із
дистанційною системою збору інформації

Виконав: студент _____ ІІ _____ курсу, групи _____ ТА-11мп
_____ Гузь Дмитро Романович
(прізвище ім’я, по батькові)

_____ (підпис)

Науковий керівник _____ ст.викл. Штіфзон Олег Йосипович
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Рецензент _____ директор ТОВ "ТД"Солар Ікс" Іваненко О.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській дисертації немає
запозичень з праць інших авторів без відповідних
посилань.

Студент _____

Київ – 2022 року

Національний технічний університет України
“Київський політехнічний інститут
імені Ігоря Сікорського”

Інститут Навчально-науковий інститут атомної та теплової енергетики

Кафедра Автоматизації енергетичних процесів

Рівень вищої освіти – другий(магістерський)

Спеціальність 151 “Автоматизація та комп’ютерно-інтегровані технології”

Освітньо-професійна програма (“Автоматизація та комп’ютерно -інтегровані технології кібер-енергетичних систем”)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____/Володимир ВОЛОЩУК/
(підпис) (імя, ПРІЗВИЩЕ)
“ “ _____ 2022 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Гузю Дмитру Романовичу
(прізвище, ім'я, по-батькові)

1. Тема дисертації Комплексний моніторинг стану навколишнього середовища із дистанційною системою збору інформації

науковий керівник дисертації Штіфзон Олег Йосипович, ст. викладач
(прізвище, ім'я, по-батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «07» 11 2022 р. № 4067-с

2. Термін подання студентом дисертації «10» грудня 2022 р.

3. Об’єкт дослідження система моніторингу стану довкілля із дистанційним збором даних

4. Вихідні дані підтримка до 5000 кінцевих пристроїв у системі, діапазон температури -40 ... 100 °С, відносна вологість повітря 0...100 %, архівація річних даних

5. Перелік завдань, які потрібно розробити систему моніторингу стану

довкільля із дистанційною передачею інформації
Програмну частину системи та кінцевого пристрою
Апаратну частину кінцевого пристрою
Демонстрацію моделювання роботи системи
Стартап-проект

6. Орієнтований перелік графічного (ілюстративного) матеріалу
Зображення архітектури системи, ілюстрації апаратних компонентів,
ілюстрації програмного коду, зображення макетного набору,
зображення вікон середовищ демонстрації.

7. Орієнтований перелік публікацій

8. Дата видачі завдання " 21 " вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Видача завдання	21.09.2022	
2	Аналітичний огляд системи	29.09.2022	
3	Проектування системи моніторингу	7.10.2022	
4	Опис системи моніторингу	14.10.2022	
5	Розробка програмної частини серверної архітектури	21.10.2022	
6	Розробка апаратної частини кінцевого пристрою	29.10.2022	
7	Розробка програмної частини кінцевого пристрою	11.11.2022	
8	Моделювання роботи системи	19.11.2022	
9	Стартап-проект	26.11.2022	
10	Підпис керівника магістерської дисертації	9.12.2022	
11	Попередній захист магістерської дисертації	16.12.2022	
12	Захист	20.12.2022	

Студент

Дмитро ГУЗЬ

(підпис)

(Ім'я, ПРІЗВИЩЕ)

Науковий керівник дисертації

Олег ШТІФЗОН

(підпис)

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

У сучасних умовах спостереження за станом навколишнього середовища відіграє значну роль, особливо на фоні кліматичних змін по всій планеті останніми роками. Беручи до уваги це, вдалим рішенням було б щонайменше вдосконалити вже наявні системи спостереження новішими технологіями із віддаленим моніторингом параметрів довкілля. В якості таких було взято сучасну бездротову технологію передачі даних радіоканалом високої дальності LoRaWAN, що володіє низьким енергоспоживанням в роботі. Граничний пристрій базується на сучасному мікроконтролері ESP32, що володіє порівняно низькою ціною та широким потенціалом для застосувань. Таке рішення може забезпечити автономний цикл роботи пристрою від акумулятора близько року.

Збір даних отриманих із віддалених точок міг би слугувати хорошим інструментом для відстежування та прогнозування параметрів навколишнього середовища, а також давати можливість раннього сповіщення про аномальні зміни.

У даній магістерській роботі запропоновано рішення комплексного моніторингу стану навколишнього середовища із дистанційною системою збору інформації. Така система є не лише розробкою з використанням справді сучасних технологій, а й дійсно безпечним рішенням із довготривалими результатами.

Отримана у результаті розробки система здатна забезпечити відповідність до вимог, що поставлені перед нею.

ABSTRACT

Nowadays environmental monitoring plays a significant role, especially while talking about climate changes across the planet in recent years. Taking this into account, a good solution would be to at least to improve existing surveillance systems with newer technologies in monitoring of remote environmental parameters. The modern wireless technology of data transmission using a long-range LoRaWAN radio channel, which has low energy consumption in operation was taken. The edge-device is based on a modern MCU ESP32, which has a relatively low price and a wide potential for applications. Such a solution can provide the autonomous operation cycle of the device from the battery for about a year.

The collection of data obtained from remote points could serve as a good tool for monitoring and forecasting environmental parameters, as well as providing an early warning of abnormal changes.

In this master's thesis, a solution for complex environmental monitoring with a remote information collection system is proposed. Such a system is not only a development using truly modern technologies, but also a safe solution with long-lasting results.

The resulting system is able to ensure compliance with the requirements set against it.

Зміст

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП	7
1. РОЗДІЛ 1. Аналітичний огляд проблеми	9
1.1. Сучасний стан галузі	9
1.2. Опис компонентів системи моніторингу	11
1.3. Загальні вимоги до системи	19
2. РОЗДІЛ 2 Проектування системи моніторингу	20
2.1. Опис системи моніторингу	20
2.2. Призначення і функції створюваної системи	21
2.3. Вимоги до реалізації функцій системи	21
3. РОЗДІЛ 3. Розробка системи віддаленого моніторингу	23
3.1. Розробка програмної частини	23
3.2. Розробка апаратної частини кінцевого пристрою на ESP32... ..	36
3.3. Розробка програмного забезпечення граничного пристрою ..	43
3.4. Моделювання роботи системи	50
4. РОЗДІЛ 4. Розробка стартап-проекту	63
4.1. Опис ідеї проекту	63
4.2. Технологічний аудит ідеї проекту	65
4.3. Аналіз ринкових можливостей запуску стартап-проекту	66
4.4. Розроблення ринкової стратегії проекту	73
4.5. Розроблення маркетингової програми стартап-проекту	74
4.6. Висновки	77
ВИСНОВКИ	78
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	79

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ПЗ – програмне забезпечення

АТ-команди – набір команд для модемної комунікації

Industry 4.0 – четверта промислова революція

LoRaWAN – Long Range Wide Area Network

IoT – Internet of Things

OTAA – over the air activation

SoC – system on chip

RISC-V – reduced instruction set computer (RISC) five

ВСТУП

У сучасних умовах спостереження за станом навколишнього середовища відіграє значну роль, особливо на фоні кліматичних змін по всій планеті останніми роками. Беручи до уваги це, вдалим рішенням було б щонайменше вдосконалити вже наявні системи спостереження новішими технологіями із віддаленим моніторингом параметрів довкілля. Збір даних отриманих із віддалених точок міг би слугувати хорошим інструментом для відстежування та прогнозування параметрів навколишнього середовища, а також давати можливість раннього сповіщення про аномальні зміни. Наприклад, встановивши пристрої обладнані сенсорами, що можуть збирати дані про забруднення повітря в безпосередній близькості до промислових об'єктів, можна ретельніше контролювати їх на предмет шкідливих викидів та оперативніше залучати відповідні установи для запобігання нанесенню шкоди людям та довкіллю.

Станом на 2022 рік питання забруднення Землі звучить все частіше й частіше. На жаль, екологія з плином часу та розвитком глобальної індустріалізації погіршується стрімкими темпами. Одним із основних параметрів якості навколишнього середовища вважають чистоту повітря, також у світі відоме значення “AQI (Air quality index)” – індекс якості повітря - параметр, що використовується для повідомлення громадськості рівню забруднення повітря у певний момент часу.

І хоча всесвітні організації працюють над поліпшенням контролю якості повітря, українська система моніторингу якості повітря не оновлювалась десятками років і, на жаль, не може надавати актуальну в часі інформацію про стан повітря, як це слід робити в умовах сучасності. Хоча цьому слід було приділити куди більше уваги, адже ми геть не на останньому місці в рейтингу забруднених міст, а з огляду на осередки металургії та поширеність ТЕЦ та ТЕС в Україні – не дивина. До 2020 Міністерство захисту довкілля та природних

ресурсів планувало закупівлю сучасного обладнання моніторингу повітря, але тоді ці кошти були передані в фонд коронавірусу.

При помітно низькій якості повітря в певному місці, коли індекс якості повітря показує, що сильний вплив може призвести до істотної шкоди здоров'ю населення, уповноважений орган може ввести надзвичайний план, який дозволить скоротити викиди основним забруднювачам повітря такі як ТЕС, ТЕЦ, підприємства, що спалюють вугілля, газ до тих пір, поки шкідливі умови не послабляться.

Частина забруднювачів повітря не пов'язані з індексом якості. Більшість країн моніторять приземний озон, тверді частинки, діоксид сірки, окис вуглецю та діоксид азоту і проводять калькуляцію індексу якості повітря на підставі цих забруднювачів.

Отже, об'єктом розробки даної роботи є система моніторингу стану довкілля із дистанційним збором даних.

Предметом дослідження є розробка системи збору та віддаленої передачі параметрів.

Ціль дослідження – це створення системи для відстежування та передачі параметрів довкілля.

Тому, на мою думку, зараз є актуальним розробка та впровадження комплексних систем моніторингу параметрів навколишнього середовища з системою дистанційного збору інформації та її обробки. Отже, метою даної роботи є розробка системи для моніторингу стану навколишнього середовища із дистанційною системою збору інформації.

За матеріалами дисертації були підготовлені тези доповіді, які прийняті до друку на XX-й міжнародній науково-практичній конференції молодих вчених та студентів "Сучасні проблеми наукового забезпечення енергетики".

1. РОЗДІЛ 1. Аналітичний огляд проблеми

1.1. Сучасний стан галузі

Українським законодавством передбачено наявність різноманітних нормативів для стаціонарних і пересувних джерел забруднення. Деякою мірою застарілих – радянських. А от індекс якості повітря AQI в Україні відсутній. Коли немає нормативу якості повітря, то немає необхідності в його контролі, станції контролю атмосфери, які працюють в режимі on-line відсутні, і газоаналізатори для станцій контролю не випускаються через відсутність потреби.

В більшості європейських країн, США, Канаді, Південно-Західній Азії створені національні мережі станцій контролю атмосферного повітря режимі on-line. В Швеції діє близько 160 стацій, які першими зафіксували викиди радіоактивних елементів в Чорнобилі. Такі станції контролюють основні забруднювачі атмосфери, які виникають у різноманітних процесах горіння в енергетиці, транспорті та промисловості: оксид вуглецю, діоксид сірки, оксиди азоту, озон (чи не найтоксичніший компонент у складі повітря), частинки пилу розміром 2,5 мкм і 10 мкм, також метеопараметри[1].

Вже довгий час у світі дослідження якості повітря відбувається з допомогою комплексної системи GEMS.

Глобальна система моніторингу навколишнього середовища (GEMS) — це колективна спроба світового співтовариства отримати за допомогою моніторингу даних, необхідних для раціонального управління навколишнім середовищем, що виникла на основі рекомендацій Конференції ООН з навколишнього середовища людини, яка проходила в Стокгольмі у 1972 році.

Система моніторингу GEMS складається з п'яти тісно взаємопов'язаних програм, які містять умови для навчання та надання технічної допомоги для забезпечення участі країн, які недостатньо забезпечені персоналом та обладнанням. До її складу входять:

- Кліматичний моніторинг;
- Моніторинг переносу забруднюючих речовин на великі відстані;
- Моніторинг чинників забруднення, які впливають на здоров'я людини;
- Моніторинг океану;
- Моніторинг наземних відновлюваних ресурсів.

Кожна з цих широких областей містить принаймні п'ять різних всесвітніх мереж моніторингу. Прикладами останніх є Всесвітній кадастр льодовиків, Мережа моніторингу фонового забруднення повітря, мережа моніторингу забруднення повітря в містах, глобальна мережа моніторингу якості води, мережа моніторингу тропічних лісів, мережа моніторингу збереження видів тощо.

Моніторингові дані збираються у відповідних координаційних центрах для кожної мережі, де створені або створюються відповідні бази даних. Дані аналізуються для проведення періодичних регіональних і глобальних оцінок, які повідомляються через проміжки часу, що відповідають змінній, яка розглядається.

1.2. Опис компонентів системи моніторингу

Система моніторингу складається:

- граничний пристрій – збір інформації з сенсорів по місцю встановлення;
- шлюз обробки даних (gateway) – маршрутизація пакетів від серверу до пристроїв та у зворотному напрямку;
- серверна частина: мережевий (network) сервер, що опрацьовує відправку даних від базової станції до серверу додатків (application server), сам сервер додатків, що може відображати функціонал для взаємодії з пристроями в мережі, дані, отримані від мережевого серверу, та відображення даних у графічній формі для візуалізації;

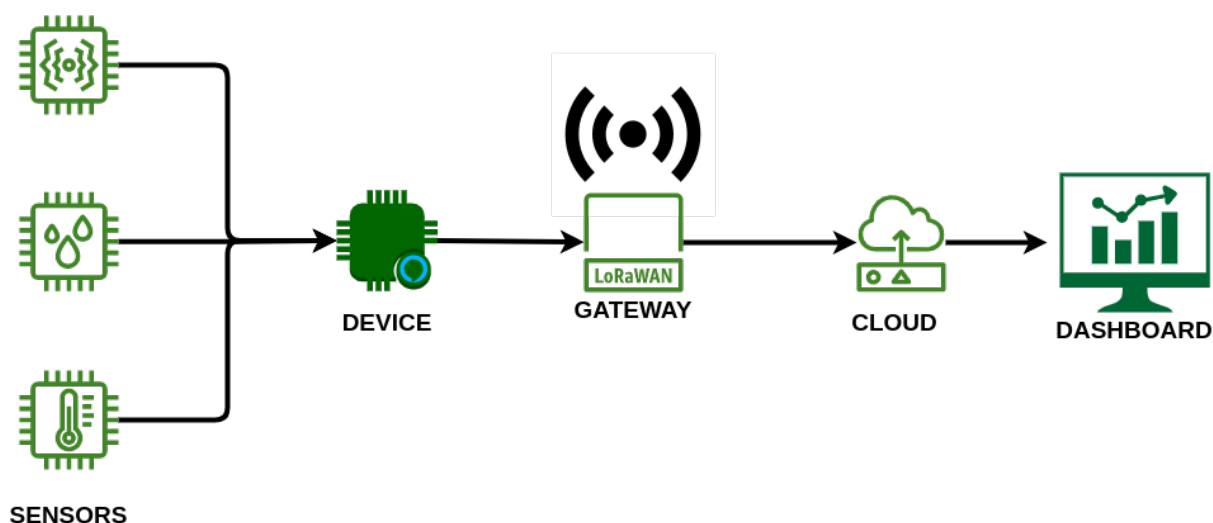


Рисунок 1.1. Технологічна схема системи моніторингу

Запропоновану систему наведено на рис.1. На нижньому рівні працює мікроконтролерний пристрій із встановленими на ньому давачами, що збиратиме інформацію для передачі на вищий рівень.

Традиційна топологія для LoRa мережі – зірка, коли багато кінцевих пристроїв спілкуються із одним шлюзом, а він у свою чергу забезпечує комунікацію із мережевим сервером, проілюстровано на Рисунку 1.2:

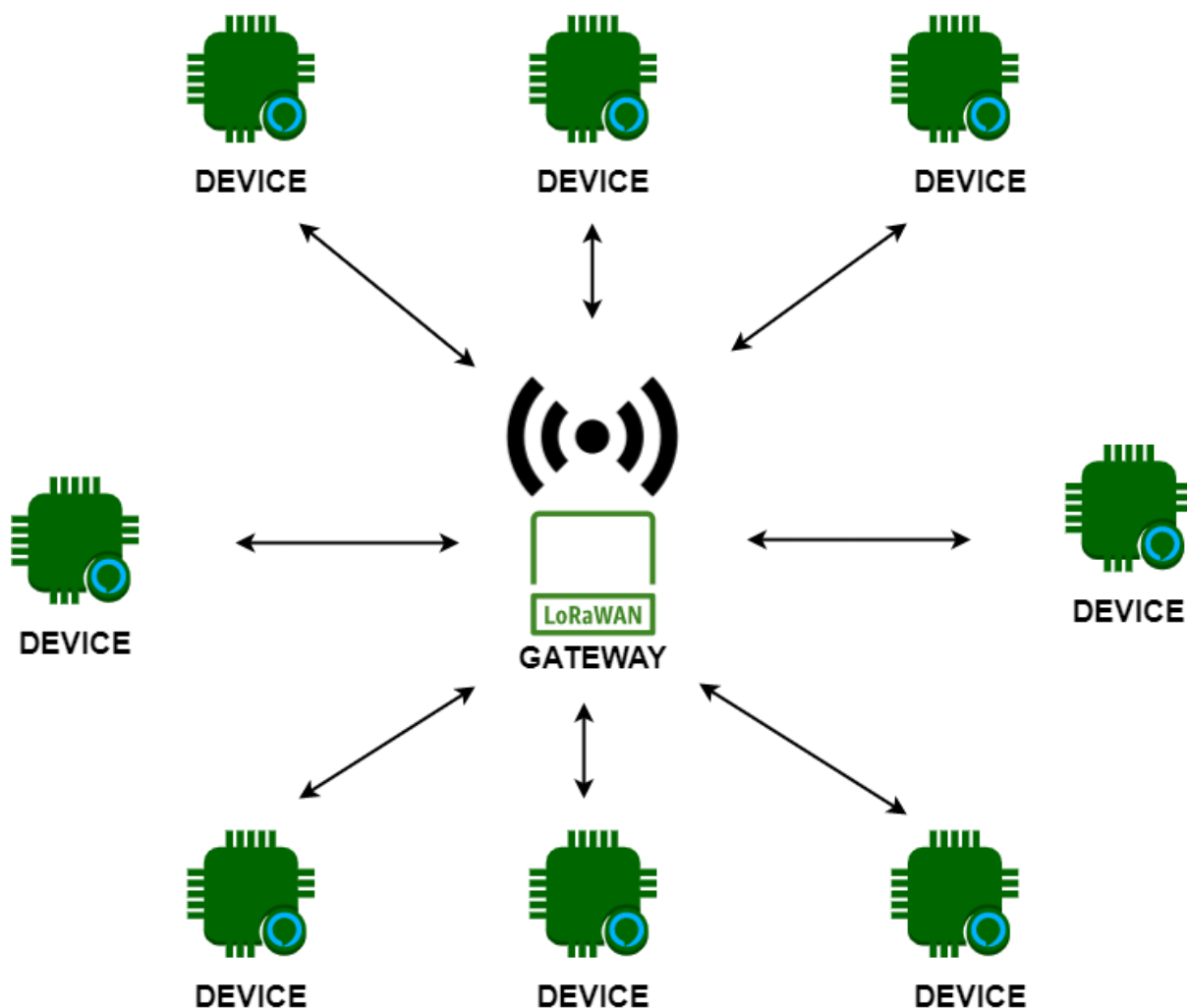


Рисунок 1.2. Топологія LoRaWAN для мережі з одним шлюзом

Частим випадком для промислових застосувань є використання кількох шлюзів обробки для покриття сигналом більшої площі. В теорії також можливий сценарій, коли навіть на малій місцевості пропускної здатності 1 шлюза буде недостатньо (як правило це до 10 000 пристроїв одночасно в мережі), тоді слід збільшити шлюзів у системі. Таке рішення забезпечить ширші можливості зв'язку, проте відповідно й здорожчає систему за рахунок додаткових гейтвеїв. Для випадків коли в мережі працює не один шлюз, застосовується топологія “зірка із зірок”, коли кілька шлюзів спілкуються із мережевим сервером, що проілюстровано на Рисунку 1.3:

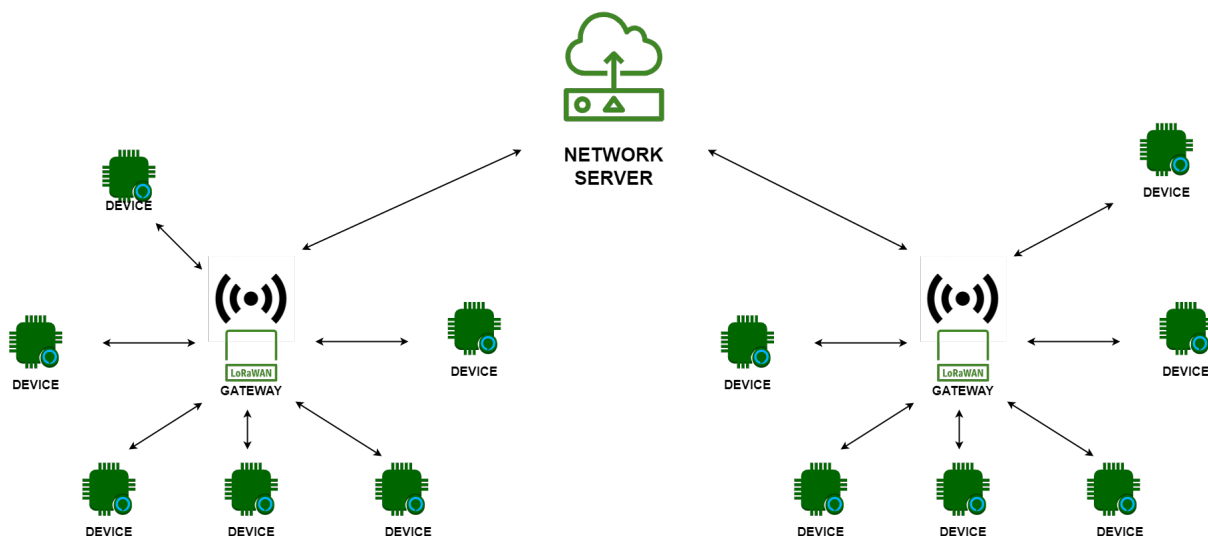


Рисунок 1.3. Топологія розширеної мережі “зірка із зірок”

Сьогодні існують варіації технологій для застосування в системах схожого характеру, наприклад LoRaWAN, Narrowband-IoT, Sigfox, Zigbee та інші. Усі вони в міру схожі між собою, деякі працюють на однаковій частоті, проте реалізація протоколів різниться. Основні характеристики цих технологій порівняно на діаграмі [2] на Рисунку 1.4:

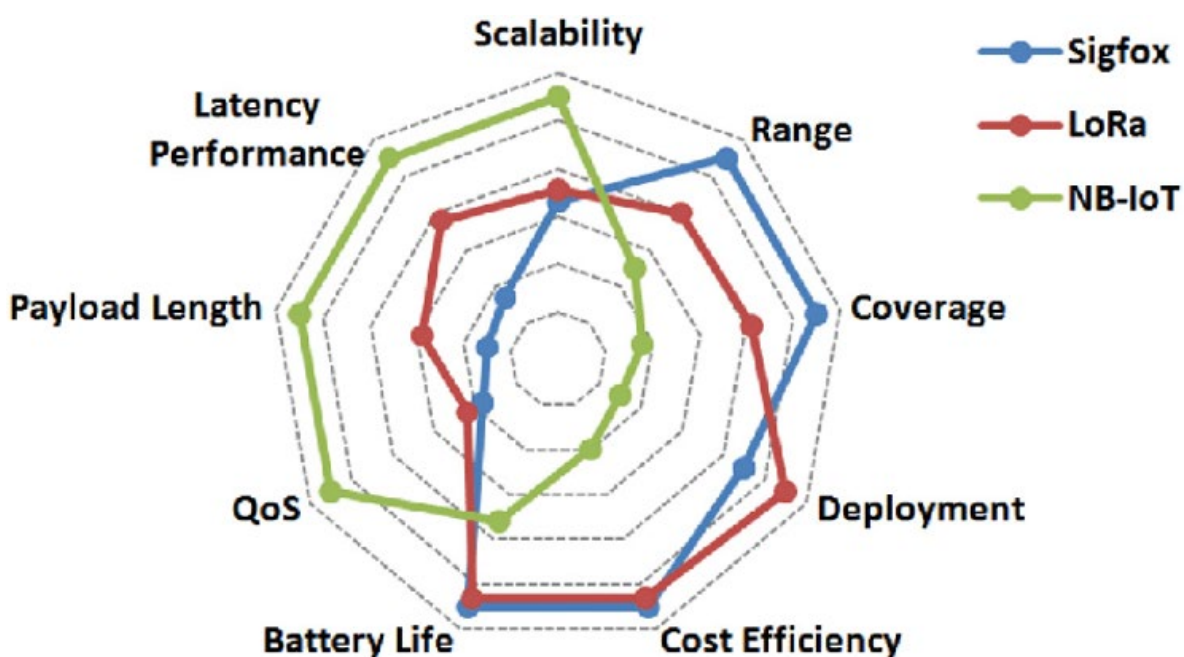


Рисунок 1.4. Порівняння LoRa, Sigfox та Narrowband-IoT

NB-IoT було створено спеціально з урахуванням малопотужних стаціонарних датчиків. Протокол пропонує широкі зони покриття з глибоким проникненням у приміщеннях.

На відміну від LoRaWAN, це ліцензований протокол, який, ймовірно, коштуватиме дорожче в довгостроковій перспективі, хоч і можливо забезпечить кращий загальний досвід для кінцевих користувачів.

На європейському ринку популярності набирає SigFox для промислових рішень, проте компанія позиціонує себе як оператор зв'язку, тому через закритість використання не слід дивитися в сторону цього протоколу в першу чергу.

Zigbee базується на стандарті персональної мережі IEEE 802.15.4. Тому певною мірою вважається альтернативою Wi-Fi і Bluetooth для деяких застосувань, у тому числі для пристроїв з низьким споживанням енергії, які не вимагають великої енерговитратності - наприклад, датчики розумного будинку. Проте для покриття міської площі великою мережею чи використання у польових умовах він навряд підійде.

Тут варто згадати перелік до двох найбільш зручних стандартів та виділити кілька пунктів:

- LoRaWAN споживає менше електроенергії порівняно з NB-IoT, що робить його більш життєздатним рішенням для проектів, які потребують вищої частоти оновлення.
- Оскільки LoRaWAN споживає менше енергії, він також забезпечує довший час роботи акумулятора порівняно з NB-IoT (15+ років порівняно з 10+ роками).
- Коли справа доходить до пропускної здатності та покриття, обидва можуть збігатися.
- Пікова пропускна здатність для NB-IoT становить 60 Кбіт/с, що трохи вище, ніж у LoRaWAN.

- NB-IoT є більш безпечним рішенням завдяки 256-бітному шифруванню 3GPP (у порівнянні зі 128-бітним AES для LoRaWAN).
- Затримка на NB-IoT іноді нижча, ніж на LoRaWAN. Хоча й затримка на LoRaWAN залежить від типу використовуваного пристрою та його класифікації.
- Обидва стандарти однаково підтримують геолокацію.

Та у випадку застосування LoRaWAN, дещо ширшим буде вибір готової продукції та компонентів, що використовуються у цьому стандарті, адже для цього не потрібне ліцензування, що й у свою чергу робить ціну доступнішою у порівнянні із NB-IoT[3].

Тому з урахуванням сказаного вище, при розробці такої системи вдалим буде застосувати недорогий та популярний RISC-V мікроконтролер ESP32 із LoRa-модемом на борту в якості граничного пристрою. Таке рішення забезпечить:

- 128-бітне шифрування при передачі даних
- Низькі енерговитрати (режими глибокого сну на боці МК)
- Порівняно низьку вартість готового пристрою
- Велику відстань передачі

Існують вже розроблені продукти схожого спрямування. Пропозиція на ринку попередніх років мала перелік пристроїв на процесорах сімейства STM32 зі схожими модулями LPWAN-технології (NB-IoT та ZigBee), які через кризу напівпровідникового виробництва є важкодоступними.

Пристрій надсилає дані до шлюзу передачі (gateway) на частоті 868 МГц, що має доступ до мережі Інтернет. Як правило, задля кращої швидкодії шлюз підключений через Ethernet інтерфейс, якщо це дозволяє місце його використання. Проте якщо розташування його не передбачає такої можливості, то більшість сучасних шлюзів мають можливість підключення до Інтернету

через стільникову мережу 3G/LTE[4]. Сьогодні в Україні найбільші оператори мобільного зв'язку також пропонують вигідні тарифи для Smart-девайсів, що робить такий спосіб підключення зручним.

Досить вдалим рішенням для покриття великої міської площі або ж відстані до 15км відкритої місцевості було б використати шлюз Tektelic Kona Macro IoT (Рисунок 1.5). До його переваг варто зарахувати можливість монтажу на дворі та широкий температурний діапазон безвідмовної роботи ($-40 \dots +60^{\circ}\text{C}$), що для нашого завдання цілком підійде.



Рисунок 1.5. Tektelic KONA Macro IoT Gateway

Шлюз пропускає інформацію лише від зареєстрованих пристроїв і передає її до сховища чи на сервер, в нашому випадку бази даних (хмари), звідки дані опрацьовуватимуться та відображатимуться у зручній формі для користувача[5].

Аби розуміти основні моменти роботи сучасної технології LoRaWAN, спочатку варто розібратися в її основах. Власне, аббревіатура “LoRa” об'єднує в собі метод модуляції LoRa у бездротових мережах LPWAN та відкритий протокол LoRaWAN. Цей протокол використовується щоб взаємодіяти між LPWAN-шлюзами, а також кінцевими пристроями.

Особливістю цієї технології є висока на сьогодні надійність передачі даних та пристойна відстань передачі інформації (10-15 км). Одна базова станція здатна обслуговувати більше тисячі пристроїв, при цьому і з двостороннім з'єднанням. Протокол LoRa забезпечує передачу в бездротовому каналі на швидкості від 0,3 до 50 Кбіт/с.

Коли кінцевий пристрій починає передачу даних, його слухають усі доступні шлюзи (базові станції). Далі вони передають інформацію на мережевий сервер (**network server**), який дивлячись на параметри про якість сигналу, віддаленість до пристрою та інші метадані вирішує, яка базова станція відповість пристрою. Після цього дані відправляються на сервери додатків (**application server**).

LoRaWAN мережа як правило має топологію “зірка із зірок”, має кінцеві вузли, які через шлюзи спілкуються з центральним сервером мережі. За такого підходу зазвичай передбачається, що шлюзами і центральним сервером володіє оператор цієї мережі, а кінцевими вузлами — абоненти. Абоненти мають прозору двобічну та захищену 128-бітним AES шифруванням передачу даних до кінцевих вузлів.

Окремо варто винести ще кілька аспектів безпеки. Використання такої мережі забезпечує повну безпеку та захист переданих користувачами даних, що забезпечуються шифруванням AES на декількох рівнях:

- на мережевому рівні з використанням унікального мережевого ключа (Unique Network key, EUIN64);

- наскрізну безпеку на рівні додатків за допомогою унікального Application key, EUI64;
- спеціального ключа пристрою (Device specific key, EUI128).

Класи пристроїв

Розрізняють три основні класи граничних пристроїв, що обслуговують різні додатки. Кожен із них має особливості, що завдяки їм пристрої представляють різний функціонал для окремих застосувань.

- Двонаправлені кінцеві пристрої **«класу А»** (Bi-directional end-devices, Class A). Пристрої цього класу застосовуються, коли важлива мінімальне споживання енергії із перевагою передачі даних до сервера. Ініціатором відкриття сеансу зв'язку виступає кінцевий вузол, надсилаючи пакет із необхідними даними, а потім відкриває два вікна, у перебігу яких чекає на дані з боку сервера. В такий спосіб передача даних від сервера можлива тільки після виходу в ефір кінцевого пристрою.
- Двонаправлені кінцеві пристрої **«класу В»** (Bi-directional end-devices, Class B). Основна різниця пристроїв «класу А» із першим полягає у виділенні додаткового вікна прийому, яке пристрій відкриває за власним розкладом. Для складання розкладу кінцевий пристрій здійснює синхронізацію по спеціальному сигналу від шлюзу. Завдяки такому вікну сервер може почати передачу даних у заздалегідь зазначений час.
- Двонаправлені кінцеві пристрої **«класу С»** з максимальним вікном для прийому даних (Bi-directional end-devices, Class C). Пристрої даного класу мають майже безперервне вікно прийому даних і закривають його виключно при потребі на визначений час передачі даних. Це дозволяє їх застосовувати для вирішення завдань, що вимагають одержання великого обсягу даних.

1.3. Загальні вимоги до системи

Задля ефективного отримання та обробки даних в умовах темпу розвитку **Industry 4.0**, в даній роботі необхідно спроектувати сучасну систему комплексного моніторинг стану навколишнього середовища із дистанційною системою збору інформації із використанням популярних та доступних технологій.

При застосуванні такого комплексу рішень варто приділити увагу можливості розширення функціоналу за такої потреби, впровадження алгоритму прогнозування отримуваних даних та забезпечити зручну взаємодію користувача із впровадженим функціоналом та візуалізацією.

При виборі технологій для розробки обов'язковим є врахування тенденцій у сфері технічного устаткування та програмного забезпечення.

Впроваджуючи таку систему вона повинна надавати безпечну передачу та зберігання даних, швидкодію при роботі, надійне обладнання та оптимальну автономність роботи у випадку застосування акумуляторних граничних пристроїв.

2. РОЗДІЛ 2 Проектування системи моніторингу

2.1. Опис системи моніторингу

У відповідності до технологічної схеми наведеної на Рисунку 1.1, розглянемо більш детально структуру граничного пристрою. На Рисунку 2.1 проілюстровані складові граничного пристрою для системи.

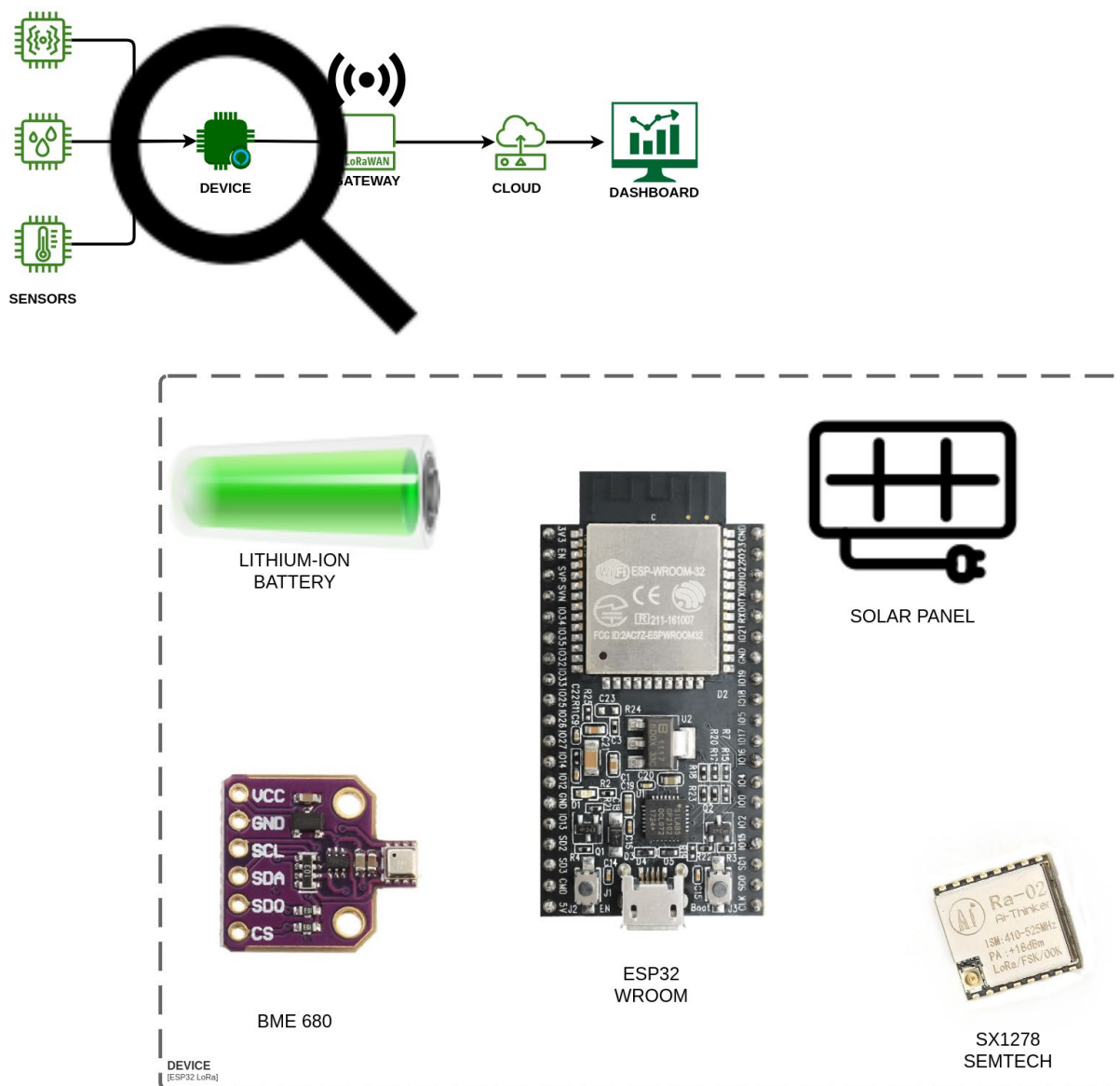


Рисунок 2.1. Структурна схема граничного пристрою

Для контролера ESP32 в якості джерела живлення застосований літій-іонний акумулятор та сонячна панель, що в більшу частину року даватиме хороші результати автономності, з огляду на те, що в основному контролер

працюватиме в режимі глибокого сну і “прокидатиметься” для опитування давачів та відправки пакету даних[6]. Модуль передавача 868 МГц із підтримкою протоколу LoRa обрано **Semtech SX1278** як один із надійних, широкоживаних і доступних. Що ж до давачів температури, вологості та якості повітря, то зазвичай для бюджетних “домашніх” рішень застосовують датчик температури та відносної вологості повітря DHT11/22, AM2032 датчик якості повітря MQ-135 або ж їх аналоги. Та за можливості краще було б обрати багатофункціональний і надійний датчик **Bosch BME680**, що на борту має можливість вимірювання температури, відносної вологості, барометричного тиску, широкого спектру газів та летючих органічних сполук, що великою мірою впливають на AQI.

2.2. Призначення і функції створюваної системи

Система, що проектується, повинна мати можливість:

- віддаленого моніторингу параметрів середовища, в якому пристрої встановлені;
- безпечної передачі та збереження даних;
- масштабування системи при потребі;
- інформативну візуалізацію вимірюваних параметрів;
- прогнозування відстежуваних параметрів;

2.3. Вимоги до реалізації функцій системи

Пропонована система перш за все повинна реалізовувати інформаційні функції: функцію передачі значень параметрів, сповіщення про стан обладнання та можливого виходу його з ладу (неможливість опитування давачів). Передача даних та їх збереження повинні мати достатній захист. Разом із цим необхідно приділити увагу надійному збереженню отримуваних даних аби можна було спиратися на історичні дані в перспективі.

Система, що відповідає таким вимогам, може замінити застарілі технології, що використовуються зараз для виконання моніторингу довкілля, спростивши роботу людям та заощаджуючи бюджет в довготерміновій перспективі. Також важливий акцент ставиться на можливості зручного розширення комплексу із додаванням пристроїв у вже налаштовану безпечну мережу.

3. РОЗДІЛ 3. Розробка системи віддаленого моніторингу

У цьому розділі основний фокус на розробці запропонованого комплексу на основних етапах. В ньому описана прийнята структура рішення, ілюстративна візуалізація і логіка використовуваних компонентів системи, взаємодії між ними.

3.1. Розробка програмної частини

У якості платформи для розгортання серверного back-end для LoRa-мережі, вирішено обрати **ChirpStack**.

ChirpStack — це мережевий сервер LoRaWAN з відкритим кодом, який можна використовувати для налаштування мереж LoRaWAN. ChirpStack надає веб-інтерфейс для керування шлюзами, пристроями, а також для налаштування інтеграції даних із основними хмарними (cloud) провайдерами, базами даних і службами, які зазвичай використовуються для обробки даних пристроїв. ChirpStack надає API (Application Programming Interface – прикладний програмний інтерфейс для взаємодії із функціоналом) на основі **gRPC**, який можна використовувати для інтеграції чи розширення ChirpStack.

gRPC — це сучасна високопродуктивна платформа Remote Procedure Call (RPC) з відкритим кодом, яка може працювати в будь-якому середовищі. Вона може ефективно підключати служби в центрах обробки даних і з їх допомогою підключається підтримка для балансування навантаження, трасування, перевірки працездатності та автентифікації. Він також застосовний в останньому етапі розподілених обчислень для підключення пристроїв, мобільних програм і браузерів до серверних служб.

Структура сервера мережі LoRaWAN представлена на Рисунок 3.1.

Основними компонентами сервера є:

Gateway Bridge — певний міст між програмою Packet Forwarder, встановленою на базовій станції (шлюзі) і безпосередньо структурою сервера LoRaWAN;

Мережевий сервер — мережевий сервер, що обробляє повідомлення мережевого рівня;

Application Server — сервер додатків, що забезпечує роботу мережі на рівні користувача, здійснює інтеграцію із зовнішніми платформами.

Допоміжні складники:

MQTT Broker Mosquitto — для внутрішнього обміну повідомленнями між компонентами сервера;

Redis - проміжна база даних для зберігання швидкоплинних даних;

PostgreSQL — база даних для постійного зберігання даних. Усе використане ПЗ має ліцензію з відкритим вихідним кодом.

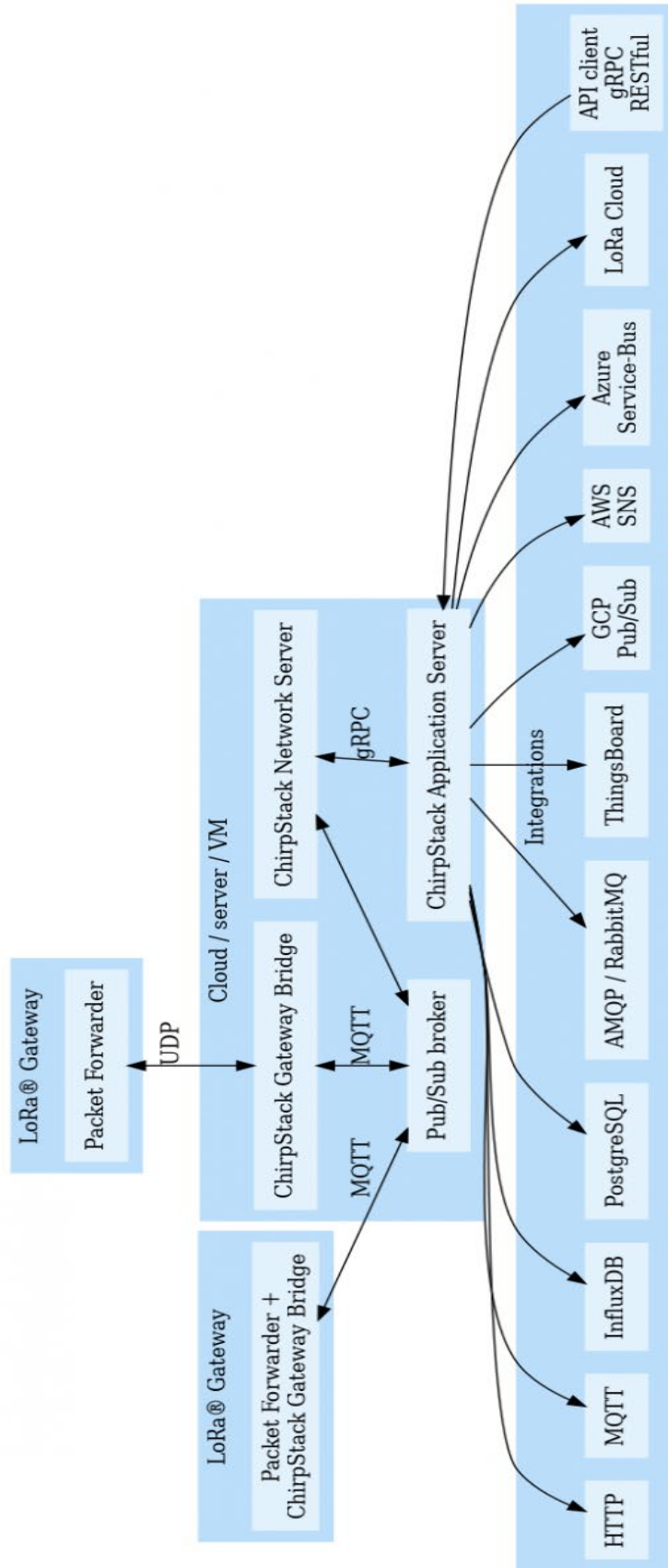


Рисунок 3.1. Структура сетевого сервера LoRaWAN ChirpStack

Встановлення середовища для ChirpStack

Для налаштування роботи серверу скористаймося рекомендаціями описаними в документації до програмного продукту.

Рекомендовано розгорнути репозиторій на базі машини із Ubuntu 22.04 LTS чи Debian 11.

Наступна команда терміналу встановлює рекомендовані компоненти ChirpStack за допомогою менеджера пакетів apt:

```
sudo apt install \
    mosquitto \
    mosquitto-clients \
    redis-server \
    redis-tools \
    postgresql
```

PostgreSQL встановлення

Щоб увійти в утиліту командного рядка для PostgreSQL:

```
sudo -u postgres psql
```

Тут слід виконати наведені нижче запити, щоб налаштувати базу даних ChirpStack. Рекомендується змінити імена користувачів і паролі.

```
-- create role for authentication
create role chirpstack with login password 'chirpstack';
-- create database
create database chirpstack with owner chirpstack;
-- change to chirpstack database
\c chirpstack
-- create pg_trgm extension
create extension pg_trgm;
-- exit psql
\q
```

Налаштування репозиторію

ChirpStack надає репозиторій для Debian/Ubuntu, який можна використовувати для встановлення останньої версії ChirpStack. Спочатку слід переконатися, що встановлено `dirmngr` і `apt-transport-https`:

```
sudo apt install apt-transport-https dirmngr
```

Далі налаштувати ключ для репозиторію ChirpStack:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
1CE2AFD36DBCCA00
```

Додати репозиторій до списку репозиторіїв:

```
sudo echo "deb https://artifacts.chirpstack.io/packages/4.x/deb
stable main" | sudo tee /etc/apt/sources.list.d/chirpstack.list
```

Оновити кеш пакетів apt:

```
sudo apt update
```

Встановлення ChirpStack Gateway Bridge

Для випадків, коли такий міст не передбачений на боці самого шлюзу, необхідно провести його первинне встановлення наступною термінальною командою із використанням apt:

```
sudo apt install chirpstack-gateway-bridge
```

Конфігурація

Файл конфігурації знаходиться за адресою `/etc/chirpstack-gateway-bridge/chirpstack-gateway-bridge.toml`. Слід оновити розділ `[integration.mqtt]`, щоб він відповідав префіксу регіону, який застосовується до цього об'єкту ChirpStack Gateway Bridge.

Приклад для EU868:

```
[integration.mqtt]
event_topic_template="eu868/gateway/{{.GatewayID}}/event/{{
.EventType }}"
command_topic_template="eu868/gateway/{{.GatewayID}}/command/#"
```

Запуск

Запуск служби ChirpStack Gateway Bridge:

```
# start chirpstack-gateway-bridge
sudo systemctl start chirpstack-gateway-bridge

# start chirpstack-gateway-bridge on boot
sudo systemctl enable chirpstack-gateway-bridge
```

Встановлення ChirpStack

Встановлення цього пакунку ПЗ також передбачає використання apt менеджера, як і в попередніх кроках:

```
apt install chirpstack
```

Конфігурація

Конфігураційні файли знаходяться в /etc/chirpstack. Можна знайти один файл глобальної конфігурації під назвою chirpstack.toml і різні файли конфігурації регіону.

Запуск сервісу ChirpStack

```
# start chirpstack
sudo systemctl start chirpstack

# start chirpstack on boot
sudo systemctl enable chirpstack
```

Вивід логів (журналу подій) ChirpStack:

```
sudo journalctl -f -n 100 -u chirpstack
```

Приєднання шлюзу до мережі

Компонент ChirpStack Gateway Bridge діє як серверна частина для засобів пересилання пакетів. Він використовує протокол **MQTT** для зв'язку з ChirpStack. Для його налаштування потрібно провести ініціалізацію для двох компонентів.

Пересилач пакетів (**packet-forwarder**), налаштований на шлюзі, повинен пересилати свої дані до ChirpStack Gateway Bridge. Оскільки він керує чіпсетом LoRa-шлюзу, він також має бути налаштований на правильні частоти. Невідповідність частот означає, що шлюз не отримуватиме вихідні канали зв'язку, надіслані пристроєм, і/або не зможе надсилати корисні дані вхідної лінії зв'язку, якщо частота вхідної лінії зв'язку виходить за межі налаштованого діапазону частот. Зазвичай постачальники шлюзів надають приклади конфігурації для різних діапазонів.

Налаштування до якого MQTT брокера підключатиметься ChirpStack Gateway Bridge встановлюються наступним чином:

```
# Generic MQTT authentication.
[integration.mqtt.auth.generic]

# MQTT servers.

# Configure one or multiple MQTT server to connect to. Each item must
be in

# the following format: scheme://host:port where scheme is tcp, ssl or
ws.

servers=[ "tcp://127.0.0.1:1883", ]
```

Username: admin

Password: admin

Додавання шлюзу

Потрібно перейти до «Gateways» у веб-інтерфейсі, натиснути «Add» та заповнити форму. Якщо все виконано успішно, тоді можна побачити шлюз у мережі.

MQTT

Оскільки ChirpStack Gateway Bridge використовує MQTT для зв'язку з ChirpStack, також можна підписатися на топіки (topics) MQTT шлюзу. Це підтвердить, що дані з шлюзу надходять принаймні до брокера MQTT.

Приклад команди для підписки на топик MQTT шлюзу за допомогою mosquitto_sub:

```
mosquitto_sub -v -t "+/gateway/#"
```

Приєднання кінцевого пристрою в мережу

Є кілька речей, які потрібно знати про пристрій, що потрібно приєднати в існуючу мережу.

- DevEUI
- LoRaWAN MAC версія імплементована на пристрої
- Регіональні параметри ревізії імплементованої на пристрої
- DevAddr
- Session-keys

Цю інформацію зазвичай надає постачальник пристрою. В нашому випадку серійний номер можна знайти на самому LoRa-модемі, а також від виробника постачається документ із мережевими ключами по визначеному серійнику, часто відображений на модулі під “FCC ID”, приклад на Рисунку 3.2:

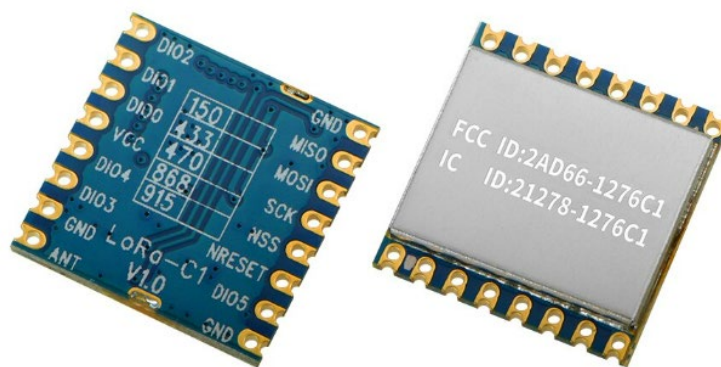


Рисунок 3.2. LoRaWAN модем із серійним номером на корпусі

Перш ніж додати пристрій до ChirpStack, потрібно створити профіль пристрою, якщо ще цього не зробили. Загалом, добре створювати окремі профілі пристроїв для різних типів пристроїв. Профіль пристрою містить можливості вашого пристрою. Наприклад, якщо він використовує ABP(Activation by personalization) або OTAA (Over the air activation) для активації, яку версію LoRaWAN і ревізію регіональних параметрів реалізовано пристроєм тощо... Його також можна налаштувати із функцією для декодування корисних даних, надісланих пристроями за допомогою профілю пристрою.

Пристрої згруповані за додатками (application). Наприклад, ви можете згрупувати датчики температури в одній програмі, а метеостанції – в іншій.

На вкладці «Пристрої» натисніть «Створити». Заповніть необхідні поля та виберіть профіль пристрою, який ви хочете пов'язати зі своїм пристроєм, і збережіть пристрій.

Залежно від конфігурації профілю пристрою для OTAA або ABP, наступна сторінка попросить вас ввести кореневі ключі пристрою (OTAA) або сеансові ключі пристрою (ABP). Якщо екземпляр ChirpStack налаштовано з join-server і пристрій використовуватиме цей сервер приєднання для активації, тоді немає необхідності вводити кореневі ключі.

Інтеграція із Node-RED

Node-RED надає редактор потоків на основі браузера, який дозволяє легко об'єднувати потоки за допомогою широкого діапазону вузлів у палітрі [7].

За замовчуванням Node-RED вимкнено, і його потрібно ввімкнути за допомогою утиліти *gateway-config*. Після запуску Node-RED доступний у браузері через порт `http://[IP-АДРЕСА]:1880`. Це означає, що якщо IP-адреса шлюзу 192.168.0.1, можна отримати доступ до Node-RED за адресою <http://192.168.0.1:1880>.

Побудова ехо-потoku

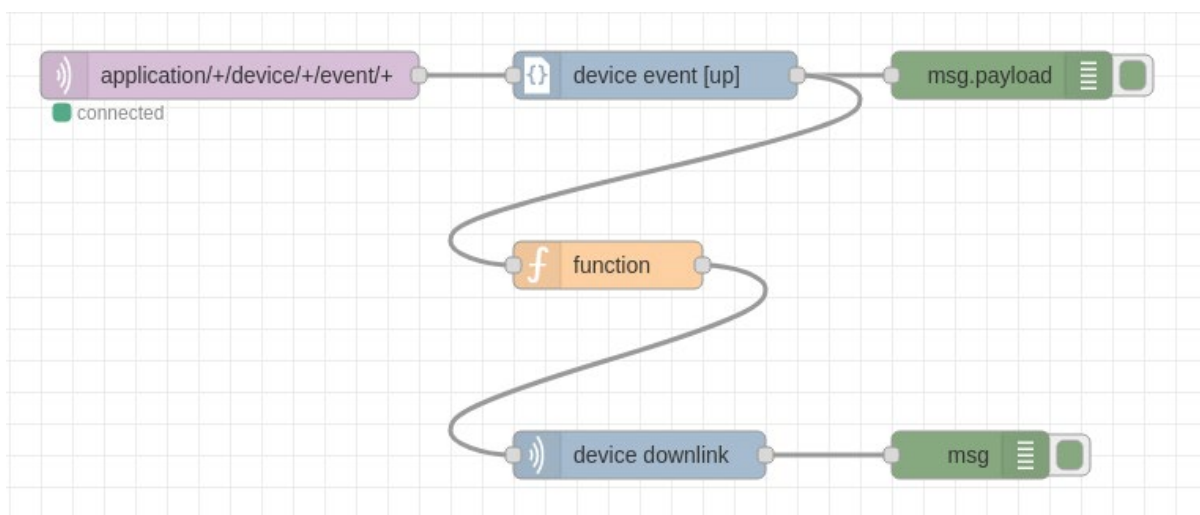


Рисунок 3.3. Потік Node-RED для базового налаштування каналу пакетів

У наведених нижче кроках описано, як побудувати ехо-потік, який поставить у чергу отримане повідомлення із пристрою як вихідне. Хоча це не охоплює реальний випадок використання, воно демонструє, як отримувати пакети інформації, обробляти їх і планувати надсилання повідомлень.

Передбачається, що ви вже налаштували шлюз і пристрій.

Отримання від MQTT

Відкривши Node-RED у своєму браузері, спочатку слід налаштувати **mqtt in** вузол. Цей вузол використовуватиметься для отримання подій пристрою, які публікує брокер MQTT, який працює на шлюзі (наданий ChirpStack Gateway OS).

Необхідно встановити такі властивості вузла:

- Server:
 - Server: localhost
 - Port: 1883
- Topic: application/+/device/+/event/+
- QoS: 0
- Output: auto-detect

Розбір подій

Тепер, коли ви стало можливим отримувати події від MQTT, черга налаштувати вузол **device event**. Цей вузол аналізує корисне навантаження (**payload**), згенероване вузлом MQTT об'єктам і виконуватиме фільтрацію на основі типів подій.

Необхідно встановити такі властивості вузла:

Event Type: Uplink

Тепер потрібно поєднати вихідну сторону вузла **mqtt-in** зі стороною входу вузла **device event**.

Echo-payload функція

Щоб змодельовати подію надсилання повідомлення як корисне навантаження в черзі вхідних, додайте **function** вузол і підключіть вхід до виходу вузла **device event**.

Необхідно встановити такі властивості вузла:

On Message

```
return {
    "payload": msg.payload.data,
    "fPort": msg.payload.fCnt,
    "confirmed": false,
    "devEUI": msg.payload.devEUI
};
```

Черга для вихідної відправки

Для цього спершу потрібно отримати API-токен від ChipStack веб-інтерфейсу. Щоб надіслати вихідне повідомлення, створене функцією **echo** до ChirpStack, слід додати вузол **device downlink** та з'єднати його вхід із виходом **function** вузла.

Для нього необхідно встановити такі властивості:

- Server: localhost:8080
- API token: the API-token отриманий від ChirpStack web-interface
- Payload Encoding: Base64

Вихідна черга (debug)

Щоб переглянути повідомлення, створене вузлом **device downlink**, потрібно додати інший вузол **debug** та з'єднати вхід із виходом вузла **device downlink**.

Після виконаних кроків такий потік може отримувати та надсилати повідомлення від серверу. При відправці вихідного пакету можна бачити такі 2 повідомлення у панелі відлагодження:

- вхідне повідомлення
- відповідь від вихідної черги

Приклад виводу із вихідної черги може виглядати наступним чином:

```
{"fCnt":1, "_msgid":"e3b9c3afa25e45f6"}
```

При передачі пакетів у мережі, вони закодовані у **base64** формат, що дає можливість суттєво зменшити об'єм навантаження, що передається. Корисне навантаження, закодоване у base64 перед передачею в багатьох випадках використання IoT(LoRa, LTE-M, NB-IoT, Sigfox тощо).

Base64 зазвичай використовується для кодування двійкових форматів. Оскільки двійковий формат є власним представленням даних для комп'ютерів, тоді це найпростіший формат для роботи з обмеженими ресурсами вбудованих пристроїв.

Як алгоритм, base64 концептуально досить простий і вимагає дуже мало ресурсів для реалізації, тому це хороший компроміс для стискання двійкових даних через текстовий канал. З іншого боку, для створення запису JSON зазвичай потрібна бібліотека JSON, яка споживає оперативну пам'ять і простір для коду.

Якщо потрібно закодувати запис даних, що складається з 32-бітної мітки часу, 8-бітного коду, що визначає тип даних (тобто температура, напруга або тиск), і 32-бітної вибірки даних, тоді буде використано 9 байт, що буде розширено до 12 байт після кодування за допомогою base64.

У порівнянні з цим, простий записом у JSON матиме 67 байтів без урахування всіх пробілів

3.2. Розробка апаратної частини кінцевого пристрою на ESP32

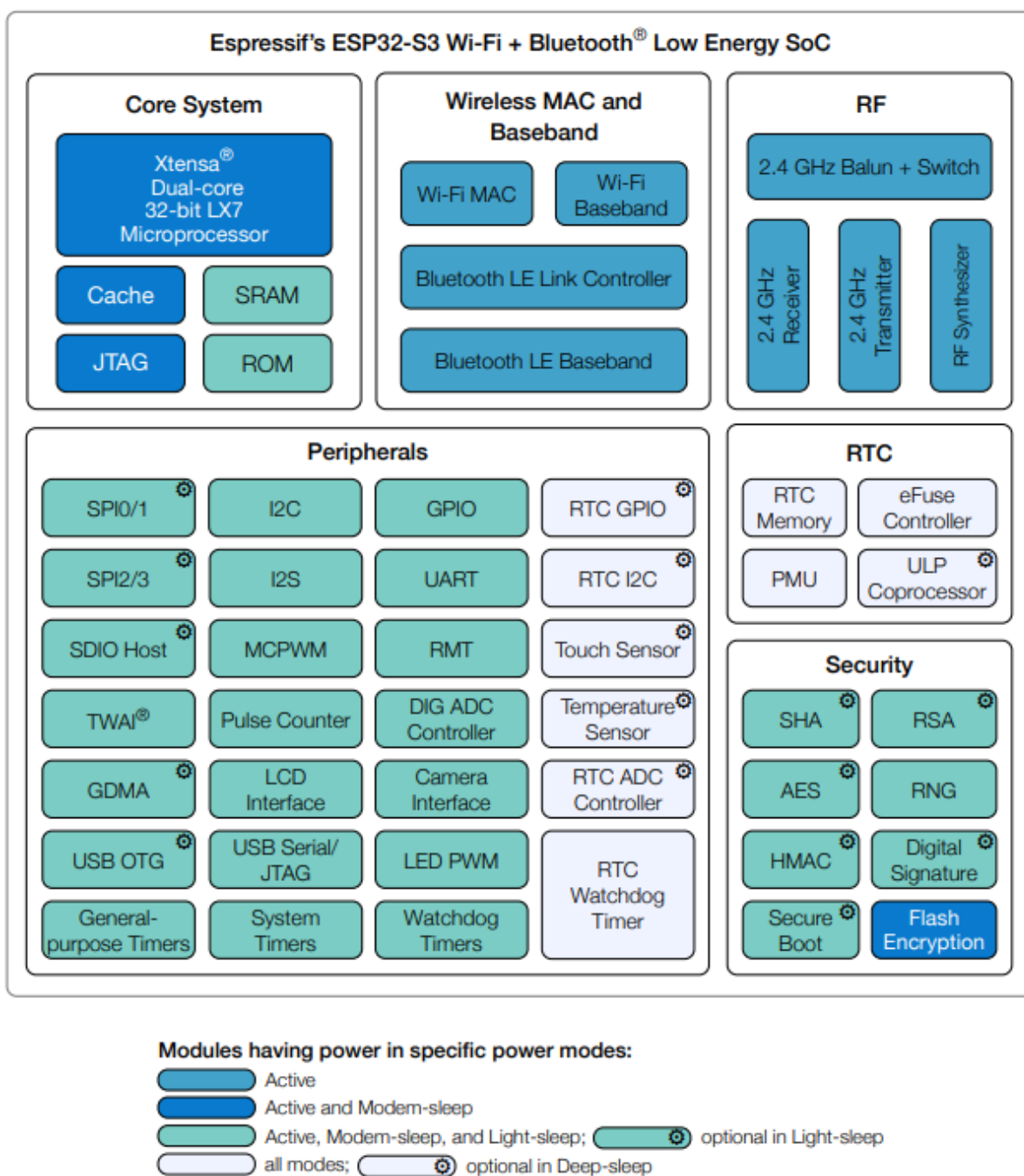


Рисунок 3.4. ESP32-S3 Блок-діаграма

На Рисунку 3.4 проілюстровані можливості використовуваного SoC. Для комунікації із LoRa-модулем використовуватиметься один із UART-інтерфейсів (universal asynchronous receiver-transmitter), яких на борту є 3.

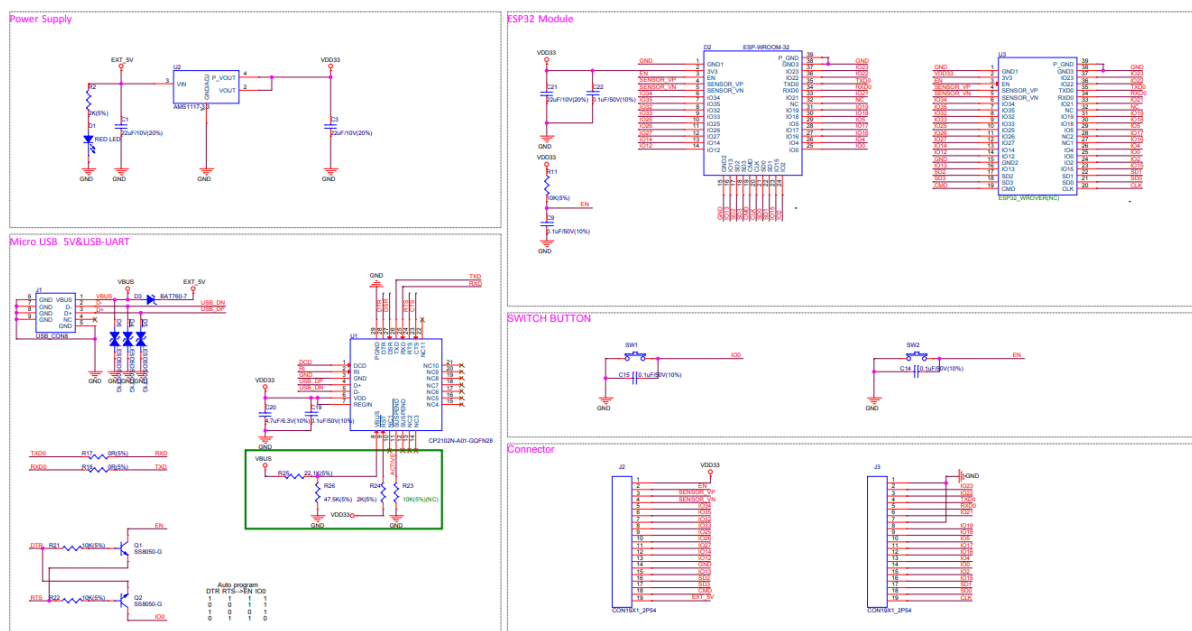


Рисунок 3.5. Схема підключень модулів Espressif ESP32 WROOVER-KIT

На схемі виробник приводить підключення входів-виходів (ІО) процесора та додаткових компонентів таких як USB-модуль, обв'язка живлення плати, кнопки та порядок ІО на “гребінці” модуля розробника.

Трошки більш детально варто розглянути сам процесор, або ж як його в таких речах прийнято називати SoC – System on chip.

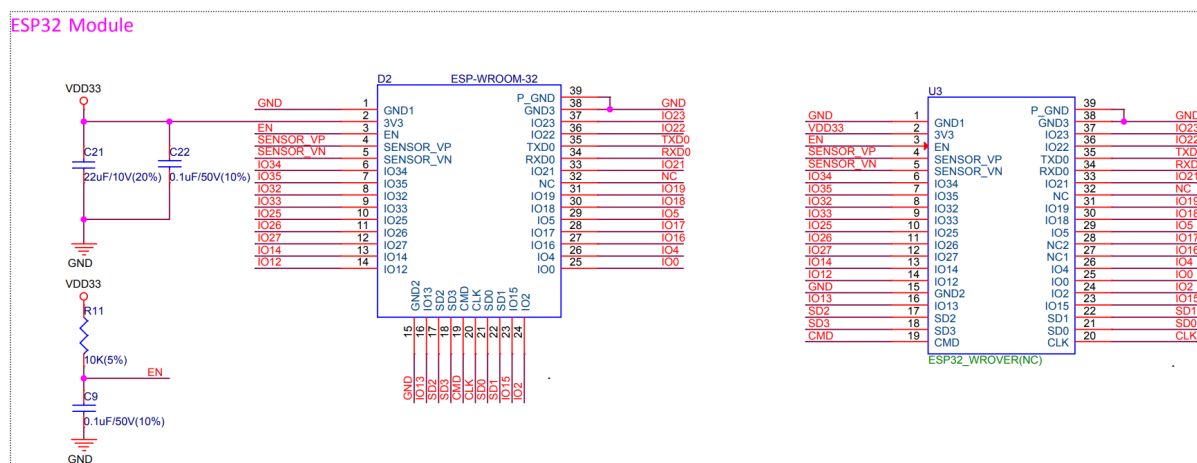


Рисунок 3.6. ESP32 Module

ESP32 — це загальна продуктова назва чіпа, розробленого компанією Espressif Systems. Він дає можливість використання Wi-Fi, а в деяких моделях дворежимне підключення Bluetooth до вбудованих пристроїв. Хоча технічно

ESP32 — це лише чіп, модулі та плати розробки, які містять цей чіп, виробники часто також називають «ESP32».

Чіп ESP32 має мікропроцесор Tensilica Xtensa LX6 як у двоядерному, так і в одноядерному варіантах із тактовою частотою понад 240 МГц.. ESP32 також включає в себе найсучасніші функції, такі як точний clock-gating (технологія зменшення енергоспоживання в цифрових системах за рахунок заборони подачі синхросигналів на невикористовувані ланки схеми, виключаючи витрачання енергії на марне переключення їх елементів), різні режими живлення та динамічне масштабування потужності. ESP32 добре інтегрований із вбудованими перемикачами антени, радіочастотним балансиrom, підсилювачем потужності, малошумним приймальним підсилювачем, фільтрами та модулями керування живленням.

Контроллер має надзвичайно широкий спектр застосувань для свого цінового класу завдяки різноманітним інтерфейсам підключень.

ESP32-DevKitC

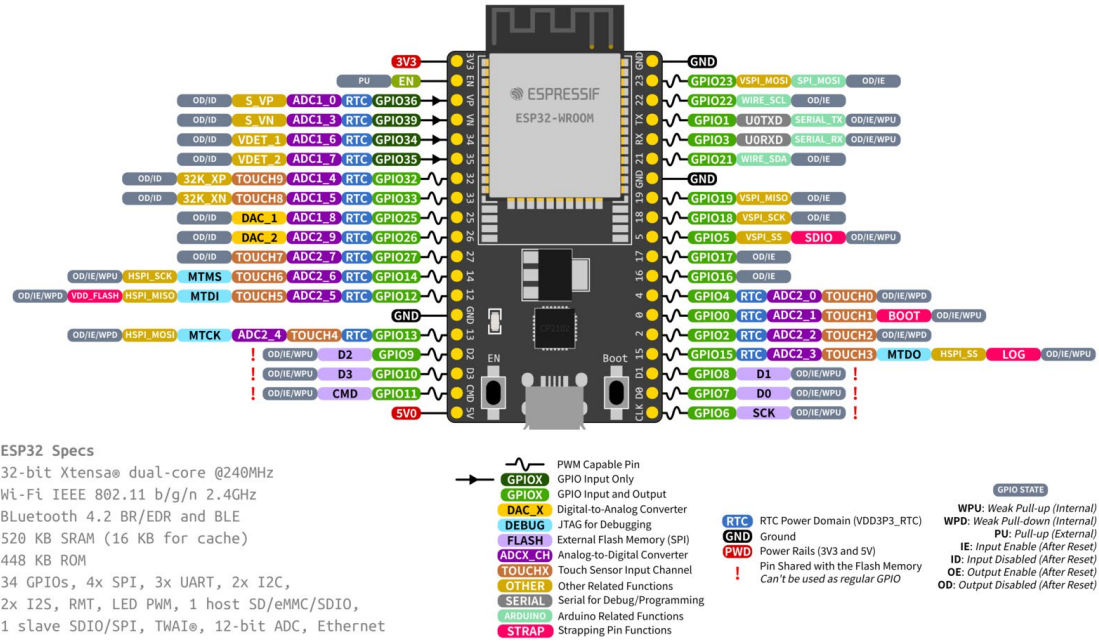


Рисунок 3.7. ESP32-DevKitC pinout

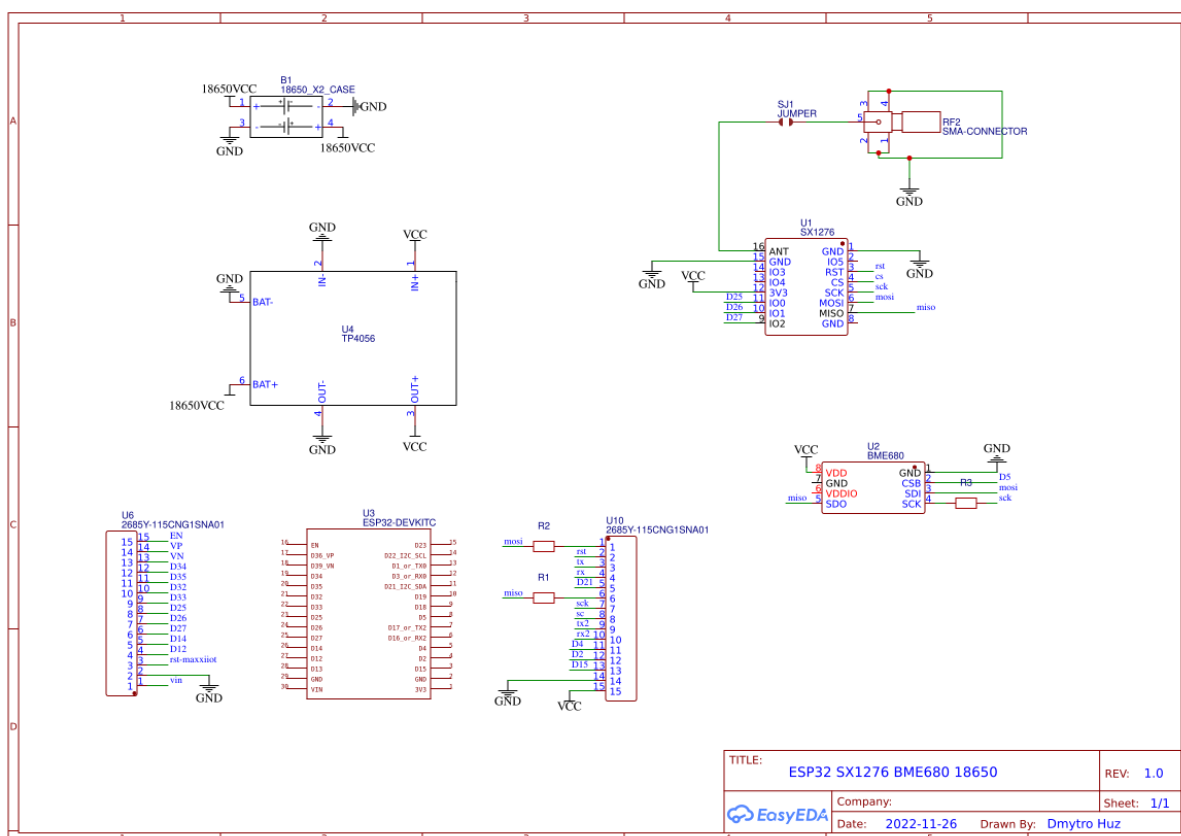


Рисунок 3.8. Схема підключень модулів граничного пристрою

На схемі зображені підключення контролера ESP32 до модему SX1276 з допомогою UART у безкоштовному середовищі розробки електронних компонентів, схем та друкованих плат **EasyEDA**.

Модуль розробника монтується у дві 15-контактні колодки, що можуть бути розташовані на друкованій платі. LoRa модуль має антенний вихід, до якого підключена SMA-антена 868 МГц через зручний конектор. Для підтримання роботи такої системи в умовах відсутності мережевого струму обрано збірку із двох акумуляторних батарей типорозміру 18650 з напругою 3.7 В. Основний спосіб живлення схеми – мережеве підключення по місцю, але також передбачається резервне підживлення батарей посередництвом дообладнання монокристалічною сонячною панеллю 1...5 Вт, що пасуватиме до розмірів корпусу готового приладу. Багатофункціональний сенсор BME680 Bosch підключений через I2C інтерфейс, по якому відбувається шляхом отримання параметрів опитування його за адресою, якими оперує система в

подальшому на інших рівнях. На Рисунку 3.9 показана спроектована друкована плата граничного пристрою, де

1. Сенсор BME680
2. RF-модуль LoRa SX1278
3. SMA вивідна антена
4. Модуль заряду TP4056
5. Корпус для акумуляторів типорозміру 18650

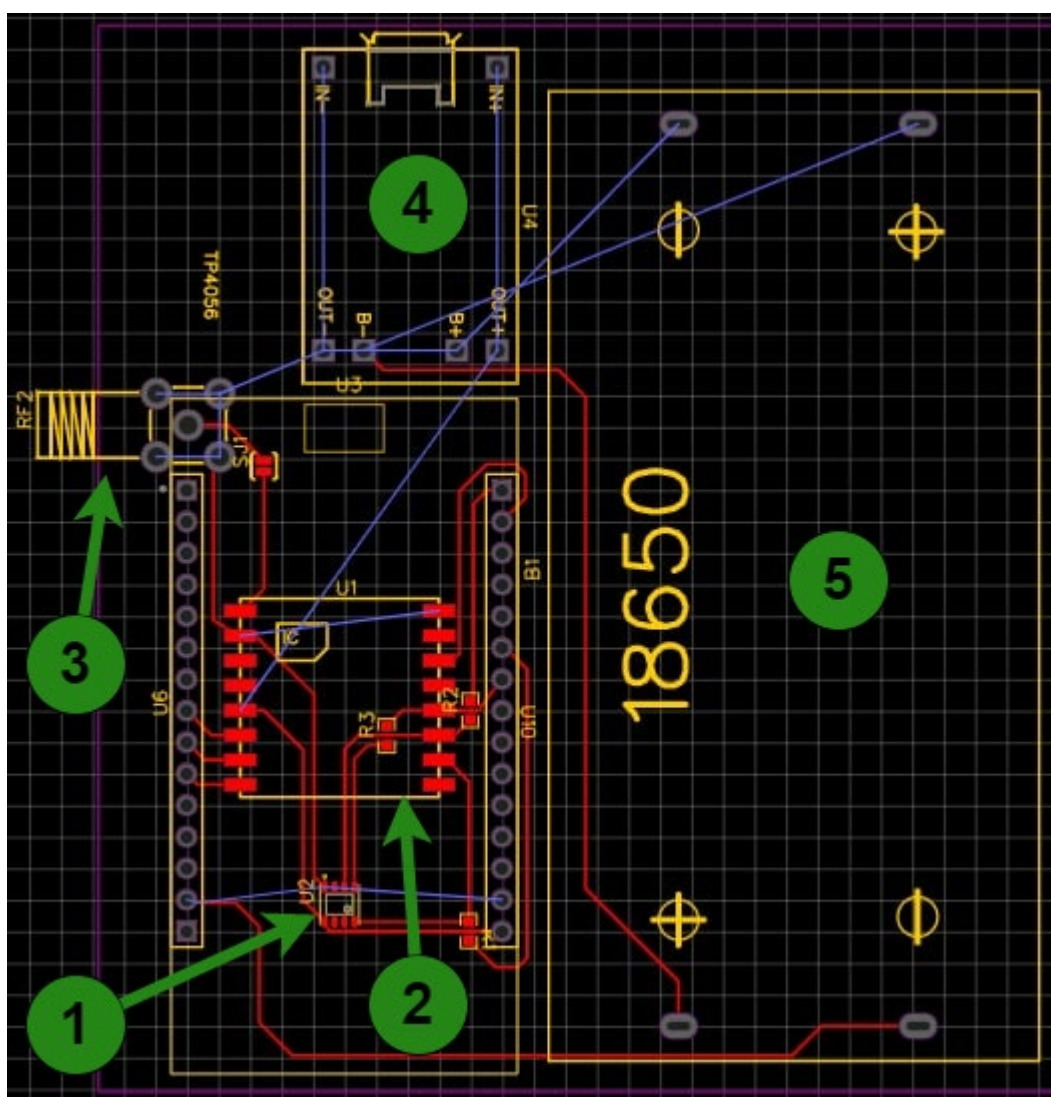


Рисунок 3.9. Схема компонентів друкованої плати

У об'ємному відображенні текстолітова плата до монтажу компонентів матиме приблизно наступний вигляд, Рисунок 3.10:

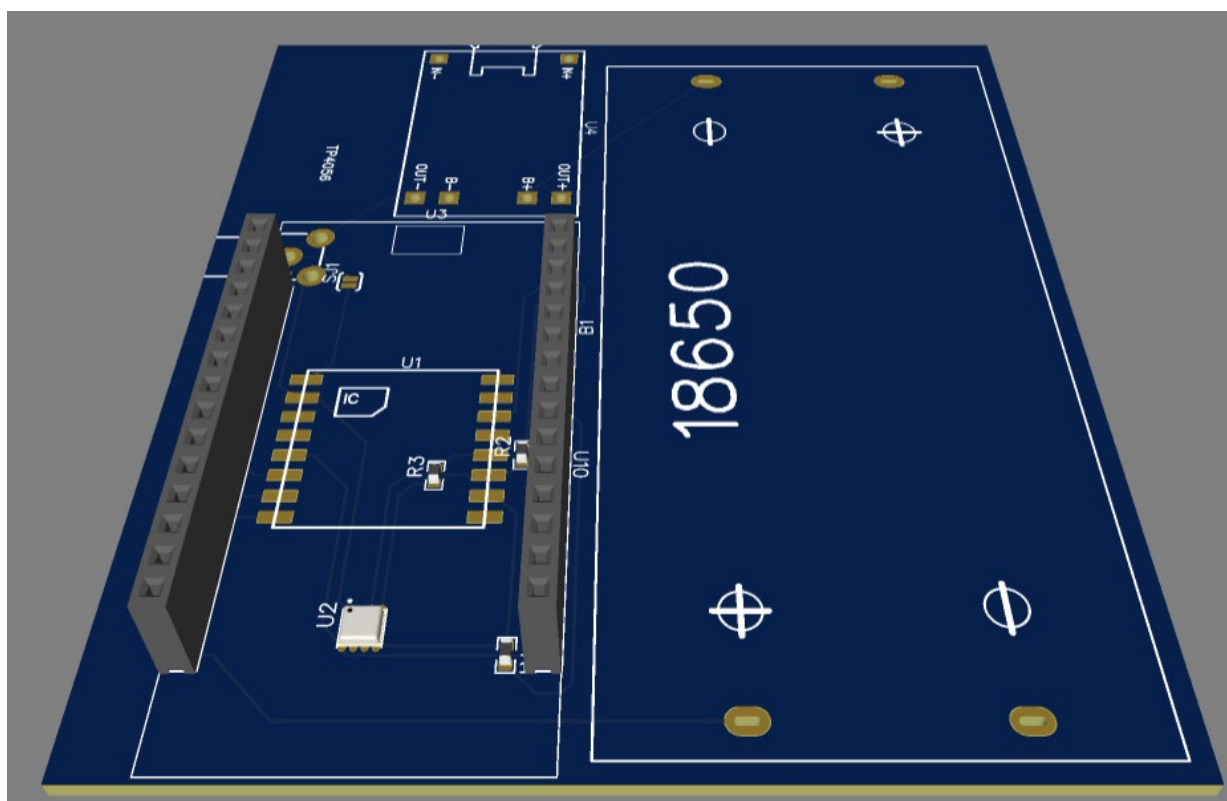


Рисунок 3.10. Відображення друкованої плати в об'ємі

3.3. Розробка програмного забезпечення граничного пристрою

Розробка програмного забезпечення для ESP32 базується на популярному C/C++ фреймворку Arduino [8]. Це дає можливість швидкого старту розробки прототипів рішень із порівняно низьким порогом входу, тобто розробнику не потрібно опановувати надмірно великий обсяг знань. Водночас це відкриває можливості застосування багатьох бібліотек для безлічі інтерфейсів і протоколів, що, як правило, підтримуються спільнотою розробників.

Розберемо основні функції, що використовуються в роботі пристрою.

Відправка повідомлення відбувається через взаємодію по Serial-інтерфейсу. LoRa-модуль повинен відправити сформовану AT-команду із бажаним корисним навантаженням на сервер. Базовий синтаксис такої команди від виробника пристрою поданий на Рисунку 3.11:

10. AT+SEND Send data to the appointed address by Command Mode.

Syntax	Response
AT+SEND=<Address>,<Payload Length>,<Data> <Address>0~65535, When the <Address> is 0, it will send data to all address (From 0 to 65535.) <Payload Length> Maximum 240bytes <Data>ASCII Format Example : Send HELLO string to the Address 50, AT+SEND=50,5,HELLO	+OK
Search last transmit data, AT+SEND?	+SEND=50,5,HELLO

Рисунок 3.11 Синтаксис команди комунікації

Реалізація функції формування та відправки команди висвітлена на Рисунку 3.12:

```
void send_lora_message(String payload)
{
    String base = "AT+LRSEND=";
    String port = String(LORA_PORT);
    base = base + port + ",1,";
    String length = String((((payload.length()) / 2) + 1));
    base = base + length + "<" + payload;
    base.toUpperCase();
    PC.println(base);
    Lora.println(base);
}
```

Рисунок 3.12 Реалізація функції відправки повідомлення на сервер

Для розбору вхідного повідомлення від серверу створена функція “LoRaEvent”, тіло якої на Рисунок 3.13:

```

151 void LoRaEvent()
152 {
153     if (Lora.available() > 0)
154     {
155         String incoming = Lora.readStringUntil('\n');
156 #ifdef PC_DEBUG
157         PC.println(incoming);
158 #endif
159         byte indexcommand = incoming.indexOf(":");
160         message.type = incoming.substring(1, indexcommand);
161         indexcommand += 1;
162         byte subindex = incoming.indexOf(",", indexcommand);
163         message.counter = incoming.substring(indexcommand, subindex);
164         subindex += 1;
165         indexcommand = incoming.indexOf(",", subindex);
166         message.port = incoming.substring(subindex, indexcommand);
167         indexcommand += 1;
168         subindex = incoming.indexOf(",", indexcommand);
169         message.rssi = incoming.substring(indexcommand, subindex);
170         subindex += 1;
171         indexcommand = incoming.indexOf(",", subindex);
172         message.snr = incoming.substring(subindex, indexcommand);
173         indexcommand += 1;
174         subindex = incoming.indexOf(",", indexcommand);
175         message.length = incoming.substring(indexcommand, subindex);
176         subindex += 2;
177         indexcommand = incoming.indexOf(",", subindex);
178         message.payload = incoming.substring(subindex, indexcommand);
179         indexcommand += 1;
180         subindex = incoming.indexOf(",", indexcommand);
181         message.freq = incoming.substring(indexcommand, subindex);
182         subindex += 1;
183         indexcommand = incoming.indexOf(",", subindex);
184         message.data = incoming.substring(subindex, indexcommand);
185         incoming = "";
186         return;
187     }
188 }

```

Рисунок 3.13. Функція розбору вхідного повідомлення “LoRaEvent”

Для роботи із повідомленнями в мережі створено структуру “struct loramessage”, що містить поля потрібних основних параметрів при взаємодії пристрою із сервером, Рисунок 3.14:

```

52  struct loramessage
53  {
54      String type;
55      String counter;
56      String port;
57      String rssi;
58      String snr;
59      String length;
60      String payload;
61      String freq;
62      String data;
63      String previous_type;
64  };
65  extern struct loramessage message;

```

Рисунок 3.14. Структура для LoRa повідомлень “*struct loramessage*”

В основному роль пристрою – надсилання зібраних із датчиків даних. Та й для отримання вхідних повідомлень передбачений функціонал: звірення часу та встановлення нового порогу значень, як показано на Рисунку 3.15:

```

void LoRaMain()
{
    if (message.type == "LRRECV")
    {
        int zerobyte = str_to_byte(message.payload.substring(0, 2));
        Serial.println("Message received at " + timer.hour_DEC + ":" + timer.minute_DEC + ":" + timer.second_DEC +
            " " + timer.dayOfMonth + "." + timer.month + ":" + timer.year);
        Serial.println(message.payload);
        switch (zerobyte)
        {
            case 0:
                send_lora_message("10", (timer.hour_HEX + timer.minute_HEX + timer.second_HEX));
                break;
            case 1:
                set_new_limit(message.payload);
                send_lora_message("11", message.payload.substring(2, 4));
                break;
        }
    }
}

```

Рисунок 3.15. Функція розбору вхідного повідомлення від серверу

В робочому циклі програми пристрій після ініціалізації інтерфейсів опитує сенсор, відправляє дані на сервер та відправляється у deep-sleep режим, споживаючи мінімум струму до наступного виходу в ефір.

На Рисунку 3.16 показані елементи периферії, що залишається активною при входженні у режим “глибокого сну” контролера ESP32: це процесор ультра низької потужності, а також RTC-область пам’яті. У звичайному ж

режимі споживання струму при роботі становить 160 ... 260 мА, а в окремих випадках досягає 790мА. Відповідно до цього, застосовуючи режим “глибокого сну” контролеру можна забезпечити 16-70 кратне заощадження споживаної енергії.

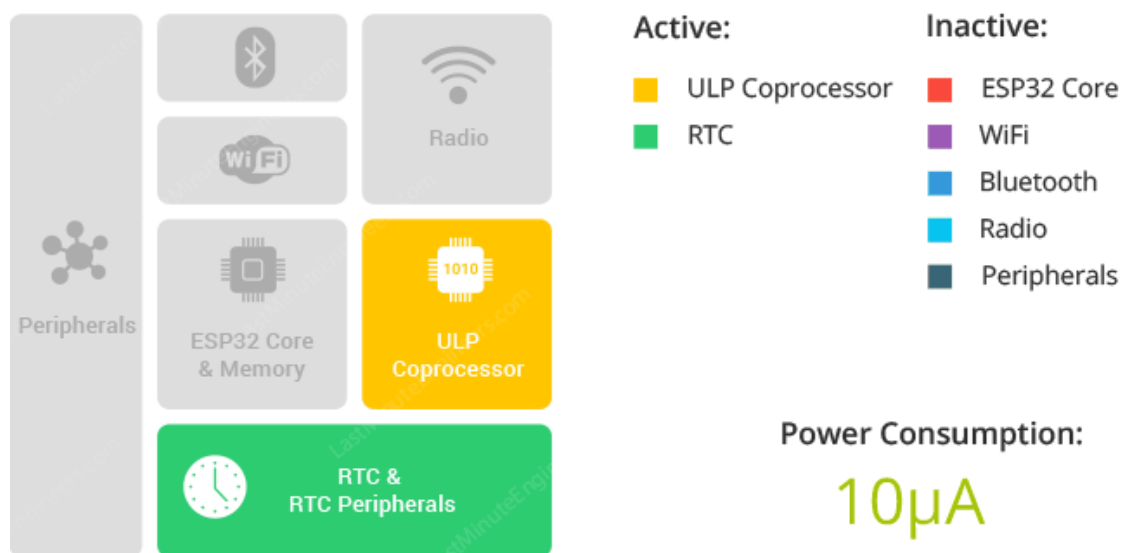


Рисунок 3.16. Активні модулі при режимі “глибокого сну” контролера

Для фільтрації зчитуваних із сенсора значень параметрів застосовано цілочисельний алгоритм фільтрації “медіанний фільтр”. Цей підхід передбачає використання пов’язаного списку. Працює він завдяки спостереженню за тим, що після сортування масиву дія видалення найстарішого значення та вставлення найновішого значення не призводить до значного розсортування масиву. У результаті цього алгоритм показує хороші результати, особливо для фільтрування масивів великих розмірів. Цей підхід може споживати помітну кількість оперативної пам’яті, оскільки потрібно зберігати як значення, так і показники. Однак це справді ефективний алгоритм, оскільки він значно швидший за алгоритми, які спираються на сортування.

Імплементація його показана на Рисунку 3.17:

```

> E: \utils\cpp > median_filter(uint16_t)
6 #define MEDIAN_FILTER_SIZE (5)
7 #define STOPPER 0 /* Smaller than any datum */
8
9 uint16_t median_filter(uint16_t datum)
10 {
11     struct pair
12     {
13         struct pair *point; /* Pointers forming list linked in sorted order */
14         uint16_t value; /* Values to sort */
15     };
16     static struct pair buffer[MEDIAN_FILTER_SIZE] = {0}; /* Buffer of nwidth pairs */
17     static struct pair *datapoint = buffer; /* Pointer into circular buffer of data */
18     static struct pair small = {NULL, STOPPER}; /* Chain stopper */
19     static struct pair big = {&small, 0}; /* Pointer to head (largest) of linked list.*/
20
21     struct pair *successor; /* Pointer to successor of replaced data item */
22     struct pair *scan; /* Pointer used to scan down the sorted list */
23     struct pair *scanold; /* Previous value of scan */
24     struct pair *median; /* Pointer to median */
25     uint16_t i;
26
27     if (datum == STOPPER)
28     {
29         datum = STOPPER + 1; /* No stoppers allowed. */
30     }
31
32     if ((++datapoint - buffer) >= MEDIAN_FILTER_SIZE)
33     {
34         datapoint = buffer; /* Increment and wrap data in pointer.*/
35     }
36
37     datapoint->value = datum; /* Copy in new datum */
38     successor = datapoint->point; /* Save pointer to old value's successor */
39     median = &big; /* Median initially to first in chain */
40     scanold = NULL; /* Scanold initially null. */
41     scan = &big; /* Points to pointer to first (largest) datum in chain */
42
43     /* Handle chain-out of first item in chain as special case */
44     if (scan->point == datapoint)
45     {
46         scan->point = successor;
47     }
48     scanold = scan; /* Save this pointer and */
49     scan = scan->point; /* step down chain */
50
51     /* Loop through the chain, normal loop exit via break. */
52     for (i = 0; i < MEDIAN_FILTER_SIZE; ++i)
53     {
54         /* Handle odd-numbered item in chain */
55         if (scan->point == datapoint)
56         {
57             scan->point = successor; /* Chain out the old datum.*/
58         }
59
60         if (scan->value < datum) /* If datum is larger than scanned value,*/
61         {
62             datapoint->point = scanold->point; /* Chain it in here. */
63             scanold->point = datapoint; /* Mark it chained in. */
64             datum = STOPPER;
65         }
66
67         /* Step median pointer down chain after doing odd-numbered element */
68         median = median->point; /* Step median pointer. */
69         if (scan == &small)
70         {
71             break; /* Break at end of chain */
72         }
73         scanold = scan; /* Save this pointer and */
74         scan = scan->point; /* step down chain */
75
76         /* Handle even-numbered item in chain. */
77         if (scan->point == datapoint)
78         {
79             scan->point = successor;
80         }
81         if (scan->value < datum)
82         {
83             datapoint->point = scanold->point;
84             scanold->point = datapoint;
85             datum = STOPPER;
86         }
87         if (scan == &small) break;
88         scanold = scan;
89         scan = scan->point;
90     }
91     return median->value;
92 }

```

Рисунок 3.17. Реалізація алгоритму медіанного фільтру

У наступному розділі буде розглянута робота змодельованої системи із використанням MQTT для вузол-серверної взаємодії та Wi-Fi інтерфейсу для бездротової передачі пакетів від кінцевого пристрою до серверного back-end. Панель оператора може мати ідентичний вигляд як для тестового, так і для продуктового вихідного виконання. Зміни стосуються лише Node-RED потоків для отримання значень від серверу.

3.4. Моделювання роботи системи

Моделювання роботи системи моніторингу проведено із використанням MQTT протоколу. В цьому розділі відображається обмін повідомлень між кінцевим пристроєм (ESP32) та серверною частиною (MQTT-брокер), яка виводить отримані дані на панель Node-RED із візуалізацією.

Було зібрано демонстраційний набір на макетній платі, на Рисунку 3.18:

- основний контролер ESP32 (підключений по USB до комп'ютера, для перевірки робочої моделі працюватиме по Wi-Fi інтерфейсу);
- датчик температури та вологості DHT11 (підключення на цифровий GPIO D4);

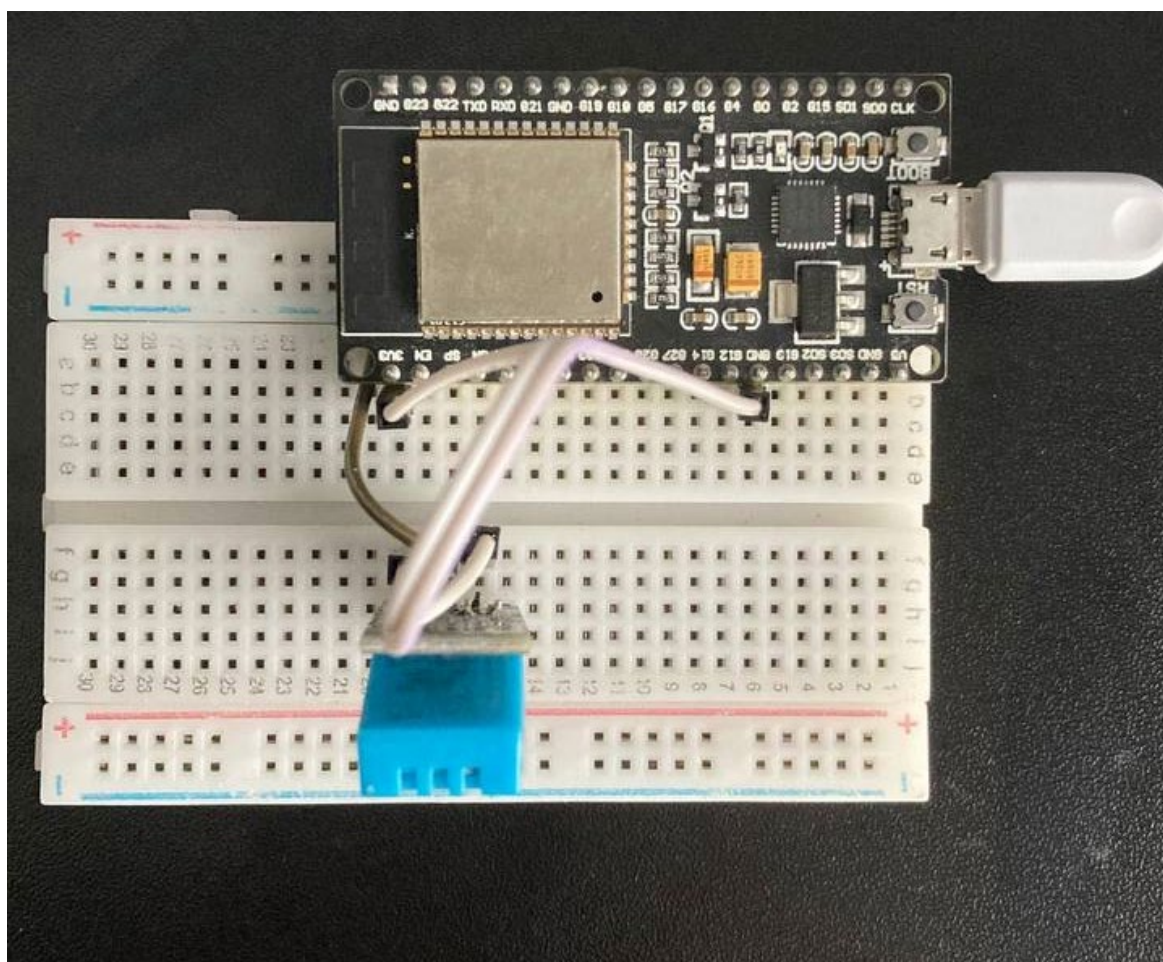


Рисунок 3.18. Збірка граничного пристрою для демонстраційної роботи

На Рисунку 3.19 представлені дві основні функції, із яких починається робота контролеру:

```

void setup() {
    Serial.begin(115200);

    WiFi.begin(ssid, pass);

    client.begin(BROKER_URI, net);
    client.onMessage(messageReceived);

    connect();
    sleep_setup();
}

void loop() {
    client.loop();
    if (!client.connected()) {
        connect();
    }
    bme680_routine();
}

```

Рисунок 3.19. Тіла основних функцій роботи пристрою

Після подачі живлення пристрій входить у свою функцію *setup()*, де відбувається первинна ініціалізація інтерфейсів та їх підключень, підключення встановленої периферії.

Для комунікації із терміналом ініціалізований послідовний порт *Serial* на швидкості у 115200 бод/с. Для під'єднання до Wi-Fi мережі використовується метод *begin* об'єкту класу *WiFi* із зазначеними параметрами мережі.

Аби встановити з'єднання із MQTT брокером викликана ініціалізація його об'єкту класу *client.begin()* і разом із цим цьому об'єкту прикріплюється callback-функція для опрацювання вхідних повідомлень від брокера.

У тілі функції приєднання до брокеру відбувається перевірка підключення Wi-Fi мережі, після чого безпосередньо саме підключення до вказаних тем (topic), Рисунок 3.20:

```

60 void connect() {
61     Serial.print("checking wifi...");
62     while (WiFi.status() != WL_CONNECTED) {
63         Serial.print(".");
64         delay(1000);
65     }
66     #ifdef DEBUG
67     Serial.println("IP address: ");
68     Serial.println(WiFi.localIP());
69     #endif // DEBUG
70
71     Serial.println("connecting...");
72     while (!client.connect("monitoring-esp32", "", "")) {
73         Serial.print(".");
74         client.disconnect();
75         delay(2000);
76     }
77
78     Serial.println("connected!");
79
80     client.subscribe("monitoring-esp32/#");
81 }

```

Рисунок 3.20. Тіло функції приєднання MQTT-брокера *connect()*

Оскільки підключення відбувається до публічного відкритого брокеру, то авторизація не потрібна. Для випадків застосування MQTTS (Secured), де потрібна авторизація TLS, дані для входу можна вказати додатковими аргументами у *connect()*-методі після топіку.

Коренева тема для підписки на події є “monitoring-esp32/”, отже підписка здійснюється на всі дочірні теми після неї, вказавши «#» після кореневої.

Після проходження *setup()* хід програми більше не повертається до неї, а переходить в циклічну функцію *loop()* , яка постійно повторюється.

У її тілі відтворюється опрацювання зв'язку із MQTT брокером і також опитування встановлених сенсорів. Функція опитування давача збирає значення параметрів та після цього відправляє їх брокеру та здійснює відлагоджувальний вивід у термінал послідовного порту, вона має вигляд Рисунок 3.21:

```

68 void bme680_routine()
69 {
70     if (millis() - lastMillis > TIMEOUT)
71     {
72         lastMillis = millis();    // to reset timer
73
74         bme680_set_params();
75
76         Serial.print(F("Reading completed at "));
77         Serial.println(millis());
78
79         Serial.print(F("Temperature = "));
80         Serial.print(bme.temperature);
81         Serial.println(F(" *C"));
82
83         Serial.print(F("Pressure = "));
84         Serial.print(bme.pressure);
85         Serial.println(F(" hPa"));
86
87         Serial.print(F("Humidity = "));
88         Serial.print(bme.humidity);
89         Serial.println(F(" %"));
90
91         Serial.print(F("Gas = "));
92         Serial.print(bme.gas_resistance);
93         Serial.println(F(" KOhms"));
94
95         bme680_publish_params();
96     }
97 }

```

Рисунок 3.21. Функція збору значень із сенсора

Оскільки обмін даними повинен відбуватися у текстовому форматі, то перед публікацією даних їх потрібно привести із чисельного формату, в якому ці дані зберігаються в об'єкті сенсора, до потрібного типу, ним обраний *String*-тип, що є Arduino-інтерпретацією масиву *char*. Функція для відправки даних по

темам показана на Рисунку , де окремо надсилаються значення температури, тиску, вологості та газових параметрів:

```
void bme680_publish_params()
{
    client.publish("monitoring-esp32/temperature", String(bme.temperature));
    client.publish("monitoring-esp32/pressure", String(bme.pressure));
    client.publish("monitoring-esp32/humidity", String(bme.humidity));
    client.publish("monitoring-esp32/gas-resistance", String(bme.gas_resistance));
}
```

Рисунок 3.22. Функція відправки даних на MQTT брокер

У терміналі Windows PowerShell запущено Node-RED сервер, потік якого підключений до публічного mqtt-брокера, Рисунок 3.23:

```

node-red
4 Dec 16:45:59 - [warn] [mqtt out:Set output] Error: Invalid topic specified
4 Dec 16:46:41 - [info] Stopping flows
4 Dec 16:46:41 - [info] Stopped flows
4 Dec 16:46:41 - [info] Updated flows
4 Dec 16:46:41 - [info] Starting flows
4 Dec 16:46:41 - [info] Started flows
4 Dec 16:46:41 - [info] [mqtt-broker:public broker] Connected to broker: mqtt://broker.emqx.io:1883
4 Dec 16:47:25 - [info] Stopping flows
4 Dec 16:47:25 - [info] Stopped flows
4 Dec 16:47:25 - [info] Updated flows
4 Dec 16:47:25 - [info] Starting flows
4 Dec 16:47:25 - [info] Started flows
4 Dec 16:47:26 - [info] [mqtt-broker:public broker] Connected to broker: mqtt://broker.emqx.io:1883
4 Dec 16:48:03 - [info] Stopping flows
4 Dec 16:48:03 - [info] Stopped flows
4 Dec 16:48:03 - [info] Updated flows
4 Dec 16:48:03 - [info] Starting flows
4 Dec 16:48:03 - [info] Started flows
4 Dec 16:48:04 - [info] [mqtt-broker:public broker] Connected to broker: mqtt://broker.emqx.io:1883
4 Dec 16:58:26 - [info] Stopping flows
4 Dec 16:58:26 - [info] Stopped flows
4 Dec 16:58:26 - [info] Updated flows
4 Dec 16:58:26 - [info] Starting flows
4 Dec 16:58:26 - [info] Started flows
4 Dec 16:58:27 - [info] [mqtt-broker:public broker] Connected to broker: mqtt://broker.emqx.io:1883
4 Dec 16:58:48 - [info] Stopping flows
4 Dec 16:58:48 - [info] Stopped flows
4 Dec 16:58:48 - [info] Updated flows
4 Dec 16:58:48 - [info] Starting flows
4 Dec 16:58:48 - [info] Started flows
4 Dec 16:58:48 - [info] [mqtt-broker:public broker] Connected to broker: mqtt://broker.emqx.io:1883
4 Dec 17:00:18 - [info] Stopping flows
4 Dec 17:00:18 - [info] Stopped flows
4 Dec 17:00:18 - [info] Updated flows
4 Dec 17:00:18 - [info] Starting flows
4 Dec 17:00:18 - [info] Started flows
4 Dec 17:00:19 - [info] [mqtt-broker:public broker] Connected to broker: mqtt://broker.emqx.io:1883
  
```

Рисунок 3.23 Термінал роботи Node-RED серверу

Мікроконтролер публікує свої вимірювання на брокер за тією ж адресою, відправку повідомлень можна перевірити через mqtt-клієнт MQTT-Explorer. На Рисунках 3.24, 3.25 видно, що по топікам дані проходять:

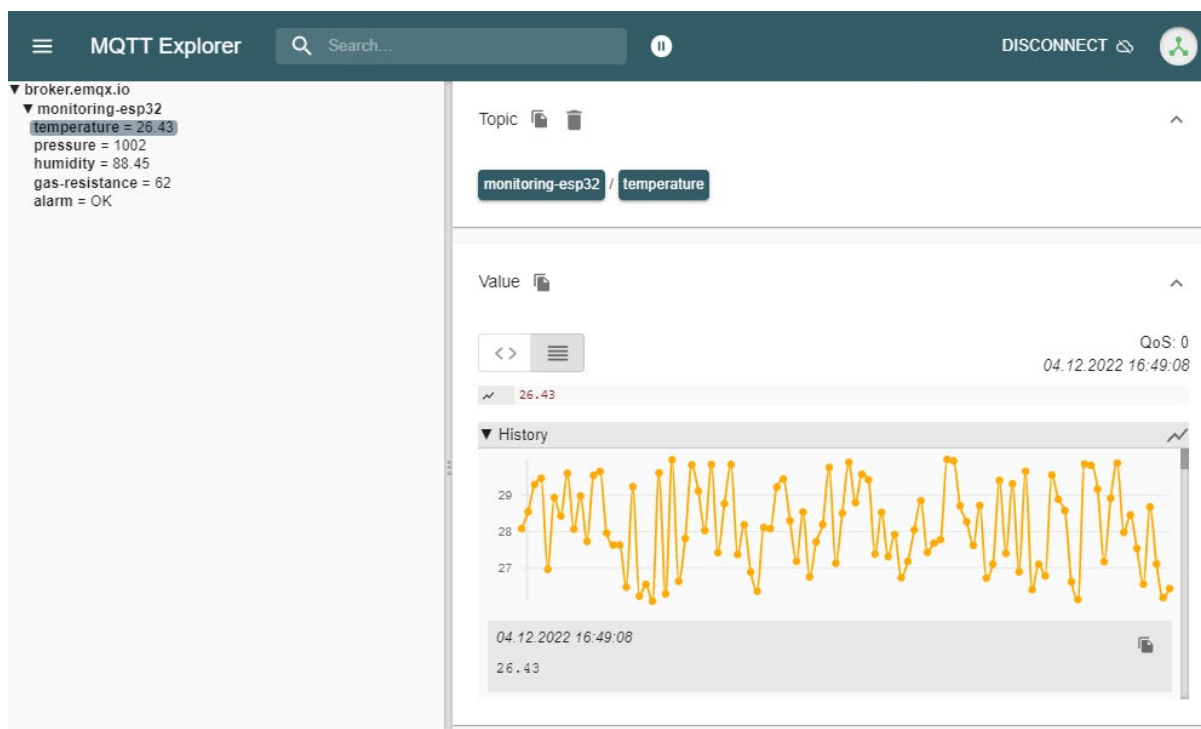


Рисунок 3.24.

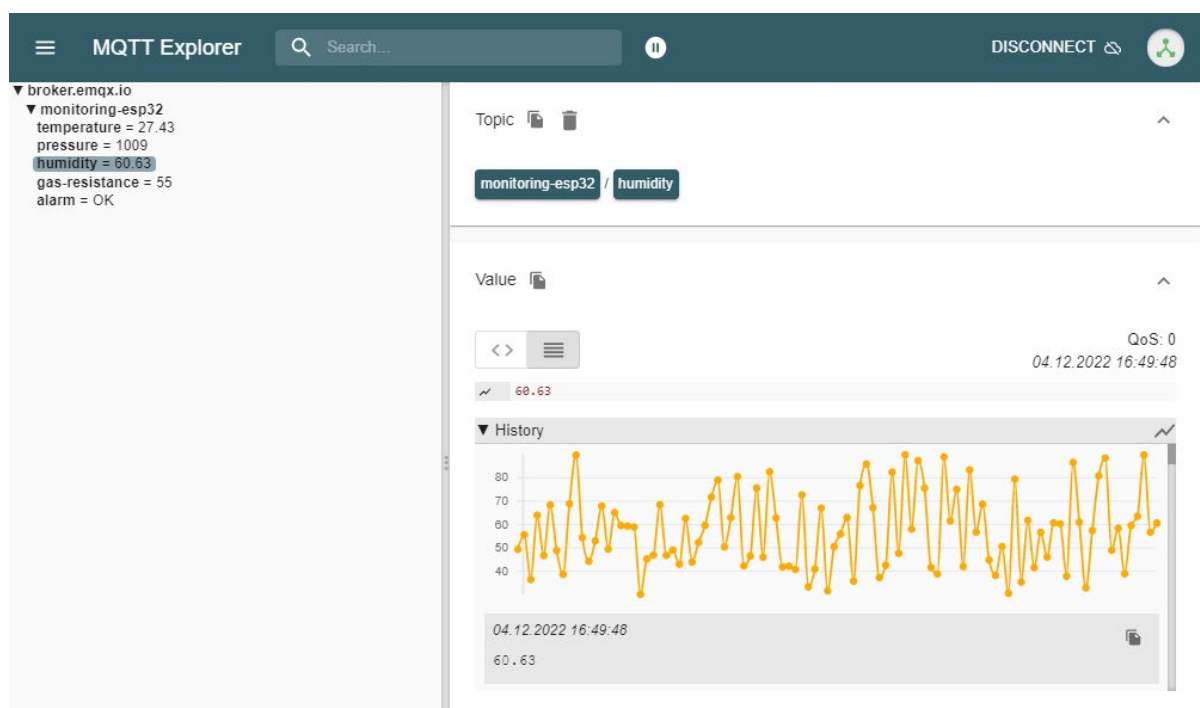


Рисунок 3.25. Отримані із пристрою дані топіку вологості “humidity”

Потік, створений для обміну інформацією, зображений на Рисунку 3.26:

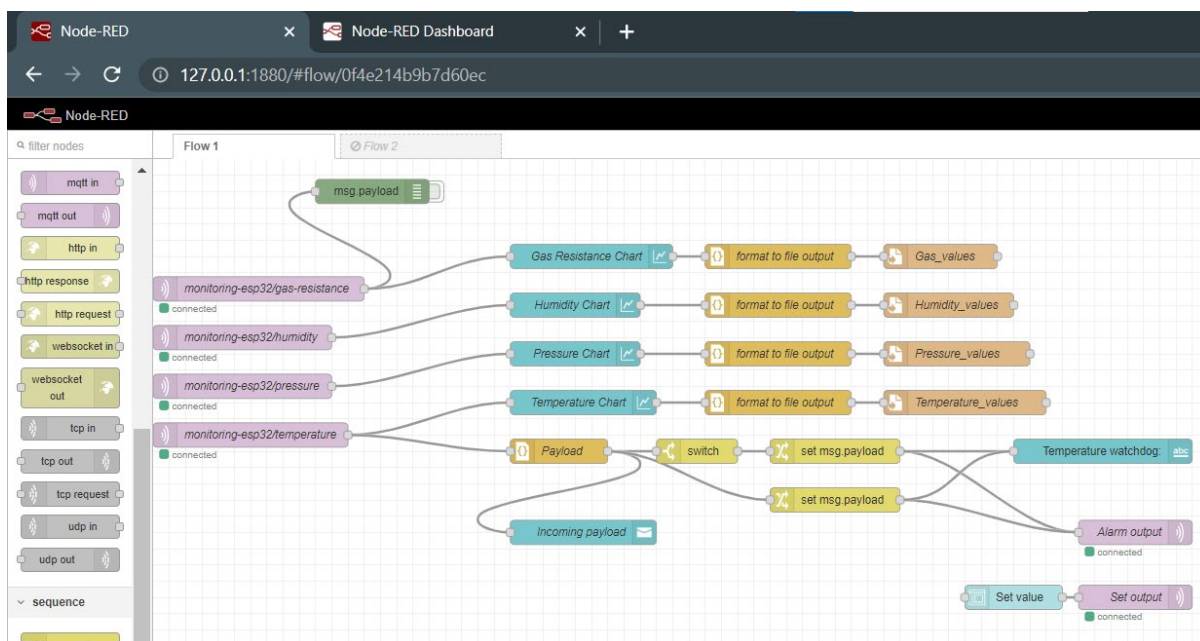


Рисунок 3.26. Потоки для значень параметрів

Mqtt-in вузли “monitoring-esp32/...” спрямовують повідомлення із брокера у потік, де вони опрацьовуватимуться. Створено 4 графіки, що відображають входні параметри: Temperature Chart (графік температури), Pressure Chart (графік тиску), Humidity Chart (графік вологості), Gas Resistance Chart (графік).

Дані із графіків архівуються в текстові файли у json-форматі в локальне сховище пристрою, на якому працює Node-RED. Приклад вмісту архівованого файлу на Рисунку 3.27.

Налаштування вузла архівації має вигляд Рисунку 3.28, шлях до файлу повинен мати абсолютний формат, або він створиться відносно директорії, в якій працює Node-RED.

 *Temperature_values: Блокнот

Файл Редагування Формат Вигляд Довідка

```

[
  {
    "series": [
      "monitoring-esp32/temperature"
    ],
    "data": [
      [],
      [
        {
          "x": 1670176263288,
          "y": 26.01
        },
        {
          "x": 1670176268293,
          "y": 29.82
        },
        {
          "x": 1670176273303,
          "y": 27.34
        },
        {
          "x": 1670176278307,
          "y": 26.51
        },
        {

```

Рисунок 3.27. Вміст архівованого файлу, де X – значення часу події від старту(сек), а Y – значення вимірюваної величини

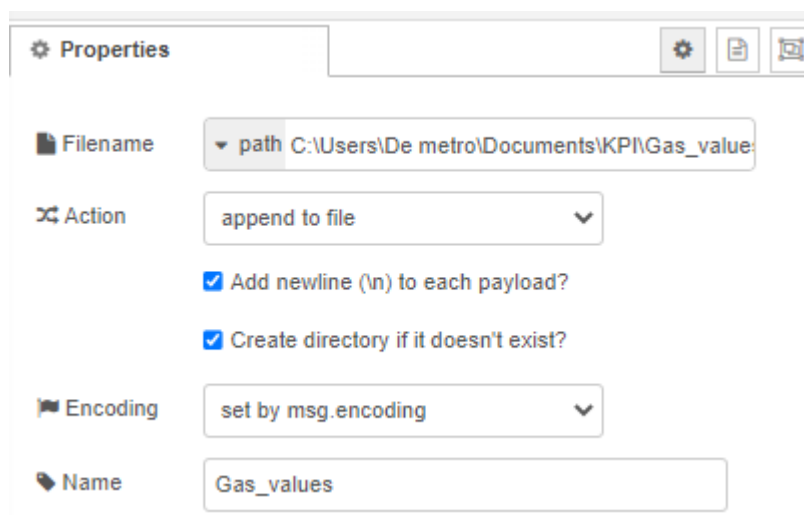


Рисунок 3.28. Приклад конфігурації вузла архівування

Для потоку температури передбачено слідкування за перевищенням встановленого значення, що у випадку спрацювання викличе зміну стану Temperature Watchdog, Рисунок 3.29:

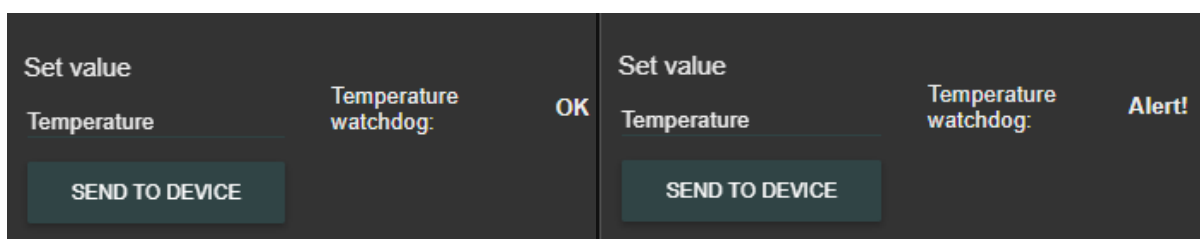


Рисунок 3.29. Стан спокою Watchdog та при перевищенні встановленої межі

Передбачена також можливість встановлення межі значень із панелі візуалізації шляхом надсилання повідомлення введеного в поле кнопкою “Send to device”.

Змодельоване вікно візуалізації роботи системи віддаленого моніторингу має наступний вигляд, Рисунок 3.30:

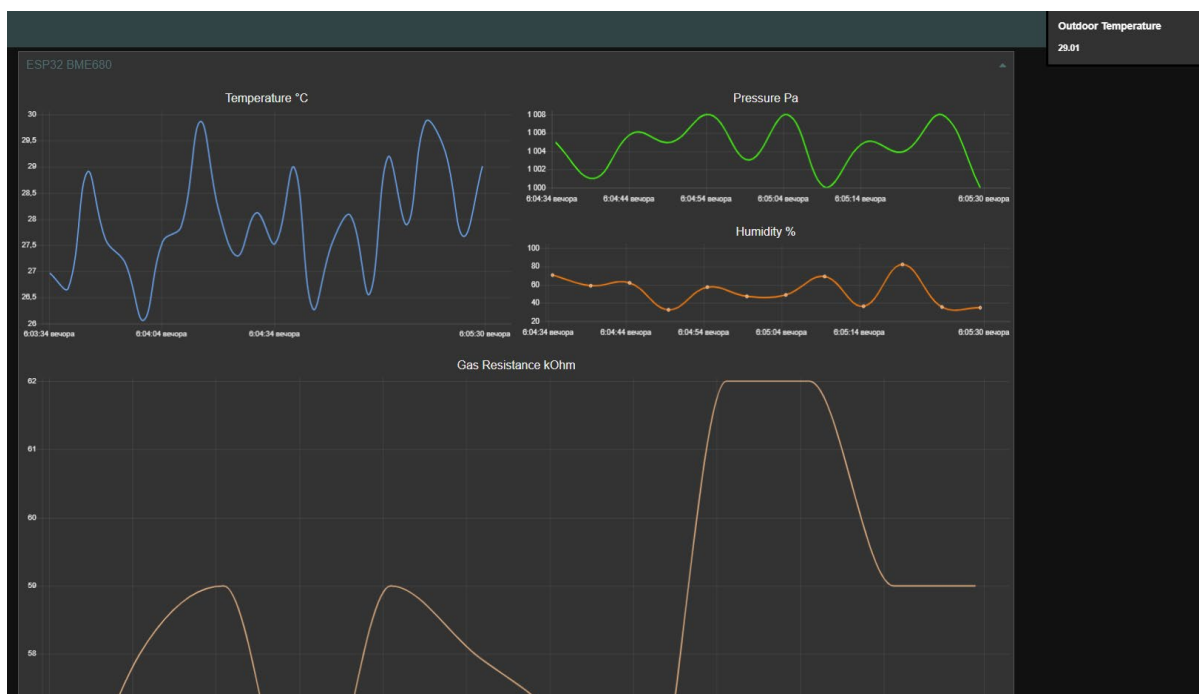


Рисунок 3.30. Загальний вигляд вікна візуалізації

Кожен параметр вимірювання у системі відображається окремим графіком.

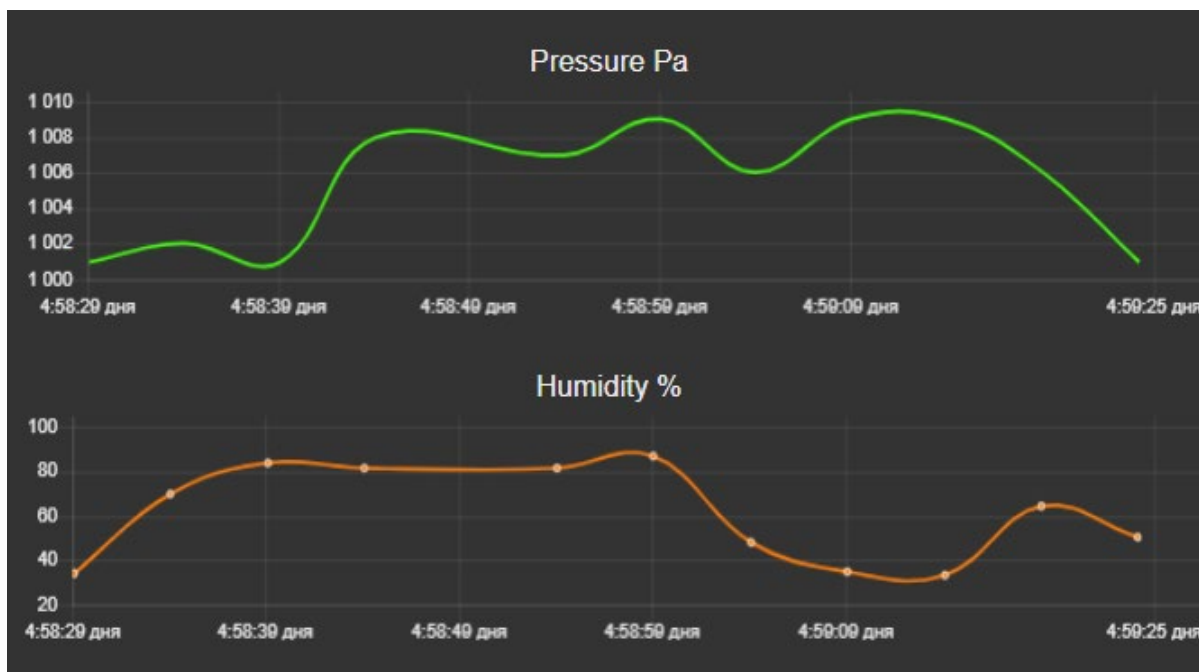


Рисунок 3.31. Графіки значень тиску та вологості повітря

При масштабованій системі моніторингу пристрої відображаються в форматі послідовного списку, коли кожен із них можна згорнути. Сповіщення

3.32:

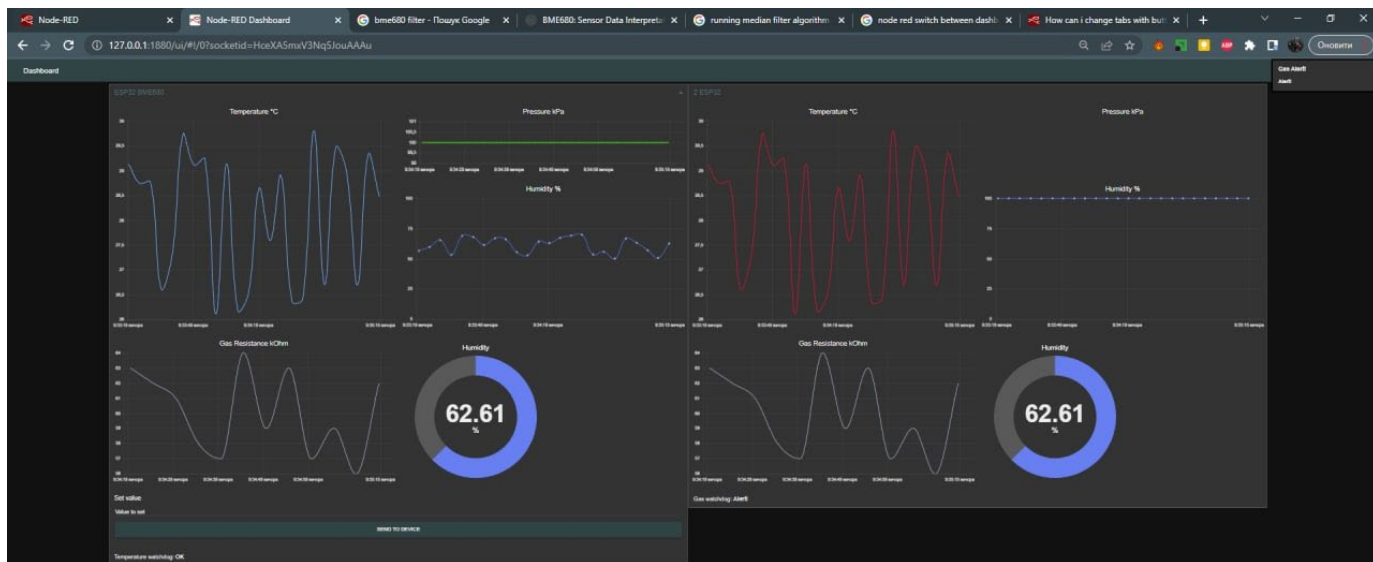


Рисунок 3.32. Панель відображення двох пристроїв системи моніторингу

4. РОЗДІЛ 4. Розробка стартап-проекту

4.1. Опис ідеї проекту

Метою цього розділу є проведення маркетингового аналізу стартап проекту для визначення принципової можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження.

Таблиця 4.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка системи комплексного моніторингу стану навколишнього середовища з дистанційною системою збору інформації	Заміна наявних систем віддаленої взаємодії із застарілими технологіями	Розширення можливостей для збору даних
	Впровадження нового комплексу для збору інформації з місцевості для оцифровування з метою безпеки передачі зашифрованої інформації	Постійний нагляд за важливим об'єктом у віддаленій місцевості, розгорнута власна безпечна закрита мережа для IoT пристроїв

Запропонований проект має схожі за спрямуванням реалізації у пропрієтарних протоколах зв'язку (SigFox, ZigBee), проте скориставшись рішенням на базі LoRaWAN досягаються оптимальні вимоги щодо безпеки і надійності мережі, а також витрат на її подальше масштабування та обслуговування.

Таблиця 4.2. Визначення сильних, слабких, та нейтральних характеристик проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W	N	S
		Мій проект	Sigfox	NB-IoT			
1	Вартість розробки	4	2	3			+
2	Безпека	5	4	4		+	
3	Покриття мережею	4	5	3	+		
4	Автономність	5	4	4			+

4.2. Технологічний аудит ідеї проекту

Таблиця 4.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології і реалізації	Наявність технологій	Доступність технологій
1	Комплекс моніторингу навколишнього середовища із дистанційною системою збору інформації	LoRaWAN	Наявна	Доступна
		SigFox	Наявна	Доступна за додаткову плату
		RISC-V MCU	Наявна	Доступна
		Живлення на сонячній батареї	Наявна	Доступна за додаткову плату
Обрана ідея реалізації проекту: в якості системи для віддаленої передачі інформації про стан місцевості протоколом LoRa.				

4.3. Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 4.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3 (Sigfox, NB-IoT, Jooby)
2	Загальний обсяг продаж, грн/за 1 рік (6 систем)	210 000 \$
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Необхідність узгодження встановлення обладнання
5	Специфічні вимоги до стандартизації та сертифікації	ДСТУ 2709-94 ДСТУ 4134-2002 ДСТУ EN 60950-22:2017
6	Середня норма рентабельності в галузі (або по ринку), % (показники взяті з офіційного джерела “Головне управління статистики України”)	13

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	1.Безпека та надійність 2.Масштабованість системи 3.Швидкість передачі даних	Організації окремих територіальних громад, міст та приватні підприємства	-конкуренція -простота входу -розмір прибутку	-Захищений обмін даних - Масштабованість системи -Швидкість обробки та передачі

Таблиця 4.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Вихід на ринок	Поріг входу достатньо високий з питань забезпечення безпеки та надійності, відповідність стандартам, проходження випробувань у спеціальних комітетах, які дають дозвіл на використання системи	Попередня перевірка відповідності стандартам забезпечення безпеки та надійності, підготовка до випробувань
2	Зниження рівня інноваційності продукту	Регулярне оновлення переліку інноваційних систем від конкурентів на ринку	Постійна модернізація та пошук рішень оптимізації для вже створеного продукту обраної технології та пошук вдосконалених альтернатив

Таблиця 4.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Скорочення об'ємів випуску нової продукції на ринку від конкурентів	Охоплення частини сегменту	Посилена робота з клієнтами з питань придбання продукції, оновлення системи та використаних технологій.
2	Співпраця з партнерами	Пришвидшення вирішення питань розробки та спрощення доступу до ресурсів	Поглиблена взаємодія з партнерами, обмін рішеннями та використання доступних ресурсів для вдосконалення
3	Зростання кількості кваліфікованих спеціалістів	Залучення кваліфікованих фахівців до розробки та модернізації системи	Розширення штату. Створення окремих підрозділів поглибленого спрямування

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Глобальний рівень	Міжнародна конференція, презентація	Просування компанії на міжнародних ринках та пошук нових партнерів для закріплення
Внутрішньогалузева	Конкуренція всередині сфери поставки товару	Орієнтація на потреби клієнта та врахування фідбеків від існуючих
Товарно-видова	Конкуренція між системами однакового призначення	Підвищення точності контролю, безпеки та надійності, простота в використанні
Цінова	Вплив ціни системи на вибір споживачів	Підтримка оптимальної ціни та зміна власної спираючись на ринкову ситуацію
Брендова	Бренд компанії	Набуття популярності бренду на ринку

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	1.Sigfox 2.NB-IoT 3.Jooby	Lifecell, Київстар, Vodafone	Espressif, Semtech, Tectelic	Організації окремих територіальних громад, міст та приватні підприємства	Готова продукція NB-IoT, SigFox, Winext
Висновки	Боротьба за ринок споживачів створюючи більш якісний продукт, який краще задовольняє потреби покупців	Інноваційний продукт під потреби клієнтів та значні зусилля на маркетинг	Зменшення унікальності постачальників шляхом підвищення якості до поставок, зменшення логістичних витрат	Більш точна система з дотримання усіх вимог безпеки	Ціна власного продукту нижча за аналоги, власне український продукт

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Підтримка клієнтів	Супровід від етапу проектування та після успішного запуску системи, обслуговування архітектури
2	Вимоги клієнтів	Вимоги клієнтів впливають на розробку та модернізацію продукту, а в кінцевому результаті на дохід
3	Ефективність	Ефективна система для кінцевого споживача
4	Маркетингова політика	Якісний маркетинг чітко сфокусований на цільову групу споживачів
5	Ціна продукту	Встановлення доступної ціни, яка конкуруватиме з присутніми на ринку аналогами, покриваючи більший функціонал

Таблиця 4.11. SWOT - аналіз стартап-проекту

Сильні сторони: ● Кінцевий продукт відповідає вимогам клієнтів ● Підтримка клієнтів на всіх етапах імплементації продукту ● Швидке реагування на зауваження та зворотній зв'язок від клієнтів	Слабкі сторони: ● Відсутність репутації на ринку ● Малий масштаб компанії ● Відсутність постійних клієнтів ● Відсутність перевірених постачальників ● Відсутність досвіду взаємодії підрозділів компанії
Можливості: ● Конкурентна ціна ● Інноваційність ● Безпека та надійність системи	Загрози: ● Нішеве середовище із відносно вузьким пулом потенційних клієнтів ● Неготовність клієнтів купувати дорогі дороговартісне обладнання

Таблиця 4.12. Альтернативи ринкового впровадження стартап-проекту

Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
● Міжнародні та Державні конференції ● Форуми ● Публікації ● Виставки ● Презентації	70%	5 місяців

4.4. Розроблення ринкової стратегії проекту

Таблиця 4.13. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачі в сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
	Організації окремих територіальних громад, міст та приватні підприємства	Висока	50%	Висока	Важко
Які цільові групи обрано: обрано компанії як державного, так і приватного сектору спрямування					

Таблиця 4.14. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Точність вимірювань	Конкурентне позиціонування	Висока точність вимірювань технологічних параметрів	● Підтримка широкого діапазону вимірювання ● Мінімальна похибка вимірювань ● Стійкість до зовнішніх впливів
2	Безпека та надійність	Конкурентне позиціонування	Реалізовані системи, які моніторять процеси	● Система спостереження ● Система сповіщення

Продовження Таблиці 4.14. Визначення стратегії позиціонування

3	Масштабованість системи	Конкурентне позиціонування	Масштабованість система згідно з потребами споживачів	● Значне масштабування без впливу на якість роботи системи ● Відмовостійкість застосовуваного обладнання ● Швидке відновлення як компонентів так і системи в цілому
4	Швидкість обробки та передачі показників вимірюваних параметрів	Конкурентне позиціонування	Швидка обробка значень параметрів та передача їх на сервер додатків кінцевому користувачеві	● висока відстань передачі даних між рівнями системи ● енергоефективність кінцевого пристрою ● швидка передача актуальних даних до серверу додатків операторові

4.5. Розроблення маркетингової програми стартап-проекту

Таблиця 4.15. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Велика дистанція передачі даних	Достатня для встановлення у міських та міжміських умовах відстань на передачу, що дозволяє ефективно розширювати мережу	Оптимальна відстань для застосування у спостереженні на місцевості
2	Безпека та надійність роботи системи	Закрита система із двостороннім шифруванням, надійне обладнання світових виробників	Контроль безпеки та надійності роботи системи

Продовження Таблиці 4.15. Визначення ключових переваг концепції
потенційного товару

3	Автономність	Мінімально можливе для схожих технологій передачі енергоспоживання в парі із розумними алгоритмами сну контролера граничного пристрою	Час роботи пристроїв з акумуляторними джерелами живлення
---	--------------	---	--

Таблиця 4.16. Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
\$35000	\$ 40 000 - 50 000	-	\$ 20 000 - 50 000

Оскільки продукт спрямований не стільки на підвищення доходів, як на покращення комфорту та безпеки для клієнта, то враховувати рівень доходів при визначенні меж встановлення ціни може бути не доречним.

Таблиця 4.17. Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Клієнти купують продукт у дилера або виробника	- налагодження контактів з покупцями - просування товару до споживача - комплектація товару згідно вимог споживача	Канал нульового рівня (виробник безпосередньо продає товар клієнту)	Безпосередня презентація комплексу моніторингу стану навколишнього середовища

Таблиця 4.18. Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Клієнти отримують інформацію про новий продукт на: конференціях, форумах, виставках, презентаціях та на офіційному сайті	-Міжнародні та Державні конференції -Форуми -Виставки -Презентації -Інтернет	Вища точність, керування безпечніше та надійніше, швидше обчислення та обробка сигналів на вході та виході	Зацікавити потенційних покупців, висвітлити переваги продукту на ринку.	Інноваційний підхід у керуванні технологічним процесом, безпека та надійність на вищому рівні

4.6. Висновки

За задумом проект забезпечує моніторинг та інформування оператора про аварійні стани об'єкта, своєчасне відпрацювання систем оповіщення. До продажу клієнти мають змогу ознайомитися з тестовим демонстраційним екземпляром та програмним забезпеченням у симуляторі з прихованою внутрішньою реалізацією. За рахунок патенту та авторських прав товар буде захищено від копіювання.

Проаналізувавши рівень цін на товари-замінники запропонованої комплексної системи моніторингу(близько \$35000), товари-аналоги (близько \$40 000 - 50 000\$) та рівень доходів цільової групи споживачів (близько \$160 000), було встановлено нижню \$20 000 та верхню \$50 000 межі встановлення ціни на товар, що дає перевагу над конкурентами.

ВИСНОВКИ

В магістерській дисертації була здійснена розробка комплексного моніторингу стану навколишнього середовища із дистанційною системою збору інформації.

Система розроблена для того, щоб здійснювати спостереження за віддаленими об'єктами на місцевості в умовах міста, а також і міжміській та польовій місцевостях, збирати дані про стан параметрів на об'єкті, забезпечувати надійну передачу та збереження даних, а також вчасно інформувати користувача про перебіг подій та відхилення ключових параметрів від номіналу.

Основна мета роботи – це створення системи для відстежування та передачі параметрів довкілля.

Пропонована система перш за все повинна реалізовувати інформаційні функції: функцію передачі значень параметрів, сповіщення про стан обладнання та можливого виходу його з ладу (неможливість опитування давачів). Передача даних та їх збереження повинні мати достатній захист. Разом із цим необхідно приділити увагу надійному збереженню отримуваних даних аби можна було спиратися на історичні дані в перспективі.

Система, що відповідає таким вимогам, може замінити застарілі технології, що використовуються зараз для виконання моніторингу довкілля, спростивши роботу людям та заощаджуючи бюджет в довготерміновій перспективі. Також важливий акцент ставиться на можливості зручного розширення комплексу із додаванням пристроїв у вже налаштовану безпечну мережу.

Розроблена система сповна забезпечує виконання висунутих до неї вимог.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. ДСТУ ISO 8756:2008 Якість повітря. Оброблення даних за температурою, тиском та відносною вологістю (ISO 8756:1994, IDT).
2. «Sigfox. LoRa. NB-IoT. Easy guide to who does it best» 2018,
[Електронний ресурс] - <https://www.iotsolutions.com.mt/post/sigfox-vs-lora-vs-nb-iot-who-s-doing-it-best>
3. Technical Overview of LoRa and LoRaWAN, TÜV Rheinland, 2015.
4. Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things, IEEE Communications Magazine 2018.
5. Pradeeka Seneviratne “Beginning LoRa Radio Networks with Arduino: Build Long Range, Low Power Wireless IoT Networks” Apress, 2019. С. 320.
6. Semtech “LoRa and LoRaWAN: A Technical Overview” Technical Paper, 2020, Режим доступу: https://lora-developers.semtech.com/uploads/documents/files/LoRa_and_LoRaWAN-A_Tech_Overview-Downloadable.pdf
7. Dogan Ibrahim “Programming with Node-RED. Design IoT Projects with Raspberry Pi, Arduino and ESP32” Elektor International Media, 2020
8. Neil Kolban “Kolban's Book on ESP32” 2018



ЕНЕРГОЗБЕРІГАЮЧІ СИСТЕМИ
АЛЬТЕРНАТИВНІ ДЖЕРЕЛА ЕНЕРГІЇ

ОФІЦІЙНИЙ ПРЕДСТАВНИК ТМ SOLARX В УКРАЇНІ
УЧАСНИК ПРОГРАМИ ЄБРР IQ ENERGY

ТОВАРИСТВО З ОБМЕЖЕНОЮ
ВІДПОВІДАЛЬНІСТЮ

«ТОРГОВИЙ ДІМ «СОЛАР ІКС»

Юридична адреса: 04073, м. Київ, вул. Ливарська, 7

Адреса для листування: 04159, м. Київ, вул. Сім'ї Кульженків 33/110

Код ЄДРПОУ 43889453

р/р UA253052990000026002036802165 в АТ КБ "ПРИВАТБАНК"

тел. (044) 223 38 55, (096) 585 99 55

Вих. №164/02 від 15.12.2022

ДОВІДКА ПРО ВПРОВАДЖЕННЯ

Надана магістерська дисертація освітнього ступеня «магістр» на тему «Комплексний моніторинг стану навколишнього середовища із дистанційною системою збору інформації» студента групи ТА-11мп Гузя Д.Р. може бути впроваджена на реальному об'єкті при розробці та пуско-налагодженні таких систем із мережею, висвітленою в роботі.

Директор ТОВ «ТД «Солар Ікс»

О.В. Іваненко

Тема: Комплексний моніторинг стану навколишнього середовища із дистанційною системою збору інформації

Виконав: магістрант групи ТА-11мп Гузь Дмитро Романович

Науковий керівник: старший викладач Штіфзон Олег Йосипович

Кількість сторінок записки: 79

Ключові слова. *LORAWAN, ESP32, МОНІТОРИНГ*

Анотація на 2-х мовах.

Магістерську дисертацію виконано на тему «Комплексний моніторинг стану навколишнього середовища із дистанційною системою збору інформації», на здобуття ступеня магістра. Пояснювальна записка складається з 4 розділів, вступу, висновку, списку літератури. Проект виконаний в обсязі 79 сторінок.

Мета та задачі дослідження. Метою магістерської дисертації є створення системи для відстежування та передачі параметрів довкілля, що дало б змогу зручного спостереження за параметрами навколишнього середовища у встановленій місцевості, оперативнішого сповіщення про аномальні події та реакцію на них. В довготерміновій перспективі це дало б покращення екологічних умов місцевості та потенційно здоров'я людей.

Об'єктом дослідження є система моніторингу стану довкілля із дистанційним збором даних.

Новизна та практичне застосування отриманих результатів. У сучасних умовах спостереження за станом навколишнього середовища відіграє значну роль, особливо на фоні кліматичних змін по всій планеті останніми роками. Беручи до уваги це, вдалим рішенням було б щонайменше вдосконалити вже наявні системи спостереження новішими технологіями із віддаленим моніторингом параметрів довкілля.

The master's thesis was completed on the topic "Complex monitoring of the state of the environment with a remote information collection system", for obtaining a master's degree. The explanatory note consists of 4 sections, an introduction, a conclusion, and a list of references. The project is completed in the volume of 79 pages.

The purpose and objectives of the research. The goal of the master's thesis is to create a system for monitoring and transmitting environmental parameters, which would allow for convenient monitoring of environmental parameters in a given area, more prompt notification of abnormal events and response to them. In the long term, this would improve the environmental conditions of the area and potentially the health of people.

The object of the study is a system for monitoring the state of the environment with remote data collection.

Novelty and practical application of the obtained results. In modern conditions, monitoring the state of the environment plays a significant role, especially against the background of climate changes across the planet in recent years. Taking this into account, a good solution would be to at least improve existing surveillance systems with newer technologies with remote monitoring of environmental parameters.

РЕЦЕНЗІЯ

на магістерську дисертацію на здобуття ступеня магістра

виконану на тему: Комплексний моніторинг стану
навколишнього середовища із дистанційною системою збору інформації
студентом Гузем Дмитром Романовичем
(прізвище, ім'я, по батькові)

Виконаний у магістерській дисертації проект відповідає затвердженій темі та завданню роботи. Створена система здатна вирішити актуальні питання порушені в роботі. Запропонована архітектура рішення може бути використана на реальному об'єкті.

Реалізоване програмне забезпечення може забезпечувати функціонал пристрою при розробці запропонованої системи, проте для встановлення такого реальній системі варто адаптувати його до повноцінних можливостей у LoRa мережі.

Продемонстровано оригінальне інженерне рішення для апаратного функціоналу кінцевого пристрою має потенціал до подальшого розвитку у розширенні можливостей застосування.

Відображене моделювання роботи системи є підтвердженням концепції, яка може розгорнутися до повноцінної системи віддаленого моніторингу.

Враховуючи все сказане, вважаю, що дисертація заслуговує оцінки «відмінно», а її автор Гузь Дмитро Романович – присвоєння 2 ступеня вищої освіти «магістр» та кваліфікації «магістр з автоматизації та комп'ютерно-інтегрованих технологій» за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології.

Рецензент Директор ТОВ «ТД «Солар Ікс»
(посада, вчені звання, ступінь)

(підпис)

О.В. Іваненко
(ініціали, прізвище)

Справ. №	
Підп. і дата	
Інв. № ауд.	
Зам. інв. №	
Підп. і дата	
Інв. № оригін	

Технологічна схема системи моніторингу

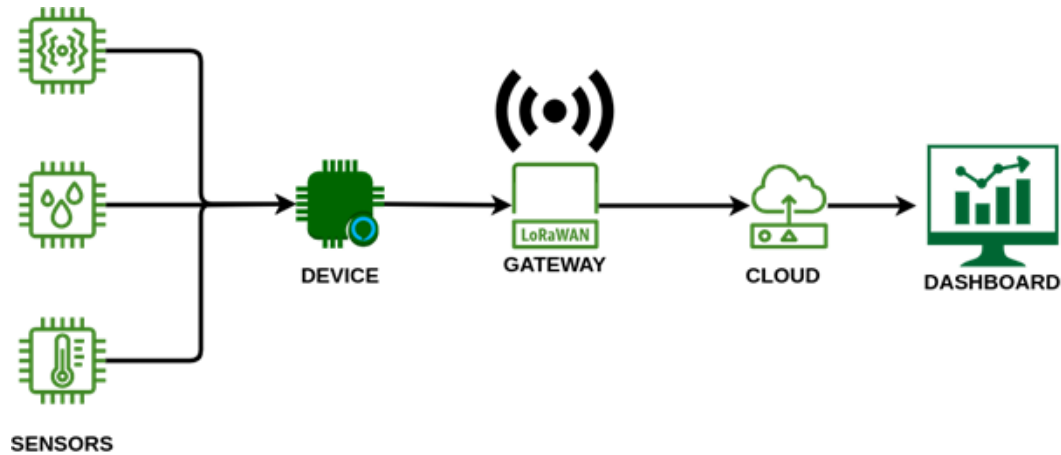
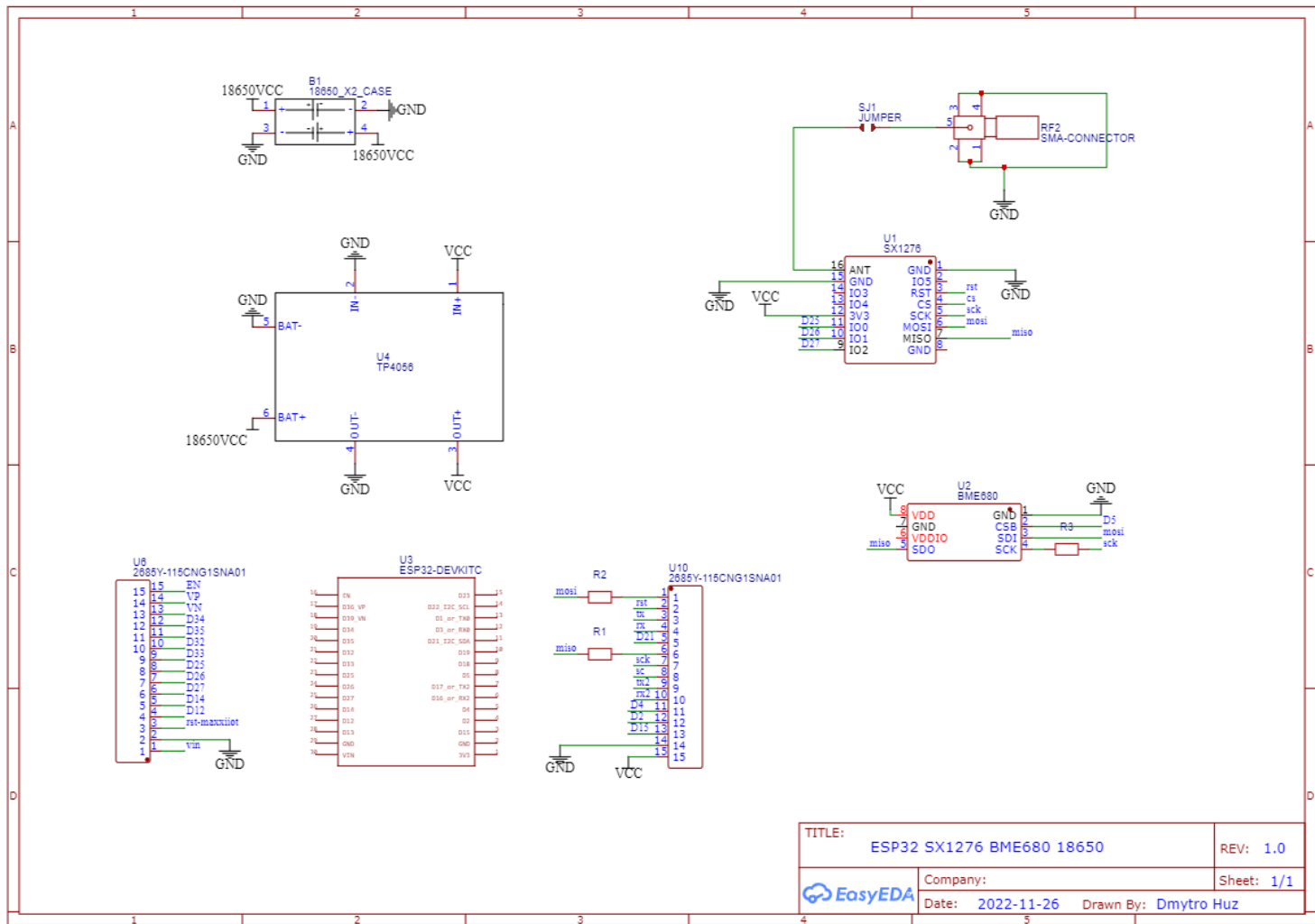
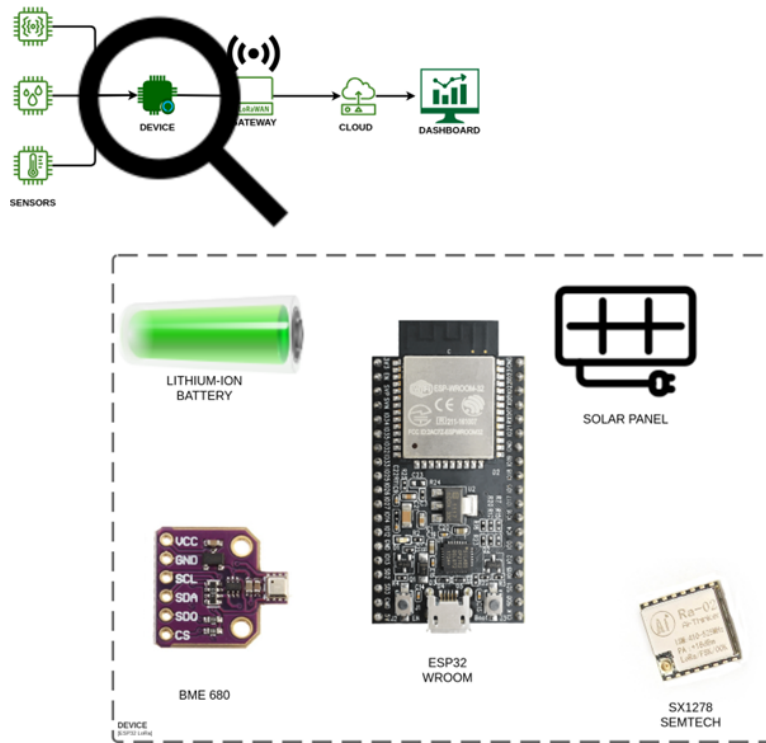


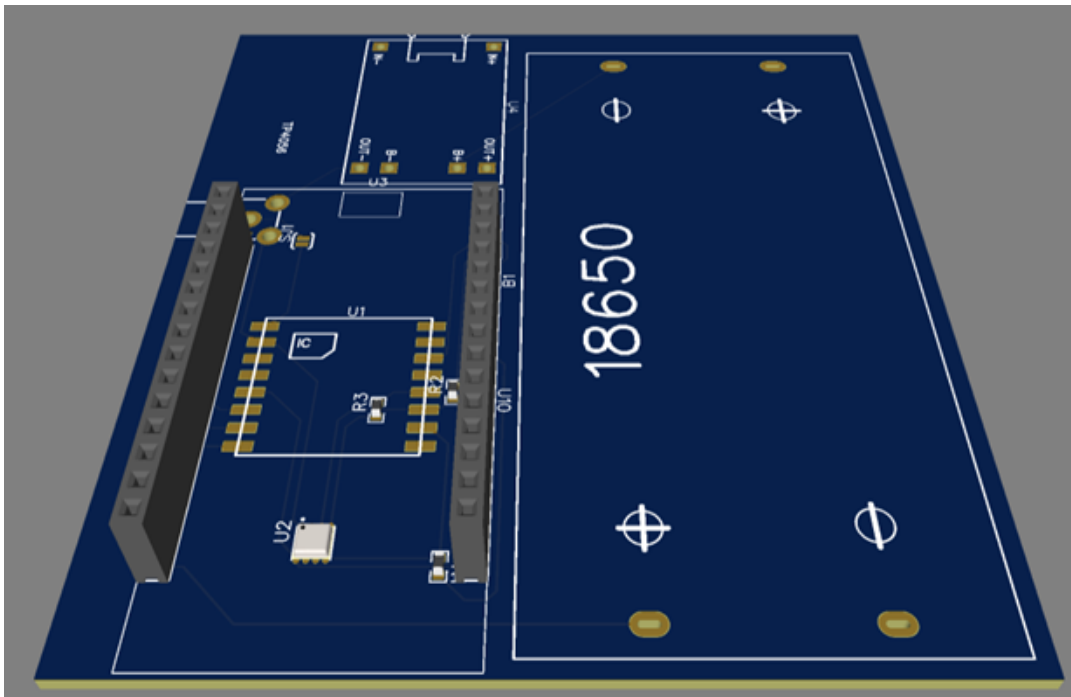
Схема підключень модулів граничного пристрою



Структурна схема граничного пристрою



Об'ємне відображення друкованої плати



Графічні ілюстрації до магістерської дисертації на тему: "Комплексний моніторинг стану навколишнього середовища із дистанційною системою збору інформації"

Виконав студент
Керівник

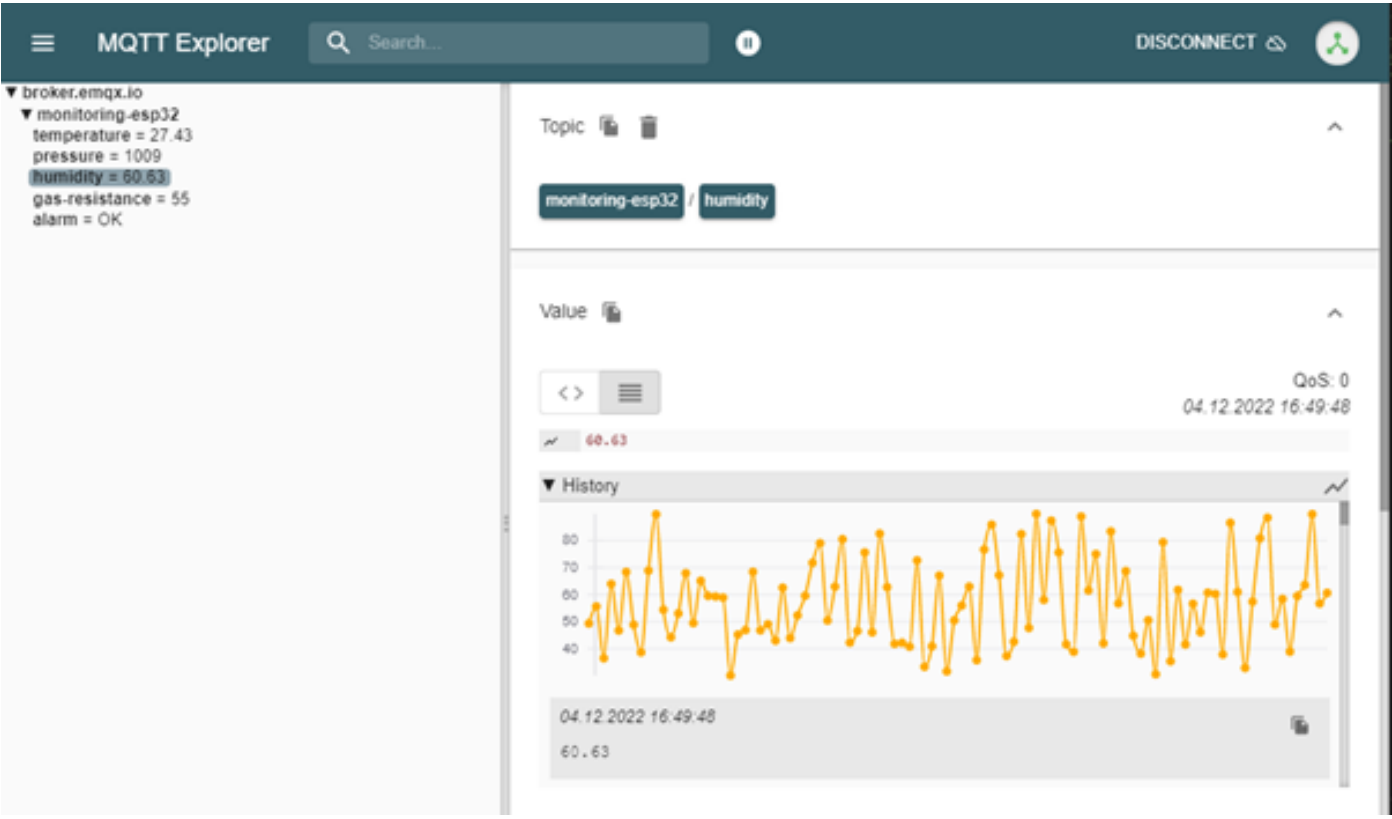
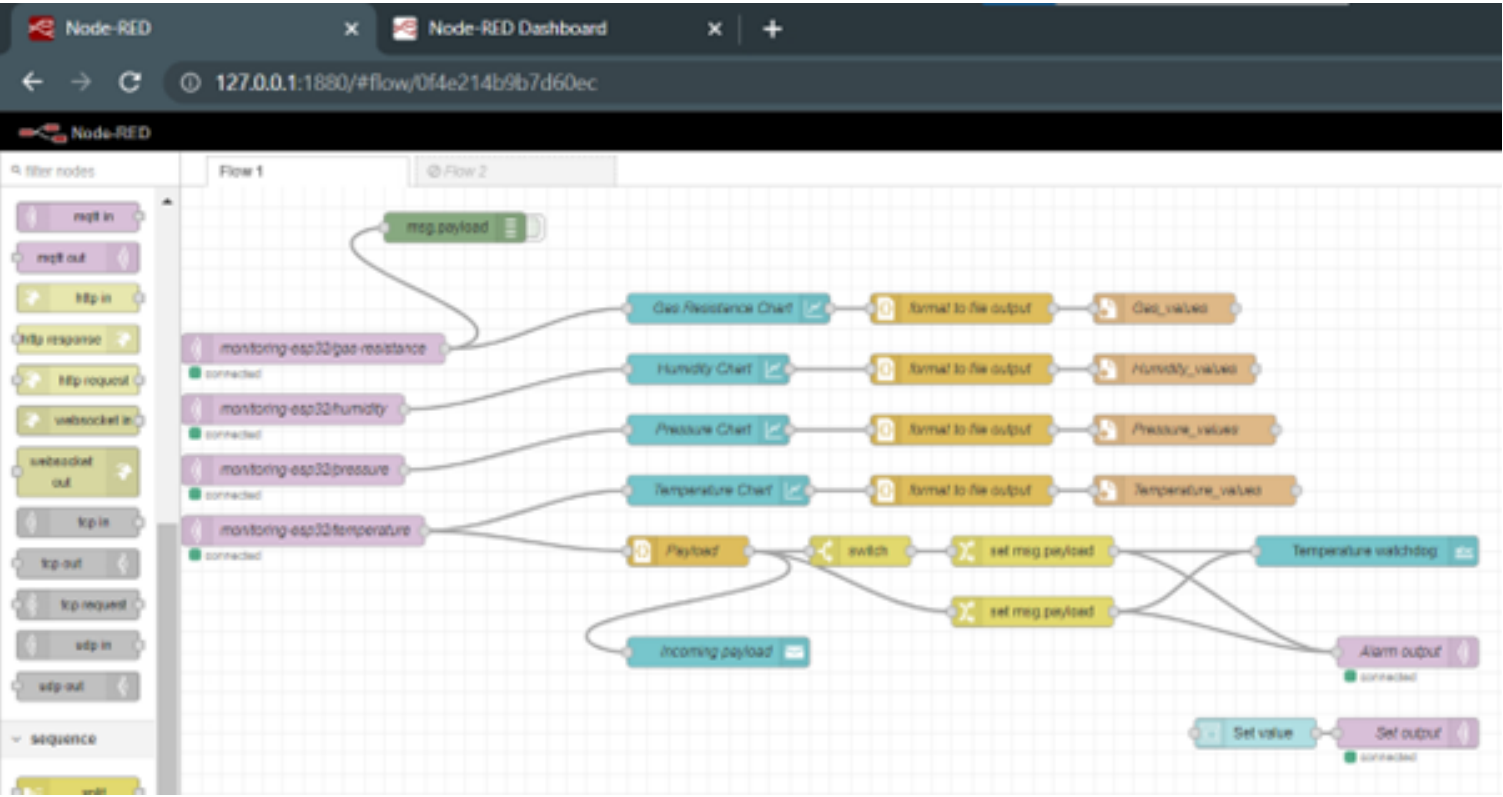
Гузь Д.Р., ІАТЕ, група ТА-11мп
Штіфзон О.Й., ст.викл.

Вікно панелі оператора



Перебіг отримання даних на MQTT брокер

Потоки для значень параметрів



Графічні ілюстрації до магістерської дисертації на тему: "Комплексний моніторинг стану навколишнього середовища із дистанційною системою збору інформації"

Виконав студент
Керівник

Гузь Д.Р., ІАТЕ, група ТА-11мп
Штіфзон О.Й., ст.викл.

Имя пользователя:
Олег Йосипович Штіфзон

ID проверки:
1013301185

Дата проверки:
14.12.2022 19:09:07 EET

Тип проверки:
Doc vs Internet + Library

Дата отчета:
14.12.2022 19:10:50 EET

ID пользователя:
94811

Название файла: Гузь Дмитро

Количество страниц: 73 Количество слов: 7832 Количество символов: 60885 Размер файла: 6.58 MB ID файла: 1013059124

Обнаружены модификации текста (могут влиять на процент совпадений)

17.4%

Совпадения

Наибольшее совпадение: 7.42% с источником из Библиотеки (ID файла: 1005707785)

6.05% Источники из Интернета

86

Страница 75

17.2% Источники из Библиотеки

760

Страница 78

0% Цитат

Исключение цитат выключено

Исключение списка библиографических ссылок выключено

0% Исключений

Нет исключенных источников

Модификации

Обнаружены модификации текста. Подробная информация доступна в онлайн-отчете.

Замененные символы

1

Подозрительное форматирование

14
страниц