

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри,
д-р техн. наук, проф.

_____ І. М. Журавська

« __ » _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА


НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА

**РОЗПОДІЛЕНА СИСТЕМА МОНІТОРИНГУ
ПАРАМЕТРІВ ОТОЧУЮЧОГО СЕРЕДОВИЩА
НА БАЗІ LORAWAN**

Спеціальність 123 Комп'ютерна інженерія

Освітня програма «Комп'ютерна інженерія»

Здобувач

 Олександр МИХАЙЛОВ
підпис

« __ » _____ 202__ р.

Керівник канд. фіз.-мат. наук, доцент

_____ Сергій ПУЗИРЬОВ
підпис

« __ » _____ 202__ р.

Факультет
Кафедра
Рівень вищої освіти
Освітній ступінь
Спеціальність
Освітня програма

Комп'ютерних наук
Комп'ютерної інженерії
Другий (магістерський)
Магістр
123 Комп'ютерна інженерія
Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерної інженерії
_____ Ірина ЖУРАВСЬКА
«___» _____ 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача

_____ Михайлова Олександра Олександровича
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

Затверджена наказом ректора ЧНУ ім. Петра Могили від «___» _____ 202__ р.
№ _____

2. Строк представлення кваліфікаційної роботи «___» _____ 202__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом роботи є апаратно-програмний комплекс розподіленої системи моніторингу параметрів оточуючого середовища на базі LoRaWAN, що дозволив би покращити автоматизацію в аграрній сфері.

4. Перелік питань, що підлягають розробці _____
- 1) проаналізувати існуючі рішення щодо використання технології LoRaWAN;
 - 2) дослідити технологію LoRaWAN та її можливості для застосування в системах моніторингу навколишнього середовища;
 - 3) розробити апаратну та програмну частини розподіленої системи моніторингу на базі LoRaWAN;
 - 4) провести тести та аналіз ефективності запропонованого рішення для моніторингу полів або тепличних масивів;
 - 5) розробити рекомендації щодо впровадження системи в аграрному сферу України.

5. Перелік графічних матеріалів

Слайди презентації

6. Завдання до спеціальної частини

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи

Керівник роботи

Особистий підпис

Сергій ПУЗИРЬОВ

Власне ім'я ПРІЗВИЩЕ

Здобувач

Особистий підпис

Олександр МИХАЙЛОВ

Власне ім'я ПРІЗВИЩЕ

Дата видачі завдання « _____ » _____ 202__ р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної магістерської роботи

Тема: Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

№ з/п	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КМР	01.09.2024	16.09.2024	<i>Виконано</i>
2.	Огляд літератури за темою роботи	17.09.2024	25.09.2024	<i>Виконано</i>
3.	Складання календарного плану КМР	26.09.2024	27.09.2024	<i>Виконано</i>
4.	Аналіз предметної області	28.09.2024	04.10.2024	<i>Виконано</i>
5.	Розробка проектних рішень	05.10.2024	09.10.2024	<i>Виконано</i>
6.	Створення принципів схем та моделі плат	10.10.2024	21.10.2024	<i>Виконано</i>
7.	Розробка апаратно-програмного комплексу	22.10.2024	29.10.2024	<i>Виконано</i>
8.	Розробка додаткових рішень для плати датчиків	30.10.2024	07.11.2024	<i>Виконано</i>
9.	Налагодження та перевірка працездатності та тестування	08.11.2024	23.11.2024	<i>Виконано</i>
10.	Відгук керівника КМР	24.11.2024	24.11.2024	<i>Виконано</i>
11.	Оформлення КМР та презентації	25.11.2024	27.11.2024	<i>Виконано</i>
12.	Попередній захист	28.11.2024	28.11.2024	<i>Виконано</i>
13.	Рецензування	10.12.2024	11.12.2024	<i>Виконано</i>
14.	Захист кваліфікаційної роботи	19.12.2024	19.12.2024	<i>Виконано</i>

Керівник роботи

Особистий підпис

Сергій ПУЗИРЬОВ

Власне ім'я ПРІЗВИЩЕ

Здобувач

Особистий підпис

Олександр МИХАЙЛОВ

Власне ім'я ПРІЗВИЩЕ

АНОТАЦІЯ

до кваліфікаційної магістерської роботи
«Розподілена система моніторингу параметрів оточуючого середовища
на базі LoRaWAN»

Здобувач гр. 605м: Михайлов Олександр Олександрович
Керівник: канд. фіз.-мат. наук, доцент Пузирьов С. В.

У кваліфікаційній роботі досліджено питання автоматизації моніторингу параметрів оточуючого середовища з використанням технології LoRaWAN для аграрного сектора. Метою роботи є розробка розподіленої системи, яка забезпечує ефективний збір, передачу та аналіз даних про стан полів для підвищення врожайності та зменшення витрат.

Актуальність теми визначається необхідністю підвищення ефективності управління аграрними процесами в умовах сучасних викликів, таких як обмеженість ресурсів і необхідність збереження екологічної рівноваги. Традиційні методи моніторингу часто є неефективними, тоді як технологія LoRaWAN забезпечує економічне, енергоефективне та масштабоване рішення для моніторингу великих площ.

Робота включає аналіз існуючих рішень у сфері моніторингу, обґрунтування вибору технології LoRaWAN, а також проектування і реалізацію апаратно-програмного комплексу. Розроблено схеми вузлів системи, спроектовано та виготовлено друковані плати, написано програмне забезпечення для збору та обробки даних, а також виконано тестування системи у реальних умовах. Запропонована система дозволяє автоматизувати моніторинг таких параметрів, як температура, вологість ґрунту, освітленість тощо, забезпечуючи своєчасне отримання даних для прийняття управлінських рішень.

Кваліфікаційна магістерська робота містить: перелік скорочень, вступ, чотири розділи, висновок, перелік джерел посилання та чотири додатки.

У першому розділі проведено аналіз існуючих методів моніторингу та їх інтеграції в аграрний сектор, а також обґрунтовано вибір технології LoRaWAN як оптимального рішення. Другий розділ містить технічні й математичні аспекти функціонування сигналів LoRa, розроблену модель передачі даних у розподіленій системі та охарактеризовані параметри, що впливають на ефективність роботи мережі. У третьому розділі спроектовано електронні схеми вузлів, розроблено друковані плати, обрано технічні компоненти для реалізації системи. Описано процес інтеграції апаратних і програмних модулів. Четвертий розділ присвячений представленню результатів складання прототипу системи, опису процедури тестування її працездатності в реальних умовах. Проведено аналіз отриманих даних та оцінено ефективність роботи розробленої системи.

Кваліфікаційна магістерська робота містить 65 сторінок (без додатків), 42 рис., 6 табл., 30 літературних джерел, 7 додатків.

Ключові слова: LoRaWAN, моніторинг середовища, автоматизація, IoT, сільське господарство.

ABSTRACT

of the Master's Thesis

“Distributed environmental monitoring system based on LoRaWAN”

Applicant: Mikhailov Olexander Olexandrovich

Supervisor: Cand. Sci. (Phys.-Math.), Docent Puzyrov S. V.

The qualification work investigates the automation of environmental monitoring using LoRaWAN technology for the agricultural sector. The aim of the work is to develop a distributed system that provides efficient collection, transmission and analysis of field condition data to increase yields and reduce costs.

The relevance of the topic is determined by the need to improve the efficiency of managing agricultural processes in the face of modern challenges, such as limited resources and the need to maintain environmental balance. Traditional monitoring methods are often inefficient, while LoRaWAN technology provides a cost-effective, energy-efficient, and scalable solution for monitoring large areas.

The work includes analysis of existing monitoring solutions, justification of the choice of LoRaWAN technology, as well as design and implementation of the hardware and software complex. Schematics of the system components were developed, printed circuit boards were designed and manufactured, software for data collection and processing was written, and the system was tested in real conditions. The proposed system allows automating the monitoring of parameters such as temperature, soil moisture, illumination, etc., providing timely data for making management decisions.

The qualification master's thesis contains: a list of abbreviations, introduction, four chapters, conclusion, list of references and four appendices.

The first section analyzes the existing monitoring methods and their integration into the agricultural sector, and justifies the choice of LoRaWAN technology as an optimal solution. The second section contains the technical and mathematical aspects of the functioning of LoRa signals, the developed model of data transmission in a distributed system, and the parameters that affect the network efficiency. In the third section, the electronic circuits of the nodes are designed, printed circuit boards are developed, and technical components for the system implementation are selected. The process of integrating hardware and software modules is described. The fourth chapter is devoted to the presentation of the results of the system prototype assembly, description of the procedure for testing its performance in real conditions. The data obtained are analyzed and the efficiency of the developed system is evaluated.

The qualifying master's thesis contains 65 pages (without appendices), 42 figures, 6 tables, 30 references, and 7 appendices.

Keywords: LoRaWAN, environmental monitoring, automation, IoT, agriculture.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП.....	5
1 ОГЛЯД ЗАСОБІВ МОНІТОРИНГУ ПАРАМЕТРІВ ОТОЧУЮЧОГО СЕРЕДОВИЩА НА БАЗІ LORAWAN	8
1.1 Огляд засобів моніторингу параметрів оточуючого середовища	8
1.2 Інтеграція розподіленої системи моніторингу за оточуючим середовищем в аграрну сферу.....	10
1.3 Обґрунтування вибору підходів для реалізації технології LoRaWAN .	13
1.4 Специфікація вимог до розподіленої системи моніторингу параметрів оточуючого середовища на базі LoRaWAN	16
Висновки до розділу 1	19
2 ПОБУДОВА МАТЕМАТИЧНОЇ МОДЕЛІ	21
2.1 Технічні та математичні концепції сигналу LoRa	21
2.2 Передача пакету.....	24
2.3 Характеристики радіо інтерфейсу LoRa	27
2.4 Робота з фізичним рівнем LoRa	29
Висновки до розділу 2	36
3 ПОБУДОВА АПАРАТНО-ПРОГРАМНОЇ СИСТЕМИ	37
3.1 Вибір апаратного забезпечення системи	37
3.2 Проєктування апаратного забезпечення системи	44
Висновки до розділу 3	50
4 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ АПАРАТНО-ПРОГРАМНОЇ СИСТЕМИ.....	51
4.1 Реалізація апаратної частини шлюзу та вузлів.....	51
4.2 Реалізація ємкісно-вимірювального модулю	52
4.3 Розробка програмного забезпечення.....	56
4.4 Тестування розробленого комплексу	61

Висновки до розділу 4	64
ВИСНОВКИ.....	66
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	67
ДОДАТОК А Код програми шлюзу	71
ДОДАТОК Б Код програми вузлу датчиків	82
ДОДАТОК В Код програми модуля ємкісного модуля ємкісного датчика вологи.....	86
ДОДАТОК Г Принципова електрична схема шлюзу	91
ДОДАТОК Д Принципова електрична схема вузлу датчиків	92
ДОДАТОК Е Принципова електрична схема модуля ємкісного датчика вологи..	93
ДОДАТОК Ж Апробація кваліфікаційної роботи	94

ПЕРЕЛІК СКОРОЧЕНЬ

АЦП	– аналого-цифровий перетворювач
ДЗЗ	– дистанційне зондування Землі
ЛЧМ	– лінійна частотна модуляція
ФНЧ	– фільтр низьких частот
AP	– Access Point
BR	– Bitrate
BW	– Broadcast Width
CoAP	– Constrained Application Protocol
CSS	– Chirp Spread Spectrum
DR	– Data Rate
GPS	– Global Positioning System
HTTP	– HyperText Transfer Protocol
IoT	– Internet of Things
LoRa	– Long Range
LoRaWAN	– Long Range Wide Area Network
MAC	– Media Access Control
MIC	– Message Integrity Code
MQTT	– Message Queuing Telemetry Transport
PWM	– Pulse-Width Modulation
SF	– Spreading Factor

ВСТУП

Сучасне сільське господарство вимагає інноваційних підходів для підвищення ефективності виробництва та раціонального використання ресурсів. Одним із важливих напрямків розвитку аграрної галузі є впровадження інформаційних технологій для моніторингу та контролю стану полів. У цьому контексті розподілені системи моніторингу, побудовані на базі інноваційних мережевих технологій, таких як LoRaWAN (англ. Long Range Wide Area Network), є ключовими рішеннями, що дозволяють значно покращити точність і ефективність управління агротехнологічними процесами.

Традиційні методи моніторингу та управління полями часто вимагають значних людських і фінансових витрат, що обмежує можливості швидкої реакції на зміну умов вирощування культур. Впровадження новітніх інформаційних технологій, зокрема систем моніторингу параметрів навколишнього середовища, стає актуальним інструментом для оптимізації агротехнічних процесів. Це дозволяє не лише збільшити врожайність, але й зменшити негативний вплив на навколишнє середовище та знизити витрати на вирощування культур.

У світовій практиці такі системи вже стали частиною розумного землеробства (англ. smart farming), де застосування цифрових технологій дозволяє з точністю визначати потреби рослин у воді, добривах, а також контролювати їх стан у реальному часі. Для України, де сільське господарство є важливим чинником стабільності та розвитку економіки, впровадження подібних технологій має надзвичайно велике значення.

Система моніторингу параметрів навколишнього середовища на базі LoRaWAN дозволяє створити ефективну та економічну інфраструктуру для відстеження таких важливих показників, як вологість ґрунту, температура повітря, рівень сонячного освітлення та інші. Завдяки широкій зоні покриття та низькому енергоспоживанню, LoRaWAN стає одним із найкращих варіантів для побудови розподілених мереж у сільськогосподарській сфері. Це дозволить аграріям своєчасно отримувати точні дані для прийняття оперативних рішень щодо догляду

за полями, підвищити продуктивність та мінімізувати витрати.

Особливо актуальною така система є для великих сільськогосподарських угідь, де регулярний фізичний огляд полів є надто витратним і неефективним. Розподілена система моніторингу на базі LoRaWAN дозволяє встановити бездротові сенсори на великих площах, які автоматично передаватимуть дані до центральної системи для їх аналізу. Це надає можливість своєчасно виявляти аномалії, змінювати полив або внесення добрив, що, в свою чергу, допоможе уникнути втрат врожаю.

Таким чином, впровадження розподілених систем моніторингу в аграрному секторі України стає важливим кроком на шляху до підвищення ефективності сільськогосподарських процесів, збереження ресурсів та покращення екологічної стійкості.

Об'єкт дослідження: процес моніторингу параметрів оточуючого середовища агропромислового комплексу для забезпечення нагляду за полями.

Предмет дослідження: методи та засоби побудови та функціонування системи моніторингу на базі LoRaWAN для аграрного сектору.

Мета дослідження: розробка розподіленої системи моніторингу параметрів оточуючого середовища на базі LoRaWAN для ефективного контролю та управління сільськогосподарськими процесами в Україні.

Завдання дослідження:

- 1) проаналізувати існуючі рішення щодо використання технології LoRaWAN;
- 2) дослідити технологію LoRaWAN та її можливості для застосування в системах моніторингу навколишнього середовища;
- 3) розробити апаратну та програмну частини розподіленої системи моніторингу на базі LoRaWAN;
- 4) провести тести та аналіз ефективності запропонованого рішення для моніторингу полів або тепличних масивів;
- 5) розробити рекомендації щодо впровадження системи в аграрному сфері

України.

Методи дослідження: абстрагування, аналіз, спостереження, порівняння, комп'ютерне моделювання.

Практичне значення роботи полягає в тому, що дана розробка дозволить автоматизувати моніторинг параметрів навколишнього середовища в аграрній сфері, покращити нагляд за станом полів, покрити велику площу за рахунок мінімальних вкладень та мінімізувати споживання електроенергії.

Робота пройшла **апробацію** на XXI Міжнародній науковій конференції «Ольвійський форум – 2024: стратегії країн Причорноморського регіону в геополітичному просторі» (Миколаїв, 20–23 червня 2024 р.) [1].

1 ОГЛЯД ЗАСОБІВ МОНІТОРИНГУ ПАРАМЕТРІВ ОТОЧУЮЧОГО СЕРЕДОВИЩА НА БАЗІ LORAWAN

1.1 Огляд засобів моніторингу параметрів оточуючого середовища

Для забезпечення ефективного управління сільськогосподарськими угіддями та оптимізації вирощування культур існує цілий ряд засобів моніторингу параметрів навколишнього середовища. Ці засоби дозволяють збирати дані про стан ґрунту, атмосферні умови, рівень освітлення, вологість та інші фактори, які впливають на продуктивність аграрних процесів. Сучасні технології моніторингу охоплюють як класичні методи збирання даних, так і новітні рішення, такі як бездротові сенсорні мережі на базі технології LoRaWAN, що мають величезний потенціал для розвитку аграрної сфери України.

1) Традиційні засоби моніторингу.

Аналіз зразків ґрунту та рослин. Цей метод передбачає взяття фізичних зразків ґрунту або рослин для подальшого лабораторного аналізу, що дозволяє визначити рівень поживних речовин, кислотність ґрунту, наявність хвороб та інших факторів, що впливають на врожайність. Однак цей метод вимагає значних ресурсів, як у плані часу, так і коштів, і не дозволяє оперативно отримувати актуальні дані для швидкого прийняття рішень.

Метеостанції – це прилади, що фіксують такі параметри, як температура, вологість, швидкість вітру та кількість опадів. Вони забезпечують цінну інформацію для прогнозування погоди і прийняття рішень щодо агротехнічних заходів. Проте встановлення метеостанцій вимагає значних початкових інвестицій, а їхня кількість на полях може бути обмежена через високі витрати на обслуговування.

2) Цифрові засоби моніторингу.

GPS-системи та дрони. Сучасні технології дозволяють використовувати безпілотні літальні апарати (дрони) для збору даних з полів. Дрони можуть обладнуватись різними сенсорами (оптичними, інфрачервоними,

мультиспектральними), які дозволяють оцінити стан культур, виявити хвороби чи шкідників, а також аналізувати ступінь розвитку рослин. GPS-навігація також дозволяє точно відслідковувати географічні координати зібраних даних. Основною перевагою таких технологій є можливість отримання великих обсягів даних за короткий час. Однак їх використання є дороговартісним і вимагає спеціальної підготовки.

Системи дистанційного зондування Землі (ДЗЗ). Системи ДЗЗ використовують супутники для збору даних про поверхню землі. Цей метод дозволяє отримувати інформацію про великі площі одночасно, що є надзвичайно корисним для моніторингу стану ґрунту та рослинності на полях. Однак точність даних, зокрема на локальному рівні, може бути недостатньою через вплив погодних умов, а інтервали збору інформації можуть бути не досить частими для оперативного моніторингу.

3) Бездротові сенсорні мережі та технології IoT.

З розвитком технологій **Інтернету речей (англ. IoT)** з'явилась можливість створення розподілених систем моніторингу, які використовують мережу бездротових сенсорів для збору інформації в режимі реального часу. У такій системі на території полів встановлюються сенсори, що фіксують такі параметри, як температура, вологість ґрунту, рівень освітленості тощо. Дані від сенсорів передаються через бездротову мережу до центральної системи для подальшого аналізу.

LoRaWAN, як інноваційна технологія, є найперспективнішою для побудови розподілених систем моніторингу в аграрному секторі. Її основною перевагою є можливість передачі даних на великі відстані з мінімальним енергоспоживанням. Це дозволяє встановлювати сенсори в віддалених і важкодоступних місцях, не потребуючи частого обслуговування та заміни батарей [2].

Основні переваги LoRaWAN для моніторингу полів:

– широке покриття: технологія забезпечує передачу даних на відстані до 15 км у сільській місцевості, що робить її ідеальною для великих аграрних угідь;

- низьке енергоспоживання: сенсори можуть працювати на одній батареї протягом декількох років;
- низька вартість обслуговування: завдяки невеликій кількості базових станцій, необхідних для покриття великих територій, та довговічності пристроїв;
- можливість роботи в реальному часі: аграрії отримують актуальні дані, що дозволяє своєчасно реагувати на зміни умов.

Таким чином, технологія LoRaWAN у поєднанні з IoT є найбільш підходящим рішенням для створення розподілених систем моніторингу параметрів оточуючого середовища. Вона дозволяє автоматизувати процес збору даних, мінімізувати людський фактор і знизити витрати, що особливо актуально для аграрного сектору України.

1.2 Інтеграція розподіленої системи моніторингу за оточуючим середовищем в аграрну сферу

Інтеграція розподілених систем моніторингу параметрів навколишнього середовища в аграрну сферу відкриває нові можливості для автоматизації управління агротехнічними процесами, підвищення ефективності використання ресурсів та поліпшення екологічної стійкості сільськогосподарських угідь. Такі системи, базовані на технології LoRaWAN та Інтернеті речей, дозволяють фермерам оперативно отримувати дані про стан полів і ухвалювати обґрунтовані рішення для підвищення врожайності. Основною метою впровадження таких рішень є оптимізація процесів нагляду за полями, зниження витрат на обслуговування та поліпшення якості продукції [7].

Ключові етапи інтеграції системи моніторингу:

1) визначення вимог та планування – перед впровадженням системи моніторингу необхідно провести детальний аналіз вимог господарства. Це включає визначення ключових параметрів навколишнього середовища, які потрібно контролювати (температура, вологість ґрунту, рівень освітлення, швидкість вітру тощо), а також планування розміщення сенсорів на полях. На цьому етапі важливо

оцінити розмір полів, топографічні особливості території, потенційні зони ризику та проблемні ділянки;

2) вибір та встановлення обладнання – після аналізу вимог наступним кроком є вибір відповідних сенсорів та інших елементів системи. Для аграрних цілей можуть бути використані такі типи сенсорів:

- сенсори вологості ґрунту для визначення рівня зволоження та оптимізації систем поливу;

- термометри та гігromетри для моніторингу температури та вологості повітря;

- сенсори освітленості для оцінки рівня сонячного випромінювання;

Ці сенсори підключаються до мережі LoRaWAN, що дозволяє їм передавати дані на великі відстані до центральної базової станції, яка потім відправляє дані до хмарної платформи для обробки та аналізу;

3) налаштування мережевої інфраструктури – LoRaWAN системи складаються з базових станцій, що приймають дані від сенсорів і передають їх на сервер через інтернет. Оскільки технологія LoRaWAN дозволяє охоплювати великі території (до 15 км у відкритій місцевості), кількість базових станцій може бути мінімальною, що знижує вартість інфраструктури. На цьому етапі важливо налаштувати мережеву архітектуру таким чином, щоб вона могла ефективно обробляти великий обсяг даних, що надходять від сенсорів;

4) інтеграція з аналітичними платформами – дані, зібрані системою моніторингу, передаються до хмарних платформ або серверів, де відбувається їх обробка та аналіз. Ці платформи за допомогою аналітичних інструментів, алгоритмів машинного навчання або системи штучного інтелекту дозволяють аграріям отримувати зрозумілу інформацію про стан полів та прогнозування змін. Системи можуть генерувати рекомендації щодо поливу, внесення добрив, захисту від шкідників або планування агротехнічних робіт;

5) автоматизація управління агротехнічними процесами – інтеграція з системами управління сільськогосподарськими процесами дозволяє

автоматизувати певні дії на основі отриманих даних. Наприклад, система може автоматично включати або вимикати системи зрошення, в залежності від вологості ґрунту, або подавати сигнали для початку захисних заходів при прогнозуванні ризиків пошкодження врожаю внаслідок зміни погодних умов. Така автоматизація дозволяє мінімізувати витрати на обслуговування полів і забезпечити більш раціональне використання ресурсів, таких як вода чи добрива.

Переваги інтеграції розподіленої системи моніторингу:

- оптимізація використання ресурсів дозволить точно визначити, коли необхідно поливати або вносити добрива, що значно скорочує надмірне використання ресурсів і знижує витрати;

- отримуючи актуальні дані, фермери можуть своєчасно вживати заходів для запобігання проблем, що негативно впливають на врожайність (шкідники, захворювання, стресові погодні умови);

- завдяки оптимізованому використанню води, добрив і пестицидів, системи моніторингу сприяють зменшенню негативного впливу на навколишнє середовище;

- система дозволяє фермерам краще адаптуватися до змін погодних умов та інших зовнішніх факторів, що можуть негативно вплинути на врожайність.

Виклики при інтеграції:

- встановлення мережі сенсорів, базових станцій та аналітичних систем потребує значних фінансових вкладень на початковому етапі;

- у віддалених регіонах можуть виникнути проблеми з підключенням до інтернету або електропостачанням, що ускладнює роботу системи;

- фермери та працівники повинні мати базові знання щодо роботи з такими системами, що вимагає проведення навчання та консультацій.

Інтеграція розподілених систем моніторингу на основі технології LoRaWAN та IoT в аграрну сферу сприяє автоматизації управління процесами, підвищенню ефективності використання ресурсів та екологічній стійкості. Системи надають фермерам оперативні дані про стан полів, допомагаючи ухвалювати обґрунтовані

рішення для підвищення врожайності. Ключові етапи включають: визначення вимог, вибір обладнання, налаштування інфраструктури, інтеграцію з аналітичними платформами та автоматизацію процесів. Основними перевагами є оптимізація ресурсів, зменшення негативного впливу на довкілля та підвищення адаптивності до змін клімату.

1.3 Обґрунтування вибору підходів для реалізації технології LoRaWAN

Вибір підходів для реалізації системи моніторингу параметрів оточуючого середовища в аграрній сфері залежить від низки факторів, зокрема специфіки сільськогосподарських процесів, характеристик полів, доступності технологій та економічної доцільності. Для аграрного сектору України, де велике значення мають ефективність використання ресурсів та зниження витрат, оптимальним є впровадження розподілених систем на базі бездротових технологій, таких як LoRaWAN.

Технологія LoRaWAN є одним з найкращих рішень для створення мереж моніторингу параметрів навколишнього середовища завдяки її здатності передавати дані на великі відстані при низькому енергоспоживанні. Це робить LoRaWAN ідеальним для розподілених мереж, де сенсори можуть бути встановлені на великих площах без необхідності частого технічного обслуговування. Основні переваги LoRaWAN [3-5]:

- LoRaWAN дозволяє передавати дані на відстані до 15 км у сільській місцевості, що є надзвичайно корисним для покриття великих аграрних угідь;
- сенсори, які працюють на LoRaWAN, можуть функціонувати на одній батареї протягом кількох років, що мінімізує витрати на обслуговування;
- система може легко розширюватися, що дозволяє додавати нові сенсори та покращувати мережу без значних витрат на інфраструктуру;

– використання технології LoRaWAN є відносно дешевим у порівнянні з іншими технологіями бездротової передачі даних (наприклад, мобільними мережами або супутниковими системами).

Детальна інформація та порівняння технологій передачі даних зібрана і представлена у табл. 1.1 та коротко представлена на рис. 1.1.

Таблиця 1.1 – Порівняння технологій передачі даних

	Bluetooth	Wi-Fi	Z-Wave	IEEE 802.15.4 (Zigbee, Thread)	LTE-M	NB-IoT	Sigfox	LoRa
Дистанція	10 м - 1,5 км	15 м - 100 м	30 м - 50 м	30 м - 100 м	1 км - 10 км	1 км - 10 км	3 км - 50 км	2 км - 20 км
Пропускна здатність	125 кбіт/с - 2 Мбіт/с	54 Мбіт/с - 1,3 Гбіт/с	10 кбіт/с - 100 кбіт/с	20 кбіт/с - 250 кбіт/с	до 1 Мбіт/с	до 200 кбіт/с	до 100 біт/с	10 кбіт/с - 50 кбіт/с
Споживана потужність	низька	середня	низька	низька	середня	низька	низька	низька
Вартість модуля	менше \$5	менше \$10	менше \$10	\$8 - \$15	\$8 - \$20	\$8 - \$20	менше \$5	\$8 - \$15
Топологія	P2P, зірка, сітка	зірка, сітка	сітка	сітка	зірка	зірка	зірка	зірка

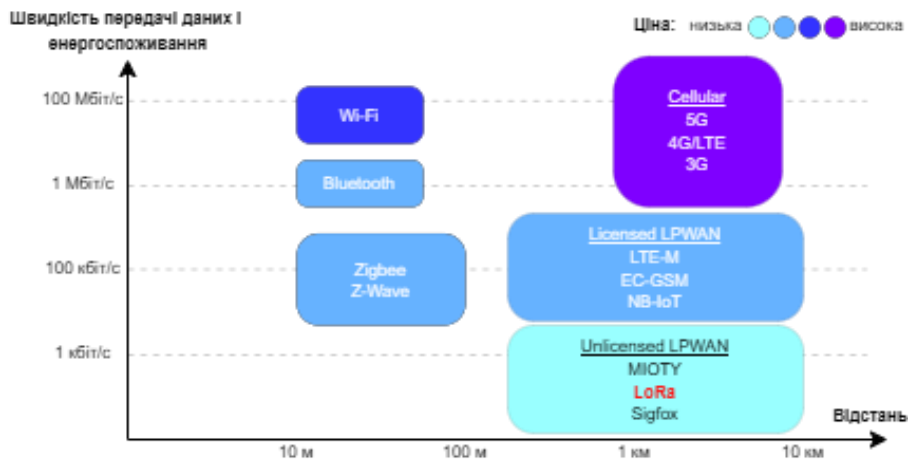


Рисунок 1.1 – Порівняння технологій передачі даних

Технологія вже активно використовується у світовій практиці для моніторингу параметрів полів, таких як вологість ґрунту, температура, рівень освітлення, що дозволяє оптимізувати використання ресурсів. Зокрема, LoRaWAN широко застосовується для автоматизації систем поливу, що допомагає знизити

витрати води та добрив. В Україні, де великі площі полів потребують ефективного управління, LoRaWAN може значно підвищити продуктивність і зменшити втрати врожаю [6].

Інтеграція IoT у сільське господарство дозволяє створити мережі сенсорів, які безперервно фіксують та передають дані про стан полів, метеоумови та інші параметри навколишнього середовища в режимі реального часу. Переваги IoT підходів:

- замість ручного збору даних з різних ділянок полів, сенсори постійно передають актуальну інформацію на сервери, що дозволяє аграріям швидко реагувати на зміни в стані полів;

- системи на базі IoT здатні обробляти дані в реальному часі, що дозволяє отримувати прогнози та аналітику щодо майбутніх агротехнічних заходів, таких як полив, внесення добрив або збирання врожаю;

- завдяки отриманню точних даних, фермери можуть налаштувати більш точний контроль за використанням води, добрив, що дозволяє знизити операційні витрати та поліпшити ефективність процесів.

LoRaWAN має доволі великий спектр використання за рахунок своєї великої зони покриття та сили сигналу, що має високу стабільність й перешкодостійкість. IoT дозволяє інтегрувати всі етапи сільськогосподарського виробництва в одну систему, де кожен сенсор надає дані, які використовуються для прийняття рішень. Це важливо для аграрної сфери України, де управління великими площами є значною проблемою через необхідність постійного моніторингу та оперативної реакції на зміну погодних умов. Крім того, використання IoT сприяє підвищенню ефективності вирощування культур, зменшуючи людський фактор та витрати. Розумні лічильники з LoRaWAN-модулями автоматично передають показники водо-, газо- та електролічильників на сервер, спрощуючи облік і формування рахунків. Технологія також дозволяє передавати дані з лічильників у кожній квартирі через мережу LoRaWAN до керуючої компанії, забезпечуючи точний та безперебійний моніторинг споживання.

Для забезпечення ефективного моніторингу параметрів навколишнього середовища у великих господарствах важливо застосовувати розподілені сенсорні мережі. Вони дозволяють встановлювати велику кількість сенсорів на значних площах, з можливістю збору даних з різних ділянок поля. Переваги розподілених сенсорних мереж:

- розподілені мережі дозволяють встановлювати сенсори в будь-яких потрібних точках поля, що дозволяє отримувати точні дані про конкретні ділянки з різними умовами;

- завдяки розподіленим мережам можна детально відстежувати зміну параметрів навколишнього середовища на кожній ділянці поля, що допомагає приймати рішення з урахуванням локальних умов;

- системи легко розширюються шляхом додавання нових сенсорів, що робить їх придатними для великих сільськогосподарських підприємств.

Розподілені сенсорні мережі особливо актуальні для великих полів в Україні, де різні ділянки можуть мати різні характеристики ґрунту, вологості, освітлення та температури. Це дозволяє аграріям здійснювати диференційований підхід до догляду за полями, що підвищує ефективність використання ресурсів і зменшує втрати врожаю через нерівномірність умов.

1.4 Специфікація вимог до розподіленої системи моніторингу параметрів оточуючого середовища на базі LoRaWAN

1. Загальні вимоги.

Апаратно-програмний комплекс розподіленої системи моніторингу параметрів оточуючого середовища повинен забезпечувати збір, передачу, обробку даних про ключові показники навколишнього середовища (температура, вологість, освітленість, рівень вологості ґрунту) у певний період часу. Система повинна бути оптимізована для сільськогосподарського використання та здатна функціонувати на великих площах полів.

2. Апаратні вимоги:

1) *сенсори* – система повинна включати набір сенсорів, що дозволяють вимірювати основні параметри навколишнього середовища:

– температура повітря: діапазон від мінус 40 °С до + 85 °С, точність вимірювання $\pm 0,5$ °С;

– вологість повітря: діапазон від 0 % до 100 % відносної вологості, точність ± 3 %;

– вологість ґрунту: можливість визначати об'ємну вологість ґрунту в діапазоні 0–100 %, точність ± 5 %;

– освітленість: вимірювання у люксах, діапазон 0–100 тис. люкс, точність ± 5 %;

– температура ґрунту: діапазон від 0 до 26 °С, точність вимірювання $\pm 0,5$ °С;

2) *комунікаційне обладнання*:

– LoRa-модулі для забезпечення зв'язку між сенсорами та базовими станціями. Діапазон зв'язку має становити до 15 км у відкритій місцевості;

– GSM-модуль для передачі даних до мережі інтернет кінцевому користувачу.

Сенсори повинні мати можливість працювати на автономних джерелах живлення (батареї або сонячні панелі) з терміном служби не менше 2-3 років без необхідності заміни батареї. Базові станції повинні підтримувати підключення до електромережі або альтернативні джерела енергії (сонячні панелі);

3) *модулі збору та передавання даних* – базові станції повинні мати вбудовані або зовнішні модулі для передачі даних через Інтернет (3G/4G) для відправки даних до хмарної платформи. Контролери сенсорів повинні підтримувати протоколи бездротового зв'язку LoRaWAN і мати інтегровані функції для збереження даних у разі тимчасової відсутності зв'язку;

4) *стійкість до умов навколишнього середовища* – сенсори та базові станції повинні мати захищені корпуси (мінімум клас захисту IP65) для роботи в різних

погодних умовах, включаючи дощ, пил, сильний вітер та екстремальні температури.

3. Програмні вимоги:

1) програмне забезпечення для збору та аналізу даних:

– апаратна платформа для збору даних;

2) інтеграція з IoT – система повинна підтримувати стандартні IoT-протоколи для інтеграції з іншими сільськогосподарськими системами (автоматичний полив, керування добривами). Підтримка інтеграції з хмарними платформами через протоколи MQTT, HTTP або CoAP для передачі даних до сторонніх аналітичних систем;

3) безпека та конфіденційність:

– шифрування даних на рівні сенсорів і при передачі даних через LoRaWAN;

– аутентифікація користувачів на всіх рівнях доступу до даних та управління системою;

4. Експлуатаційні вимоги:

1) надійність і безперервність роботи – система повинна забезпечувати безперервну роботу з доступністю не менше 99,5 % часу. У разі відключення однієї базової станції, інші елементи системи повинні продовжувати функціонувати автономно.

2) легкість монтажу та обслуговування – сенсори повинні бути легко інтегровані в польові умови з можливістю швидкого налаштування без залучення висококваліфікованих спеціалістів. Система повинна мати можливість дистанційного моніторингу та діагностики сенсорів для зменшення потреби в фізичному обслуговуванні;

3) масштабованість – система повинна підтримувати можливість масштабування шляхом додавання нових сенсорів і базових станцій без зміни архітектури;

4) енергетична ефективність – система повинна забезпечувати мінімальне енергоспоживання для тривалого використання автономних джерел живлення.

Сенсори повинні підтримувати можливість переходу в енергозберігаючий режим при відсутності необхідності в передачі даних.

5. Економічні вимоги.

Система повинна мати конкурентоспроможну ціну та бути економічно вигідною для впровадження в аграрних підприємствах України. Витрати на обслуговування повинні бути мінімальними, а термін служби основних компонентів – не менше 5 років;

6. Функціональні вимоги:

1) *моніторинг параметрів* – збір і відображення таких параметрів як температура, вологість повітря і ґрунту, освітленість, швидкість вітру в реальному часі;

2) *нотифікації та тривоги* – система повинна генерувати автоматичні повідомлення та тривоги у разі виходу параметрів за межі допустимих значень;

3) звіти та історія даних:

– генерація звітів на основі зібраних даних (щоденні, тижневі, місячні звіти);
– можливість зберігання та перегляду історичних даних для аналізу довгострокових змін.

Ця специфікація вимог до апаратно-програмного комплексу розподіленої системи моніторингу параметрів оточуючого середовища на базі LoRaWAN визначає основні технічні та експлуатаційні характеристики системи, що дозволяє забезпечити ефективний моніторинг та управління агротехнічними процесами. Система повинна бути надійною, масштабованою, економічно вигідною та легкою у використанні.

Висновки до розділу 1

Розробка розподіленої системи моніторингу параметрів оточуючого середовища на базі технології LoRaWAN для аграрної сфери України є важливим кроком до підвищення ефективності сільськогосподарського виробництва. Впровадження сучасних IoT-рішень та сенсорних мереж дозволяє фермерам

отримувати актуальні дані про стан полів, аналізувати їх та оперативно реагувати на зміни кліматичних і агротехнічних умов.

Основні переваги системи:

– технологія LoRaWAN забезпечує передачу даних на великі відстані з мінімальним енергоспоживанням, що дозволяє моніторити великі площі полів з низькими витратами на інфраструктуру;

– сенсори та базові станції дозволяють отримувати точні дані про стан ґрунту, атмосферні умови та кліматичні зміни, забезпечуючи оптимальне управління ресурсами (водою, добривами тощо);

– інтеграція IoT дозволяє автоматизувати процеси та інтегрувати їх в єдину систему управління господарством, що підвищує точність і зменшує людський фактор;

– масштабованість і енергетична ефективність системи роблять її гнучкою і придатною для впровадження на різних типах аграрних підприємств, від невеликих господарств до великих агрохолдингів.

Система сприятиме підвищенню продуктивності, скороченню витрат на ресурси та оптимізації агротехнічних заходів, що, в свою чергу, може позитивно вплинути на розвиток аграрної галузі України та її економічну стійкість в умовах сучасних умов.

2 ПОБУДОВА МАТЕМАТИЧНОЇ МОДЕЛІ

2.1 Технічні та математичні концепції сигналу LoRa

LoRa (англ. Long Range) – це технологія бездротового зв'язку, яка використовується для передачі даних на великі відстані з мінімальним енергоспоживанням. Вона базується на модуляції Chirp Spread Spectrum (CSS), що забезпечує високу чутливість приймача та стійкість до перешкод, навіть у складних умовах роботи. Завдяки цим характеристикам LoRa є основою для побудови енергоефективних та надійних мереж Інтернету речей (IoT).

Головна перевага LoRa – її здатність забезпечувати зв'язок на відстанях до 15-20 км у сільській місцевості та до 2-5 км у міських умовах. Технологія оптимізована для передачі невеликих обсягів даних, таких як показники сенсорів, що робить її ідеальною для моніторингу оточуючого середовища, аграрних систем та інших IoT-застосувань.

Фізичний радіо інтерфейс LoRa ґрунтується на використанні широкопasmових радіосигналів з великою базою B , що набагато більша за одиницю. Цей вид радіосигналів має дві головні особливості:

- ширина спектра радіосигналу BW значно більша за швидкість передавання даних R_b ($BW \gg R_b$);
- кореляційна функція істотно вужча за кореляційну функцію вузькосмугового радіосигналу з базою $B \sim 1$.

Частотна надмірність широкопasmового радіосигналу зумовлює його високу завадостійкість, а вузька кореляційна функція - високу точність часової синхронізації.

Широкопasmовий радіосигнал LoRa являє собою сигнал із лінійною частотною модуляцією (ЛЧМ) або CSS. Частота CSS радіосигналу може як збільшуватися (англ. up-chirp), так і зменшуватися (англ. down-chirp). Математично ЛЧМ сигнал подається у вигляді виразу:

$$x(t) = A_0 \times \cos \cos \left(\omega_0 \times t + \frac{\mu}{2} \times t^2 \right), \quad \text{де } \frac{T_{sym}}{2} \leq t < \frac{T_{sym}}{2}$$

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

та описується наступними параметрами:

BW – ширина спектра радіосигналу;

f_0 ; $\omega_0 (= 2\pi f_0)$ – центральна (несуча) частота радіосигналу - центральна (несуча) частота радіосигналу;

$f_H (= f_0 - \frac{BW}{2})$; $\omega_H (= 2\pi f_H)$ – нижня частота радіосигналу;

$f_B (= f_0 + \frac{BW}{2})$; $\omega_B (= 2\pi f_B)$ – верхня частота радіосигналу;

SF – коефіцієнт розширення спектра (змінюється в діапазоні від 7 до 12);

$T_{sym} = \frac{2^{SF}}{BW}$ – тривалість радіосигналу;

$\mu = \frac{BW}{T_{sym}}$ – швидкість зміни частоти радіосигналу;

$B = BW \times T_{sym} = 2^{SF}$ – база радіосигналу.

Коефіцієнт розширення спектра (SF) визначає розрядність символу даних (у бітах), переданого через радіо інтерфейс за час T_{sym} .

На рис. 2.1 наведено вигляд ЛЧМ-сигналу в часовій області, а на рис. 2.2 і рис. 2.3 показано його спектр із $BW = 125$ кГц і базою, що дорівнює 128 ($SF = 7$) і 4096 ($SF = 12$) відповідно [8-9].

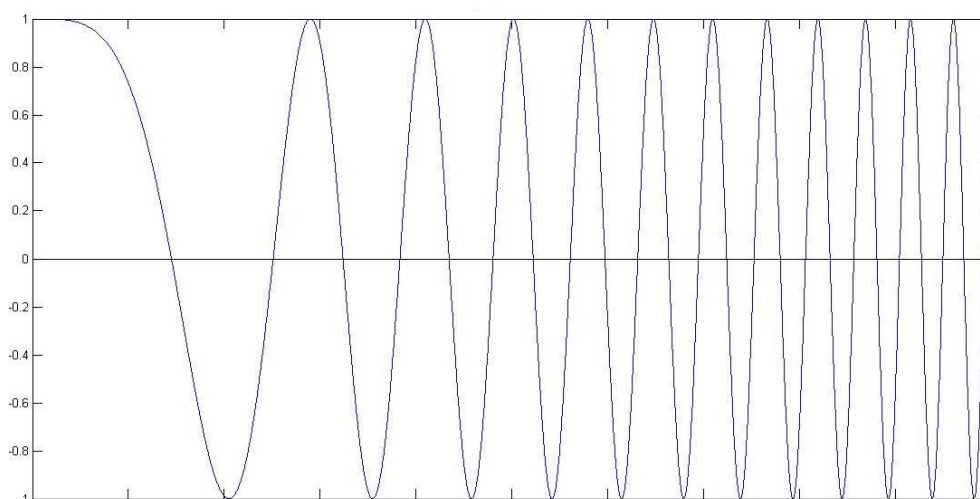


Рисунок 2.1 – ЛЧМ-сигнал

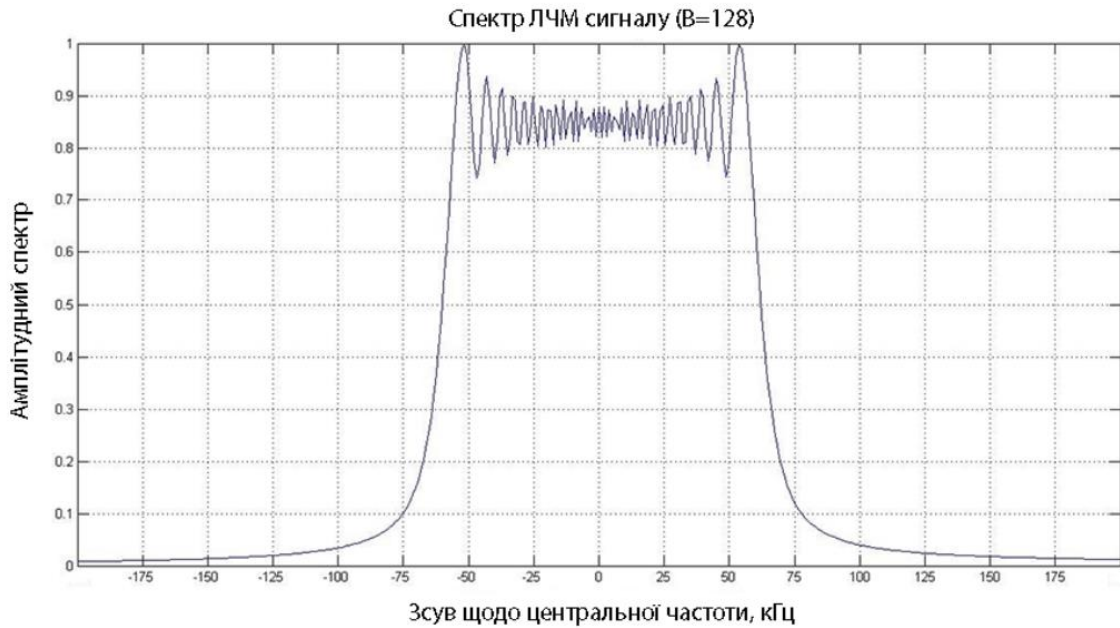


Рисунок 2.2 – Спектр ЛЧМ сигналу ($B = 128$)

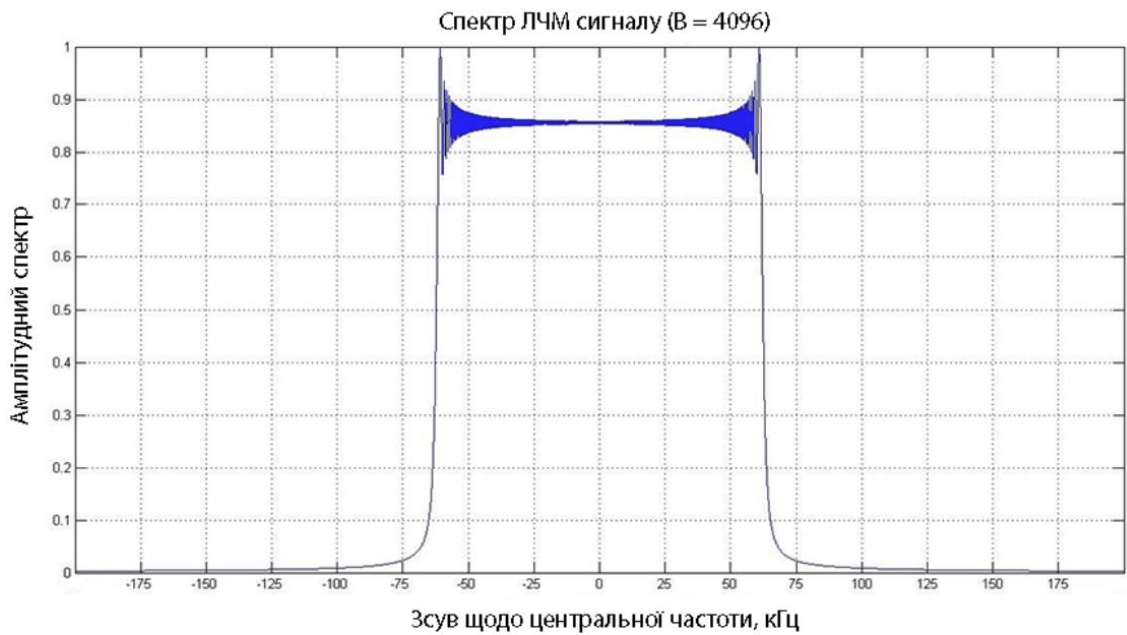


Рисунок 2.3 – Спектр ЛЧМ сигналу ($B = 4096$)

Передавачі LoRa формують CSS радіосигнали з шириною спектра (BW) 125, 250 або 500 кГц. За фіксованої ширини спектра радіосигналу BW зміна його бази здійснюється завдяки зміні тривалості T_{sym} і швидкості зміни частоти μ (рис. 2.4).

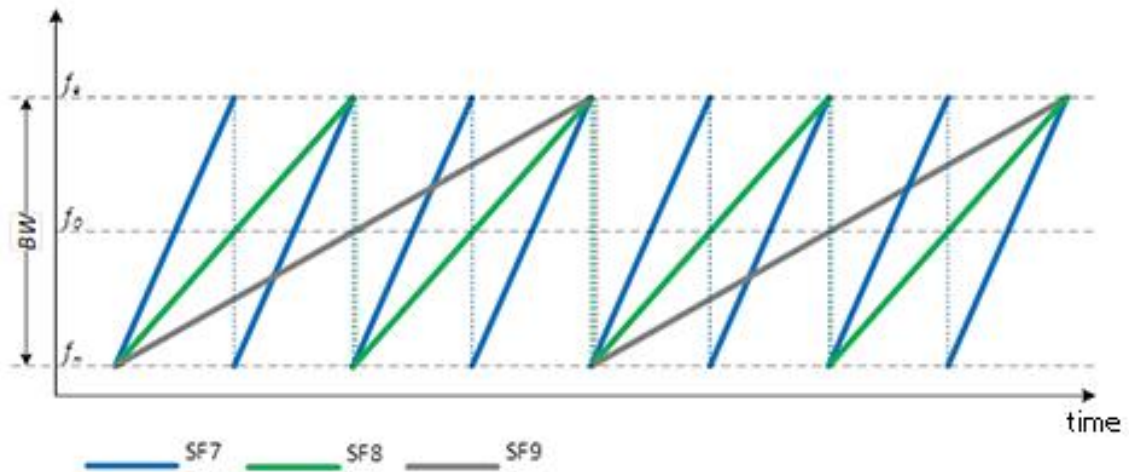


Рисунок 2.4 – Залежність SF від часу та ширини радіосигналу

Spreading Factor (SF) – коефіцієнт розширення спектру. Чим менший SF, тим менший час треба на передачу одного символу, тим вища швидкість передачі даних. Головна ціль – це передавати сигнал на більшу дистанцію, але з меншою швидкістю. Також чим більший SF, тим менший вплив перешкод в радіоканалі.

2.2 Передача пакету

Для успішного функціонування будь-якої системи обміну інформацією необхідна взаємна синхронізація приймача і передавача, що дає змогу визначити тимчасові межі приймання-передавання як цілого блоку даних, так і одиничних символів [12].

Технологія LoRa використовує асинхронний режим приймання та передачі, за якого передавач може почати генерацію радіосигналу в будь-який момент часу. У цьому разі потрібен механізм, що забезпечує синхронізацію приймача за сигналом від передавача (аналог «старт-біта» протоколу RS232). Як такий механізм використовується преамбула, що передує кожному сеансу зв'язку. Преамбула містить послідовність символів, що дають змогу приймачу виявити активність передавача, визначити коефіцієнт розширення спектра (SF), який використовує передавач, і виконати символну синхронізацію. Тривалість преамбули є величиною, яку можна конфігурувати і має бути не меншою за $T_1 + 2 \times T_2$, де T_1

визначає максимальний час знаходження приймача в стані «сну» (Sleep), T_2 - визначає час пошуку приймачем преамбули (рис. 2.5).

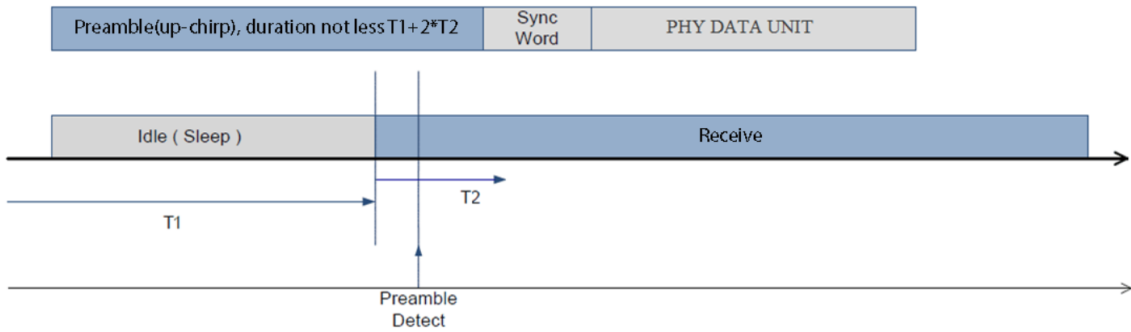


Рисунок 2.5 – Синхронізація приймача та передавача

Після завершення преамбули слідує слово синхронізації (англ. Sync Word) і блок даних фізичного рівня. Довжина слова синхронізації налаштовується в діапазоні від 1 до 8 байт. Специфікацією LoRa визначено низку специфічних значень Sync Word – 0x34 для публічних мереж (англ. public networks), 0x12 – для приватних мереж (англ. private networks) і 0xC194C1 - для каналів із FSK модуляцією.

На рис. 2.6 наведено загальну структуру кадру, що забезпечує передачу одного блоку даних.

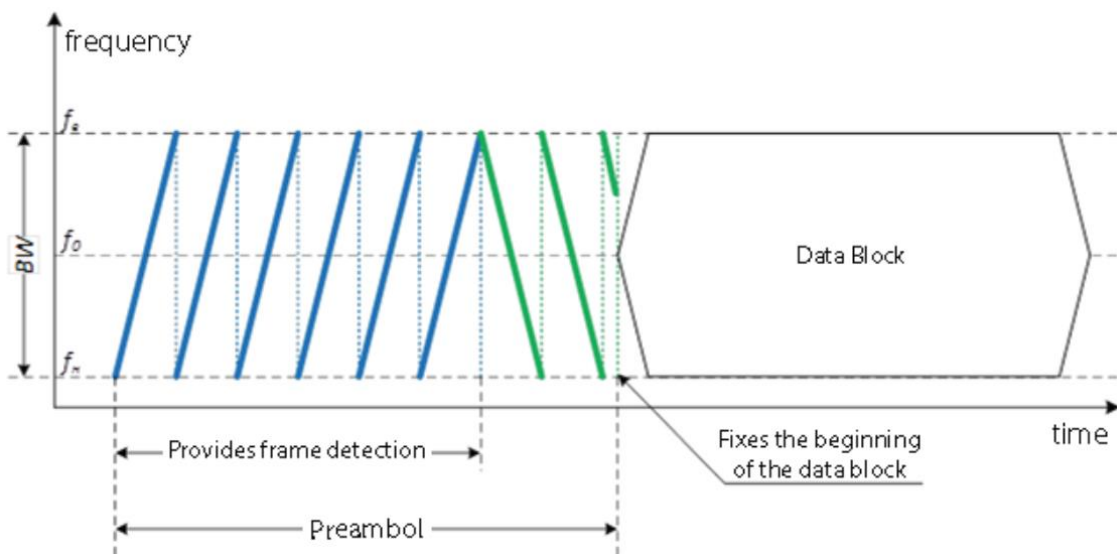


Рисунок 2.6 – Структура кадру

Механізм функціонування детектора преамбули ґрунтується на використанні узгодженого фільтра, чия імпульсна характеристика комплексно пов'язана з CSS радіосигналом у частотній ділянці та має дзеркальне відображення його в часі:

$$h(t) = A_1 \times \cos \cos \left(\omega_0 \times (T_{sym} - t) - \frac{\mu}{2} \times (T_{sym} - t)^2 \right), \quad 0 \leq t < T_{sym}$$

Принцип передавання символів інформації блоку даних фізичного рівня (PHY DATA UNIT) за допомогою широкосмугового радіосигналу LoRa полягає в частотному зсуві $e^{j \times \Delta\omega \times k \times t}$ відносно опорного ЛЧМ радіосигналу $e^{j(\omega_n \times t + \mu \times t^2)}$, де $k = 0, 1, 2, \dots, 2^{SF}$ – інформаційний символ, розмірністю SF біт (рис. 2.7) [10]:

$$x(t) = \begin{cases} A_0 \times \cos \left(\omega_0 \times t + \Delta\omega \times k \times t + t \frac{\mu}{2} \times t^2 \right), & 0 \leq t < T_0 \\ A_0 \times \cos \left(\omega_0 \times t + \Delta\omega \times k \times -BW \times t + t \frac{\mu}{2} \times t^2 \right), & T_0 \leq t < T_{sym} \end{cases}$$

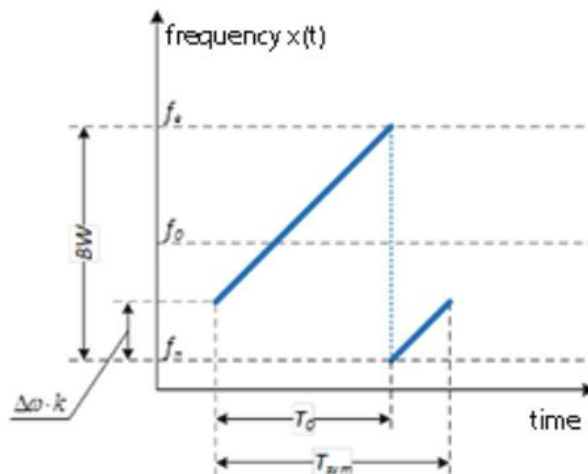


Рисунок 2.7 – Принцип передавання символів інформації блоку даних фізичного рівня

Приклад залежності частоти радіосигналу від часу для LoRa-кадру показано на рис. 2.8.

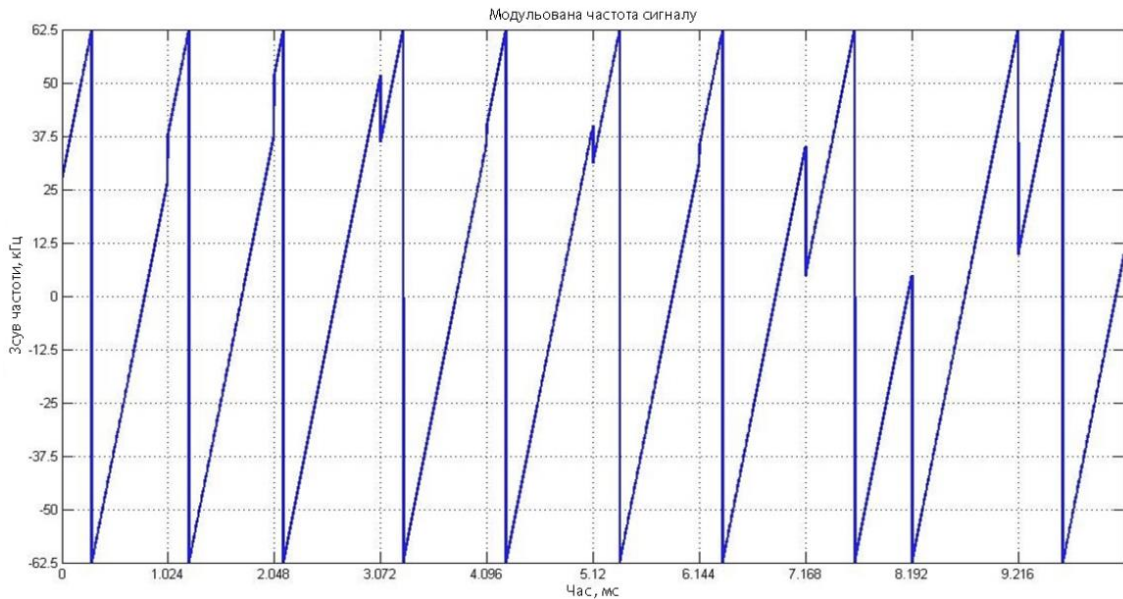


Рисунок 2.8 – Приклад залежності частоти радіосигналу від часу для LoRa

2.3 Характеристики радіо інтерфейсу LoRa

Основні характеристики радіо інтерфейсу LoRa:

- зведені дані за різними регіональними частотними діапазонами (табл. 2.1);
- дані щодо швидкості передавання інформації, розрахованої за формулою $R_b = SF \times BW / 2^{SF}$, для різних швидкостей кодування (табл. 2.2);
- дані щодо допустимих швидкостей передачі даних (англ. Data Rate – DR) (табл. 2.3);
- дані щодо сумарної агрегованої швидкості одного радіоканалу з урахуванням того, що шлюз LoRa має змогу в одному радіочастотному каналі одночасно приймати сигнали від декількох кінцевих пристроїв, які передають сигнал із різними коефіцієнтами розширення спектра (табл. 2.4);
- параметри регулювання потужності передавача кінцевого пристрою за командою від мережевого сервера (табл. 2.5).

Таблиця 2.1 – Зведені дані за регіональними частотними діапазонами

	Європа	Північна Америка
Частотний діапазон, МГц	863–870	902–928
Максимальна кількість каналів	35	64(UL)+8(UL)+8(DL)
Ширина спектра радіосигналу UL, кГц	125/250	125/500
Ширина спектра радіосигналу каналу DL, кГц	125	500
Модуляція	LORA, GFSK, MSK	LORA, GFSK, MSK
Потужність передачі UL, дБм	2–14 (20)	10–28
Потужність передачі UL, мВт	1–25 (100)	1–630
Потужність передачі DL, дБм	14	27
Фактор розширення спектра SF	7–12	7–10

Таблиця 2.2 – Дані про швидкості передавання інформації

SF	7	8	9	10	11	12
W, kHz	125	125	125	125	125	125
Rb, CR=1 (без кодування FEC), біт/с	6835,94	3906,25	2197,27	1220,70	671,39	366,21
Rb, CR=4/5, біт/с	5468,75	3125,00	1757,81	976,56	537,11	292,97
Rb, CR=4/6, біт/с	4557,29	2604,17	1464,84	813,80	447,59	244,14
Rb, CR=4/7, біт/с	3906,25	2232,14	1255,58	697,54	383,65	209,26
Rb, CR=4/8, біт/с	3417,97	1953,13	1098,63	610,35	335,69	183,11

Таблиця 2.3 – Дані щодо допустимих швидкостей передачі даних

Швидкість передачі даних	Конфігурація (SF/W)	Швидкість передавання даних на фізичному рівні, біт/с
DR0	LoRa: SF12 / 125 кГц	183–293
DR1	LoRa: SF11 / 125 кГц	335–537
DR2	LoRa: SF10 / 125 кГц	610–976
DR3	LoRa: SF9 / 125 кГц	1098–1757
DR4	LoRa: SF8 / 125 кГц	1953–3 25
DR5	LoRa: SF7 / 125 кГц	3417–5468
DR6	LoRa: SF7 / 250 кГц	6835–10937
DR7	FSK: 50 kbps	50000

Таблиця 2.4 – Дані щодо сумарної агрегованої швидкості

Rb, CR=1 (без кодування FEC), біт/с	15197,76
Rb, CR=4/5, біт/с	12158,20
Rb, CR=4/6, біт/с	10131,83
Rb, CR=4/7, біт/с	8684,42
Rb, CR=4/8, біт/с	7598,88

Таблиця 2.5 – Параметри регулювання потужності

TXPower	Потужність передавача, дБм
0	20
1	14
2	11
3	8
4	5
5	2

З даних таблиць можна побачити залежність різних параметрів від інших. Для подальшого калібрування системи необхідно враховувати характерні залежності технології передачі LoRa.

2.4 Робота з фізичним рівнем LoRa

Для вирішення різних завдань і застосувань у мережі LoRaWAN передбачено три класи кінцевих пристроїв:

Двонаправлені кінцеві пристрої «класу А» (англ. Bidirectional end-devices, Class A). Кінцеві пристрої «класу А» дають змогу організувати двонаправлений обмін. Причому зв'язок може ініціювати тільки кінцевий пристрій, після чого виділяються два часових вікна, протягом яких очікується відповідь від мережі. Інтервал передачі планується кінцевим пристроєм на основі власних потреб у зв'язку з невеликими випадковими часовими флуктуаціями (протокол типу ALOHA). Кінцеві пристрої «класу А» застосовують у застосунках, де передавання даних від мережі можливе тільки як відповідна реакція на отримання даних від

кінцевого пристрою і потрібен максимальний час роботи від автономного джерела живлення [14].

Двонаправлені кінцеві пристрої «класу В» (англ. Bidirectional end-devices, Class B) на додачу до функцій пристроїв «класу А», відкривають додаткові вікна приймання за розкладом. Для того, щоб відкрити вікно прийому, кінцевий пристрій синхронізується за спеціальними сигналами від шлюзу (за маяками - англ. Beacon). Це дає змогу мережі знати час, коли кінцевий пристрій готовий приймати дані.

Двонаправлені кінцеві пристрої «класу С» з максимальним приймальним вікном (англ. Bidirectional end-devices, Class C). Кінцеві пристрої «класу С» мають майже безперервно відкрите вікно приймання. Приймальне вікно закривається тільки на час передавання даних. Цей тип кінцевих пристроїв підходить для завдань, коли необхідно отримувати великі обсяги даних і не потрібна тривала робота від автономного джерела живлення. На рис. 2.9 показано стек протоколів технології LoRa [15].

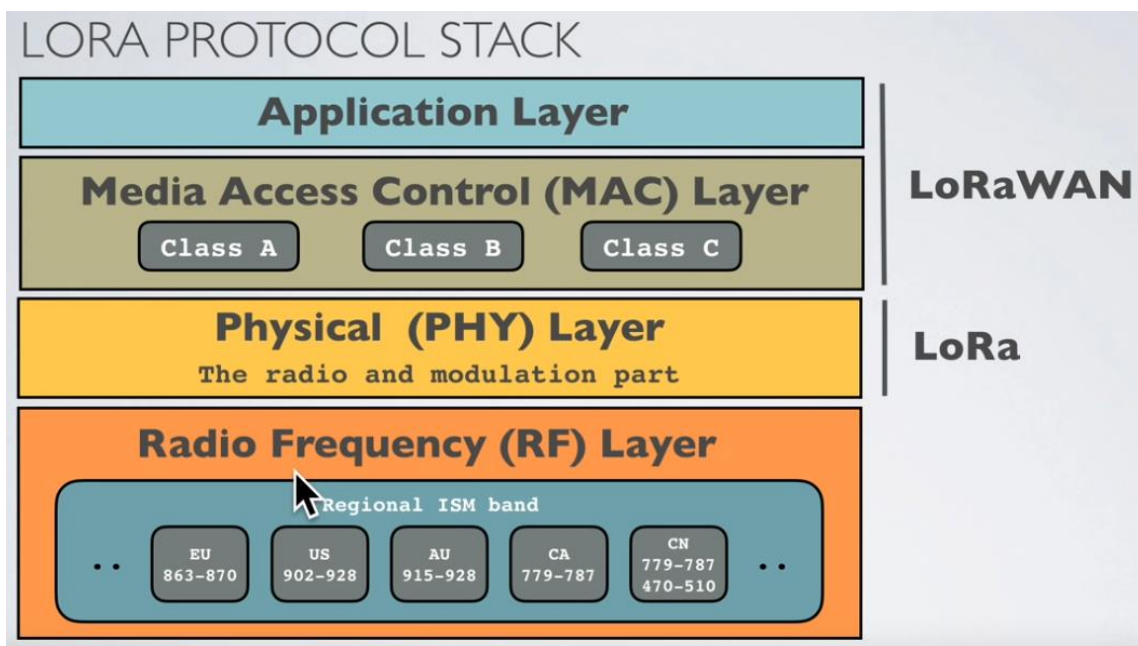


Рисунок 2.9 – Стек протоколів технології LoRa

На фізичному рівні забезпечується негарантована передача блоків даних між кінцевим пристроєм (англ. End Node) і шлюзом LoRa (англ. Gateway).

На стороні передавача виконується:

- приймання блоку даних від MAC рівня (PHYPayload);

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

- формування фізичного заголовка пакета (PHDR + PHDR_CRC);
- кодування фізичного заголовка пакета (PHDR + PHDR_CRC) з фіксованою швидкістю 4/8;
- обчислення контрольної суми блоку корисних даних PHYPayload (CRC);
- кодування блоку корисних даних (PHYPayload + CRC) з попередньо встановленою швидкістю CR;
- передача по радіоканалу преамбули;
- модуляція і передача по радіоканалу фізичного блоку даних.

На стороні приймача виконується:

- виявлення преамбули і визначення початку фізичного блоку даних;
- демодуляція сигналу;
- декодування фізичного заголовка пакета (PHDR + PHDR_CRC) і перевірка його контрольної суми;
- декодування блоку корисних даних (PHYPayload + CRC) і перевірка його контрольної суми;
- підтвердження прийнятих даних (для відповідних типів повідомлень);
- передача даних на MAC рівень.

На рис. 2.10 нижче наведено формати фізичних блоків даних низхідного (DL) і висхідного (UL) каналів:

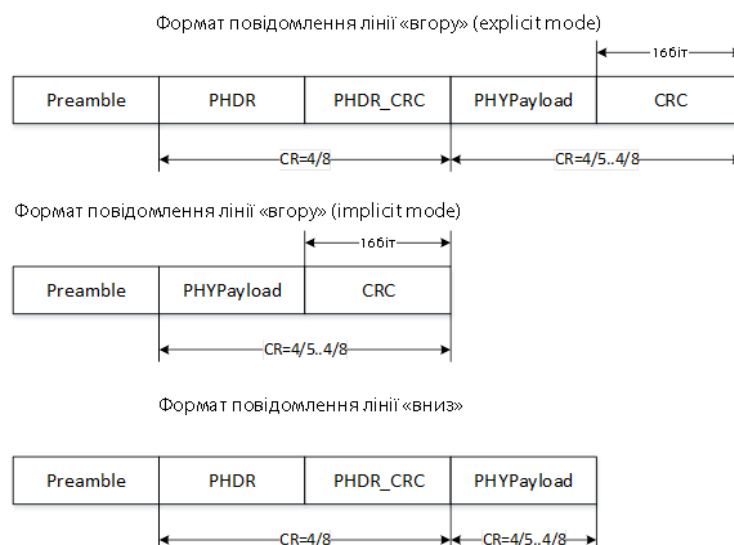


Рисунок 2.10 – Передача блоків даних між кінцевим пристроєм (End Node) і шлюзом LoRa (Gateway)

Preamble – преамбула, яка використовується для синхронізації приймача з вхідним потоком і визначення початку фізичного блоку даних. Довжина преамбули для чипа Semtech SX1272 є програмованою.

PHDR – фізичний заголовок пакета. Присутній тільки при використанні явного режиму (англ. explicit mode) і містить:

- довжину корисного навантаження в байтах;
- швидкість кодування;
- наявність у фізичному блоці даних опціонального поля CRC.

У разі використання неявного режиму (англ. implicit mode) фізичний заголовок пакета не передається, і пристрої працюють із попередньо встановленими параметрами.

PHDR_CRC – контрольна сума поля PHDR.

PHYPayload – корисне навантаження (блок даних, отриманий від рівня MAC / переданий на рівень MAC).

CRC – контрольна сума поля PHYPayload (опціональне поле).

При цьому заголовок PHDR кодується надлишковим кодом з фіксованою швидкістю 4/8; корисне навантаження – з програмованою швидкістю.

Для пристроїв класу «А» після завершення кожного сеансу передавання даних кінцевим пристроєм (у лінії «вгору») відкриваються два коротких часових вікна, протягом яких кінцевий пристрій може прийняти дані від мережі (у лінії «вниз»). Водночас протягом тривалості вікон приймання передавання даних кінцевим обладнанням заборонено [16].

Для передавання даних шлюзом LoRa в першому часовому вікні (RX1) використовуються ті самі параметри передавання (включно з номером частотного каналу і швидкістю передавання даних), які використовувалися для передавання даних кінцевим пристроєм.

Для передавання даних шлюзом LoRa у другому часовому вікні (RX2) використовуються попередньо встановлені параметри передавання (включно з номером частотного каналу та швидкістю передавання даних).

Тривалість тимчасового вікна встановлюється заздалегідь і має бути достатньою для приймання преамбули. Якщо мережі потрібно передати кінцевому пристрою (англ. End Node) більше інформації, ніж може бути передано в рамках одного вікна прийому (англ. receive window), LoRa gateway (шлюз) запитує у End Node виділення йому додаткового вікна, шляхом встановлення біта FPending заголовка MAC рівня. У цьому випадку End Node повинен виконати передачу up-link повідомлення (в т.ч. порожнього, якщо відсутні корисні дані для передачі), після закінчення якого і будуть відкриті додаткові вікна для отримання даних від мережі (рис. 2.11).

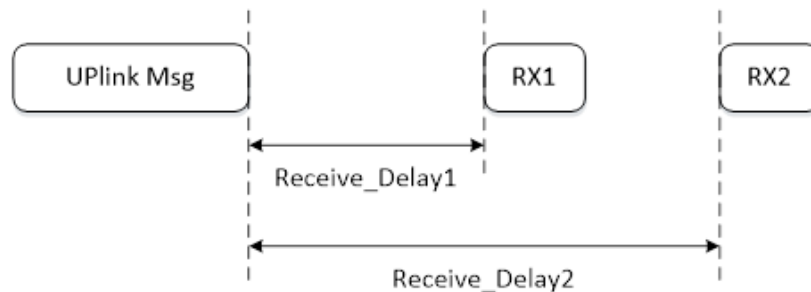


Рисунок 2.11 – Вікно прийому інформації

У технології LoRa передбачено механізми адаптації швидкості передавання даних кінцевих пристроїв для того, щоб оптимізувати завантаження мережі та забезпечити кожному кінцевому пристрою можливість роботи на максимальних швидкостях, які забезпечують належну стійкість до завад у тих радіоумовах, у яких цей пристрій перебуває.

Адаптацію швидкості передачі даних кінцевих пристроїв виконує мережевий сервер за допомогою відповідних MAC команд. Рішення про вибір тієї чи іншої швидкості приймається на підставі оцінки якості прийнятого від End Node сигналу.

Механізми адаптації швидкості доречно використовувати тільки на пристроях, місце розташування яких постійне і не змінюється згодом (статичні пристрої), тому що для таких пристроїв і радіоумови загалом будуть досить стабільні від одного сеансу зв'язку до іншого. На мобільних пристроях, наприклад, встановлених на автомобілях, тваринах тощо радіоумови між сеансами зв'язку змінюються непередбачувано. Отже, на таких пристроях доречно використовувати

постійні (встановлені за дефолтом) швидкості передачі. Статичні пристрої повинні ініціювати використання мережею режиму адаптації за допомогою встановлення ADR біта заголовка MAC-рівня.

Якщо кінцевий пристрій використовує швидкість передавання даних, вищу за попередньо встановлену дефолтну швидкість (відповідно до команди MAC рівня, отриманої від мережевого сервера), він має періодично контролювати факт одержання мережею повідомлень (навіть у разі використання режиму передавання без підтвердження) відповідно до такої процедури:

- кінцевий пристрій (End Node) інкрементує лічильник `ADR_ACK_CNT` під час кожного переданого у висхідному каналі повідомленні (UL-Msg) та скидає його під час отримання вхідного повідомлення низхідним каналом (DL-Msg) у вікні прийому (receive window);

- у разі досягнення лічильником `ADR_ACK_CNT` порога `ADR_ACK_LIMIT` кінцевий пристрій (за допомогою встановлення біта `ADRACKReq`) запитує мережу надіслати йому будь-який DL-Msg, підтвердивши тим самим, що повідомлення від цього кінцевого пристрою досягають мети; підтвердження повинне бути надіслане у вікні приймання одного з наступних UL-Msg (але не більше, ніж задано порогом `ADR_ACK_DELAY`);

- за відсутності підтвердження кінцевий пристрій знижує швидкість передачі на один крок;

- подальше зниження швидкості передавання на один крок відбуватиметься після передавання кожних `ADR_ACK_LIMIT` UL-Msg до отримання підтвердження, або до досягнення попередньо встановленої дефолтної швидкості.

Функціональність класу «B» оптимізована для мобільних і стаціонарних кінцевих пристроїв, що працюють від автономних джерел живлення.

Кінцеві пристрої здійснюють роботу в класі «B» за наявності запитів на відкриття приймальних вікон у фіксовані інтервали часу для можливості забезпечення надсилання повідомлень у DL каналі на вимогу сервера.

Для мережі з підтримкою кінцевих пристроїв класу «В» усі шлюзи мають синхронно передавати сигнал маяка (beacon), забезпечуючи відліки синхронізації для кінцевих пристроїв. Ґрунтуючись на цих відліках синхронізації, кінцеві пристрої можуть періодично відкривати прийомні вікна (ping slots), які можуть використовуватися мережевою інфраструктурою для ініціювання передавання каналом DL. Ініційоване мережею передання в DL каналі в одному з прийомних вікон (ping slot), називається «ping». Шлюз, призначений для передавання даних у DL каналі, обирається мережевим сервером, ґрунтуючись на індикаторах якості сигналу кінцевого пристрою в UL каналі (у межах останнього сеансу зв'язку). У зв'язку з цим, під час переміщення кінцевого пристрою та виявлення змін ідентифікатора в маяку, що приймається, кінцевий пристрій має надіслати повідомлення по UL каналу мережевому серверу, щоб сервер оновив базу даних шляху маршрутизації каналу DL.

Кінцеві пристрої класу «С» – це пристрої з постійно відкритим приймальним вікном у DL каналі. Такі пристрої використовуються у випадках, коли є достатня ємність по живленню і в зв'язку з цим не потрібно обмежувати час роботи приймача. Кінцеві пристрої класу «С» не можуть працювати в режимі класу «В».

Кінцеві пристрої класу «С» постійно слухають радіоефір з параметрами вікна RX2, за винятком випадків, коли пристрій передає дані або відкриває приймальне вікно RX1 відповідно до опису класу «А». Таким чином, приймальне вікно з параметрами RX2 відкривається кінцевим пристроєм:

- між закінченням передачі в UL каналі та початком приймального вікна RX1;
- після закриття приймального вікна RX1 і аж до початку передачі в UL каналі.

Специфікація LoRa не визначає конкретних повідомлень кінцевого пристрою для інформування мережевого сервера про те, що пристрій працює в режимі класу «С». Тільки сервер додатків знає, що кінцевий пристрій працює в класі «С» на підставі минулої процедури активації.

Висновки до розділу 2

Технологія LoRa являє собою ефективний спосіб бездротового зв'язку, який забезпечує передачу даних на великі відстані за мінімального енергоспоживання. Завдяки використанню модуляції CSS, LoRa характеризується високою чутливістю приймача, стійкістю до перешкод і точністю часової синхронізації, що дозволяє працювати навіть у складних умовах. Технологія оптимізована для передачі невеликих обсягів даних, таких як показники сенсорів, що робить її ідеальною для застосувань у сфері IoT, зокрема в аграрному секторі та для моніторингу параметрів оточуючого середовища.

Фізичний рівень LoRa побудований на використанні широкосмугових радіосигналів з лінійною частотною модуляцією (ЛЧМ), які забезпечують високу завадостійкість і низьку швидкість передачі даних, що компенсується можливістю роботи на великих відстанях (до 15–20 км у сільській місцевості). Технологія також підтримує гнучке налаштування параметрів передачі, таких як ширина спектра, коефіцієнт розширення спектра (SF) і потужність сигналу, що дозволяє адаптувати її до конкретних потреб мережі.

Асинхронний режим роботи з використанням преамбули і слова синхронізації забезпечує надійну синхронізацію між передавачем і приймачем. Три класи кінцевих пристроїв (A, B і C) дозволяють оптимізувати енергоспоживання і режими роботи відповідно до вимог конкретних застосувань. Високий рівень налаштувань і підтримка широкого спектра регіональних частот робить LoRa універсальним вибором для побудови масштабованих IoT-мереж.

3 ПОБУДОВА АПАРАТНО-ПРОГРАМНОЇ СИСТЕМИ

3.1 Вибір апаратного забезпечення системи

Розподілена система моніторингу параметрів навколишнього середовища складається з двох основних компонентів: вузлів датчиків (нод) та центрального шлюзу (хабу). Обидві частини працюють у тісній взаємодії, забезпечуючи збір, та передачу даних за певним розкладом.

Шлюз є точкою збору даних від усіх вузлів у радіусі покриття. Він забезпечує зв'язок із зовнішньою мережею для подальшої обробки та збереження інформації.

Основою для розробки розподіленої системи моніторингу параметрів оточуючого середовища є технологія передачі даних на великі відстані LoRa (рис. 3.1).



Рисунок 3.1 – LoRa Ra-01 на базі чипа SX1278

LoRa Ra-01 – це бездротовий модуль для довготривалої передачі даних, розроблений компанією Ai-Thinker на базі чипа SX1278 від Semtech. Він підтримує технологію LoRa, що забезпечує високу дальність зв'язку до 10 км і більше в умовах прямої видимості при мінімальному енергоспоживанні. Модуль працює в частотному діапазоні 433–525 МГц, що робить його придатним для використання в регіонах із дозволеним застосуванням цієї частоти [25].

Ra-01 використовує метод модуляції LoRa, який забезпечує високу стійкість до перешкод, дозволяючи стабільно працювати в середовищах із високим рівнем шуму. Завдяки цьому він підходить для систем IoT, де надійність і енергоефективність є ключовими параметрами. Модуль підтримує кілька режимів

роботи, зокрема передачу даних з використанням стандартів LoRaWAN або власних протоколів.

Модуль використовує інтерфейс SPI, напівдуплексний зв'язок, з CRC, до 256-байтним механізмом пакетів. Також має компактний розмір, але потребує зовнішньої антени для оптимальної роботи, що підключається через паяльний контакт. Живлення здійснюється напругою 3,3 В, максимальний вихід +22 дБм, максимальний робочий струм 140 мА, а енергоспоживання залежить від обраного режиму, залишаючись низьким під час очікування та помірним у режимі передачі даних.

Головним ядром шлюзу є ESP E-12. Таке рішення було зумовлене необхідністю надати користувачу зручний доступ до налаштувань шлюзу та налагодження зв'язку між вузлами (рис 3.2).



Рисунок 3.2 – Багатофункціональний Wi-Fi модуль ESP E-12

Дана ESP – це багатофункціональний Wi-Fi модуль, розроблений на основі чипа ESP8266 від Espressif Systems. Він є одним із найпопулярніших рішень для проектів з підключенням до Інтернету завдяки своїй доступності, універсальності та підтримці великої кількості функцій. Чип ESP8266, який є ядром модуля, оснащений 32-бітним процесором Tensilica Xtensa L106, здатним працювати на частоті до 160 МГц. Він забезпечує достатню обчислювальну потужність для виконання завдань, пов'язаних з передачею даних через Wi-Fi та обробкою інформації на локальному рівні [26].

Модуль має вбудовану флеш-пам'ять обсягом 4 Мбайт, яка використовується для зберігання програмного забезпечення, бібліотек і необхідних даних. Він

підтримує повноцінний TCP/IP стек, що дозволяє йому працювати як клієнт або сервер, забезпечуючи передачу даних через Wi-Fi у стандарті 802.11 b/g/n. Завдяки підтримці режиму SoftAP модуль може створювати власну точку доступу, до якої можуть підключатися інші пристрої.

Однією з головних переваг ESP-12E є велика кількість доступних GPIO-виводів, що дозволяють підключати датчики, світлодіоди, реле та інші компоненти. Також підтримуються периферійні інтерфейси, такі як UART, SPI та I²C, що розширює можливості інтеграції в складні системи. Крім того, модуль оснащений вбудованою керамічною антеною, яка забезпечує стабільний прийом і передачу сигналу, і екрануванням, що зменшує електромагнітні перешкоди.

ESP-12E відзначається компактними розмірами, що робить його зручним для інтеграції в пристрої з обмеженим простором. Він підтримує напругу живлення 3,3 В і працює з помірним енергоспоживанням, що є важливим для IoT-проектів з автономним живленням.

Для передачі отриманого пакету даних кінцевому отримувачу використано GSM модуль. Хоч LoRa може передавати інформацію на великі відстані, але вона може комунікувати в рамках однакових технологічних модулів.

SIM800L – це компактний GSM/GPRS модуль, розроблений для забезпечення мобільного зв'язку пристроїв із мережею стільникового зв'язку (рис. 3.3). Він підтримує передачу голосу, SMS, даних і навіть інтернет-з'єднання через мережі 2G, що робить його універсальним рішенням для IoT-проектів, де потрібне надійне з'єднання в реальному часі. Завдяки підтримці частот 850/900/1800/1900 МГц модуль працює в більшості країн світу, забезпечуючи глобальну сумісність[27].

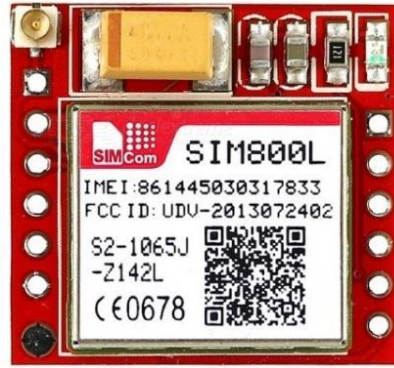


Рисунок 3.3 – GSM/GPRS модуль SIM800L

Цей модуль вирізняється невеликими розмірами (приблизно 25 мм x 23 мм), що робить його ідеальним для інтеграції в компактні пристрої. Він може передавати дані на швидкості до 85,6 кбіт/с (GPRS Class 12) і підтримує управління через стандартний набір AT-команд, що забезпечує простоту у використанні з мікроконтролерами та іншими системами. Інтерфейс UART дозволяє легко інтегрувати модуль до вже обраного контролера.

SIM800L також підтримує широкий набір додаткових функцій, включаючи визначення геолокації на основі стільникових веж (LBS), що дозволяє отримувати приблизні координати без використання GPS, але дану функція використовуватись не буде. Він оснащений виходом для підключення зовнішньої антени через інтерфейс IPEX або паяльний контакт, що дає гнучкість у виборі антенного рішення залежно від умов використання.

Модуль працює з живленням у діапазоні від 3,4 В до 4,4 В і вимагає стабільного джерела енергії, здатного забезпечити пікові струми до 2 А під час роботи з мережею.

Він має широкий діапазон робочих температур (від мінус 40 °С до + 85 °С), що дозволяє використовувати його в промислових та екстремальних умовах. Завдяки доступній ціні та простоті інтеграції SIM800L є популярним вибором для створення рішень, що потребують надійного зв'язку через GSM, як-от смарт-системи, сигналізації, автоматизовані системи збору даних і моніторинг у віддалених регіонах.

Для забезпечення автономної роботи системи моніторингу протягом тривалого часу можна використовувати **сонячну батарею** як основне джерело енергії. Вона дозволяє перетворювати сонячне світло на електричну енергію, яка забезпечує роботу всіх компонентів системи. Для зберігання енергії, особливо у періоди з низькою сонячною активністю (ніч, похмура погода), слід використовувати **акумулятор**. Його вибір залежить від енергоспоживання системи, необхідного запасу енергії, а також умов експлуатації. Оптимальне поєднання сонячної батареї та акумулятора забезпечує безперервну роботу системи незалежно від зовнішніх чинників.

Головна плата вузла

Для вузлу датчиків було обрано мікроконтролер CH32V003F4P6.

CH32V003F4P6 – це надзвичайно компактний і доступний 32-бітний мікроконтролер, створений компанією WCH (Nanjing Qinheng Microelectronics). Він базується на відкритій архітектурі RISC-V, що робить його гнучким і незалежним від пропрієтарних рішень (рис. 3.4). Мікроконтролер розрахований на роботу з частотою до 48 МГц, що забезпечує достатню продуктивність для широкого спектра задач. Він оснащений 16 КБ флеш-пам'яті для зберігання програмного коду та 2 КБ оперативної пам'яті (SRAM) для тимчасового зберігання даних під час виконання програм [28].



Рисунок 3.4 – Мікроконтролер CH32V003F4P6

CH32V003 підтримує основні периферійні інтерфейси, включаючи UART, SPI та I²C, що дозволяє легко інтегрувати його у проекти, пов'язані з передачею даних і комунікацією. Завдяки наявності 10-бітного АЦП мікроконтролер може перетворювати аналогові сигнали в цифрову форму, що є важливим для роботи з

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN датчиками. Крім того, у нього є до 18 конфігурованих GPIO, які можна налаштувати для роботи як входи, виходи або спеціалізовані функції.

Працює пристрій при робочій напрузі 3,3 В і має низьке енергоспоживання, що робить його ідеальним для застосувань, де важлива енергоефективність. Мікроконтролер доступний у різних компактних корпусах, зокрема TSSOP20, QFN20, SOP16, SOP8, що підходить для компактних конструкцій та масового виробництва.

Завдяки низькій вартості CH32V003 є привабливим рішенням для бюджетних проектів, особливо в галузі IoT, де він може використовуватися для підключення простих датчиків або керування пристроями. Відкритість архітектури RISC-V дозволяє розробникам використовувати вільно доступні інструменти, такі як GCC та OpenOCD, що спрощує процес створення та налагодження програмного забезпечення.

Для вузлу використаний той самий модуль LoRa Ra-01 на базі чипа SX1278.

Для вимірювання показників вологості та температури повітря використано модуль SHT3x (рис. 3.5), а для вимірювання освітлення – BH1750 (рис. 3.6).



Рисунок 3.5 – Цифровий сенсор вологості та температури SHT3x

SHT3x – це серія високоточних цифрових сенсорів вологості та температури від компанії Sensirion. Вони забезпечують точні вимірювання завдяки використанню запатентованої технології CMOSens®. Сенсори здатні вимірювати відносну вологість у діапазоні від 0% до 100% з похибкою $\pm 2\%$, а температуру в межах від -40°C до $+125^{\circ}\text{C}$ з точністю до $\pm 0.2^{\circ}\text{C}$. Вони використовують цифровий інтерфейс I²C для передачі даних і підтримують декілька адрес, що дозволяє легко інтегрувати їх у багатосенсорні системи. Компактні розміри, низьке

енергоспоживання та висока стабільність роблять SHT3x ідеальними для застосувань у системах моніторингу клімату, побутовій техніці, IoT-рішеннях та промислових пристроях. Має широкий діапазон напруги живлення, від 2,4 В до 5,5 [29].



Рисунок 3.6 – Цифровий сенсор освітленості BH1750

BH1750 — це цифровий сенсор освітленості, розроблений для вимірювання рівня світла в люксах (lux). Він працює на основі технології фотодіодів і перетворювача струму в цифровий сигнал, забезпечуючи точне вимірювання освітленості в діапазоні від 1 до 65,535 люкс. Сенсор використовує інтерфейс I²C, що дозволяє легко отримувати дані для обробки. BH1750 є енергоефективним і має можливість регулювання чутливості для точного вимірювання у різних умовах освітлення. Завдяки простоті інтеграції він широко використовується в системах автоматичного регулювання освітлення, мобільних пристроях, інтелектуальних будинках і фотоелектричних системах. Робоча напруга від 3 В до 5 В [30].

Модуль вологості та температури ґрунту

Щоб забезпечити вимірювання вологості ґрунту вирішено створити окремий модуль для запобігання значних похибок. В його основі лежить мікроконтролер CN32V003J4M6 того ж самого виробника. Такий вибір зумовлений відсутністю необхідності в великій кількості конфігурованих GPIO. Контролер має 8 пінів для роботи з UART технологією та забезпечення живлення.

Усі датчики мають бути аналоговими, оскільки один із них є частиною плати, а два інших у вигляді мікросхем були б занадто дорогими. Висока точність у цьому

випадку не потрібна, а враховуючи значну кількість пристроїв, це було б економічно невиправдано.

На противагу датчику вологості з відкритими електродами, за допомогою якого можна грубо вимірювати опір ґрунту, використано ємнісний датчик вологості, що запобігає швидкому виходу з ладу й розщепленню електродів і, відповідно, утворенню небажаних з'єднань. Звісно, він має бути захищений смолою або паяльна маска, що являє собою тонкий шар полімеру.

Для вимірювання температури ґрунту вирішено обрати мініатюрний термістор NCP18XH103F03RB типорозміру 0603. Термістор буде на самому кінці штиря, щоб вимірювання проводилися на глибині 10–15 см.

3.2 Проєктування апаратного забезпечення системи

Для розробки схем та моделей плат для системи моніторингу параметрів оточуючого середовища використовувався безкоштовний застосунок KiCad. Цей інструмент обрано через його відкритий код, широкий функціонал, підтримку великої кількості форматів файлів і доступність без додаткових витрат, що є важливим фактором при створенні економічно ефективних рішень.

Першим етапом було проєктування схем роботи для кожного апаратного елемента, які складають основу системи. Це дало змогу створити деталізовану структурну модель, яка враховує особливості функціонування кожного компонента. Основна увага приділялася оптимальному поєднанню апаратної частини з програмним забезпеченням для забезпечення максимальної продуктивності та надійності системи.

У процесі розробки було виділено три основні типи плат, кожна з яких виконує свою унікальну функцію:

1) **плата шлюзу** – центральний елемент системи, який забезпечує передачу даних між вузлами датчиків і сервером. Її функціонал включає прийом даних із

сенсорних вузлів через мережу LoRaWAN та їх подальшу обробку і передачу до хмарного сховища;

2) **головна плата вузла датчиків** – відповідальна за обробку даних із підключених сенсорів. Ця плата слугує інтерфейсом між окремими датчиками та бездротовим модулем, забезпечуючи коректну обробку сигналів перед їхньою передачею до шлюзу;

3) **модуль для вимірювання вологості та температури ґрунту** – спеціалізована плата, розроблена для підключення датчиків, що контролюють ключові параметри стану ґрунту. Вона забезпечує зчитування аналогових сигналів, їх перетворення та підготовку для подальшого використання системою.

Для кожної з цих плат у KiCad було створено принципову схему, яка відображає логіку їх роботи, зв'язки між компонентами та електричні параметри. Ці схеми є важливим етапом проєктування, адже вони слугують основою для подальшого виготовлення друкованих плат та тестування апаратної частини. Усі розроблені принципові схеми включено в додатки до дипломної роботи для детального ознайомлення [17-18].

Проєктування шлюзу:

Для шлюзу необхідно було створити окремі обв'язки стабілізаторів напруги через різні потреби компонентів. Наприклад, GSM-модуль потребує більшу напругу для роботи аніж LoRa та ESP. Додатково зроблено управління живленням для відключення не потрібних для постійної роботи компонентів, таких як LoRa та SIM800L, що вмикаються лише в процесі передачі та прийому інформації. Для автономного живлення шлюзу було розміщено роз'єм для сонячної панелі, що з'єднаний з DC/DC MPPT для того, щоб максимально ефективно витягувати енергію з панелі та заряджати акумулятор. Спілкування між ESP та SIM800L виконується через протокол UART. Спілкування між ESP та LoRa-модулем виконується через протокол SPI. Додатково виведений роз'єм для додаткових датчиків за протоколом I²C.

Спроєктована модель та структурна схема шлюзу наведені на рис. 3.7–3.9.

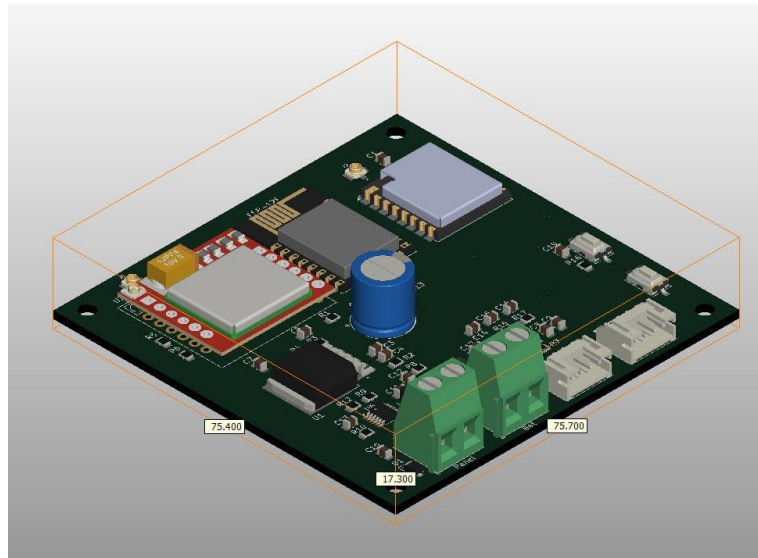


Рисунок 3.7 – 3D-вигляд плати шлюзу

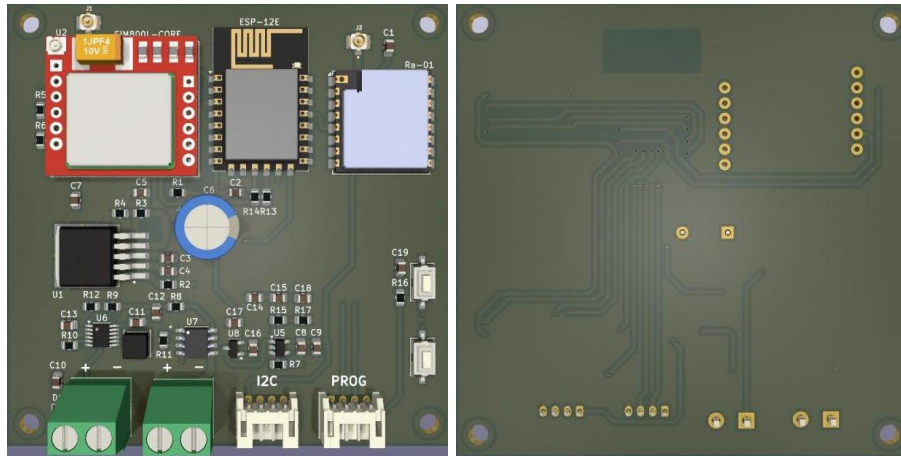


Рисунок 3.8 – Вигляд плати шлюзу зверху та знизу

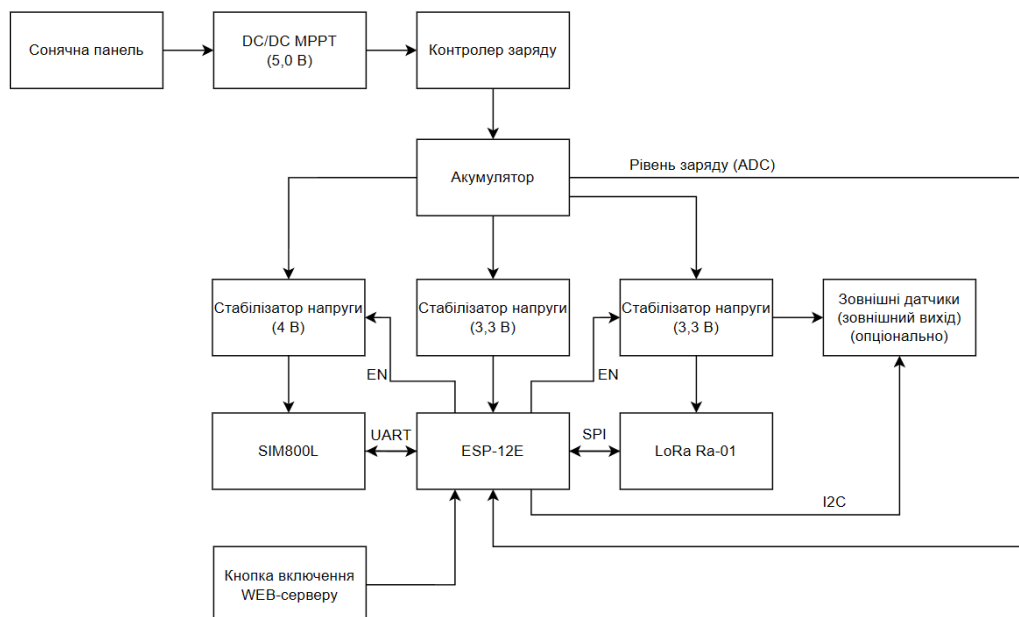


Рисунок 3.9 – Структурна схема плати шлюзу

Проектування вузлу датчиків:

Для вузлу датчиків було використано акумуляторне живлення, тобто акумулятор підключається через роз'єм та живить всю плату через окремий стабілізатор. Також на окрему ніжку мікроконтролера підключено вимірювання напруги на акумуляторі для зчитування заряду батареї. Контроль живлення датчиків контролюється мікроконтролером тому, що системі для економії заряду не потрібно працювати з датчиками і заживлювати їх. Спілкування між мікроконтролером та LoRa-модулем виконується через протокол SPI. За протоколом I²C виконується спілкування з датчиками температури та вологості повітря й освітленням. Для вимірювання вологості та температури ґрунту було використано протокол UART та вивід на окрему плату. Усі датчики та акумулятор мають свої роз'єми для модульності системи. Це дозволить в майбутньому модифікувати набір датчиків [11].

Спроектвана модель та структурна схема вузлу датчиків наведені на рис. 3.10–3.12.

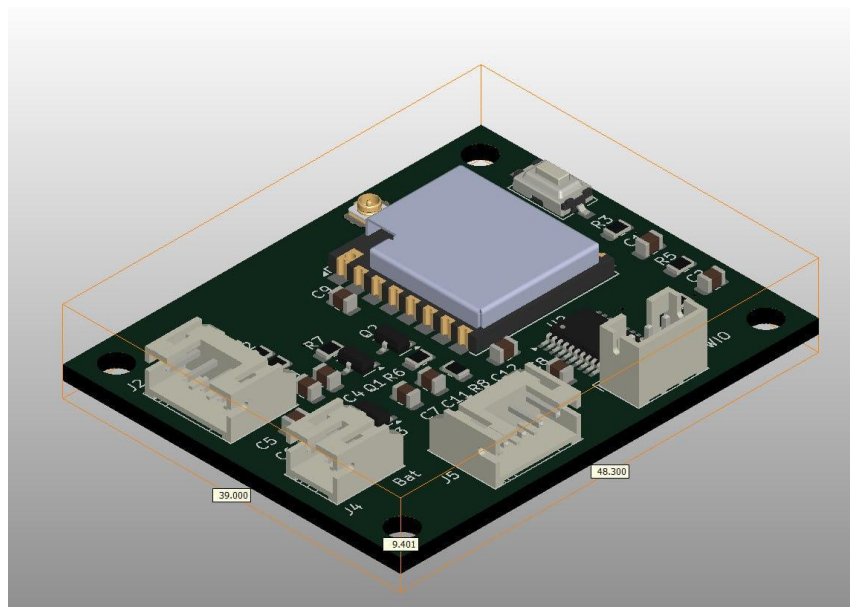


Рисунок 3.10 – 3D-вигляд головної плати вузлу

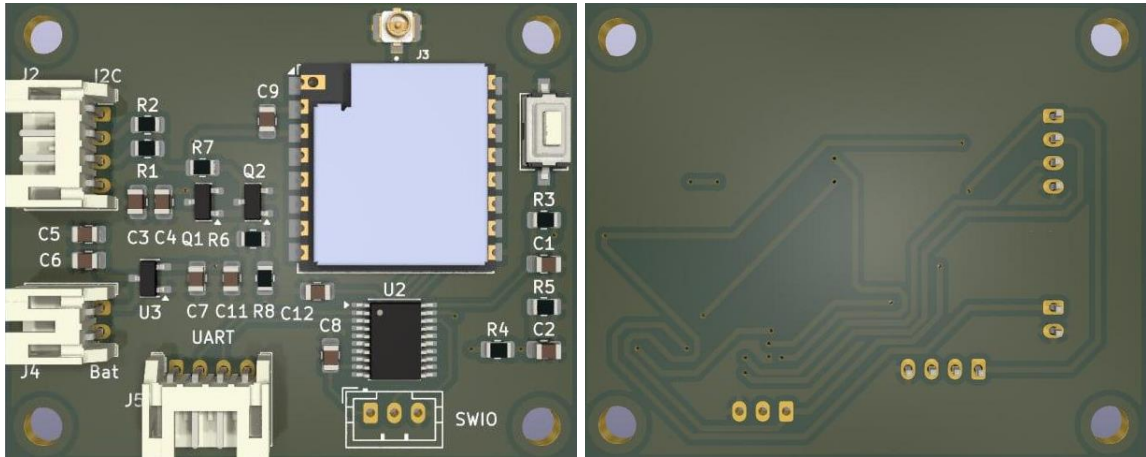


Рисунок 3.11 – Вигляд головної плати вузлу зверху та знизу

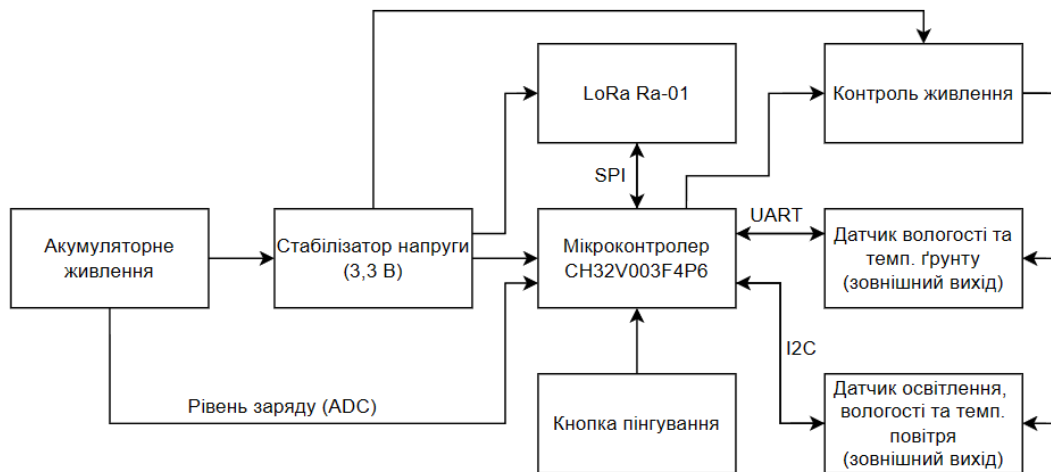


Рисунок 3.12 – Структурна схема головної плати вузлу

Проектування плати вимірювання вологості та температури ґрунту:

Для плати вимірювання вологості та температури ґрунту застосоване зовнішнє живлення, тобто дана плата не буде працювати без головної плати вузлу. Для зчитування показників стану ґрунту були використані аналогові апаратні рішення по типу двох пластин для конденсатора, на ємкість якого буде впливати зміна вологості, та термістора для виміру температури вглибині. Мікроконтролер генерує сигнал, що проходить через ємкісний елемент та обв'язку частотного фільтру й АМ-детектор. На виході ми будемо зчитувати падіння напруги з елементів (датчиків) і конвертує їх в відповідні показники вологості та температури. Повертати дані мікроконтролер буде через протокол UART [19-21].

Спроектвана модель та структурна схема плати вимірювання вологості та температури ґрунту наведені на рис. 3.13–3.15.

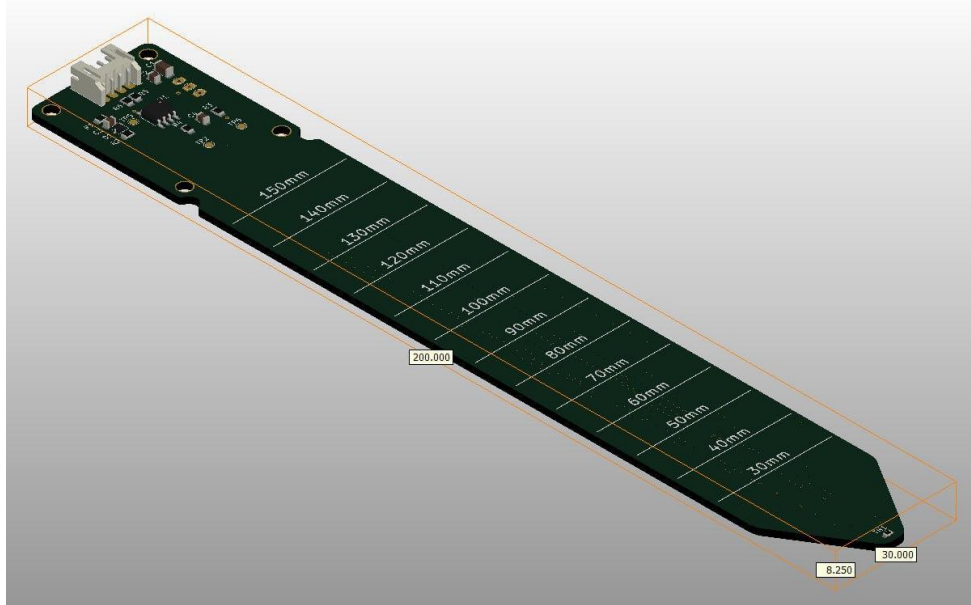


Рисунок 3.13 – 3D вигляд модуля вимірювання вологості ґрунту

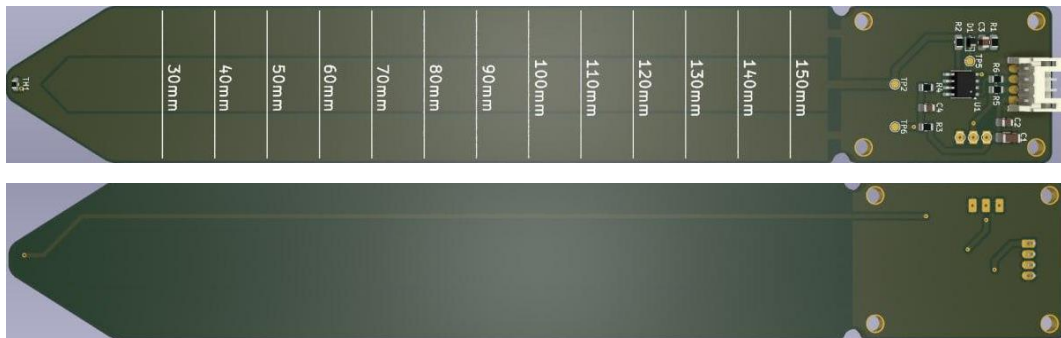


Рисунок 3.14 – Вигляд модуля вимірювання вологості ґрунту зверху та знизу

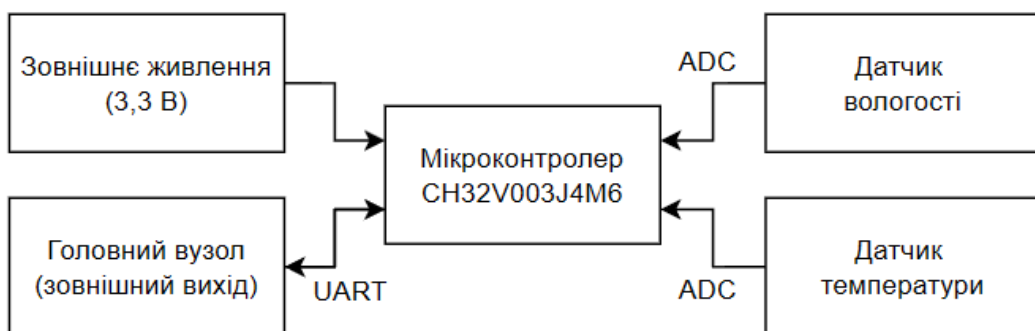


Рисунок 3.15 – Структурна схема модуля вимірювання вологості ґрунту

У фіналі отримано три друковані плати та схеми до них, що дозволить замовити виробництво готових плат в серійному масштабі та подальшого прототипування та розробки системи.

Висновки до розділу 3

Результатом виконаної роботи стало проектування апаратного комплексу розподіленої системи моніторингу параметрів навколишнього середовища, що складається з вузлів датчиків та центрального шлюзу. Було проаналізовано основні компоненти системи, такі як LoRa-модулі, Wi-Fi та GSM-модулі, а також сенсори для вимірювання параметрів середовища, зокрема температури, вологості та освітленості. Представлені моделі та структурні схема плат.

Розробка кожного з елементів базувалася на використанні енергоефективних, компактних і доступних компонентів, що забезпечують надійність і функціональність системи навіть у складних умовах експлуатації. Завдяки використанню відкритих стандартів, таких як LoRa та RISC-V, було досягнуто гнучкості в інтеграції різноманітних датчиків для вимірювання показників навколишнього середовища.

Система здатна забезпечити передачу даних на великі відстані з мінімальним енергоспоживанням, а підтримка автономної роботи завдяки сонячній енергії робить її придатною для застосування в польових умовах. Розроблені схеми та конструкції плат оптимізовані для економічного використання ресурсів та зручності масштабування.

Результати цього етапу створюють основу для подальшої інтеграції апаратного забезпечення із програмним, забезпечуючи функціональність та ефективність розподіленої системи моніторингу.

4 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ АПАРАТНО-ПРОГРАМНОЇ СИСТЕМИ

4.1 Реалізація апаратної частини шлюзу та вузлів

На основі розроблених принципових схем для шлюзу та вузлів датчиків, створених у програмному забезпеченні KiCad, було зібрано функціональні прототипи на макетній платі. Використання макетної плати дозволило швидко перевірити працездатність схеми без необхідності виготовлення друкованих плат на початковому етапі розробки.

Для збирання прототипів було обрано стандартні макетні плати, які забезпечують легке підключення елементів без пайки. Це дозволило зручно організувати компоненти й забезпечити можливість оперативного внесення змін у схему. Основними компонентами прототипів стали:

- контролери CH32V003F4P6 та плата розробки ESP8266: забезпечують обробку даних і управління всіма підключеними елементами;
- модулі LoRa Ra-01: відповідає за передачу даних через бездротову мережу на великі відстані;
- датчики SHT3x та BH1750: для вузлів датчиків використовувалися сенсори вологості та температури ґрунту, а також універсальні датчики для зчитування інших параметрів;
- додаткові компоненти: резистори, конденсатори та інші пасивні елементи для підтримки коректної роботи.

Для прототипу шлюзу було встановлено ESP8266 на макетній платі та з'єднано з модулями SIM800L та LoRaWAN через стандартні з'єднувальні дроти. Модуль LoRa підключено до контролера через SPI-інтерфейс, а SIM800L через UART [22]. Особливу увагу приділили правильності підключення живлення, адже GSM-модуль для роботи потребує окреме живлення. Усі підключення перевірялися мультиметром. Тестувалася передача даних між шлюзом і сервером через мережевий модуль. На рис. 4.1 наведений прототип готового шлюзу.

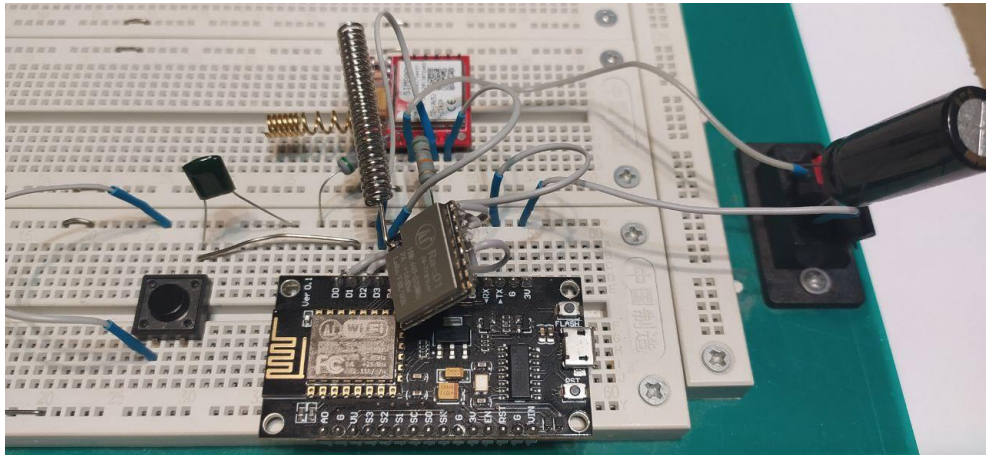


Рисунок 4.1 – Фото прототипу шлюзу

Якщо розглядати вузол датчиків, то для нього використаний мікроконтролер CN32V003F4P6 як центральний елемент вузла. До нього підключалися сенсори й LoRa-модуль. До вузла були підключені датчики для вимірювання вологості й температури ґрунту. Виходи сенсорів підключалися до відповідних пінів контролера. Датчики спілкуються з контролером через протокол I²C. На рис. 4.2 наведений прототип готового вузла датчиків.

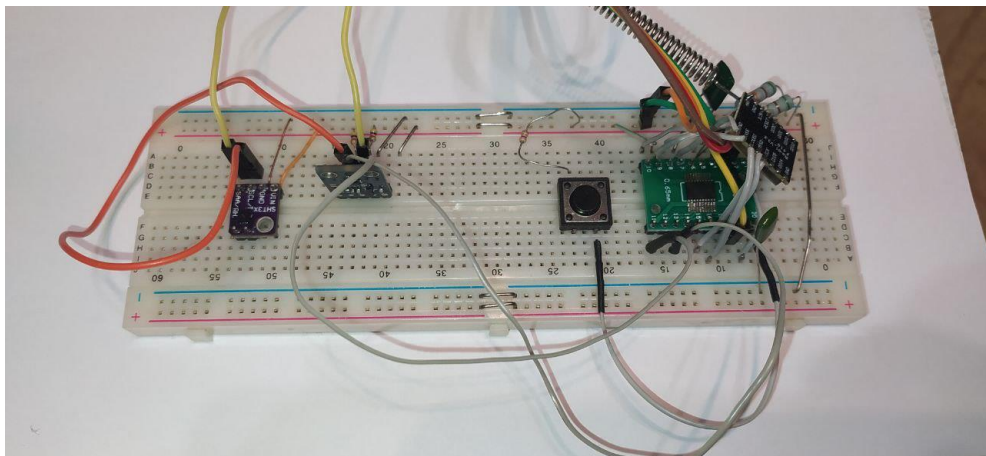


Рисунок 4.2 – Фото прототипу вузла датчиків

Ці фотографії наочно демонструють принципи збирання, розташування компонентів і загальний вигляд прототипів у робочому стані. Вони також ілюструють, як саме реалізована розроблена схема на практиці, що є важливим етапом підготовки до виготовлення друкованих плат.

4.2 Реалізація ємкісно-вимірювального модулю

Якщо взяти два шматки фольгового склотекстоліту і розташувати їх паралельно мідним площами всередину на невеликій відстані, то отримаємо конденсатор, що розташований вздовж однієї площини. Таким чином, плавно занурюючи цей конденсатор в ґрунт, також плавно і пропорційно буде змінюватися ємність, що дає можливість ефективно відстежувати стан вологості.

Для реалізації вимірювання параметрів вологості та температури ґрунту створено прототип додаткового модуля до головного вузлу модулів. Дана плата є вивідною частиною, що одним кінцем через кабель з'єднується з головною платою та спілкується з нею через протокол UART, а основним об'ємом встромляється в землю на глибину до 150 мм.

Для збірки прототипу було розведено та виготовлено плату (рис. 4.3)



Рисунок 4.3– Прототип плати

Через окрему обв'язку здійснюється вирівнювання сигналу в постійну величину для подальшого зчитування (рис. 4.4) Основним принципом вимірювання вологості ґрунту є зчитування перепаду напруги між пластинами імпровізованого конденсатора на великій площі, розведеної на платі. За допомогою таймеру МК здійснюється генерації PWM-сигнал частотою 1 МГц наповненістю 50% (рис. 4.5). Підключивши послідовно з резистором конденсатор, отримаємо фільтр нижніх частот (ФНЧ). Виходить діляник напруги, де у верхнього плеча R2 опір не змінюється, а ємнісний опір нижнього плеча TP2 змінюється залежно від частоти. На цей сигнал впливає вже перепади вологи в ґрунті, змінюючи його ємність. Для виділення амплітуди застосовується АМ-детектор (рис. 4.6). Головна суть полягає в необхідності підібрати ємність і опір таким чином, щоб конденсатор встигав підзаряджтися під час збільшення сигналу, а під час зменшення розряджався за час низького рівня, але під час зміни сигналу обвідна змінювалася [23].

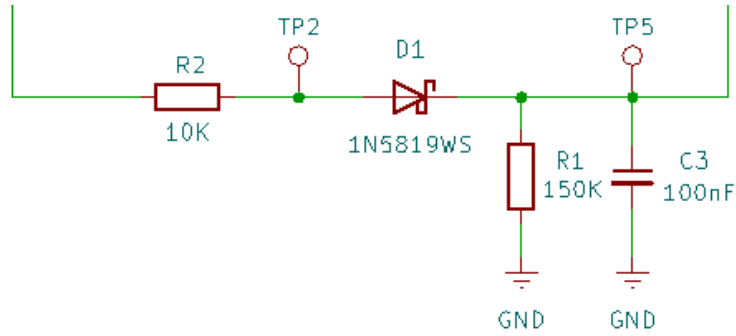


Рисунок 4.4 – Принципова електрична схема ФНЧ

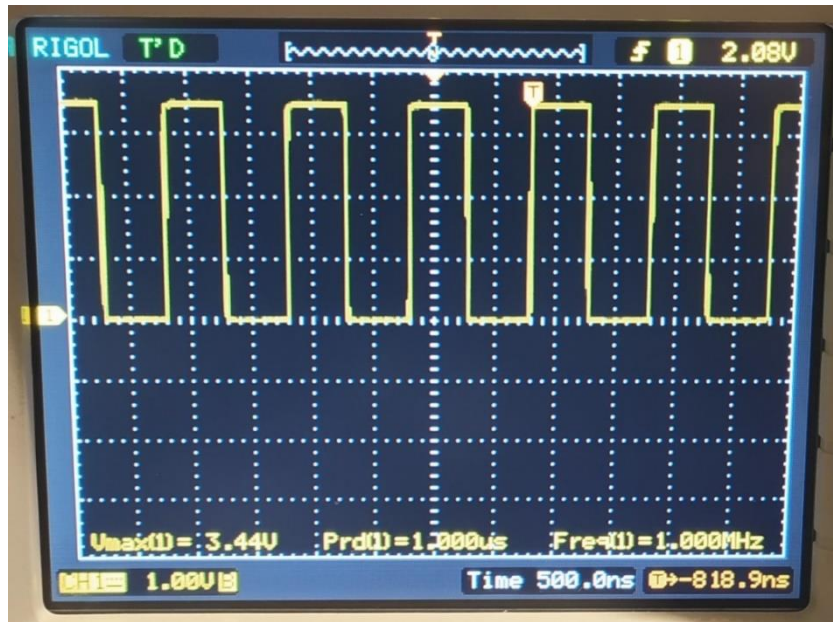


Рисунок 4.5 – PWM сигнал частотою 1 МГц з МК

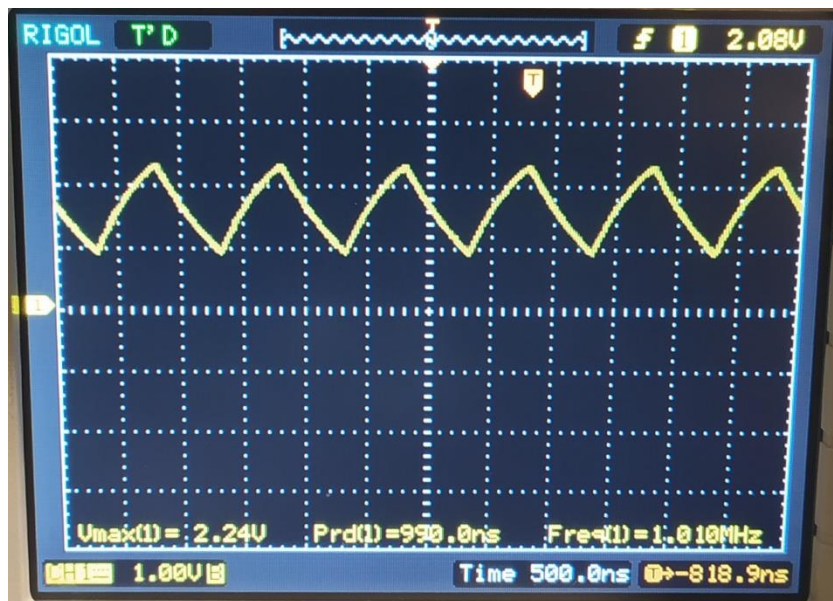


Рисунок 4.6 – Сигнал після ФНЧ

На виході з цього ланцюга отримуємо постійну напругу, що має діапазон від 2 В до 1,4 В, відповідно до проценту вологості від 0% до 100%. Чим менша напруга, тим вище ємкість і відповідно вища вологість. Залежність напруги від зміни вологи на ведено на рис. 4.7-4.8.

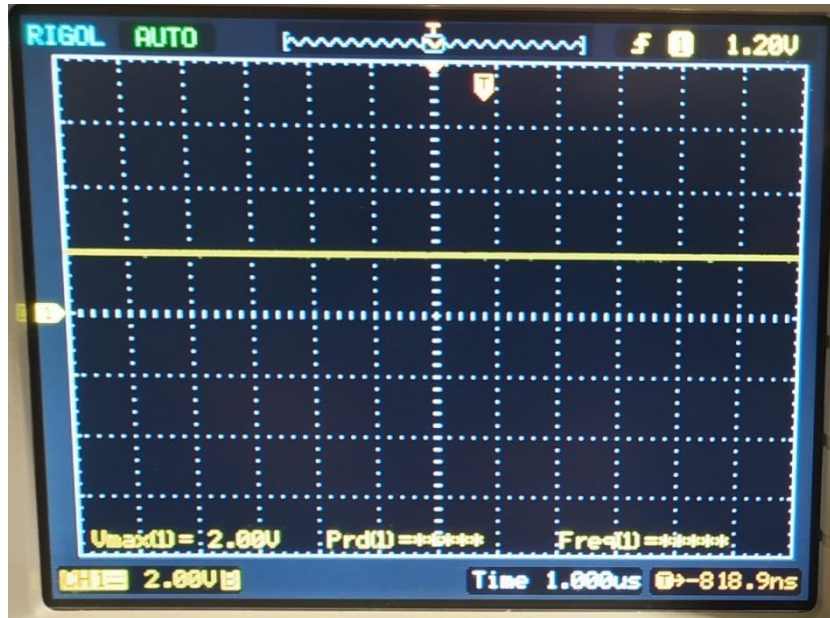


Рисунок 4.7 – Постійна напруга при 0% вологості

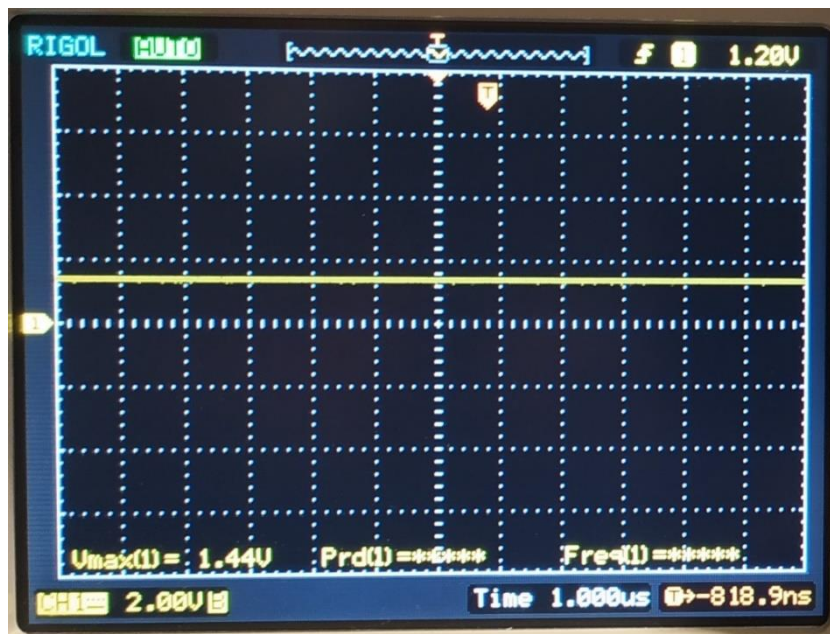


Рисунок 4.8 – Постійна напруга при 100% вологості

Перевагою ємнісного датчика перед просто двома голими електродами є відсутність електрохімічної реакції (електролізу), під час якої на контактах

відновлюватимуться речовини (з розчину) та псуватимуть ґрунт, а крім цього, вони будуть самі піддаватися корозії [24].

Виготовлення ємнісного датчика легко здійснюється при використанні технологій виготовлення друкованих плат, особливо це добре, коли інші компоненти розташовуються там же.

Оскільки одержувані показання залежать від параметрів датчика, то він потребує калібрування.

4.3 Розробка програмного забезпечення

Написання коду для шлюзу було здійснено з допомогою Arduino IDE. Ця середовище дозволяє легко інтегрувати компоненти системи, такі як ESP, LoRa-модулі та SIM800L, використовуючи готові бібліотеки та зручний інтерфейс. Реалізація основного циклу, ініціалізації модулів, обробки даних і передачі інформації виконувалася в цій IDE, що забезпечує зручність розробки та налагодження.

Для вузлів датчиків код було написано за допомогою Moon River Studio. Цей інструмент забезпечує розширені можливості для роботи з мікроконтролерами, орієнтованими на збір даних. Завдяки його функціоналу вдалося ефективно реалізувати зчитування показників датчиків, середньоарифметичні розрахунки та передачу даних до шлюзу.

Принцип роботи шлюзу

При першому включенні шлюз виконує ряд підготовчих кроків для налаштування і запуску основної роботи. Спочатку ініціалізується ESP-модуль, який відповідає за управління всією системою. Після натискання кнопки активується точка доступу і веб-сервер на ESP, що дозволяє користувачеві здійснювати налаштування (рис. 4.9).

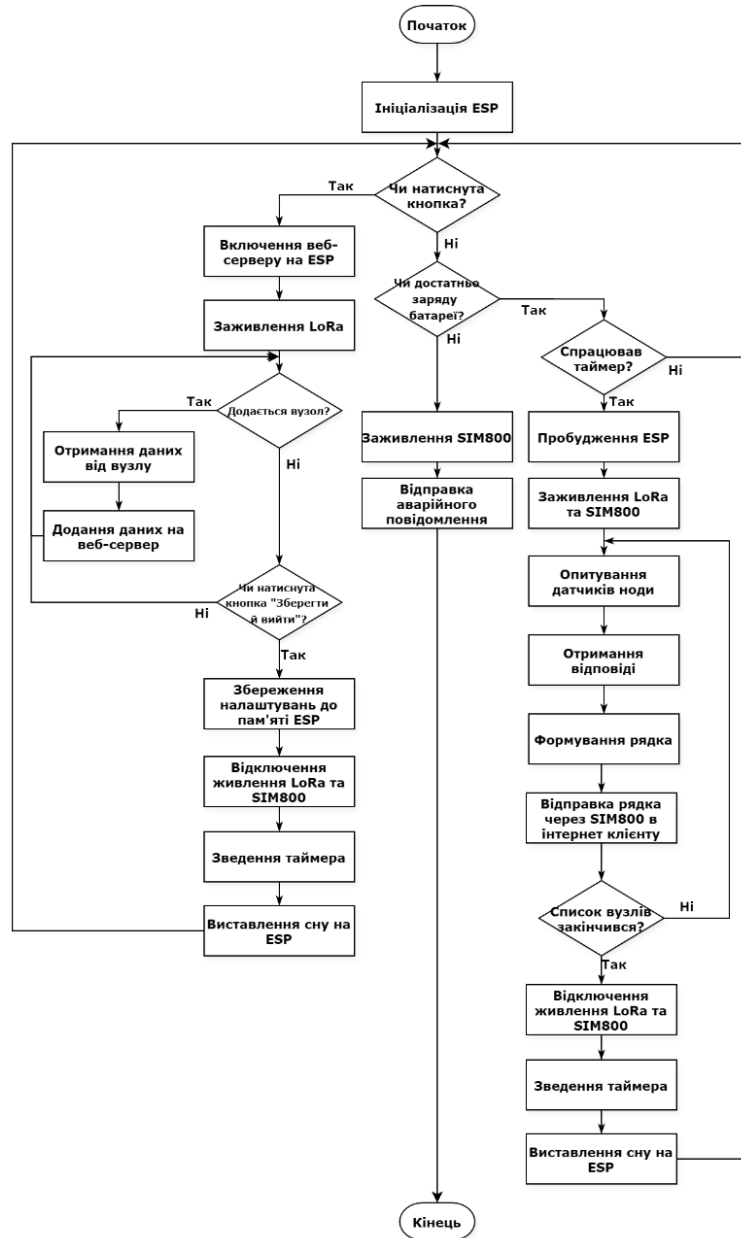


Рисунок 4.9 – Блок-схема програми шлюзу

Через веб-інтерфейс користувач може додати ідентифікатори вузлів. Для цього, після натискання відповідної кнопки на веб-сервері, LoRa-модуль переходить у режим прослуховування ефіру. При натисканні кнопки на вузлі, його ID надсилається шлюзу. Далі користувач вказує посилання для передачі інформації та визначає кількість і частоту збору даних. Після завершення налаштувань і натискання кнопки "зберегти і вийти" система зберігає введені параметри в пам'ять ESP і вимикає точку доступу. Після цього шлюз переходить до виконання основного циклу.

Основний цикл збору інформації починається з пробудження ESP. Подальшим кроком є подача живлення на модулі LoRa і SIM800L. Шлюз, орієнтуючись на список підключених вузлів, послідовно надсилає команду з ID кожному вузлу для запиту даних. Вузли відповідають рядками даних, які послідовно додаються до загальної структури.

Після отримання даних з усіх вузлів формується спільна строка, яка включає всю отриману інформацію. Ця строка передається кінцевому користувачеві через мобільний інтернет за допомогою SIM800L. Після успішної передачі живлення для модулів LoRa і SIM800L вимикається, щоб заощадити енергію.

Далі активується таймер, який визначає інтервал до наступного циклу збору даних. ESP переходить у режим сну до моменту, поки таймер не спрацює, після чого цикл повторюється.

Принцип роботи вузлу датчиків

Після запуску програми, мікроконтролер виконує послідовність дій для налаштування і початкової роботи з LORA. На першому етапі відбувається ініціалізація виводів і блоків мікроконтролера. Використовуються основні периферійні інтерфейси, такі як I²C, SPI, UART, та ADC, які забезпечують взаємодію з зовнішніми датчиками, комунікаційними модулями, та іншими компонентами системи (рис. 4.10).

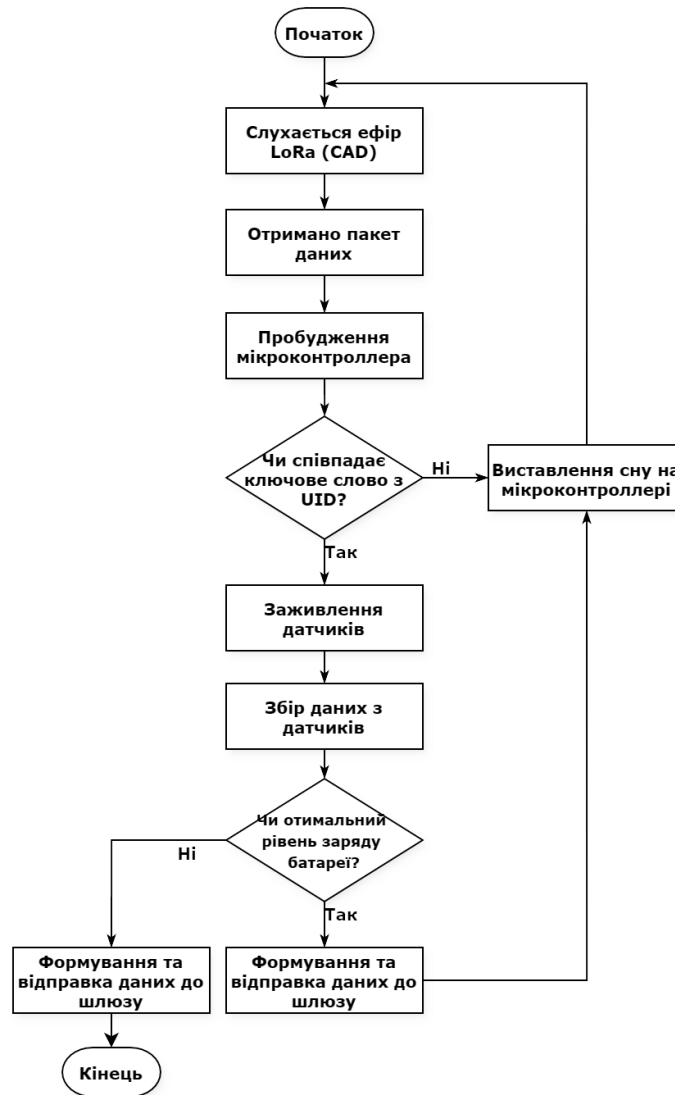


Рисунок 4.10 – Блок-схема програми вузлу датчиків

Далі проводиться ініціалізація підключених периферійних пристроїв та датчиків. Це включає налаштування адрес та параметрів взаємодії, специфічних для кожного компонента. Після цього LORA-модуль переводиться в режим прослуховування ефіру для отримання сигналів. З метою економії енергії, живлення датчиків відключається, а сам мікроконтролер переводиться в режим стенд-бай до надходження сигналу.

Коли LORA-модуль виявляє сигнал у ефірі, мікроконтролер активується для аналізу отриманого пакета. Перевіряється ідентифікатор пакета, і якщо сигнал адресовано даному пристрою, вмикається живлення датчиків для зчитування їх показників. Якщо сигнал не призначений цьому пристрою, мікроконтролер знову переходить у режим стенд-бай.

При отриманні запиту до пристрою, дані зчитуються з датчиків, включаючи показники температури, вологості ґрунту, або інші параметри. Разом із рівнем заряду батареї ці дані формуються у структуру для передачі на хаб. Після цього живлення датчиків знову вимикається, а мікроконтролер повертається в режим стенд-бай.

Принцип роботи датчика вологості

Коли на датчик вологості подається живлення, мікроконтролер ініціалізує свої виводи та блоки для роботи, зокрема UART та ADC. Після ініціалізації мікроконтролер переходить у режим очікування команд через інтерфейс UART (рис. 4.11).



Рисунок 4.11 – Блок-схема програми датчика вологості

При отриманні команди з кодом 2, мікроконтролер виконує десять вимірювань вологості та температури через ADC, після чого обчислює середнє значення отриманих даних. Якщо отримано команду з кодом 3, пристрій передає поточне значення температури. При команді 4 передається значення вологості ґрунту. Після виконання відповідної дії мікроконтролер знову переходить у режим очікування команд.

Живлення датчика вимикається, коли його робота завершена, що забезпечує енергоефективність системи.

4.4 Тестування розробленого комплексу

Для перевірки працездатності системи було проведено симуляцію реального циклу роботи.

На початку передбачається, що робітник компанії приїжджає на місце роботи для встановлення та налагодження системи моніторингу. Прибувши на місце, він відкриває кришку коробки шлюзу, під'єднає акумулятор та сонячну панель, тим самим подає живлення на головну плату, що запускає мікроконтролер. Наступним кроком є налаштування системи. Для цього необхідно натиснути на кнопку, що розташована на головній платі, після чого підіймається точка доступу, на якій розгортається веб-сторінка.

Щоб отримати доступ до налаштувань шлюзу робітник повинен з телефону, планшету, або ноутбуку підключитися до нещодавно створеної Wi-Fi мережі за паролем, встановлений від виробника і відомий тільки компанії (потім для звичного користування пароль може бути змінено). Далі робітник отримує доступ до веб сторінки, де може приступити до подальшого налагодження системи. На рис. 4.10 наведено зовнішній вигляд такої сторінки. На сторінці не повинно бути багато елементів та важких для системи дизайнерських рішень, тільки поля для налаштування для економії пам'яті та спроможності мікроконтролера.

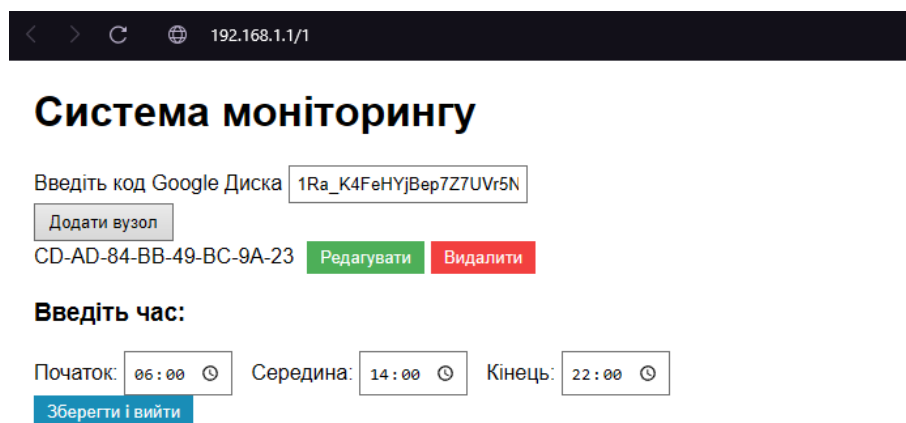


Рисунок 4.12 – Веб-сторінка системи моніторингу

На веб-сторінці можна побачити поле для введення посилання користувача, куди будуть приходити звіти з даними про стан навколишнього середовища. Далі присутня кнопка та список вже збережених вузлів з їх ID. Натиснувши на кнопку додати, мікроконтролер вмикає модуль LoRa та встановлює його в режим прослуховування ефіру. Система буде чекати поки обітних натисне на кнопку на вузлу датчиків, що у свою чергу надішле свій ID до шлюзу. Ці дії злінують шлюз та вузол, що дозволить їм в майбутньому комунікувати. На практиці можна побачити на рис 4.10, що система оперує ID “CD-AD-84-BB-49-BC-9A-23” розміром в 8 байт, відображений у системі Нех. Кожен з вузлів можна спокійно видалити зі списку.

Додатково можна відкалібрувати та налаштувати параметри вузлу датчиків відповідно від відстані, на які вони будуть встановлені, та сигналу, що вони передають. Параметри, якими ми керуємо у кодї зображені на рис. 4.11.

```
LoraClient.frequency           = 433;                // default = 433 MHz
LoraClient.spreadingFactor     = SF_7;                // default = SF_7
LoraClient.bandwidth           = BW_125KHz;           // default = BW_125KHz
LoraClient.crcRate             = CR_4_5;             // default = CR_4_5
LoraClient.power               = POWER_20db;         // default = 20db
LoraClient.overCurrentProtection = 100;              // default = 100 mA
LoraClient.preamble            = 8;                  // default = 8;
```

Рисунок 4.13 – Параметри вузлу датчиків

Наступним головним етапом є встановлення таймерів та проміжків для зчитування даних з датчиків та передачі їх клієнту. За замовчуванням система опитує вузли три рази на день.

Після завершення налаштування треба натиснути на кнопку «Зберегти та вийти», що призведе до записування донних до внутрішньої постійної пам'яті контролера та запуску основного циклу опитування датчиків.

Підключивши та налаштувавши систем, робітник розставляє вузли по території аграрного сектору.

Далі перевіримо що дані приходять коректно. На рис. 4.12 представлені дані, що надходять вузлу, а на рис. 4.13 зображені дані, що надходять з вузла до шлюзу. Дані зчитані через serial port в програмі Arduino IDE та MounRiver Studio.

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

Expression	Type	Value
▼ ntxdata	u8 [25]	0x20000074 <ntxdata>
↳ ntxdata[0]	u8	205 'f'
↳ ntxdata[1]	u8	171 'e'
↳ ntxdata[2]	u8	132 '\204'
↳ ntxdata[3]	u8	187 's'
↳ ntxdata[4]	u8	73 'l'
↳ ntxdata[5]	u8	188 '¼'
↳ ntxdata[6]	u8	154 '\232'
↳ ntxdata[7]	u8	35 '#'
↳ ntxdata[8]	u8	1 '\001'

Рисунок 4.14 – Дані з консолі налагодження вузлу

```

Output Serial Monitor x
Message (Enter to send message to 'Generic ESP8266 Module' on 'COM3')

TxDone
Received
Structure:
voltBat 67
soilMoisture 63
soilTemperature 18.4
airTemperature 22.5
airHumidity 47
lightLevel 7

```

Рисунок 4.15 – Дані з консолі налагодження шлюзу

Після того, як система зібрала всі дані, виконується передача через мобільну мережу до серверу користувача. На даному етапі проєкту таким сховищем даних є Excel таблиця на сервері гугл. На рис. 4.14 зображені збережені дані з датчиків готової системи.

A	B	C	D	E	F	G	H
Дата	Час	Заряд шлюзу %	Заряд вузлу %	Вологість ґрунту %	Темп. ґрунту	Вологість повітря %	Темп. повітря °C
21.11.2024	6:00	100	87	65	18,3	45	22,7
21.11.2024	14:01	100	85	42	18,6	54	23,1
21.11.2024	22:00	100	85	23	19,2	48	24,2
22.11.2024	6:00	100	84	12	19,8	62	23,1
22.11.2024	14:01	100	84	74	16,6	57	22,8
22.11.2024	22:01	100	83	58	17,1	47	23,5
23.11.2024	6:00	100	82	32	17,9	49	22,8
23.11.2024	14:00	100	82	15	18,7	54	23,1
23.11.2024	22:00	100	81	7	19,3	51	23,6

Рисунок 4.16 – Збережені дані

Результатом тестування системи є поступово заповнювана Excel таблиця, де зображені параметри заряду шлюзу, заряду вузлу, вологості й температури ґрунту, вологості й температури повітря, освітлення. Ці дані можуть бути використані для подальшого аналізу та впровадження дій, що до поливу тощо.

Висновки до розділу 4

На основі розроблених принципових схем для шлюзу та вузлів датчиків, створених у програмному забезпеченні KiCad, були зібрані функціональні прототипи на макетній платі. Для створення прототипів використовувались стандартні макетні плати, які забезпечували легке підключення елементів без пайки, що спрощувало організацію компонентів і оперативне внесення змін у схеми.

Прототипи шлюзу та вузлів датчиків включали ключові компоненти: мікроконтролери CH32V003F4P6 і плати ESP8266 для обробки даних та управління; модулі LoRa Ra-01 для передачі даних через бездротову мережу; датчики SHT3x та BH1750 для вимірювання вологості, температури ґрунту та інших параметрів; а також додаткові пасивні елементи для забезпечення стабільної роботи. Особливу увагу було приділено правильності підключення живлення, зокрема для GSM-модуля SIM800L, який потребував окремого джерела енергії.

Прототип шлюзу складався з плати ESP8266, підключеної до модулів LoRa та SIM800L через SPI- та UART-інтерфейси. Тестувалася передача даних між шлюзом і сервером через мобільний інтернет, що підтвердило працездатність розробленої системи. Прототип вузла датчиків, заснований на мікроконтролері CH32V003F4P6, забезпечував зчитування даних сенсорів через I2C-протокол та передачу їх через LoRa.

Для реалізації ємнісного вимірювального модуля було розроблено плату, яка дозволяє відстежувати вологість ґрунту на основі зміни ємності імпровізованого конденсатора. Зчитування результатів здійснювалося через обробку сигналу з використанням фільтрів і АМ-детектора. Таке рішення дозволяє уникнути корозії контактів і забезпечити тривалу експлуатацію датчика.

У ході тестування системи проведено збір даних про параметри навколишнього середовища в реальних умовах. Дані від вузлів датчиків передавалися на шлюз і надалі зберігалися на сервері. Для зручності аналізу результати автоматично експортувалися у формат Excel-таблиці, що дозволяє легко

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

оцінювати стан навколишнього середовища за ключовими показниками, такими як температура, вологість ґрунту, освітленість тощо.

Загалом, виконані розробки, тестування та отримання кінцевих результатів у вигляді Excel-таблиці підтвердили працездатність запропонованої системи моніторингу, що є важливим етапом у підготовці до серійного виготовлення друкованих плат і розгортання системи на практиці.

ВИСНОВКИ

За час виконання кваліфікаційної магістерської роботи були виконані всі поставлені завдання та досягнуті основні цілі, які стосувалися

У процесі дослідження було розроблено розподілену систему моніторингу параметрів оточуючого середовища на базі технології LoRaWAN для аграрної сфери України. Проведений аналіз сучасних підходів до моніторингу та управління полями підтвердив ефективність впровадження інформаційних технологій для підвищення продуктивності сільського господарства.

Запропоноване рішення дозволяє:

- забезпечити автоматизацію збору даних про стан навколишнього середовища;
- скоротити витрати на обслуговування завдяки низькому енергоспоживанню пристроїв та широкій зоні покриття;
- своєчасно виявляти відхилення від нормальних умов, що дозволяє оперативно приймати управлінські рішення;
- підвищити ефективність використання ресурсів, таких як вода, добрива та енергія.

Результати тестування системи підтвердили її працездатність у реальних умовах, а також виявили високий потенціал для масштабування на великі території. Система демонструє конкурентоспроможність як у технічному, так і в економічному аспектах, що робить її доцільною для впровадження в українському агропромисловому комплексі.

Практичне значення роботи полягає в створенні ефективного інструменту для моніторингу стану полів, який сприяє зменшенню негативного впливу на довкілля, оптимізації агротехнічних процесів та підвищенню стійкості аграрного сектора України. Запропоновані рекомендації щодо впровадження системи можуть бути використані підприємствами для реалізації концепції «розумного землеробства».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Михайлов О. О., Пузирьов С. В. Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN. *Ольвійський форум – 2024: стратегії країн Причорноморського регіону в геополітичному просторі. Технічні науки та інженерія*. Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2024. С. 174-177. URL: <http://surl.li/wwdjlw> (дата звернення: 05.10.2024).
2. LoRa Alliance®. URL: <https://lora-alliance.org> (Last accessed: 10.10.2024).
3. LoRa® and LoRaWAN®: A Technical Overview - Semtech LoRa. URL: https://lora-developers.semtech.com/uploads/documents/files/LoRa_and_LoRaWAN-A_Tech_Overview-Downloadable.pdf (Last accessed: 10.10.2024).
4. LoRa documentation. LoRa. URL: <https://lora.readthedocs.io/en/latest/#modulation-types-and-chirp-spread-spectrum> (Last accessed: 10.10.2024).
5. Semtech FAQ. Semtech. URL: <https://www.semtech.com/design-support/faq/P120> (Last accessed: 10.10.2024).
6. LoRaWAN Architecture. The Things Network. URL: <https://www.thethingsnetwork.org/docs/lorawan/architecture> (Last accessed: 10.10.2024).
7. Mukarram A. M., Waheb A. J., Noorazliza S., Suliman A. A Survey on LoRaWAN Technology: Recent Trends, Opportunities, Simulation Tools and Future Directions. *Electronics, MDPI*, 2022. Vol. 11(1), p. 164. DOI: 10.3390/electronics11010164 (Last accessed: 10.10.2024).
8. Marco C., Ahmed E. On the LoRa Modulation for IoT: Waveform Properties and Spectral Analysis. *IEEE Internet of Things Journal*, 2019. Vol. 6, pp. 8463-8470. DOI: 10.1109/JIOT.2019.2919151 (Last accessed: 10.10.2024).
9. Quantao Y., Hua W., Dongxuan H., Zhiping L. Enhanced Group-Based Chirp Spread Spectrum Modulation: Design and Performance Analysis. *IEEE Xplore*, 2023. DOI: 10.1109/JIOT.2024.3486126 (Last accessed: 21.11.2024).

10. Ali W. A., Raed S., Marwa C. Chirp Spread Spectrum-Based Waveform Design and Detection Mechanisms for LPWAN-Based IoT: A Survey. *IEEE Xplore*, 2024. Vol. 12, pp. 24949 - 25017 DOI: 10.1109/10391276. (Last accessed: 21.11.2024).
11. Thomas A., Patrick V. T., Hendrik R. A Compact Low-Power LoRa IoT Sensor Node with Extended Dynamic Range for Channel Measurements. *Sensors, MDPI*, 2018. Vol. 18(7), p. 2137. DOI: 10.3390/s18072137 (Last accessed: 21.11.2024).
12. Salma S. Ahmed E., Mohamed Y. S., Atef M. G., Mohamed F. A. Mobility of LoRaWAN Gateways for Efficient Environmental Monitoring in Pristine Sites. 2023. Vol. 23(3), 1698. DOI: 10.3390/s23031698 (Last accessed: 21.11.2024).
13. Vicky B., Brandon C., Sang G. Y. A Systematic Literature Review of LoRaWAN: Sensors and Applications. 2023. Vol. 23(20), 8440. DOI: 10.3390/s23208440 (Last accessed: 21.11.2024).
14. Pongsarun B., Thanamate T. Environment Monitoring System through LoRaWAN for Smart Agriculture. 2020. *IEEE Conference Publication*. DOI: 10.1109/Conference.2022.9310977 (Last accessed: 21.11.2024).
15. Paidikondala V., Amarlapudi S., Uppala S., K. Shri Ramtej. LoRaWAN based Environmental Monitoring for Smart Campus. 2023. *IEEE Transactions*. DOI: 10.1109/10441519 (Last accessed: 21.11.2024).
16. Juan J. L., Paula L. Rapid IoT Prototyping: A Visual Programming Tool and Hardware Solutions for LoRa-Based Devices. *Sensors, MDPI*, 2023. Vol. 3(17), p. 7511. DOI: 10.3390/s23177511 (Last accessed: 21.11.2024).
17. Pottie G. J., Kaiser W. J. Wireless Integrated Network Sensors. *Communications of the ACM*, 2000. pp. 51-58. DOI: 10.1145/332833.332838 (Last accessed: 21.11.2024).
18. Roderval M., Luan C. C., Renan C., Yuri C. & Vilson G. Internet of Things Applied to Precision Agriculture. *Lecture Notes in Networks and Systems*. 2018. Vol. 22, pp. 499-509. DOI: 10.1007/978-3-319-64352-6_46 (Last accessed: 21.11.2024).
19. Muthumanickam D., Poongodi C., Kumaraperumal R., Sellaperumal P., Rangunath K. Smart Farming: Internet of Things (IoT)-Based Sustainable Agriculture.

Digital Innovations in Agriculture. India, 2022. DOI: 10.3390/agriculture12101745 (Last accessed: 21.11.2024).

20. Jetmir H., Eli De Poorter, Ingrid M., Jeroen H. A Survey of LoRaWAN for IoT: From Technology to Application. *Department of Information Technology at Ghent University—IMEC*. 9052 Ghent, Belgium, 2018. DOI: 10.3390/s18113995 (Last accessed: 21.11.2024).

21. Mohamed R. M. K., Kuala L. IoT Applications in Smart Agriculture: Issues and Challenges. *IEEE Conference on Open Systems*. Malaysia, 2020. DOI: 10.1109/ICOS50156.2020.9293672. (Last accessed: 21.11.2024).

22. Yansong W., Yi H., Chaoyun S. A New Smart Sensing System Using LoRaWAN for Environmental Monitoring. *Communications and IoT Applications*. 2019 DOI: 10.1109/9018829 (Last accessed: 21.11.2024).

23. Srijeeta M., Gummadala K., Perli H., Himanshi, Sonu C., Khauhelo P. S., Taj T., Jatin V. Assessing Low-cost Capacitive Soil Moisture Sensors: Accurate, Affordable, and IoT-ready Solutions for Soil Moisture Monitoring. *International Journal of Environment and Climate Change*, 2023. Vol. 13(11), pp. 2233-2242. DOI: 10.9734/ijecc/2023/v13i113386 (Last accessed: 21.11.2024).

24. Abdelaziz M. O., Hasnaa G. I., Adel H. E., Khaled M. K., Zaher M. Y., Salah E. Designing Low-Cost Capacitive-Based Soil Moisture Sensor and Smart Monitoring Unit Operated by Solar Cells for Greenhouse Irrigation Management. *Sensors, MDPI*, 2021. Vol. 21(16), p. 5387. DOI: 10.3390/s21165387 (Last accessed: 21.11.2024).

25. LoRa Ra-01 SPECIFICATION. URL: https://docs.ai-thinker.com/_media/lora/docs/ra-01s_specification.pdf (Last accessed: 21.11.2024).

26. ESP-12E WiFi Module. URL: <https://components101.com/sites/default/files/2021-09/ESP12E-Datasheet.pdf>

27. SIM800L Hardware design v1.0. URL: https://www.makerhero.com/img/files/download/Datasheet_SIM800L.pdf (Last accessed: 21.11.2024).

28. 32-bit general-purpose RISC-V MCU-CH32V003 documentation. URL: <https://www.wch-ic.com/products/CH32V003.html> (Last accessed: 21.11.2024).
29. Sht3x Datasheet. URL: <http://surl.li/fkzjem> (Last accessed: 21.11.2024).
30. BH1750 Ambient Light Sensor Module. URL: <http://surl.li/tukvps> (Last accessed: 21.11.2024).

ДОДАТОК А

Код програми шлюзу

```
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <ESP_EEPROM.h>
#include <SPI.h>
#include <LoRa.h>
#include <vector>

#define ss 15
#define rst 16
#define dio0 4
#define RX 0
#define TX 2
#define WebButt 5
#define ConfGetSensData 1
#define ConfGetSettingsData 2
#define ConfPullSettingsData 3

struct Data_Client{
    u16  voltBat;
    u16  soilMoisture;
    u16  soilTemperature;
    u16  airTemperature;
    u16  airHumidity;
    u16  lightLevel;
}DataClient;

struct LoRa_setting
{
    int    frequency;
    u8     spreadingFactor;
    u8     bandWidth;
    u8     crcRate;
    u16    preamble;
    u8     power;
    u8     overCurrentProtection;
}StructParamLoRa;

const int EEPROM_SIZE = 512;
const int UID_Size = 8;
const int COUNTER_ADDRESS = 0;
const int TransferBufferSize = 32;

std::vector<uint8_t> TransferBuff;
std::vector<uint8_t> SendBuffer;

const char* ssid = "MainSab";
```

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

```
const char* password = "1234512355";
const char* server_google = "";
const char* google_script_id = "";
IPAddress local_ip(192,168,1,1);
IPAddress gateway(192,168,1,1);
IPAddress subnet(255,255,255,0);

SoftwareSerial SIM800(TX, RX);
ESP8266WebServer server(80);

bool IsPacketResived = false;
bool IsPacketSended = false;
bool AddNewNode = false;
bool ButtPress = false;
bool SendComm2 = false;
bool apEnabled = false;
bool ResiveEnabled = false;

void setup(){
  pinMode(WebButt, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(WebButt), CheckButt, RISING);

  Serial.begin(115200);
  SIM800.begin(115200);
  EEPROM.begin(EEPROM_SIZE);

  Serial.println("Start");
  SIM800.println("AT");

  LoRa.setPins(ss, rst, dio0);
  while (!Serial);

  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }

  SIM800.println("AT");
  delay(500);
  if (SIM800.find("OK")) {
    Serial.println("Модуль SIM800L готов.");
  } else {
    return;
  }
  SIM800.println("AT+CREG?");
  delay(500);

  SIM800.println("AT+SAPBR=3,1,\"CONTTYPE\", \"GPRS\");
  delay(500);
  SIM800.println("AT+SAPBR=3,1,\"APN\", \"Internet\");
  delay(500);
```

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

```
SIM800.println("AT+SAPBR=1,1");
delay(2000);
SIM800.println("AT+SAPBR=2,1");

LoRa.onReceive(RecvFromLORA);
LoRa.onTxDone(onTxDone);
LoRa_rxMode();
}

void loop()
{
if(ButtPress){
  if(IsPacketSended){
    TransferBuff.clear();
    for(size_t i = 0; i < sizeof(UID); i++){
      TransferBuff.push_back((uint8_t)UID[i]);
    }
    TransferBuff.push_back((uint8_t)1);
    LoRa_sendMessage();
    IsPacketSended = false;
  }
  if(IsPacketResived){
    IsPacketResived = false;
    uint32_t structureSize = sizeof(DataClient);
    uint8_t *dataRX = (uint8_t *)&DataClient;
    for(uint8_t i = 0; i < structureSize; i++){
      dataRX[i] = TransferBuff[i+8];
    }
    printInfo(DataClient);
    ButtPress = false;
  }
  if(IsPacketResived){
    IsPacketResived = false;

    uint32_t structureSize = sizeof(DataClient);
    uint8_t *dataRX = (uint8_t *)&DataClient;
    for(uint8_t i = 0; i < structureSize; i++){
      dataRX[i] = TransferBuff[i+8];
    }
    printInfo(DataClient);
    ButtPress = false;
  }
}
if (AddNewNode){
  GetUID();
}

if (SendComm2){
  Serial.println("Button pressed");
  SendComm2 = false;
}
```

```

    getSensorsData();
  }
}

ICACHE_RAM_ATTR void CheckButt(){
  SendComm2 = true;
}

void getSensorsData(){
  byte count = EEPROM.read(COUNTER_ADDRESS);
  for (int i = 0; i < count; i++) {
    int startAddress = 1 + (i * UID_Size);
    byte id[UID_Size];
    for (int j = 0; j < UID_Size; j++) {
      id[j] = EEPROM.read(startAddress + j);
    }

    LoRa.beginPacket();
    for (int j = 0; j < UID_Size; j++) {
      LoRa.write(id[j]);
    }
    LoRa.write(ConfigGetSensData);
    LoRa.endPacket(true);

    if(isIDEqual(id)){
      Serial.println("UID Approved, Send to .Json");
      for(size_t i = 8; i < TransferBuff.size(); i++){
        Serial.println(TransferBuff[i]);
      }
    }else Serial.println("Package not Approved");
    Serial.println();
  }
}

void getSensorsParam(int UIDNumber){
  int startAddress = 1 + (UIDNumber * UID_Size);
  for (int i = 0; i < UID_Size; i++)
  {
    SendBuffer.push_back((uint8_t)EEPROM.read(startAddress + i));
  }
  SendBuffer.push_back((uint8_t)2);
  SendToLORA();

  if(isIDEqual(UID))
  {
    Serial.println("UID Approved");
    uint32_t structureSize = sizeof(StructParamLoRa);
    uint8_t *dataRX = (uint8_t *)&StructParamLoRa;
    for(uint8_t i=0; i<structureSize;i++)
    {
      dataRX[i] = TransferBuff[i+8];
    }
  }
}

```

```
    }
    printDeviceInfo(StructParamLoRa);
    SendToClient();
} else Serial.println("Package not Approved");
}

void printDeviceInfo(const LoRa_setting &Lora1) {
    Serial.println("Structure; ");
    Serial.println(Lora1.frequency, DEC);
    Serial.println(Lora1.spredingFactor, DEC);
    Serial.println(Lora1.bandWidth, DEC);
    Serial.println(Lora1.crcRate, DEC);
    Serial.println(Lora1.preamble, DEC);
    Serial.println(Lora1.power, DEC);
    Serial.println(Lora1.overCurrentProtection, DEC);
}

void pulSensorsParam(uint8_t UID[UID_Size]){
    LoRa.beginPacket();
    for (int i = 0; i < UID_Size; i++) {
        LoRa.write(UID[i]);
    }
    LoRa.write(ConfPullSettingsData);
    uint32_t structureSize = sizeof(StructParamLoRa);
    uint8_t *dataRX = (uint8_t *)&StructParamLoRa;
    for(uint8_t i=0; i<structureSize;i++){
        LoRa.write(dataRX[i]);
    }
    LoRa.endPacket(true);
}

void TalkToSIM(){
    if(SIM800.available()){
        Serial.write(SIM800.read());
    }
    if(Serial.available()){
        SIM800.write(Serial.read());
    }
}

void SendToClient(){
    String data = "{\"temperature\":\" + String(temperature) + "\",\"humidity\":\" + String(humidity) + "\"}";
    SIM800.println("AT+HTTPINIT");
    delay(100);
    SIM800.println("AT+HTTTPARA=\"CID\",1");
    delay(100);
    SIM800.println("AT+HTTTPARA=\"URL\", \"https://\" + String(server_google) + "/macros/s/\" +
String(googel_script_id) + "/exec\"");
    delay(100);
    SIM800.println("AT+HTTTPARA=\"CONTENT\", \"application/json\"");
    delay(100);
}
```


Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

```

SIM800.print("AT+HTTPDATA=");
SIM800.println(String(data.length()) + ",10000");
delay(500);
SIM800.println(data);
delay(500);
SIM800.println("AT+HTTPACTION=1");
delay(1000);
ResiveEnabled = false;
String response = "";
while (SIM800.available()) {
    response += char(SIM800.read());
}
Serial.println(response);
SIM800.println("AT+HTTPTERM");
delay(100);
}

bool isIDEqual(const byte id2[UID_Size]) {
    for (int i = 0; i < UID_Size; i++) {
        if (TransferBuff[i] != id2[i]) {
            return false;
        }
    }
    return true;
}

bool isUIDExists() {
    byte count = EEPROM.read(COUNTER_ADDRESS);
    for (int i = 0; i < count; i++) {
        int startAddress = 1 + (i * UID_Size);
        byte existingID[UID_Size];
        for (int j = 0; j < UID_Size; j++) {
            existingID[j] = EEPROM.read(startAddress + j);
        }
        if (isIDEqual(existingID)) {
            return true;
        }
    }
    return false;
}

void addUID(){
    byte count = EEPROM.read(COUNTER_ADDRESS);
    int startAddress = 1 + (count * UID_Size);

    if (startAddress + UID_Size > EEPROM_SIZE) {
        Serial.println("Error");
        return;
    }
    if(count == 0){
        for (int i = 0; i < UID_Size; i++) {

```

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

```

EEPROM.write(startAddress + i, TransferBuff[i]);
}
}
else if(!isUIDExists()){
  for (int i = 0; i < UID_Size; i++) {
    EEPROM.write(startAddress + i, TransferBuff[i]);
  }
}
else {
  return;
}

EEPROM.write(COUNTER_ADDRESS, count + 1);
EEPROM.commit();
}

void readAllIDs(){
  byte count = EEPROM.read(COUNTER_ADDRESS);
  Serial.print("Количество ID в памяти: ");
  Serial.println(count);
  for (int i = 0; i < count; i++){
    int startAddress = 1 + (i * UID_Size);
    byte id[UID_Size];
    for (int j = 0; j < UID_Size; j++){
      id[j] = EEPROM.read(startAddress + j);
    }

    Serial.print("ID ");
    Serial.print(i + 1);
    Serial.print(": ");
    for (int j = 0; j < UID_Size; j++) {
      Serial.print(id[j]);
      if (j < UID_Size - 1) Serial.print(":");
    }
    Serial.println();
    SendToLORA();
  }
}

void GetUID(){
  if(IsPacketResived){
    if(TransferBuff.size() == UID_Size){
      addUID();
      readAllIDs();
      AddNewNode = false;
    }
  }
}

void SendToLORA(){
  Serial.print("Send package ");

```

```
LoRa_txMode();
LoRa.beginPacket();
for(size_t i = 0; i < SendBuffer.size(); i++){
  LoRa.write(SendBuffer[i]);
}
LoRa.endPacket(true);
Serial.println(" - Sended");
}

void ReciveFromLORA(int packetSize){
  TransferBuff.clear();
  for (int i = 0; i < packetSize; i++) {
    TransferBuff[i] = (uint8_t)LoRa.read();
  }
  Serial.println("Received");
  IsPacketResived = true;
}

void LoRa_rxMode(){
  LoRa.receive();
}
void LoRa_txMode(){
  LoRa.idle();
}
void onTxDone() {
  Serial.println("TxDone");
  LoRa_rxMode();
}

void ChangeWebStat(){
  if (apEnabled)
  {
    WebServerON();
  } else
  {
    WebServerOFF();
  }
}
void WebServerON(){
  WiFi.softAP(ssid, password);
  WiFi.softAPConfig(local_ip, gateway, subnet);
  Serial.println(WiFi.softAPIP());
  server.on("/", handle_Root);
  server.on("/1", handle_Root);
  server.onNotFound(handle_NotFound);
  server.begin();
}
void WebServerOFF(){
  server.stop();
  WiFi.softAPdisconnect();
}
```

```
void handle_Root(){
  server.send(200, "text/html", SendHTML());
}

void handle_NotFound(){
  server.send(404, "text/plain", "Not found");
}

String SendHTML(){
  client.println(R"rawliteral(
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Додавання та редагування рядків</title>
<style>
  body {
    font-family: Arial, sans-serif;
    margin: 20px;
  }
  .row {
    display: flex;
    align-items: center;
    margin-bottom: 5px;
  }
  .row p {
    margin: 0;
    padding-right: 10px;
  }
  .row button {
    background-color: #f44336;
    color: white;
    border: none;
    padding: 5px 10px;
    cursor: pointer;
    margin-right: 5px;
  }
  .row button:hover {
    background-color: #d32f2f;
  }
  button.edit {
    background-color: #4CAF50;
  }
  button.edit:hover {
    background-color: #45a049;
  }
  input[type="time"], input[type="text"] {
    margin-right: 10px;
    padding: 5px;
  }
rawliteral)";
}
```

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

```

}
button {
  padding: 5px 10px;
  cursor: pointer;
}
button.add {
  background-color: #4CAF50;
  color: white;
  border: none;
}
button.add:hover {
  background-color: #45a049;
}
button.save-exit {
  background-color: #008CBA;
  color: white;
  border: none;
}
button.save-exit:hover {
  background-color: #0073A1;
}
</style>
</head>
<body>
  <h1>Система моніторингу</h1>

  <div>
    <label for="userInput">Введіть код Google Диска</label>
    <input type="text" id="userInput" placeholder="Введіть текст здесь">
  </div>
  <button id="addRowButton">Додати вузол</button>
  <div id="rowsContainer">
  </div>
  <div id="timeInputs">
    <h3>Введіть час:</h3>
    <label for="startTime">Початок:</label>
    <input type="time" id="startTime">
    <label for="middleTime">Середина:</label>
    <input type="time" id="middleTime">
    <label for="endTime">Кінець:</label>
    <input type="time" id="endTime">
  </div>

  <button id="saveExitButton" class="save-exit">Зберегти і вийти</button>
  <script>
    const addRowButton = document.getElementById('addRowButton');
    const saveExitButton = document.getElementById('saveExitButton');
    const userInputField = document.getElementById('userInput');
    const rowsContainer = document.getElementById('rowsContainer');

```

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

```

let counter = 1;

addRowButton.addEventListener('click', () => {
  const userInput = userInputField.value.trim();

  if (userInput) {
    const row = document.createElement('div');
    row.className = 'row';
    const text = document.createElement('p');
    text.textContent = `${userInput}`;
    const deleteButton = document.createElement('button');
    deleteButton.textContent = 'Видалити';
    deleteButton.onclick = () => {
      rowsContainer.removeChild(row);
    };

    const editButton = document.createElement('button');
    editButton.textContent = 'Редагувати';
    editButton.className = 'edit';
    editButton.onclick = () => {
      const newText = prompt('Введіть новий текст:', text.textContent.replace(""));
      if (newText !== null) {
        text.textContent = `${newText}`;
      }
    };
    row.appendChild(text);
    row.appendChild(editButton);
    row.appendChild(deleteButton);
    rowsContainer.appendChild(row);

    userInputField.value = "";
    counter++;
  } else {
    alert('Будь ласка, введіть текст!');
  }
});

saveExitButton.addEventListener('click', () => {
  const startTime = document.getElementById('startTime').value;
  const middleTime = document.getElementById('middleTime').value;
  const endTime = document.getElementById('endTime').value;

  if (startTime && middleTime && endTime) {
    alert(`Час збережено:\nПочаток: ${startTime}\nСередина: ${middleTime}\nКінець:
    ${endTime}`);
  } else {
    alert('Будь ласка, заповніть усі поля часу!');
  }
});
</script>
</body>
</html>rawliteral");}

```

ДОДАТОК Б

Код програми вузлу датчиків

```
#include "debug.h"
#include "config.h"
#include "LoRa.h"
#include "bh1750.h"
#include "sht3x.h"
#include <string.h>

LoRa LoraClient;
DataClient dataClient;
u8 rxtxdata[25];
int    RSSI;
volatile u8 LoRa_MoD = 0
u16 ADC_val = 0;
u8 test = 0;

int main(void)
{
    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_1);
    SystemCoreClockUpdate();
    Delay_Init();

    USART_Printf_Init(115200);

    printf("SystemClk:%d\r\n",SystemCoreClock);
    printf( "ChipID:%08x\r\n", DBGMCU_GetCHIPID() );

    USARTx_CFG();

    I2Cx_CFG();
    SPIx_CFG();
    GPIOx_CFG();
    EXTI_INT_CFG();
    ADC_CFG();

    dataClient.airHumidity = 0;
    dataClient.airTemperature = 0;
    dataClient.lightLevel = 0;
    dataClient.soilMoisture = 0;
    dataClient.soilTemperature = 0;
    dataClient.voltBat = 0;

    LoraClient = newLoRa();
    uint32_t structureSize = sizeof(LoraClient);
    uint32_t *dataPtr = (uint32_t *)&LoraClient;
    for(u8 i = 0; i < structureSize/4; i++){
        dataPtr[i] = *(u32*)(0x08003FC0 + 4 * i);
    }
}
```

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

```

if(LoraClient.run1 != 0){
    LoraClient.CS_port      = GPIOC;
    LoraClient.CS_pin      = GPIO_Pin_0;
    LoraClient.reset_port  = GPIOD;
    LoraClient.reset_pin   = GPIO_Pin_0;
    LoraClient.DIO0_port   = GPIOD;
    LoraClient.DIO0_pin    = GPIO_Pin_2;

    LoraClient.frequency   = 433;
    LoraClient.spredingFactor = SF_7;
    LoraClient.bandWidth   = BW_125KHz;
    LoraClient.crcRate     = CR_4_5;
    LoraClient.power       = POWER_20db;
    LoraClient.overCurrentProtection = 100;
    LoraClient.preamble    = 8;
    LoraClient.run1        = 0;

    uint32_t structureSize = sizeof(LoraClient);
    FLASH_Unlock_Fast();
    FLASH_ErasePage_Fast(0x08003FC0);
    FLASH_BufReset();
    uint32_t *dataPtr = (uint32_t *)&LoraClient;
    for(u8 i=0; i<structureSize/4; i++){
        FLASH_BufLoad(0x08003FC0+4*i, dataPtr[i]);
    }
    FLASH_ProgramPage_Fast(0x08003FC0);
    FLASH_Lock_Fast();
}
LoRa_reset(&LoraClient);
LoRa_init(&LoraClient);

LoRa_startCAD(&LoraClient);
test=LoRa_read(&LoraClient, RegOpMode);
Delay_Ms(5);

while(1)
{
    if(LoRa_MoD == JOINT){
        memset(rtxdata,0,sizeof(rtxdata));
        buffer_unique_ID(rtxdata);
        LoRa_transmit(&LoraClient, rtxdata, 8, 500);
        LoRa_clearIRQ(&LoraClient);
        LoRa_startReceiving(&LoraClient);
        LoRa_MoD = 0;
    }
    else if(LoRa_MoD == DATA_TX){
        memset(rtxdata,0,sizeof(rtxdata));
        buffer_unique_ID(rtxdata);
        GPIO_SetBits(GPIOA, GPIO_Pin_2);
        BH1750_Init();
        Delay_Ms(100);
    }
}

```


Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

```

SHT3x_Read(&dataClient.airTemperature, &dataClient.airHumidity);
dataClient.soilMoisture = 0;
dataClient.soilTemperature = 0;
readBatt(&dataClient.voltBat);
BH1750_Read(BH1750_ADDR, &dataClient.lightLevel);
GPIO_ResetBits(GPIOA, GPIO_Pin_2);
uint32_t structureSize = sizeof(dataClient);
u8 *dataPtr = (u8 *)&dataClient;
for(u8 i=0; i<structureSize;i++){
    rtxdata[i+8] = dataPtr[i];
}
printf("Command DATA_TX.\r\n");
LoRa_transmit(&LoraClient, rtxdata, 20, 500);
LoRa_clearIRQ(&LoraClient);
LoRa_startCAD(&LoraClient);
LoRa_MoD = 0;
}
else if(LoRa_MoD == SETTING_TX){
    memset(rtxdata,0,sizeof(rtxdata));
    buffer_unique_ID(rtxdata);

    uint32_t structureSize = sizeof(LoraClient);
    u8 *dataPtr = (u8 *)&LoraClient;
    for(u8 i=0; i<structureSize-32;i++){
        rtxdata[i+8] = dataPtr[i+28];
    }

    LoRa_transmit(&LoraClient, rtxdata, 20, 500);
    LoRa_clearIRQ(&LoraClient);
    LoRa_startCAD(&LoraClient);
    LoRa_MoD = 0;
}
else if(LoRa_MoD == SETTING_RX){
    uint32_t structureSize = sizeof(LoraClient);
    u8 *dataRX = (u8 *)&LoraClient;
    for(u8 i=0; i<structureSize-32;i++){
        dataRX[i+28] = rtxdata[i+9];
    }

    FLASH_Unlock_Fast();
    FLASH_ErasePage_Fast(0x08003FC0);
    FLASH_BufReset();
    uint32_t *dataPtr = (uint32_t *)&LoraClient;
    for(u8 i=0; i<structureSize/4; i++){
        FLASH_BufLoad(0x08003FC0+4*i, dataPtr[i]);
    }
    FLASH_ProgramPage_Fast(0x08003FC0);
    FLASH_Lock_Fast();
    LoRa_reset(&LoraClient);
    LoRa_init(&LoraClient);
    LoRa_clearIRQ(&LoraClient);

```

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

```

    LoRa_startCAD(&LoraClient);
    memset(rtxdata,0,sizeof(rtxdata));
    printf("Command SETTING_RX.\r\n");
    LoRa_MoD = 0;
}
Delay_Ms(200);
}
}
void USART1_IRQHandler(void) __attribute__((interrupt("WCH-Interrupt-fast")));
void USART1_IRQHandler(void)
{
    if(USART_GetITStatus(USART1, USART_IT_RXNE) != RESET)
    {
        USART_SendData(USART1, USART_ReceiveData(USART1));
        if(USART_ReceiveData(USART1) == 1){
            LoRa_MoD = DATA_TX;
        } else if(USART_ReceiveData(USART1) == 2){
            LoRa_MoD = SETTING_TX;
        } else if(USART_ReceiveData(USART1) == 3){
            LoRa_MoD = SETTING_RX;
        }
    }
}
void EXTI7_0_IRQHandler(void) __attribute__((interrupt("WCH-Interrupt-fast")));
void EXTI7_0_IRQHandler(void)
{
    if (EXTI_GetITStatus(EXTI_Line2) != RESET) {
        memset(rtxdata,0,sizeof(rtxdata));
        LoRa_receive(&LoraClient, rtxdata, 25);
        printf("Receive data:%x\r\n", rtxdata);
        if(compare_unique_ID(rtxdata)==1)
        {
            if(rtxdata[8] == 0x01){LoRa_MoD = DATA_TX;}
            else if(rtxdata[8] == 0x02){LoRa_MoD = SETTING_TX;}
            else if(rtxdata[8] == 0x03){LoRa_MoD = SETTING_RX;}
        }
        else {
            LoRa_startCAD(&LoraClient);
            LoRa_clearIRQ(&LoraClient);
            memset(rtxdata,0,sizeof(rtxdata));
        }

        EXTI_ClearITPendingBit(EXTI_Line2);
    }
    if (EXTI_GetITStatus(EXTI_Line3) != RESET) {
        LoRa_MoD = JOINT;
        EXTI_ClearITPendingBit(EXTI_Line3);
    }
}
}

```

ДОДАТОК В

Код програми модуля ємкісного модуля ємкісного датчика вологи

```
#include "debug.h"

u8 TxBuffer[] = "OK";
u8 RxBuffer[] = {0};
u8 mode = 0;

u16 ADC_Temp = 0;
u16 ADC_Humi = 0;

void USARTx_CFG(void)
{
    GPIO_InitTypeDef GPIO_InitStructure = {0};
    USART_InitTypeDef USART_InitStructure = {0};
    NVIC_InitTypeDef NVIC_InitStructure = {0};

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOID | RCC_APB2Periph_USART1,
    ENABLE);

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_30MHz;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
    GPIO_Init(GPIOID, &GPIO_InitStructure);
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
    GPIO_Init(GPIOID, &GPIO_InitStructure);

    USART_InitStructure.USART_BaudRate = 115200;
    USART_InitStructure.USART_WordLength = USART_WordLength_8b;
    USART_InitStructure.USART_StopBits = USART_StopBits_1;
    USART_InitStructure.USART_Parity = USART_Parity_No;
    USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
    USART_InitStructure.USART_Mode = USART_Mode_Tx | USART_Mode_Rx;

    USART_Init(USART1, &USART_InitStructure);
    USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);

    NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&NVIC_InitStructure);

    USART_Cmd(USART1, ENABLE);
}

void TIM2_PWMOut_Init(u16 arr, u16 psc, u16 ccp){
    GPIO_InitTypeDef GPIO_InitStructure={0};
```

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

```
TIM_OCInitTypeDef TIM_OCInitStructure={0};
TIM_TimeBaseInitTypeDef TIM_TimeBaseInitStructure={0};

RCC_APB2PeriphClockCmd(RCC_APB2Periph_AFIO|RCC_APB2Periph_GPIOC, ENABLE);
RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2, ENABLE );
GPIO_PinRemapConfig(GPIO_PartialRemap1_TIM2, ENABLE);

GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_30MHz;
GPIO_Init(GPIOC, &GPIO_InitStructure);

TIM_TimeBaseInitStructure.TIM_Period = arr;
TIM_TimeBaseInitStructure.TIM_Prescaler = psc;
TIM_TimeBaseInitStructure.TIM_ClockDivision = TIM_CKD_DIV1;
TIM_TimeBaseInitStructure.TIM_CounterMode = TIM_CounterMode_Up;
TIM_TimeBaseInit( TIM2, &TIM_TimeBaseInitStructure);

TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;
TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_Pulse = ccp;
TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;
TIM_OC2Init( TIM2, &TIM_OCInitStructure );

TIM_CtrlPWMOutputs(TIM2, ENABLE );
TIM_OC2PreloadConfig( TIM2, TIM_OCPreload_Disable );
TIM_ARRPreloadConfig( TIM2, ENABLE );
}

void ADC_CFG(void)
{
    ADC_InitTypeDef ADC_InitStructure = {0};
    GPIO_InitTypeDef GPIO_InitStructure = {0};

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA|RCC_APB2Periph_GPIOC, ENABLE);
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1, ENABLE);
    RCC_ADCClockConfig(RCC_PCLK2_Div8);

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
    GPIO_Init(GPIOA, &GPIO_InitStructure);

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
    GPIO_Init(GPIOC, &GPIO_InitStructure);

    ADC_DeInit(ADC1);
    ADC_InitStructure.ADC_Mode = ADC_Mode_Independent;
    ADC_InitStructure.ADC_ScanConvMode = DISABLE;
    ADC_InitStructure.ADC_ContinuousConvMode = DISABLE;
```

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

```
ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None;
ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
ADC_InitStructure.ADC_NbrOfChannel = 1;
ADC_Init(ADC1, &ADC_InitStructure);

ADC_RegularChannelConfig(ADC1, ADC_Channel_0, 1, ADC_SampleTime_241Cycles);
ADC_InjectedChannelConfig(ADC1, ADC_Channel_2, 1, ADC_SampleTime_241Cycles);
ADC_Calibration_Vol(ADC1, ADC_CALVOL_50PERCENT);
ADC_AutoInjectedConvCmd(ADC1, ENABLE);;
ADC_Cmd(ADC1, ENABLE);

ADC_ResetCalibration(ADC1);
while(ADC_GetResetCalibrationStatus(ADC1));
ADC_StartCalibration(ADC1);
while(ADC_GetCalibrationStatus(ADC1));

ADC_SoftwareStartConvCmd(ADC1, ENABLE);
while(!ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC));
Delay_Ms(20);
ADC_GetConversionValue(ADC1);
ADC_GetInjectedConversionValue(ADC1, ADC_InjectedChannel_1);
ADC_SoftwareStartConvCmd(ADC1, DISABLE);
ADC_ClearFlag(ADC1, ADC_FLAG_EOC);
}
void readValue(u16 *temp, u16 *humi){
    *temp = 0; *humi = 0;
    TIM2_PWMOut_Init(48-1, 0, 24-1);
    TIM_Cmd( TIM2, ENABLE );
    Delay_Ms(50);
    for(u8 i = 0; i<10; i++){
        ADC_SoftwareStartConvCmd(ADC1, ENABLE);
        while(!ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC));
        Delay_Ms(20);
        *humi += ADC_GetConversionValue(ADC1);
        *temp += ADC_GetInjectedConversionValue(ADC1, ADC_InjectedChannel_1);
        ADC_SoftwareStartConvCmd(ADC1, DISABLE);
        ADC_ClearFlag(ADC1, ADC_FLAG_EOC);
    }

    *temp = *temp/10;
    *humi = *humi/10;

    TIM2_PWMOut_Init(48-1, 0, 0);
    TIM_Cmd( TIM2, DISABLE );
}

int main(void){
    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_1);
    SystemCoreClockUpdate();
    Delay_Init();
```

Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN

```
USART_Printf_Init(115200);

printf("SystemClk:%d\r\n",SystemCoreClock);
printf("ChipID:%08x\r\n",DBGMCU_GetCHIPID());

USARTx_CFG();
ADC_CFG();
TIM2_PWMOut_Init(48-1, 0, 24-1);
while(1){
    if(mode == 1){
        TIM2_PWMOut_Init(48-1, 0, 24-1);
        TIM_Cmd(TIM2, ENABLE);
        mode=0;
    }
    else if(mode == 2){
        TIM2_PWMOut_Init(48-1, 0, 0);
        TIM_Cmd(TIM2, DISABLE);
        mode=0;
    }
    else if(mode==3){
        TIM2_PWMOut_Init(48-1, 0, 24-1);
        TIM_Cmd(TIM2, ENABLE);
        Delay_Ms(50);
        ADC_SoftwareStartConvCmd(ADC1, ENABLE);
        while(!ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC));
        Delay_Ms(20);
        ADC_Humi = ADC_GetConversionValue(ADC1);
        ADC_Temp = ADC_GetInjectedConversionValue(ADC1, ADC_InjectedChannel_1);
        ADC_SoftwareStartConvCmd(ADC1, DISABLE);
        ADC_ClearFlag(ADC1, ADC_FLAG_EOC);

        TIM2_PWMOut_Init(48-1, 0, 0);
        TIM_Cmd(TIM2, DISABLE);

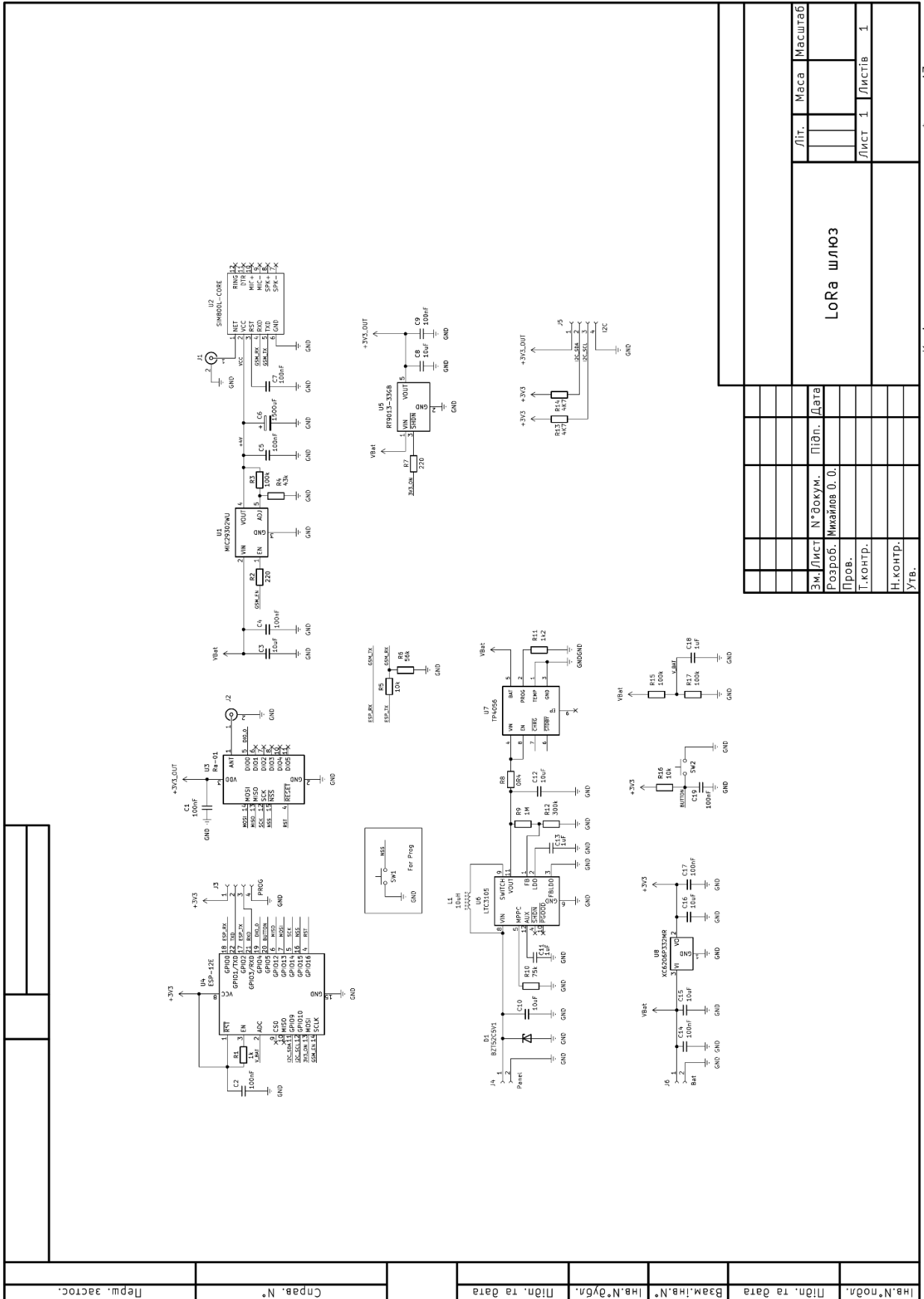
        printf("Humidity:%d\r\n",ADC_Humi);
        printf("Temperature:%d\r\n",ADC_Temp);

        Delay_Ms(1000);
    }
    else if(mode==4){
        readValue(&ADC_Temp, &ADC_Humi);
        USART_SendData(USART1, ADC_Temp);
        printf("Temperature:%d\r\n",ADC_Temp);
        mode = 0;
    }
    else if(mode==5){
        readValue(&ADC_Temp, &ADC_Humi);
        USART_SendData(USART1, ADC_Humi);
        printf("Humidity:%d\r\n",ADC_Humi);
        mode = 0;
    }
}
```

```
    }  
  }  
  void USART1_IRQHandler(void) __attribute__((interrupt("WCH-Interrupt-fast")));  
  
  void USART1_IRQHandler(void){  
    if(USART_GetITStatus(USART1, USART_IT_RXNE) != RESET){  
      USART_SendData(USART1, USART_ReceiveData(USART1));  
      if(USART_ReceiveData(USART1) == 1){  
        mode=1;  
      } else if(USART_ReceiveData(USART1) == 2){  
        mode=2;  
      } else if(USART_ReceiveData(USART1) == 3){  
        mode=3;  
      } else if(USART_ReceiveData(USART1) == 4){  
        mode=4;  
      }else if(USART_ReceiveData(USART1) == 5){  
        mode=5;  
      }  
    }  
  }  
}
```

ДОДАТОК Г

Принципова електрична схема шлюзу



Лит.	Маса	Масштаб
Лист 1	Листів 1	
LoRa шлюз		
Зм. Лист	№ докум.	Підп. Дата
Розроб.	Михайлов О. О.	
Пров.		
Т.контр.		
Н.контр.		
Утв.		

Копіював

Формат А3

ДОДАТОК Д

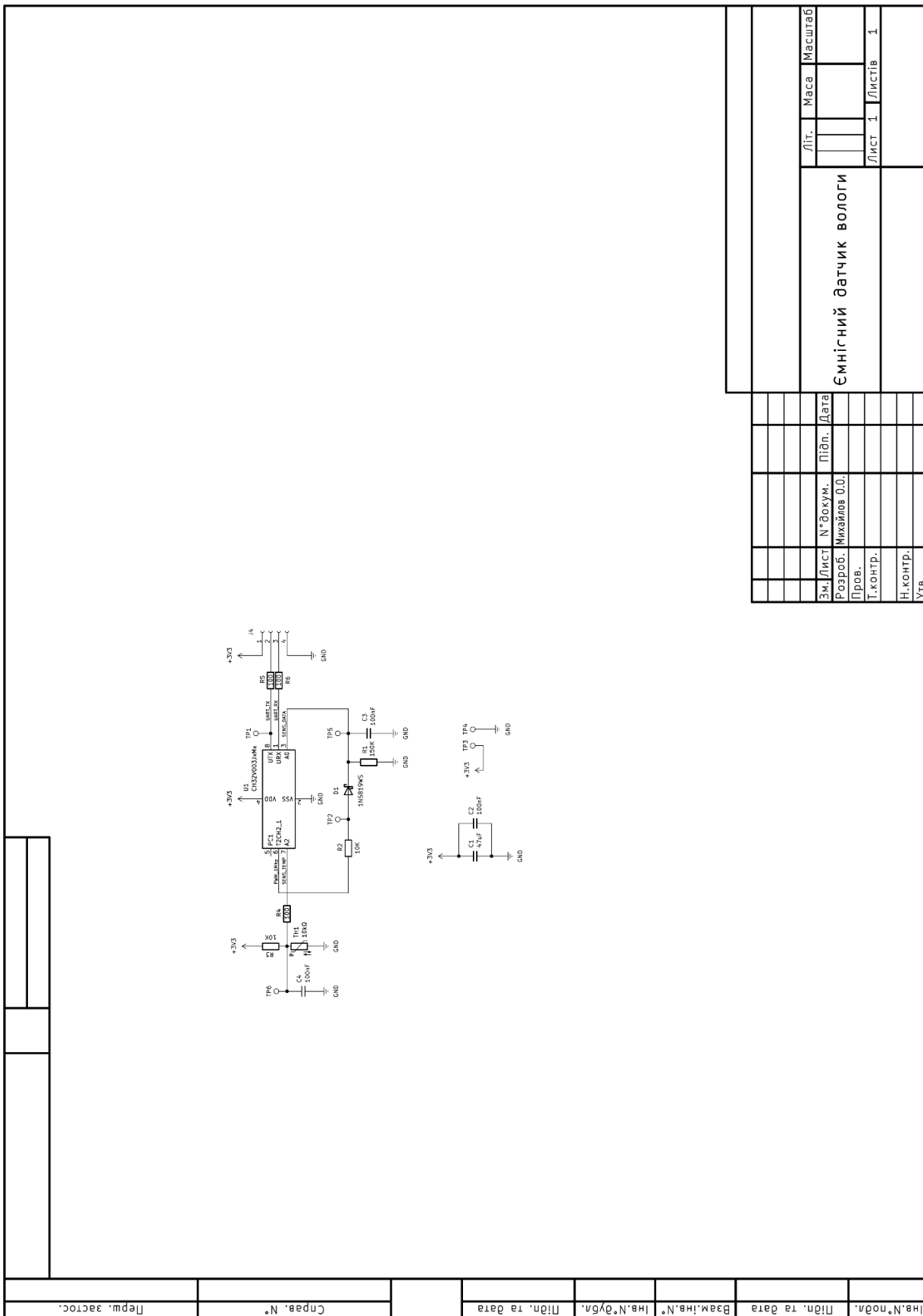
Принципова електрична схема вузлу датчиків

Лист №	LoRa вузол	Лист №	Масштаб
Розроб.	Михайлов О.О.	Лист 1	Листів 1
Т.контр.			
Н.контр.			
Утв.			
Копіював		Формат А3	

Лист №	LoRa вузол	Лист №	
Розроб.	Михайлов О.О.	Лист 1	
Т.контр.			
Н.контр.			
Утв.			
Копіював		Формат А3	

ДОДАТОК Е

Принципова електрична схема модуля ємкісного датчика вологи



ДОДАТОК Ж
Апробація кваліфікаційної роботи

Міністерство освіти і науки України
Чорноморський національний університет імені Петра Могили
Національна академія наук України
Південний науковий центр НАН і МОН України
Інститут української археографії та джерелознавства
ім. М.С. Грушевського НАН України
Державний архів Миколаївської області
ДУ «Національний науковий центр радіаційної медицини НАМН України»
Донецький національний медичний університет
Technical University of Moldova (Moldova)
Jan Dlugosz University in Czestochowa (Poland)
Adam Mickiewicz University (Poland)
Leipzig University of Applied Sciences (Germany)
Rzeszow University of Technology (Poland)
Ca' Foscari University (Italy)



ОЛЬВІЙСЬКИЙ ФОРУМ – 2024:
стратегії країн Причорноморського регіону
в геополітичному просторі

XXI Міжнародна наукова конференція

ТЕЗИ

ТЕХНІЧНІ НАУКИ ТА ІНЖЕНЕРІЯ

20–23 червня 2024 р., м. Миколаїв, Україна

Миколаїв – 2024

УДК 004.623

Михайлов О. О.,
магістрант кафедри комп'ютерної інженерії,
Пузирьов С. В.,
канд. фіз.-мат. наук, доцент кафедри комп'ютерної інженерії,
ЧНУ імені Петра Могили, м. Миколаїв, Україна

РОЗПОДІЛЕНА СИСТЕМА МОНІТОРИНГУ ПАРАМЕТРІВ ОТОЧУЮЧОГО СЕРЕДОВИЩА НА БАЗІ LoRaWAN

В сучасному світі, коли забруднення навколишнього середовища стає однією з найбільш гострих проблем, впровадження ефективних систем моніторингу стає важливим кроком у напрямку збереження природи та забезпечення розвитку як в містах так і за їх межами. Виявлення та контроль рівнів токсичних речовин у повітрі, воді та ґрунті дозволяє уникнути потенційно небезпечних ситуацій та запобігти захворюванням, пов'язаним з забрудненням довкілля. Моніторинг та аналіз даних є ключовим інструментом для розуміння впливу людської діяльності на екосистеми, що дозволяє розробляти та впроваджувати стратегії збереження природних ресурсів. Комплексне вивчення стану навколишнього середовища в межах міста або на територіях господарських підприємств, а саме збір та аналіз даних показників якості повітря, вологості ґрунту, температур тощо, дозволяє нам розробляти стратегії адаптації та зменшення впливу нашої діяльності на клімат.

Для забезпечення збору та передачі даних розумно буде використовувати технологію широкосмутового зв'язку великої дальності LoRa. Технологія LoRa, що є частиною Інтернету речей (IoT), відкриває нові горизонти для систем моніторингу, забезпечуючи низьке споживання енергії та великий радіус дії, що робить її ідеальним рішенням для збору даних про стан довкілля. У цьому контексті важливо розглянути можливість та переваги розподіленої системи моніторингу параметрів оточуючого середовища на базі технології LoRa.

Датчики, обладнані LoRa-технологією, розміщуються на різних ділянках території, що підлягає моніторингу. Це можуть бути міста з різними умовами середовища, такі як міські райони, промислові зони, лісові масиви тощо. Датчики збирають дані про різні параметри середовища (температура, вологість, рівень забруднення повітря, тощо) та передають їх у сліду мережу LoRaWAN. Дані, отримані від датчиків, агрегуються та аналізуються централізовано в спеціалізованому

174

ПІДСЕКЦІЯ: Комп'ютерна інженерія

- Алексейко В. О., Павлова О. О.** Застосування машинного навчання для прогнозування температури земної поверхні відповідно до кліматичного районування..... 132
- Баландін Я. В., Журавська І. М.** Система попередження ДТП шляхом виявлення засинання водія..... 135
- Басов Д. Є., Пузирьов С. В.** Розподілена система детектування рухомих об'єктів та обміну інформацією на базі LoRaWAN..... 138
- Буряк М. Р.** Методи та засоби оптимізації використання ресурсів в Kubernetes кластера 141
- Варанкін Д. В., Обухова К. О.** Система дального радіозв'язку для контролю датчиків на основі одношплатних мікрокомп'ютерів з LoRa..... 144
- Гончаров Д. С.** Емпірична оцінка якості вимірювань датчика пульсу шляхом статистичної обробки даних..... 146
- Данишова О. М., Бурлаченко І. С.** Апаратно-програмний комплекс моделювання руху протезів..... 151
- Завгородній К. С., Бурлаченко І. С.** Система відеоспостереження для запобігання промисловим травмам..... 155
- Козяк Л. О., Пузирьов С. В.** Адаптивне управління освітленням в приміщенні за допомогою датчиків та алгоритмів машинного навчання 159
- Косілап І. Л., Бурлаченко І. С.** ERP для обліку комплектуючих на базі Raspberry Pi 162
- Кравченко П. К., Бурлаченко І. С.** Використання CNN та Spring Boot на базі Raspberry Pi Zero W для виявлення гострого лімфоцитарного лейкозу..... 165
- Лосіцький П. М., Журавська І. М.** Розробка та вдосконалення алгоритмів управління БПЛА для місії пошуку та порятунку 168
- Лялюк В. М., Бурлаченко І. С.** KYC-верифікація для вебплатформи торгівлі товарами повільного призначення на базі Vapora Pi M3 171
- Михайлов О. О., Пузирьов С. В.** Розподілена система моніторингу параметрів оточуючого середовища на базі LoRaWAN..... 174

246