МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМ. С.З. ҐЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему:

«РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДИСТАНЦІЙНОГО МОНІТОРИНГУ СТАНУ НАВКОЛИШНЬОГО СЕРЕДОВИЩА»

Виконала: здобувачка групи Іт-61з
спеціальності 126 «Інформаційні системи та технології»

_____ Брень І. А.
(прізвище та ініціали)

Керівник: _____ Пташник В. В.
(прізвище та ініціали)

ДУБЛЯНИ-2025

Created by Universal Document Converter

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАОЧНОЇ ТА ПІСЛЯДИПЛОМНОЇ ОСВІТИ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Другий (магістерський) рівень вищої освіти
Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____
(підпис)
д.т.н., професор, Тригуба А. М.
(вч. звання, прізвище, ініціали)
“____” _____202 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Брень Ірині Андріївні
(прізвище, ім’я, по батькові)

1. Тема роботи *«Розробка інформаційної системи дистанційного моніторингу стану навколишнього середовища»*
керівник роботи *к. т. н., доцент., Пташник В. В.*
(наук.ступінь, вч. звання, прізвище, ініціали)
затверджені наказом Львівського НУП *від 30.12.2024 року № 887/к-с.*

2. Строк подання студентом роботи *8 грудня 2025 року*
3. Вихідні дані до роботи: *технічні характеристики сенсорів температури, вологості, атмосферного тиску та газового складу повітря (зокрема комплексного датчика BME680), бездротових модулів зв’язку LoRa/LoRaWAN та мікроконтролерних платформ ESP32-S3. Паспорти й специфікації апаратних компонентів граничних пристроїв і шлюзів, документація на платформу ChirpStack, середовище Node-RED та MQTT-брокер, а також науково-технічна література й нормативні вимоги до систем екологічного моніторингу та обробки телеметричних даних.*
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
Вступ
1. Аналіз предметної області та сучасних рішень
2. Структурне проектування інформаційної системи дистанційного моніторингу
3. Імплементація системи дистанційного моніторингу
4. Охорона праці та безпека в надзвичайних ситуаціях
5. Економічна ефективність
Висновки
Список використаних джерел

Created by Universal Document Converter

5. Перелік графічного матеріалу

Графічний матеріал подається у вигляді презентації

6. Консультанти розділів

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата		Відмітка про виконання
		завдання видав	завдання прийняв	
1, 2, 3, 5	*Пташник В. В., к.т.н., доцент*			
4	*Городецький І. М., к.т.н., доцент*			

7. Дата видачі завдання *31 грудня 2024 р.*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Відмітка про виконання
1	*Складання інженерної характеристики систем моніторингу стану навколишнього середовища та IoT-сенсорних платформ*	*01.01.2025 – 31.03.2025*	
2	*Вибір принципів, методів та засобів розробки інформаційної системи дистанційного моніторингу навколишнього середовища*	*01.04.2025 – 31.05.2025*	
3	*Оптимізація інформаційного обміну та обробки телеметричних даних в IoT-системі екологічного моніторингу*	*01.06.2025 – 30.09.2025*	
4	*Розгляд питань з охорони праці та безпеки у надзвичайних ситуаціях*	*01.10.2025 – 20.10.2025*	
5	*Оцінка економічної ефективності прийнятих рішень*	*21.10.2025 – 15.11.2025*	
6	*Завершення оформлення розрахунково-пояснювальної записки та презентаційного матеріалу*	*16.11.2025 – 30.11.2025*	
7	*Завершення роботи в цілому. Підготовка до захисту кваліфікаційної роботи*	*01.06.2025 – 08.12.2025*	

Здобувач _____ *Брень І. А.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Пташник В. В.*
(підпис) (прізвище та ініціали)

Created by Universal Document Converter

УДК 681.521 / 681.518

Розробка інформаційної системи дистанційного моніторингу стану навколишнього середовища. Брень І. А. Кафедра інформаційних технологій – Дубляни, Львівський НУВМБ ім. С.З. Ґжицького, 2025.

Кваліфікаційна робота: 72 сторінки текстової частини, 33 рисунки, 6 таблиць, 18 джерел літератури.

Мета кваліфікаційної роботи полягає в аналізі та розробленні інформаційної системи дистанційного моніторингу параметрів навколишнього середовища на основі розподіленої IoT-сенсорної мережі з використанням технології LoRaWAN, із забезпеченням надійного передавання телеметричних даних, їх обробки та візуалізації у режимі, наближеному до реального часу.

Об'єктом дослідження є процеси збору, передавання та моніторингу параметрів навколишнього середовища в розподілених IoT-системах на базі бездротових сенсорних мереж.

Предмет дослідження становлять методи й засоби проєктування інформаційної системи екологічного моніторингу на базі LoRaWAN, включно з архітектурою сенсорних вузлів і шлюзів, механізмами інтеграції з мережевим сервером, передаванням даних за MQTT та організацією їх подальшої обробки і візуалізації.

У кваліфікаційній роботі проаналізовано підходи до побудови IoT-систем екологічного моніторингу; обґрунтовано вибір LoRaWAN як технології зв'язку; спроєктовано структуру сенсорних вузлів і мережевої частини з використанням платформи керування мережею; реалізовано збір телеметрії та її подання у вебінтерфейсі; оцінено працездатність запропонованого рішення та його енергоефективність..

Ключові слова: інформаційна система, Інтернет речей (IoT), LoRaWAN, сенсорна мережа, екологічний моніторинг, телеметричні дані, ESP32, BME680, ChirpStack, Node-RED, MQTT, візуалізація даних, енергоефективність.

Created by Universal Document Converter

ЗМІСТ

Created by Universal Document Converter

Created by Universal Document Converter

ВСТУП

Технологічний прогрес відбувається з безпрецедентною швидкістю, проникаючи в кожен аспект сучасного життя. Четверта промислова революція суттєво посилила цей розвиток, а її ключові характеристики включають:

- інтерактивність коли різні системи працюють спільно та узгоджено;
- віртуалізацію через створення штучних об'єктів та середовищ;
- децентралізацію планування та виконання процесів;
- виконання процесів у режимі реального часу;
- сервісну орієнтованість де користувач може впливати на кінцевий продукт;
- модульність системи за рахунок використання незалежних модулів, що дозволяє швидко та легко замінювати їх без впливу на інші компоненти системи.

Інтернет речей (IoT) є важливим технологічним компонентом Четвертої промислової революції, що втілює більшість її характеристик і значно розширює та спрощує зв'язок між людьми, технологіями та навколишнім світом.

Саме Інтернет робить можливими технології на основі IoT. Інтернет речей (IoT) використовує мережеву інфраструктуру для з'єднання різних незалежних компонентів у єдину систему, забезпечуючи вільний обмін даними між ними. Інтернет речей може як спрощувати, так і оптимізувати щоденну діяльність людей, а також підтримувати налаштування й реалізацію масштабних процесів виробництва, моніторингу та аналізу даних.

У контексті IoT особливий інтерес становлять системи моніторингу навколишнього середовища в режимі реального часу. Такі системи відіграють вирішальну роль у спостереженні та прогнозуванні стану навколишнього середовища, а також у запобіганні й реагуванні на небезпечні ситуації. Вони допомагають оптимізувати розподіл ресурсів, економити енергію та підвищувати продуктивність. Завдяки IoT ми можемо отримувати детальні дані

Created by Universal Document Converter

про стан навколишнього середовища, що розширює наші можливості щодо впливу на його якість і стабільність.

З огляду на такі глобальні виклики, як зміна клімату, забруднення навколишнього середовища та виснаження природних ресурсів, важливість моніторингу довкілля стає дедалі очевиднішою. Здатність відстежувати та аналізувати ці процеси в режимі реального часу має вирішальне значення для нашого виживання та сталого розвитку. Для нашої країни головним викликом є повоєнна відбудова та перетворення України на сучасну, розвинену європейську державу. Наше покоління стикається із завданнями відбудови та будівництва нових міст, цивільної та стратегічної інфраструктури. Для майбутнього нації важливо розвиватися відповідно до новітніх тенденцій Інтернету речей (IoT), зокрема через застосування технологій розумного міста. Системи IoT, здатні збирати, обробляти та аналізувати дані про стан навколишнього середовища, відіграють життєво важливу роль у процесі модернізації.

Метою роботи є створення системи збору, обробки та аналізу даних про навколишнє середовище, що дасть змогу своєчасно реагувати на їх зміни та вживати заходів. У межах дослідження розробляється концепція системи, визначаються технології та програмні засоби, необхідні для її впровадження, а також аналізуються проблеми й перешкоди, які можуть виникнути під час реалізації такої системи, що передбачає виконання наступних завдань:

- проаналізувати існуючі програмно-апаратні рішення віддаленого моніторингу навколишнього середовища;

- обґрунтувати вибір архітектури інформаційної системи моніторингу, апаратної платформи сенсорних вузлів (мікроконтролер, лічильники, датчики) та бездротової технології передавання даних (LoRaWAN);

- розробити структурну й інформаційну модель системи: схему взаємодії вузлів, модель бази даних, потоків телеметрії та аналітичних показників для користувачів;

- створити програмне забезпечення граничних вузлів і серверної частини системи, а також інтерфейс візуалізації даних та звітності для користувача.

Created by Universal Document Converter

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СУЧАСНИХ РІШЕНЬ

1.1 Характеристика об’єкта дослідження та його призначення

Моніторинг навколишнього середовища є важливим компонентом багатьох сфер діяльності, зокрема екології, сільського господарства та метеорології. Він дає змогу виявляти різноманітні зміни у стані довкілля та своєчасно на них реагувати.

Основні завдання моніторингу навколишнього середовища включають:

- відстеження змін стану навколишнього середовища;
- моделювання та прогнозування впливу діяльності людини на довкілля;
- об’єктивну оцінку стану навколишнього середовища;
- прогнозування можливих змін у навколишньому середовищі.

Основні принципи, яких необхідно дотримуватися під час побудови та експлуатації систем моніторингу навколишнього середовища це об’єктивність, надійність, систематичність, програмно-апаратна узгодженість та комплексність оцінки.

Таким чином будь-яка система, що збирає та обробляє дані, повинна бути перевіреною, стабільною та надійною. Похибки та інші фактори впливу необхідно враховувати під час аналізу даних. Оскільки збір даних, їх подальша обробка та аналіз є частинами єдиного процесу моніторингу, система повинна працювати послідовно й безперервно. Це дасть змогу отримувати більш об’єктивну та повну інформацію про стан навколишнього середовища. Для забезпечення коректної роботи системи взаємодія між її різними компонентами має бути правильно налаштована. Важливе значення відіграють як програмні так і апаратні засоби, а однією з основних задач розробника є їх узгодження. Також

Created by Universal Document Converter

очевидно, що оцінювання стану об'єкта моніторингу повинно враховувати всі його характеристики та особливості, зовнішні впливи, можливі похибки вимірювань тощо.

Моніторинг навколишнього середовища можна поділити на дві основні категорії: дистанційний та наземний моніторинг [1]. Дистанційний моніторинг навколишнього середовища передбачає збір даних про стан довкілля за допомогою супутникових знімків, аерофотозйомки, радарів та інших подібних технологій. Цей підхід потребує спеціалізованого програмного забезпечення для обробки даних з метою отримання інформації про стан навколишнього середовища на великих територіях або у важкодоступних регіонах. Дистанційний моніторинг особливо важливий для дослідження змін клімату, вирубки лісів, забруднення водних ресурсів і моніторингу стихійних лих.

Наземний моніторинг передбачає безпосередній збір даних на місці за допомогою таких інструментів, як метеорологічні станції, буї, зонди, проби води та ґрунту. Він може бути дуже точним і детальним та дає змогу отримувати дані про конкретні локації й параметри навколишнього середовища. Наземний моніторинг може використовуватися для збору даних про якість повітря й води, стан ґрунту, біорізноманіття тощо. Варто зазначити, що ці два методи моніторингу зазвичай застосовують у поєднанні для отримання найповнішої та найточнішої інформації про стан навколишнього середовища.

Структура системи моніторингу змін навколишнього середовища включає такі складові: «спостереження», «оцінка фактичного стану», «прогноз стану», «оцінка прогнозованого стану» та «регулювання якості системи» [2]. Ці елементи взаємопов'язані петлями прямого та зворотного зв'язку (рис. 1.1). Враховуючи завдання, принципи та види моніторингу довкілля, очевидно, що ефективність таких систем залежить не лише від правильного вибору параметрів спостереження, але й від здатності своєчасно збирати, передавати, зберігати та аналізувати дані. Структурні компоненти моніторингу – «спостереження», «оцінка фактичного стану», «прогнозування стану», «оцінка прогнозованого стану» та «регулювання якості системи» – вимагають надійної інформаційної

Created by Universal Document Converter

інфраструктури для забезпечення безперервного обміну даними між датчиками, обчислювальними ресурсами та користувачами.

Як зазначалося вище, узгодженість апаратного та програмного забезпечення є ключовою вимогою до архітектури системи, вибору протоколу зв'язку та програмної платформи для обробки вимірювань. У зв'язку з цим постає потреба в подальшому дослідженні інформаційних і програмних інструментів для практичної реалізації вищезгаданих концепцій моніторингу.

Розглянемо етапи побудови багаторівневої системи Інтернету речей, що включає периферійні пристрої з датчиками, шлюзи обробки даних, вебсервери та сервери застосунків, а також вибір технологій бездротового зв'язку (зокрема LoRaWAN та альтернативних рішень NB-IoT, Sigfox і Zigbee). Попередній аналіз архітектури та протоколів вказує на можливість інтеграції апаратних компонентів, мережевих сервісів та програмного забезпечення в єдину систему моніторингу, що відповідає вимогам надійності, систематичності та комплексності, необхідним для оцінювання впливу на навколишнє середовище.

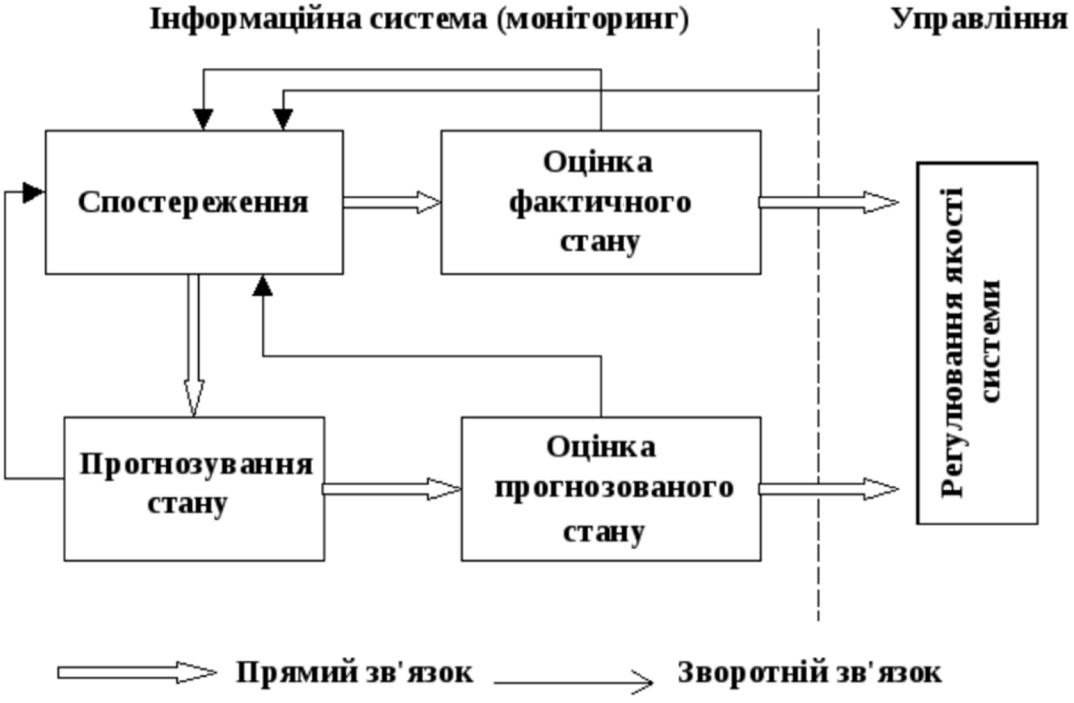

Рисунок 1.1 – Функціональна схема взаємодії модулів інформаційної системи моніторингу навколишнього середовища

Created by Universal Document Converter

Застосування Інтернету речей (IoT) у цьому процесі надає суттєві переваги. Датчики IoT можуть працювати безперервно, збираючи інформацію з високою точністю та передаючи її в режимі реального часу. Вони здатні вимірювати різноманітні показники, зокрема температуру, вологість, концентрацію вуглекислого газу, рівень шуму, якість води тощо. Крім того технології IoT дають змогу автоматизувати процеси збору даних, зменшуючи потребу в ручних операціях. Це не лише знижує витрати на оплату праці, а й імовірність помилок. IoT забезпечує моніторинг та збір даних у режимі реального часу, що дозволяє швидко реагувати на різноманітні зміни в навколишньому середовищі. Водночас датчики IoT генерують значні обсяги даних, аналіз яких дає змогу отримати глибокі та цінні висновки про стан довкілля. Це допомагає приймати обґрунтовані рішення щодо захисту та управління навколишнім середовищем. Використовуючи аналітику великих даних і методи штучного інтелекту, дані з IoT-датчиків можна застосовувати для прогнозування та запобігання небезпечним ситуаціям. Відкритість (доступність) зібраних даних сприяє прозорості дій зацікавлених сторін у сфері захисту довкілля. Вони також можуть слугувати важливим освітнім інструментом для підвищення обізнаності населення щодо екологічних проблем.

Хоча технологія Інтернету речей пропонує численні переваги для моніторингу довкілля, її застосування також має низку помітних недоліків, які слід мінімізувати. Перш за все встановлення та обслуговування IoT-датчиків може бути дорогим. Витрати включають придбання обладнання, монтаж, технічну підтримку та заміну пристроїв. Зберігання й обробка великих масивів даних, а також розроблення програмного забезпечення також потребують значних ресурсів. Крім того датчики Інтернету речей можуть працювати некоректно, давати збої або виходити з ладу через вплив навколишнього середовища. Це може призводити до втрати даних або спотвореного відображення реальних умов. Датчики IoT та системи передавання даних є потенційно вразливими до несанкціонованого доступу чи кібератак. Це може спричинити порушення конфіденційності, втрату або фальсифікацію даних. І ще

Created by Universal Document Converter

одним потенційним недоліком технології може стати енергетична залежність системи, адже датчики Інтернету речей потребують енергопостачання для роботи, що може бути значною проблемою у віддалених або важкодоступних районах.

1.2 Структура та ключові компоненти системи моніторингу

За результатами попереднього аналізу можна зробити попередній висновок, що система моніторингу стану навколишнього середовища повинна складатись з таких компонентів:

– периферійне обладнання для збору інформації з датчиків, встановлених на об’єкті;

– шлюз обробки даних, що забезпечує маршрутизацію пакетів даних від сервера до периферійних пристроїв і навпаки;

– серверна інфраструктура, зокрема мережевий сервер обробляє дані, що надсилаються від базової станції до сервера застосунків, а сервер застосунків реалізує функції взаємодії з пристроями в мережі, отримує дані від мережевого сервера та відображає їх графічно для візуалізації.

На рисунку 1.2 подано схему запропонованої системи, що включає усі вищеперелічені компоненти. На нижньому рівні функціонує мікроконтролер, оснащений датчиками, який збирає інформацію та передає її на верхній рівень.

Традиційна топологія мережі LoRa – це «зірка», у якій кілька кінцевих пристроїв зв’язуються зі шлюзом, а той, у свою чергу, – з мережевим сервером, як показано на рисунку 1.3.

У промислових застосуваннях широко використовується кілька шлюзів обробки даних для охоплення більшої зони покриття сигналом. Теоретично навіть на відносно невеликій площі пропускної здатності одного шлюзу може

Created by Universal Document Converter

бути недостатньо (зазвичай до мережі може одночасно підключатися до 10 000 пристроїв), що зумовлює необхідність збільшення кількості шлюзів у системі. Такий підхід розширює комунікаційні можливості, але водночас підвищує вартість системи через потребу в додаткових шлюзах.

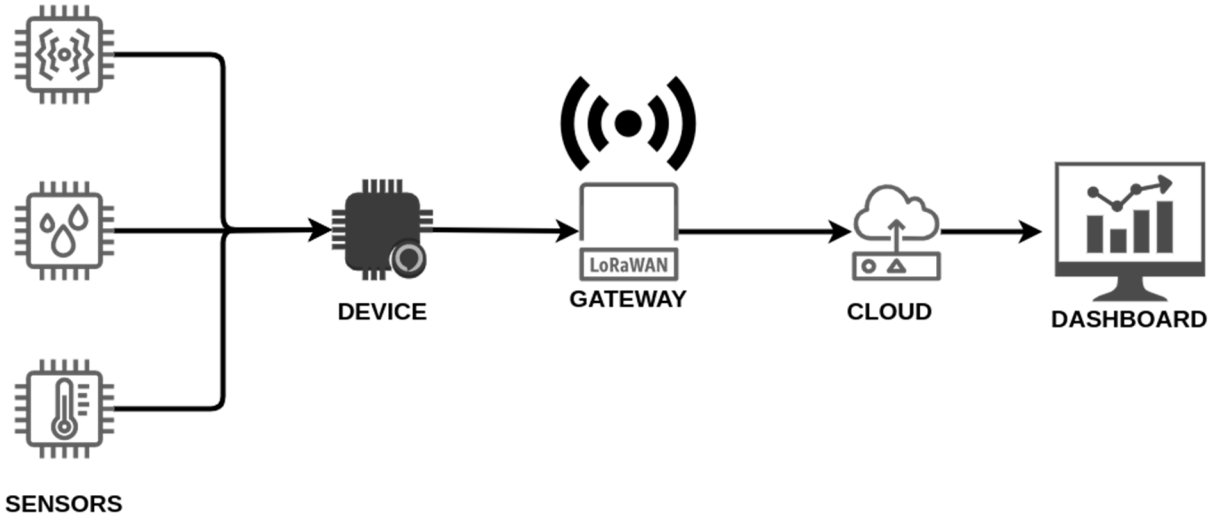

Рисунок 1.2 – Архітектура інформаційної системи контролю параметрів навколишнього середовища

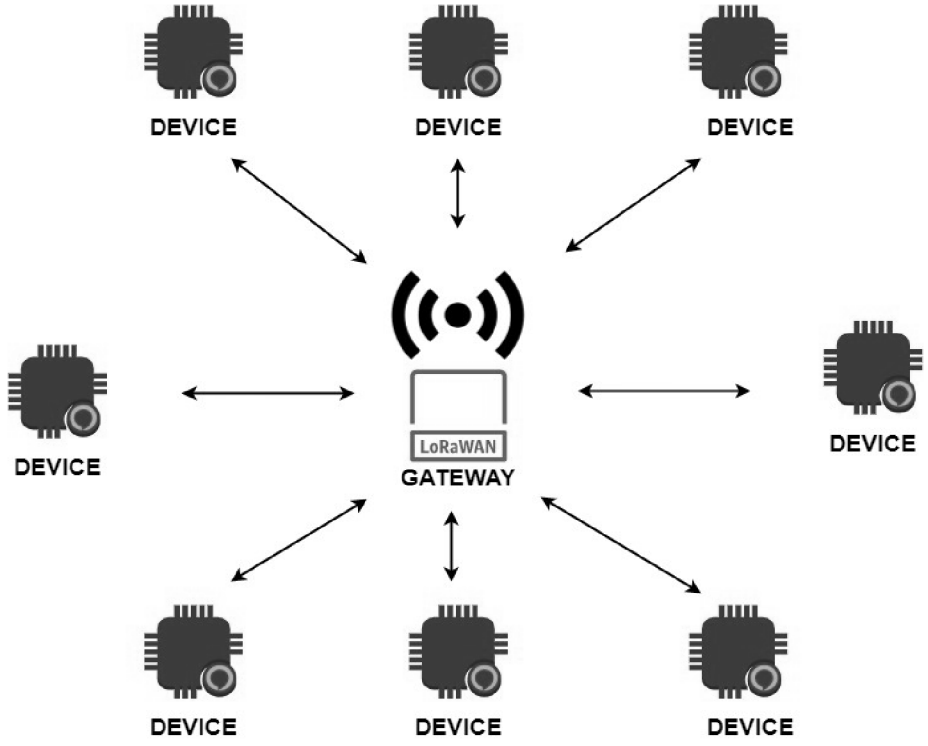

Рисунок 1.3 – Базова топологія мережі LoRaWAN типу «зірка»

Created by Universal Document Converter

У сценаріях, які передбачають використання кількох шлюзів у мережі, зазвичай застосовується зіркова топологія, у якій декілька шлюзів взаємодіють з одним мережевим сервером, як показано на рисунку 1.4.

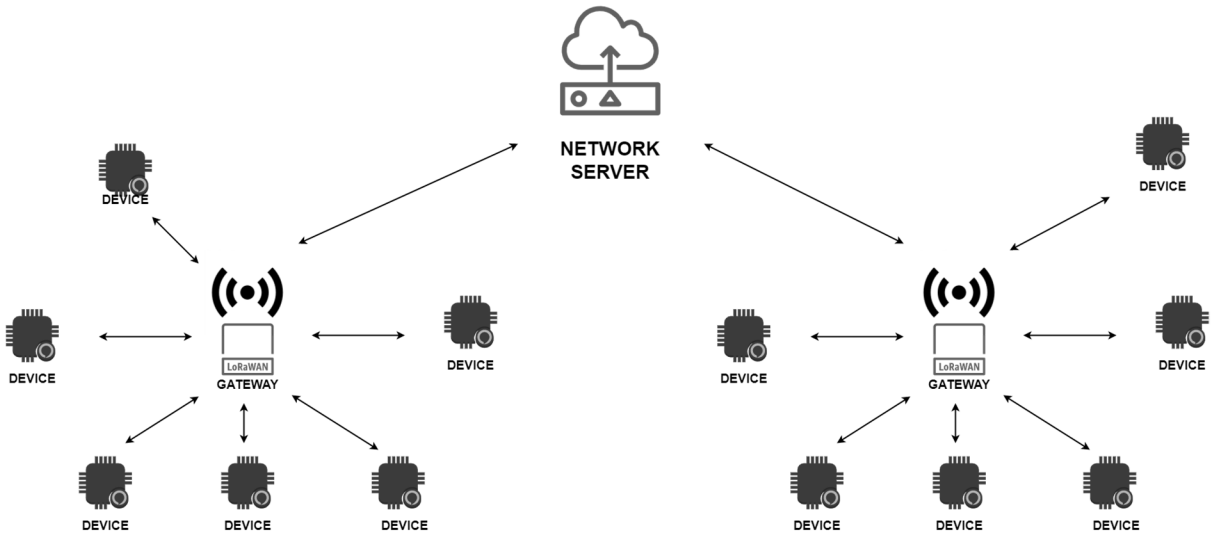

Рисунок 1.4 – Ієрархічна топологія мережі LoRaWAN за схемою «зірка у зірці»

Сьогодні існує багато технологій, що використовуються для подібних систем, зокрема LoRaWAN, вузькосмуговий Інтернет речей (NB-IoT), Sigfox, Zigbee тощо. Вони мають подібні характеристики, а деякі навіть працюють на однакових частотах, проте реалізація протоколів у них відрізняється. Діаграма на рисунку 1.5 [2] ілюструє порівняння основних характеристик цих технологій.

NB-IoT спочатку був розроблений для малопотужних стаціонарних датчиків. Протокол забезпечує широке покриття та добре проникнення сигналу всередину приміщень.

На відміну від LoRaWAN, NB-IoT є ліцензованим протоколом, що може виявитися дорожчим у довгостроковій перспективі, хоча водночас здатен забезпечити кращий загальний досвід для кінцевого користувача.

Sigfox набирає популярності на європейському ринку промислових рішень, однак компанія позиціонує себе як оператор зв’язку, тому цей протокол не варто розглядати як пріоритетний через його закрите операційне середовище.

Created by Universal Document Converter

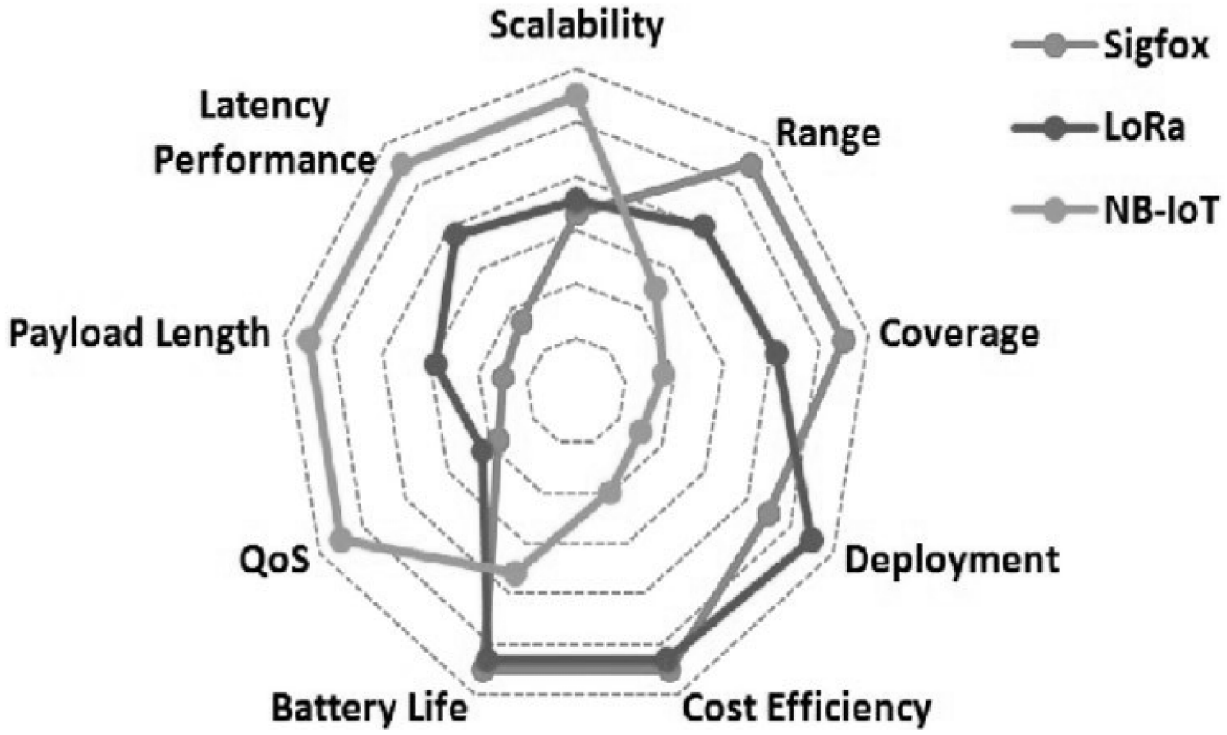

Рисунок 1.5 – Порівняння основних параметрів технологій LoRa, NB-IoT та Sigfox

Zigbee базується на стандарті WPAN IEEE 802.15.4. Тому для деяких застосувань, зокрема для малопотужних пристроїв з низькими вимогами до енергоспоживання (наприклад, датчики розумного будинку), Zigbee розглядають як альтернативу Wi-Fi та Bluetooth. Водночас він навряд чи підходить для покриття великих міських територій або відкритих просторів.

Розглянемо детальніше стандарти LoRaWAN та NB-IoT, щоб визначити особливості їх технічної інтеграції. LoRaWAN споживає менше енергії, ніж NB-IoT, що робить його більш придатним для проєктів, які потребують вищої частоти оновлення. Завдяки нижчому енергоспоживанню LoRaWAN також характеризується тривалішим терміном служби акумулятора, ніж NB-IoT (понад 15 років проти понад 10 років).

Щодо пропускної здатності та покриття, обидва стандарти є порівнянними. NB-IoT має пікову пропускну здатність близько 60 Кбіт/с, що дещо вище, ніж у LoRaWAN. NB-IoT забезпечує вищий рівень безпеки завдяки 256-бітному шифруванню за стандартами 3GPP (LoRaWAN використовує 128-бітне

Created by Universal Document Converter

шифрування AES). NB-IoT іноді має меншу затримку, ніж LoRaWAN. Водночас затримка в LoRaWAN також залежить від типу та класу використовуваних пристроїв. Обидва стандарти підтримують геолокацію.

Однак для LoRaWAN, оскільки ліцензування не потрібне, існує ширший асортимент готових продуктів і компонентів, а їхня вартість є доступнішою, ніж у NB-IoT [3]. Тому, з огляду на вищезазначені фактори, використання недорогого та широко застосовуваного мікроконтролера RISC-V ESP32 (з інтегрованим LoRa-модемом) як периферійного пристрою є доцільним підходом під час реалізації кваліфікаційної роботи. Такий вибір апаратної частини забезпечить: 128-бітове шифрування під час передавання даних, низьке енергоспоживання (режим глибокого сну на стороні мікроконтролера), відносно низьку вартість готового виробу та велику дальність передавання даних.

Аналогічні продукти вже присутні на ринку. У попередні роки з'явилася низка пристроїв на базі процесорів серії STM32 з використанням схожих технологічних модулів LPWAN (NB-IoT та Zigbee).

Пристрій надсилає дані на підключений до Інтернету шлюз на частоті 868 МГц. Зазвичай, для досягнення кращої продуктивності, шлюз підключається через інтерфейс Ethernet (якщо це дозволяють умови розташування). Водночас більшість сучасних шлюзів можуть під'єднуватися до Інтернету через стільникові мережі 3G/4G/LTE, якщо дротове підключення неможливе [4].

Шлюз отримує інформацію лише від зареєстрованих пристроїв і передає її на пристрій зберігання даних або сервер (до бази даних / хмарного середовища). Дані обробляються на сервері та відображаються в зручному для користувача форматі [5].

Технологія LoRa характеризується високою надійністю передавання даних і значною дальністю зв'язку (10–15 км). Одна базова станція може забезпечити двонапрямний зв'язок для понад тисячі пристроїв. Протокол LoRa передає дані зі швидкістю від 0,3 до 50 кбіт/с у бездротовому каналі.

Коли термінальний пристрій починає передавати дані, усі доступні шлюзи (базові станції) прослуховують сигнал і надсилають отриману інформацію на

Created by Universal Document Converter

мережевий сервер. Мережевий сервер визначає, яка базова станція має відповісти пристрою, на основі таких параметрів, як якість сигналу, відстань до пристрою та інші метадані. Після цього дані передаються на сервер застосунків.

Мережі LoRaWAN зазвичай використовують зіркову топологію, у якій кінцеві вузли зв'язуються з центральним мережевим сервером через шлюз. Така архітектура передбачає, що шлюз і центральний сервер належать мережевому оператору, тоді як кінцеві вузли належать користувачам. Між користувачами та кінцевими вузлами можлива прозора двонапрямна передача даних, захищена 128-бітним AES-шифруванням.

1.3 Порівняльний огляд наявних систем моніторингу навколишнього середовища

Сучасні системи моніторингу довкілля охоплюють широкий спектр технологічних рішень – від побутових сенсорів до високоточних професійних метеорологічних комплексів. Кожна платформа має власні підходи до збору, передавання й аналізу екологічних даних, що визначає її точність, сферу застосування та експлуатаційні обмеження. Нижче подано огляд найпоширеніших систем, що використовуються у побутових, міських та науково-виробничих сценаріях.

Платформа IQAir (AirVisual) інтегрує апаратні та аналітичні засоби, забезпечуючи широкий спектр параметрів моніторингу, включно з концентрацією твердих частинок різних розмірів та окремих газоподібних забруднювачів. Система має надійну хмарну інфраструктуру, мобільні додатки, інструменти прогнозування та готові сервіси для відображення стану навколишнього середовища в певних районах. Тому IQAir широко застосовується як у домашніх умовах, так і в корпоративних рішеннях. Водночас її експлуатаційні витрати є вищими порівняно з недорогими сенсорними

Created by Universal Document Converter

платформами, а в наукових дослідженнях іноді виникає потреба в додаткових контрольних калібрувальних вимірюваннях.

Для завдань, що вимагають високої точності, довгострокової стабільності показників та відповідності міжнародним стандартам, у професійній практиці зазвичай використовують пристрої Vaisala. Це складні метеорологічні та екологічні станції моніторингу, які широко застосовуються національними та муніципальними відомствами, в авіаційній промисловості та науково-дослідних установах. Їхні можливості охоплюють широкий спектр параметрів, зокрема концентрацію забруднювальних газів, атмосферні характеристики, дані про вітер та метеорологічні явища. Такі пристрої забезпечують найвищий рівень надійності, проте є дорогими та складними у розгортанні, що робить їх малопридатними для недорогих розподілених сенсорних мереж.

Система PurpleAir є однією з найпоширеніших платформ моніторингу якості повітря в режимі реального часу, яка обслуговує широкий спектр користувачів – від приватних домогосподарств до державних екологічних організацій. Вона базується на невеликих лазерних датчиках, що вимірюють концентрацію дрібних твердих частинок (PM2.5) та пов’язані з ними параметри мікроклімату, зокрема температуру, вологість і тиск повітря. Ці пристрої передають дані в хмару через Wi-Fi, після чого відповідні показники відображаються на глобальній онлайн-карті. Такий підхід дозволяє PurpleAir формувати одну з найщільніших неурядових мереж моніторингу якості повітря, що діють сьогодні, з даними, які оновлюються майже в режимі реального часу та є загальнодоступними.

Ключовими перевагами PurpleAir є низька вартість обладнання, що дає змогу розгортати датчики у великій кількості та створювати детальні карти забруднення. Простота встановлення та невисокі вимоги до технічних знань дозволяють широким верствам громадськості долучатися до моніторингу, тоді як висока частота оновлення даних забезпечує можливість швидкого реагування на локальні зміни якості повітря, зокрема на пожежі, промислові викиди та інші екологічні події. Ще однією важливою перевагою є відкритість даних, що

Created by Universal Document Converter

спрощує проведення досліджень і підвищує прозорість у сфері моніторингу довкілля.

Водночас PurpleAir має й низку суттєвих недоліків, насамперед пов'язаних із точністю вимірювань. Ці датчики не є еталонними приладами та є чутливими до вологості, температури й інших факторів, що можуть спричиняти похибки. Тому потрібне застосування калібрувальних моделей або порівняння з професійними метеостанціями. Тривала експлуатація іноді призводить до погіршення характеристик сенсорних компонентів, а залежність від Wi-Fi обмежує можливість встановлення пристроїв у віддалених районах. З огляду на ці особливості, PurpleAir доцільніше використовувати як інструмент для загального моніторингу якості повітря та попередньої оцінки, тоді як для точних наукових або регуляторних вимірювань мають застосовуватися еталонні прилади.

Пристрій Arable Mark 2 – це високотехнологічний датчик Інтернету речей, призначений для моніторингу росту сільськогосподарських культур і умов навколишнього середовища. Він може вимірювати широкий спектр параметрів, зокрема температуру, вологість, інтенсивність освітлення, тиск повітря, швидкість вітру, кількість опадів і фотосинтетичну активність.

Датчики Arable Mark 2 розміщені в міцному та довговічному корпусі, який надійно захищає їх від негативного впливу навколишнього середовища. Для передавання зібраних даних на центральний сервер, де вони обробляються та аналізуються, використовується бездротовий зв'язок.

Однією з ключових особливостей Arable Mark 2 є його адаптивність до різноманітних сільськогосподарських умов. Завдяки вимірюванню параметрів, що впливають на здоров'я рослин і врожайність (температура, вологість, освітленість, тиск повітря тощо), фермери отримують змогу краще розуміти динаміку росту культур і відповідно планувати свою діяльність.

Libelium Smart Water – це IoT-рішення, розроблене спеціально для моніторингу якості води. Система складається з водонепроникних датчиків, які можна встановлювати в будь-яких водоймах, та центрального сервера для

Created by Universal Document Converter

обробки й аналізу даних. Датчики Libelium Smart Water вимірюють різноманітні параметри, зокрема розчинений кисень, електропровідність, pH, температуру та інші показники якості води.

Зібрані дані передаються на сервер бездротовим способом. Libelium Smart Water використовує сучасні технології для забезпечення високої точності та надійності моніторингу якості води. Завдяки цьому установи, відповідальні за управління водними ресурсами, можуть отримувати достовірні дані про стан води в режимі реального часу та вживати необхідних заходів для захисту безпеки й здоров'я населення.

Cisco Kinetic for Cities – це рішення Інтернету речей, яке забезпечує комплексний підхід до моніторингу різних аспектів міського середовища. Система збирає, обробляє та аналізує дані з різноманітних джерел, зокрема датчиків якості повітря, шумових датчиків і метеостанцій.

Платформа Cisco Kinetic for Cities базується на модульній архітектурі, що спрощує додавання нових датчиків і розширення функціональних можливостей системи. Кожен датчик передає дані бездротовим способом на центральний сервер для подальшої обробки та аналізу. Така архітектура робить систему гнучкою та легко масштабованою.

Ключовою перевагою Cisco Kinetic for Cities є здатність надавати повні та точні дані про стан міського середовища в режимі реального часу. Це дає змогу муніципальним органам влади та іншим установам оперативно реагувати на зміни умов довкілля та ухвалювати своєчасні управлінські рішення.

Система також використовує алгоритми машинного навчання для виявлення закономірностей і тенденцій у даних. Наприклад, вона може виявляти певні патерни забруднення повітря або зміни рівня шуму. Отриману інформацію можна використовувати для вдосконалення міського планування та управління ресурсами. У таблиці 1.1 подано порівняння та опис наявних систем моніторингу навколишнього середовища за низкою основних параметрів.

Created by Universal Document Converter

Таблиця 1.1 – Порівняльний аналіз технічних та функціональних можливостей сучасних систем моніторингу

Назва системи	Контрольовані параметри	Технології передачі даних	Топологія та архітектура	Аналіз інтеграційних можливостей
AirVisual	PM2.5, PM10, CO_2, температура, вологість, іноді додаткові газові сенсори залежно від моделі	Wi-Fi, хмарна передача даних через мобільний застосунок	Хмарно-орієнтована система з локальним сенсорним модулем і централізованою обробкою даних	Підтримка API, взаємодія з мобільними застосунками, можливість вбудовування у веб-сервіси та сторонні аналітичні платформи
Libelium Smart Water	Концентрація розчиненого кисню, електропровідність, водневий показник, температура, мутність, колірність води	Wi-Fi, 3G/4G, LoRaWAN, ZigBee	Модульна мережева архітектура з інтегрованими бездротовими сенсорами	Можливість інтеграції зі сторонніми системами через вбудований API
PurpleAir	PM1.0, PM2.5, PM10, температура, вологість, атмосферний тиск	Wi-Fi з передачею даних на онлайн-карту PurpleAir	Децентралізована мережа великої кількості локальних сенсорів із загальною відкритою хмарною платформою	Публічне API, можливість інтеграції з дослідницькими платформами, екологічними картами та сторонніми сервісами
Arable Mark 2	Кліматичні та мікрокліматичні показники, рівень фотосинтезу рослин	LoRaWAN, LTE	Централізована мережа, що передає інформацію від сенсорів на сервер	Можливість інтеграції зі сторонніми системами через вбудований API
Cisco's Kinetic for Cities	Метеорологічні дані, рівень шуму, показники якості повітря	LTE, Wi-Fi	Централізована мережа, що передає інформацію від сенсорів на сервер	Можливість інтеграції зі сторонніми системами через вбудований API

Created by Universal Document Converter

1.4 Функціональне призначення та задачі розроблюваної системи

Розроблена система повинна мати такі функції:

- дистанційний моніторинг різних параметрів середовища, в якому встановлено обладнання;
- безпечна передача та зберігання даних;
- можливість розширення системи за потреби;
- візуалізація виміряних параметрів;
- прогнозування контрольованих параметрів.

Система повинна виконувати передусім інформаційні функції: передавати значення параметрів, повідомляти про стан обладнання та потенційні збої (наприклад, неможливість опитування датчиків). Передача та зберігання даних мають бути достатньо захищеними. Крім того, пріоритетом має бути надійне зберігання отриманих даних, щоб забезпечити подальше використання та аналіз історичних показників.

Система, що відповідає цим вимогам, може замінити застарілі технології, які нині використовуються для моніторингу навколишнього середовища, тим самим спрощуючи роботу персоналу та заощаджуючи кошти в довгостроковій перспективі. Окрім цього, система повинна передбачати зручне розширення шляхом додавання нових пристроїв до вже налаштованої захищеної мережі.

.

Created by Universal Document Converter

РОЗДІЛ 2
СТРУКТУРНЕ ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДИСТАНЦІЙНОГО МОНІТОРИНГУ

2.1 Вибір програмної платформи та архітектури мережевого сервера LoRaWAN

Для розгортання бекенд-сервера мережі LoRa було обрано платформу ChirpStack. ChirpStack – це мережевий сервер LoRaWAN з відкритим кодом, призначений для налаштування та керування мережами LoRaWAN. Він надає вебінтерфейс для керування шлюзами, пристроями та налаштування інтеграції з основними постачальниками хмарних послуг, базами даних і поширеними сервісами обробки даних пристроїв. ChirpStack також надає API на основі gRPC для інтеграції та розширення своєї функціональності.

gRPC – це сучасна високопродуктивна платформа віддаленого виклику процедур з відкритим кодом, яка може працювати в будь-якому середовищі. Вона ефективно підключає сервіси в центрі обробки даних і підтримує балансування навантаження, трасування, перевірку справності та автентифікацію. gRPC також підходить для завершальних етапів розподілених обчислень, підключення пристроїв, мобільних застосунків і браузерів до серверних служб.

Архітектуру мережевого сервера LoRaWAN подано на рисунку 2.1.

Основні компоненти сервера включають: шлюзовий міст, який з'єднує генератор пакетів, установлений на базовій станції (шлюзі), з архітектурою сервера LoRaWAN; мережевий сервер для роботи з повідомленнями мережевого рівня та сервер застосунків, що забезпечує мережеві операції на рівні користувача та інтегрується із зовнішніми платформами.

Created by Universal Document Converter

Серед допоміжних компонентів сервера відзначимо: MQTT-брокер Mosquitto, який відповідає за внутрішню передачу повідомлень між компонентами сервера; проміжну базу даних для зберігання тимчасових даних Redis та базу даних для зберігання постійних даних PostgreSQL.

Усе вищезгадане програмне забезпечення розповсюджується за ліцензією програмного забезпечення з відкритим вихідним кодом.

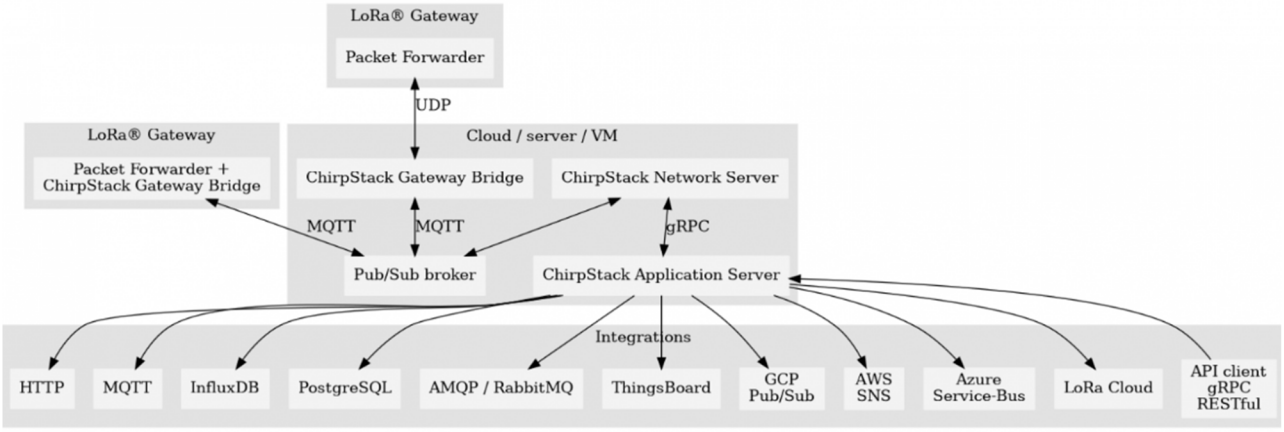

Рисунок 2.1 – Структура граничного вузла системи дистанційного моніторингу

2.2 Розгортання та налаштування серверної інфраструктури ChirpStack

Розглянемо етапи налаштування сервера ChirpStack згідно з рекомендаціями, наведеними у документації до програмного продукту. Перш за все рекомендується розгортати програмне забезпечення на комп'ютері з Ubuntu 22.04 LTS або Debian 11.

Наведені нижче команди терміналу активують менеджер пакетів для встановлення рекомендованих компонентів ChirpStack:

```
sudo apt install \
    mosquitto \
    mosquitto-clients \
```

Created by Universal Document Converter

```
    redis-server \
    redis-tools \
    postgresql.
```

Доступ до утиліти командного рядка PostgreSQL можна отримати за допомогою команди:

```
sudo -u postgres psql.
```

Потрібно виконати наведений нижче запит, щоб налаштувати базу даних ChirpStack. Рекомендується при цьому змінити ім'я користувача та пароль:

```
-- 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.
```

ChirpStack надає репозиторій програмного забезпечення для Debian/Ubuntu, який можна використовувати для встановлення останньої версії ChirpStack. Спочатку перевіримо чи встановлено dirmngr та apt-transport-https за допомогою команди:

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

Далі налаштуємо ключ репозиторію ChirpStack:

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

та додамо репозиторій до списку джерел пакетів:

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

Після цього слід оновити кеш пакетів APT:

```
sudo apt update.
```

Created by Universal Document Converter

Якщо шлюз не забезпечує роботу ChirpStack Gateway Bridge, необхідно виконати початкове встановлення через APT, використавши наведену нижче команду в терміналі:

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

Наступний етап передбачає налаштування та конфігурування даного мережевого моста. Сам файл конфігурації розташований за адресою `/etc/chirpstack-gatewaybridge/chirpstack-gateway-bridge.toml`. Необхідно оновити розділ `[integration.mqtt]`, щоб він відповідав префіксу зони, який застосовується до цього екземпляра ChirpStack Gateway Bridge.

Приклад для EU868:

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

далі запускаємо 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.
```

Для встановлення самого пакета ChirpStack, як і на попередніх етапах, використовується менеджер пакетів APT:

```
apt install chirpstack.
```

Файли його конфігурації розташовані в каталозі /etc/chirpstack. Тут розміщено глобальний файл конфігурації chirpstack.toml, а також окремі файли конфігурації, що відповідають конкретним регіонам.

Запуск сервісу ChirpStack здійснюється за командою:

```
# start chirpstack
sudo systemctl start chirpstack
```

Created by Universal Document Converter

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

а вивід логів (журналу подій) ChirpStack реалізовано через:

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

2.3 Конфігурування шлюзів і кінцевих пристроїв LoRaWAN

Компонент моста **ChirpStack Gateway Bridge** виконує роль сервера для засобу пересилання пакетів (packet forwarder). Він взаємодіє з платформою ChirpStack за допомогою протоколу MQTT. Щоб виконати налаштування, необхідно коректно ініціалізувати обидва компоненти.

Засіб пересилання пакетів, налаштований на шлюзі, повинен передавати свої дані до компонента ChirpStack Gateway Bridge. Оскільки він керує чипсетом LoRa-шлюзу, його також потрібно налаштувати на правильний діапазон частот. Невідповідність частот призведе до того, що шлюз не зможе приймати вихідні канали зв'язку від пристрою та/або надсилати вхідні корисні навантаження, якщо частота вхідного каналу виходить за межі налаштованого діапазону. Постачальники шлюзів зазвичай надають приклади конфігурацій для різних діапазонів частот.

Налаштування, які визначають, до яких параметрів брокера MQTT підключатиметься компонент ChirpStack Gateway Bridge, наведено нижче:

```
# 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", ].
```

Created by Universal Document Converter

Для додавання шлюзу необхідно перейти до розділу «Шлюзи» у вебінтерфейсі, натиснути «Додати» та заповнити форму. Якщо все пройде успішно то шлюз з'явиться у мережі.

Оскільки міст ChirpStack Gateway Bridge використовує MQTT для зв'язку з ChirpStack також можна підписатися на MQTT-розсилку шлюзу. Це дасть змогу підтвердити, що дані зі шлюзу принаймні досягають MQTT-сервера.

Нижче наведено приклад команди для підписки на MQTT-розсилку шлюзу за допомогою mosquitto_sub:

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

Щоб підключити кінцевий пристрій до наявної мережі, необхідно знати таку інформацію:

- DevEUI;
- версію LoRaWAN MAC, реалізовану на пристрої;
- налаштування регіональної версії, реалізовані на пристрої;
- DevAddr;
- сеансовий ключ.

Цю інформацію зазвичай надає постачальник пристрою. Серійний номер може висвітлюватись на LoRa-модемі, а виробник також надає документ, що містить мережевий ключ для цього конкретного серійного номера, який зазвичай розміщено під написом «FCC ID» на модулі як показано на рисунку 2.2:

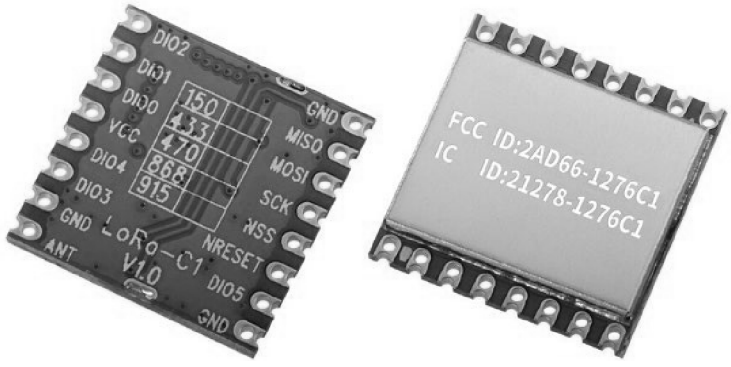

Рисунок 2.2 – Зовнішній вигляд LoRaWAN-модема із зазначенням серійного номера

Created by Universal Document Converter

Перш ніж додавати пристрій до ChirpStack, необхідно створити профіль пристрою. Рекомендується створювати окремі профілі для різних типів пристроїв. Профіль пристрою містить інформацію про його можливості, зокрема про те, чи активується пристрій за допомогою ABP (персоналізована активація) чи OTAA (активація «по повітрю»), які версії LoRaWAN та регіональні налаштування він підтримує тощо. У профілі також можна налаштувати функцію декодування корисних навантажень, що надсилаються пристроями, пов'язаними з цим профілем. Пристрої групуються, наприклад, можна об'єднати датчики температури в одну програму, а метеостанції – в іншу.

Щоб створити пристрій на вкладці «Пристрої» слід натиснути кнопку «Створити». Потім треба заповнити обов'язкові поля, вибрати профіль пристрою та зберегти його. Залежно від того, чи налаштовано профіль пристрою для OTAA чи ABP, на наступній сторінці буде запропоновано ввести кореневий ключ пристрою (OTAA) або сеансові ключі пристрою (ABP). Якщо екземпляр ChirpStack налаштовано з використанням сервера приєднання і пристрій буде активовано через цей сервер, вводити кореневий ключ не потрібно.

2.4 Інтеграція з Node-RED та обробка телеметричних даних

Далі необхідно реалізувати інтеграцію розробки з Node-RED, яке надає браузерний редактор потоків, що дає змогу легко будувати логіку обробки даних із використанням різних вузлів з палітри [7].

Node-RED потрібно активувати за допомогою утиліти gateway-config. Після ввімкнення доступ до Node-RED здійснюється через браузер за адресою http://[IP-адреса]:1880. Наприклад, якщо IP-адреса шлюзу – 192.168.0.1, доступ до Node-RED буде за адресою http://192.168.0.1:1880. Побудова echo-потоків показана на рис. 2.3.

Created by Universal Document Converter

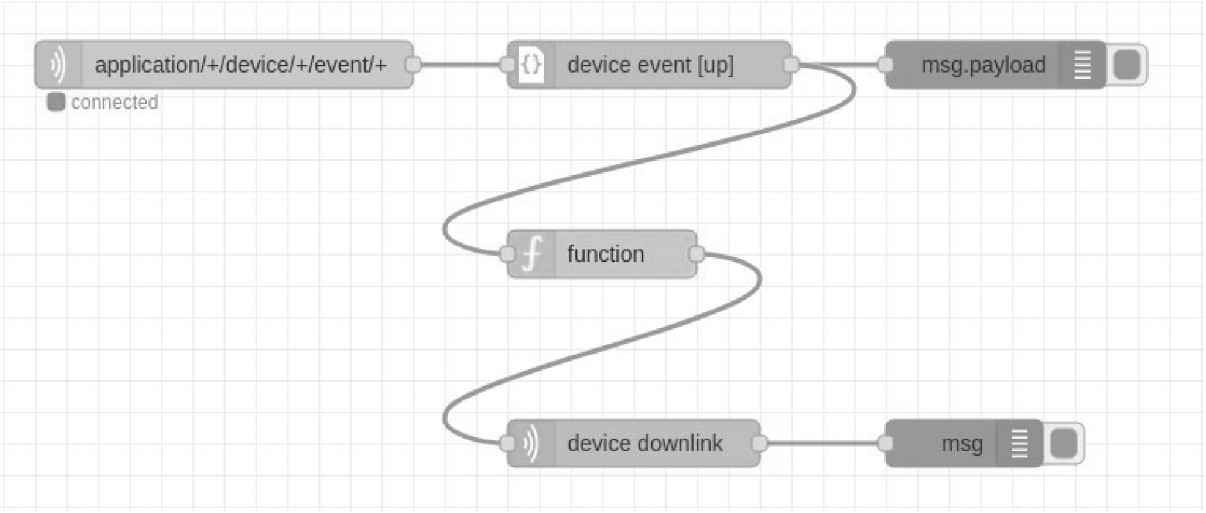

Рисунок 2.3 – Потік обробки пакетів у середовищі Node-RED для каналу LoRaWAN

Наведені нижче кроки описують, як створити echo-потік, який ставить у чергу повідомлення, отримані від пристрою, як вихідні дані.

Після відкриття Node-RED у браузері спочатку потрібно налаштувати вузол MQTT. Цей вузол буде використовуватися для отримання подій від пристроїв, опублікованих брокером MQTT, який працює на шлюзі (надається ОС ChirpStack Gateway). Для цього потрібно встановити такі властивості вузла:

Сервер: localhost;

Порт: 1883;

Тема: application/+/device/+/event/+;

QoS: 0;

Вивід: автоматичне визначення;

Синхронізація подій.

Наступним кроком є налаштування вузла події пристрою. Цей вузол аналізує корисне навантаження, сформоване вузлом MQTT, і фільтрує його відповідно до типу події. Потрібно встановити такі властивості вузла:

Тип події: вихідне повідомлення.

Після цього необхідно підключити вихід вузла mqtt-in до входу вузла події пристрою.

Created by Universal Document Converter

Щоб змоделювати подію, яка надсилає повідомлення як корисне навантаження у вхідну чергу треба додати вузол функції та підключити її вхід до виходу вузла події пристрою та встановити такі властивості вузла:

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

Також необхідно отримати токен API у вебінтерфейсі ChirpStack. Щоб надсилати повідомлення, згенеровані функцією echo, до ChirpStack, потрібно додати вузол низхідного каналу пристрою та підключити його вхід до виходу вузла функції встановивши такі властивості:

Сервер: localhost:8080;

Токен API: токен API, отриманий із вебінтерфейсу ChirpStack;

Кодування корисного навантаження: Base64;

Черга надсилання (налагодження).

Щоб переглядати повідомлення, згенеровані вузлом низхідного каналу пристрою, потрібно додати ще один вузол налагодження та підключити його вхід до виходу вузла низхідного каналу пристрою.

Після виконання цих кроків створений потік зможе отримувати й надсилати повідомлення із сервера. Під час передавання пакетів даних на панелі налагодження можна побачити два типи повідомлень:

– вхідне повідомлення;

– відповідь із вихідної черги.

Пакети даних кодуються у форматі Base64 під час передавання мережею, що дає змогу суттєво зменшити обсяг корисного навантаження. У багатьох IoT-

Created by Universal Document Converter

застосунках (таких, як LoRa, LTE-M, NB-IoT, Sigfox тощо) корисне навантаження перед кодуванням у JSON зазвичай спершу кодується в Base64.

Base64 переважно використовують для кодування двійкових форматів. Оскільки двійковий формат є природним представленням даних для комп'ютерів, він є найпростішим варіантом для вбудованих пристроїв з обмеженими ресурсами. Алгоритм Base64 концептуально дуже простий і потребує незначних ресурсів для реалізації, що робить його придатним для передачі двійкових даних через текстові канали. Натомість формування JSON-записів зазвичай вимагає використання JSON-бібліотеки, яка споживає додаткову пам'ять і місце в коді.

Кодування запису даних, який містить 32-бітну позначку часу, 8-бітний код типу даних (наприклад, температура, напруга або тиск) та 32-бітний зразок даних, потребує 9 байтів, що після кодування Base64 збільшуються до 12 байтів. Натомість простий JSON-запис (без жодних пробілів) займатиме 67 байтів.

Created by Universal Document Converter

РОЗДІЛ 3
ІМПЛЕМЕНТАЦІЯ СИСТЕМИ ДИСТАНЦІЙНОГО МОНІТОРИНГУ

3.1 Розробка загальної архітектури та принципів роботи системи дистанційного моніторингу стану навколишнього середовища

На основі блок-схеми процесу, поданої на рисунку 1.1, розглянемо структуру граничного пристрою (рис. 3.1).

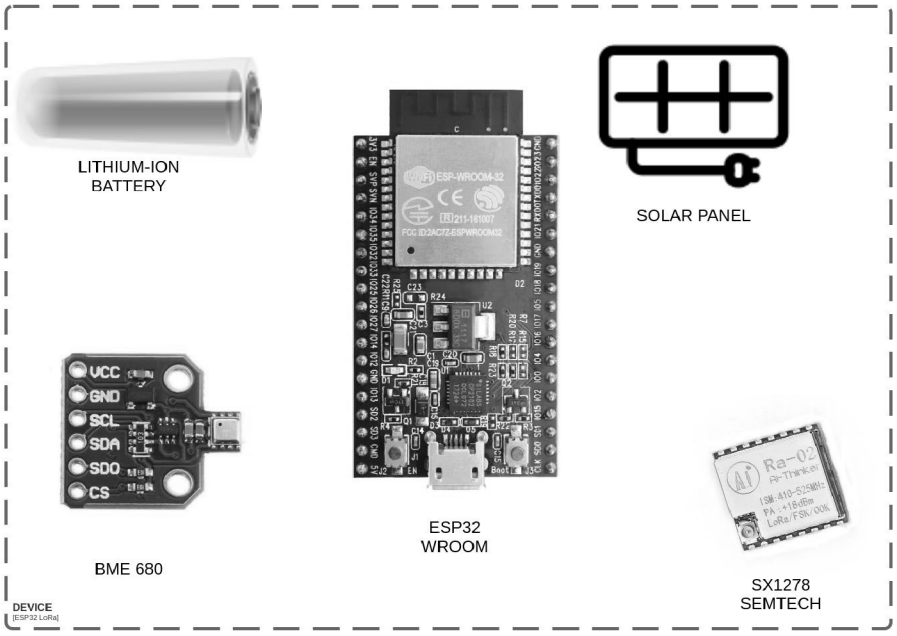

Рисунок 3.1 – Компонування функціональних модулів системи моніторингу стану навколишнього середовища на базі мікроконтролера ESP32-S3

Контролер ESP32 живиться від літій-іонного акумулятора та сонячної панелі. Оскільки контролер переважно перебуває в режимі глибокого сну та «прокидається» лише під час опитування датчиків і надсилання пакетів даних [6], він може забезпечувати тривалий час роботи від акумулятора протягом більшої частини року.

Created by Universal Document Converter

Модуль передавача, що підтримує протокол LoRa, і Semtech SX1278 було обрано, оскільки він є надійним, широко використовуваним та доступним. Що стосується датчиків температури, вологості та якості повітря, то економічні «домашні» рішення зазвичай використовують датчики температури й вологості DHT11/22, датчик якості повітря AM2032, MQ-135 або аналогічні продукти. Також для цих задач можна використати універсальний датчик Bosch BME680, який може вимірювати температуру, відносну вологість, тиск повітря, а також вміст різноманітних газів та летких органічних сполук, що істотно впливають на індекс якості повітря.

На рисунку 3.2 наведено структурну схему мікроконтролера на кристалі ESP32-S3, що використовується як базовий обчислювальний модуль у розробленій IoT-системі. Чип містить двоядерний 32-бітний процесор Xtensa LX7 з кеш-пам'яттю, вбудованою SRAM та ПЗП, а також інтерфейс налагодження JTAG. Окремий блок Wireless MAC and Baseband забезпечує роботу вбудованих інтерфейсів Wi-Fi та Bluetooth Low Energy, тоді як RF-тракт включає приймач, передавач і узгоджувальні ланки для роботи в діапазоні 2,4 ГГц. У блоці периферії інтегровано інтерфейси SPI, I²C, UART, I²S, USB OTG, контролери АЦП, таймери, ШІМ-канали для керування світлодіодами, інтерфейс камери, сенсор дотику та загальновживані GPIO-виводи. Окрема підсистема реального часу з власною пам'яттю, GPIO та I²C забезпечує енергоощадні режими сну й роботу допоміжного ULP-процесора. Підсистема безпеки містить апаратну підтримку алгоритмів SHA, AES, RSA, генератор випадкових чисел, модулі цифрового підпису, захищеного завантаження та шифрування зовнішньої флеш-пам'яті, що підвищує захищеність IoT-вузла. Для організації обміну даними з зовнішнім радіомодулем LoRa в даній роботі передбачається використання одного з трьох вбудованих апаратних інтерфейсів UART, які забезпечують надійну послідовну передачу телеметричної інформації.

Created by Universal Document Converter

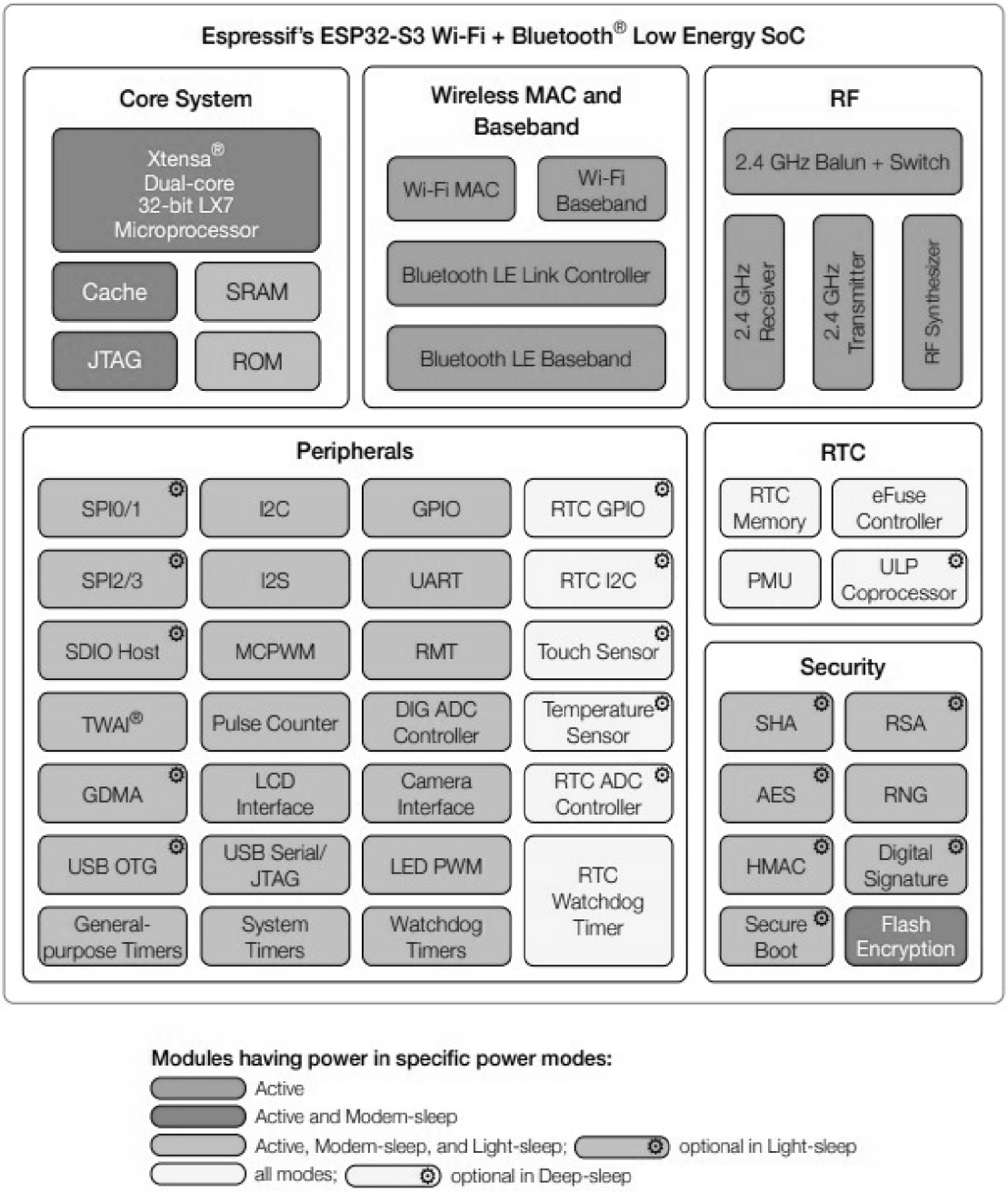

Рисунок 3.2 – Блок-схема основних функціональних модулів мікроконтролера ESP32-S3

На рис. 3.3 показано схему підключення інтерфейсів введення/виведення (I/O) процесора до інших компонентів, зокрема USB-модулів, ліній живлення плати та кнопок, а також розташування контактів на інтерфейсі підключення модуля розробки. Розташування фізичних виводів плати ESP32-DevKitC відображено на рис. 3.4.

Created by Universal Document Converter

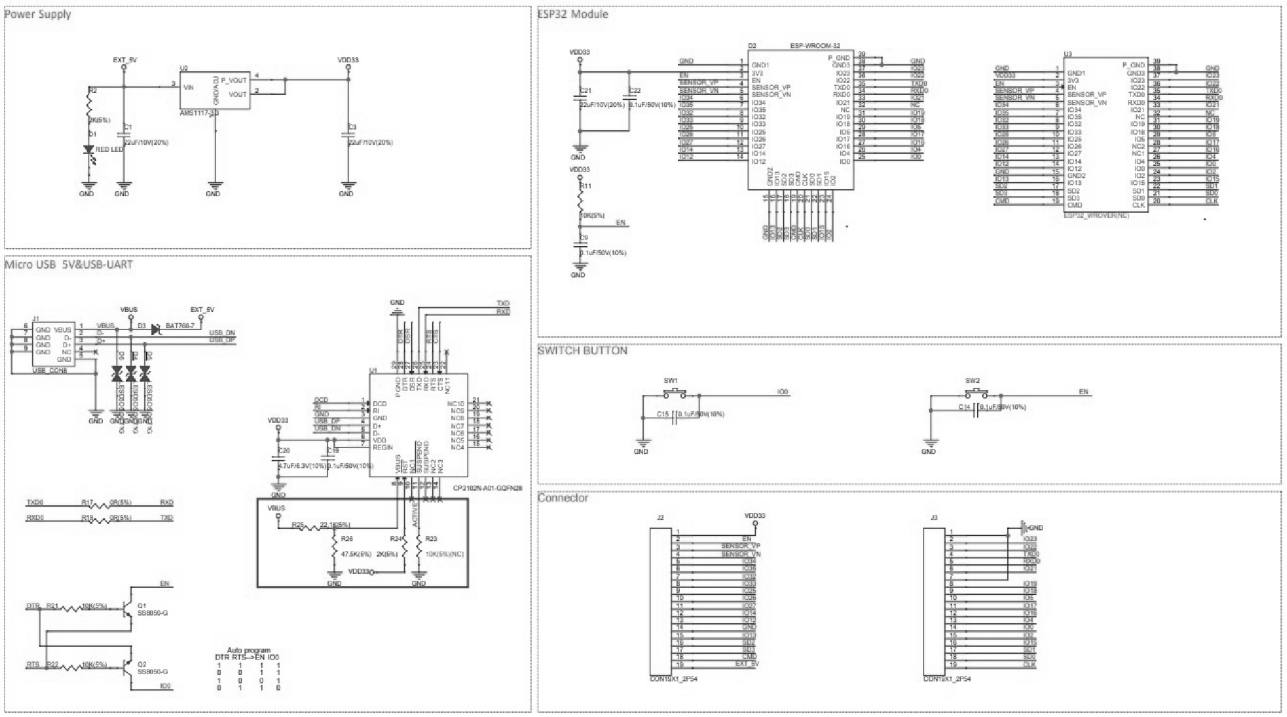

Рисунок 3.3 – Електрична схема підключення плати розробки Espressif ESP32 WROOVER-KIT

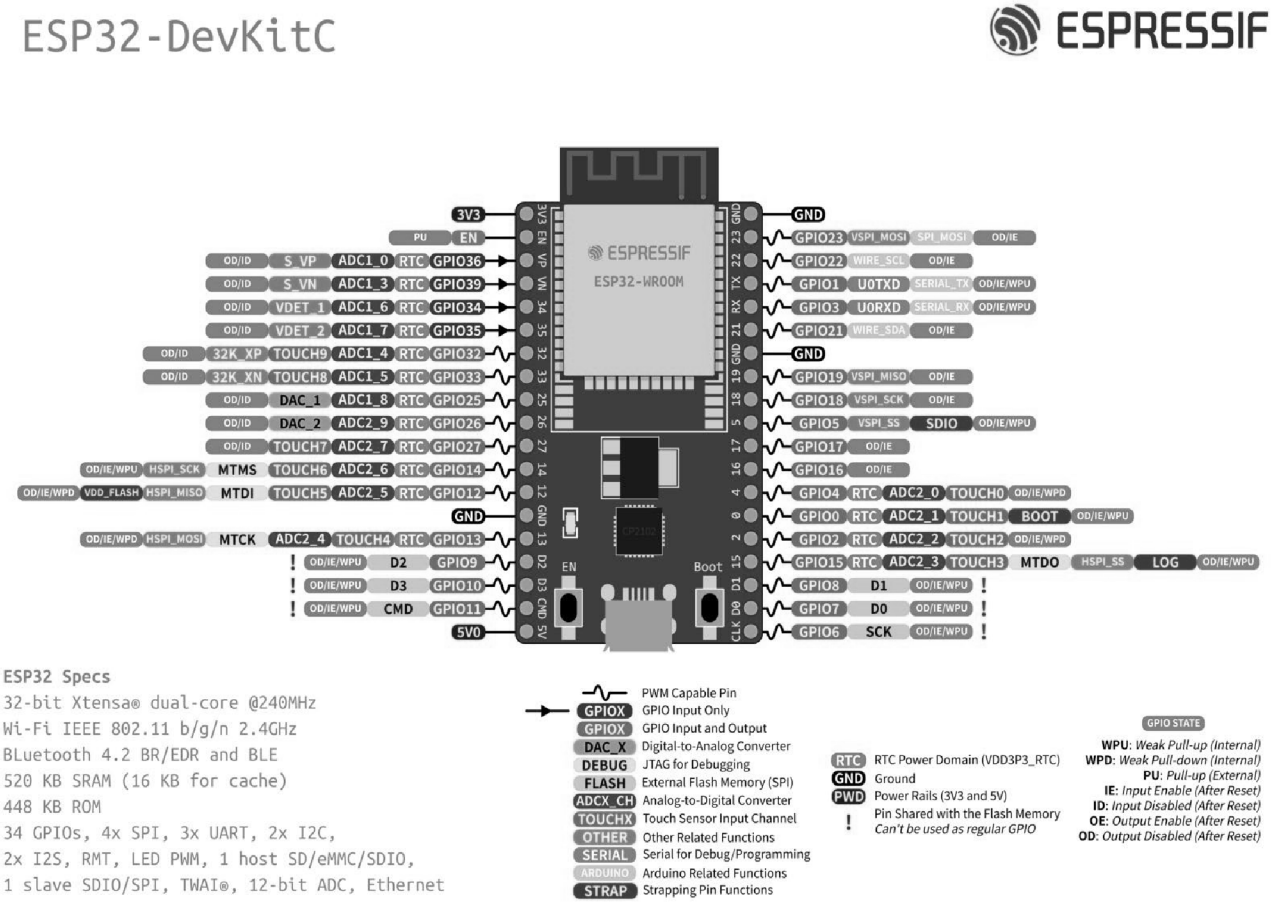

Рисунок 3.4 – Розташування фізичних виводів плати ESP32-DevKitC

Created by Universal Document Converter

На рис. 3.5 подано фрагмент принципової електричної схеми IoT-пристрою на базі мікроконтролера ESP32. Живлення системи забезпечує акумулятор формату 18650, підключений через тримач B1 до шини 18650VCC та загальної шини GND. Заряджання й захист акумулятора реалізовано мікросхемою TP4056 (U4), у якій вхідні виводи IN+ та IN– підключені до джерела 18650VCC/GND, а вихідні OUT+ та OUT– формують стабілізовану шину живлення VCC і загальний провід для решти вузлів. До цієї шини підключено плату контролера ESP32-DEVKITC (U3), що виступає центральним обчислювальним модулем системи та керує периферійними пристроями.

Радіомодуль SX1276 (U1) реалізує бездротовий зв’язок LoRa. Лінії MOSI, MISO, SCK, CS та інші сигнальні виводи модуля підключені до відповідних GPIO ESP32, а вихід ANT з’єднаний із зовнішньою антеною 868 МГц через SMA-роз’єм RF2, для якого передбачено джампер SJ1.

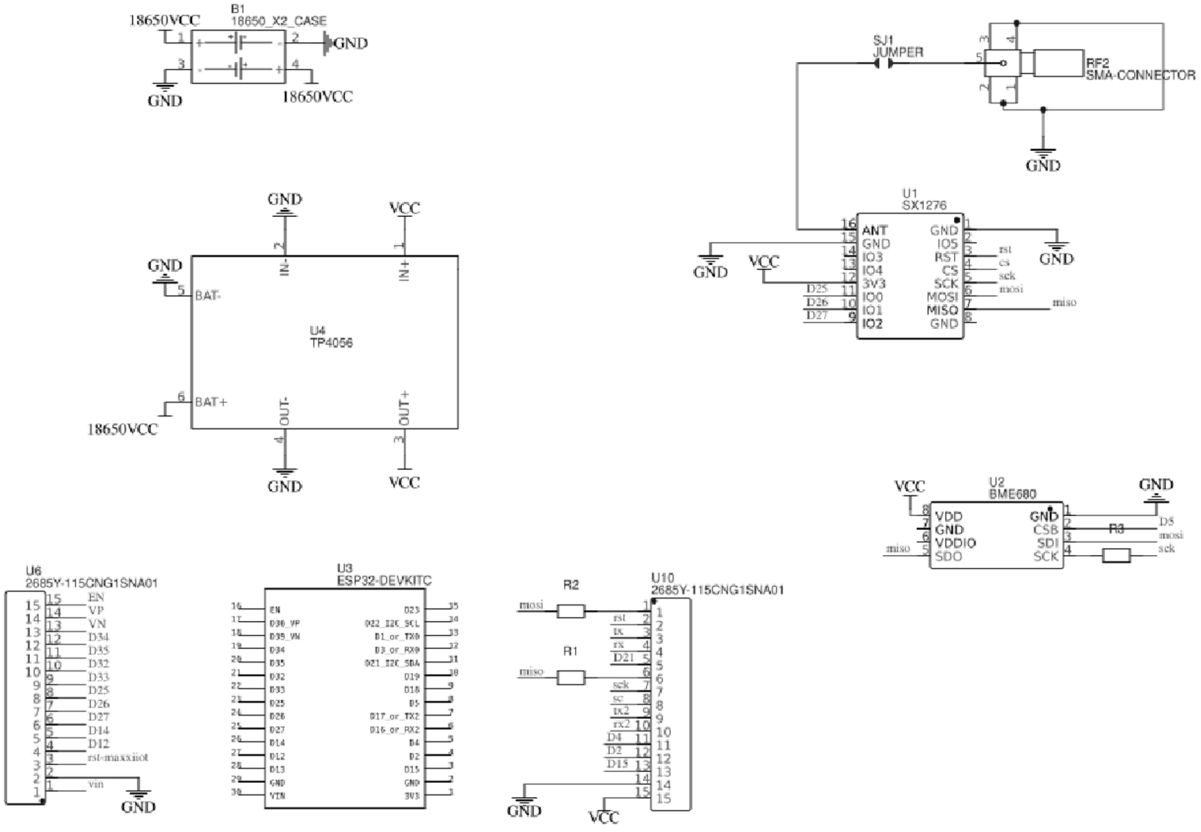

Рисунок 3.5 – Електрична схема підключення датчиків та LoRa-модема граничного пристрою

Created by Universal Document Converter

Для підтримки автономної роботи системи було обрано двоелементний акумуляторний блок 3,7 В формату 18650. Основним джерелом живлення для схеми є мережеве підключення, однак резервне живлення від акумуляторів може підтримуватися шляхом додавання монокристалічної кремнієвої сонячної панелі потужністю 1…5 Вт, розмір якої відповідає габаритам корпусу готового пристрою.

Окремим вузлом є датчик якості повітря й параметрів довкілля BME680 (U2), який підключається до ESP32 по послідовному інтерфейсу (SPI) за допомогою виводів SDO, SDI, SCK, CSB, з'єднаних з контактами мікроконтролера. У схемі також присутні допоміжні роз'єми/модулі U6 та U10 для розведення сигнальних ліній, а також резистори R1 і R2, які виконують функції узгодження або підтягування ліній інтерфейсу. Таким чином, рисунок демонструє пов'язану між собою систему живлення від літієвого акумулятора, зарядний модуль TP4056, обчислювальний модуль ESP32, LoRa-радіомодуль SX1276 з зовнішньою антеною та датчик BME680, що разом утворюють бездротовий сенсорний вузол.

3.2 Імплементація вбудованого програмного забезпечення граничного пристрою

Розроблення програмного забезпечення для ESP32 базується на популярному C/C++-фреймворку Arduino [8]. Це дає змогу розробникам швидко розпочати створення прототипів рішень завдяки відносно низькому порогу входу, що не потребує глибоких спеціальних знань. Крім того, використання цього фреймворку дозволяє застосовувати численні бібліотеки для підтримки різних інтерфейсів і протоколів, які активно розвиваються спільнотою розробників.

Created by Universal Document Converter

Розглянемо основні функції, що використовуються в роботі пристрою. Надсилання повідомлень здійснюється через послідовний інтерфейс. Модуль LoRa повинен отримувати AT-команди, які містять необхідне корисне навантаження для передавання на сервер. Основний синтаксис таких команд, наданий виробником пристрою, показано на рисунку 3.6.

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 3.6 – Формат командного повідомлення для обміну даними з сервером

Реалізацію функцій генерації команд і планування подано на рисунку 3.7.

```
// Функція формує та надсилає AT-команду з даними payload через LoRa
void send_lora_message(String payload)
{
    // Початкова частина AT-команди для відправки даних
    String base = "AT+LRSEND=";

    // Перетворюємо номер порту LoRa у рядок
    String port = String(LORA_PORT);
    base = base + port + ",1,";  // додаємо порт та інші параметри до команди

    // Обчислюємо довжину корисного навантаження (у символах/байтах) і перетворюємо в рядок
    String length = String((((payload.length()) / 2) + 1));
    base = base + length + ",<" + payload;     // дописуємо довжину та сам payload

    // Перетворюємо усю команду у верхній регістр (для сумісності з модемом)
    base.toUpperCase();

    // Виводимо сформовану команду в консоль ПК та на модуль LoRa
    PC.println(base);
    Lora.println(base);
}
```

Рисунок 3.7 – Фрагмент функції надсилання повідомлення на сервер LoRaWAN

Created by Universal Document Converter

Для аналізу вхідних повідомлень від сервера створено функцію LoRaEvent, тіло якої подано на рисунку 3.8.

```
void LoRaEvent()               // Обробка події отримання повідомлення LoRa
{
    if (Lora.available() > 0)  // Перевірка, чи є нові дані від модуля LoRa
    {
        String incoming = Lora.readStringUntil('\n');  // Зчитування вхідного рядка до символу нового рядка
#ifdef PC_DEBUG
        PC.println(incoming);  // Виведення сирого повідомлення у консоль для налагодження
#endif
        byte indexcommand = incoming.indexOf(":");     // Пошук першого роздільника та виділення типу повідомлення
        message.type = incoming.substring(1, indexcommand);
        indexcommand += 1;
        byte subindex = incoming.indexOf(",", indexcommand);
        message.counter = incoming.substring(indexcommand, subindex);
        subindex += 1;
        indexcommand = incoming.indexOf(",", subindex);
        message.port = incoming.substring(subindex, indexcommand);
        indexcommand += 1;
        subindex = incoming.indexOf(",", indexcommand);
        message.rssi = incoming.substring(indexcommand, subindex);
        subindex += 1;
        indexcommand = incoming.indexOf(",", subindex);
        message.snr = incoming.substring(subindex, indexcommand);
        indexcommand += 1;
        subindex = incoming.indexOf(",", indexcommand);
        message.length = incoming.substring(indexcommand, subindex);
        subindex += 2;
        indexcommand = incoming.indexOf(",", subindex);
        message.payload = incoming.substring(subindex, indexcommand); // Виділення корисного навантаження з рядка
        indexcommand += 1;
        subindex = incoming.indexOf(",", indexcommand);
        message.freq = incoming.substring(indexcommand, subindex);    // Отримання робочої частоти повідомлення
        subindex += 1;
        indexcommand = incoming.indexOf(",", subindex);
        message.data = incoming.substring(subindex, indexcommand);    // Збереження основних даних повідомлення
        incoming = "";                                                // Очищення буфера вхідного рядка
        return;                                                       // Вихід із обробника події
    }
}
```

Рисунок 3.8 – Алгоритм обробки вхідних подій LoRa у функції «LoRaEvent»

Для обробки повідомлень у мережі створено структуру struct loramessage, яка містить основні параметри, необхідні для взаємодії із сервером, як показано на рис. 3.9. До її складу входять поля type, counter, port, rssi, snr, length, payload, freq, data та previous_type, що дає змогу в межах одного об’єкта зберігати тип повідомлення, його порядковий номер і порт, параметри якості радіоканалу (рівень сигналу RSSI та відношення сигнал/шум SNR), довжину й вміст корисного навантаження, робочу частоту передавання, додаткові службові дані, а також тип попередньо обробленого повідомлення. Оголошення глобального

Created by Universal Document Converter

екземпляра цієї структури забезпечує централізований доступ до параметрів поточного LoRa-пакета з різних модулів програмного проєкту, що спрощує обробку, логування та подальшу передачу даних на сервер.

```
// Структура для збереження параметрів прийнятого LoRa-повідомлення
struct loramessage
{
    String type;          // Тип повідомлення
    String counter;       // Лічильник / номер пакета
    String port;          // Номер порту
    String rssi;          // Рівень сигналу (RSSI)
    String snr;           // Відношення сигнал/шум (SNR)
    String length;        // Довжина корисного навантаження
    String payload;       // Корисні дані повідомлення
    String freq;          // Робоча частота передавання
    String data;          // Додаткові службові/корисні дані
    String previous_type; // Тип попереднього обробленого повідомлення
};
// Оголошення глобального екземпляра структури, визначеного в іншому модулі
extern struct loramessage message;
```

Рисунок 3.9 – Оголошення структури loramessage для зберігання параметрів LoRa-повідомлень

Крім того контролер має можливість отримувати вхідні команди – зокрема щодо перевірки поточного часу та встановлення нових порогових значень контрольованих параметрів, як показано на рис. 3.10. У наведеному фрагменті коду реалізовано функцію LoRaMain(), яка викликається після приймання пакета від шлюзу. Спочатку перевіряється тип отриманого повідомлення, і лише у випадку, коли він має значення "LRRECV", тобто є справжнім прийнятим пакетом, виконується подальша обробка. Перші два символи поля payload інтерпретуються як код команди, який перетворюється на числове значення і використовується в операторі switch для вибору потрібної гілки алгоритму. Перед виконанням команд у послідовний порт виводиться час отримання повідомлення, сформований на основі показів внутрішнього таймера, а також вміст корисного навантаження, що полегшує налагодження роботи системи. Якщо код команди дорівнює нулю, пристрій надсилає у відповідь поточний час у HEX-форматі; у випадку команди з кодом 1 відбувається оновлення граничних

Created by Universal Document Converter

значень контрольованого параметра функцією set_new_limit(), після чого на сервер повертається підтвердження з оновленими даними. Після завершення обробки прийнятого пакета та чергового циклу опитування датчиків пристрій надсилає результати на сервер і переходить у режим глибокого сну, забезпечуючи надзвичайно низьке споживання струму до наступного сеансу передавання даних.

```
// Головна функція обробки отриманих LoRa-повідомлень
void LoRaMain()
{
    // 1. Перевіряємо, чи тип отриманого повідомлення - "LRRECV" (прийнятий пакет)
    if (message.type == "LRRECV")
    {
        // 2. Перетворюємо перші два символи payload у байт - код команди
        int zerobyte = str_to_byte(message.payload.substring(0, 2));

        // 3. Виводимо час отримання повідомлення в серійний монітор
        Serial.println("Message received at " + timer.hour_DEC + ":" +
                       timer.minute_DEC + ":" + timer.second_DEC + " " +
                       timer.dayOfMonth + "." + timer.month + "." + timer.year);

        // 4. Виводимо вміст корисного навантаження для налагодження
        Serial.println(message.payload);

        // 5. Обробляємо команду залежно від значення першого байта
        switch (zerobyte)
        {
            case 0:
                // Надсилаємо відповідь з поточним часом у HEX-форматі
                send_lora_message("10",
                                  (timer.hour_HEX +
                                   timer.minute_HEX +
                                   timer.second_HEX));
                break;

            case 1:
                // Оновлюємо граничні значення та надсилаємо підтвердження
                set_new_limit(message.payload);
                send_lora_message("11", message.payload.substring(2, 4));
                break;
        }
    }
}
```

Рисунок 3.10 – Код функції обробки вхідних повідомлень від сервера LoRaWAN

Created by Universal Document Converter

На рисунку 3.11 показано зовнішні компоненти, які залишаються активними, коли контролер ESP32 переходить у режим глибокого сну, зокрема процесор з наднизьким енергоспоживанням та пам’ять RTC. У звичайному режимі споживання струму становить орієнтовно від 160 до 260 мА, а в деяких випадках може досягати 790 мА. Використовуючи режим глибокого сну, контролер може зменшити рівень енергоспоживання у 1000 разів.

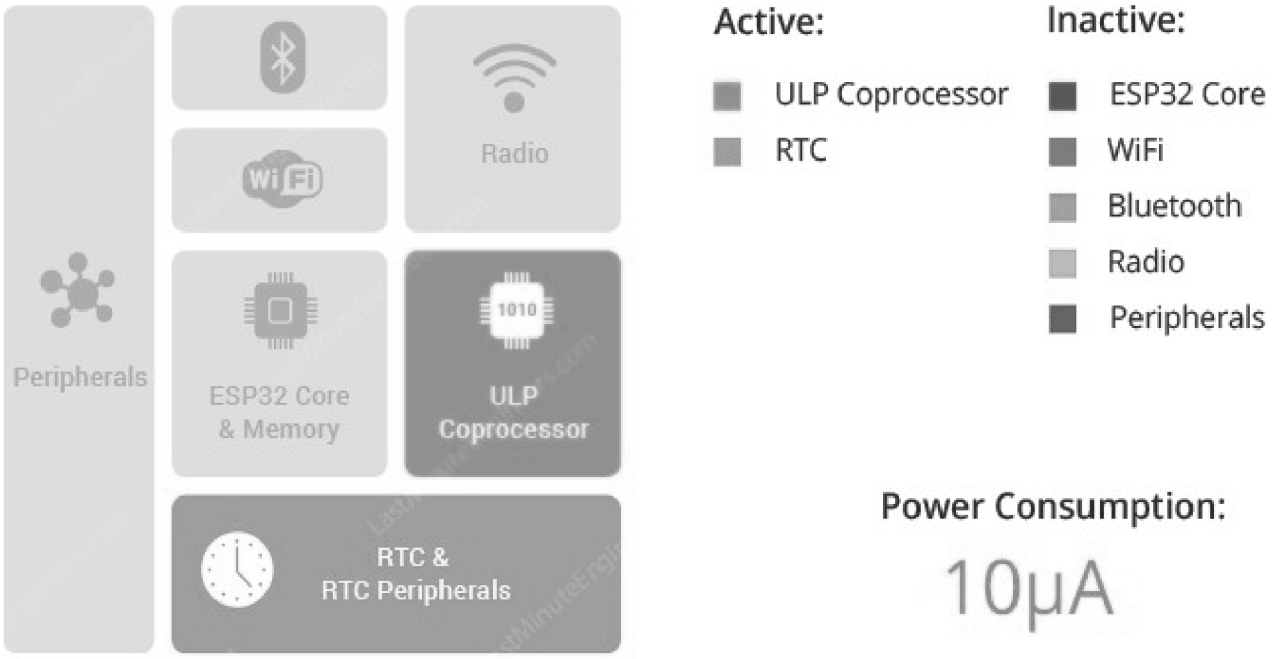

Рисунок 3.11 – Схема активних модулів контролера в режимі глибокого сну

Для фільтрації значень параметрів, зчитаних із датчика, було використано алгоритм цілочисельної фільтрації під назвою «медіанний фільтр». Цей метод використовує зв’язані списки. Його принцип роботи полягає в тому, що сортування масиву, видалення найстарішого значення та вставлення найновішого значення суттєво не змінюють результат сортування масиву. Тому цей алгоритм добре працює, особливо під час фільтрації великих масивів.

Оскільки він потребує зберігання значень і вказівників, цей метод може споживати значну кількість пам’яті. Водночас це дуже ефективний алгоритм, значно швидший за підходи, що ґрунтуються на повторному сортуванні. Його реалізацію подано на рисунку 3.12.

Created by Universal Document Converter

```
// Ковзний медіанний фільтр для згладжування вимірюваних значень
#define MEDIAN_FILTER_SIZE 5          // Розмір вікна фільтра (кількість зразків)
#define STOPPER 0                     // Значення, менше за будь-яке реальне вимірювання
uint16_t median_filter(uint16_t datum)
{
    struct pair                       // Елемент зв'язаного списку відсортованих значень
    {
        struct pair *point;           // Вказівник на наступний елемент у списку
        uint16_t value;               // Значення, що сортується
    };
    static struct pair buffer[MEDIAN_FILTER_SIZE] = {0}; // Кільцевий буфер елементів фільтра
    static struct pair *datapoint = buffer;              // Поточна позиція в буфері
    static struct pair small = {NULL, STOPPER};          // «Стопер» наприкінці списку
    static struct pair big = {&small, 0};                // Голова списку (найбільший елемент)
    struct pair *successor;           // Наступник заміненого елемента
    struct pair *scan;                // Поточний елемент при проході по списку
    struct pair *scanold;             // Попередній елемент при проході
    struct pair *median;              // Вказівник на медіану
    uint16_t i;
    if (datum == STOPPER) datum = STOPPER + 1;           // Забороняємо використання значення STOPPER
    if (++datapoint - buffer >= MEDIAN_FILTER_SIZE)      // Рух по кільцевому буферу
        datapoint = buffer;
    datapoint->value = datum;         // Записуємо нове значення у буфер
    successor = datapoint->point;     // Запам'ятовуємо наступник старого значення
    median = &big;                    // Початково медіана — голова списку
    scanold = NULL;                   // Початково попередній елемент відсутній
    scan = &big;                      // Починаємо сканування з голови списку
    if (scan->point == datapoint)     // Окремий випадок: видаляємо перший елемент у списку
        scan->point = successor;
    scanold = scan;                   // Зберігаємо попередній елемент
    scan = scan->point;               // Переходимо до наступного елемента
    for (i = 0; i < MEDIAN_FILTER_SIZE; ++i)             // Основний цикл проходу по відсортованому списку
    {
        if (scan->point == datapoint)                    // Якщо натрапили на старе значення
            scan->point = successor;                     // Виводимо його зі списку
        if (scan->value < datum)                         // Вставляємо нове значення у відсортовану позицію
        {
            datapoint->point = scanold->point;
            scanold->point = datapoint;
            datum = STOPPER;                             // Позначаємо, що вставка виконана
        }
        median = median->point;                          // Зсуваємо вказівник на медіану після непарного елемента
        if (scan == &small) break;                       // Кінець списку
        scanold = scan;                                  // Переходимо до наступного елемента
        scan = scan->point;
        if (scan->point == datapoint)                    // Обробка парного елемента списку
            scan->point = successor;
        if (scan->value < datum)                         // Повторна перевірка для парного елемента
        {
            datapoint->point = scanold->point;
            scanold->point = datapoint;
            datum = STOPPER;
        }
    }
    return median->value;                                // Повертаємо поточне медіанне значення
}
```

Рисунок 3.12 – Фрагмент коду реалізації медіанного фільтру для згладжування результатів вимірювань

Created by Universal Document Converter

3.3 Імітаційне моделювання та тестування функціонування системи

Для тестування розробленої системи моніторингу стану навколишнього середовища проведено моделювання її роботи. На етапі моделювання використовується протокол MQTT для зв'язку між вузлами та сервером, а також інтерфейс Wi-Fi для бездротової передачі пакетів від кінцевих пристроїв до серверної частини. При цьому панель керування як для тестового, так і для робочого середовищ може мати однаковий вигляд – змін зазнає лише потік даних Node-RED, який отримує значення від сервера.

Апаратну частину зібрано у мінімалістичному варіанті, як показано на рисунку 3.13. Макет включає контролер ESP32 (підключений до комп'ютера через USB для перевірки роботи моделі через Wi-Fi) та комплексний датчика стану навколишнього середовища BME680.

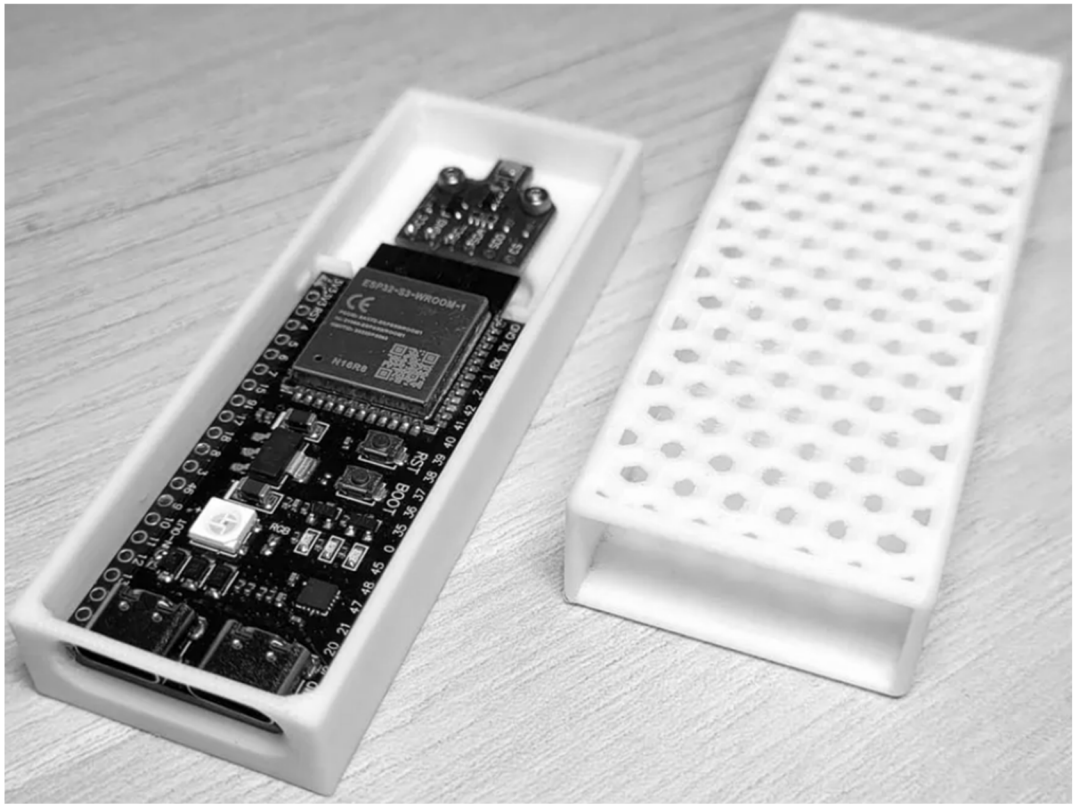

Рисунок 3.13 – Зовнішній вигляд граничного пристрою системи віддаленого моніторингу

Created by Universal Document Converter

На рисунку 3.14 наведено функції setup() та loop(), які по суті ініціалізують роботу бездротового контролера. Після ввімкнення пристрою керування передається до функції setup(), яка ініціалізує інтерфейси та з'єднання, а також підключає всі встановлені периферійні пристрої.

Для зв'язку з терміналом послідовний порт ініціалізується зі швидкістю 115200 бод. Для підключення до мережі Wi-Fi використовується метод begin() об'єкта класу WiFi, у якому задаються параметри мережі.

Для встановлення з'єднання з сервером MQTT викликається метод ініціалізації об'єкта client.begin(), до якого приєднується функція зворотного виклику для обробки вхідних повідомлень від сервера.

```
void setup() {
    Serial.begin(115200);                  // Старт порту для налагодження
    WiFi.begin(ssid, pass);                // Підключення до Wi-Fi

    client.begin(BROKER_URI, net);         // Ініціалізація MQTT-клієнта
    client.onMessage(messageReceived);     // Обробник вхідних повідомлень

    connect();                             // Підключення до брокера
    sleep_setup();                         // Налаштування енергозбереження
}

void loop() {
    client.loop();                         // Обробка MQTT-подій

    if (!client.connected()) {             // Перепідключення за потреби
        connect();
    }

    bme680_routine();                      // Опитування датчика BME680
}
```

Рисунок 3.14 – Функції керування роботою граничного вузла

У тілі функції, що відповідає за підключення до сервера, спочатку перевіряється наявність з'єднання з мережею Wi-Fi, а потім встановлюється прямий зв'язок із заданою темою (підпискою) MQTT, як показано на рис. 3.15.

Created by Universal Document Converter

```
void connect() {
    Serial.print("checking wifi...");              // Перевірка підключення до Wi-Fi

    while (WiFi.status() != WL_CONNECTED) {        // Чекаємо, поки Wi-Fi не з'єднається
        Serial.print(".");
        delay(1000);
    }

#ifdef DEBUG
    Serial.println("IP address: ");                // Вивід IP-адреси в режимі налагодження
    Serial.println(WiFi.localIP());
#endif

    Serial.println("connecting...");               // Спроба підключення до MQTT-брокера
    while (!client.connect("monitoring-esp32", "", "")) {
        Serial.print(".");
        client.disconnect();                       // Очистка з'єднання перед новою спробою
        delay(2000);
    }

    Serial.println("connected!");                  // Успішне підключення до брокера
    client.subscribe("monitoring-esp32/#");        // Підписка на потрібні MQTT-топіки
}
```

Рисунок 3.15 – Функції connect() для підключення до MQTT-брокера

Оскільки підключення здійснюється до загальнодоступного проксі-сервера, авторизація не потрібна. Для захищених MQTT-застосунків, що використовують TLS і потребують авторизації, облікові дані можна вказати як додатковий параметр після теми в методі connect().

Коренева тема для підписки на події – monitoring-esp32/, тому всі дочірні теми підписуються шляхом додавання символу # після кореневої теми.

Після виконання функції setup() потік програми більше не повертається до неї, а переходить до функції циклу loop() (рис. 3.14), яка виконується багаторазово у циклічному режимі.

Функція циклу забезпечує зв'язок із проксі-сервером MQTT та опитування підключених датчиків. Вона опитує датчик, збирає значення параметрів, надсилає їх на проксі-сервер і формує налагоджувальний вивід у послідовний термінал, як показано на рисунку 3.16.

Created by Universal Document Converter

```
void bme680_routine() {
    // Періодичний запуск рутини за таймером
    if (millis() - lastMillis > TIMEOUT) {
        lastMillis = millis();              // Оновлення часу останнього вимірювання

        bme680_set_params();                // Зчитування/оновлення параметрів датчика

        Serial.print(F("Reading completed at "));
        Serial.println(millis());           // Час завершення вимірювання

        Serial.print(F("Temperature = "));
        Serial.print(bme.temperature);
        Serial.println(F(" *C"));

        Serial.print(F("Pressure = "));
        Serial.print(bme.pressure);
        Serial.println(F(" hPa"));

        Serial.print(F("Humidity = "));
        Serial.print(bme.humidity);
        Serial.println(F(" %"));

        Serial.print(F("Gas = "));
        Serial.print(bme.gas_resistance);
        Serial.println(F(" kOhms"));        // Вивід усіх параметрів у монітор порту

        bme680_publish_params();            // Публікація даних (MQTT чи інший канал)
    }
}
```

Рисунок 3.16 – Функція bme680_routime() для опитування датчика та зчитування виміряних параметрів

Обмін даними з MQTT-брокером має здійснюватися в текстовому форматі, тому отримані з датчика значення спочатку зберігаються в числовому вигляді у відповідних полях об’єкта bme, а перед публікацією конвертуються у потрібний тип. Для цього використано тип String, оскільки саме так Arduino IDE типово інтерпретує символьні масиви та коректно передає їх до функції client.publish(), яка очікує текстовий буфер. Фактично на кожному кроці виконання функції числове значення температури, тиску, вологості чи якості повітря перетворюється на рядок методом String(...). На рис. 3.17 показано реалізацію функції надсилання даних bme680_publish_params(), у якій для кожного параметра датчика BME680 використовується окрема тема («monitoring-esp32/temperature», «monitoring-esp32/pressure», «monitoring-esp32/humidity», «monitoring-esp32/gas-resistance»). Такий підхід дає змогу розділити інформаційні потоки за фізичними величинами, спрощує підписку клієнтів та полегшує подальшу обробку показників на стороні сервера або в хмарі.

Created by Universal Document Converter

```
// Функція для публікації параметрів датчика BME680 до MQTT-брокера
void bme680_publish_params()
{
    // Надсилання значення температури на топік "monitoring-esp32/temperature"
    client.publish("monitoring-esp32/temperature", String(bme.temperature));

    // Надсилання значення тиску на топік "monitoring-esp32/pressure"
    client.publish("monitoring-esp32/pressure", String(bme.pressure));

    // Надсилання значення вологості на топік "monitoring-esp32/humidity"
    client.publish("monitoring-esp32/humidity", String(bme.humidity));

    // Надсилання значення опору газового сенсора на топік "monitoring-esp32/gas-resistance"
    client.publish("monitoring-esp32/gas-resistance", String(bme.gas_resistance));
}
```

Рисунок 3.17 – Функція публікації телеметричних даних на MQTT-брокер

Сервер Node-RED запущено в терміналі Windows PowerShell, а його потік даних підключено до публічного брокера MQTT, як показано на рисунку 3.18.

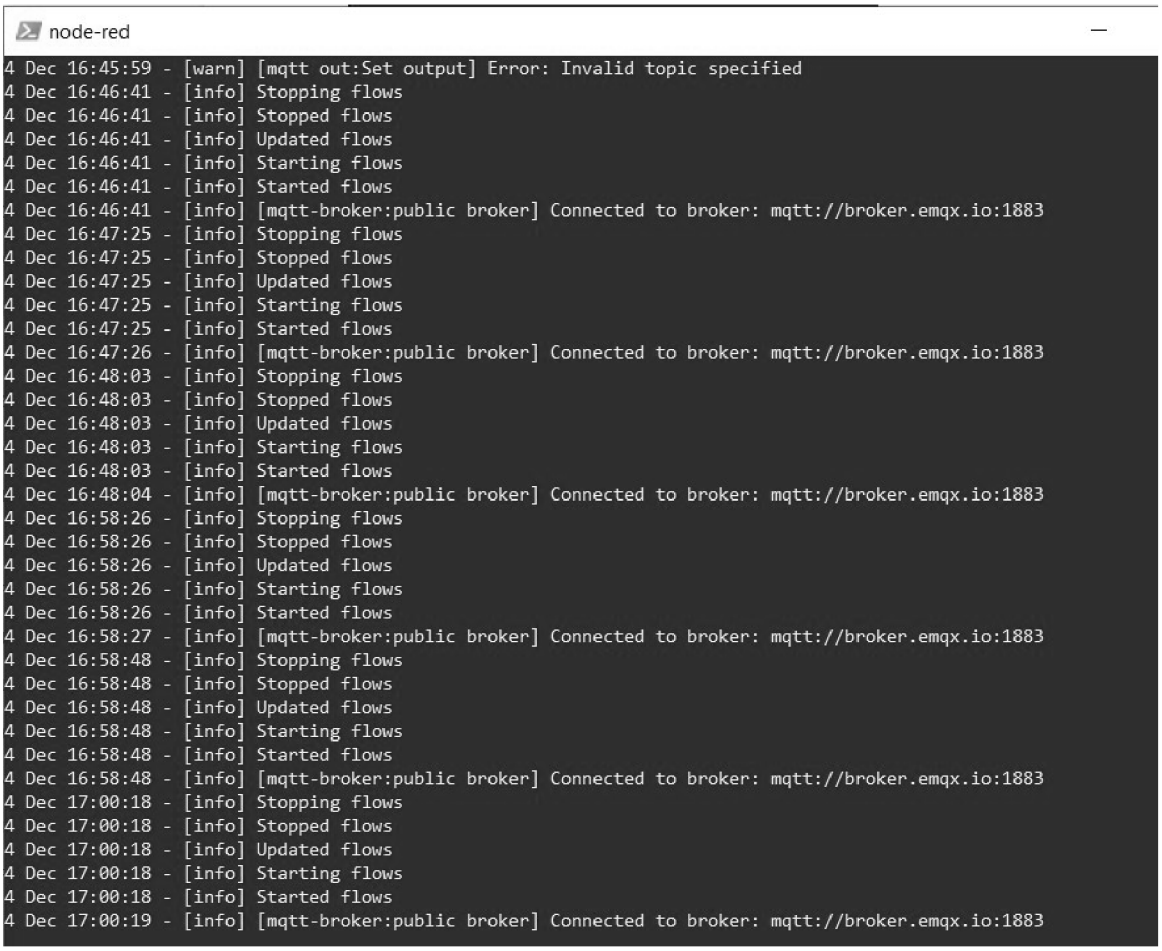

Рисунок 3.18 – Консольний лог роботи сервера Node-RED під час обробки повідомлень

Created by Universal Document Converter

Мікроконтролер публікує результати вимірювань на брокерському сервері за тією самою адресою, а доставку повідомлень можна перевірити за допомогою MQTT-клієнта MQTT Explorer. На рисунках 3.19 показано передачу даних за відповідними темами.

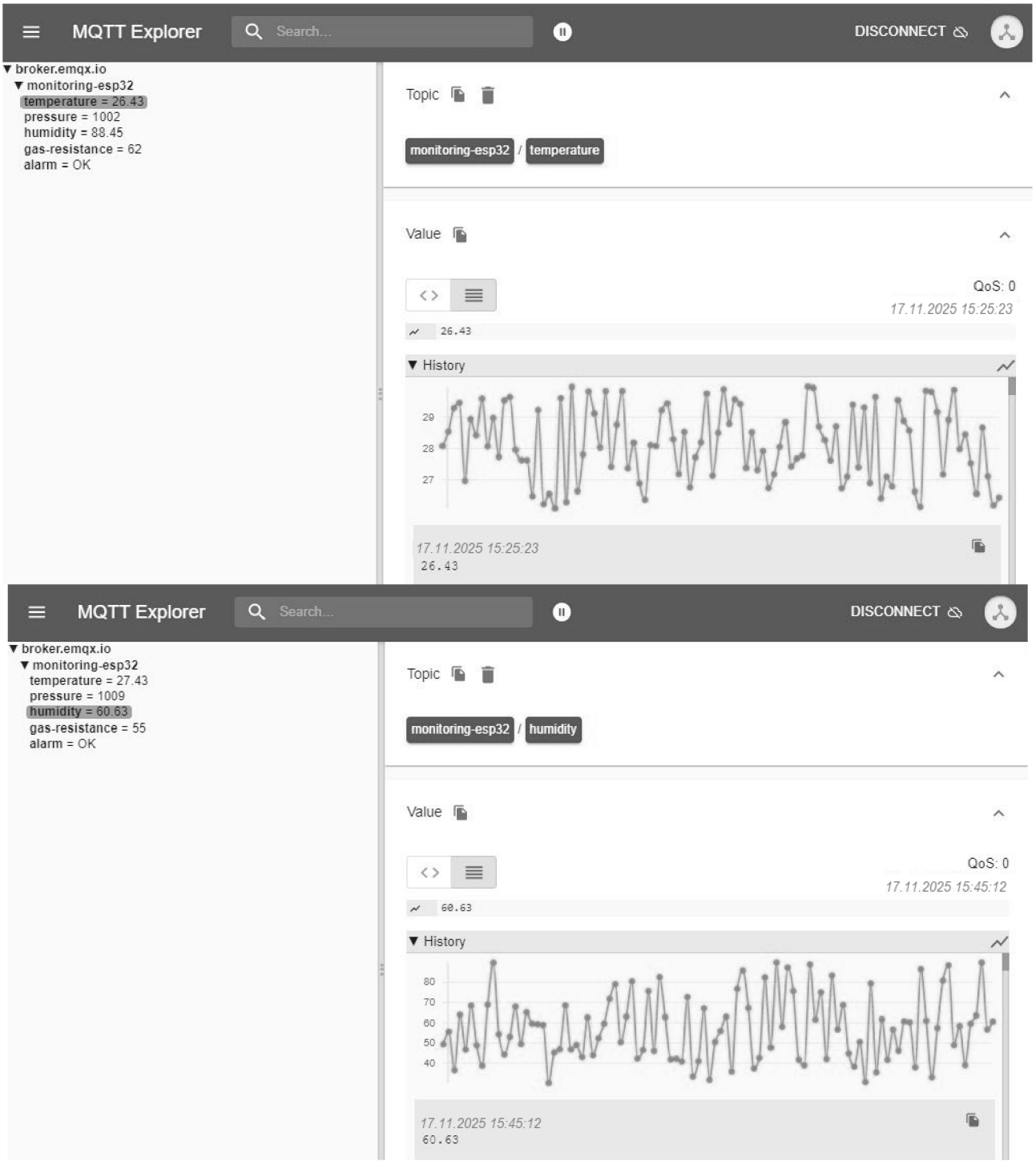

Рисунок 3.20 – Приклад отриманих із пристрою значень вологості з топіка “temperature” та “humidity”

Created by Universal Document Converter

На рисунку 3.20 подано процес, розроблений для обміну інформацією:

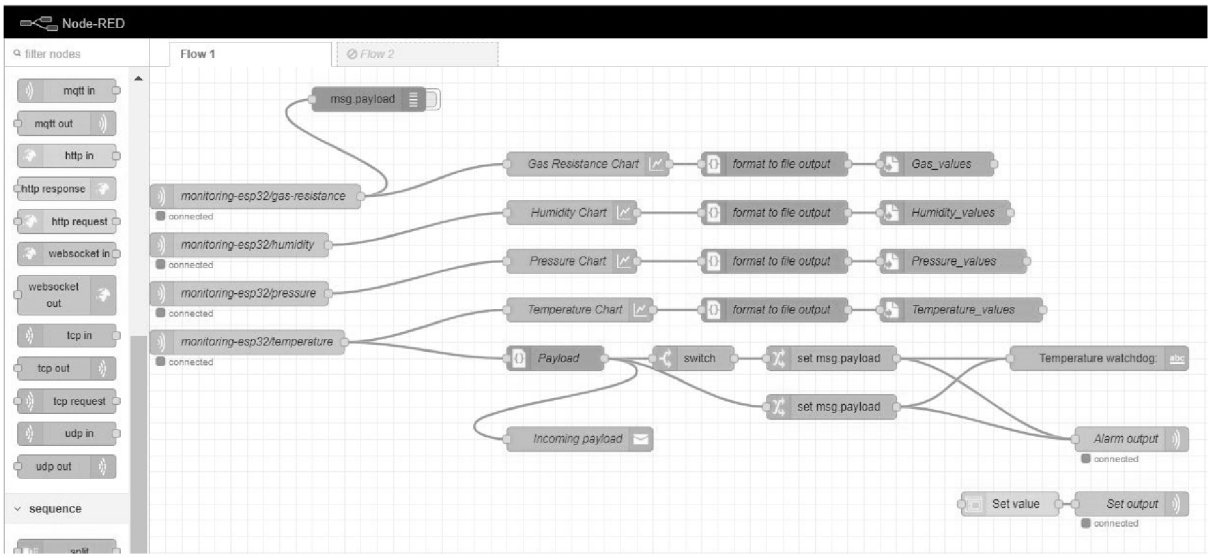

Рисунок 3.20 – Схема потоків Node-RED для обробки окремих параметрів середовища

Вхідний вузол MQTT monitoring-esp32/... отримує та надсилає повідомлення безпосередньо від брокера до відповідного процесу Node-RED для їх подальшої обробки в потоках даних. На основі цих повідомлень система формує окремі графіки для відображення вхідних параметрів: температури, тиску, вологості і якості повітря, що дозволяє спостерігати зміну кожного з них.

Дані, які використовуються для побудови графіків, додатково архівуються у форматі JSON до локального сховища пристрою, на якому запущено Node-RED, що забезпечує можливість подальшого аналізу історичних значень та резервного зберігання. На рисунку 3.21 наведено приклад вмісту такого архівного файлу з послідовністю записів виміряних параметрів.

Конфігурацію архівного вузла подано на рисунку 3.22. У ній, зокрема, задається шлях до файлу, режим дописування та формат збереження записів. Важливо, щоб шляхи до файлів були зазначені як абсолютні, інакше вони будуть інтерпретовані як відносні щодо каталогу, з якого запущено Node-RED, що може призвести до створення файлів у неочікуваному місці або до помилок під час зчитування архіву.

Created by Universal Document Converter

```
[
    {
        "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 3.21 – Структура архівного файлу з часовими мітками та виміряними значеннями параметрів

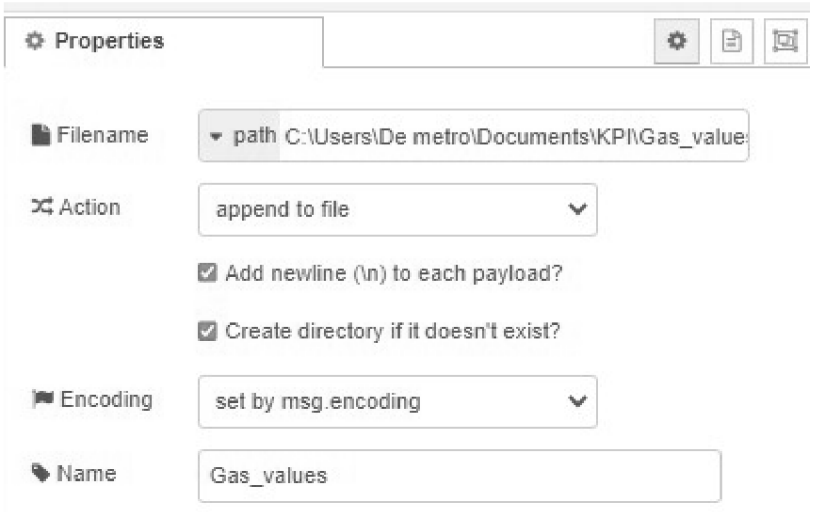

Рисунок 3.22 – Налаштування вузла архівування даних у середовищі Node-RED

Температура та якість повітря контролюються, щоб запобігти перевищенню встановлених порогових значень. У разі перевищення встановлених значень стан вікна контролю температури змінюється, як показано на рисунку 3.23.

Created by Universal Document Converter

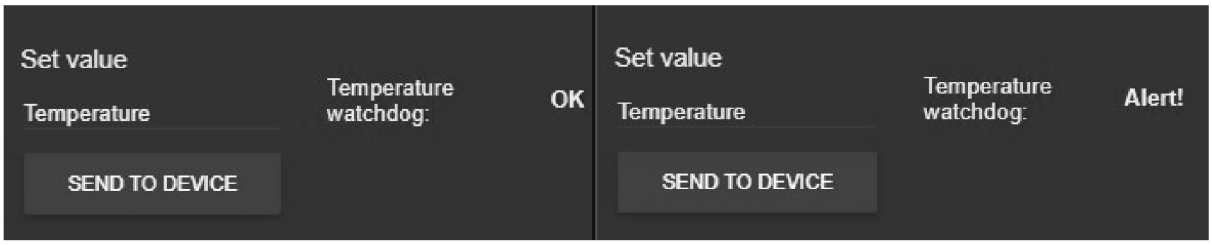

Рисунок 3.23 – Зміна станів Watchdog за нормальної роботи та при перевищенні граничного порогу температури

Критичні значення також можна задавати через панель візуалізації, натиснувши кнопку «Надіслати на пристрій», щоб передати введене значення у відповідне поле. Головне вікно візуалізації моделі системи віддаленого моніторингу стану навколишнього середовища показано на рисунку 3.24.

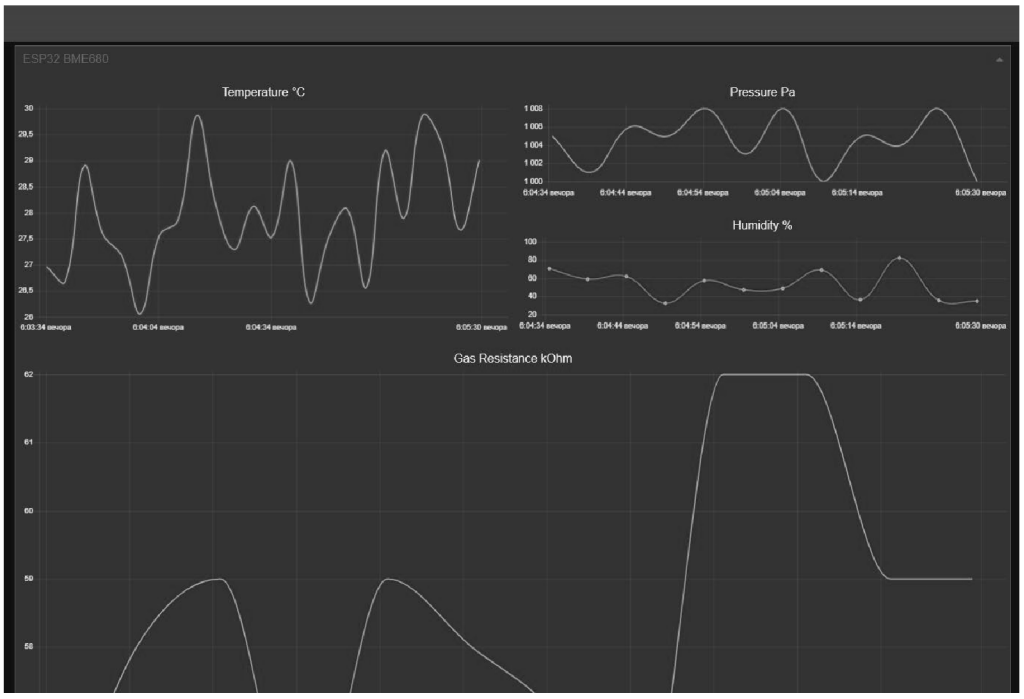

Рисунок 3.24 – Інтерфейс панелі візуалізації параметрів системи моніторингу

Кожен вимірюваний параметр у системі відображається на окремій діаграмі (рис. 3.25), яка підтримує масштабування та перелистування для перегляду історії наявних записів.

Created by Universal Document Converter

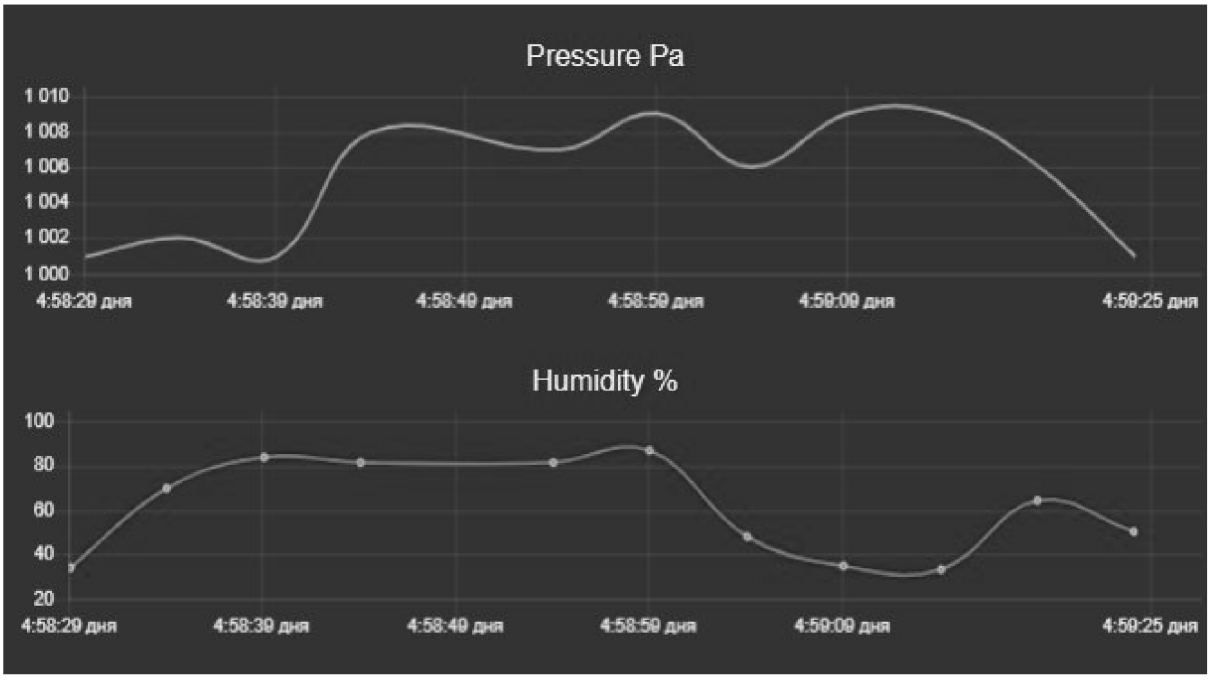

Рисунок 3.25 – Графіки динаміки тиску та вологості повітря

У масштабованій системі моніторингу пристрої відображаються в послідовному списку, і кожен з них можна розгортати для перегляду. Завдяки цьому оператор має змогу швидко переключатися між різними вузлами мережі, порівнювати їхні поточні показники та аналізувати динаміку змін у часі. За потреби користувач може змінювати порогові значення для окремих датчиків, одразу оцінюючи результат на відповідних графіках і у вікні контролю стану. Передбачена можливість використання даних історії вимірювань для формування звітів, виявлення довгострокових тенденцій та налаштування більш ефективних сценаріїв реагування. Сукупність цих функцій забезпечує зручний моніторинг параметрів навколишнього середовища, своєчасне виявлення відхилень та підвищує загальну надійність і інформативність системи. Таким чином, розроблена панель візуалізації є важливим інструментом для практичної експлуатації системи дистанційного екологічного моніторингу.

Created by Universal Document Converter

РОЗДІЛ 4
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Аналіз умов праці при розробці та експлуатації інформаційної системи

Розробка інформаційної системи дистанційного моніторингу стану навколишнього середовища передбачає виконання робіт, пов'язаних із програмуванням, тестуванням, налагодженням апаратного та програмного забезпечення, а також із подальшою експлуатацією системи. Основна діяльність виконується у приміщеннях, обладнаних комп'ютерною технікою, а також на відкритому повітрі під час встановлення сенсорних модулів і перевірки роботи мережевих пристроїв.

Умови праці розробників та обслуговуючого персоналу належать до категорії робіт з переважно розумовим характером навантаження. Основними факторами, які можуть впливати на працівників у процесі виконання робіт, є: недостатнє освітлення, надмірна яскравість екранів, електромагнітне випромінювання від моніторів і комп'ютерів, шум від обладнання, недостатній повітрообмін у приміщенні, а також тривале перебування у статичній позі. Такі умови можуть спричиняти зорове стомлення, підвищену втому, зниження концентрації уваги та ризик розвитку професійних захворювань.

Особливу увагу необхідно приділяти організації робочого місця. Робоче місце користувача комп'ютера повинно відповідати вимогам ДСанПіН 3.3.2.007-98 та ДСТУ ISO 9241-5:2003, які регламентують ергономічні вимоги до робочого місця з відеотерміналом. Висота столу має становити 720-750 мм, а відстань від очей користувача до екрана монітора – 500-700 мм. Монітор слід розташовувати таким чином, щоб верхній край екрана був на рівні очей. Робоче крісло має забезпечувати регулювання висоти сидіння та нахилу спинки.

Created by Universal Document Converter

Важливо також забезпечити належні мікрокліматичні умови. Температура повітря в приміщенні повинна підтримуватися в межах 18–24 °С, відносна вологість – 40–60 %, швидкість руху повітря – не більше 0,1 м/с. Освітлення має бути рівномірним, із рівнем освітленості робочої поверхні не менше 300 лк. Джерела світла бажано розміщувати так, щоб уникати відблисків на поверхні екрана.

У процесі експлуатації системи можливе виконання робіт на відкритому повітрі під час встановлення сенсорів або діагностики стану обладнання. У таких випадках працівники можуть зазнавати впливу метеорологічних факторів – високої або низької температури, вітру, опадів, підвищеної вологості. Для зменшення ризиків необхідно забезпечити персонал відповідним спецодягом, засобами індивідуального захисту та переносним інструментом, який відповідає вимогам електробезпеки.

Отже, аналіз умов праці показує, що основними небезпечними та шкідливими факторами при розробці та експлуатації системи є вплив електромагнітних випромінювань, напруження зору, статичне навантаження, недотримання ергономічних вимог, а також несприятливі погодні умови під час монтажних робіт. Для забезпечення безпечних і комфортних умов праці слід організувати раціональний режим праці та відпочинку, проводити профілактичні медичні огляди, а також регулярно перевіряти стан робочих місць і технічного обладнання.

4.2 Ідентифікація небезпечних та шкідливих виробничих факторів

У процесі розробки, налаштування та експлуатації інформаційної системи дистанційного моніторингу стану навколишнього середовища працівники можуть зазнавати впливу низки небезпечних і шкідливих факторів, що впливають на їхнє здоров’я, працездатність і безпеку. Виявлення та оцінка таких

Created by Universal Document Converter

факторів є важливою складовою системи управління охороною праці на підприємстві.

До небезпечних виробничих факторів під час роботи з апаратною частиною системи належать:

– ураження електричним струмом при роботі з електрообладнанням, блоками живлення, контролерами та датчиками;

– ризик короткого замикання або займання внаслідок несправності електропроводки чи перевантаження мережі;

– можливість падіння з висоти або травмування при монтажі сенсорів у польових умовах (на стовпах, фасадах будівель тощо);

– механічні пошкодження рук чи очей під час використання інструментів для кріплення елементів системи;

– небезпека ураження блискавкою під час роботи на відкритій місцевості або при обслуговуванні обладнання на висоті.

До шкідливих виробничих факторів, що впливають на розробників і користувачів програмної частини системи, належать:

– підвищене зорове навантаження при тривалій роботі з монітором;

– статичне навантаження на опорно-руховий апарат;

– вплив електромагнітного випромінювання від дисплеїв, комп’ютерної техніки, маршрутизаторів та іншого периферійного обладнання;

– підвищений рівень шуму від вентиляційних систем, комп’ютерів або серверів;

– нераціональні параметри мікроклімату в приміщенні.

Окрему групу становлять психофізіологічні фактори, серед яких розумова перевтома, емоційна напруга, стресові ситуації при виконанні складних програмних завдань або в разі виникнення технічних збоїв системи. Тривале перевантаження може знижувати концентрацію уваги та підвищувати ризик помилок, що впливають як на якість роботи, так і на безпеку працівника.

Created by Universal Document Converter

При монтажі системи в польових умовах можуть діяти природні фактори: низька або висока температура, дощ, вітер, підвищена вологість, слизька поверхня, укус комах. Такі умови створюють додаткові ризики для життя і здоров'я персоналу, тому необхідно використовувати засоби індивідуального захисту, зокрема захисні рукавички, діелектричне взуття, каски, окуляри та спецодяг.

Усі виявлені небезпечні та шкідливі фактори підлягають контролю, а їхній рівень має не перевищувати гранично допустимих значень, установлених чинними нормативно-правовими актами з охорони праці:

– ДСанПіН 3.3.2.007-98 «Державні санітарні норми і правила роботи з відеодисплейними терміналами ЕОМ»;

– ДСТУ EN 50110-1:2015 «Експлуатація електроустановок»;

– НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електроустановок споживачів».

Отже, ідентифікація потенційних небезпечних і шкідливих виробничих факторів дає змогу своєчасно розробити ефективні заходи для їх усунення або мінімізації. Це забезпечує безпечні умови праці розробників і технічного персоналу, підвищує надійність експлуатації системи та знижує ризик виникнення аварійних ситуацій.

4.3 Заходи з охорони праці під час розробки та експлуатації системи

Забезпечення безпечних умов праці при створенні та використанні інформаційної системи дистанційного моніторингу стану навколишнього середовища є одним із головних завдань організації виробничого процесу. Комплекс заходів з охорони праці повинен охоплювати як етап розробки

Created by Universal Document Converter

програмного забезпечення, так і етапи монтажу, налагодження та експлуатації технічних компонентів системи.

У процесі розробки програмної частини основну увагу необхідно приділяти організації робочого місця працівника, який працює за комп'ютером. Робоча поверхня повинна бути чистою та вільною від сторонніх предметів, а розташування обладнання має забезпечувати зручність, безпечну позу і можливість зміни положення тіла. Висота столу й стільця, а також положення монітора, клавіатури й миші повинні відповідати вимогам ергономіки. Для запобігання перевтомі працівникам рекомендується дотримуватися режиму праці та відпочинку – робити короткі перерви через кожні 1–2 години роботи, виконувати вправи для очей і рук, а також змінювати положення тіла.

У приміщеннях, де розташоване комп'ютерне обладнання, важливо забезпечити належні параметри мікроклімату. Оптимальна температура має становити від 18 до 24 °C, вологість – у межах 40–60 %, а швидкість руху повітря не повинна перевищувати 0,1 м/с. Система освітлення має бути комбінованою, із використанням природного та штучного світла. Не допускається утворення відблисків на поверхні моніторів або інших екранів. З метою зниження рівня шуму та вібрації необхідно регулярно проводити технічне обслуговування комп'ютерів, вентиляторів і блоків живлення.

Під час роботи з електричним і мережевим обладнанням необхідно дотримуватись правил електробезпеки. До експлуатації електроустановок допускаються лише особи, які пройшли відповідне навчання і мають кваліфікаційну групу з електробезпеки не нижче II. Усі електропристрої повинні бути заземлені, а кабелі – ізольовані та захищені від механічних пошкоджень. Використання несправних розеток, подовжувачів або вилок неприпустиме.

Особливі вимоги висуваються до працівників, які здійснюють монтаж або обслуговування сенсорних модулів системи в польових умовах. Для них необхідно передбачити використання спецодягу, діелектричного взуття, захисних рукавичок і касок. У разі виконання робіт на висоті або поблизу ліній електропередачі обов'язковим є застосування страхувальних поясів і

Created by Universal Document Converter

дотримання відстані безпеки. Під час грози або несприятливих погодних умов виконання зовнішніх робіт має бути припинене.

Організаційні заходи з охорони праці включають проведення вступного та періодичних інструктажів, навчання персоналу безпечним методам роботи, перевірку знань з охорони праці та пожежної безпеки, а також забезпечення працівників засобами колективного й індивідуального захисту. Адміністрація має контролювати дотримання вимог чинних нормативних документів, таких як ДСТУ ISO 45001:2019 «Системи управління охороною здоров'я та безпекою праці» та НПАОП 0.00-4.21-10 «Типове положення про навчання з питань охорони праці».

Таким чином, реалізація комплексу технічних, організаційних та профілактичних заходів дозволяє створити безпечні умови праці для всіх учасників процесу розробки й експлуатації інформаційної системи. Це сприяє підвищенню продуктивності, зниженню ризику професійних захворювань і травматизму, а також забезпечує відповідність вимогам законодавства України у сфері охорони праці.

4.4 Дії персоналу в разі виникнення надзвичайних ситуацій

Діяльність, пов'язана з розробкою та експлуатацією інформаційної системи дистанційного моніторингу стану навколишнього середовища, може супроводжуватись ризиками виникнення надзвичайних ситуацій техногенного або природного характеру. До таких ситуацій належать пожежі, короткі замикання, вибухи електрообладнання, витоки небезпечних речовин у зоні встановлення сенсорів, аварії систем електроживлення, а також стихійні лиха – сильні зливи, повені, грози чи буревії. Для мінімізації наслідків подібних подій необхідно чітко визначити порядок дій персоналу, спрямований на збереження життя і здоров'я людей та запобігання матеріальним збиткам.

Created by Universal Document Converter

У разі виникнення пожежі працівник, який першим виявив загоряння, повинен негайно повідомити про це відповідальну особу та службу охорони праці, а також викликати пожежно-рятувальну службу за телефоном 101. Після цього необхідно, не ризикуючи власним життям, розпочати гасіння вогню за допомогою первинних засобів пожежогасіння – вогнегасників або пожежних покривал. Електричне обладнання, що живиться від мережі, слід попередньо відключити. Якщо пожежу ліквідувати неможливо, працівники повинні організовано залишити приміщення через найближчий евакуаційний вихід і зібратися у визначеному безпечному місці.

У ситуації, коли виникає загроза ураження електричним струмом, персонал повинен негайно відключити електроживлення за допомогою рубильника або автоматичного вимикача. Якщо людина потрапила під дію електричного струму, необхідно безпечно звільнити її від дії струму та надати першу домедичну допомогу, викликавши медичну службу. Усі роботи з усунення несправностей можуть виконуватися лише після повного зняття напруги та перевірки відсутності живлення.

Під час виконання зовнішніх робіт можливе виникнення небезпечних ситуацій, спричинених погодними умовами. При наближенні грози, сильному вітрі або зливі працівники повинні негайно припинити монтажні чи ремонтні роботи, особливо якщо вони проводяться на висоті або поблизу електричних ліній. Необхідно відійти на безпечну відстань від металевих конструкцій і дерев та укритися в захищеному приміщенні.

У разі виникнення надзвичайної ситуації природного характеру, зокрема повені чи землетрусу, дії персоналу повинні відповідати затвердженим інструкціям цивільного захисту. Працівники зобов’язані виконувати розпорядження керівника об’єкта або уповноважених служб, зберігати спокій, допомагати евакуювати людей, а також за можливості забезпечити збереження технічного обладнання.

Created by Universal Document Converter

РОЗДІЛ 5
ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ

5.1 Формування та обґрунтування концепції стартап-проєкту

Одним із завдань кваліфікаційної роботи є маркетинговий аналіз стартап-проєкту, що дозволяє оцінити можливість його виведення на ринок, визначити цільові сегменти споживачів і очікувані вигоди. Проєкт передбачає створення системи комплексного моніторингу стану навколишнього середовища на основі розподіленої мережі датчиків, які бездротовими каналами передають дані до центрального вузла для обробки й візуалізації.

Пропонується апаратно-програмний комплекс для збору інформації з віддалених або важкодоступних територій. Дані оцифровуються, шифруються та передаються захищеними каналами. Використання LoRaWAN забезпечує оптимальний баланс між далекістю зв'язку, енергоспоживанням пристроїв і витратами на побудову та обслуговування мережі. Для користувачів це означає можливість отримувати екологічні й технологічні показники майже в реальному часі, здійснювати постійний нагляд за об'єктами у віддаленій місцевості та своєчасно реагувати на відхилення. Додатковою перевагою є розгортання власної захищеної IoT-мережі без залежності від сторонніх операторів.

Порівняльний аналіз із пропрієтарними протоколами Sigfox та NB-IoT показав конкурентоспроможність проєкту. За п'ятибальною шкалою вартість розробки оцінено у чотири бали (для Sigfox – два, для NB-IoT – три), що свідчить про відносну економічність рішення. Рівень безпеки досягає п'яти балів завдяки використанню сучасних механізмів шифрування й розмежування доступу. Покриття мережі оцінено у чотири бали: воно менше, ніж у глобальних мереж Sigfox, але достатнє для більшості сценаріїв, де пріоритетом є автономність.

Created by Universal Document Converter

Остання, завдяки енергоефективним радіомодулям та оптимізованим протоколам, отримала максимальну оцінку – п’ять балів.

Узагальнюючи, до сильних сторін проєкту належать прийнятна вартість, високий рівень інформаційної безпеки та значна автономність сенсорних вузлів. Відносним недоліком є менш масштабне покриття порівняно з глобальними операторами LPWAN, однак це компенсується можливістю розгортання власної інфраструктури та адаптації системи до потреб конкретного замовника. Отже, запропонована система моніторингу має реальний потенціал для успішного виходу на ринок і подальшого масштабування.

5.2 Оцінка технологічної реалізовності та ризиків впровадження проєкту

У межах проєкту розглядається ідея створення комплексу моніторингу навколишнього середовища з дистанційною системою збору інформації. Для реалізації цієї ідеї проаналізовано низку технологій, придатних для формування комунікаційної, обчислювальної та енергетичної підсистем системи. Як основні засоби бездротового зв’язку розглянуто протоколи LoRaWAN і SigFox, як обчислювальну платформу – мікроконтролери архітектури RISC-V MCU, а як джерело живлення – сонячні батареї.

Усі перелічені технології є наявними на сучасному ринку й технічно придатними для інтеграції до складу комплексу. Протокол LoRaWAN доступний для впровадження без істотних обмежень, характеризується відкритою специфікацією та широкою підтримкою розробників. Технологія SigFox також є наявною, однак її використання пов’язане з пропрієтарною інфраструктурою та потребує додаткової плати за підключення й обслуговування. Мікроконтролери RISC-V MCU представлені різними виробниками, мають добру доступність і можуть бути використані як основа обчислювального модуля системи.

Created by Universal Document Converter

Використання сонячних батарей як джерела живлення також є технічно можливим, але передбачає додаткові витрати.

З урахуванням наявності та доступності розглянутих рішень, а також їх технічних характеристик, обрано таку ідею реалізації проєкту: у якості системи для віддаленої передачі інформації про стан місцевості використовується протокол LoRa, що забезпечує необхідний радіус дії, енергоефективність і гнучкість масштабування мережі моніторингу.

5.3 Дослідження ринкового потенціалу та можливостей комерціалізації

У межах маркетингового аналізу визначено ключову ринкову потребу, на яку відповідає розроблювана система. Вона пов'язана з надійним і захищеним передаванням інформації, можливістю гнучкого масштабування мережі за кількістю вузлів і площею покриття, а також підтриманням високої швидкості обміну даними, наближеної до режиму реального часу. Поєднання цих характеристик формує стійкий інтерес користувачів до впровадження системи.

Цільовою аудиторією є організації територіальних громад, органи міського управління та приватні підприємства, що потребують постійного моніторингу інфраструктури, виробничих процесів та параметрів довкілля. Їхня поведінка на ринку зумовлюється рівнем конкуренції, бар'єрами входу для нових гравців і очікуваними економічними вигодами від застосування систем моніторингу, які дозволяють оптимізувати витрати та підвищити якість послуг (табл. 5.1).

Споживачі очікують від продукту гарантованого захищеного обміну даними, збереження продуктивності під час розширення мережі та високої швидкості обробки й передавання інформації. Відповідність цим критеріям визначає ринкову привабливість і конкурентоспроможність рішення у сегменті систем моніторингу та керування розподіленими об'єктами.

Created by Universal Document Converter

Таблиця 5.1 – Оцінка місткості та структури цільового ринку стартап-проєкту

№ з/п	Показник стану ринку (найменування)	Характеристика
1	Сукупний річний обсяг реалізації рішень моніторингу навколишнього середовища (6 конкурентних систем)	Близько 210 000 дол. США
2	Кількість ключових постачальників на ринку	Орієнтовно три основні гравці (технології Sigfox, NB-IoT, рішення Jooby)
3	Середній рівень рентабельності в галузі	Показник прибутковості становить приблизно 13 % за даними офіційної статистики України
4	Характер зміни місткості ринку	Ринок демонструє тенденцію до зростання
5	Бар'єри для виходу на ринок (обмеження для нових учасників)	Потрібне попереднє погодження проєктів та процедур встановлення обладнання
6	Спеціальні вимоги щодо стандартизації та сертифікації продукції	Необхідність відповідності національним і міжнародним нормативам: ДСТУ 2709-94, ДСТУ 4134-2002, ДСТУ EN 60950-22:2017

Паралельно проаналізовано зовнішні загрози (табл. 5.2). Серед них вирізняється складність виходу на ринок, пов'язана з жорсткими вимогами до безпеки й надійності, необхідністю відповідати стандартам і проходити випробування в спеціалізованих комітетах. Для подолання цього бар'єру компанія має заздалегідь забезпечити відповідність системи нормативним документам, відпрацювати рішення щодо безпечної експлуатації та підготувати повний пакет технічної документації. Ще однією загрозою є ризик втрати інноваційності продукту через постійну появу оновлених рішень конкурентів. Для збереження новизни система повинна регулярно модернізуватися.

Водночас зовнішнє середовище відкриває можливості для розвитку. Скорочення обсягів випуску нових продуктів конкурентами створює умови для активнішого охоплення ринку за рахунок посиленої роботи з клієнтами, поліпшення сервісу та своєчасного оновлення системи. Перспективними є також поглиблення співпраці з партнерами, що прискорює вирішення технічних і організаційних питань, та залучення кваліфікованих фахівців, яке дає змогу розширити штат, створити спеціалізовані підрозділи й підвищити загальний рівень конкурентоспроможності продукту.

Created by Universal Document Converter

Таблиця 5.2 – Аналіз конкурентного середовища на цільовому ринку

Особливості конкурентного середовища	**У чому проявляється дана характеристика**	**Вплив на діяльність підприємства**
Внутрішньогалузева конкуренція	Суперництво між компаніями однієї галузі, що постачають подібні товари й послуги	Орієнтація на потреби клієнтів, систематичний збір і врахування відгуків існуючих споживачів, удосконалення сервісу
Цінова конкуренція	Відмінності у вартості систем, які впливають на вибір покупця	Встановлення економічно обґрунтованої, привабливої для споживача ціни; гнучке коригування цін залежно від ринкової ситуації
Глобальний рівень конкуренції	Участь компанії в міжнародних виставках, конференціях, презентаціях	Активне просування рішень на закордонні ринки, пошук іноземних партнерів і дилерів для розширення присутності та закріплення позицій
Товарно-видова конкуренція	Змагання між системами з подібним функціональним призначенням	Поліпшення технічних характеристик: підвищення точності вимірювань, надійності й безпеки, спрощення інтерфейсу та експлуатації
Брендова конкуренція	Відмінності у сприйнятті торговельних марок споживачами	Формування впізнаваного іміджу компанії, посилення репутації та популярності бренду на цільових ринках

Таблиця 5.3 – Характеристика галузевої конкуренції за моделлю п’яти сил М. Портера

	Прямі конкуренти в галузі	**Потенційні конкуренти**	**Постачальники**	**Клієнти**	**Товари-замінники**
Компоненти аналізу	NB-IoT, Sigfox, Jooby	Київстар, Lifecell, Vodafone	Espressif, Tektelic, Semtech	Приватні підприємства, територіальні громади	Готові рішення на базі NB-IoT, SigFox, Winext
Результат	Боротьба за ринок споживачів шляхом створення більш якісного продукту, який краще задовольняє потреби користувачів	Необхідність пропонувати інноваційний продукт, орієнтований на потреби клієнтів, та інвестувати значні ресурси в маркетинг	Зниження залежності від окремих постачальників за рахунок підвищення якості поставок і оптимізації логістичних витрат	Розробка більш точної системи з дотриманням усіх вимог безпеки	Формування конкурентної переваги за рахунок нижчої ціни порівняно з аналогами та позиціонування як вітчизняного українського продукту

Created by Universal Document Converter

У межах аналізу конкурентоспроможності виокремлено кілька ключових факторів, щоб порівняти його з іншими рішеннями. Перший – підтримка клієнтів, яка передбачає повний супровід замовника: допомогу на етапі проєктування, підтримку під час запуску та подальше обслуговування архітектури. Це підвищує довіру користувачів і робить продукт привабливішим за рішення з обмеженим сервісом. Зведена SWOT-матриця наведена у табл. 5.4.

Вимоги клієнтів розглядаються як рушій модернізації системи та розвитку її функціоналу. Орієнтація на реальні потреби ринку дозволяє формувати дорожню карту продукту, своєчасно додавати затребувані можливості й позитивно впливати на фінансові результати та лояльність замовників. Важливим чинником є й ефективність системи для кінцевого споживача: здатність виконувати поставлені завдання з оптимальними витратами ресурсів, забезпечуючи стабільність, зручність використання та очікуваний результат.

Таблиця 5.4 – SWOT-матриця сильних і слабких сторін, можливостей і загроз стартап-проєкту

Сильні сторони	**Слабкі сторони**	**Можливості**	**Загрози**
Відповідність продукту вимогам клієнтів. Комплексна підтримка користувачі на різних етапах імплементації нового продукту. Оперативні відповіді на зауваження та побажання користувачів.	Відсутність сформованої репутації. Немає лояльних клієнтів. Малий масштаб компанії. Немає перевірених постачальників. Незначний досвід взаємодії між окремими підрозділами.	Інноваційний продукт. Конкурентна ціна. Високий рівень безпеки та надійності системи.	Сегментне середовище з обмеженим пулом потенційних клієнтів. Неготовність замовників інтегрувати дороговартісне апаратне забезпечення.

Суттєву роль відіграє маркетингова політика. Якісний маркетинг, зорієнтований на чітко визначену цільову аудиторію, допомагає правильно позиціонувати продукт, донести його переваги та забезпечити стійку присутність на ринку. Додатковим фактором конкурентоспроможності є ціна: встановлення доступної вартості, що не поступається аналогам і при цьому супроводжується ширшим функціоналом, створює вигідне співвідношення «ціна – можливості» та мотивує клієнтів обирати саме це рішення.

Created by Universal Document Converter

5.4 Формування стратегії виходу на ринок і позиціонування продукту

У табл. 5.5 наведено основні вимоги цільової аудиторії до системи віддаленого моніторингу стану навколишнього середовища, обрану базову стратегію її розвитку, а також ключові конкурентні переваги стартап-проєкту. Окремо подано асоціації, з якими має ототожнюватися продукт на ринку, що формують його цілісне позиціювання у сегменті IoT-рішень для моніторингу стану навколишнього середовища.

Таблиця 5.5 – Формування стратегії позиціонування продукту на ринку

№ з/п	Вимоги до продукту цільової аудиторії	Базова стратегія розвитку	Ключові конкурентні переваги стартап-проєкту	Асоціації, що формують комплексне ринкове позиціювання
1	Можливість розширення та модернізації системи	Конкурентне позиціонування	Система легко нарощується відповідно до зростання кількості об’єктів і вимог користувачів	Значне збільшення масштабу без втрати якості роботи; висока відмовостійкість застосованого обладнання; швидке відновлення окремих модулів і всієї системи
2	Висока точність вимірювань параметрів	Конкурентне позиціонування	Забезпечення коректного вимірювання технологічних показників із мінімальною похибкою	Підтримка широкого діапазону вимірювань; стабільні результати за наявності зовнішніх впливів; довіра користувачів до отриманих даних
3	Швидка обробка та передавання результатів вимірювань	Конкурентне позиціонування	Оперативна обробка зібраних даних і їхнє спрямування на сервер або в прикладні сервіси кінцевого користувача	Передавання даних на значні відстані між рівнями системи; енергоощадність кінцевих пристроїв; швидка доставка актуальної інформації до сервера й робочих додатків оператора
4	Надійність функціонування та безпечна робота системи	Конкурентне позиціонування	Реалізовані механізми постійного контролю стану об’єктів та своєчасного інформування про відхилення	Інтегрована система спостереження; механізми повідомлення про несправності чи небезпечні ситуації; підтримка безперервного моніторингу процесів

Created by Universal Document Converter

ВИСНОВКИ

Розроблено комплексну інформаційну систему моніторингу навколишнього середовища, орієнтовану на дистанційний збір даних та експлуатацію у міських, міжміських і польових умовах, із забезпеченням надійної передачі, зберігання та оперативного інформування користувача про відхилення параметрів.

Уточнено призначення системи як засобу відстеження та передавання параметрів довкілля, а також сформульовано вимоги до безпечної передачі/зберігання даних і масштабованості рішення, що дозволяє розглядати систему як практичну альтернативу застарілим підходам моніторингу.

Обґрунтовано вибір програмної платформи мережевого сервера LoRaWAN та побудовано серверну частину на базі ChirpStack із типовими компонентами інфраструктури (зокрема Redis, PostgreSQL, Mosquitto, gRPC API), що формує основу для подальшої інтеграції сервісів обробки/візуалізації та керування пристроями.

Імплементовано граничний пристрій і логіку передавання телеметрії: розроблено прототип сенсорного вузла на базі ESP32-S3 з радіомодулем SX1276 і сенсором BME680, а для підвищення енергоефективності реалізовано режим глибокого сну зі споживанням 6,5 μA та сценарій циклічної роботи «опитування–передавання–сон». Для інтеграції даних і користувацького перегляду реалізовано обробку телеметрії через Node-RED із підпискою на MQTT-повідомлення (у форматі JSON), що забезпечує основу для оперативної візуалізації та подальшого розширення логіки обробки даних.

Результати працездатності перевірено шляхом імітаційного моделювання та тестування, зокрема підтверджено коректність ключових інформаційних функцій системи (передавання параметрів, контроль стану обладнання, фіксація потенційних несправностей), а також доцільність накопичення історичних даних для подальшого аналізу.

Created by Universal Document Converter

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Клименко М.О., Прищепа А.М., Вознюк Н.М. Моніторинг довкілля : підручник. – 2-ге вид., допов. та переробл. – Рівне : НУВГП, 2023. – 350 с.

2. Моніторинг довкілля : підручник / [Боголюбов В.М., Клименко М.О., Мокін В.Б. та ін.] ; за ред. В.М. Боголюбова. – 2-ге вид., перероб. і доп. – Вінниця : ВНТУ, 2010. – 232 с.

3. Екологічний моніторинг довкілля : навч. посіб. / [Боголюбов В.М., Мокін В.Б., Крижановський Є.М. та ін.]. – Київ : НУБіП України, 2023. – 201 с.

4. Дистанційні методи моніторингу довкілля : навч. посіб. / [Бондар О.І. та ін.] ; за ред. О.І. Бондаря, П.Я. Унгуряна ; Держ. екол. акад. післядиплом. освіти та упр. – Київ : ОЛДІ-ПЛЮС, 2019. – 297 с.

5. Посудін Ю.І. Моніторинг довкілля з основами метрології : підручник. – К., 2012. – 426 с.

6. Моделювання і прогнозування стану довкілля : підручник / [Лаврик В.І., Боголюбов В.М., Полетаєва Л.М. та ін.] ; за ред. В.І. Лаврика. – К. : ВЦ «Академія», 2010. – 400 с.

7. Прилади і методи дослідження стану довкілля : навч. посіб. / Л.С. Старикович, К.П. Дудок, Н.М. Любас. – Львів : ЛНУ ім. І. Франка, 2014. – 195 с.

8. Радловська К. Локальний моніторинг довкілля для адміністративних районів і територіальних громад : монографія / за ред. О.С. Волошкіної. – Івано-Франківськ : Петраш К.Т., 2015. – 184 с.

9. Комп'ютеризовані регіональні системи державного моніторингу поверхневих вод: моделі, алгоритми, програми : монографія / за ред. В.Б. Мокіна. – Вінниця : Вид-во ВНТУ «УНІВЕРСУМ-Вінниця», 2005. – 315 с.

10. Патика В.П., Тараріко О.Г. Агроекологічний моніторинг та паспортизація сільськогосподарських земель. – К. : Фітосоціоцентр, 2002. – 256 с.

Created by Universal Document Converter

11. Погребенник В., Мельник М., Бойчук М. Екологічний моніторинг: концепції, принципи, системи // Вимірювальна техніка та метрологія. – 2005. – № 65. – С. 164–171.

12. Міхеєва І.Л., Орлов М.О., Трокоз В.А. Система моніторингу довкілля м. Києва // Вісник НТУУ «КПІ». Приладобудування : зб. наук. пр. – 2004. – Вип. 28. – С. 37–46.

13. Універсальна інформаційно-вимірювальна система оперативного екологічного моніторингу з використанням мобільних пристроїв / Мокін В.Б., Дзюняк Д.Ю., Горячев Г.В., Бондалєтов К.О. // Вісник Вінницького політехнічного інституту. – 2015. – № 5 (122). – С. 116–122.

14. Створення інформаційної системи моніторингу забруднення атмосферного повітря міста на основі технології «Інтернет речей» / Мокін В.Б., Собко Б.Ю., Дратований М.В., Крижановський Є.М., Горячев Г.В. // Вісник Вінницького політехнічного інституту. – 2017. – № 3. – С. 5–13.

15. Біловус А.С., Толстоусова О.В. Використання Інтернету речей для проведення моніторингу навколишнього середовища // Безпека людини у сучасних умовах = Human security in modern conditions : зб. тез наук. доп. 10-ї міжнар. наук.-метод. конф. та міжнар. конф. EAS, 6–7 грудня 2018 р. – Харків : Панов А.М., 2018. – С. 133–134.

16. Котелянець В.В. Інформаційна технологія моніторингу навколишнього середовища на базі концепції Інтернету речей : дис. … канд. техн. наук : 05.13.06 – Інформаційні технології. – Черкаси, 2019.

17. Давидік М.І. Інформаційно-вимірювальна система оперативного екологічного моніторингу водного середовища з прогнозом стану якості води на основі машинного навчання : магістерська робота ; спец. 174 «Автоматизація та комп'ютерно-інтегровані технології та робототехніка». – Миколаїв : НУК, 2024. – 80 с.

18. Карпенко М.І. Технології Інтернету речей для розробки програмно-апаратних комплексів моніторингу параметрів довкілля : кваліфікаційна робота бакалавра / Нац. ун-т харч. технологій. – Київ : НУХТ, 2022.

Created by Universal Document Converter