

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра спеціалізованих комп'ютерних систем

Палійчук Христина Мирославівна

**IoT система обліку водопостачання/ IoT water supply
accounting system**

спеціальність: 152 – Метрологія та інформаційно-вимірвальна
техніка
освітньо-професійна програма – Технології інтернету речей

Кваліфікаційна робота

Виконав студент групи МТІР - 41
Х.М. Палійчук

Науковий керівник
к.т.н., О.М.Заставний

Кваліфікаційну роботу допущено
до захисту:

« ____ » _____ 2026 р.

Завідувач кафедри

_____ А.І.Сегін

ТЕРНОПІЛЬ – 2026

Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра спеціалізованих комп'ютерних систем
Освітній ступінь "бакалавр"
Спеціальність: 152 – Метрологія та інформаційно-вимірювальна техніка
Освітньо-професійна програма – Технології інтернету речей

“ЗАТВЕРДЖУЮ”

Завідувач кафедри СКС

_____ А.І.Сегін
“ ____ ” _____ 2025 р.

З А В Д А Н Н Я

НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ПАЛІЙЧУК Христині Мирославівні

(прізвище, ім'я по-батькові)

1. Тема випускної кваліфікаційної роботи

IoT система обліку водопостачання/ IoT water supply accounting system

керівник роботи к.т.н., Заставний О.М.

затверджені наказом по університету від 1.12.2025р №894

2. Строк подання студентом закінченої випускної кваліфікаційної роботи:
25.05.2026р.

3. Вихідні дані до випускної кваліфікаційної роботи:

1. IoT системи автоматизованого обліку водопостачання та їх
компоненти

2. Сенсори та засоби обліку систем водопостачання

3. Засоби керування технологічними процесами

4. Основні питання, які потрібно розробити:

1. Аналіз IoT систем обліку водопостачання

2. Проектування IoT системи обліку водопостачання

3. Розроблення IoT системи обліку водопостачання

5. Перелік графічного матеріалу у роботі:

1. Структурна схема IoT системи обліку водопостачання

2. Структурна схема облікового модуля системи

3. Блок-схема алгоритму роботи вузла обліку водоспоживання

6. Консультанти розділів випускної кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Заставний О.М.		
2	Заставний О.М.		
3	Заставний О.М.		

7. Дата видачі завдання 1 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назви етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз IoT систем обліку водопостачання	01.12.2025р. – 15.01.2026р.	виконано
2	Проектування IoT системи обліку водопостачання	16.01.2026р. – 15.03.2026р.	виконано
3	Розроблення IoT системи обліку водопостачання	16.03.2026р. – 30.04.2026р.	виконано
4.	Остаточне оформлення та подача кваліфікаційної роботи на перевірку щодо плагіату	1.05.2026р. – 25.05.2026	виконано

Студент

(підпис)

Палійчук Х.М.

Керівник роботи

(підпис)

к.т.н., Заставний О.М.

АНОТАЦІЯ

Палійчук Х.М. IoT система обліку водопостачання. – Рукопис.

Дослідження на здобуття освітнього ступеня «бакалавр» за спеціальністю 152 «Метрологія та інформаційно-вимірювальна техніка», освітньо-професійна програма – Технології інтернету речей. Західноукраїнський національний університет, Тернопіль, 2026.

У роботі проведено аналіз сучасних систем дистанційного обліку водопостачання, технологій бездротової передачі даних та засобів побудови енергоефективних IoT мереж. Розглянуто архітектуру існуючих рішень, принципи збору та передачі даних, а також особливості використання механічних водомірів із дистанційним зчитуванням показників.

У процесі проєктування розроблено структуру вузла обліку на базі мікроконтролера ESP32-C6, сенсора Холла та автономного джерела живлення. Запропоновано використання Thread mesh-мережі для передачі даних між вузлами обліку та центральним шлюзом із подальшою передачею інформації на сервер.

Розроблено алгоритми роботи вузла обліку, серверної частини та клієнтського програмного забезпечення. Виконано оцінку енергоспоживання системи та обґрунтовано можливість забезпечення багаторічної автономної роботи вузлів обліку.

Запропоноване рішення характеризується низькою вартістю впровадження, масштабованістю та можливістю інтеграції у сучасні системи моніторингу комунальних ресурсів. Використання Thread mesh-мережі дозволяє мінімізувати потребу у додатковій мережевій інфраструктурі.

Ключові слова: Інтернет речей (IoT), облік водопостачання, дистанційний моніторинг, водомір, ESP32-C6, сенсор Холла, Thread, mesh-мережа, MQTT, автономне живлення, бездротова передача даних, енергоефективність, автоматизована система обліку.

ANNOTATION

Palijchyk H.M. IoT water supply accounting system. - Manuscript.

Research for obtaining the Bachelor's Degree in Specialty 152 "Metrology and Information-Measuring Technology", Educational and Professional Program – Internet of Things Technologies. West Ukrainian National University, Ternopil, 2026.

In this work, an analysis of modern remote water metering systems, wireless data transmission technologies, and methods for building energy-efficient IoT networks is carried out. The architecture of existing solutions, principles of data collection and transmission, as well as the features of using mechanical water meters with remote reading capabilities are examined. Based on the conducted analysis, technical requirements were formulated and an IoT-based water supply metering system for multi-apartment residential buildings was designed.

During the design process, a metering node structure based on the ESP32-C6 microcontroller, a Hall-effect sensor, and an autonomous power supply was developed. The use of a Thread mesh network is proposed for data transmission between metering nodes and a central gateway, followed by forwarding the collected information to a server.

Algorithms for the operation of the metering node, server-side application, and client software were developed. The system's power consumption was evaluated, and the feasibility of providing long-term autonomous operation of the metering nodes was justified.

The proposed solution is characterized by low deployment cost, scalability, and the possibility of integration into modern utility resource monitoring systems. The use of a Thread mesh network minimizes the need for additional network infrastructure and ensures reliable data collection in multi-apartment residential buildings.

Keywords: Internet of Things (IoT), water supply metering, remote monitoring, water meter, ESP32-C6, Hall-effect sensor, Thread, mesh network, MQTT, battery-powered operation, wireless data transmission, energy efficiency, automated metering system.

ЗМІСТ

ВСТУП.....	7
1. АНАЛІЗ ІоТ СИСТЕМ ОБЛІКУ ВОДОПОСТАЧАННЯ	10
1.1 Загальні концепції розвитку систем обліку водопостачання	10
1.2 Аналіз сучасних систем дистанційного обліку водопостачання	11
1.3 Постановка задачі	14
2. ПРОЕКТУВАННЯ ІОТ СИСТЕМИ ОБЛІКУ ВОДОПОСТАЧАННЯ	17
2.1 Структурна схема ІоТ системи обліку водопостачання	17
2.2 Обґрунтування вибору апаратної платформи ESP32-C6 та протоколу Thread	19
2.3 Вибір водоміра.....	23
2.4 Обґрунтування вибору сенсора DRV5032FBLPGM.....	27
2.5 Структура проектованого вузла обліку водоспоживання.	30
2.5 Алгоритм роботи вузла обліку водоспоживання.....	33
2.6 Проєктування структури бази даних MongoDB.	39
3. РОЗРОБЛЕННЯ ІОТ СИСТЕМИ ОБЛІКУ ВОДОПОСТАЧАННЯ	43
3.1 Аналіз середовища проєктування ESP-IDF	43
3.2 Розроблення програмного забезпечення вузла обліку на базі ESP32-C6..	45
3.3 Розрахунок енергоспоживання вузла обліку.	50
3.4 Налаштування MQTT-брокера Mosquitto.	54
3.6 Розроблення серверного застосунку обробки даних.	57
3.7 Розроблення клієнтського програмного забезпечення.....	61
3.8 Розроблення модуля перегляду поточних показників.	66
ВИСНОВКИ	72
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	74
ДОДАТОК А	77

					ДП.МТІР.10658567.00.00.00.000 ПЗ			
Розроб	Палійчук Х.				ІоТ система обліку водопостачання/ IoT water supply accounting system	Літ.	Арк.	Аркушів
Перевірів	Заставний О.							
Консульт	Заставний О.					ЗУНУ.ФКІТ.МТІР-41		
Н.Контроль	Заставний О.							
Затвердив	Сегін А.І.							

ВСТУП

Актуальність роботи. Актуальність теми дослідження полягає у зростаючій потребі ефективного використання водних ресурсів та автоматизації процесів контролю водоспоживання. У сучасних умовах традиційні системи обліку водопостачання мають низку недоліків, серед яких необхідність ручного збору показників, затримка виявлення аварійних ситуацій, недостатня точність моніторингу та високі витрати на обслуговування.

Стрімкий розвиток технологій Інтернету речей (IoT) відкриває нові можливості для створення інтелектуальних систем обліку водопостачання, які забезпечують автоматичний збір, передачу та аналіз даних у режимі реального часу. Використання IoT-рішень дозволяє оперативно виявляти витоки води, контролювати рівень споживання, підвищувати точність обліку та зменшувати втрати ресурсів.

Особливої актуальності впровадження таких систем набуває в умовах зростання навантаження на комунальну інфраструктуру, необхідності енергозбереження та переходу до концепції «розумного міста» (Smart City). Автоматизовані системи моніторингу сприяють оптимізації роботи водопостачальних мереж, підвищенню якості обслуговування споживачів та зниженню експлуатаційних витрат.

Таким чином, розробка IoT системи обліку водопостачання є актуальним напрямом дослідження, що має практичне значення для підвищення ефективності управління водними ресурсами та впровадження сучасних цифрових технологій у сферу житлово-комунального господарства.

Мета роботи. Метою даної роботи є розробка IoT системи обліку водопостачання для автоматизованого збору, передачі та моніторингу даних про споживання води в режимі реального часу з метою підвищення точності обліку, зменшення втрат водних ресурсів та оптимізації роботи систем водопостачання.

Для досягнення поставленої мети необхідно:

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підп.	Дата		

- проаналізувати існуючі системи обліку водопостачання та сучасні IoT-технології;
- обрати апаратні та програмні засоби для реалізації системи;
- розробити структуру та алгоритм роботи системи;
- реалізувати передачу даних між пристроями;
- створити інтерфейс для моніторингу показників водоспоживання;
- провести тестування роботи системи та оцінити її ефективність.

Предметом дослідження. Предметом дослідження є методи та засоби автоматизованого обліку водопостачання із застосуванням технологій Інтернету речей (IoT), а також процеси збору, передачі, обробки та моніторингу даних про споживання води в режимі реального часу.

Об’єкт дослідження. Об’єктом дослідження є система водопостачання та процес обліку споживання води з використанням сучасних інформаційних і телекомунікаційних технологій.

Методи дослідження.

У роботі використано методи аналізу та узагальнення науково-технічної літератури, методи системного аналізу для проєктування структури IoT системи, методи програмування та моделювання для реалізації програмного забезпечення, а також методи експериментального дослідження для перевірки працездатності та оцінки ефективності системи обліку водопостачання.

Практичне значення одержаних результатів.

Практичне значення одержаних результатів полягає у розробці та реалізації IoT системи обліку водопостачання на базі мікроконтролера ESP32-C6 із використанням протоколу MQTT, що забезпечує автоматизований збір, передачу та моніторинг даних про споживання води в режимі реального часу.

Розроблена система може бути використана в житлових будинках, комунальних підприємствах та промислових об’єктах для підвищення точності обліку води, оперативного виявлення витоків та зменшення втрат водних ресурсів. Використання мікроконтролера ESP32-C6 дозволяє забезпечити низьке

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		8

енергоспоживання, бездротову передачу даних через Wi-Fi та можливість інтеграції з вебсервісами і мобільними застосунками.

Застосування протоколу MQTT забезпечує надійний обмін даними між пристроями системи, швидку передачу повідомлень та масштабованість рішення для подальшого розширення функціоналу в межах концепції «розумного будинку» або «розумного міста».

Апробація. І. Мелько, Х. Палійчук, В. Сокальська, О. Заставний Система автоматизованого керування електричними зарядними станціями для електромобілів. – Збірник матеріалів всеукраїнської науково-практичної конференції «Автоматизація та комп'ютерно-інтегровані технології» (АКІТ - 2026), Тернопіль, 2026. – ст. 245-248.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підп.	Дата		

1. АНАЛІЗ ІоТ СИСТЕМ ОБЛІКУ ВОДОПОСТАЧАННЯ

1.1 Загальні концепції розвитку систем обліку водопостачання

Сучасні системи обліку водопостачання активно впроваджують технології Інтернету речей (ІоТ)[1], що дозволяє автоматизувати процес збору та обробки даних про водоспоживання. Основною метою таких систем є підвищення точності обліку, своєчасне виявлення витоків та зменшення експлуатаційних витрат.

Одним із поширених рішень є системи AMR[4] (Automated Meter Reading), які забезпечують автоматичне зчитування показників лічильників та передачу даних на сервер без необхідності ручного збору інформації. Такі системи підвищують точність обліку, проте зазвичай передають дані через великі проміжки часу та мають обмежені можливості оперативного моніторингу.

Більш сучасним підходом є використання АМІ [2,3,4,5] (Advanced Metering Infrastructure), що передбачає двосторонній обмін даними між лічильниками та сервером. АМІ-системи дозволяють отримувати інформацію про водоспоживання в режимі реального часу, автоматично формувати звіти, виявляти аварійні ситуації та контролювати стан мережі.

Комерційні рішення у сфері Smart Water Management, такі як системи компаній Verteco[6,7] та iEngineering[8,9], використовують ІоТ-пристрої для безперервного моніторингу споживання води, аналізу даних та виявлення витоків. Дані передаються на сервери через бездротові мережі та відображаються у вебінтерфейсах або мобільних застосунках. Такі системи підтримують інтеграцію з SCADA-системами та промисловими платформами моніторингу.

У сучасних дослідженнях значна увага приділяється використанню ІоТ для побудови масштабованих систем моніторингу водорозподільних мереж. Запропоновані архітектури дозволяють реалізовувати контроль витрат води, аналіз споживання та автоматичне виявлення витоків із використанням хмарних сервісів та периферійних обчислень (Edge Computing).

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		10

Перспективним напрямком є створення недорогих систем на базі мікроконтролерів сімейства ESP32. Зокрема, у роботі AquOculus запропоновано систему автоматизованого обліку води на базі ESP32 та Wi-Fi, орієнтовану на зниження вартості впровадження та забезпечення функцій виявлення витоків у водопровідних мережах.

Для передачі даних у IoT-системах широко використовуються протоколи MQTT, Modbus TCP та REST API. Особливо популярним є протокол MQTT завдяки низькому мережевому навантаженню, високій швидкості обміну повідомленнями та можливості роботи на пристроях з обмеженими ресурсами. MQTT активно застосовується в системах моніторингу водопостачання для передавання телеметричних даних між датчиками, шлюзами та сервером.

Проведений аналіз показав, що існуючі рішення забезпечують ефективний моніторинг водоспоживання, проте більшість комерційних систем характеризуються високою вартістю впровадження та складністю масштабування. Тому актуальним є створення власної IoT системи обліку водопостачання на базі мікроконтролера ESP32-S3 та протоколу MQTT, яка поєднуватиме низьку вартість, простоту інтеграції та можливість моніторингу даних у режимі реального часу.

1.2 Аналіз сучасних систем дистанційного обліку водопостачання

Розвиток технологій Інтернету речей (IoT) сприяв широкому впровадженню систем дистанційного обліку комунальних ресурсів, зокрема водопостачання. Сучасні системи дистанційного обліку дозволяють автоматизувати процес збору показників, зменшити витрати на обслуговування мереж, підвищити точність розрахунків та забезпечити оперативне виявлення аварійних ситуацій і витоків води. На ринку представлені як окремі інтелектуальні водоміри, так і комплексні системи автоматизованого збору та обробки даних.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		11

Одним із відомих європейських виробників є компанія Apator Powogaz, яка пропонує серію водомірів JS Smart+[10]. Дані пристрої являють собою механічні крильчасті водоміри з можливістю встановлення модулів дистанційного зчитування показників через імпульсний вихід, Wireless M-Bus або M-Bus інтерфейс. Конструкція водоміра дозволяє встановлювати модулі зв'язку без порушення заводського пломбування та метрологічних характеристик приладу. Крім того, система підтримує функції виявлення витоків, зворотного потоку води та несанкціонованого втручання.

Для ілюстрації даної системи доцільно використати фотографії водомірів JS Smart+ та модулів дистанційного зчитування з офіційного сайту Apator

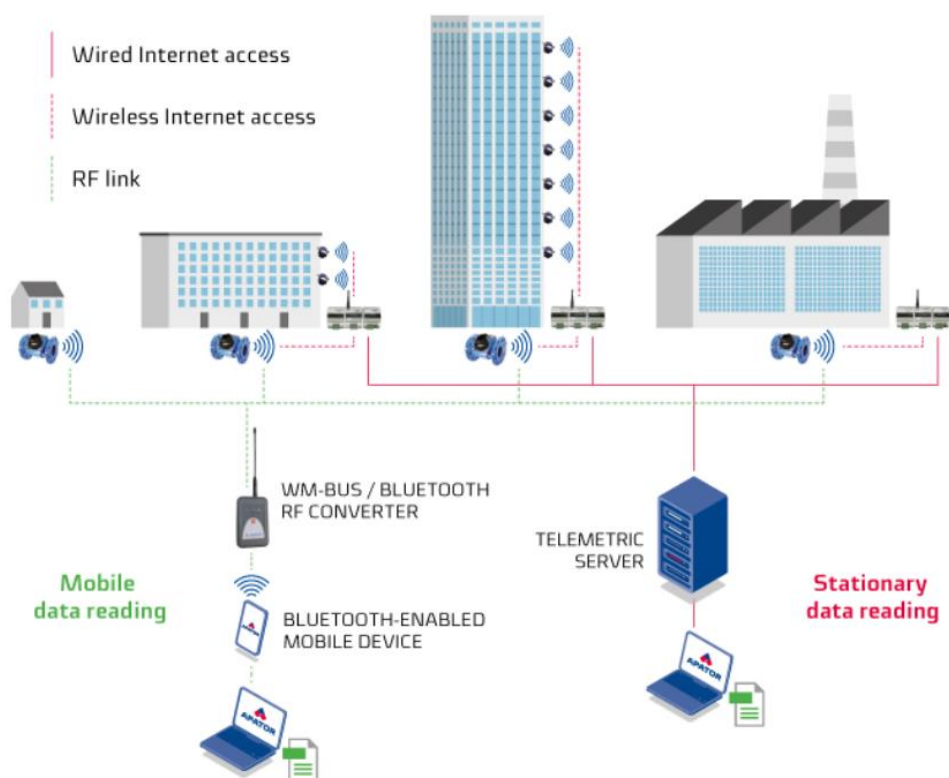


Рисунок 1.1 – Структурна схема з використанням AMR Apator

Іншим поширеним рішенням є системи компанії ZENNER, які використовують технології Wireless M-Bus та LoRaWAN для організації дистанційного збору показників. Водоміри ZENNER можуть передавати дані через радіомодулі на локальні шлюзи або безпосередньо до LoRaWAN-мережі.

Перевагою даних систем є можливість централізованого збору інформації з великої кількості вузлів, а також інтеграція з платформами моніторингу та аналітики. Сучасні рішення ZENNER підтримують автоматичне виявлення витоків, формування статистики споживання та прогнозування навантаження на водопровідні мережі.

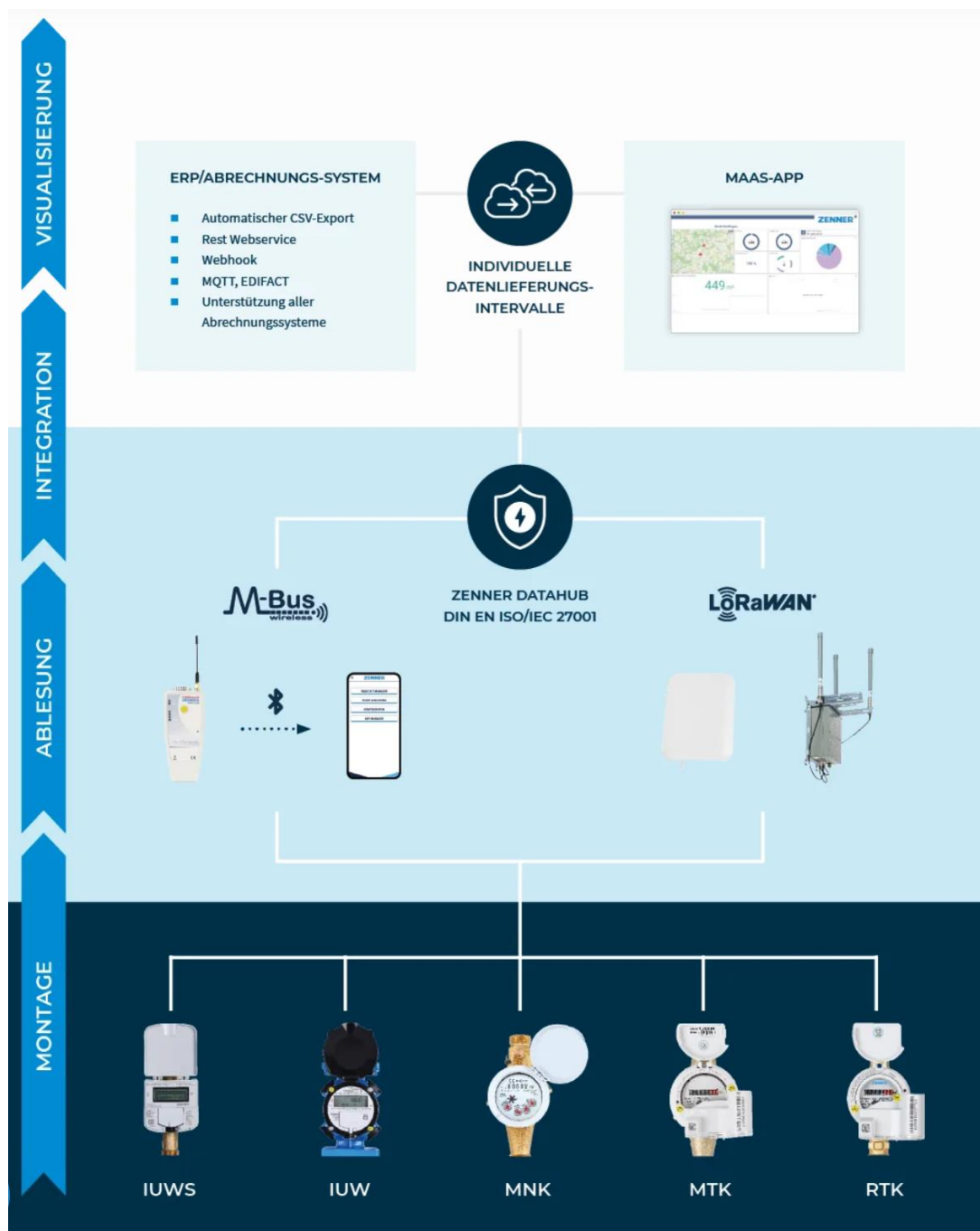


Рисунок 1.2 – Структура системи V.One компанії Zenper

Особливий інтерес становлять ультразвукові водоміри нового покоління, наприклад серія IUWS B.One[11] компанії ZENNER. Дані пристрої не містять механічних рухомих елементів, що підвищує точність вимірювання та збільшує термін експлуатації. Водоміри підтримують інтеграцію з радіомодулями дистанційного зчитування та забезпечують додаткові функції контролю аварійних ситуацій.

До недоліків більшості існуючих систем можна віднести використання спеціалізованих протоколів зв'язку, необхідність розгортання окремої інфраструктури Wireless M-Bus або LoRaWAN, а також порівняно високу вартість обладнання. Крім того, значна частина комерційних рішень орієнтована на централізовані системи обліку водоканалів і не завжди є економічно доцільною для впровадження в окремих багатоквартирних будинках.

Аналіз сучасних рішень показує, що найбільш перспективним напрямком розвитку є використання енергоефективних IoT-технологій, які забезпечують автоматизований збір даних, тривалий термін автономної роботи та можливість побудови масштабованих мереж. Водночас існуючі комерційні системи переважно використовують зіркоподібну архітектуру або потребують додаткової інфраструктури передачі даних. Це обґрунтовує доцільність розроблення власної системи на базі протоколу Thread та технології mesh-маршрутизації, яка дозволяє використовувати самі вузли обліку як елементи мережевої інфраструктури та зменшити витрати на розгортання системи в багатоквартирних будинках.

1.3 Постановка задачі

Сучасні системи обліку водопостачання в багатоквартирних будинках переважно базуються на механічних або електронних лічильниках, показники яких зчитуються вручну або передаються за допомогою локальних засобів зв'язку. Такий підхід потребує значних трудових витрат, не забезпечує оперативного отримання інформації про споживання води та ускладнює виявлення аварійних ситуацій і витоків.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		14

На відміну від систем обліку електроенергії, вузли обліку водопостачання зазвичай не мають постійного джерела електроживлення. Використання енергії потоку води для живлення електронних пристроїв є технічно складним і недостатньо ефективним через низький рівень доступної потужності. Тому доцільним є застосування автономних вузлів обліку з батарейним живленням. Враховуючи, що міжповірочний інтервал більшості побутових водолічильників становить близько чотирьох років, система повинна забезпечувати автономну роботу без заміни елементів живлення протягом не менше п'яти років.

Особливістю багатоквартирних будинків є компактне розташування вузлів обліку. Як правило, лічильники води встановлюються на стояках кожного поверху, причому в одній квартирі може використовуватися декілька лічильників для холодної та гарячої води. Невелика відстань між пристроями створює сприятливі умови для побудови бездротової самоорганізованої мережі типу Mesh, у якій кожен вузол може виступати ретранслятором даних для сусідніх вузлів. Такий підхід дозволяє підвищити надійність зв'язку та зменшити вимоги до потужності передавачів.

У зв'язку з цим виникає задача розробки IoT системи обліку водопостачання для багатоквартирних будинків, яка забезпечуватиме автоматизований збір даних з водолічильників, їх передачу через бездротову Mesh-мережу до центрального шлюзу та подальшу доставку на сервер за допомогою мережі Ethernet і протоколу MQTT.

Для вирішення поставленої задачі необхідно:

- провести аналіз існуючих технологій обліку водоспоживання та бездротової передачі даних;
- розробити архітектуру IoT системи з автономними вузлами обліку на базі батарейного живлення;
- реалізувати бездротову Mesh-мережу для обміну даними між вузлами;
- забезпечити передачу інформації до центрального шлюзу та сервера збору даних через протокол MQTT;

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		15

- розробити програмне забезпечення для збору, зберігання та відображення інформації про водоспоживання;
- оцінити енергоспоживання системи та можливість забезпечення автономної роботи протягом заданого терміну експлуатації.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підп.	Дата		

2. ПРОЕКТУВАННЯ ІОТ СИСТЕМИ ОБЛІКУ ВОДОПОСТАЧАННЯ

2.1 Структурна схема ІоТ системи обліку водопостачання

Структурна схема ІоТ системи обліку водопостачання для багатоквартирних будинків наведена на рисунку 2.1. Система складається з вузлів обліку водоспоживання, бездротової mesh-мережі Thread[12], центрального шлюзу (Border Router), серверної частини та клієнтських застосунків.

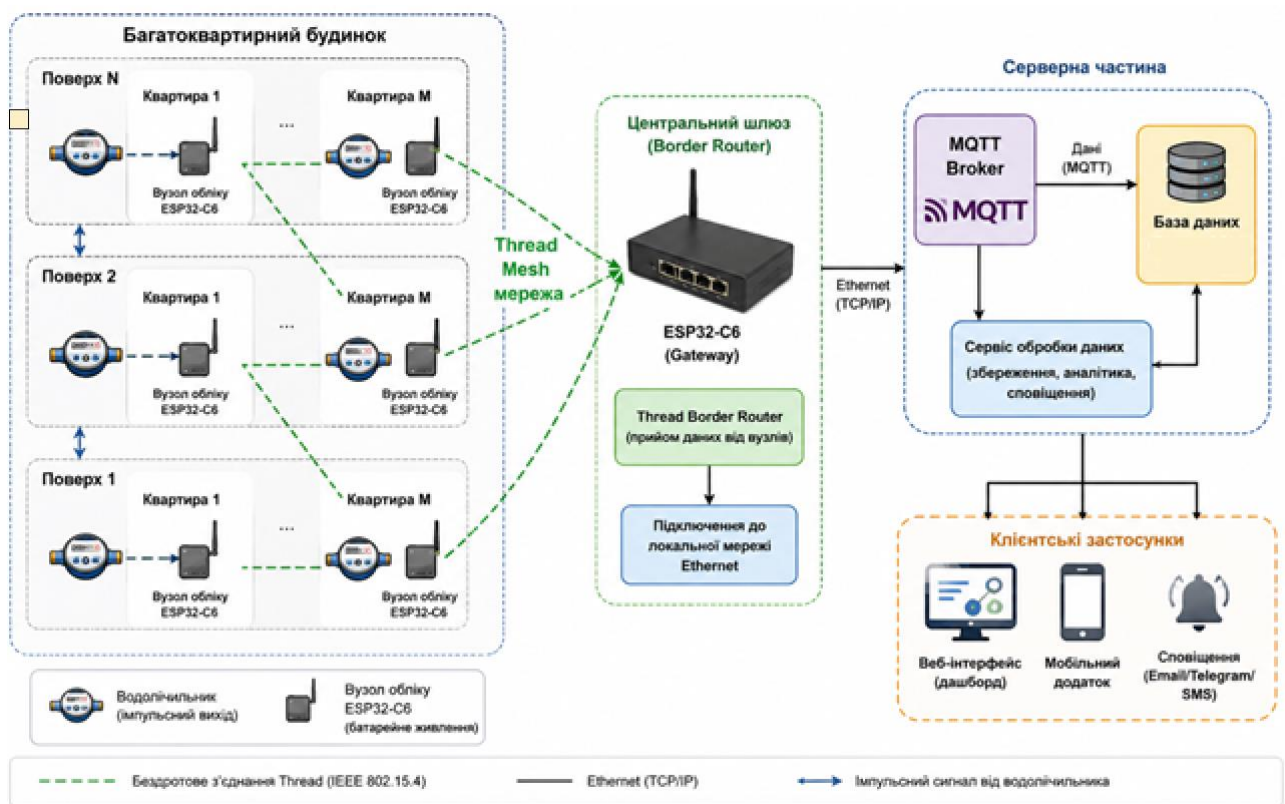


Рисунок 2.1 - Структурна схема ІоТ системи обліку водопостачання

Кожен вузол обліку побудований на базі мікроконтролера ESP32-C6[13] та підключений до водолічильника з імпульсним виходом. Під час проходження певного об'єму води лічильник формує імпульси, які підраховуються мікроконтролером. Для забезпечення тривалої автономної роботи вузол живиться від батареї та більшу частину часу перебуває в режимі зниженого

енергоспоживання, активуючись лише для обробки імпульсів та передачі накопичених даних.

Передача інформації між вузлами здійснюється через бездротову mesh-мережу Thread, побудовану на базі стандарту IEEE 802.15.4. Особливістю такої мережі є можливість ретрансляції пакетів через сусідні вузли, що дозволяє забезпечити надійне покриття всієї будівлі без необхідності прямого зв'язку кожного пристрою із центральним шлюзом. Завдяки компактному розташуванню лічильників на поверхах багатоквартирного будинку формується щільна мережа взаємопов'язаних вузлів, яка характеризується високою відмовостійкістю та можливістю автоматичного відновлення маршрутів передачі даних.

Усі дані від вузлів надходять до центрального шлюзу, який виконує функції Thread Border Router. Шлюз забезпечує взаємодію між локальною мережею Thread та локальною комп'ютерною мережею будинку. Для підключення до серверної частини використовується інтерфейс Ethernet, що забезпечує стабільний канал зв'язку та не залежить від якості бездротового покриття.

Обмін даними між шлюзом та сервером здійснюється за допомогою протоколу MQTT[14], який реалізує модель publish/subscribe. Кожен вузол обліку передає дані до відповідного MQTT-топіка, що дозволяє організувати централізований збір інформації та спростити масштабування системи.

Серверна частина включає MQTT-брокер, сервіс обробки даних та базу даних MongoDB[15]. MQTT-брокер приймає повідомлення від шлюзу та передає їх сервісу обробки даних. Отримані показники проходять перевірку, агрегування та зберігаються в базі даних MongoDB. Використання MongoDB дозволяє ефективно працювати з часовими рядами та документно-орієнтованими структурами даних, що добре відповідає характеру інформації про водоспоживання.

Для доступу до інформації користувачів передбачено веб-інтерфейс та мобільний застосунок. Вони забезпечують перегляд поточних показників

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підп.	Дата		

лічильників, аналіз історії споживання води, формування звітів та отримання сповіщень про аварійні ситуації, такі як можливі витoki води або тривале безперервне споживання.

Таким чином, запропонована архітектура поєднує енергоефективні вузли обліку на базі ESP32-C6, надійну Thread mesh-мережу, MQTT-протокол передачі повідомлень та базу даних MongoDB, що забезпечує створення масштабованої та відмовостійкої системи автоматизованого обліку водопостачання для багатоквартирних будинків.

2.2 Обґрунтування вибору апаратної платформи ESP32-C6 та протоколу Thread

Одним із ключових завдань під час проєктування IoT системи обліку водопостачання є вибір апаратної платформи та технології бездротового зв'язку, які забезпечуватимуть надійну роботу системи протягом тривалого часу при мінімальному енергоспоживанні. Оскільки вузли обліку планується встановлювати безпосередньо біля водолічильників, де зазвичай відсутнє зовнішнє джерело живлення, основною вимогою до обладнання є можливість автономної роботи від батареї протягом декількох років.

Для вибору апаратної платформи вузла обліку було проведено порівняння декількох популярних мікроконтролерів (Таблиця 2.1), які можуть використовуватися в IoT системах з батарейним живленням.

Таблиця 2.1. Порівняння можливих апаратних платформ

Характеристика	ESP32-C6-Zero	ESP32-S3-Zero	nRF52840
Архітектура	RISC-V 32-bit	Xtensa LX7	ARM Cortex-M4
Тактова частота	до 160 МГц	до 240 МГц	до 64 МГц
Wi-Fi	Wi-Fi 6	Wi-Fi 4	Відсутній
Bluetooth LE	Так	Так	Так
IEEE 802.15.4 (Thread/Zigbee)	Так	Ні	Так

Продовження таблиці 2.1

Характеристика	ESP32-C6-Zero	ESP32-S3-Zero	nRF52840
Thread	Так	Ні	Так
Mesh-мережі	Так	Лише Wi-Fi Mesh	Так
Обсяг Flash	4 МБ	4 МБ	до 1 МБ
Орієнтація на батареїне живлення	Висока	Середня	Дуже висока
Вартість модуля	Низька	Низька	Висока
Складність інтеграції з MQTT та Ethernet-шлюзом	Низька	Низька	Середня

Як видно з таблиці, ESP32-S3-Zero хоча й має вищу продуктивність, проте не підтримує стандарт IEEE 802.15.4, а отже не може використовувати протокол Thread без додаткового радіомодуля. Це унеможлиблює створення енергоефективної mesh-мережі, яка є основою запропонованої системи.

Мікроконтролер nRF52840 широко використовується в Thread та Zigbee мережах і характеризується дуже низьким енергоспоживанням. Однак він не має вбудованого Wi-Fi інтерфейсу та зазвичай має вищу вартість у порівнянні з рішеннями сімейства ESP32. Крім того, для побудови шлюзу та інтеграції з локальною мережею можуть знадобитися додаткові апаратні компоненти.

ESP32-C6-Zero поєднує підтримку Wi-Fi 6, Bluetooth LE та IEEE 802.15.4 в одному мікроконтролері, що дозволяє реалізувати як вузли обліку, так і шлюз системи на єдиній апаратній платформі. Наявність підтримки Thread забезпечує побудову енергоефективної mesh-мережі, а низька вартість модуля робить систему економічно доцільною для масового впровадження в багатоквартирних будинках.

Таким чином, для реалізації вузлів обліку обрано мікроконтролер ESP32-C6 компанії Espressif. Даний мікроконтролер побудований на базі 32-бітного ядра RISC-V та підтримує одночасно декілька бездротових технологій: Wi-Fi 6, Bluetooth Low Energy та IEEE 802.15.4. Наявність підтримки IEEE 802.15.4 є особливо важливою для даного проєкту, оскільки дозволяє використовувати

сучасні протоколи побудови mesh-мереж, зокрема Thread. На рисунку 2.2 наведено зовнішній вигляд модуля ESP32-C6-Zero.

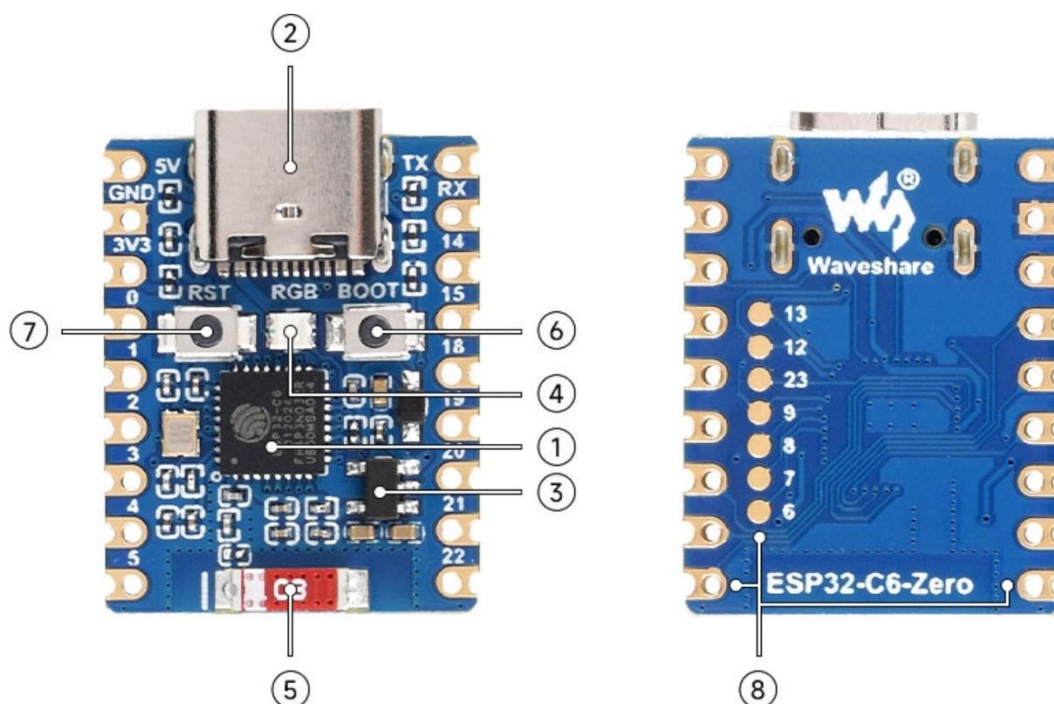


Рисунок 2.2 - Зовнішній вигляд модуля ESP32-C6-Zero.

На даному рисунку позначені наступні компоненти:

1. Мікроконтролер ESP32-C6FH4;
2. USB-Type C;
3. Стабілізатор напруги ME6217C33M5G (LDO);
4. Керований RGB світлодіод WS2812;
5. Керамічна антена 2.4G;
6. Кнопка BOOT, для прошивки мікроконтролера;
7. Кнопка RESET;
8. Основні та додаткові GPIO контакти.

У порівнянні з ESP32-S3, який орієнтований переважно на використання Wi-Fi та Bluetooth, мікроконтролер ESP32-C6 краще підходить для створення автономних IoT пристроїв завдяки підтримці малопотужних мереж IEEE 802.15.4. Це дозволяє значно зменшити енергоспоживання вузлів під час передачі даних та збільшити термін служби елементів живлення.

Вибір протоколу Thread обумовлений його архітектурними особливостями та орієнтацією на пристрої Інтернету речей з батарейним живленням. На відміну від традиційних Wi-Fi мереж, де кожен пристрій повинен підтримувати прямий зв'язок із точкою доступу, Thread використовує mesh-топологію, що дозволяє передавати дані через проміжні вузли. Це особливо актуально для багатоквартирних будинків, де водолічильники розташовані компактно на поверхових стояках та можуть взаємодіяти між собою.

Важливою перевагою Thread є підтримка механізмів самоорганізації та самовідновлення мережі. У випадку виходу з ладу окремого вузла або погіршення якості радіоканалу мережа автоматично перебудовує маршрути передачі даних без втручання користувача. Завдяки цьому забезпечується висока надійність роботи системи та зменшується ймовірність втрати даних.

Додатковою перевагою Thread є використання IPv6-адресації. Кожен вузол мережі може мати власну IP-адресу, що значно спрощує інтеграцію системи з мережевими сервісами та платформами Інтернету речей. Для підключення Thread-мережі до локальної мережі використовується спеціальний пристрій Border Router, який забезпечує маршрутизацію трафіку між Thread та Ethernet сегментами мережі.

З точки зору енергоспоживання Thread також має суттєві переваги над Wi-Fi. Передача невеликих пакетів даних через IEEE 802.15.4 потребує значно менше енергії, ніж використання Wi-Fi-з'єднання. Оскільки показники водоспоживання змінюються відносно повільно та не потребують безперервної передачі великих обсягів інформації, вузол обліку може перебувати в режимі глибокого сну більшу частину часу та активуватись лише для обробки імпульсів лічильника і періодичної передачі накопичених даних.

Таким чином, поєднання мікроконтролера ESP32-C6 та протоколу Thread дозволяє створити енергоефективну, масштабовану та відмовостійку систему обліку водопостачання, яка відповідає вимогам автономної роботи протягом тривалого часу та забезпечує надійну передачу даних у межах багатоквартирного будинку.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		22

2.3 Вибір водоміра

Одним із ключових елементів системи автоматизованого обліку водопостачання є водомір, який забезпечує вимірювання об'єму спожитої води та формування даних для подальшої обробки інформаційною системою. Від типу водоміра залежать точність вимірювань, можливість автоматизації збору даних, енергоспоживання вузла обліку та загальна вартість системи.

У сучасних системах водопостачання найбільшого поширення набули механічні, електронні та ультразвукові водоміри.

Механічні водоміри (Рисунок 2.3) є найпоширенішими у житловому секторі завдяки простій конструкції, невисокій вартості та відсутності потреби у зовнішньому живленні. Принцип їх роботи базується на обертанні крильчатки під дією потоку води. Кількість обертів через механічну передачу або магнітну муфту перетворюється у показники спожитого об'єму води. Основними перевагами таких лічильників є надійність, тривалий термін експлуатації та низька вартість. Недоліком є відсутність штатних засобів дистанційного збору даних у базових моделях.



Рисунок 2.3 – Зовнішній вигляд типового механічного водоміра

Для автоматизації обліку широко використовуються механічні водоміри з імпульсним виходом (рисунок 2.4). У таких пристроях кожному певному об'єму води відповідає електричний імпульс, який може бути зчитаний зовнішньою

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		23

електронною системою. Найчастіше імпульсний вихід реалізується за допомогою геркона, що спрацьовує під впливом магніту, встановленого на обертовому елементі лічильника. Такі водоміри добре підходять для інтеграції з автоматизованими системами обліку, однак герконові контакти схильні до дребезгу та мають обмежений ресурс перемикачів.



Рисунок 2.4 – Зовнішній вигляд типового механічного водоміра з імпульсним виходом на базі геркона

Електронні водоміри оснащуються вбудованими мікроконтролерами та цифровими інтерфейсами передачі даних, такими як M-Bus, Wireless M-Bus, LoRaWAN або NB-IoT. Вони забезпечують високу точність вимірювання та готовність до інтеграції у системи дистанційного моніторингу. Водночас їхня вартість суттєво вища порівняно з механічними аналогами, а використання власних протоколів зв'язку часто ускладнює інтеграцію з розроблюваними системами.



Рисунок 2.5 - Зовнішній вигляд водоміра з LoRaWAN

Ультразвукові водоміри є найбільш сучасним типом вимірювальних приладів (рисунок 2.6). Вони визначають витрату води шляхом вимірювання часу проходження ультразвукового сигналу через потік рідини. Такі пристрої характеризуються високою точністю, відсутністю рухомих частин та мінімальним зношуванням. Однак висока вартість та складність конструкції обмежують їх застосування переважно промисловими та комерційними об'єктами.



Рисунок 2.6 – Зовнішній вигляд ультразвукового водоміра

Для розроблюваної IoT системи обліку водопостачання найбільш доцільним є використання механічних водомірів із можливістю формування імпульсного сигналу. Такі лічильники широко використовуються у

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		25

багатоквартирних будинках, мають доступну вартість та забезпечують достатню точність вимірювання для побутового обліку води.

Слід відмітити що водоміри з герконовим виходом є зазвичай разів в 4 дорожчі від звичайних механічних. При цьому опрацювання так званого «сухого контакту», в даному випадку яким виступає геркон, є зазвичай алгоритмічно складнішим, оскільки додається алгоритм обробки «брязкіту» контактів. Хоча він і не сильно складний і зачасту просто додає тривалу паузу для ігнорування деяких спрацювань сенсора (в даному випадку геркона) поки контакт сенсора не стабілізується і стане постійно замкнутим, аналогічно при розмиканні. Незважаючи на всю простоту алгоритму це не дозволяє максимально використовувати наявні можливості сучасних мікроконтролерів. Тобто наприклад це унеможлиблює використання переривань чи апаратних лічильників, які дозволяють рахувати імпульси в режимі «сну», оскільки кожне переривання «пробуджує» процесор мікроконтролера та таким чином значною мірою підвищує енергоспоживання системи.

На відміну від герконового датчика, сенсор Холла не містить механічних контактів, не створює дребезгу сигналу та має практично необмежений ресурс спрацювань. Крім того, використання малопотужного сенсора, наприклад DRV5032, дозволяє реалізувати надійну систему підрахунку імпульсів із мінімальним впливом на загальне енергоспоживання вузла обліку.

Окрім того, зараз на ринку існує чимало механічних лічильників адаптованих під використання сенсорів (електронних або герконів). Такі водоміри зазвичай вдвічі дорожчі від простих механічних, проте дозволяють використовувати свої сенсори без втручання в роботу водоміра та непошкоджуючи пломб. Приклад таких лічильників наведено на рисунку 2.7.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		26

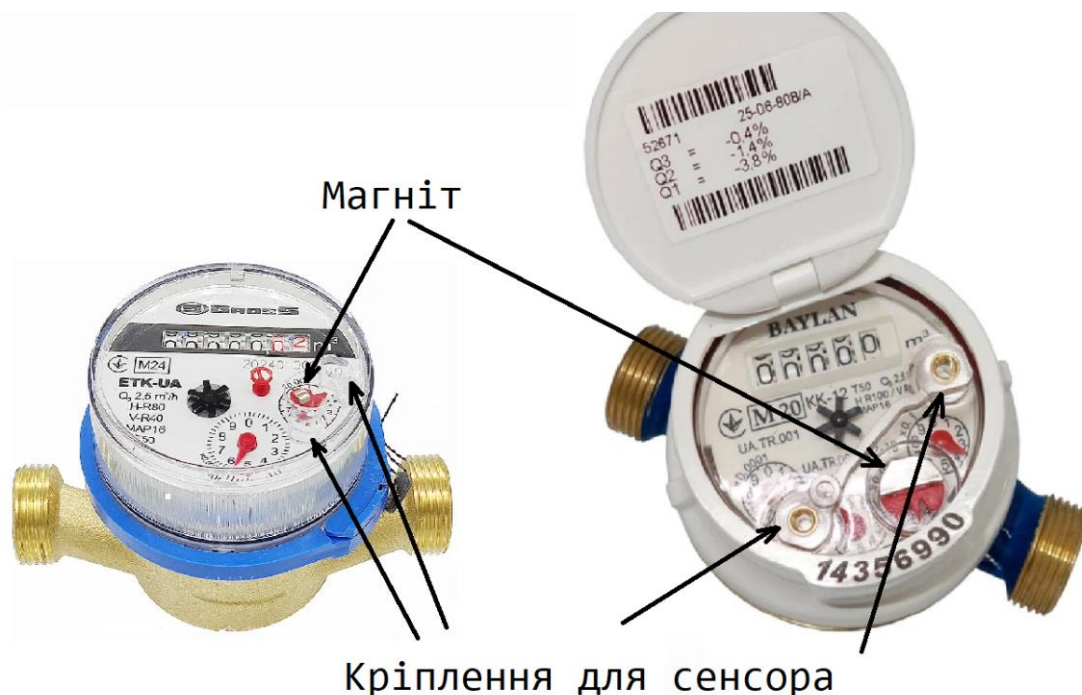


Рисунок 2.7 – Механічні водоміри виробників Gross та Baylan з додатковими магнітами та кріпленнями під сенсорні вузли

На даному рисунку видно, що вони містять додатковий невеликий магніт на механізмі що обертається, та відповідні кріплення для того щоб надійно закріпити сам сенсор. Єдиним недоліком є відсутність уніфікації таких систем, тому кріплення мають бути різні для різних моделей водомірів та різних виробників.

Таким чином, для реалізації системи автоматизованого обліку водопостачання доцільно обрати механічний водомір із магнітом для можливості безконтактного зчитування показників за допомогою цифрового сенсора Холла DRV5032. Таке рішення забезпечує оптимальне поєднання вартості, надійності, точності та енергоефективності.

2.4 Обґрунтування вибору сенсора DRV5032FBLPGM.

Для реєстрації обертів механізму водоміра в розроблюваній системі обрано цифровий магнітний сенсор Холла DRV5032FBLPGM[16]. Даний сенсор призначений для батарейних пристроїв та характеризується надзвичайно

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		27

низьким енергоспоживанням, що робить його придатним для використання в автономних IoT-вузлах обліку водоспоживання.

Основні технічні характеристики DRV5032FBLPGM наведено в таблиці 2.2

Таблиця 2.1. Технічні характеристики DRV5032FBLPGM

Параметр	Значення
Напруга живлення	1,65–5,5 В
Тип виходу	Цифровий (Open Drain)
Частота опитування магнітного поля	5 Гц
Середній струм споживання	близько 0,54 мкА
Робочий температурний діапазон	від -40°C до +85°C
Корпус	ТО-92
Тип сенсора	Unipolar Hall Effect Switch

Сенсор DRV5032FBLPGM[16] працює за принципом періодичного опитування магнітного поля. На відміну від класичних датчиків Холла, які безперервно контролюють магнітне поле та споживають десятки або сотні мікроампер, DRV5032 виконує короткочасні вимірювання через задані інтервали часу та більшу частину часу перебуває у внутрішньому режимі сну. Завдяки цьому середній струм споживання становить лише близько 0,54 мкА.

У більшості сучасних побутових водомірів передбачена можливість встановлення зовнішнього імпульсного сенсора. Для цього на літровій стрілці рахункового механізму встановлюється постійний магніт, а на корпусі індикаторної частини передбачаються спеціальні отвори, посадкові місця або кріпильні бонки для монтажу датчика. Найчастіше виробниками пропонуються герконові модулі, однак принцип формування сигналу залишається однаковим незалежно від типу сенсора — один повний оберт літрової стрілки відповідає одному літру води та формує один імпульс.

Такий підхід забезпечує певну уніфікацію конструкції водомірів різних виробників і дозволяє реалізувати автоматизований збір показників без втручання у вимірювальний механізм приладу. Встановлення сенсора здійснюється зовні лічильника у спеціально передбаченому виробником місці,

тому не потребує розбирання корпусу, порушення заводських пломб або внесення змін до метрологічної частини приладу.

Для розроблюваної системи пропонується використовувати цифровий датчик Холла DRV5032FBLPGM, який встановлюється у штатному місці кріплення імпульсного модуля. Під час проходження магніту, закріпленого на літрової стрілці, сенсор формує цифровий сигнал, який надходить на мікроконтролер ESP32-C6. Таким чином забезпечується безконтактне зчитування показників водоміра із збереженням його штатної конструкції та метрологічних характеристик.

Оскільки один оберт літрової стрілки відповідає одному літру води, сенсор формує один імпульс на літр. Для типового побутового водоміра DN15 із максимальною витратою $Q_3 = 2,5 \text{ м}^3/\text{год}$ максимальна частота імпульсів становить приблизно 0,7 Гц. З урахуванням того, що модифікація DRV5032FBLPGM виконує опитування магнітного поля з частотою 5 Гц, сенсор має більш ніж семикратний запас за швидкодією навіть при максимальній витраті води, що гарантує надійну реєстрацію кожного оберту літрової стрілки.

Для розрахунку достатності частоти опитування розглянемо максимальний режим роботи лічильника. Типовий побутовий водомір класу DN15 має максимальну витрату близько $2,5 \text{ м}^3/\text{год}$, що відповідає 2500 літрам води на годину. Якщо один повний оберт вимірювального механізму відповідає одному літру води, максимальна частота формування імпульсів становитиме:

$$2500 / 3600 = 0,694 \text{ імпульса за секунду.}$$

Сенсор DRV5032FBLPGM виконує вимірювання магнітного поля з частотою 5 Гц, тобто кожні 200 мс. За одну секунду здійснюється п'ять вимірювань стану магнітного поля. Відповідно за мінімальний інтервал між імпульсами (0,7 с) сенсор встигає виконати приблизно 4 циклів вимірювання, що забезпечує надійне виявлення кожного оберту лічильника навіть при максимальній витраті води.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		29

Важливою перевагою використання DRV5032FBLPGM є відсутність механічних контактів. На відміну від герконових датчиків, сенсор Холла не створює дребезгу контактів і не потребує додаткової програмної або апаратної фільтрації сигналу. Це спрощує програмне забезпечення вузла обліку та зменшує кількість пробуджень мікроконтролера, що позитивно впливає на загальне енергоспоживання системи.

Таким чином, сенсор DRV5032FBLPGM повністю задовольняє вимоги розроблюваної системи щодо точності реєстрації імпульсів, енергоефективності, надійності та простоти інтеграції з механічними побутовими водомірами. Його використання дозволяє реалізувати вузол обліку з мінімальним енергоспоживанням та створює передумови для забезпечення автономної роботи системи протягом декількох років від одного елемента живлення.

2.5 Структура проектованого вузла обліку водоспоживання.

Проектований вузол обліку водоспоживання призначений для автоматичного збору даних з побутового водоміра та їх передачі до IoT системи моніторингу. Вузол складається з механічного водоміра з магнітним виходом, цифрового сенсора Холла DRV5032FBLPGM, модуля ESP32-C6-Zero та автономного джерела живлення. На рисунку 2.8 наведено блок-схему даного вузла.

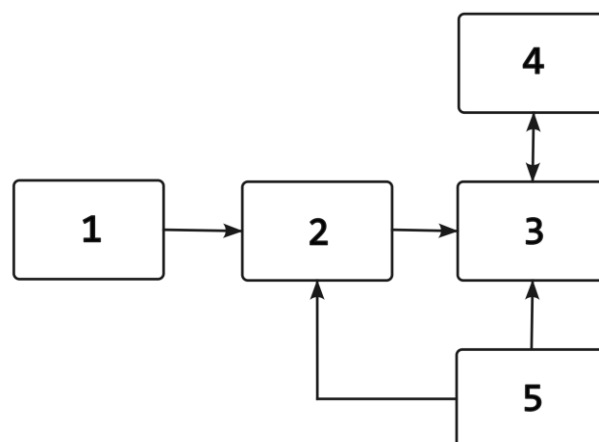


Рисунок 2.8 – Структурна схема облікового модуля системи

На даному рисунку позначені наступні елементи:

- 1 – механічний лічильник з магнітним виходом;
- 2 – сенсор Холла;
- 3 – модуль мікроконтролера ESP32-C6;
- 4 – модуль керування та індикації;
- 5 – елемент живлення.

Основним вимірювальним елементом системи є механічний водомір (1). Для формування цифрових даних використовується магніт, розташований на літрової стрілці лічильника. Під час кожного повного оберту стрілки сенсор Холла DRV5032FBLPGM (2) реєструє проходження магнітного поля та формує цифровий імпульс. Оскільки один оберт літрової стрілки відповідає одному літру води, кожен зафіксований імпульс еквівалентний одному літру спожитої води.

Обробка імпульсів здійснюється мікроконтролерним модулем ESP32-C6-Zero(3). Мікроконтролер виконує підрахунок отриманих імпульсів, зберігає поточне значення лічильника у енергонезалежній пам'яті та формує повідомлення для передачі в мережу. Для забезпечення мінімального енергоспоживання більшу частину часу ESP32-C6 перебуває в режимі Deep Sleep та активується лише при надходженні імпульсу від сенсора або у визначені моменти часу для передачі накопичених даних.

Передача інформації до системи верхнього рівня здійснюється через бездротову мережу Thread, побудовану на базі стандарту IEEE 802.15.4. Кожен вузол є учасником mesh-мережі та може передавати власні дані або брати участь у маршрутизації повідомлень від інших пристроїв. Завдяки цьому забезпечується висока надійність зв'язку та можливість масштабування системи без встановлення додаткового мережевого обладнання.

Живлення вузла здійснюється від автономного джерела живлення (5). З урахуванням низького енергоспоживання сенсора DRV5032FBLPGM та використання режимів енергозбереження ESP32-C6 передбачається забезпечення багаторічної автономної роботи без заміни елемента живлення.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		31

Особливістю обраного модуля ESP32-C6-Zero є наявність вбудованого адресного RGB-світлодіода та кнопки BOOT, які використовуються як модуль керування та індикації (4). Дані елементи можуть бути використані для первинного налаштування пристрою під час введення в експлуатацію. Після встановлення вузла користувач або сервісний інженер може натиснути кнопку BOOT для переведення пристрою в режим конфігурування. У цьому режимі мікроконтролер активує Bluetooth Low Energy та стає доступним для підключення зі смартфона через спеціалізований мобільний застосунок.

Після встановлення з'єднання користувач отримує можливість виконати початкове налаштування вузла. Зокрема можуть бути задані параметри підключення до існуючої Thread-мережі, ідентифікатор пристрою, адреса MQTT-брокера, інтервал передачі показників, порогові значення для діагностичних повідомлень та інші експлуатаційні параметри. Після завершення налаштування конфігурація зберігається у внутрішній енергонезалежній пам'яті мікроконтролера, а Bluetooth-модуль вимикається для мінімізації енергоспоживання.

Для інформування користувача про поточний стан пристрою використовується вбудований RGB-світлодіод. За допомогою різних кольорів та режимів індикації можуть відображатися режими роботи пристрою, процес налаштування, стан підключення до Thread-мережі, передача даних на сервер, рівень заряду батареї або виникнення аварійних ситуацій.

Для спрощення процедури введення пристрою в експлуатацію передбачається використання режиму Commissioning Mode (режим первинного налаштування). Після першого ввімкнення або після скидання до заводських налаштувань вузол автоматично переходить у режим конфігурування та активує модуль Bluetooth Low Energy. У цьому режимі пристрій стає доступним для виявлення мобільним застосунком сервісного інженера або адміністратора системи.

Під час процедури налаштування користувач може виконати пошук доступних Thread-мереж, вибрати необхідну мережу та передати параметри

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		32

авторизації для приєднання вузла до існуючої інфраструктури. Крім того, через мобільний застосунок можуть задаватися індивідуальні параметри пристрою, зокрема ідентифікатор водоміра, місце встановлення, інтервал передачі даних, параметри MQTT-з'єднання, часові інтервали службових повідомлень та інші конфігураційні дані.

Після успішного завершення процедури налаштування всі параметри зберігаються у енергонезалежній пам'яті мікроконтролера. Далі пристрій виконує підключення до Thread-мережі та переходить у штатний режим роботи. Оскільки Bluetooth Low Energy використовується виключно на етапі введення в експлуатацію, після завершення конфігурації BLE-модуль повністю вимикається. Це дозволяє усунути зайве енергоспоживання та підвищити рівень інформаційної безпеки пристрою шляхом зменшення кількості активних бездротових інтерфейсів.

Повторна активація режиму Commissioning Mode можлива лише після натискання кнопки BOOT або виконання процедури скидання налаштувань. Такий підхід забезпечує зручність обслуговування пристрою протягом усього терміну експлуатації та водночас мінімізує енергоспоживання в штатному режимі роботи.

Таким чином, запропонована структура вузла обліку поєднує простоту монтажу, низьке енергоспоживання, можливість бездротового налаштування та інтеграцію в сучасну IoT інфраструктуру на базі протоколів Thread та MQTT.

2.5 Алгоритм роботи вузла обліку водоспоживання.

Алгоритм роботи вузла обліку водоспоживання побудований таким чином, щоб забезпечити мінімальне енергоспоживання при збереженні надійності збору та передачі даних. Основна ідея полягає в тому, що більшу частину часу пристрій перебуває в режимі глибокого сну (Deep Sleep), а активується лише під час виконання необхідних операцій. Блок-схема алгоритму роботи вузла обліку водоспоживання наведено в додатку А.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		33

Після подачі живлення мікроконтролер ESP32-C6 виконує ініціалізацію апаратних ресурсів, перевіряє наявність збереженої конфігурації та визначає режим подальшої роботи. Якщо пристрій запускається вперше або було виконано скидання налаштувань, активується режим первинного налаштування Commissioning Mode.

У режимі Commissioning Mode вмикається модуль Bluetooth Low Energy та запускається сервіс конфігурування. За допомогою мобільного застосунку користувач може підключитися до пристрою та виконати налаштування параметрів роботи, зокрема задати ідентифікатор водоміра, вибрати або створити Thread-мережу, налаштувати параметри MQTT-з'єднання, встановити період передачі даних та інші експлуатаційні параметри. Після завершення налаштування всі параметри записуються до енергонезалежної пам'яті мікроконтролера. Далі Bluetooth Low Energy повністю вимикається, а пристрій переходить у штатний режим роботи.

У штатному режимі вузол перебуває у режимі Deep Sleep. Цифровий сенсор Холла DRV5032FBLPGM контролює положення магніту на літровій стрілці водоміра та формує цифровий імпульс під час кожного повного оберту. Оскільки один оберт стрілки відповідає одному літру води, кожен зареєстрований імпульс збільшує внутрішній лічильник спожитої води на один літр.

Після надходження імпульсу від сенсора відбувається пробудження мікроконтролера через механізм переривань GPIO. Мікроконтролер фіксує подію, оновлює накопичене значення спожитого об'єму води та за необхідності записує нові дані до енергонезалежної пам'яті. Після завершення обробки пристрій знову переходить у режим сну.

Передача даних до IoT системи виконується не після кожного імпульсу, а через задані інтервали часу. Для цього використовується внутрішній таймер реального часу (RTC), який періодично пробуджує мікроконтролер. Після пробудження формується пакет даних, що містить ідентифікатор пристрою,

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		34

поточні показники водоміра, рівень заряду батареї, службову інформацію та часову мітку.

Сформований пакет передається через Thread mesh-мережу до центрального шлюзу. Шлюз приймає повідомлення та передає їх через локальну мережу до MQTT-брокера. Після успішного підтвердження доставки даних вузол повертається у режим низького енергоспоживання.

Для контролю працездатності пристрою передбачено періодичне передавання службових повідомлень. Такі повідомлення можуть містити інформацію про рівень заряду батареї, якість зв'язку з мережею Thread, кількість втрачених пакетів та інші діагностичні параметри. Це дозволяє своєчасно виявляти несправності та планувати технічне обслуговування обладнання.

Індикація стану роботи пристрою виконується за допомогою вбудованого RGB-світлодіода модуля ESP32-C6-Zero. У штатному режимі світлодіод вимкнений для економії енергії. Індикація використовується лише під час налаштування, оновлення програмного забезпечення або сигналізації про аварійні ситуації.

Таким чином, запропонований алгоритм забезпечує автоматизований збір показників водоспоживання, їх надійну передачу до інформаційної системи та мінімальне енергоспоживання вузла, що є необхідною умовою для багаторічної автономної роботи від батарейного живлення.

Робота IoT системи обліку водопостачання базується на автоматизованому зборі даних з вузлів обліку, їх передачі через бездротову мережу до серверної частини, подальшому зберіганні у базі даних та наданні доступу до інформації користувачам через веб- та мобільні застосунки.

Після введення вузла обліку в експлуатацію та його підключення до Thread-мережі пристрій переходить у штатний режим роботи. Під час споживання води літрова стрілка механічного водоміра здійснює обертання, а встановлений магніт проходить повз сенсор Холла DRV5032FBLPGM. Кожен повний оберт стрілки відповідає одному літру води та формує один цифровий імпульс.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		35

Мікроконтролер ESP32-C6-Zero реєструє отримані імпульси та накопичує значення спожитого об'єму води у внутрішній енергонезалежній пам'яті. Одночасно вузол контролює службові параметри, такі як рівень заряду батареї, якість радіозв'язку та стан мережевого підключення.

Через задані проміжки часу, наприклад кожні 3 або 6 годин, вузол активує радіомодуль та формує пакет телеметричних даних. До складу пакета входять ідентифікатор пристрою, поточне значення лічильника, рівень заряду батареї, часові мітки та службова інформація. Після формування пакет передається через Thread mesh-мережу до центрального шлюзу.

Передача даних здійснюється за принципом багатострибкової маршрутизації. Якщо шлюз знаходиться в зоні прямої видимості вузла, повідомлення передається безпосередньо. В іншому випадку пакет може бути переданий через один або декілька проміжних вузлів мережі. Завдяки компактному розташуванню водомірів у багатоквартирному будинку формується щільна мережа, яка забезпечує альтернативні маршрути доставки повідомлень.

Після отримання даних центральний шлюз виконує їх перевірку та перетворення до формату MQTT-повідомлень. Для взаємодії із серверною частиною використовується модель Publish/Subscribe. Дані кожного вузла публікуються у відповідному MQTT-топіку, що дозволяє легко масштабувати систему та додавати нові пристрої без зміни архітектури програмного забезпечення.

MQTT-брокер приймає повідомлення від шлюзу та передає їх сервісу обробки даних. На цьому етапі виконується перевірка коректності отриманої інформації, формування часових міток, обробка службових повідомлень та виявлення можливих аварійних ситуацій.

Після обробки дані зберігаються в базі даних MongoDB. Використання документно-орієнтованої бази даних дозволяє ефективно працювати з телеметричною інформацією та зберігати показники водоспоживання у вигляді

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		36

структурованих документів. Для кожного вузла зберігаються поточні показники, історія споживання, службові параметри та журнали подій.

На основі накопичених даних серверна частина формує статистичну інформацію та виконує аналіз споживання води. Система може розраховувати добове, місячне та річне споживання, виявляти аномалії, контролювати рівень заряду батарей та формувати повідомлення про можливі несправності обладнання.

Доступ до інформації здійснюється через веб-інтерфейс або мобільний застосунок. Користувач може переглядати поточні показники лічильників, історію споживання, статистичні звіти та інформацію про стан системи. Адміністратор додатково отримує можливість контролювати мережеву інфраструктуру, виконувати налаштування обладнання та аналізувати журнали подій.

У разі виявлення аварійної ситуації вузол обліку може виконати позачергову спробу передачі повідомлення через Thread-мережу. Якщо доступний маршрут до шлюзу існує, повідомлення буде доставлено негайно. Якщо ж мережевий маршрут тимчасово недоступний, інформація про подію зберігається у локальній пам'яті та передається під час найближчого успішного сеансу зв'язку.

Таким чином, запропонований алгоритм забезпечує автоматизований збір, передачу, обробку та зберігання даних про водоспоживання, а також надає користувачам зручні засоби моніторингу та аналізу роботи системи в режимі реального часу.

На рисунку 2.9 наведено діаграму потоків даних (DFD) IoT системи обліку водопостачання. Діаграма відображає основні компоненти системи та напрямки передавання інформації від моменту реєстрації споживання води до відображення результатів користувачу.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		37

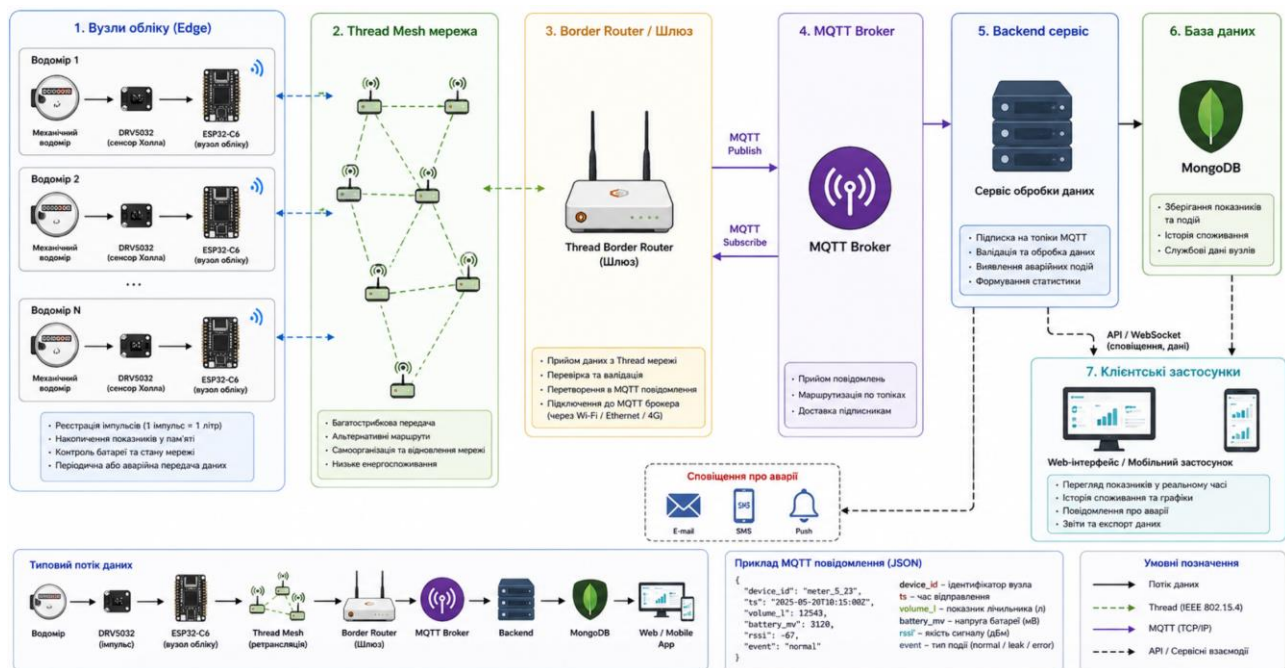


Рисунок 2.9 – DFD потік даних системи обліку водопостачання

Первинним джерелом даних є вузли обліку водоспоживання, встановлені на водомірах. Кожен вузол складається з механічного водоміра, сенсора Холла DRV5032FBLPGM та модуля ESP32-C6-Zero. Під час проходження води через водомір літрова стрілка здійснює обертання, а магніт, встановлений на ній, формує імпульси, які реєструються сенсором Холла. Отримані імпульси обробляються мікроконтролером, який накопичує показники споживання води та формує телеметричні повідомлення.

Передача даних від вузлів обліку здійснюється через бездротову Thread mesh-мережу, побудовану на базі стандарту IEEE 802.15.4. У мережі використовується багатострибова маршрутизація, що дозволяє доставляти повідомлення до шлюзу навіть у випадках відсутності прямого радіозв'язку між вузлом обліку та центральним маршрутизатором. Завдяки компактному розташуванню водомірів у багатоквартирному будинку формується щільна мережа з декількома альтернативними маршрутами передачі даних.

Центральний шлюз виконує функції Thread Border Router та забезпечує взаємодію між мережею IEEE 802.15.4 та локальною IP-мережею будинку. Шлюз

приймає повідомлення від вузлів обліку, перевіряє їх коректність та виконує перетворення отриманих даних до формату MQTT.

Подальша взаємодія між шлюзом та серверною частиною системи здійснюється за допомогою протоколу MQTT. Дані публікуються у відповідних MQTT-топіках та передаються через MQTT-брокер до сервісу обробки даних. Такий підхід дозволяє реалізувати слабозв'язану архітектуру системи, спростити масштабування та забезпечити підтримку великої кількості підключених пристроїв.

Сервіс обробки даних виконує аналіз отриманої інформації, перевірку коректності показників, обробку службових повідомлень та виявлення аварійних ситуацій. Після обробки дані зберігаються у базі даних MongoDB, де формується історія показників водоспоживання, журнали подій та інформація про стан вузлів обліку.

Для доступу до накопичених даних використовуються клієнтські застосунки у вигляді веб-інтерфейсу та мобільного додатка. Через API серверної частини користувачі можуть переглядати поточні показники лічильників, історію споживання води, статистичні звіти та повідомлення про аварійні ситуації. За необхідності система може формувати та надсилати сповіщення за допомогою електронної пошти, SMS-повідомлень або push-сповіщень.

Таким чином, наведена діаграма потоків даних демонструє повний цикл функціонування системи — від реєстрації кожного літра спожитої води до зберігання, аналізу та відображення інформації кінцевому користувачу.

2.6 Проектування структури бази даних MongoDB.

Одним із ключових компонентів IoT системи обліку водопостачання є база даних, яка забезпечує довготривале зберігання показників водоспоживання, службової інформації та журналів подій. Для реалізації серверної частини системи обрано документно-орієнтовану систему керування базами даних MongoDB.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		39

Вибір MongoDB обумовлений необхідністю ефективної роботи з телеметричними даними, що надходять від великої кількості вузлів обліку. На відміну від реляційних баз даних, MongoDB дозволяє зберігати інформацію у вигляді JSON-подібних документів, що добре відповідає структурі MQTT-повідомлень та спрощує масштабування системи при збільшенні кількості підключених пристроїв.

Для зберігання даних пропонується використовувати декілька основних колекцій.

Колекція devices призначена для зберігання інформації про встановлені вузли обліку. Для кожного пристрою зберігається унікальний ідентифікатор, серійний номер, версія програмного забезпечення, дата введення в експлуатацію, поточний стан пристрою, рівень заряду батареї та параметри підключення до мережі.

Приклад документа колекції devices:

```
{
  "_id": "ESP32C6_001",
  "meter_id": "WATER_001",
  "building": "Building_1",
  "apartment": "12",
  "firmware_version": "1.0.0",
  "battery_level": 92,
  "status": "online",
  "last_seen": "2025-05-10T12:00:00Z"
}
```

Колекція meters містить інформацію про водоміри та місця їх встановлення. Для кожного водоміра зберігаються його тип, заводський номер, дата перевірки, коефіцієнт перерахунку імпульсів та прив'язка до відповідного вузла обліку.

Приклад документа колекції meters:

```
{
  "_id": "WATER_001",
  "serial_number": "123456789",
  "type": "DN15",

```

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		40

```

    "pulse_factor": 1,
    "verification_date": "2025-01-15",
    "device_id": "ESP32C6_001"
  }

```

Основною колекцією системи є readings, яка містить показники водоспоживання. Кожен запис формується після отримання телеметричних даних від вузла обліку та містить часову мітку, накопичений показник лічильника та службові параметри.

Приклад документа колекції readings:

```

{
  "_id": "6612fae1",
  "device_id": "ESP32C6_001",
  "meter_id": "WATER_001",
  "timestamp": "2025-05-10T12:00:00Z",
  "counter_value": 15432,
  "battery_level": 92,
  "rssi": -68
}

```

Для зберігання інформації про аварійні ситуації та службові події використовується колекція events. У ній можуть реєструватися повідомлення про можливий витік води, низький заряд батареї, втрату зв'язку з вузлом обліку, перезавантаження пристрою або інші події, що потребують уваги адміністратора.

Приклад документа колекції events:

```

{
  "_id": "event_001",
  "device_id": "ESP32C6_001",
  "timestamp": "2025-05-10T03:15:00Z",
  "event_type": "leak_detected",
  "severity": "warning",
  "description": "Тривале безперервне споживання води"
}

```

Для забезпечення доступу до системи передбачена колекція users, яка містить інформацію про зареєстрованих користувачів, їх ролі та права доступу.

Приклад документа колекції users:

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підп.	Дата		

```
{
  "_id": "user_001",
  "username": "admin",
  "role": "administrator",
  "email": "admin@example.com"
}
```

Запропонована структура бази даних забезпечує розділення телеметричних даних, конфігурації пристроїв та журналів подій, що спрощує обробку інформації та підвищує продуктивність системи. Використання окремих колекцій також дозволяє реалізувати гнучку систему аналітики, формування звітів та моніторингу технічного стану обладнання.

Таким чином, використання MongoDB дозволяє створити масштабовану серверну інфраструктуру для IoT системи обліку водопостачання, здатну ефективно працювати з великими обсягами телеметричних даних та забезпечувати швидкий доступ до інформації для користувачів і адміністраторів системи.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підп.	Дата		

3. РОЗРОБЛЕННЯ ІОТ СИСТЕМИ ОБЛІКУ ВОДОПОСТАЧАННЯ

3.1 Аналіз середовища проєктування ESP-IDF

Для розробки програмного забезпечення вузла обліку водоспоживання було обрано програмну платформу ESP-IDF (Espressif IoT Development Framework), яка є офіційним середовищем розробки компанії Espressif Systems для сімейства мікроконтролерів ESP32.

ESP-IDF являє собою набір програмних бібліотек, драйверів та інструментів, що дозволяють створювати вбудовані застосунки для пристроїв Інтернету речей. Платформа підтримує програмування мовами C та C++, використовує операційну систему реального часу FreeRTOS та забезпечує доступ до всіх апаратних можливостей мікроконтролерів серії ESP32. Зовнішній вигляд середовища проєктування ESP-IDF наведено на рисунку 3.1.

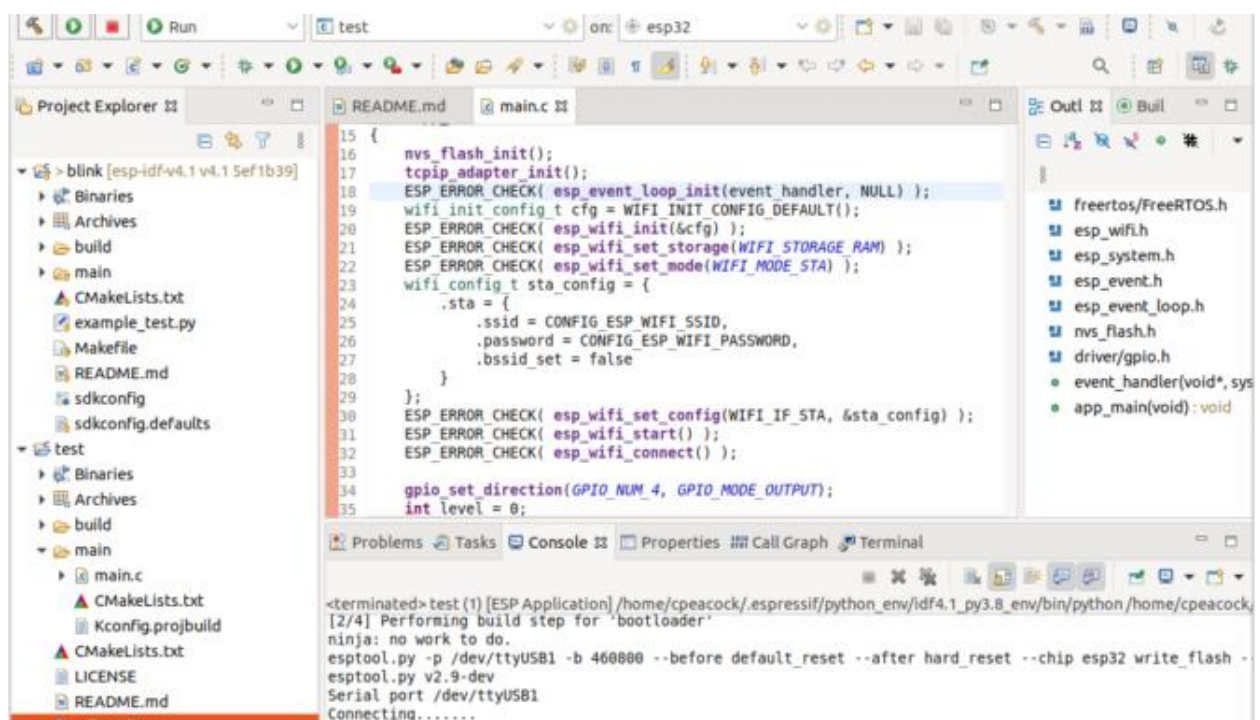


Рисунок 3.1 - Зовнішній вигляд середовища проєктування ESP-IDF з інтеграцією в Eclipse.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		43

Вибір ESP-IDF для реалізації даного проєкту обумовлений тим, що мікроконтролер ESP32-C6 є відносно новим пристроєм і повна підтримка його функціональних можливостей реалізована саме в офіційному SDK виробника. На відміну від альтернативних середовищ розробки, таких як Arduino Framework або PlatformIO, ESP-IDF надає прямий доступ до низькорівневих механізмів керування живленням, мережевими інтерфейсами та стеком протоколів IEEE 802.15.4.

Однією з ключових переваг ESP32-C6 є підтримка стандарту IEEE 802.15.4, на базі якого реалізуються протоколи Thread та Zigbee. Для роботи з даними протоколами в ESP-IDF використовується інтегрований стек OpenThread, який забезпечує побудову mesh-мереж, маршрутизацію повідомлень, автоматичне формування топології мережі та взаємодію з Thread Border Router.

Крім підтримки Thread, ESP-IDF забезпечує роботу з Bluetooth Low Energy, який використовується у проєкті для початкового налаштування вузла обліку. Засоби BLE дозволяють реалізувати режим Commissioning Mode, у якому користувач може виконати первинне налаштування пристрою за допомогою мобільного застосунку, після чого Bluetooth вимикається для зменшення енергоспоживання.

Важливою особливістю ESP-IDF є розвинена система керування енергоспоживанням. Платформа підтримує режими Light Sleep та Deep Sleep, механізми пробудження від зовнішніх переривань, таймерів RTC та GPIO-сигналів. Саме ці можливості дозволяють реалізувати багаторічну автономну роботу вузла обліку від батарейного живлення.

Для зберігання конфігураційних параметрів у ESP-IDF використовується підсистема NVS (Non-Volatile Storage), яка дозволяє зберігати налаштування Thread-мережі, параметри MQTT-з'єднання, ідентифікатори пристрою та інші експлуатаційні дані у внутрішній Flash-пам'яті мікроконтролера.

Архітектура ESP-IDF базується на компонентному підході. Програмне забезпечення розділяється на окремі модулі, що спрощує розробку, тестування

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		44

та подальшу модернізацію системи. У рамках даного проєкту можуть бути виділені окремі компоненти для роботи із сенсором Холла, підрахунку імпульсів водоміра, управління енергоспоживанням, взаємодії з Thread-мережею, передачі даних через MQTT та реалізації режиму початкового налаштування через Bluetooth Low Energy.

Для компіляції проєкту ESP-IDF використовує систему збірки CMake та інструментарій Ninja. Налагодження програмного забезпечення може виконуватися через послідовний інтерфейс UART або за допомогою JTAG-інтерфейсу. Додатково платформа надає засоби моніторингу журналів подій, профілювання продуктивності та аналізу енергоспоживання пристрою.

Таким чином, середовище ESP-IDF забезпечує повний набір програмних засобів, необхідних для реалізації вузла обліку водоспоживання на базі ESP32-C6. Використання офіційного SDK дозволяє отримати доступ до всіх можливостей мікроконтролера, реалізувати підтримку Thread mesh-мережі, Bluetooth Low Energy та сучасних механізмів енергозбереження, що є необхідною умовою для створення автономної IoT системи обліку водопостачання.

3.2 Розроблення програмного забезпечення вузла обліку на базі ESP32-C6

Програмне забезпечення вузла обліку водоспоживання розробляється у середовищі ESP-IDF та має модульну структуру. Основними функціональними частинами програми є ініціалізація апаратних ресурсів, зчитування імпульсів від сенсора Холла, збереження показників у енергонезалежній пам'яті, робота з режимами енергозбереження, підключення до Thread-мережі, передача даних до шлюзу та режим первинного налаштування через Bluetooth Low Energy.

На рисунку 3.2 наведено UML-діаграма компонентів програмного забезпечення вузла обліку на базі ESP32-C6. Діаграма відображає логічну структуру прошивки, її взаємодію з апаратними елементами вузла, системними службами ESP-IDF та зовнішніми компонентами IoT системи.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		45

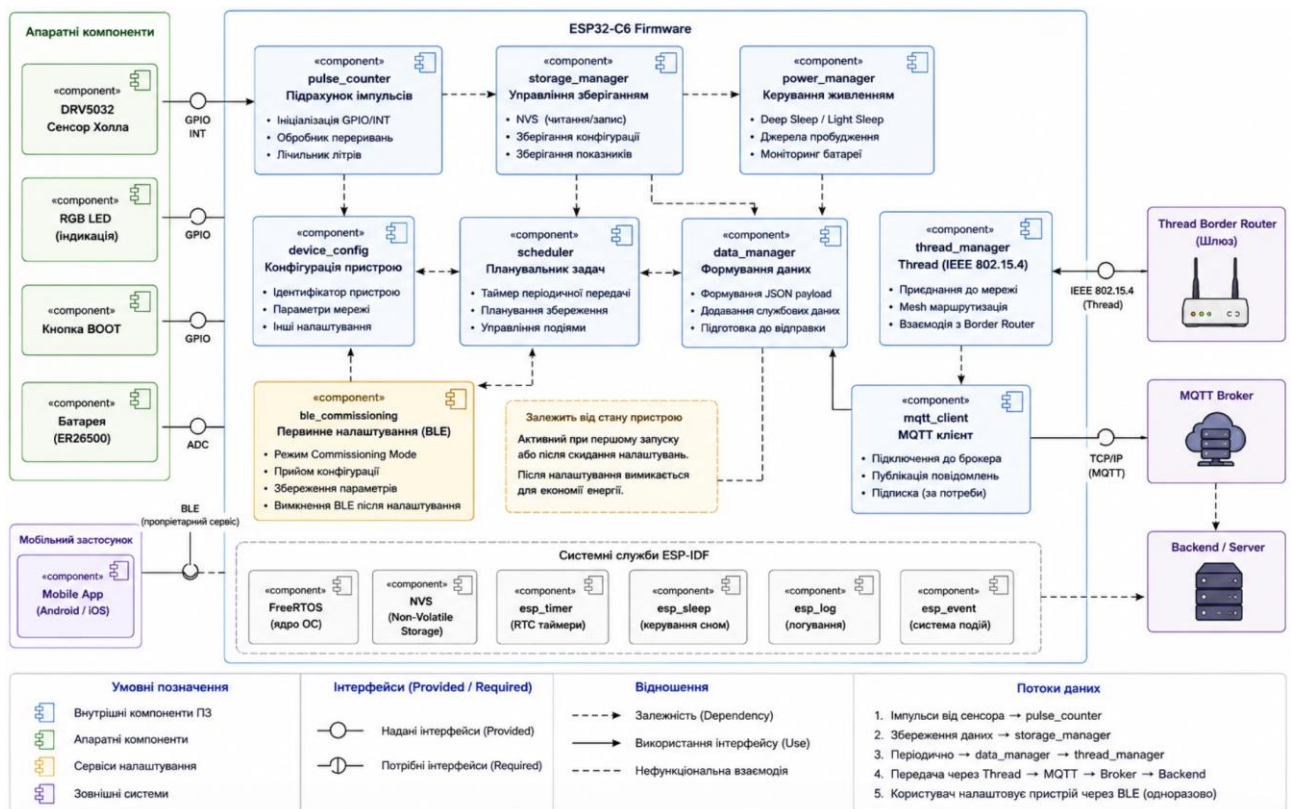


Рисунок 3.2 - UML-діаграма компонентів програмного забезпечення вузла обліку на базі ESP32-C6

Програмне забезпечення вузла поділено на окремі функціональні компоненти. Компонент pulse_counter відповідає за ініціалізацію GPIO-входу, обробку переривань від сенсора Холла DRV5032 та підрахунок імпульсів, кожен з яких відповідає одному літру води. Отримані значення передаються до компонента storage_manager, який забезпечує збереження показників і конфігураційних параметрів у NVS-пам'яті.

Компонент power_manager відповідає за керування режимами енергозбереження, зокрема перехід у Deep Sleep, налаштування джерел пробудження та контроль рівня заряду батареї. Компонент scheduler визначає моменти періодичної передачі даних, збереження показників і виконання службових операцій.

Формування телеметричного повідомлення виконує компонент data_manager. Він об'єднує показники лічильника, ідентифікатор пристрою,

рівень заряду батареї та службові параметри у формат JSON. Після цього дані передаються до `thread_manager`, який забезпечує приєднання до Thread-мережі та передачу повідомлень до Thread Border Router.

Окремим компонентом виділено `ble_commissioning`, який використовується лише під час первинного налаштування пристрою. Через Bluetooth Low Energy мобільний застосунок передає параметри конфігурації, після чого вони зберігаються у пам'яті, BLE вимикається, а вузол переходить у штатний режим роботи.

Нижній рівень програмного забезпечення утворюють системні служби ESP-IDF: FreeRTOS, NVS, RTC-таймери, механізми керування сном, система логування та обробки подій. Саме ці засоби забезпечують взаємодію між окремими компонентами прошивки та апаратними ресурсами ESP32-C6.

Таким чином, запропонована компонентна структура дозволяє розділити програмне забезпечення вузла на незалежні модулі, що спрощує розробку, тестування та подальшу модернізацію прошивки. Після визначення основних компонентів системи можна перейти до опису їх програмної реалізації та розгляду фрагментів коду.

На початку роботи пристрій виконує ініціалізацію NVS-пам'яті, GPIO, таймерів та перевіряє наявність збереженої конфігурації. Якщо конфігурація відсутня, вузол переходить у режим первинного налаштування.

```
void app_main(void)
{
    nvs_flash_init();
    gpio_init();
    pulse_counter_init();

    if (!config_is_saved()) {
        start_commissioning_mode();
    } else {
        thread_network_start();
        normal_mode_start();
    }
}
```

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		47

У наведеному фрагменті функція `config_is_saved()` перевіряє, чи були раніше збережені параметри пристрою. Якщо пристрій запускається вперше, активується режим `Commissioning Mode`, у якому користувач може виконати налаштування через `Bluetooth Low Energy`.

Для підрахунку імпульсів від сенсора Холла `DRV5032FBLPGM` використовується `GPIO`-вхід з перериванням. Оскільки один імпульс відповідає одному літру води, при кожному спрацюванні сенсора значення внутрішнього лічильника збільшується на одиницю.

```
static volatile uint32_t water_counter_liters = 0;

static void IRAM_ATTR hall_sensor_isr_handler(void *arg)
{
    water_counter_liters++;
}
```

Даний обробник переривання виконується при зміні стану сигналу від сенсора Холла. Використання переривань дозволяє не опитувати вхід постійно, що зменшує енергоспоживання мікроконтролера.

Періодично накопичені показники зберігаються у `NVS`-пам'яті. Це необхідно для збереження даних у разі перезавантаження пристрою або тимчасового зникнення живлення.

```
void save_meter_value(uint32_t value)
{
    nvs_handle_t handle;
    nvs_open("meter", NVS_READWRITE, &handle);
    nvs_set_u32(handle, "liters", value);
    nvs_commit(handle);
    nvs_close(handle);
}
```

Передача даних виконується не після кожного імпульсу, а з певною періодичністю. Це дозволяє суттєво зменшити кількість радіообмінів і продовжити термін автономної роботи вузла.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		48

```

void send_meter_data(void)
{
    char payload[128];

    snprintf(payload, sizeof(payload),
             "{\"device_id\":\"meter_001\", \"volume_1\":%lu, \"battery_mv\":%u\", \"water_counter_liters\", \"read_battery_voltage()\"};

    thread_send_message(payload);
}

```

У цьому фрагменті формується JSON-повідомлення, яке містить ідентифікатор пристрою, поточний показник лічильника та напругу батареї. Далі повідомлення передається через Thread-мережу до шлюзу, який перетворює його у MQTT-повідомлення та передає до серверної частини системи.

Для зменшення енергоспоживання після завершення обробки імпульсу або передачі даних пристрій переходить у режим Deep Sleep.

```

void enter_deep_sleep(uint64_t sleep_time_us)
{
    esp_sleep_enable_timer_wakeup(sleep_time_us);
    esp_sleep_enable_ext0_wakeup(HALL_SENSOR_GPIO, 1);
    esp_deep_sleep_start();
}

```

У даному випадку передбачено два джерела пробудження: таймер RTC для періодичної передачі даних та зовнішній GPIO-сигнал від сенсора Холла. Такий підхід дозволяє пристрою більшу частину часу перебувати в режимі мінімального енергоспоживання.

Окремо реалізується режим первинного налаштування. Він активується при першому запуску пристрою або при натисканні кнопки BOOT. У цьому режимі вмикається Bluetooth Low Energy, через який мобільний застосунок передає параметри конфігурації.

```

void start_commissioning_mode(void)
{
    rgb_led_set_color(0, 0, 255); // синій колір – режим налаштування
}

```

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		49

```

    ble_start();

    while (!config_received()) {
        vTaskDelay(pdMS_TO_TICKS(500));
    }

    save_device_config();
    ble_stop();
    rgb_led_off();

    thread_network_start();
}

```

Після отримання конфігурації параметри зберігаються у Flash-пам'яті, Bluetooth Low Energy вимикається, а пристрій підключається до Thread-мережі та переходить у штатний режим роботи. Вимикання BLE після налаштування дозволяє зменшити енергоспоживання та підвищити безпеку пристрою.

Таким чином, програмне забезпечення вузла обліку реалізує повний цикл роботи пристрою: первинне налаштування, підрахунок імпульсів водоміра, збереження показників, періодичну передачу даних через Thread-мережу та перехід у режим енергозбереження. Така структура забезпечує надійну роботу вузла та дозволяє досягти тривалої автономної експлуатації від батарейного живлення.

3.3 Розрахунок енергоспоживання вузла обліку.

Однією з основних вимог до вузла обліку водоспоживання є забезпечення тривалої автономної роботи без заміни елемента живлення. Оскільки міжповірочний інтервал побутових водомірів зазвичай становить близько чотирьох років, доцільно проєктувати вузол із розрахунковим терміном автономної роботи не менше п'яти років.

У складі вузла обліку використовуються модуль ESP32-C6-Zero, сенсор Холла DRV5032FBLPGM та літієва батарея типу ER26500. Вибір ER26500 обумовлений її більшою ємністю порівняно з ER18505 за близької вартості, що

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підп.	Дата		

дозволяє отримати більший запас автономності. Для розрахунку приймемо ємність батареї ER26500 рівною 8500 мА·год.

У режимі очікування мікроконтролер ESP32-C6 перебуває в режимі Deep Sleep. Для оцінкового розрахунку приймемо струм споживання мікроконтролера в режимі сну 10 мкА. Сенсор Холла DRV5032FBLPGM працює постійно та споживає близько 0,54 мкА. Тоді сумарний струм у режимі очікування становить:

$$I_{\text{оч}} = 10 + 0,54 = 10,54 \text{ мкА.}$$

Добове споживання заряду в режимі очікування:

$$Q_{\text{оч}} = 10,54 \times 24 = 252,96 \text{ мкА} \cdot \text{год} = 0,253 \text{ мА} \cdot \text{год/добу.}$$

За п'ять років:

$$Q_{\text{оч}_5\text{р}} = 0,253 \times 365 \times 5 = 461,7 \text{ мА} \cdot \text{год.}$$

Розглянемо енергоспоживання під час передачі даних. Приймемо, що один сеанс зв'язку триває 3 секунди, а середній струм під час активної роботи радіомодуля та мікроконтролера становить 80 мА. Якщо вузол передає дані один раз на годину, за добу виконується 24 сеанси зв'язку.

Добове споживання на передачу власних даних:

$$Q_{\text{tx}} = (80 \times 3 \times 24) / 3600 = 1,6 \text{ мА} \cdot \text{год/добу.}$$

Оскільки вузол працює у mesh-мережі, необхідно врахувати додаткові витрати енергії на ретрансляцію повідомлень від інших вузлів. Точне значення залежить від розташування пристрою, якості радіоканалу, кількості сусідніх

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		51

вузлів та побудованих маршрутів. Для інженерного розрахунку введемо коефіцієнт mesh-навантаження K_{mesh} , який враховує додаткові витрати на ретрансляцію. Для консервативної оцінки приймемо $K_{\text{mesh}} = 2$.

Тоді добове споживання на передачу та ретрансляцію:

$$Q_{\text{tx_mesh}} = Q_{\text{tx}} \times K_{\text{mesh}} = 1,6 \times 2 = 3,2 \text{ мА} \cdot \text{год/добу}.$$

За п'ять років:

$$Q_{\text{tx_mesh_5p}} = 3,2 \times 365 \times 5 = 5840 \text{ мА} \cdot \text{год}.$$

Загальне споживання за п'ять років:

$$Q_{\text{заг}} = Q_{\text{оч_5p}} + Q_{\text{tx_mesh_5p}} = 461,7 + 5840 = 6301,7 \text{ мА} \cdot \text{год}.$$

Порівняємо отримане значення з ємністю батареї ER26500:

$$K_{\text{зап}} = 8500 / 6301,7 = 1,35.$$

Отже, при використанні батареї ER26500 та передачі даних один раз на годину вузол може забезпечити автономну роботу протягом п'яти років навіть з урахуванням коефіцієнта mesh-навантаження $K_{\text{mesh}} = 2$. Запас автономності становить приблизно 35 %.

Водночас слід зазначити, що режим передачі даних один раз на годину є досить інтенсивним для системи обліку водопостачання. Оскільки показники водоспоживання не потребують оновлення в режимі реального часу, доцільно розглянути збільшення інтервалу передачі до 3, 6 або 12 годин. Наприклад, при передачі даних один раз на 6 годин кількість сеансів зв'язку зменшується у 6 разів:

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підп.	Дата		

$$Q_{\text{tx_mesh_6h}} = 3,2 / 6 = 0,533 \text{ мА} \cdot \text{год/добу}.$$

За п'ять років:

$$Q_{\text{tx_mesh_6h_5p}} = 0,533 \times 365 \times 5 = 972,7 \text{ мА} \cdot \text{год}.$$

Загальне споживання за п'ять років:

$$Q_{\text{заг_6h}} = 972,7 + 461,7 = 1434,4 \text{ мА} \cdot \text{год}.$$

У цьому випадку коефіцієнт запасу становить:

$$K_{\text{зап_6h}} = 8500 / 1434,4 = 5,93.$$

Таким чином, використання батареї ER26500 забезпечує значний запас автономності вузла обліку. Навіть за умови щогодинної передачі даних і врахування додаткових витрат на ретрансляцію в mesh-мережі розрахунковий термін роботи перевищує п'ять років. Оптимальним режимом для практичного використання є передача даних один раз на 3–6 годин, що дозволяє зменшити енергоспоживання, зберегти достатню оперативність отримання показників і забезпечити великий запас роботи батареї.

Окремо слід врахувати, що у повністю батарейній mesh-мережі нехайна доставка аварійних повідомлень не завжди може бути гарантована, оскільки частина вузлів може перебувати в режимі сну. Тому для аварійних ситуацій доцільно передбачити режим позачергової спроби передачі. У разі виявлення тривалого безперервного споживання або різкого збільшення витрати вузол активує радіомодуль і намагається передати аварійне повідомлення. Якщо поблизу є активний мережевий вузол або доступний маршрут до шлюзу, повідомлення буде доставлено оперативно. Якщо маршрут недоступний, подія

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		53

зберігається у пам'яті та передається під час найближчого планового сеансу зв'язку.

Для підвищення надійності mesh-мережі без суттєвого збільшення вартості системи доцільно передбачити можливість встановлення додаткових постійно живлених мережевих вузлів не на кожному поверсі, а в ключових точках будівлі, наприклад один вузол на 4–5 поверхів. Це дозволяє зменшити навантаження на батарейні пристрої, підвищити ймовірність оперативної доставки аварійних повідомлень і водночас зберегти простоту та економічну доцільність системи.

3.4 Налаштування MQTT-брокера Mosquitto.

Для організації обміну повідомленнями між вузлами обліку водоспоживання та серверною частиною системи використовується MQTT-брокер Mosquitto. Дане програмне забезпечення є одним із найпоширеніших відкритих рішень для реалізації протоколу MQTT та широко застосовується в системах Інтернету речей завдяки простоті налаштування, невисоким вимогам до ресурсів та високій надійності роботи.

У розроблюваній системі Mosquitto виконує функцію центрального вузла обміну повідомленнями між шлюзом Thread Border Router, сервером обробки даних та клієнтськими застосунками. Шлюз публікує телеметричні дані у відповідні MQTT-топіки, а серверна частина підписується на них та виконує подальшу обробку інформації.

Архітектура взаємодії побудована за моделлю Publish/Subscribe. У такій моделі вузли обліку та шлюз не взаємодіють безпосередньо із сервером бази даних, а передають повідомлення брокеру. Це дозволяє зменшити зв'язність компонентів системи та спростити її масштабування.

Для роботи системи достатньо виконати базове налаштування MQTT-брокера. Основний конфігураційний файл Mosquitto має вигляд:

```
listener 1883
```

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
						54
Зм.	Арк.	№ докум.	Підп.	Дата		

```
allow_anonymous false

password_file /etc/mosquitto/passwd

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

autosave_interval 1800
```

У наведеній конфігурації відкривається стандартний MQTT-порт 1883, забороняється анонімний доступ та вмикається автентифікація користувачів за допомогою файлу паролів. Додатково активується механізм збереження повідомлень та журналювання роботи брокера.

Створення користувача для підключення шлюзу та серверної частини може виконуватися командою:

```
mosquitto_passwd -c /etc/mosquitto/passwd water_system
```

Після виконання команди адміністратор задає пароль, який надалі використовується клієнтами MQTT для авторизації.

Для передачі телеметричних даних у системі пропонується використовувати ієрархічну структуру топіків. Приклад структури MQTT-топіків наведено нижче:

```
water/building1/apartment12/meter1/data
water/building1/apartment12/meter1/event
water/building1/apartment12/meter1/status
```

Топік data використовується для передачі показників водоспоживання, топік event — для аварійних повідомлень та службових подій, а топік status — для передачі інформації про стан вузла, рівень заряду батареї та параметри зв'язку.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		55

Приклад телеметричного повідомлення, що надходить від вузла обліку:

```
{
  "device_id": "meter_001",
  "counter_value": 15432,
  "battery_mv": 3560,
  "rssi": -68,
  "timestamp": "2025-05-10T12:00:00Z"
}
```

Для підвищення надійності доставки повідомлень доцільно використовувати другий рівень якості обслуговування MQTT (QoS 1), який гарантує доставку повідомлення щонайменше один раз.

Приклад публікації повідомлення:

```
esp_mqtt_client_publish(
    client,
    "water/building1/apartment12/meter1/data",
    payload,
    0,
    1,
    0
);
```

Параметр QoS встановлено рівним 1, що забезпечує підтвердження отримання повідомлення брокером.

Для підвищення безпеки системи MQTT-брокер може працювати через захищене TLS-з'єднання. У такому випадку використовується окремий порт 8883 та цифрові сертифікати.

Приклад налаштування TLS:

```
listener 8883

cafile /etc/mosquitto/ca.crt
certfile /etc/mosquitto/server.crt
keyfile /etc/mosquitto/server.key

allow_anonymous false
```

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		56

Використання TLS забезпечує шифрування трафіку між шлюзом та серверною частиною системи, що запобігає перехопленню або модифікації переданих даних.

Таким чином, MQTT-брокер Mosquitto є центральним елементом серверної інфраструктури розроблюваної системи. Він забезпечує надійну маршрутизацію повідомлень між вузлами обліку, шлюзом та серверними сервісами, дозволяє реалізувати масштабовану архітектуру та створює основу для подальшого зберігання й аналізу телеметричних даних у базі MongoDB.

3.6 Розроблення серверного застосунку обробки даних.

Серверний застосунок обробки даних є проміжною ланкою між MQTT-брокером Mosquitto, базою даних MongoDB та клієнтськими застосунками. Його основними завданнями є підключення до MQTT-брoкeра, отримання телеметричних повідомлень від вузлів обліку, перевірка коректності даних, збереження показників у базі MongoDB та формування подій у разі виявлення аварійних ситуацій.

Для реалізації серверного застосунку доцільно використати платформу Node.js, оскільки вона добре підходить для обробки подієво-орієнтованих потоків даних та має готові бібліотеки для роботи з MQTT і MongoDB. Для підключення до MQTT-брoкeра використовується бібліотека `mqtt`, а для взаємодії з MongoDB — бібліотека `mongoose`.

Підключення до MQTT-брoкeра Mosquitto реалізоване таким чином:

```
const mqtt = require('mqtt');

const mqttClient = mqtt.connect('mqtt://localhost:1883', {
  username: 'water_system',
  password: 'password',
  clientId: 'backend_service'
});

mqttClient.on('connect', () => {
  console.log('Connected to MQTT broker');
```

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		57

```

mqttClient.subscribe('water/+/+/+/data', { qos: 1 });
mqttClient.subscribe('water/+/+/+/event', { qos: 1 });
mqttClient.subscribe('water/+/+/+/status', { qos: 1 });
});

```

У цьому фрагменті серверний застосунок підключається до брокера Mosquitto та підписується на групу MQTT-топиків. Символ + використовується як шаблон одного рівня топика, що дозволяє приймати повідомлення від усіх будинків, квартир і лічильників без необхідності окремо вказувати кожен пристрій.

Для збереження даних у MongoDB створюється модель показників водоспоживання:

```

const mongoose = require('mongoose');

const readingSchema = new mongoose.Schema({
  device_id: String,
  meter_id: String,
  counter_value: Number,
  battery_mv: Number,
  rssi: Number,
  timestamp: Date,
  created_at: {
    type: Date,
    default: Date.now
  }
});

const Reading = mongoose.model('Reading', readingSchema);

```

Дана модель описує структуру документа, який зберігається у колекції readings. У ньому фіксується ідентифікатор пристрою, ідентифікатор вodomіра, поточне значення лічильника, напруга батареї, якість зв'язку та час формування повідомлення.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		58

Обробка вхідних MQTT-повідомлень виконується у відповідному обробнику подій:

```
mqttClient.on('message', async (topic, message) => {
  try {
    const payload = JSON.parse(message.toString());

    if (!payload.device_id || payload.counter_value ===
undefined) {
      console.log('Invalid message format');
      return;
    }

    const reading = new Reading({
      device_id: payload.device_id,
      meter_id: payload.meter_id,
      counter_value: payload.counter_value,
      battery_mv: payload.battery_mv,
      rssi: payload.rssi,
      timestamp: payload.timestamp ? new
Date(payload.timestamp) : new Date()
    });

    await reading.save();

    console.log(`Saved reading from
${payload.device_id}`);
  } catch (error) {
    console.error('MQTT message processing error:',
error.message);
  }
});
```

У наведеному фрагменті сервер отримує повідомлення, перетворює його з JSON-формату, перевіряє наявність основних полів і записує дані до MongoDB. Якщо повідомлення має неправильний формат, воно не зберігається, що дозволяє уникнути потрапляння некоректних даних до бази.

Окремо в серверному застосунку реалізується перевірка аварійних ситуацій. Наприклад, якщо вузол передає подію про тривале безперервне споживання води або низький заряд батареї, сервер створює запис у колекції events.

```
const eventSchema = new mongoose.Schema({
  device_id: String,
  event_type: String,
```

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		59

```

    severity: String,
    description: String,
    timestamp: Date,
    created_at: {
      type: Date,
      default: Date.now
    }
  });

const Event = mongoose.model('Event', eventSchema);

async function createEvent(deviceId, type, severity,
description) {
  const event = new Event({
    device_id: deviceId,
    event_type: type,
    severity: severity,
    description: description,
    timestamp: new Date()
  });

  await event.save();
}

```

Функція `createEvent()` використовується для реєстрації службових та аварійних повідомлень. Такі події можуть надалі відображатися у веб-інтерфейсі або використовуватися для формування push-, SMS- чи email-сповіщень.

Приклад перевірки рівня заряду батареї:

```

if (payload.battery_mv && payload.battery_mv < 3000) {
  await createEvent(
    payload.device_id,
    'low_battery',
    'warning',
    'Низький рівень заряду батареї вузла обліку'
  );
}

```

Після збереження показників серверний застосунок може оновлювати поточний стан пристрою у колекції `devices`. Це дозволяє швидко отримувати інформацію про останній зв'язок із вузлом, його стан батареї та актуальні показники.

```

await Device.updateOne(
  { device_id: payload.device_id },
  {

```

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підп.	Дата		

```

    $set: {
      last_seen: new Date(),
      battery_mv: payload.battery_mv,
      last_counter_value: payload.counter_value,
      status: 'online'
    }
  },
  { upsert: true }
);

```

Таким чином, серверний застосунок виконує роль центрального програмного компонента, який приймає дані від MQTT-брокера, перевіряє їх, зберігає у MongoDB та формує службові або аварійні події. Такий підхід дозволяє відокремити рівень збору телеметрії від рівня зберігання та відображення даних, що спрощує масштабування системи та подальший розвиток її функціональних можливостей.

3.7 Розроблення клієнтського програмного забезпечення.

Завершальним елементом IoT системи обліку водопостачання є клієнтське програмне забезпечення, призначене для взаємодії користувачів із серверною частиною системи. Основним завданням клієнтського застосунку є відображення поточних показників водоспоживання, статистичної інформації, стану вузлів обліку та повідомлень про аварійні ситуації.

Для реалізації клієнтської частини доцільно використовувати веборієнтовану архітектуру. Такий підхід дозволяє забезпечити доступ до системи з персональних комп'ютерів, планшетів та смартфонів без необхідності встановлення спеціалізованого програмного забезпечення. Як технологічну основу може бути використана бібліотека React, яка забезпечує створення сучасних односторінкових застосунків (SPA) та підтримує ефективне оновлення інтерфейсу при надходженні нових даних.

Клієнтський застосунок взаємодіє із серверною частиною через REST API. Після авторизації користувач отримує доступ до інформації про власні вузли

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
						61
Зм.	Арк.	№ докум.	Підп.	Дата		

обліку, поточні показники лічильників, історію споживання води та статистичні звіти.

Структурно клієнтське програмне забезпечення можна розділити на декілька функціональних модулів:

- модуль авторизації користувачів;
- модуль перегляду поточних показників;
- модуль статистики водоспоживання;
- модуль моніторингу стану пристроїв;
- модуль перегляду аварійних повідомлень;
- модуль адміністрування системи.

Після входу до системи користувач потрапляє на головну інформаційну панель (Dashboard), де відображаються основні параметри системи. Для отримання актуальних даних використовується HTTP-запит до серверного API:

```
async function loadMeters() {  
  const response = await fetch('/api/meters');  
  const data = await response.json();  
  
  return data;  
}
```

Сервер повертає список водомірів, закріплених за поточним користувачем, разом із їхніми показниками та службовою інформацією.

Для відображення поточних показників може використовуватися окремий React-компонент:

```
function MeterCard({ meter }) {  
  return (  
    <div className="meter-card">  
      <h3>{meter.name}</h3>  
      <p>Показник: {meter.counter_value} л</p>  
      <p>Батарея: {meter.battery_level}%</p>  
    </div>  
  );  
}
```

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		62

Даний компонент відображає інформацію про конкретний вузол обліку, включаючи поточний показник лічильника та рівень заряду батареї.

Для формування статистики водоспоживання клієнтський застосунок отримує історичні дані з MongoDB через серверний API:

```
async function loadStatistics(meterId) {
    const response = await fetch(
        `/api/statistics/${meterId}`
    );

    return await response.json();
}
```

На основі отриманих даних користувач може переглядати добове, тижневе та місячне споживання води. Серверна частина виконує агрегацію даних у MongoDB та повертає готову статистичну інформацію для побудови графіків.

Окремий модуль призначений для моніторингу технічного стану вузлів обліку. Він дозволяє контролювати рівень заряду батарей, час останнього зв'язку із сервером, версію програмного забезпечення пристрою та якість радіозв'язку.

Приклад отримання інформації про стан пристрою:

```
async function loadDeviceStatus(deviceId) {
    const response = await fetch(
        `/api/devices/${deviceId}`
    );

    return await response.json();
}
```

Важливою складовою системи є підсистема оповіщення. При виникненні аварійної ситуації серверний застосунок формує запис у колекції events, після

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		63

чого клієнтський інтерфейс отримує відповідне повідомлення та відображає його користувачу.

Для отримання аварійних повідомлень може використовуватися окремий API-запит:

```
async function loadEvents() {  
    const response = await fetch('/api/events');  
  
    return await response.json();  
}
```

До аварійних подій можуть належати:

- тривале безперервне споживання води;
- можливий витік води;
- низький рівень заряду батареї;
- втрата зв'язку з вузлом обліку;
- відсутність телеметричних повідомлень протягом заданого часу;
- перезавантаження пристрою.

Для адміністратора системи передбачаються додаткові функції керування обладнанням. Через адміністративну панель можна переглядати перелік усіх вузлів обліку, контролювати стан мережі, переглядати журнали подій та виконувати налаштування параметрів системи.

З метою підвищення швидкодії клієнтського застосунку доцільно використовувати механізм кешування даних та асинхронне оновлення інформації. Це дозволяє зменшити навантаження на сервер та забезпечити оперативне відображення актуальних показників користувачам.

Таким чином, розроблене клієнтське програмне забезпечення забезпечує зручний доступ до даних про водоспоживання, надає інструменти для аналізу статистики, контролю технічного стану вузлів обліку та оперативного реагування на аварійні ситуації. Використання сучасних вебтехнологій дозволяє створити масштабовану та зручну систему моніторингу, яка може

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		64

використовуватися як окремими споживачами, так і організаціями, що здійснюють облік водних ресурсів.

На рисунку 3.3 представлено головне вікно авторизації користувача вебзастосунку IoT системи обліку водопостачання. Дане вікно є початковою сторінкою системи та призначене для автентифікації користувачів перед отриманням доступу до функціональних можливостей програмного комплексу.

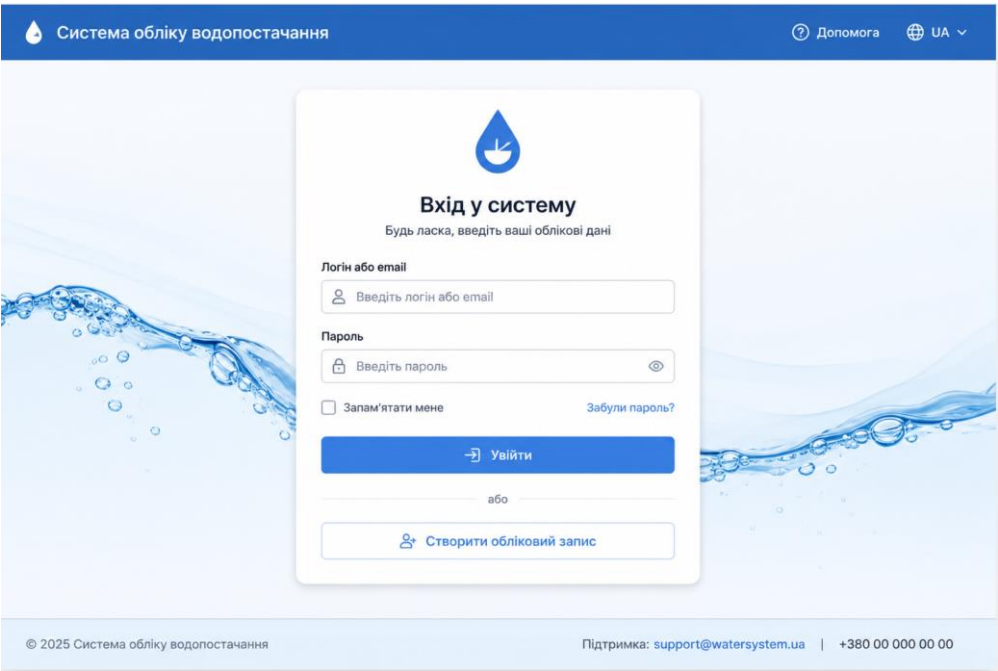


Рисунок 3.3 – Зовнішній вигляд вікна входу в систему

У центральній частині інтерфейсу розташована форма входу, яка містить поля для введення логіна або адреси електронної пошти та пароля користувача. Після введення облікових даних користувач натискає кнопку «Увійти», після чого клієнтський застосунок формує запит до серверного API для перевірки введеної інформації та отримання JWT-токена авторизації.

Для підвищення зручності використання передбачено функцію «Запам'ятати мене», яка дозволяє зберігати токен авторизації у локальному сховищі браузера та автоматично відновлювати сеанс користувача під час наступних відвідувань системи. Також реалізовано посилання «Забули пароль?».

за допомогою якого користувач може ініціювати процедуру відновлення доступу до облікового запису.

Для нових користувачів передбачено кнопку «Створити обліковий запис», яка відкриває форму реєстрації та дозволяє створити новий обліковий запис у системі.

Після успішної авторизації користувач перенаправляється на головну інформаційну панель (Dashboard), де відображаються поточні показники водоспоживання, статистичні дані, стан вузлів обліку та повідомлення про аварійні ситуації.

3.8 Розроблення модуля перегляду поточних показників.

Одним із основних функціональних модулів клієнтського програмного забезпечення є модуль перегляду поточних показників водоспоживання. Його призначенням є надання користувачу актуальної інформації про стан вodomірів, поточні показники споживання води, рівень заряду батареї вузлів обліку та час останнього отримання даних від пристрою.

Після успішної авторизації користувач потрапляє на головну інформаційну панель (Dashboard), яка містить перелік закріплених за ним вузлів обліку. Для кожного вodomіра відображаються основні параметри, необхідні для оперативного контролю стану системи.

Дані про вodomіри отримуються із серверної частини через REST API. Після відкриття сторінки клієнтський застосунок виконує HTTP-запит до серверного застосунку для отримання актуальних показників.

Приклад функції отримання даних:

```
async function loadMeters() {  
  
    const token =  
        localStorage.getItem("token");  
  
    const response = await fetch(  
        "/api/meters",
```

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		66

```

        {
            headers: {
                Authorization:
                    `Bearer ${token}`
            }
        }
    );

    return await response.json();
}

```

Після отримання відповіді сервера дані передаються до компонентів інтерфейсу користувача для подальшого відображення.

Для представлення інформації про окремий вузол обліку використовується компонент картки водоміра.

```

function MeterCard({ meter }) {

    return (
        <div className="meter-card">

            <h3>
                {meter.name}
            </h3>

            <p>
                Показник:
                {meter.counter_value} л
            </p>

            <p>
                Батарея:
                {meter.battery_level} %
            </p>

            <p>
                Останній зв'язок:
                {meter.last_seen}
            </p>

        </div>
    );
}

```

У даному компоненті відображається назва водоміра, поточне значення лічильника, рівень заряду батареї та час останнього успішного зв'язку з сервером.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		67

Для побудови головної інформаційної панелі використовується контейнерний компонент, який виконує завантаження даних та формує список карток водомірів.

```
function Dashboard() {  
  
  const [meters, setMeters] =  
    useState([]);  
  
  useEffect(() => {  
  
    loadMeters()  
      .then(setMeters);  
  
  }, []);  
  
  return (  
  
    <div>  
  
      {meters.map(meter => (  
  
        <MeterCard  
          key={meter.id}  
          meter={meter}  
        />  
  
      ))}  
  
    </div>  
  
  );  
}
```

Після отримання даних від сервера React автоматично оновлює інтерфейс користувача та відображає актуальні показники для всіх доступних водомірів.

Для зменшення навантаження на сервер та забезпечення актуальності даних використовується механізм періодичного оновлення інформації. Наприклад, дані можуть оновлюватися кожні 60 секунд.

```
useEffect(() => {  
  
  const timer =  
    setInterval(() => {  
  
      loadMeters()  

```

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
						68
Зм.	Арк.	№ докум.	Підп.	Дата		

```

        .then(setMeters);

        }, 60000);

    return () =>
        clearInterval(timer);

    }, []);

```

У результаті користувач отримує практично актуальну інформацію без необхідності ручного оновлення сторінки.

Особливу увагу приділено відображенню стану вузлів обліку. Для цього застосовується кольорова індикація, яка дозволяє швидко оцінити працездатність пристроїв. Наприклад:

зелений колір — пристрій працює нормально;

жовтий колір — низький рівень заряду батареї;

червоний колір — відсутній зв'язок із пристроєм або зафіксована аварійна подія.

Приклад визначення статусу пристрою:

```

function getDeviceStatus(meter) {

    if (!meter.online)
        return "offline";

    if (
        meter.battery_level < 20
    )
        return "warning";

    return "normal";
}

```

Крім поточного показника водоміра, на головній панелі можуть відображатися агреговані параметри системи, зокрема загальна кількість підключених пристроїв, кількість активних аварійних повідомлень та сумарне споживання води за поточний місяць.

Дані для відображення формуються серверним застосунком на основі інформації, що зберігається у MongoDB. Це дозволяє мінімізувати навантаження

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		69

на клієнтську частину та забезпечити швидке завантаження сторінки навіть при великій кількості вузлів обліку.

Таким чином, модуль перегляду поточних показників забезпечує користувачу оперативний доступ до актуальної інформації про водоспоживання та технічний стан вузлів обліку. Використання React-компонентів, REST API та механізмів автоматичного оновлення даних дозволяє створити зручний та інформативний інтерфейс моніторингу роботи IoT системи обліку водопостачання.

На рисунку 3.4 представлено головне вікно модуля перегляду поточних показників IoT системи обліку водопостачання. Даний інтерфейс є основною інформаційною панеллю користувача та призначений для оперативного моніторингу стану вузлів обліку та поточного водоспоживання.

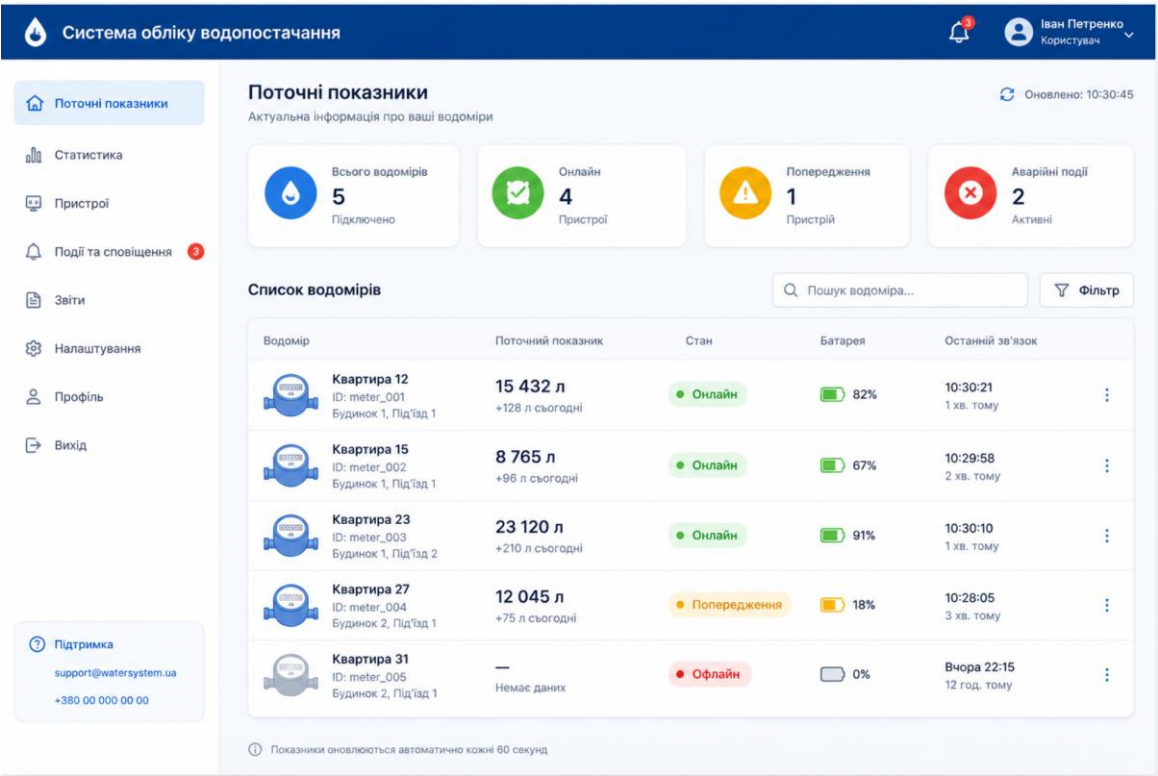


Рисунок 3.4 - Головне вікно модуля перегляду поточних показників

У верхній частині вікна відображаються узагальнені показники системи, зокрема загальна кількість підключених водомірів, кількість активних пристроїв, попереджень та аварійних подій. Це дозволяє користувачу швидко оцінити

поточний стан системи без необхідності перегляду детальної інформації по кожному пристрою.

Основну частину інтерфейсу займає таблиця водомірів, у якій для кожного вузла обліку відображаються поточний показник лічильника, стан пристрою, рівень заряду батареї та час останнього зв'язку із сервером. Для спрощення аналізу використовується кольорова індикація статусів, що дозволяє швидко виявляти проблемні пристрої або аварійні ситуації.

У лівій частині вікна розташовано навігаційне меню, яке забезпечує доступ до інших функціональних модулів системи, таких як перегляд статистики, керування пристроями, журнал подій, формування звітів та налаштування користувацького профілю.

Для зручності роботи користувача передбачено функції пошуку та фільтрації водомірів, що особливо важливо при експлуатації системи з великою кількістю підключених вузлів обліку. Дані на інформаційній панелі автоматично оновлюються через задані інтервали часу, забезпечуючи відображення актуальної інформації про стан системи в режимі, наближеному до реального часу.

Таким чином, розроблений модуль перегляду поточних показників забезпечує зручний доступ до основної інформації про водоспоживання та технічний стан обладнання, дозволяючи користувачам оперативно контролювати роботу всієї системи обліку водопостачання.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		71

ВИСНОВКИ

У результаті виконання дипломного проєкту було розроблено концепцію IoT системи обліку водопостачання для багатоквартирних житлових будинків, яка забезпечує автоматизований збір, передачу, зберігання та аналіз даних про водоспоживання.

Під час виконання роботи проведено аналіз існуючих систем дистанційного обліку води та визначено основні вимоги до сучасних IoT рішень. Встановлено, що для багатоквартирних будинків актуальною є побудова енергоефективної бездротової мережі, яка дозволяє мінімізувати витрати на монтаж та експлуатацію обладнання.

На основі проведеного аналізу для реалізації вузла обліку було обрано мікроконтролер ESP32-C6, який підтримує стандарти Wi-Fi 6, Bluetooth Low Energy та IEEE 802.15.4. Використання даного мікроконтролера дозволяє реалізувати енергоефективну Thread mesh-мережу, що забезпечує передачу даних між вузлами обліку та центральним шлюзом без необхідності прокладання додаткових комунікаційних ліній.

У роботі запропоновано архітектуру вузла обліку, до складу якого входять механічний водомір, сенсор Холла DRV5032FBLPGM, модуль ESP32-C6-Zero та автономне джерело живлення. Обґрунтовано використання сенсора Холла замість традиційних герконових датчиків, що дозволило усунути проблему «брязкіту» контактів, підвищити надійність вимірювання та зменшити енергоспоживання системи.

Розроблено структурну схему IoT системи обліку водопостачання та алгоритм її функціонування. Передбачено передачу даних від вузлів обліку через Thread mesh-мережу до центрального шлюзу, який виконує функції Thread Border Router та забезпечує подальшу передачу інформації до серверної частини за допомогою протоколу MQTT.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		72

Виконано оцінку енергоспоживання вузла обліку. Встановлено, що використання батареї ER26500 ємністю близько 8500 мА·год забезпечує автономну роботу пристрою протягом п'яти років навіть з урахуванням періодичної передачі даних та участі вузлів у роботі mesh-мережі. Додатково запропоновано механізм оперативного інформування про аварійні ситуації та можливість використання окремих постійно живлених мережевих вузлів для підвищення надійності передачі повідомлень.

Для реалізації програмного забезпечення вузла обліку обрано середовище ESP-IDF, яке забезпечує повну підтримку апаратних можливостей ESP32-C6, включаючи роботу зі стеком OpenThread, Bluetooth Low Energy та механізмами енергозбереження. Запропоновано модульну структуру програмного забезпечення, що спрощує його подальший розвиток та модернізацію.

У роботі також розроблено концепцію серверної частини системи на базі MQTT-брокера Mosquitto та бази даних MongoDB. Запропоновано структуру зберігання телеметричних даних, інформації про пристрої, користувачів та аварійні події. Для обробки повідомлень розроблено архітектуру серверного застосунку, який забезпечує прийом телеметрії, збереження даних та формування повідомлень про аварійні ситуації.

Крім того, спроектовано клієнтське програмне забезпечення у вигляді вебзастосунку, що забезпечує перегляд поточних показників водоспоживання, моніторинг стану вузлів обліку, аналіз статистики та роботу із системними повідомленнями.

Отримані результати підтверджують можливість створення масштабованої та енергоефективної IoT системи обліку водопостачання на базі сучасних бездротових технологій. Запропонована архітектура може бути використана як основа для подальшої практичної реалізації систем автоматизованого обліку водних ресурсів у житловому секторі та об'єктах комунального господарства.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		73

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Технології інтернету речей. Навчальний посібник [Електронний ресурс]: навч. посіб. для студ. спеціальності 126 «Інформаційні системи та технології», спеціалізація «Інформаційне забезпечення робототехнічних систем» / Б. Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 12,5 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2021. – 271 с – [Електронний ресурс]. – Режим доступу: <https://ela.kpi.ua/handle/123456789/42078> – Назва з екрана.
2. Gungor V. C., Sahin D., Kocak T., Ergut S., Buccella C., Cecati C., Hancke G. P. Smart Grid Technologies: Communication Technologies and Standards // IEEE Transactions on Industrial Informatics. 2011. Vol. 7, No. 4. P. 529–539. DOI: 10.1109/TII.2011.2166794.
3. Ibhaze A. E., Akpabio M. U., Akinbulire T. O. A Review on Smart Metering Infrastructure // International Journal of Energy Technology and Policy. 2020. Vol. 16, No. 1. P. 1–23.
4. Katkar S. S., Patil A., Samad K., Diggikar S. Advanced Metering Infrastructure and Its Role in Building a Smart and Sustainable Power Distribution System: A Comprehensive Review from India's Frame of Reference // International Journal of Scientific Research in Engineering and Management. 2024. Vol. 8, Issue 3. P. 1–12.
5. Putra I. M. A. A., Mahardika I. B. G., Wijaya I. N. Literature Review: Challenges and Technologies in the Development of Advanced Metering Infrastructure (AMI) for Smart Grid Systems // Majalah Ilmiah Teknologi Elektro. 2025. Vol. 24, No. 1. P. 35–46.
6. VERTECO. Smart Water Management | Intelligent Water Systems | IoT [Електронний ресурс]. – Режим доступу: <https://www.verteco.com/what-we-do/smart-water-management> (дата звернення: 20.05.2026).

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		74

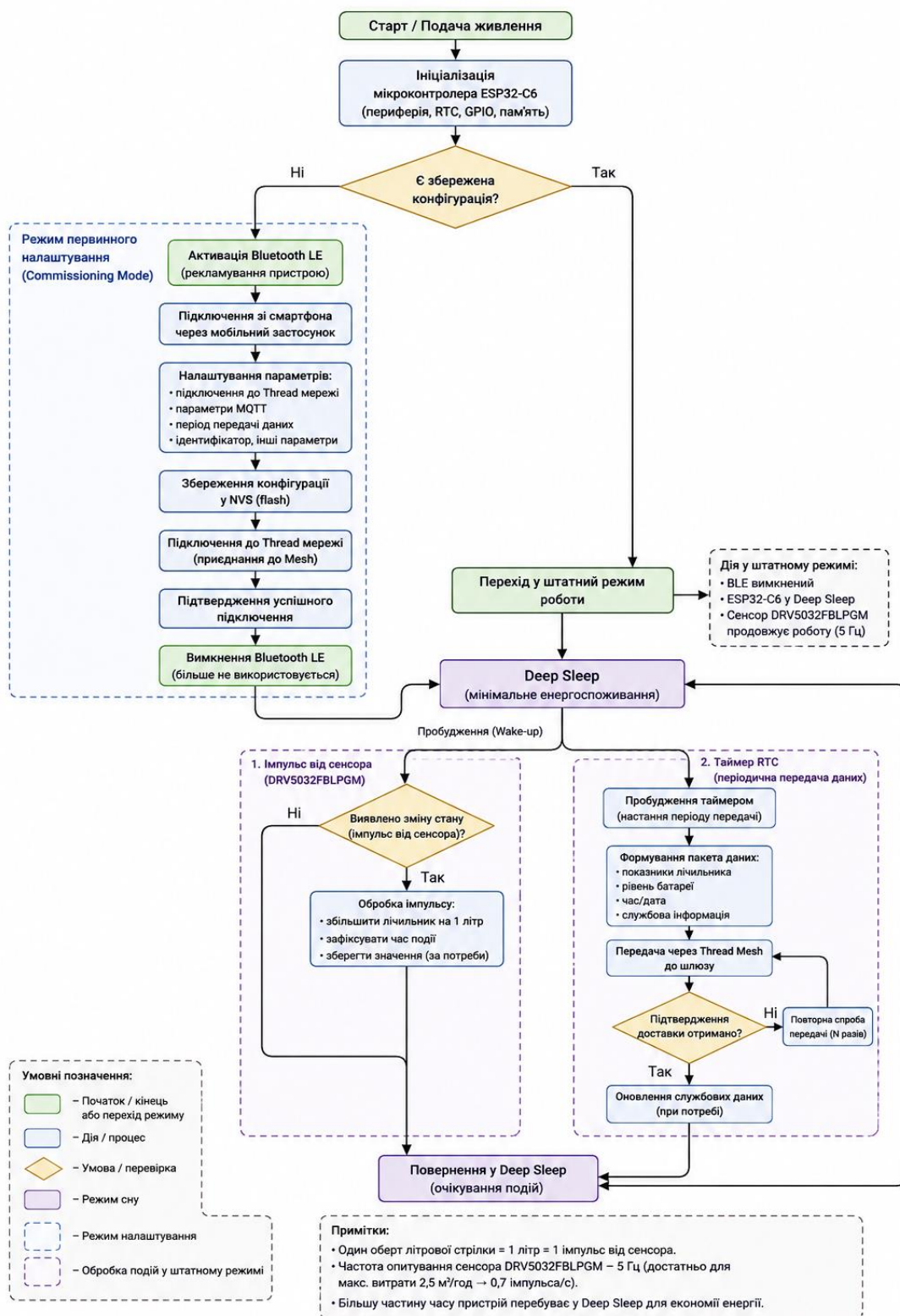
7. VERTECO. What is IoT Water Management? [Електронний ресурс]. – Режим доступу: <https://www.verteco.com/blog/smart-water-management-using-iot-what-is-it> (дата звернення: 20.05.2026)
8. iEngineering. Smart Water Management – IoT Water Metering [Електронний ресурс]. – Режим доступу: <https://smartwatermeter.ieng.tech/> (дата звернення: 20.05.2026)
9. iEngineering Group. Smart Water Management Solution [Електронний ресурс]. – Режим доступу: <https://ieng.tech/smart-water-management-solution/> (дата звернення: 20.05.2026)
10. Apator Powogaz S.A. JS Smart+ (DN15–20) – Single-jet water meters [Електронний ресурс]. – Режим доступу: <https://www.apator.com/en/our-solutions/water-and-heat/water-meters/vane-wheel-dn-15-20/js-smart-dn15-20> (дата звернення: 20.05.2026).
11. ZENNER International GmbH & Co. KG. IUWS B.One – Ultrasonic Water Meter [Електронний ресурс]. – Режим доступу: <https://zenner.com/product/iuws-b-one/> (дата звернення: 20.06.2026)
12. Thread Group. Thread Network Fundamentals [Електронний ресурс]. – Режим доступу: https://www.threadgroup.org/Portals/0/documents/support/Thread%20Network%20Fundamentals_v3.pdf (дата звернення: 20.05.2026)
13. Espressif Systems. ESP32-C6 Series Datasheet [Електронний ресурс]. – Режим доступу: https://www.espressif.com/sites/default/files/documentation/esp32-c6_datasheet_en.pdf (дата звернення: 20.05.2026)
14. MQTT.org. MQTT – The Standard for IoT Messaging [Електронний ресурс]. – Режим доступу: <https://mqtt.org/> (дата звернення: 20.05.2026)
15. MongoDB Inc. MongoDB Documentation [Електронний ресурс]. – Режим доступу: <https://www.mongodb.com/docs/> (дата звернення: 20.06.2026).

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		75

16. Texas Instruments. DRV5032 Ultra-Low-Power Digital-Switch Hall Effect Sensor Datasheet [Електронний ресурс]. – Режим доступу: <https://www.ti.com/lit/ds/symlink/drv5032.pdf> (дата звернення: 20.05.2026).
17. Шматко О. П. Основи проєктування систем автоматизованого управління : навч. посіб. / О. П. Шматко. – Дніпро : ДНУ, 2020. – 211 с.
18. Деркач В. М., Левченко І. С. Мікропроцесорні засоби та системи автоматизації : навч. посіб. / В. М. Деркач, І. С. Левченко. – Харків : ХНАДУ, 2019. – 232 с.

					ДП.МТІР.10658567.00.00.00.000 ПЗ	Арк.
						76
Зм.	Арк.	№ докум.	Підп.	Дата		

Алгоритм роботи вузла обліку водоспоживання



Змн.	Арк.А	№ докум.№	Підпис	Дата
Розроб.	Палійчук Х.М.			
Перевірив.	Заставний О.М			
Консульт.	Заставний О.			
Н. Контр.	Заставний О.			
Затверд.	Сегін			

ДП.МТІР.10658567.00.00.00.000 ПЗ

Блок-схема алгоритму
роботи вузла обліку
водоспоживання

Літ.	Арк.	Акрушів

ЗУНУ.ФКІТ.МТІР-41