

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

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Освітній рівень

Комп'ютерна мережа збору даних з розумних лічильників електроенергії

Назва теми

КВРКІ 210102.21.01.30 ПЗ

Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

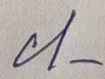
Виконав: студент IV курсу, група KI2-21-1

  
Підпис

Анна БІЛА

Ініціали, прізвище

Керівник

  
Підпис, дата

Світлана САЧЕНКО

Ініціали, прізвище

Нормоконтролер

  
Підпис, дата

Тетяна КИСІЛЬ

Ініціали, прізвище

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

  
Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

« 20 » червня 2025 р.

Хмельницький 2025

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 – ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2025 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Анні БІЛІЙ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Комп'ютерна мережа збору даних з розумних лічильників електроенергії

Керівник проекту (роботи) Світлана САЧЕНКО, к.е.н., доцент.

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

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 20.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) Мережа збору даних з розумних лічильників електроенергії, архітектура мережі, креслення, код для передачі даних

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) архітектура мережі зв'язку, архітектура програмного забезпечення, креслення проекту

6. Консультанти розділів дипломного проекту (роботи)

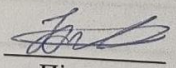
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСЛІВ, доцент кафедри КПС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КПС		

7. Дата видачі завдання « 10 » 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

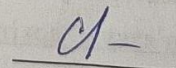
№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником		виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження		виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі		виконано
4	Робота над розділом 2 – аналіз та розробка архітектури проекту		виконано
5	Робота над розділом 3 – реалізація та моделювання частин проекту		виконано
6	Оформлення пояснювальної записки згідно вимог		виконано
7	Попередній захист ВКР		виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент

  
Підпис

Анна БІЛІА  
Ініціали, прізвище



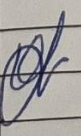
Керівник роботи

  
Підпис

Світлана САЧЕНКО  
Ініціали, прізвище

№ р я д к а	Ф о р м а т	Позначення	Найменування	К і л · л и с т і в	№ ек з	П р и м і т к а
			<u>Текстові документи</u>			
1		КВРКІ 210102.21.01.30 ПЗ	Пояснювальна записка	57		
			<u>Графічні матеріали</u>			
2		КВРКІ 210102.21.01.30 Е8	Архітектура ПЗ проєкту	1		
3		КВРКІ 210102.21.01.30 Е8	Архітектура ПЗ для кіберфізичної системи	1		
4		КВРКІ 210102.21.01.30 Е8	Апаратне забезпечення проєкту	1		

КВРКІ 210102.21.01.30 ВП

Зм	Арк	№ докум	Підпис	Дата	Літера	Аркуш	Аркушів
Розробив		Біла		19.06.25			
Перевір.		Саченко		19.06.25	ХНУ, КІ2-22-1		
Н. контр.		Кисіль					
Зав.		Павлова		19.06.25			

Відомість проєкту

ХНУ, КІ2-22-1

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Мережа збору даних з розумних лічильників електроенергії».

Автор роботи: Анна БІЛА.

Керівник роботи: Світлана Іванівна САЧЕНКО.

Пояснювальна записка: 57 с., 30 рис., 4 табл., 4 дод., 41 джерел.

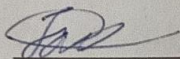
Графічна частина: 4 креслення.

Метою дипломної роботи є розробка концепції комп'ютерної мережі збору даних з розумних лічильників електроенергії з урахуванням технічних вимог, особливостей передавання даних та кібербезпеки.

Об'єктом дослідження є системи лічильників, їх комунікація із зовнішніми сервісами та сервером, створення мережі передачі даних.

Предметом дослідження є принцип функції комп'ютерної мережі збору даних з розумних лічильників електроенергії, технології зв'язку, принципи роботи її компонентів.

Під час проведення даного дослідження був використаний метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації.



Підпис студента

30.05.2025

Дата

## ЗМІСТ

ВСТУП.....	4
<b>1 КОМП'ЮТЕРНА МЕРЕЖА ЗБОРУ ДАНИХ З РОЗУМНИХ ЛІЧИЛЬНИКІВ ЕЛЕКТРОЕНЕРГІЇ .....</b>	<b>5</b>
1.1 Аналіз предметної області мереж збору даних з розумних лічильників електроенергії.....	5
1.2 Аналіз труднощів та структури створення мереж передачі даних з розумних лічильників електроенергії .....	9
1.3 Аналіз та постановка задачі роботи .....	10
1.4 Висновки до першого розділу.....	11
<b>2 ПРОЄКТУВАННЯ МЕРЕЖІ ЗБОРУ ДАНИХ З РОЗУМНИХ ЛІЧИЛЬНИКІВ ЕЛЕКТРОЕНЕРГІЇ .....</b>	<b>13</b>
2.1 Обґрунтування вибору технічних способів вирішення задачі та їх вивчення .....	13
2.1.1 Огляд та вибір технологій зв'язку .....	13
2.1.2 Визначення апаратних та програмних підсистем програмно-технічного засобу .....	19
2.2 Розробка архітектури комп'ютерної мережі .....	25
2.2.1 Ієрархічний поділ .....	25
2.2.2 Модель взаємодії між ієрархічними рівнями.....	26
2.2.3 Реалізація алгоритму роботи мережі.....	29
2.3 Вибір обладнання та програмного забезпечення.....	30
<b>3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ МЕРЕЖІ ЗБОРУ ДАНИХ З РОЗУМНИХ ЛІЧИЛЬНИКІВ ЕЛЕКТРОЕНЕРГІЇ .....</b>	<b>37</b>
3.1 Опис реалізації модулів апаратного та програмного забезпечення програмно-технічного засобу.....	37
3.2 Схема апаратного забезпечення та опис його елементів .....	38

КвРКІ.190130.19.01.19 ПЗ

Зм.	Арк.	№докум.	Підпис	Дата	Мережа збору даних з розумних лічильників електроенергії. Пояснювальна записка	Літера	Аркуш	Аркушів
Виконав		Анна БІЛА		20.06		у		2
Перевір.		Світлана САЧЕНКО		22.06				
Н.контр.		Гетяна КИСІЛЬ		19.06.15				
Затвер.		Ольга ПАВЛОВА						

ХНУ КІ2-22-1

## ВСТУП

Наш світ невпинно розвивається, з'являються нові мережі та прилади, тож осучаснення торкається також й енергетичних систем. Зокрема, запроваджуються нові способи збору та опрацювання отриманих даних для збільшення ефективності роботи системи. Одним із таких способів осучаснення та оцифрування є розумні лічильники електроенергії, які автоматизують збір, передачу та аналіз даних споживання енергетичних ресурсів.

Запровадження комп'ютерних мереж збору даних з таких пристроїв відкриває нові можливості для енергопостачальних компаній, зокрема: своєчасне виявлення перевантажень і втрат, підвищення точності розрахунків, зменшення витрат на обслуговування та покращення взаємодії з кінцевими споживачами.

Актуальність теми обумовлена необхідністю впровадження інтелектуальних систем в інфраструктуру, відповідно до світових тенденцій розвитку розумних міст, а також енергетики майбутнього. Водночас таке ефективне функціонування систем потребує ретельно спроектованої мережевої інфраструктури, яка забезпечує надійний та захищений обмін даними між лічильниками та серверними системами.

					КВРКІ 210102.21.01.30 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

## КОМП'ЮТЕРНА МЕРЕЖА ЗБОРУ ДАНИХ З РОЗУМНИХ ЛІЧИЛЬНИКІВ ЕЛЕКТРОЕНЕРГІЇ

### 1.1 Аналіз предметної області мереж збору даних з розумних лічильників електроенергії

Сама тема розумних пристроїв завжди була важливим кроком в розвитку технологій, оскільки впровадження такої техніки принесло значні зміни в розвиток та функції різноманітної інфраструктури, а також суцільно змінило наше життя. Розумні лічильники вносять зміни в процес щомісячної подачі показників, полегшуючи та роблячи дані щодо використання електроенергії/води/газу більш доступними для звичайних людей.

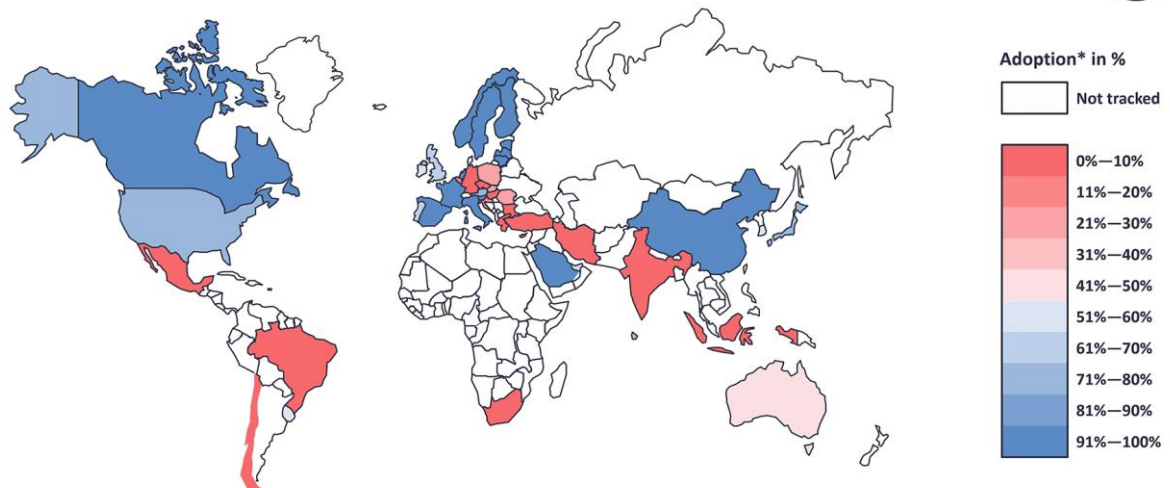
«Розумний» лічильник – це частина автоматичної системи комерційного обліку електроенергії (АСКОЕ). Він автоматично передає показання, а також інформацію про аварії, перевантаження мережі, втрати, несправність приладу тощо.” - витяг із офіційного сайту компанії “YASNO” яка займається постачанням енергоресурсів в Україні[1].

У сучасних енергетичних системах все активніше впроваджуються технології автоматизації, і такий розвиток дозволяє підвищити ефективність використання ресурсів та оперативність управління. Одним із основних напрямків цього розвитку є використання розумних пристроїв для збору та обробки даних, зокрема лічильників електроенергії, які вимірюють споживання електроенергії та передають ці дані в реальному часі до централізованих систем обліку та аналізу даних. Це прекрасний спосіб осучаснення та автоматизації таких буденних речей, як щоразовий збір показників лічильників та їх передача до державних установ, що виконувалось мануально за допомогою показникових книг до впровадження розумних пристроїв.

Надалі, розглянемо зображення, яке показує популярність використання розумних лічильників електроенергії в різних частинах світу.

					КВРКІ 210102.21.01.30 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

## Global smart electricity meter adoption 2024



\* Smart meter adoption (as of Q4 2023) = The share of smart (AMI) electricity meters installed as a share of the total number of metering points that includes residential, commercial, and industrial segments.  
Source: IoT Analytics Research 2024 – Global Smart Meter Market Tracker 2020-2030. We welcome the republishing of images but ask for source citation with a link to the original post or company website.

Рисунок 1.1 – Глобальне використання розумних лічильників електроенергії [2]

Аналізуючи зображення, можемо відмітити, що використання розумних лічильників електроенергії дуже розповсюджене в країнах з високим рівнем розвитку, таких як Канада, Китай, Японія, Італія, Франція, тощо. В таких країнах як зокрема Україна, статистика не відображена через замалий відсоток використання, або ж через недостатність дослідницьких даних. Однак, враховуючи класифікацію нашої країни по рівню розвитку до країн що розвиваються, можемо зробити припущення що відсоток використання розумних лічильників в Україні досить низький.

Однак, ми можемо спостерігати, що попри часте використання застарілих методів збору показників, наша країна також активно впроваджує системи з використанням розумних лічильників. Про це свідчить наявність відповідних розділів на офіційних сайтах українських енергетичних компаній, а також наявність компаній які працюють на українському ринку. Прикладом є компанія НІК(З), яка активно співпрацює з нашою країною, постачаючи розумні технології, зокрема розумні лічильники.



створенні макету мережі для збору даних, а саме: вибір обладнання, яке буде використане при створенні мережі, протоколи зв'язку між її елементами, програмне забезпечення, яке буде обробляти та зберігати данні, засоби для захисту інформації, а також додаткові протоколи для забезпечення безперервного функціонування.

Для налаштування та роботи комп'ютерної мережі важливим фактором є засоби зв'язку та передачі інформації. Для реалізації цього є декілька способів зв'язку розумних лічильників електроенергії, кожен з яких має власні переваги та недоліки. Розглянемо їх у таблиці 1.1

Таблиця 1.1 – Засоби організації зв'язку з розумними лічильниками

1.	PLC (Power Line Communication) - передача даних через електромережу	+ Не потребує додаткових комунікаційних ліній (використовується вже існуюча електромережа). + Підходить для щільної забудови типу міст, мегаполісів, багатоквартирних будинків. - Можливість виникнення завад від електроприладів. - Низька швидкість передачі даних. - Обмеження в можливості масштабування системи.
2.	GSM/3G/4G (мобільний зв'язок)	+ Добре підходить для віддалених/сільських територій. + Не потребує додаткового прокладання кабелів. + Підтримується операторами мобільного зв'язку. - Витрати на абонплату за кожен лічильник. - Залежність від якості мобільної мережі. - Потреба в SIM-картах/додатковому адмініструванні.
3.	Wi-Fi / Ethernet	+ Висока швидкість передачі даних. + Можливість інтеграції з локальними мережами. - Обмежений радіус дії. - Потреба у стабільному живленні та охопленні Wi-Fi мережею. - Непридатні для локацій без інфраструктури.

Кінець таблиці 1.1

4.	LPWAN (Low Power Wide Area Network) - LoRa, NB-IoT	<ul style="list-style-type: none"> <li>+ Досить низьке енергоспоживання.</li> <li>+ Дальність дії до кількох кілометрів.</li> <li>+ Велика масштабованість.</li> <li>- Низька пропускна здатність (лише для періодичних коротких повідомлень).</li> <li>- Потреба у створенні спеціалізованої інфраструктури.</li> <li>- Має певні частотні обмеження.</li> </ul>
5.	Гібридні способи	<ul style="list-style-type: none"> <li>+ Можливість адаптації до різних умов експлуатації.</li> <li>+ Підвищення надійності та резервування каналів зв'язку.</li> <li>- Вища ніж у інших методів складність впровадження та обслуговування.</li> <li>- Додаткові витрати на координацію компонентів.</li> </ul>

В книзі «Design and Implementation of an Innovative Internet of Things (IoT) Based Smart Energy Meter» запропоновано розробку та впровадження інноваційного Інтернету речей на основі розумного лічильника енергії. Ця модель описує його конструкцію разом з його роботою. Це дозволить зробити лічильник зручним і точно вимірювати споживання енергії, також це зменшить витрати енергії. (5)

## 1.2 Аналіз труднощів та структури створення мереж передачі даних з розумних лічильників електроенергії

Різні виробники розумних лічильників використовують власні протоколи обміну даними, інтерфейси та формати збереження інформації. Це створює деякі труднощі з інтеграцією обладнання в єдину мережу, особливо у великих системах, де одночасно використовуються пристрої різних брендів. Це можна вирішити

створенням єдиного стандарту, або ж створенням адаптування різних мереж та систем.

У районах країни далеких від міста(наприклад сільська місцевість) часто спостерігається нестабільний мобільний зв'язок або повна його відсутність. Це ускладнює передачу даних у режимі реального часу. Вирішення проблеми може займати великий відрізок часу, оскільки напряду пов'язаний з мережним покриттям країни, або ж має комбінувати в собі сучасні з більш застарілими методами, що дещо суперечить концепції.

Також великою проблемою може стати доступність розумних лічильників, оскільки нові впровадження часто мають високу ціну на їх старті перед тим, як увійти в широке коло користування, що може бути проблемою для багатьох підприємств що мають обмежений бюджет.

Варто зазначити, що розумні лічильники, особливо ті, які встановлені у важкодоступних місцях або на значній відстані, складно обслуговувати у разі їх виходу з ладу або втрати з'єднання. Також усі працівники мають бути кваліфіковані та розумітися у принципах роботи лічильників, що також є значним ускладненням для масового впровадження систем розумних лічильників.

Також, великою проблемою для всіх новітніх пристроїв залишаються кібератаки та безпека передачі інформації. Тому важливо забезпечити захист передачі даних, хмарного серверу та самих програм моніторингу.

### 1.3 Аналіз та постановка задачі роботи

Для виконання роботи потрібен відповідальний підхід до розробки ефективної та зручної системи. Окрім вибору правильного обладнання, ця стратегія повинна включати створення програмної логіки, мережевої архітектури та забезпечення протоколів безпеки інформації.

**Адаптивність та модульність архітектури.** Встановлення модульної мережевої інфраструктури, яка спрощує розширення або модифікацію компонентів системи без шкоди для її цілісності. Це означає, що основну систему збору даних

					КВРКІ 210102.21.01.30 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

можна підключити до нових видів лічильників, додаткових протоколів зв'язку або аналітичних модулів без необхідності комплексного перепроєктування системи.

**Технології зв'язку.** Не існує універсальної технології передачі даних, тому найоптимальнішим рішенням є обрати найбільш зручні способи передачі даних. Наприклад, використання гібридних схем є підходящим способом вирішення проблеми. Головне – зберегти баланс та не перенасичити систему великою кількістю рішень. Обрати найпідходящі для реалізації способи буде оптимальним варіантом для реалізації технології зв'язку.

**Хмарні платформи та API.** Використання хмарних сервісів з відкритим API може гарантувати гнучкий доступ до даних з будь-якого місця, а також інтеграцію з іншими сервісами. Завдяки цьому буде спрощене виведення даних до зовнішніх систем, мобільних додатків тощо.

**Локальна буферизація та кешування.** Проміжне локальне сховище даних у лічильнику або на шлюзі гарантуватиме безперервність збору даних у разі втрати з'єднання. Це допоможе уникнути втрати даних. Також, шлюзи можуть мати набір додаткових функцій, які можуть забезпечити більш продуктивну роботу системи.

**Інтеграція з існуючими бізнес-інформаційними системами.** Завдяки підтримці поширених форматів обміну (JSON, XML, MQTT, т.д.) та API, системи управління енергією, аналітики або автоматизації можуть бути інтегровані з системою збору даних. Це підвищить суцільну ефективність системи.

**Інформаційна безпека.** Усі елементи мережі повинні відповідати сучасним вимогам кібербезпеки, тому обов'язковим буде введення таких способів захисту як: шифрування даних під час передачі (наприклад TLS/SSL), автентифікація пристроїв, обмеження доступу за IP адресою, контроль цілісності інформації та регулярне оновлення ПО.

#### 1.4 Висновки до першого розділу

У теоретичній частині цієї роботи було розглянуто предметну область, а також окреслено її основні проблеми, наприклад такі як відсутність стандартів,

					КВРКІ 210102.21.01.30 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

нестабільність каналів зв'язку, кіберзагрози та складність обслуговування. Було визначено основні функції роботи розумних лічильників електроенергії.

На основі порівняльного аналізу наявних технологій зв'язку (PLC, GSM, LPWAN, Wi-Fi, гібридні моделі) було визначено, що жодне рішення не є універсальним, тому ефективна система повинна поєднувати декілька різних підходів до вирішення, залежно від наданих умов.

У висновку, отримані в ході практики теоретичні знання, навички та аналіз предметної області створюють основу для реалізації ефективною цифровою системи збору даних з використання електроенергії, яка відповідає сучасним технічним, економічним вимогам та вимогам безпеки.

					КВРКІ 210102.21.01.30 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2 ПРОЄКТУВАННЯ МЕРЕЖІ ЗБОРУ ДАНИХ З РОЗУМНИХ ЛІЧИЛЬНИКІВ ЕЛЕКТРОЕНЕРГІЇ

2.1 Обґрунтування вибору технічних способів вирішення задачі та їх вивчення

Після проведеного аналізу задачі у розділі 1 було виявлено, що її рішення потребує гнучкого та адаптивного підходу, оскільки є безліч факторів, які можуть повпливати на якість та можливість роботи мережі.

Від вибору архітектури мережі та технологій зв'язку в мережі збору даних залежить стабільність самої системи. Хоч гібридні системи й мають перелік недоліків, вони будуть найбільш оптимальним рішенням для різних типів інфраструктури. Оскільки було обрано гібридний вид систем зв'язку, варто обрати й гібридну архітектуру мережі.

Гібридна архітектура мережі забезпечує можливість зберігання переданих даних на центральному хмарному сховищі, як і централізована архітектура, однак передача даних також розподілена між додатковими локальними вузлами, що допомагає уникнути деяких недоліків суто централізованої архітектури. Перевагою даної системи буде гнучкість, що дозволить підлаштувати її під різні способи комунікації між розумними лічильниками, базами даних та серверами.

Способи мереж зв'язку залежатимуть від типу інфраструктури, де вони розташовуватимуться. Після аналізу недоліків та переваг кожного із способів, виділимо найбільш підходящі для створення якісної мережі передачі даних та зв'язку.

### 2.1.1 Огляд та вибір технологій зв'язку

У межах розвитку архітектури комп'ютерної мережі збору даних були розглянуті технології LPWAN, в які також входила технологія NB-IoT. Narrowband

					КВРКІ 210102.21.01.30 ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

IoT(або ж інтернет речей) – це вузькосмуговий стандарт радіозв'язку, який був розроблений спеціально для пристроїв інтернету речей.

Цей технічний спосіб вирізнявся високою ефективністю для задач з енергомоніторингу. NB-IoT був стандартизований у 3GPP Release 13 і побудований поверх LTE, але з вузькою смугою 200 кГц, забезпечуючи кращий обхват і менш енергоспоживання в порівнянні з іншими схожими способами. Він працює завдяки мережам мобільних операторів й часто працює в мержі мобільних операторів (частіше всього на базі 4G/5G операторів). Зв'язок через інтернет речей ідеально підійде для таких систем, де буде необхідна передача невеликого обсягу даних з великої кількості пристроїв.

Адаптивні алгоритми передачі, які змінюють частоту звернень залежно від зміни даних, дозволяють знизити кількість передач на 86–87 %, що суттєво економить ресурс батареї, гарантуючи надійність та своєчасність передавання усіх даних.

На фізичному рівні NB-IoT використовує модифіковані LTE-канали (OFDM для DL, SC-FDMA для UL), що забезпечить стабільне покриття навіть у важкодоступних місцях. Сигнали витримують приблизно на 20 дБ глибше затухання, ніж LTE, що покращує проникнення сигналу наприклад у підвальні приміщення чи товсті стіни будівель. (8)

Інтернет речей може використовуватися для сільських, віддалених територій або об'єктів. Також, в залежності від вибору користувачів, можуть бути встановлені лічильники з модулем зв'язку NB-IoT, якщо в обраній місцевості є покриття 4G. Це надійний спосіб, оскільки система працюватиме безперебійно навіть зі слабким сигналом. Також, NB-IoT має низьке енергоспоживання. (8)

Для підключення до покриття Інтернету потрібні модеми, які можуть бути встановлені окремо або ж бути вбудовані в самі розумні лічильники, модеми підключаються до мережі й передають дані до центрального сервера, як зображено на рисунку 2.1

					КВРКІ 210102.21.01.30 ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

Power Line Communication – це технологія для передачі даних вже існуючими лініями електропостачання. Інакше кажучи, це комунікація між розумними лічильниками та сервером.

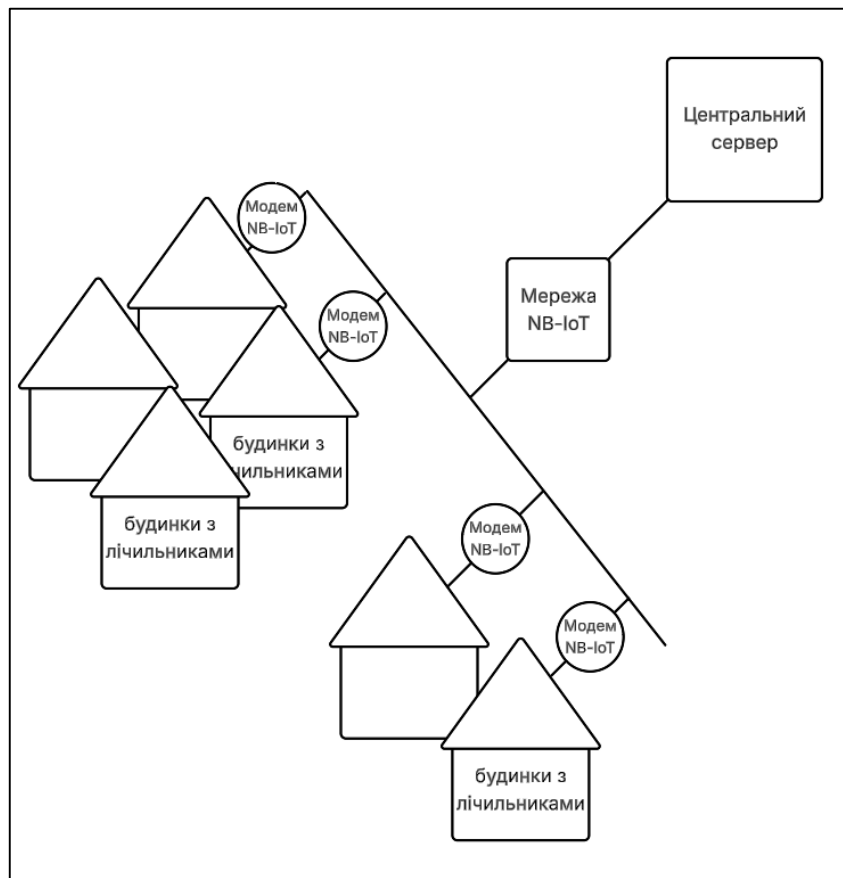


Рисунок 2.1 – Схематичне зображення роботи мережі з технологією підключення NB-IoT

PLC може використовуватись в міських місцевостях, оскільки в більшості інфраструктури проведено електричне з'єднання. Цей спосіб мережевого покриття досить легкий в реалізації, оскільки не потребує додаткового проведення ліній та існує поверх уже існуючої мережі, а також добре працюватиме навіть в густонаселених містах. З'єднання по такій мережі відбувається з допомогою додаткових підстанцій для буферизації та адаптування сигналу, після чого дані через шлюз надходять до центрального серверу. (зображено на рисунку 2.2). (7)

PLC забезпечить одночасну роботу з декількома лічильниками, зменшить кількість точок обслуговування та дозволить централізовано збирати показники з

лічильників через єдиний шлюз. Це зробить розроблену архітектуру системи простішою та менш затратною.

Технологія PLC, а саме її варіант NB-PLC(Narrowband PLC), базується на модуляції сигналу у вузькому частотному діапазоні біля 3–148 кГц, що дозволить здійснювати передачу інформації на відстані до кількох сотень метрів при мінімальному впливі завад від електричних пристроїв що також підключені до мережі. У реальних умовах стабільна робота PLC-каналу можлива приблизно в межах 100–300 метрів. Проте, при реалізації проекту можуть бути передбачені невеликі відстані для передачі даних по PLC, до 20-ти метрів. (8)

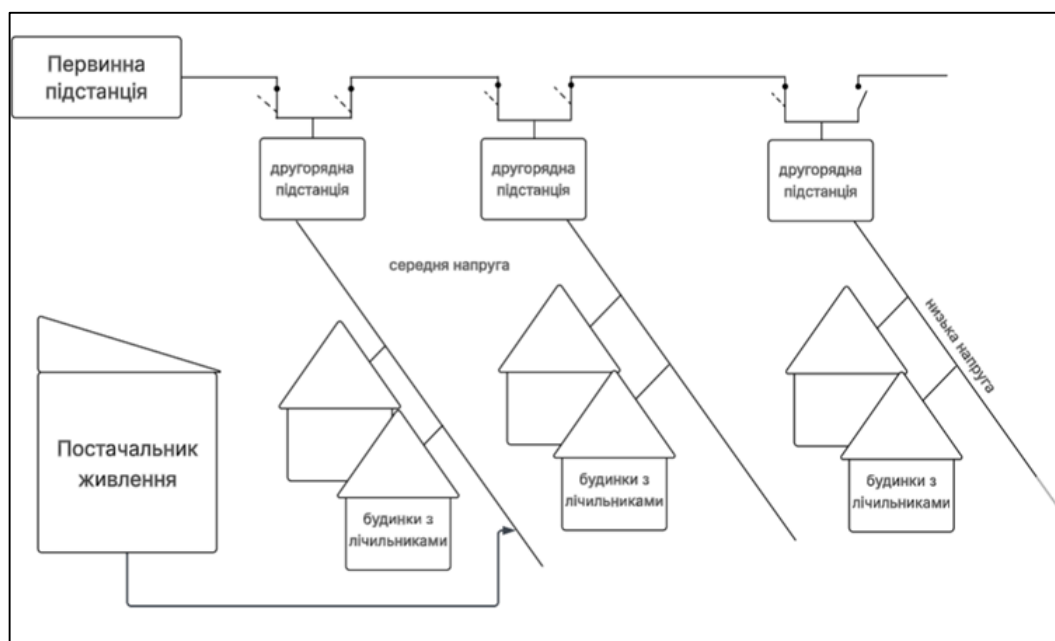


Рисунок 2.2 – Схематичне зображення роботи PLC мережі та комунікації її компонентів

GSM це одна з найрозповсюдженіших у світі технологій зв'язку, вона забезпечує передачу таких даних як голос, SMS та інформаційні дані. Для цілей мереж збору даних з розумних лічильників може використовуватися GSM GPRS/EDGE (2G) або 3G/4G LTE, що дозволить пристроям передавати показники на сервер бездротовим способом через мережу мобільного оператора(9). Дещо схожа з технологією NB-IoT але є більш доступною та універсальною, хоч і має деякі недоліки у вигляді потреби у сім-карті та абонплати. За допомогою технології

SMS лічильники передають дані в мережу, дані передаються до сервера оператора мобільної мережі після чого надходять до центрального серверу збору даних, як показано на рисунку 2.3.

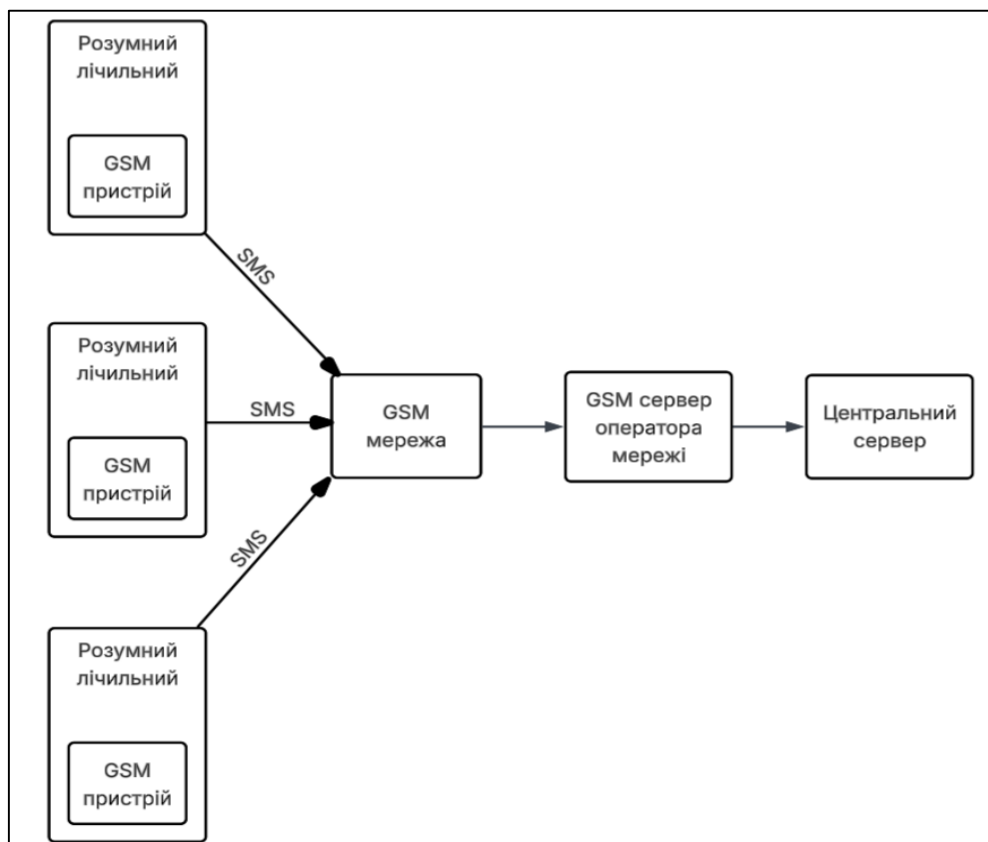


Рисунок 2.3 – Схематичне зображення комунікації через технологію зв'язку GSM

В розумних лічильниках, що передають зібрані дані через GSM, вбудовано додаткові пристрої, що забезпечать передачу даних до мережі. Мережа інтегрується з сервером оператора зв'язку, сім-карта якого використовується в пристрої, після чого дані по швидкій лінії передачі надходять до центрального серверу. Комунікацію розумних лічильників та центрального серверу за допомогою технології зв'язку GSM схематично зображено на рисунку 2.3. GSM або 4G в межах проекту може бути використано як резервну технологію, або ж технологію, що покриватиме ті інфраструктурні ділянки, яких не досягатиме ні PLC ні NB-ІоТ.

Таблиця 2.1 – Характеристика технологій зв'язку

	Інтернет речей	PLC	GSM
Тип зв'язку	бездротовий (мобільна мережа)	провідний (через електромережу)	бездротовий (через операторів зв'язку)
Радіус дії	до 15-ти км	до 1 км	до 35-ти км
Швидкість	100–250 кбіт/с	10–500 кбіт/с (NB-PLC)	50–200 кбіт/с (2G), до 1–2 Мбіт/с (3G)
Енергоспоживання	дуже низьке	низьке / середнє	середнє / високе
Надійність	висока	середня, дуже залежить від електромережі	висока але залежить від покриття
Застосування	для розподілених об'єктів та віддалених відстаней	для багатоквартирних будинків та щільної забудови	для віддаленої місцевості та далеких сільських регіонів

В таблиці 2.1 наведено основні характеристики усіх трьох технологій зв'язку у порівнянні, показані їх особливості, що допоможе детальніше проаналізувати їх відмінності й схожості для подальшого проектування мережі.

На рисунку 2.1 було схематично представлено загальну структурну архітектуру розроблюваної системи збору даних з розумних лічильників електроенергії. Вона охоплює споживацьку та серверну секції, канали зв'язку,

модулі безпеки, а також механізми зовнішньої інтеграції, схематично зображуючи зв'язок між ними

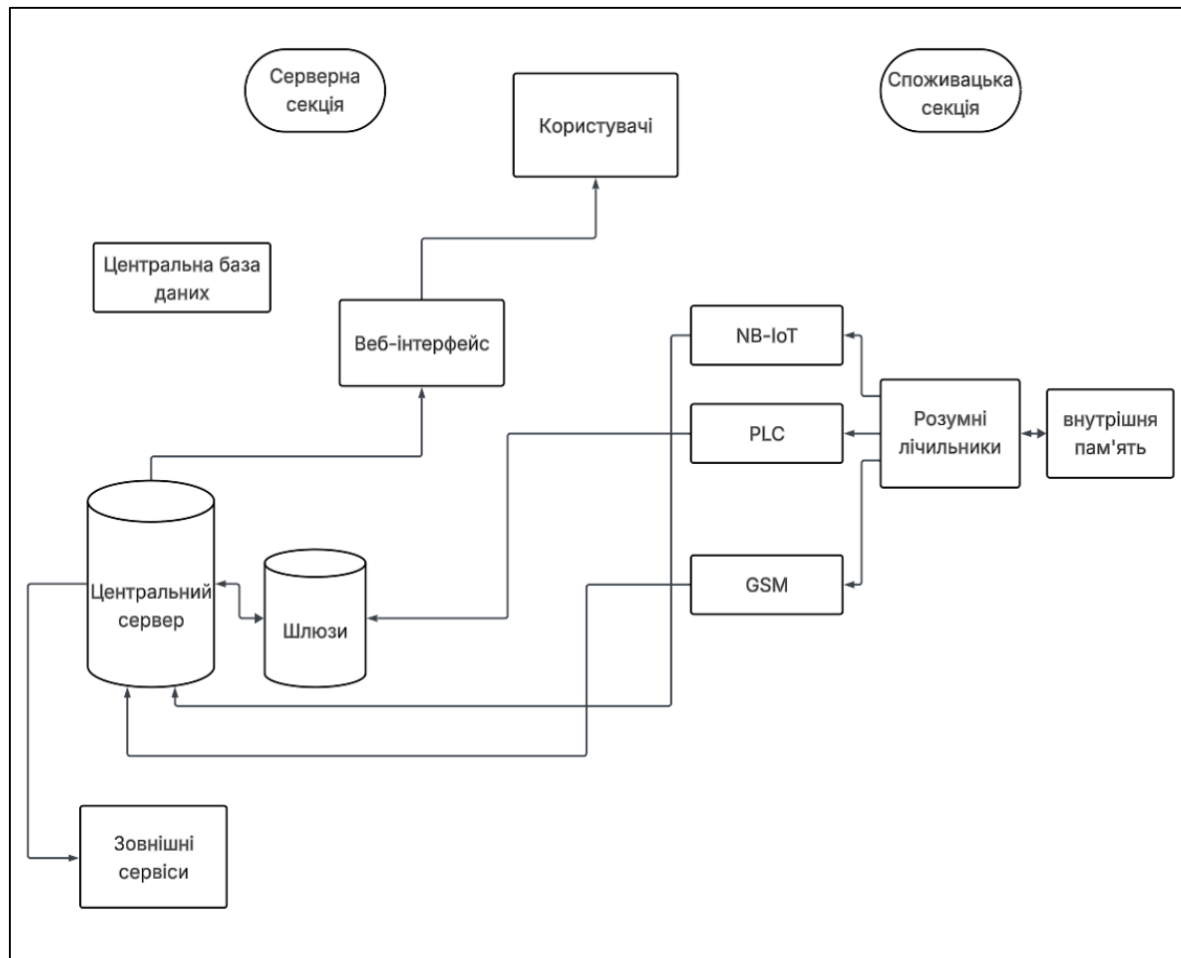


Рисунок 2.4 – Схематичне зображення мережі

### 2.1.2 Визначення апаратних та програмних підсистем програмно-технічного засобу

Для побудови ефективної комп'ютерної мережі збору даних з розумних лічильників електроенергії потрібно визначити структуру програмно-технічного комплексу (ПТЗ), який буде реалізовувати всі ключові для роботи функції: збір, передача, обробка, зберігання і захист інформації.

Програмно-технічний комплекс складається з двох підсистем:

#### **Апаратна підсистема.**

Це система в яку входить усе апаратне обладнання мережі. Вони пов'язані між собою різними способами зв'язку, а саму комунікацію обладнання між собою та користувачем забезпечує програмна підсистема.

Вона складається з таких компонентів:

- 1) Розумні лічильники електроенергії. Вони вимірюватимуть споживання електроенергії та формуватимуть пакети даних готових до передачі;
- 2) Комунікаційні модулі/шлюзи. Вони забезпечуватимуть передачу даних від лічильників до центральної системи;
- 3) Локальні контролери або пристрої збору даних (концентратори даних). Вони поєднуюватимуть зібрану з декількох лічильників інформацію, забезпечуватимуть попередню обробку/кешування об'єднаних даних;
- 4) Додаткове інфраструктурне обладнання, яке складатиметься з мережевих маршрутизаторів, комутаторів, блоків для забезпечення безперебійного живлення, датчиків температури чи вологості тощо.



Рисунок 2.4 – Схема апаратної підсистеми

На рисунку 2.4 схематично зображено взаємодію модулів апаратної підсистеми мережі, їх пряму ціль та складові.

Апаратна підсистема включає в себе 2 модулі, які реалізуються всередині мережі на апаратному рівні.

### **Збір даних**

Модуль здійснює фіксацію показників електроспоживання та формування телеметричних повідомлень й складається із таких компонентів:

- 1) Розумні лічильники;
- 2) EEPROM або Flash для кешу;

Виконує такі функції, як вимірювання енергії, запис подій(таких як перенапруга або відключення), а також зберігає дані при мережевих збогах щоб запобігти їх втраті та забезпечити відсилання при відновленні зв'язку.

### **Передача та маршрутизація**

Це модуль який здійснює доставку даних від пристроїв збору до центрального вузла обробки.

Складові модулю передачі та маршрутизації:

- 1) Комунікаційні модулі (NB-IoT модем, PLC-модуль, GSM-модуль);
- 2) Проміжні вузли: шлюзи, маршрутизатори;
- 3) Протоколи: MQTT, CoAP, DLMS/COSEM, VPN.

Модуль передачі та маршрутизації включає в себе пристрої, які виконують функції агрегації даних з групи лічильників, перетворення протоколів для забезпечення сумісності, шифрування трафіку, виявлення мережевих втрат та ведення журналів подій безпеки.

## **2. Програмна підсистема**

Програмна підсистема – це елемент ПТЗ, який відповідає за справну роботу апаратної підсистеми, передачу шифровку та дешифровку даних, їх підготовку до читання тощо. Програмне забезпечення системи включатиме в себе:

					КВРКІ 210102.21.01.30 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

- 1) Програмне забезпечення лічильників і шлюзів - це прошивки, що забезпечуватимуть збір показань, шифрування даних та зв'язок з мережею;
- 2) Проміжне програмне забезпечення шлюзів (middleware) оброблятиме дані, забезпечуватиме їх кешування, буферизацію та ретрансляцію;
- 3) Програмне забезпечення серверу для збору даних. Воно складатиметься з таких компонентів як:
  - a) Системи для зберігання (бази даних, файлові сховища);
  - b) Програми для обробки та аналізу даних;
  - c) Модулі безпеки (автентифікація, контроль доступу, тощо);
- 4) Інтерфейс адміністратора. Цей інструмент існуватиме для адміністрованого налаштування та моніторингу роботи усієї системи;
- 5) Користувацький сайт або мобільний додаток, що буде використовуватись користувачами для того, аби мати можливість моніторити дані зі свого лічильника. Він відображатиме усі потрібні дані;
- б) Також можуть бути запроваджені системи й можливості зв'язку з державними установами у разі виникнення проблем.

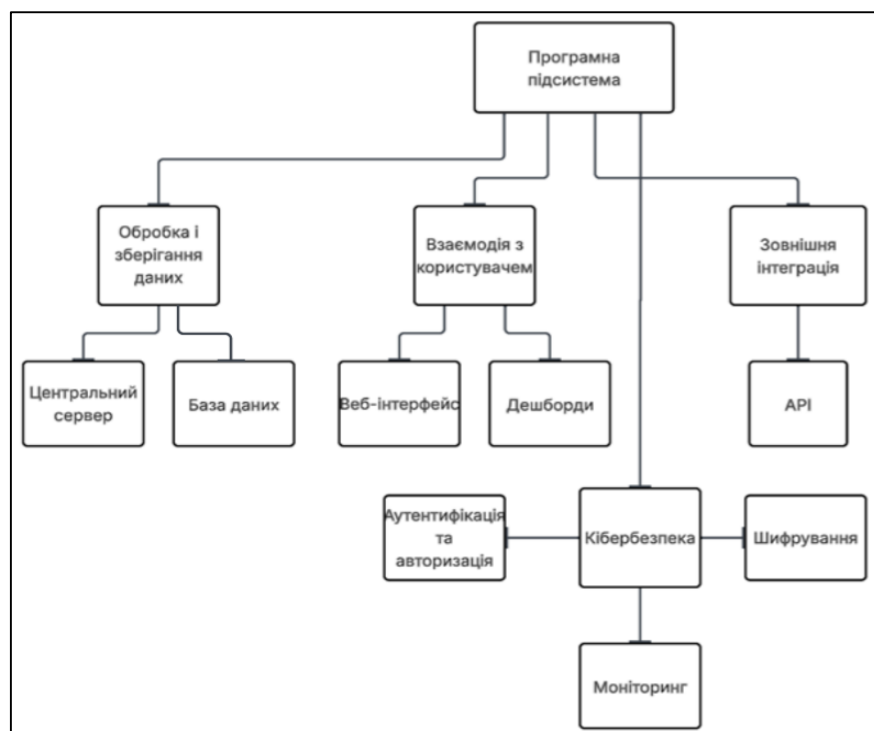


Рисунок 2.6 – Схема програмної підсистеми

Зображена на рисунку 2.6 програмна підсистема забезпечуватиме обробку інформації, що надходить та виконуватиме функцію, яка дозволить взаємодію самої комп'ютерної мережі та користувачів. Також, буде забезпечена мережева безпека. Підсистема розділена на 4 модулі, кожен з яких виконуватиме свою програмну функцію. Їх ціль – бути проміжною ланкою між користувачем та апаратною підсистемою.

Детальніше розглянемо ці 4 модулі:

### **Обробка та зберігання даних**

Модуль виконує аналіз отриманих даних, фільтрує їх та зберігає. Цей модуль складається з таких компонентів:

- 1) Центральний сервер або хмара;
- 2) База даних (PostgreSQL, InfluxDB, TimescaleDB, MongoDB);
- 3) Обробники подій (скрипти на Python, Node.js, cron-завдання).

Прямими задачами цього модулю є сортування даних за такими характеристиками як дата надходження, ID пристрою з якого було передано дані, тип споживання, виявлення стрибків напруги, пікових навантажень, підготовка даних для API або виведення на інтерфейс, а також створення резервних копій та бекапів переданих даних тощо.

### **Взаємодія з користувачем**

Призначення цього модулю – надавати доступ до інформації користувачам, операторам, споживачам та адміністраторам. ПЗ візуалізує отримані дані, роблячи їх зручними для читання. Складається з таких компонентів:

- 1) Веб-інтерфейс (React, Vue.js);
- 2) Дешборди (Grafana, Kibana).

Веб-інтерфейс виконуватиме такі важливі для розуміння інформації користувачем функції, як перегляд поточних показників, а також може мати історію раніше переданих, зможе формувати звіти, матиме користувацькі налаштування для зручностей (наприклад налаштування сповіщень), а також налаштування доступу за дозволами.

					КВРКІ 210102.21.01.30 ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

## **Зовнішня інтеграція (API)**

Цей модуль несе в собі функцію інтеграції з різними інформаційними та обліковими системами, що може поліпшити доступність.

- 1) REST API,
- 2) WebSocket API;

Він виконуватиме такі прості задачі, як наприклад надавати дані зовнішнім системам таким як CRM, ERP, SCADA, також забезпечити обмін інформацією з обліковими реєстрами держави, а також робить можливим використання API для покращення роботи мережі.

## **Кібербезпека**

Функція модулю - захищати передані дані, попереджати крадіжки або витоки, а також шифрувати їх перед передачею. Складається з таких складових:

- 1) Системи шифрування (TLS, SSL, VPN);
- 2) Аутентифікація та авторизація (ACL, 2FA, токени);
- 3) Моніторинг та IDS/IPS.

Конкретні задачі для модулю кібербезпеки – це захищати усю мережу у випадках загрози цілісності даних. Протоколи кібербезпеки вивлятимуть спроби несанкціонованого доступу, запобігатимуть підміні або втраті даних а також автоматично блокуватимуть показ даних при підозрілій активності.

Апаратна й програмна частини матимуть тісний зв'язок між собою, й робота однієї підсистеми напряму залежатиме від іншої. Ціллю апаратної підсистеми буде збирання даних та їх безперервну передачу, в той час як програмна частина забезпечуватиме їх обробку, зберігання, захист та надання користувачу доступу до них. Те, наскільки надійною та функціональною буде уся система залежатиме від спільної роботи цих двох підсистем.

					КВРКІ 210102.21.01.30 ПЗ	Арк. 23
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2.2 Розробка архітектури комп'ютерної мережі

Архітектура комп'ютерної мережі – дуже важлива складова системи, оскільки це структура, на якій в подальшому будуватиметься уся мережа. Тому, при проектуванні архітектури треба обдуманно підійти до структури, типу й принципу її роботи.

База архітектури буде гібридною(через різні технології зв'язку), матиме ієрархію рівнів, а обробка даних відбуватиметься з допомогою центрального серверу та додаткових вузлів, що виконуватимуть роль полегшуючих факторів для найбільш навантажених частин системи та зможуть перетримувати дані в разі довготривалих втрат зв'язку із центральним сервером. Найближчим до розроблюваної архітектури буде тип «термінал – головний комп'ютер» суть якої – існування центрального комп'ютера, терміналів(якими виступатимуть лічильники) та мультиплексорів, якими виступатимуть шлюзи. (14)

### 2.2.1 Ієрархічний поділ

Ієрархічний поділ матиме три рівні: найвищий(центральный сервер), середній(проміжний) та найнижчий або ж кінцевий(самі лічильники):

1) Кінцевий рівень. На цьому рівні відбувається первинний збір даних та їх підготовка по передачі. Рівень архітектури складатиметься з двох компонентів, а саме:

а) Самі розумні лічильники. Це пристрої, що вимірюватимуть споживання електроенергії та передаватимуть дані на наступні рівні архітектури мережі;

б) Технології зв'язку, які було обрано в минулому підрозділі а саме NB-IoT, PLC та GSM. Це модеми або вбудовані всередину лічильників модулі.

2) Проміжний рівень

Шлюзи займатимуться агрегацією даних з лічильників, будуть здійснювати попередню обробку та передаватимуть їх на центральний сервер.

					КВРКІ 210102.21.01.30 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

Використовуватимуться в як адаптери для лічильників, що підключені до PLC, а також для додаткової обробки даних при перенавантаженні системи.

Вони виконуватимуть такі функції як:

- c) Буферизація даних та їх опрацювання у разі навантаження, або ж перетримка даних при втратах зв'язку із сервером;
- d) Перевірка цілісності даних та їх шифрування;
- e) Об'єднання інформації, а також її фільтрація;
- f) Слугуватимуть проміжними пунктами (адаптерами) між центральним сервером та PLC лічильниками.

### 3) Центральний рівень

На центральному рівні знаходитиметься центральний сервер, в якому відбуватиметься основна обробка, зберігання та аналіз даних. Після чого дані будуть передаватися уже до користувачів.

Функції центрального рівня включатимуть в себе:

- a) Зберігання історичних даних;
- b) Аналітику та візуалізацію споживання;
- c) Інтеграцію з іншими системами або зовнішніми сервісами з допомогою API.

Така структуризація допоможе стабілізувати систему й отримати максимальну версативність мережі, вона зможе працювати навіть при зміні деяких факторів.

### 2.2.2 Модель взаємодії між ієрархічними рівнями

На відміну від інших, ті лічильники, які використовуватимуть технологію зв'язку NB-IoT не потребуватимуть додаткових шлюзів, оскільки можуть довготривало зберігати дані, не потребуючи додаткових впроваджень, а також не потребують адаперного підключення. Через структуру покриття та потребу в адаптерах, якими також можуть виступати шлюзи, а також можливість

перенавантаження, шлюзи будуть забезпечувати безперебійну роботу систем, які зв'язуються із сервером через PLC (8).

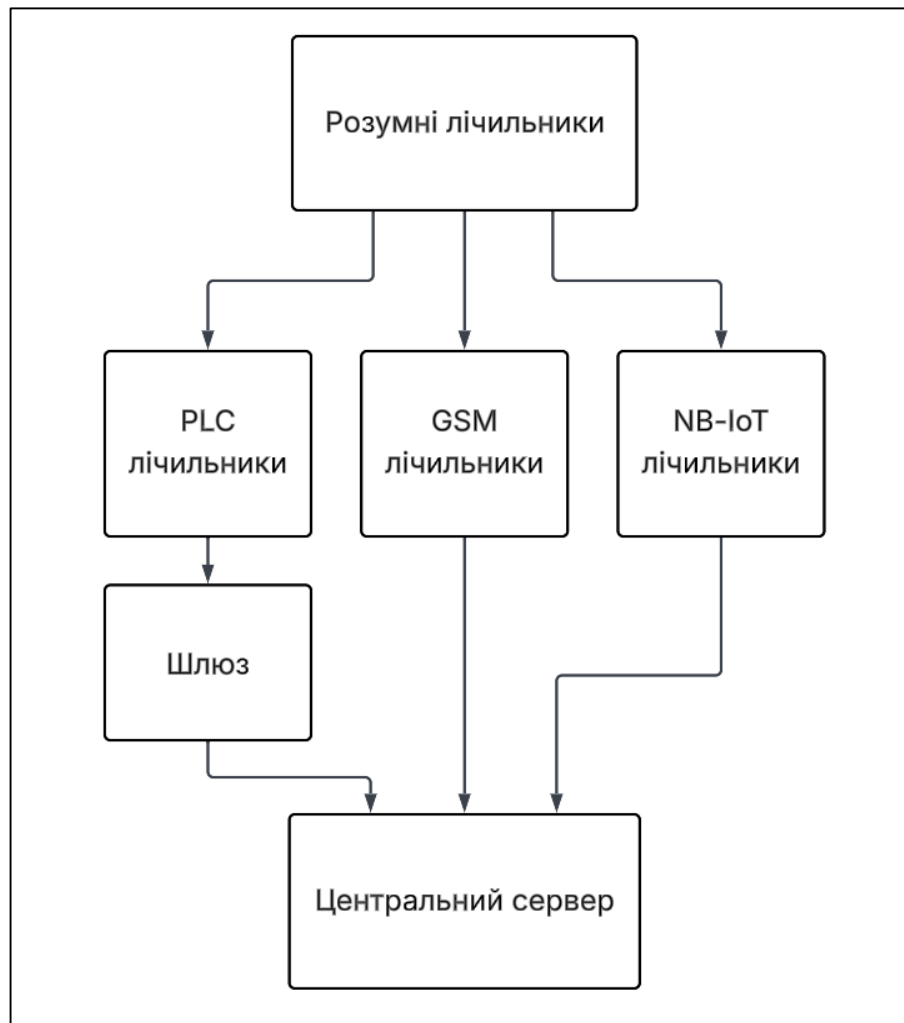


Рисунок 2.7 – Ієрархія архітектури мережі

Взаємодія між рівнями архітектури базуватиметься на таких стандартних протоколах та інтерфейсах:

1) Протоколи для передачі даних

MQTT протокол буде гарним рішенням для передачі телеметричних даних. MQTT - це легкий протокол обміну повідомленнями для малих сенсорів і мобільних пристроїв, оптимізований для мереж з високою затримкою або ненадійних з'єднань. Вважається легким протоколом обміну повідомленнями для відправки невеликих пакетів інформації в додатках з обмеженою пропускнуою здатністю [12].

HTTP або ж HTTPS протокол може використовуватися для взаємодії з різними веб-інтерфейсами та API. DLMS або COSEM є стандартними протоколом для взаємодії даних та розумних лічильників, тому буде застосовано для лічильників, які не підтримують MQTT. DLMS/COSEM є найбільш використовуваним протоколом для систем АМІ, також дозволяє інтеграцію в мережеві сегменти, які підтримують 5G, використовуються для підвищення надійності обміну енергетичними вимірюваннями та доступності достатніх даних для розрахунку енергетичного попиту. [13]

## 2) IP-адресація:

NB-IoT матиме динамічні IPv6-адреси з NAT (через мобільного оператора); PLC та шлюзи матимуть внутрішні приватні IPv4-адреси (наприклад, 192.168.0.0/16);

Сервер матиме статичну публічну IP-адресу з відкритим портом API (наприклад 443, TLS).

## 3) Протоколи які забезпечать безпеку мережі

Найважливішими протоколами для захисту мережі будуть TLS чи SSL шифрування даних при передачі, і використовуватимуться вони для того, щоб захистити їх від перехоплень, втручань або спроб зламу. Додатковим протоколом для забезпечення захисту від стороннього втручання, реалізований на рівні взаємодії користувача та системи буде двох-етапна автентифікація: за допомогою токенів або сертифікатів. В той же час контроль доступу до даних забезпечуватиме рольову модель доступу до них. [11]

Обрана архітектура хоч і є доволі складною в реалізації, має свої переваги. Наприклад, вона доволі гнучка й може бути адаптована до різних умов експлуатації, технологій зв'язку, баз даних тощо. Також є надійною, може забезпечити безперервний збір даних навіть при перебоях з'єднання. І найважливіше – не буде проблеми масштабування мережі, оскільки після додавання нових користувачів(пристроїв) структура не змінюватиметься.

					КВРКІ 210102.21.01.30 ПЗ	Арк. 27
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2.2.3 Реалізація алгоритму роботи мережі

Після аналізу усіх існуючих даних, було сформовано алгоритм роботи комп'ютерної мережі враховуючи визначені протоколи та архітектуру:

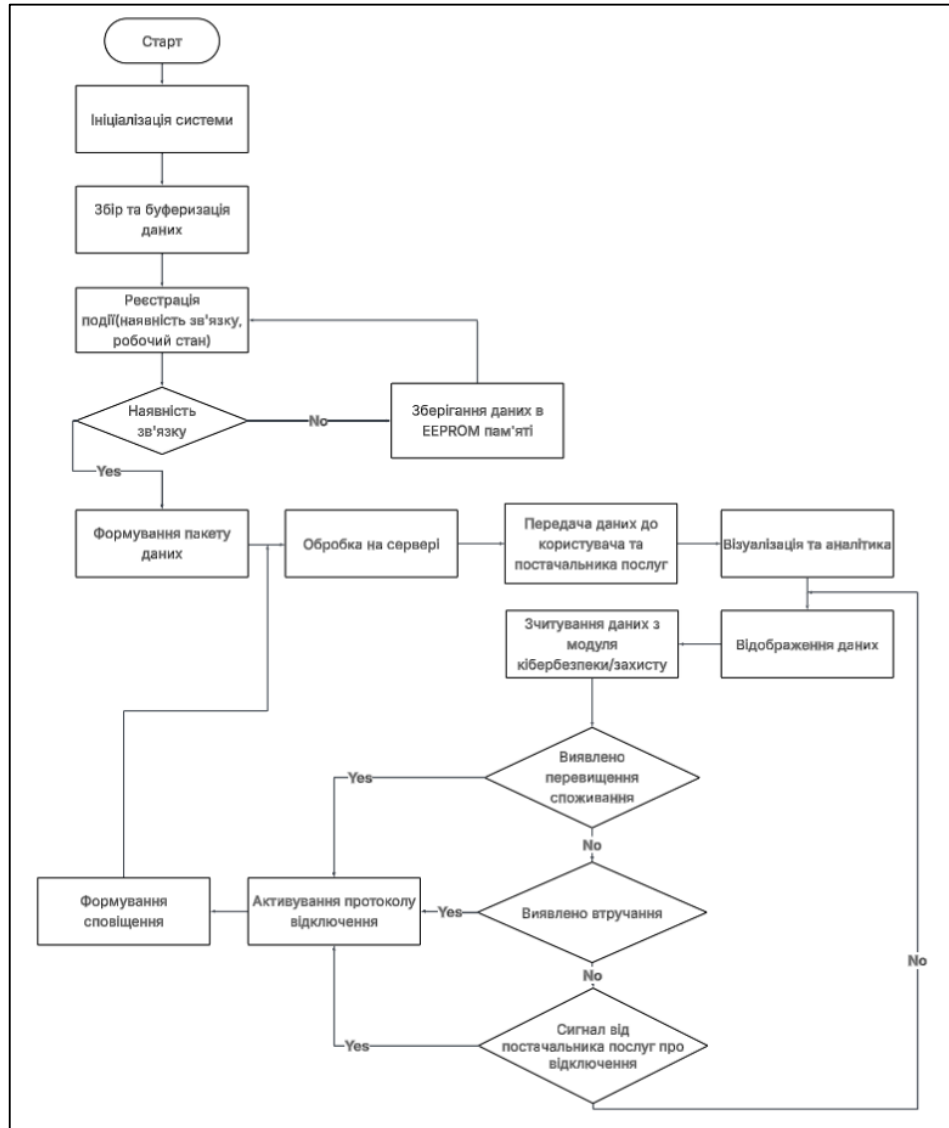


Рисунок 2.8 – Алгоритм мережі збору даних з лічильників

Алгоритм роботи мережі починається з ініціалізації системи. Починають працювати лічильники, модеми зв'язку, сервер та інші компоненти системи. Після ініціалізації, пристрої збору даних починають фіксувати рівень споживання, показники та інші дані, які можна отримати з розумного лічильника електроенергії. Система проходить два етапи перевірки на наявність подій. Перший етап –

перевірка можливості передати дані(зв'язок, робочий стан пристроїв), другий – перевірка цілісності даних, їх безпеку та перевірка на сигнали від постачальника послуг.

При збоях на першому етапі, дані будуть збережені у кеш пам'яті лічильників до відновлення роботи, після чого алгоритм перезапуститься. При збоях на другому етапі перевірки, буде запущено реле відключення, і до користувача буде надіслано сповіщення про тип події та повідомлення про відключення. Після першої перевірки у разі справної роботи зібрані дані формуються у пакет та обробляються, після чого відправляються в користувацький модуль, де ПЗ візуалізує отримані дані та робить їх зручними для сприйняття людиною. Після виконання роботи з даними, відбувається звертання до модулю безпеки, і в разі виявлення однієї з декількох подій, запускається відключення. Після відновлення доступу алгоритм перезапускається, при їх відсутності продовжує роботу в штатному режимі.

### 2.3 Вибір обладнання та програмного забезпечення

Реалізація правильно працюючої комп'ютерної мережі потребує грамотно підібраних складових апаратного та програмного забезпечення, тому цей підрозділ присвячено вибору компонентів, які будуть використані для реалізації мережі збору даних з розумних лічильників електроенергії. Обладнання повинне підтримувати обрані технології зв'язку, а також забезпечити безперервність та швидкість збору даних, їх захист, а також гнучкість та можливість масштабування.

#### **Розумні лічильники**

Лічильники для міської та сільської місцевості матимуть різні моделі, оскільки потребуватиметься підтримка різних способів передачі даних на шлюзи чи сервер. Тому, для міста та промислових об'єктів може бути використано лічильники моделі D103-600 які підтримуватимуть PRIME PLC для передачі даних через електромережу, матимуть модульну конструкцію, що дозволить в разі потреби додавати NB-IoT або GSM модеми в разі використання в інших умовах, багатотарифний облік, можливість реєстрації подій, енергоспоживання а також

					КВРКІ 210102.21.01.30 ПЗ	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		

вбудовану пам'ять. Цей лічильник буде універсальним рішенням для реалізації мережі. В межах проекту буде використаний для з'єднання із системою через PLC.



Рисунок 2.9 – Розумний лічильник D103-600 від компанії SmartMaic [4]

Для сільської місцевості або ж для точок без дротової інфраструктури доцільним буде використати моделі з вбудованим NB-IoT модемом, наприклад NIK 2307 з додатковим вбудованим модемом NB-IoT.

Зображений на рисунку 2.10 лічильник матиме можливість автономної роботи, вбудований журнал подій, а також вбудоване реле відключення.

Комунікаційні модулі:

1. SIMCom A7670 може використовуватись для зв'язку по GSM/4G;
2. Модуль Quectel BC95 або ж Quectel BC66 може використовуватись для NB-IoT;
3. STM32 з PLC-адаптером використовуватиметься для PLC. Також може використовуватись ESP32, як аналогове рішення для комунікаційного модулю.

Зм.	Арк.	№ докум.	Підпис	Дата



Рисунок 2.10 – Розумний лічильник моделі NIK 2307 [3]

### Обладнання проміжного рівня – шлюзи та концентратори даних

Вимоги до пристроїв які виконуватимуть роль шлюзу для забезпечення продуктивної роботи:

Таблиця 2.2 – технічні вимоги до шлюзу

CPU	ARM Cortex-A7 / A53
RAM	≥ 512 МБ
OS	OpenWRT або Embedded Linux
Інтерфейс	RS-485, Ethernet, USB

Для агрегації інформації з групи лічильників буде використовуватися шлюзовий пристрій, наприклад:

Arduino Module – універсальна плата, яку можна буде модифікувати під власні потреби, забезпечить урахування усіх особливостей мережі.

Raspberry Pi + модуль RS485/PLC буде підходящим та недорогим пристроєм для малих об'єктів.

Industrial IoT Gateway WISE-3610 (Advantech) підійде для промислових середовищ, адже він має високий рівень надійності.

Шлюз виконуватиме такий набір функцій як:

- a) Кешуватиме та агрегуватиме отримані дані;
- b) Зберігатиме й повторно передаватиме дані у разі зброїв або тимчасової відсутності мережевого зв'язку;
- c) Шлюз також виконуватиме первинну фільтрацію та шифрування отриманої інформації;
- d) Виконуватиме роль адаптера для деяких випадків.

#### **Обладнання верхнього рівня – серверна частина**

Для сервера напоганим рішенням буде реалізація VPS або bare-metal серверу з такими вимогами:

Таблиця 2.3 – технічні вимоги для серверу

CPU	2 ядра
RAM	4-8 ГБ
ОС	Ubuntu 22.04 LTS / GigaCloud, Hetzner, DigitalOcean
СУБД	MongoDB
MQTT-брокер	Eclipse Mosquitto
Back-end	Python (Flask), Node.js
Front-end	Grafana, Vue.js або React

Сервер може бути реалізований як:

- 1) локальний сервер на базі Linux-системи (наприклад Ubuntu Server)
- 2) віртуальний сервер (VPS) в українському або європейському дата-центрі.

Додаткові вимоги до сервера включатимуть в себе такі речі, як: підтримка баз даних(наприклад PostgreSQL), інтеграція з API та веб-інтерфейсом а також модулі шифрування та автентифікації.

Для реалізації проекту найвигіднішим буде використання саме Ubuntu Server, так як він досить зручний у використанні й не потребує додаткових онлайн сервісів для встановлення та реалізації, що буде зручно для забезпечення усіх вимог при виконанні задачі.

### **Програмне забезпечення**

На рівні лічильників і шлюзів:

- 1) Вбудоване ПЗ виробника лічильника(прошивка з підтримкою MQTT а також DLMS/COSEM);
- 2) Open-source MQTT клієнти (наприклад, Eclipse Paho для шлюзів);
- 3) NB-ІоТ SDK для кастомізації протоколів передачі.

На рівні серверу:

Таблиця 2.4 – приклади рекомендованого програмного забезпечення для організації сервера

Компонент	ПЗ
DLMS/COSEM стек	OpenDLMS/Gurux DLMS (Java, C#, Python)/libDLMS
Збір даних	Python-скрипт з Gurux Python API або ж Gurux DLMS Client
База даних	PostgreSQL з підтримкою TimeSeries (TimescaleDB) або MongoDB
Аналітика та перегляд	Grafana + PostgreSQL Data Source

## Кінець таблиці 2.4

Інтерфейс керування	Flask/FastAPI (Python)
API	Swagger/OpenAPI + FastAPI

Інтерфейс користувача може бути реалізований, як Web-інтерфейс для адміністратора (на React/Vue.js або простий HTML5) або ж як мобільний додаток.

### Безпека

Для забезпечення безпеки мережі будуть застосовані такі засоби захисту:

- 1) Шифрування TLS/SSL на рівні передавання;
- 2) Аутентифікація за API-ключами або токенами (OAuth 2.0);
- 3) Журналювання доступу та оновлення ПЗ шлюзів і серверу.

Передача даних між розумними лічильниками, шлюзами та центральним сервером буде здійснюватись із застосуванням криптографічного шифрування TLS/SSL (Transport Layer Security / Secure Sockets Layer). Цей рівень захисту гарантуватиме, що всі наявні телеметричні пакети, що надходять від пристроїв через канали зв'язку будуть захищені від перехоплення та спроб дешифрування даних сторонніми особами. Таке шифрування буде охоплювати звичайні дані виміру а також службові повідомлення (наприклад, це можуть бути такі команди як ті, що використовуватимуться для підтвердження подій або ж для вимкнення).

На рівні ідентифікації та контролю доступу може бути реалізовано механізм автентифікації користувачів та пристроїв за допомогою таких речей як, до прикладу API ключі, або ж авторизаційні токени, відповідно до стандарту OAuth 2.0. Цей підхід допоможе призначити унікальні параметри доступу для кожного користувача або ж підсистеми, обмежити їхні дії згідно з правами доступу, а також оперативно відкликати або змінювати токени в разі підозри на стороннє втручання.

Не менш важливою складовою захисту мережі буде журналювання доступу. Це означатиме що всі виконані всередині мережі дії, такі як запити до центрального

					КВРКІ 210102.21.01.30 ПЗ	Арк. 34
Зм.	Арк.	№ докум.	Підпис	Дата		

сервера, шлюзів, тощо, буде записано в логи з вкачання таких параметрів як час, сесія, результат операції а також джерело, з якого було складено лог. Журналювання дозволить проводити перевірку дій в середині системи та виявляти підозрілу активність або ж спроби втручання.

Це не кінцеві рішення щодо обладнання та ПЗ, в даному підрозділі надані найбільш оптимальні варіанти, надалі з яких буде обрано підходящі для реалізації проекту.

#### 2.4 Висновки до другого розділу

У межах розділу 2 було обрано та обґрунтовано технічні рішення для створення мережі збору даних з розумних лічильників електроенергії, спроектовано схему роботи усієї мережі.

Описано основні підсистеми ПТЗ: апаратну (збір, буферизація, передача даних), програмну (обробка, зберігання, візуалізація, інтеграція), модуль кібербезпеки, а також реалізовано логічну взаємодію між ними. Запропонована мережева архітектура підтримує протокол DLMS/COSEM, забезпечує сумісність з існуючими інформаційними системами та гарантує цілісність і безпечність передачі інформації.

Було проаналізовано та обрано припустиме обладнання для забезпечення роботи мережі, підібрані протоколи для кіберзахисту та зв'язку. Створено та описано схематичні зображення роботи складових мережі.

На основі порівняльного аналізу було обґрунтовано використання двох основних типів лічильників: NIK 2307 NB-IoT для прямих підключень та D103-600 PLC для підключень через шлюзи. Було забезпечено мінімізацію кількості проміжного обладнання, без втрати надійності передачі даних.

					КВРКІ 210102.21.01.30 ПЗ	Арк. 35
Зм.	Арк.	№ докум.	Підпис	Дата		

### **3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ МЕРЕЖІ ЗБОРУ ДАНИХ З РОЗУМНИХ ЛІЧИЛЬНИКІВ ЕЛЕКТРОЕНЕРГІЇ**

3.1 Опис реалізації модулів апаратного та програмного забезпечення програмно-технічного засобу

У межах цього підрозділу подано опис реалізації основних модулів апаратного та програмного забезпечення, які утворюють програмно-технічний засіб комп'ютерної мережі збору даних з розумних лічильників електроенергії. З урахуванням відсутності обладнання, практична реалізація здійснена у вигляді функціонального моделювання, що дозволить імітувати ключові процеси збору, передачі, обробки, представлення даних. Також, при цьому зберігається логіка взаємодії компонентів та їх архітектурна відповідність.

Апаратна частина моделюється з урахуванням конфігурацій реальних пристроїв, а саме: розумні лічильники моделей NIK 2307 з підтримкою NB-IoT та D103-600 з каналом зв'язку PRIME PLC. Під час моделювання мережі передбачається, що ці пристрої мають в собі вбудовані вимірювальні модулі, внутрішню пам'ять EEPROM для буферизації показників та засоби зв'язку для відправлення даних на центральний сервер. У випадку з PLC-зв'язком передбачається використання шлюзового пристрою, який виконує роль адаптера між енергетичною мережею та IP-мережею. Цей шлюз також забезпечує базову маршрутизацію та шифрування трафіку. У варіантах із зв'язком по NB-IoT – він буде здійснюватися безпосередньо між лічильником і сервером через мобільну мережу оператора або в локальній мережі (LAN).

Програмна частина представлена окремими модулями, що відповідають за прийом, обробку, зберігання і подальшу інтерпретацію даних. Сервер збору даних моделюється у вигляді умовного програмного середовища, яке приймає пакети з показниками у форматі, сумісному з протоколом DLMS/COSEM. У рамках імітації на рівні прикладного коду буде реалізовано логіку декодування

структурованих телеметричних повідомлень, їх зберігання у документно-орієнтованій базі даних типу MongoDB з часовою розміткою, а також логіку виведення показників через веб-інтерфейс. Інтерфейс адміністрування, написаний з використанням інструментів Node.js або NodeJS, дозволяє відображати ключові параметри у зручному для оператора вигляді. У моделі також передбачено наявність веб-сокетів, завдяки яким дані можуть бути експортовані до зовнішніх інформаційних систем таких як аналітичні або також системи звітності в режимі реального часу.

### 3.2 Схема апаратного забезпечення та опис його елементів

Дана система забезпечує функції збору, буферизації, передачі даних та захисту інформації, маючи в основі реальні апаратні компоненти або ж їх еквіваленти.

Згідно з розробленою в розділі архітектурою, структура апаратної частини мережі базується на двох основних моделях лічильників: NIK 2307 що підтримує NB-IoT та D103-600 зі зв'язком через PLC PRIME. Обидва пристрої мають вбудовані вимірювальні блоки, модулі обліку та EEPROM для зберігання даних.

Лічильники моделі NIK 2307 має інтегрований NB-IoT модем, який забезпечує безпосередній вихід у мережу операторів стандарту LTE Cat NB1/NB2 та LAN 2.5-5.0. У такому випадку апаратна реалізація не потребує додаткового шлюзу, оскільки сам лічильник виступає джерелом даних та передавачем. Основні апаратні інтерфейси, які реалізуються на платі пристрою — це UART для локальної діагностики, SPI або I2C для зчитування з внутрішніх датчиків, а також живлення від електромережі.

У лічильниках D103-600 передача даних здійснюється по електромережі за технологією PLC. Через те, що PLC не забезпечує прямого виходу в IP-мережу, дані агрегуються на проміжному вузлі - шлюзі, його еквівалентом по функціональності є мікрокомп'ютер поделі Raspberry Pi або вбудована система на

					КВРКІ 210102.21.01.30 ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

базі універсальної плати Arduino. Шлюз забезпечує приймання даних через інтерфейс RS-485 або через PLC-модем, а далі виконує їх передавання по Ethernet або 4G-каналу до центрального сервера. На апаратному рівні шлюз включає контролер зв'язку, модем, інтерфейси USB/UART та схеми захисту по живленню і лініях передачі.

Для побудови макетної моделі системи можна використовувати мікроконтролер ESP32, підключений до NB-IoT модему (наприклад, Quectel BC66) через UART. У якості живлення використовується стабілізований блок живлення 5В або 9В із захистом по струму IP 65, а дані можуть передаватися на серверну частину по TCP або через емулятор DLMS-пакетів(в якості емулятора можна використати роутер)

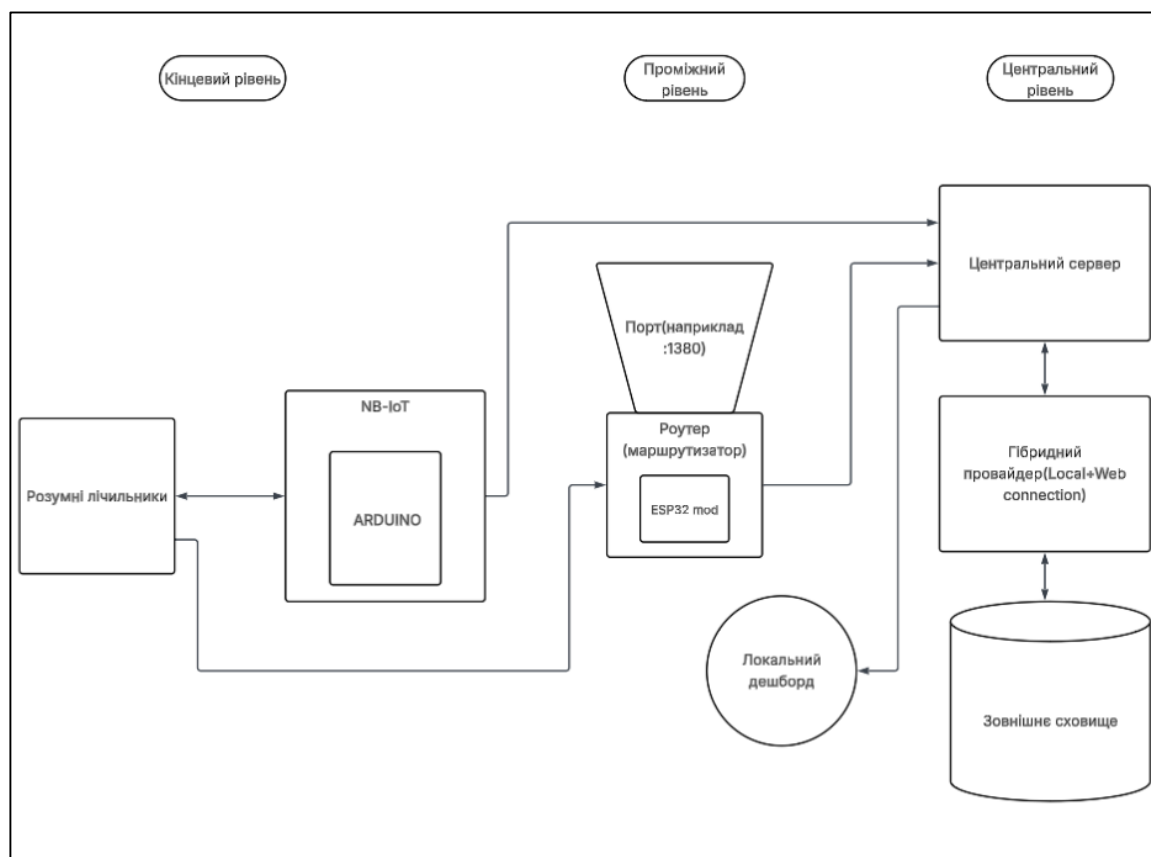


Рисунок 3.1 – Схематичне зображення програмно-апаратної частини системи

На рисунку 3.1 було наведено демонстративний варіант апаратної частини системи, вона демонструє розміщення основних вузлів, їхні інтерфейси взаємодії

Зм.	Арк.	№ докум.	Підпис	Дата

та канали зв'язку між ними. Така схема буде використана для подальшого моделювання прототипу мережі.

### 3.3 Апаратне забезпечення мережі збору даних з лічильників електроенергії

Система включатиме в себе моделювання невеликої локальної мережі збору даних з лічильників електроенергії, щоб демонструвати її принцип роботи із заданими умовами. Підберемо робочі компоненти та складові даної мережі для подальшого моделювання схеми та створення електричної схеми. Після визначення конкретних елементів та складових системи буде сформована принципова електрична схема зв'язку між лічильниками, шлюзом, мікроконтроллером та сервером.

Почнемо з кінцевого рівня мережевої архітектури: розумних лічильників електроенергії.

До моделі мережі підключаємо один лічильник моделі NIK 2307 по технології зв'язку NB-IoT та два лічильника D103-600 по мережі PRIME PLC. Лічильники моделі NIK 2307 підключатимуться безпосередньо до серверу по NB-IoT(Cat-NB1/Cat-NB2), отримуватиме живлення від мережі 220V з допомогою AC/DC адаптеру 220V => 9V/2A.

Лічильники моделі D103-600 підключатимуться до адаптерного шлюзу по протоколу PRIME PLC(а саме IEC 61334-5-1), отримуватимуть живлення від тієї ж мережі та адаптерів, додатково також реально локально організувати відстань 5-10 метрів для демонстрації роботи передачі даних з допомогою електричних кабелів PLC.

Лічильники вимірюватимуть дані, що надходять по трьохфазному каналу, після чого передаватимуть дані виміру на мікроконтроллер, який забезпечить подальшу роботу. Мікроконтроллер перетворює дані й надає можливість передавати їх комунікаційному модулю. У випадку з лічильниками D103-600 вони підключатимуться до лінії електропередачі, а NIK 2307 до NB-IoT через

					КвРКІ 210102.21.01.30 ПЗ	Арк. 39
Зм.	Арк.	№ докум.	Підпис	Дата		

вбудований у лічильник модем. Функційна схема роботи лічильника зображена на рисунку 3.2

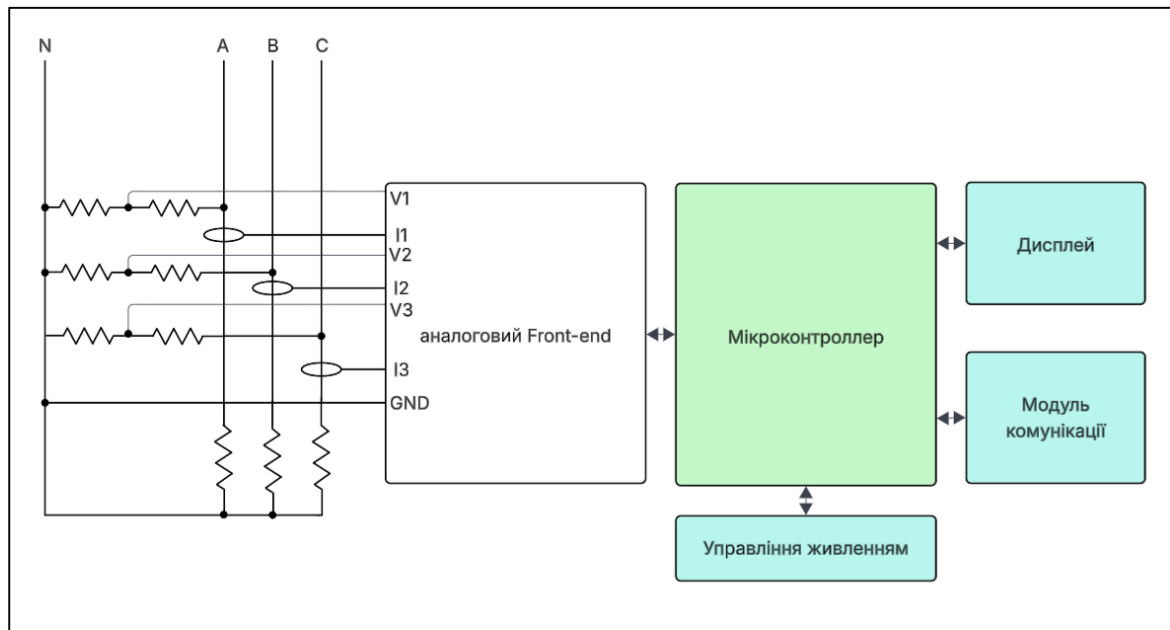


Рисунок 3.2 – Схема роботи лічильника

Реалізація NB-IoT-зв'язку базується на використанні спеціалізованого стільникового модему, зокрема Quectel BC66, який підключається до основного контролера ESP32 через UART-інтерфейс.

Передача даних у NB-IoT відбувається з використанням стека протоколів TCP/IP, який реалізовано на рівні модема. Керування модемом здійснюється через AT-команди відповідно до стандарту 3GPP. У програмній частині мікроконтролера реалізовано алгоритм ініціалізації модему, встановлення з'єднання з мобільною мережею, налаштування контексту PDP та безпосередньої передачі телеметричних даних у вигляді JSON або HEX-пакетів на IP-адресу сервера. Дані проходять через захищений TLS-канал, що гарантує їх конфіденційність та цілісність у процесі передавання.

Для підтримки енергозбереження в моделі реалізовано програмну емуляцію режиму PSM (Power Saving Mode з англійської – режим збереження енергії), в якому модем переходить у стан низького енергоспоживання між сесіями передачі.

Також враховано затримку повторної реєстрації у мережі(вона є типовою для NB-IoT), та оптимізовано періодичність надсилання даних відповідно до логіки обліку наприклад, передача кожні 30 хвилин або при зміні показника більше ніж на визначений поріг.

З боку апаратного забезпечення, модем NB-IoT живиться від стабілізованої лінії 3.3 В із допустимим піковим струмом до 250 мА під час передачі. У схемі передбачено конденсатори-фільтри ємністю 10  $\mu$ F та 100 nF на вході живлення, а також RC-фільтр і TVS-діод для захисту UART-ліній. Модем підключається до ESP32 за допомогою 4-пінового UART-інтерфейсу, де два провідники використовуються для обміну даними, а ще два для керування станами (наприклад, вивід PWRKEY або DTR).

Надалі переходимо до характеристики PLC шлюзу, з яким з'єднуватимуться лічильники D103-600. MCU шлюзу використовуватиме мікроконтролер ESP-WROOM-32 (наприклад візьмемо модель Xtensa LX6 Dual-Core, яка підтримує UART/SPI/I2C, Wi-Fi або Bluetooth). Технічні характеристики мікроконтролера: 240 MHz, 520 KB оперативна пам'ять, 4 MB флеш-пам'ять. Мікроконтролер забезпечує компактність, достатній об'єм оперативної пам'яті, вбудовані інтерфейси UART та SPI, а також можливість підключення до NB-IoT-модема через AT-команди. Також згідно рекомендацій виробника мікроконтролера, до схеми буде додано RC-фільтр на вході живлення, байпасні конденсатори на лініях живлення, а також TVS-діоди на зовнішніх лініях для захисту GPIO. UART0 ESP32 використовується для зв'язку з модемом Quectel BC66, а UART1 — для керування інтерфейсом RS-485 через MAX485.

Далі підбираємо підходящий PRIME PLC контролер для модуля шлюзу, підходящим варіантом є модель Atmel ATPL250A. ATPL250A відповідає за обмін даними з лічильниками моделі D103-600 через двопровідну лінію PLC, використовуючи модуляцію OFDM(у діапазоні 3–148 кГц). ATPL250A розроблений для використання у зв'язці з зовнішнім мікроконтролером (MCU). Зовнішній мікроконтролер використовує бібліотеку PHY-рівня G3, надану



Останній, у свою чергу, здійснює подальшу маршрутизацію пакетів у напрямку центрального сервера або вивід у системний журнал.

Окрему увагу в реалізації приділено структурі шифрування та цілісності даних. Оскільки шифрування реалізовано на каналному рівні за допомогою AES-128, основна обробка ключів виконується апаратно, це дозволяє знизити навантаження на контролер. Цей підхід підвищує стійкість до атак і є критично важливим при роботі в умовах відкритого енергетичного середовища.

Для LTE модему обираємо модель Quectel BC66(Cat 4 LTE), забезпечуємо з'єднання по мережі Ethernet використавши 10/100 Mbps контролер, для забезпечення стабільного живлення використаємо 5-9В DC блок, передбачуване споживання: до 15В. Захист шлюзу відповідатиме рівню захисту IP65 (пилзахист + захист від водяних струменів). Передбачений температурний діапазон шлюзу: -20°C до +60°C.

Інтерфейси шлюзу: PRIME PLC інтерфейс для підключення, D103-600 Ethernet порт (RJ45), LTE антенний роз'єм (SMA), живлення 5-9В DC LED, індикатори стану а також кнопка Reset

Специфікація PRIME PLC включає в себе: стандарт ITU-T G.9904 (PRIME v1.4), частотний діапазон від 3-х до 148.5-ти кГц, модуляція типу OFDM, максимальна кількість вузлів становить до 1023-х. Топологічна структура специфікації: дерево з автоматичним маршрутизуванням.

LTE модуль включатиме в себе: стандарт LTE Cat 4, швидкість до 150 Мбіт/с (DL) або 50 Мбіт/с (UL), підтримку частотних діапазонів українських операторів а також зовнішню всеспрямовану антену 5dBi.

1. Живлення лічильників:

- a) Джерело: мережа 220В AC
- b) Споживання NIK 2307:  $\leq 10$  ВА
- c) Споживання D103-600:  $\leq 8$  ВА кожен

2. Живлення PLC-шлюзу:

- a) Вхідна напруга: 5-9В DC

					КВРКІ 210102.21.01.30 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		



На схемі було зображено основні підключення контролера та мікроконтролера між собою і до мережі, перетворювачі напруги, підключення пінів та заземлення.

3.4 Реалізація програмної частини централізованої системи. Організація серверної частини передачі даних між розумними лічильниками та центральним сервером

Щоб обробляти та зберігати отримані дані з розумних лічильників зазвичай використовують мініатюрне локальне сховище. Для даного випадку, щоб мати повний контроль над отриманою інформацією буде використано локальний VPS(з англ. - віртуальний приватний сервер). В якості операційної системи, було використано Linux-дистрибутив, котрий призначений для серверних задач, а саме Ubuntu Server версії 20.04 LTS. Дана версія забезпечує стабільну роботу програм, а також команд, без зайвого навантаження на саму систему в цілому, без додаткових налаштувань чи встановлення внутрішніх обмежень на ОС. Окрім інсталюваної системи для отримання і зберігання даних ми будемо використовувати самописний отримувач на базі NodeJS та бібліотеки node:ws. Спираючись на назву бібліотеки можемо дійти висновку, що буде використовуватись універсальний спосіб комунікації - WebSockets (надалі - веб-сокети). Основний принцип роботи веб-сокетів полягає у наступному:

Таблиця 3.1 – порівняння WebSockets та HTTP

Критерій	HTTP (1.1 / 2.0)	WebSockets
Тип зв'язку	Однонаправлений (клієнт → сервер)	Двосторонній (клієнт ⇔ сервер)

Продовження таблиці 3.1

<b>З'єднання</b>	Нове з'єднання на кожен запит (HTTP 1.1) або мультиплекс (HTTP/2)	Постійне з'єднання після handshake
<b>Протокол</b>	http:// або https://	ws:// або wss://
<b>Початкова ініціалізація</b>	Стандартний запит (GET, POST і т.д.)	Починається з HTTP-запиту, потім "upgrades" до WS
<b>Передача даних</b>	Текст (HTML, JSON, XML тощо)	Будь-що: текст, JSON, бінарні дані (ArrayBuffer, Blob), тощо
<b>Формат передачі</b>	Повні HTTP-пакети	Без заголовків HTTP
<b>Затримка (latency)</b>	Вища, через постійне відкриття-закриття з'єднань	Низька, адже з'єднання підтримується постійно
<b>Передача в реальному часі</b>	Складно, треба елементи опитування (інша назва - polling) або SSE	Ідеально для real-time: миттєві push/receive
<b>Підтримка браузерами</b>	Універсальна	Підтримується всіма сучасними браузерами
<b>Обсяг трафіку</b>	Більше (багато хедерів, часто дублюються)	Менше (мінімальні заголовки)

Кінець таблиці 3.1

Підтримка стандартів	REST, GraphQL поверх HTTP	Потрібно писати власні протоколи обміну
Безпека	TLS (https)	TLS (wss)
Масштабованість	Легко скейлиться за допомогою звичних HTTP проху	Потребує специфічної інфраструктури (наприклад, WS-aware проксі)
Сценарії використання	CRUD, API, сайти, завантаження сторінок, REST	Чати, онлайн-ігри, біржі, трекери, push-нотифікації

Як бачимо - WebSockets для даного завдання підходить більше, аніж прямі http з'єднання і ось чому:

1) Розумні лічильники отримують та відправляють метрики кожен секунду або в момент змін показників, що на даному етапі вимагає нам постійної передачі інформації до бази даних;

2) На момент коли ми повинні відображати результати метрик в реальному часі, прямі http з'єднання не забезпечує нам постійної та активної передачі даних;

3) Вебсокети працюють за принципом телефонної розмови, що не дає можливостей прослоховувати комунікації між клієнтом та сервером, даючи безпеку при комунікації;

Задля тимчасового зберігання(далі - кешування) метрик ми використовуємо документно-орієнтовану базу даних(далі - БД) MongoDB. Розглянемо відмінності MongoDB від реляційних баз даних(друга назва - SQL Databases):



Кінець таблиці 3.2

<b>Гнучкість</b>	Можна зберігати напівструктуровані дані без змін схеми	Потрібно заздалегідь проектувати структуру
<b>Кейс-використання</b>	IoT, real-time data, швидкі MVP, документообіг	ERP, банкінг, бухгалтерія, складні зв'язки, ACID-критично
<b>Тип зберігання</b>	BSON (розширений JSON)	Рядки та типізовані стовпці
<b>Підтримка ACID</b>	Часткова / повна з 4.0+, але складніше конфігурувати	Повна підтримка ACID від початку
<b>Інструменти аналітики</b>	MongoDB Aggregation Framework	Потужні SQL-запити + CTE, Window functions, функції
<b>Легкість старту</b>	Швидко підняти, не треба морочитись зі схемами	Потрібна чітка структура і планування
<b>Крива навчання</b>	Простий старт, але агрегації мають свою складність	Вивчення SQL бази - обов'язковий для багатьох фахів
<b>Вбудовані механізми безпеки</b>	Хороші, але конфігуруються окремо	Зазвичай мають детальну політику доступу та ролі
<b>Приклади</b>	MongoDB, Couchbase, Firebase Firestore	PostgreSQL, MySQL, MariaDB, SQLite

Звідси робимо висновок, що MongoDB забезпечує якісну організацію обробки інформації, а також ідеально підходить для зберігання метрик чи організації елементів розумного будинку або інших елементів IoT, на відміну від SQL баз даних, завдяки простій та динамічній структурі таблицю, даючи нам гнучкість і простоту у масштабуванні.

Після вибору усіх потрібних елементів, далі ми організуємо основні алгоритми роботи ключових системи, а саме: Сервер, Клієнт та Веб інтерфейс,

розглянемо кожен із них в деталях. Для роботи сервера із використанням WebSocket-а та бази даних MongoDB алгоритм буде виглядати наступним чином:

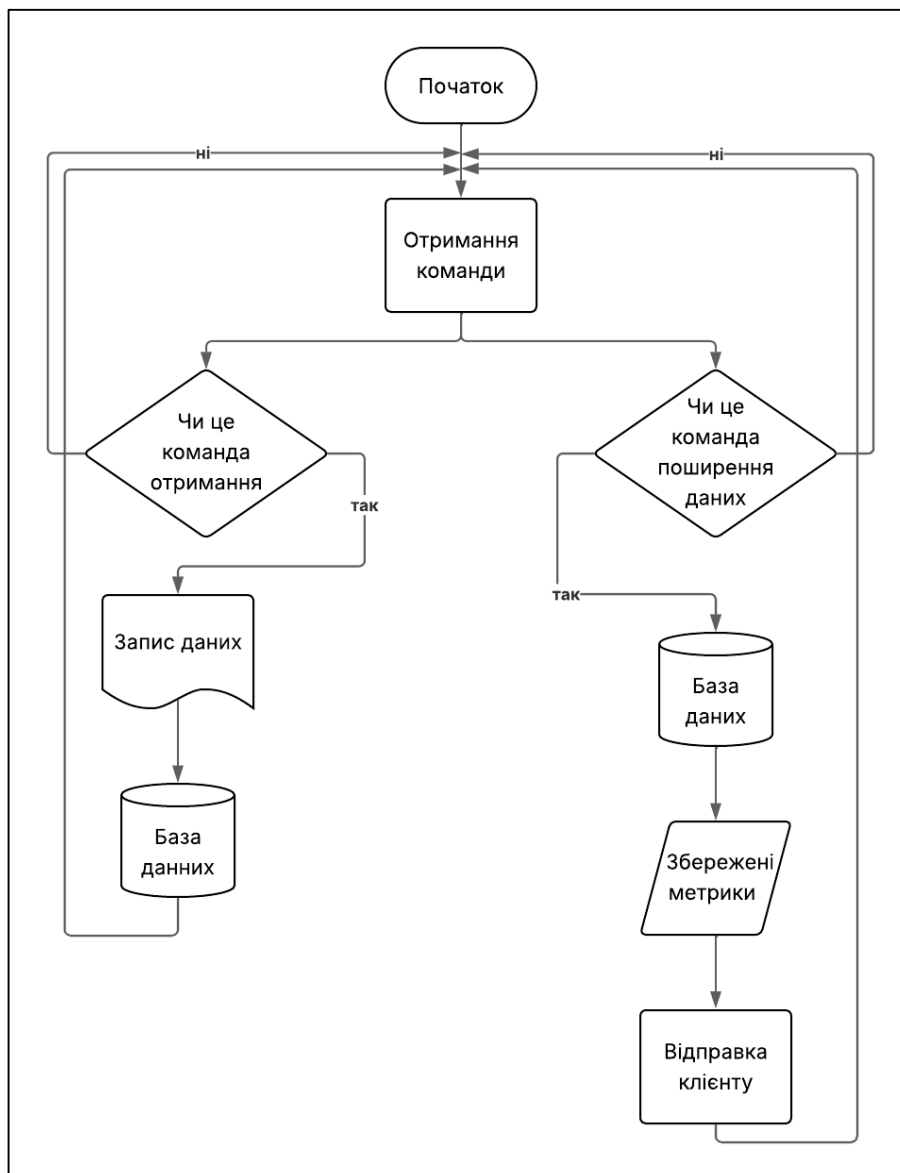


Рисунок 3.5 – Алгоритм роботи сервера з MongoDB та WebSockets

Як можемо помітити, сервер має доволі просту логіку роботи, котра дозволяє нам безперервно обробляти запити в незалежності який формат зв'язку відбувається між ним та клієнтом. У нього є 2 основні команди, котрі забезпечують нам роботу - отримання даних та поширення поточних записів метрик. У разі якщо команди ніякі не було відправлено - сервер працює і надалі без зупинки роботи та закриття зв'язку. Команда прийняття даних відправляє метри до бази даних у той

час як команда поширення бере записи і відправляє клієнту. Сам клієнт працює інакше:

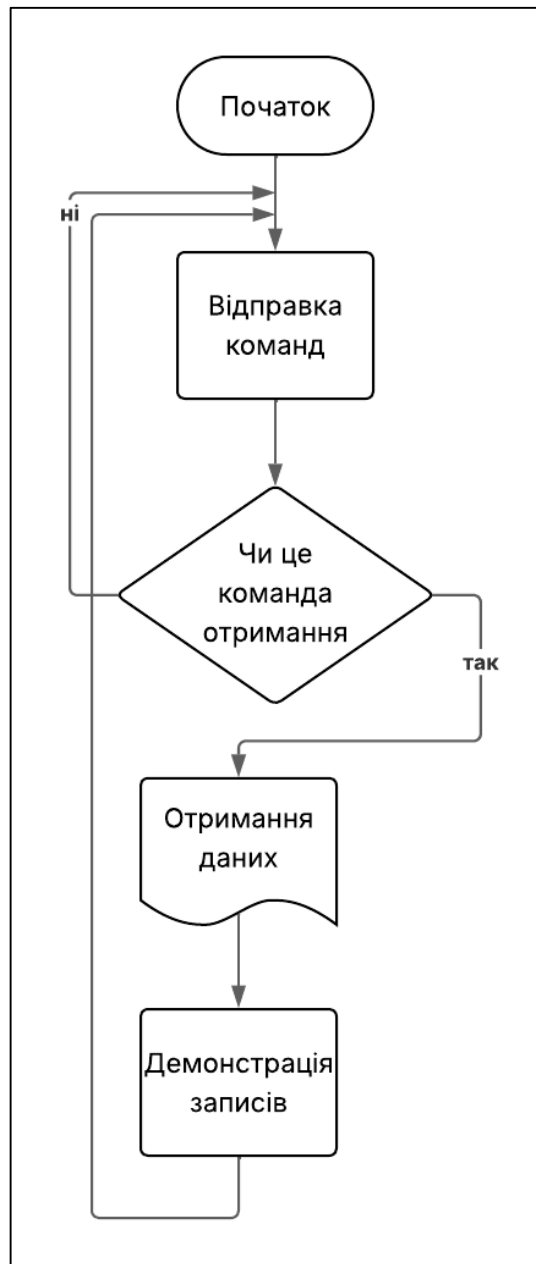


Рисунок 3.6 – Алгоритм роботи клієнта

Клієнт має виключно одну команду, а саме «Отримання записів з серверу». Ця команда слугує нам для того, щоби веб-сервіси, котрі будуть відображати дані змогли без проблем мати актуальну інформацію кожні 3 секунди на стороні веб клієнта. Щодо веб клієнта, він працює виключно як клієнт-запрошувач:

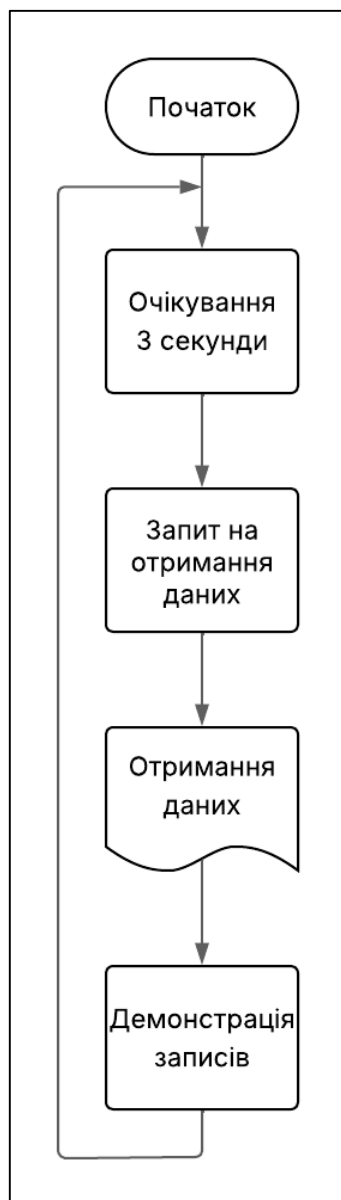


Рисунок 3.7 – Алгоритм роботи веб клієнта

Наступним кроком є написання кодової частини центрального серверу. Для цього використаємо мову програмування JavaScript та допоміжну бібліотеку, котра забезпечує додаткову функціональність - express.js. Код серверної частини виглядає наступним чином:

```

import express from 'express'
import { WebSocketServer } from 'ws'
import mongoose from 'mongoose'
import http from 'http'

//Сетап бази даних
await mongoose.connect('mongodb+srv://<username>:<password>@cluster0.mongodb.net/meters?retryWrites=true&w=majority')
console.log('Connected to MongoDB')

//База даних
const meterSchema = new mongoose.Schema({
  meterId: String,
  timestamp: Date,
  value: Number,
})

const MeterData = mongoose.model('MeterData', meterSchema)

const app = express()
const server = http.createServer(app)
const wss = new WebSocketServer({ server })

//Вебсокети
wss.on('connection', (ws) => {
  console.log('New client connected')

  ws.on('message', async (message) => {
    try {
      const parsed = JSON.parse(message)

      if (parsed.type === 'meter_data') {
        const { meterId, value } = parsed.data

        await MeterData.create({
          meterId,
          value,
          timestamp: new Date(),
        })
      }
    }
  })
})

```

Рисунок 3.9 – Підключення та ініціалізація бази даних та сокету, обробка команди запису інформації

```

ws.send(JSON.stringify({ type: 'status', status: 'saved' }))
}

if (parsed.type === 'get_data') {
  const data = await MeterData.find({ meterId: parsed.meterId }).sort({ timestamp: -1 }).limit(50)
  ws.send(JSON.stringify({ type: 'data_response', data }))
}

} catch (err) {
  console.error('Error handling message:', err)
  ws.send(JSON.stringify({ type: 'error', message: 'Invalid format or server error' }))
}
})

ws.on('close', () => {
  console.log('Client disconnected')
})

//Старт серверу
const PORT = process.env.PORT || 3000
server.listen(PORT, () => {
  console.log(`WebSocket server running on port ${PORT}`)
})

```

Рисунок 3.10 – Обробка команди відправлення записів клієнту

Також паралельно з цим відбувається перехоплення помилок та закриття каналу зв'язку на стороні клієнта.

Даний сервер завжди отримує команду прийняття метрик, записування їх у базу даних (документ), а згодом, через команду відправки, сервер надає всі записи клієнту, котрий їх і запросив. Сервер працює на порті 3000, а для того щоб веб-сокет працював належно ми обгортаємо його у HTTP модуль завдяки бібліотеки `express.js`, щоб була відповідність в заголовках запиту для протоколу "ws". Для клієнта реалізовано окремі http запити, задля забезпечення отримання метрик для веб-інтерфейсу у разі, якщо клієнтська частина коду не може працювати. Наступним кроком - написання скрипту, котрий буде працювати, як основне джерело отримання інформації. Код виглядає наступним чином:

```
import axios from 'axios'

const SERVER_URL = 'http://localhost:3000'
const meterId = 'METER_001'

async function fetchData() {
  try {
    const response = await axios.get(`${SERVER_URL}/api/meter/${meterId}`)
    console.table(response.data.map(x => ({
      Time: new Date(x.timestamp).toLocaleString(),
      Value: x.value
    })))
  } catch (e) {
    console.error('HTTP error:', e.message)
  }
}

// Можна викликати вручну або раз на N секунд:
fetchData()

// setInterval(fetchData, 10000) // <- автозапит кожні 10 сек
```

Рисунок 3.11 – Загальний код клієнта

Щоби відображати дані не через термінал, а мати чітку візуалізацію та аналізувати попередні та теперішні показники, реалізуємо веб-інтерфейс. Інтерфейс буде складатися з однієї сторінки, буде доречно використати мову гіпертекстової розмітки HTML та інтеграцію скриптової мови JavaScript всередині

самої HTML сторінки. Ключові елементи логіки будуть виглядати наступним чином:

```
//Підключення до WebSocket сервера
connectWebSocket() {
  try {
    this.websocket = new WebSocket(this.serverUrl);

    this.websocket.onopen = () => {
      console.log('WebSocket з\'єднання встановлено');
      this.isConnected = true;
      this.updateConnectionStatus();
      this.requestInitialData();
    };

    this.websocket.onmessage = (event) => {
      try {
        const message = JSON.parse(event.data);
        this.handleServerMessage(message);
      } catch (error) {
        console.error('Помилка при обробці повідомлення:', error);
      }
    };

    this.websocket.onclose = () => {
      console.log('WebSocket з\'єднання закрито');
      this.isConnected = false;
      this.updateConnectionStatus();
      //Спроба повторного підключення через 5 секунд
      setTimeout(() => this.connectWebSocket(), 5000);
    };

    this.websocket.onerror = (error) => {
      console.error('WebSocket помилка:', error);
      this.isConnected = false;
      this.updateConnectionStatus();
    };
  } catch (error) {
    console.error('Не вдалося підключитися до WebSocket:', error);
    this.fallbackToSimulation();
  }
}
```

Рисунок 3.12 – Підключення до серверу та обробка подій веб-сокета та обробка помилок на стороні клієнта та серверу

Під час підключення до серверу клієнт запрошує дані, щоб впевнитись, що з'єднання встановлено і все працює коректно. У разі невдалого чи провального підключення інтерфейс подає запит на симуляцію даних допоки не відновиться підключення до мережі. У разі, якщо інтерфейс зміг доєднатися до серверу, діють

наступні мікро-алгоритми. Для того щоб кожні 30 секунд система могла отримати дані, виконується запуск функції `handleServerMessage`. Дана функція виконує роль перехоплювача, котрий в режимі реального часу отримує дані з серверу і надає можливість іншим алгоритмам демонструвати отримані дані. Загальний код функції виглядає так:

```
//Обробка повідомлень від сервера
handleServerMessage (message) {
  if (message.type === 'data_response' && message.data) {
    message.data.forEach(record => {
      const meterType = this.meterMapping[record.meterId];
      if (meterType && this.data.hasOwnProperty(meterType)) {
        this.data[meterType] = record.value;

        //Оновлення графіка для нових даних
        this.updateChartWithNewData(meterType, record.value, record.timestamp);
      }
    });

    this.updateMetrics();
    this.updateLastUpdated();
  }
}
```

Рисунок 3.13 – Отримання записів з серверу

Допоміжним елементом є функція під назвою `setupDataRequest`. Вона забезпечує безперебійне отримання даних із серверу і розподіляє їх між елементами інтерфейсу.

```
//Налаштування періодичних запитів даних
setupDataRequest() {
  setInterval(() => {
    if (this.isConnected && this.websocket && this.websocket.readyState === WebSocket.OPEN) {
      const request = {
        type: 'data_request',
        meterIds: Object.keys(this.meterMapping),
        timeRange: {
          start: new Date(Date.now() - 3 * 1000).toISOString(),
          end: new Date().toISOString()
        }
      };
      this.websocket.send(JSON.stringify(request));
    }
  }, 3000);
}
```

Рисунок 3.14 – Періодична обробка запитів

Для демонстрації основного вигляду системи, запрошуються початкові дані. Інформація фільтрується за часовими параметрами, відправляється і формується у масив даних, котрі конвертуються у загальний вигляд. Код функції `requestInitialData` виглядає наступним чином:

```
//Запит початкових даних
requestInitialData() {
  if (this.websocket && this.websocket.readyState === WebSocket.OPEN) {
    const request = {
      type: 'data_request',
      meterIds: Object.keys(this.meterMapping),
      timeRange: {
        start: new Date(Date.now() - 24 * 60 * 60 * 1000).toISOString(),
        end: new Date().toISOString()
      }
    };
    this.websocket.send(JSON.stringify(request));
  }
}
```

Рисунок 3.15 – Запит на початкові дані

Після обробки усіх отриманих даних ініціалізується запуск дашборду, який буде використовуватись, як веб-інтерфейс для зчитування оброблених даних користувачем.

```
// Запуск дашборду
document.addEventListener('DOMContentLoaded', () => {
  new SmartMetersDashboard();
});
```

Рисунок 3.16 – Запуск веб-інтерфейсу

Дашборд з'єднується напряму із сервером, отримуючи дані, після чого запускає генерацію графіку споживання електроенергії лічильником, базуючись на отриманій інформації.



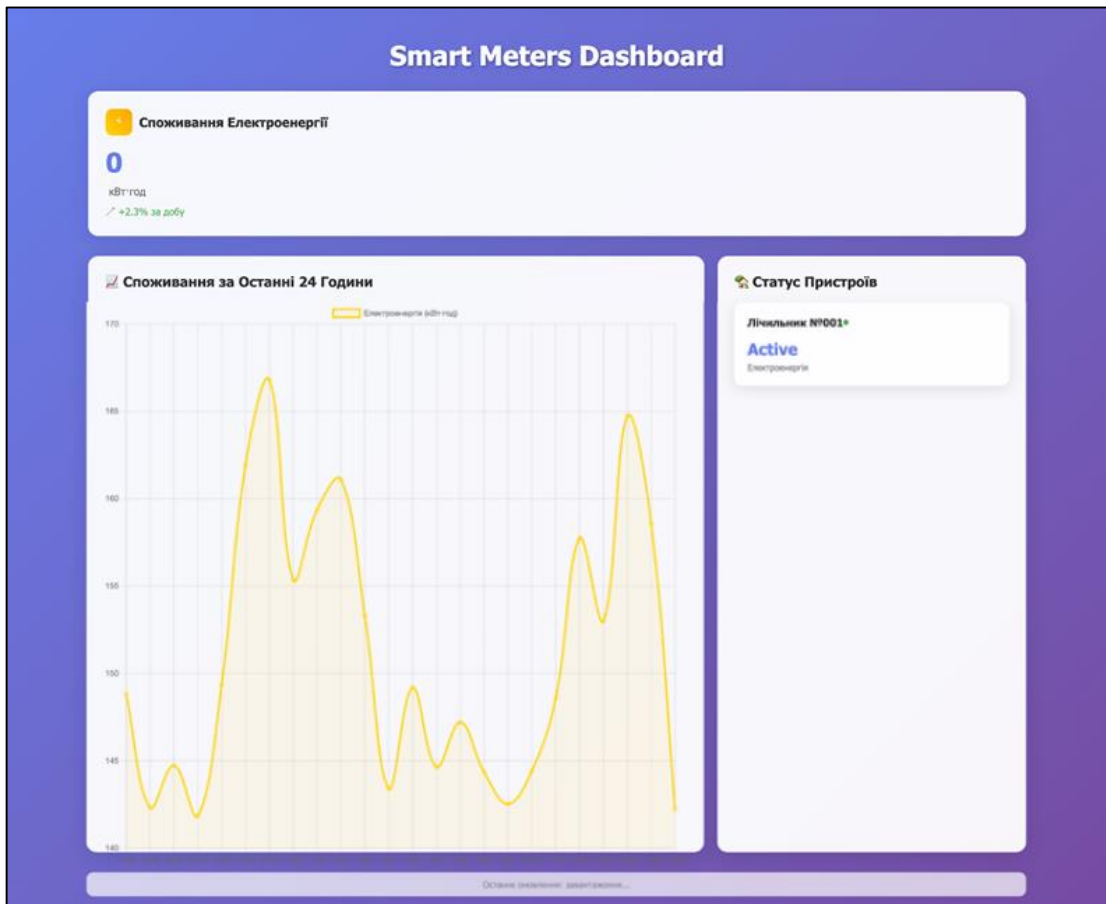


Рисунок 3.17 – загальний вигляд веб-інтерфейсу

Реалізувавши усі заплановані програмні та апаратні частини, маємо схему проекту мережі збору даних розумних лічильників електроенергії:

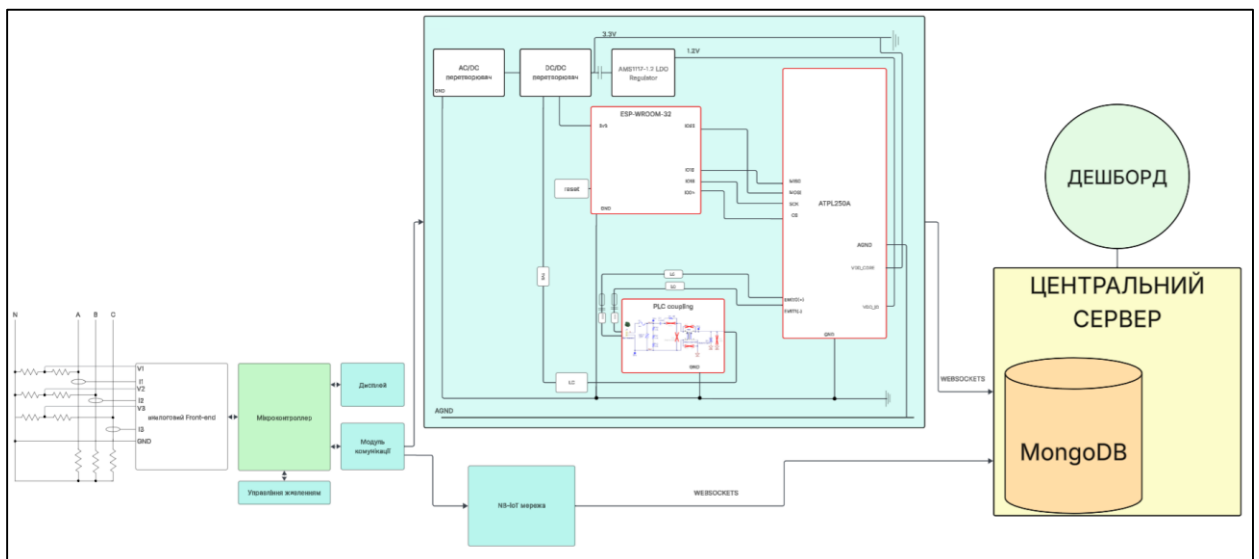


Рисунок 3.18 – Загальна схема проекту

На схемі зображені усі архітектурні рівні, які було змодельовано під час роботи над проектом, показана їх взаємодія та показане схематичне зображення підключення апаратної частини в лічильнику та більш детально в PLC шлюзі, який має забезпечити роботу передачі даних через кабелі.

### 3.5 Висновки до третього розділу

В третьому розділі було фіналізовано підбір апаратних та програмних компонентів для забезпечення роботи комп'ютерної мережі збору даних з розумних лічильників електроенергії, було реалізовано моделювання та зображення усіх компонентів проекту, складено схему функціонування PLC-шлюзу для передачі даних, написано програмний код для комунікації між сервером, базою даних та веб-інтерфейсом, а також описано роботу і зв'язок між основними компонентами мережі.

					КВРКІ 210102.21.01.30 ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

За результатами виконаних теоретичних та практичних досліджень було отримано навички роботи та організації комп'ютерних мереж, вивчено класифікації технологій зв'язку між серверами та розумними лічильниками електроенергії, здобуто поглиблене розуміння роботи PLC та NB-IoT.

У першому розділі проведено аналіз ситуації щодо використання розумних лічильників у світі та Україні, а також проаналізовано проблеми та рішення, які можуть виникнути під час розробки мережі. Було сформовано перші ідеї та порівняння підходів до вирішення задачі.

У другому розділі проведено підбір підходящих технологій та більш глибоке дослідження предметної області для подальшого формування та виконання проекту.

У третьому розділі було реалізовано функційне моделювання роботи мережі збору даних, забезпечено передавання даних на сервер, а також складено схематичні креслення для реалізації мережі збору даних з розумних лічильників електроенергії.

					КВРКІ 210102.21.01.30 ПЗ	Арк.
						61
Зм.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Офіційний сайт компанії YASNO. URL: [YASNO.com.ua](https://yasno.com.ua) (дата звернення 22.05.2025)
2. [Smart electricity meter market 2024: Global adoption landscape](https://iot-analytics.com/smart-meter-adoption/) by Adarsh Krishnan. URL: <https://iot-analytics.com/smart-meter-adoption/> (дата звернення 22.05.2025)
3. Компанія NIK. URL: <https://nik.net.ua> (дата звернення 24.05.2025)
4. Лічильник smart-MAIC D103. URL: <https://smart-maic.com/uk/> (дата звернення 24.05.2025)
5. Dr.Adithya Tiwary, Manish Mahato, Mohit Tripathi, Mayank Shrivastava, Matnak Kumar Chandrol, Abhitesh Chidar (2018), "Design and Implementation of an Innovative Internet of Things(IOT) Based Smart Energy Meter"(IJFRCSCCE), 244-247
6. «Optimizing IoT Energy Efficiency: Real-Time Adaptive Algorithms for Smart Meters with LoRaWAN and NB-IoT» by Kanar Alaa Al-Sammak, Sama Hussein Al-Gbur, Ion Marghescu, Ana-Maria Claudia Drăgulinesc, Cristina Marghescu, Alexandru Martian, Nawar Alaa Hussein Al-Sammak, George Suciu, and Khattab M. Ali Alheeti. URL: <https://www.mdpi.com/1996-1073/18/4/987>
7. «Smart Metering Systems Based on Power Line Communications» by José A. Cortés, J. Miguel Idiago. URL: [https://link.springer.com/chapter/10.1007/978-981-13-1768-2\\_4](https://link.springer.com/chapter/10.1007/978-981-13-1768-2_4) (дата звернення 28.05.2025)
8. «Strategies for Power Line Communications Smart Metering Network Deployment» by Alberto Sendin, Ivan Peña, Pablo Angueira. URL: <https://www.mdpi.com/1996-1073/7/4/2377>(дата звернення 28.05.2025)
9. «GSM for remote reading of electricity meters by Smart Energy International» Sep 30, 2004. URL: <https://www.smart-energy.com/regional-news/europe-uk/gsm-for-remote-reading-of-electricity-meters/>(дата звернення 28.05.2025)
10. «Advances in smart electricity meter field diagnostics» by Cosimo Carriero. URL: [https://www.power-and-beyond.com/advances-in-smart-electricity-meter-field-diagnostics-a-8e7a38b8197fbe137cbcbf6ac2ee7142/?cmp=go-ta-art-trf-PuB\\_DSA-](https://www.power-and-beyond.com/advances-in-smart-electricity-meter-field-diagnostics-a-8e7a38b8197fbe137cbcbf6ac2ee7142/?cmp=go-ta-art-trf-PuB_DSA-)

					КВРКІ 210102.21.01.30 ПЗ	Арк. 62
Зм.	Арк.	№ докум.	Підпис	Дата		

20250123-coma&gad\_source=1&gad\_campaignid=21513720705

&gclid=Cj0KCQjwu7TCBhCYARIsAM\_S3NjIkBOwJwY2QH9FSZD62ZuLNC3ff85  
GKP9q9drzmsle1UQ\_pWSaCkYaAkN7EALw\_wcB(дата звернення 28.05.2025)

11. «Cyber security and encryption: Best practices for your business» by Emrick Etheridge. URL: <https://www.dataguard.com/blog/cyber-security-measures-secure-your-business-with-encryption/>(дата звернення 29.05.2025)

12. «MQTT protocol for SME foundries: potential as an entry point into industry 4.0, process transparency and sustainability» by Eric Riedel. URL: [https://www.researchgate.net/publication/359083564\\_MQTT\\_protocol\\_for\\_SME\\_foundries\\_potential\\_as\\_an\\_entry\\_point\\_into\\_industry\\_40\\_process\\_transparency\\_and\\_sustainability](https://www.researchgate.net/publication/359083564_MQTT_protocol_for_SME_foundries_potential_as_an_entry_point_into_industry_40_process_transparency_and_sustainability)(дата звернення 29.05.2025)

13. Risk Assessment Method for 5G-oriented DLMS/COSEM Communications by Alexios Lekidis, Elpiniki I Papageorgiou. URL: [https://www.researchgate.net/publication/378720242\\_Risk\\_Assessment\\_Method\\_for\\_5G-oriented\\_DLMSCOSEM\\_Communications](https://www.researchgate.net/publication/378720242_Risk_Assessment_Method_for_5G-oriented_DLMSCOSEM_Communications)(дата звернення 30.05.2025)

14. Організація комп'ютерних мереж: підручник: для студ. спеціальності 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського ; Ю. А. Тарнавський, І. М. Кузьменко. – Електронні текстові дані (1 файл: 45,7 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. 21-32 с(дата звернення 30.05.2025)

15. Github Smart Meter data collector example code by raymar9. URL: <https://github.com/scs/smartmeter-datacollector>(дата звернення 30.05.2025)

16. Github Smart Meter decoders for smart meter manufacturers. URL: [https://github.com/BitBangingBytes/gr-smart\\_meters](https://github.com/BitBangingBytes/gr-smart_meters)(дата звернення 30.05.2025)

17. Bradley Fisher, Jeff Haynes «Managing your data and dashboards with Oracle Cloud ERP». URL: <https://www.bakertilly.com/insights/managing-your-data-and-dashboards-with-oracle-cloud> (дата звернення 30.05.2025)

18. Michal Wlosik «What Is a CRM and How Does It Work». URL: <https://clearcode.cc/blog/how-does-crm-work/> (дата звернення 30.05.2025)

					КВРКІ 210102.21.01.30 ПЗ	Арк. 63
Зм.	Арк.	№ докум.	Підпис	Дата		

19. Dr.Adithya Tiwary, Manish Mahato, Mohit Tripathi,Mayank Shrivastava,Matnak Kumar Chandrol,Abhitesh Chidar (2018), «Design and Implementation of an Innovative Internet of Things(IOT) Based Smart Energy Meter»(IJFRCSCCE)

20. Xu, Y.; Yan, C.; Liu, H.; Wang, J.; Yang, Z.; Jiang, Y. «Smart Energy Systems: A Critical Review on Design and Operation Optimization. Sustain. Cities» Soc. 2020, 62, 102369.

21. M.Faisal ,Thahia Fahrin Karim, Abu Ridwan Pavel, M.S.Hossain Lipu(2019) «Development of Smart Energy Meter For Energy Cost Analysis of Conventional Grid and Solar Energy»,(IEEE)

22. German M. «A Mobile Recognition System for Analog Energy Meter Scanning» M.German, G. Shalunts, D. Albertini., 2016. p. 247 – 256.

23. D. M. Svakha i A. Y. Varfolomieiev, «Система автоматичного візуального розпізнавання показань лічильника», Мікросист., Електрон. та Акуст., т. 23, вип. 6,с. 22–28, Груд 2018.

24. Smart Meter Outage Prediction by Sapdo Utomo. URL: <https://github.com/ICCC-Platform/Smart-Meter-Outage-Prediction>(дата звернення 30.05.2025)

25. LPWAN (low-power wide area network) By Sharon Shea, Executive Editor. URL: <https://www.techtarget.com/iotagenda/definition/LPWAN-low-power-wide-area-network> (дата звернення 26.05.2025)

26. LPWAN Explained: Low Power Wide Area Networks by Jean-Eudes Ambroise. URL: <https://www.emnify.com/blog/lpwan>(дата звернення 26.05.2025)

27. AN5715G3-Hybrid PLC & RF software. URL: [https://www.st.com/resource/en/application\\_note/an5715-g3hybrid-plc--rf-software-package-stmicroelectronics.pdf](https://www.st.com/resource/en/application_note/an5715-g3hybrid-plc--rf-software-package-stmicroelectronics.pdf) (дата звернення 28.05.2025)

28. Протокол MQTT by Олександр Пупена. URL: <https://pupenas.github.io/TI40/%D0%9B%D0%B5%D0%BA%D1%86/MQTT.html>( дата звернення 30.05.2025)

					КВРКІ 210102.21.01.30 ПЗ	Арк. 64
Зм.	Арк.	№ докум.	Підпис	Дата		

29. An extended guide to NB-IoT: Practical insights for IoT projects by Jazz Rosende. URL: <https://onomondo.com/blog/what-is-nb-iot-definitive-guide/>(дата звернення 29.05.2025)

30. Wang, G.; Giannakis, G.B.; Chen, J.; Sun, J. Distribution System State Estimation: An Overview of Recent Developments. Front. Inf. Technol. Electron.Eng. 2019, 20, 4–17.

31. [A comprehensive guide to cyber security protocols and best practices](#) by dataguard. URL: <https://www.dataguard.com/blog/cyber-security-protocols-and-best-practices/>(дата звернення 29.05.2025)

32. [Data Protection Strategies in Cybersecurity](#) By Jason Miller [Електронний ресурс] URL: <https://www.bitlyft.com/resources/data-protection-strategies-in-cybersecurity>(дата звернення 29.05.2025)

33. Ashna K, Sudhesh N George (2013), “GSM Based Automatic Energy Meter Reading With Instant Billing “(IEEE), 65-71

34. Shubham Pahurkar<sup>1</sup>, Subhash Diwakar<sup>2</sup>, Harshal Nerkar<sup>3</sup>, Sunita Patil<sup>4</sup> (2017) IOT Based Electric Bill Generation (IJARCCE)

35. Pandurang G.Kate, Jitendra R.Rana (2015),” ZigBee Based Monitoring Theft Detection & Automatic Electricity Meter Reading” (ICESA), 258-262

36. Jayant.P.Pawar, Amirthaganesh.S, ArunKumar.S, Satiesh Kumar.B(2016),” Real time energy measurement using smart meter”(IC-GET)

37. ATPL Series Power Line Communications Device DATASHEET. URL: <https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/doc43079F.pdf>(дата звернення 06.06.2025)

38. «UART протокол», автор: Сергій Матвієнко. URL: <https://itmaster.biz.ua/directory/standarts/uart.html>(дата звернення 06.06.2025)

39. ESP32-WROOM-32 Datasheet Explained: Specifications, Pinout & Performance by: Kapil Kumar. URL:

					КВРКІ 210102.21.01.30 ПЗ	Арк. 65
Зм.	Арк.	№ докум.	Підпис	Дата		

<https://www.electronifyindia.com/blogs/news/esp32-wroom-32-datasheet>(дата звернення 06.06.2025)

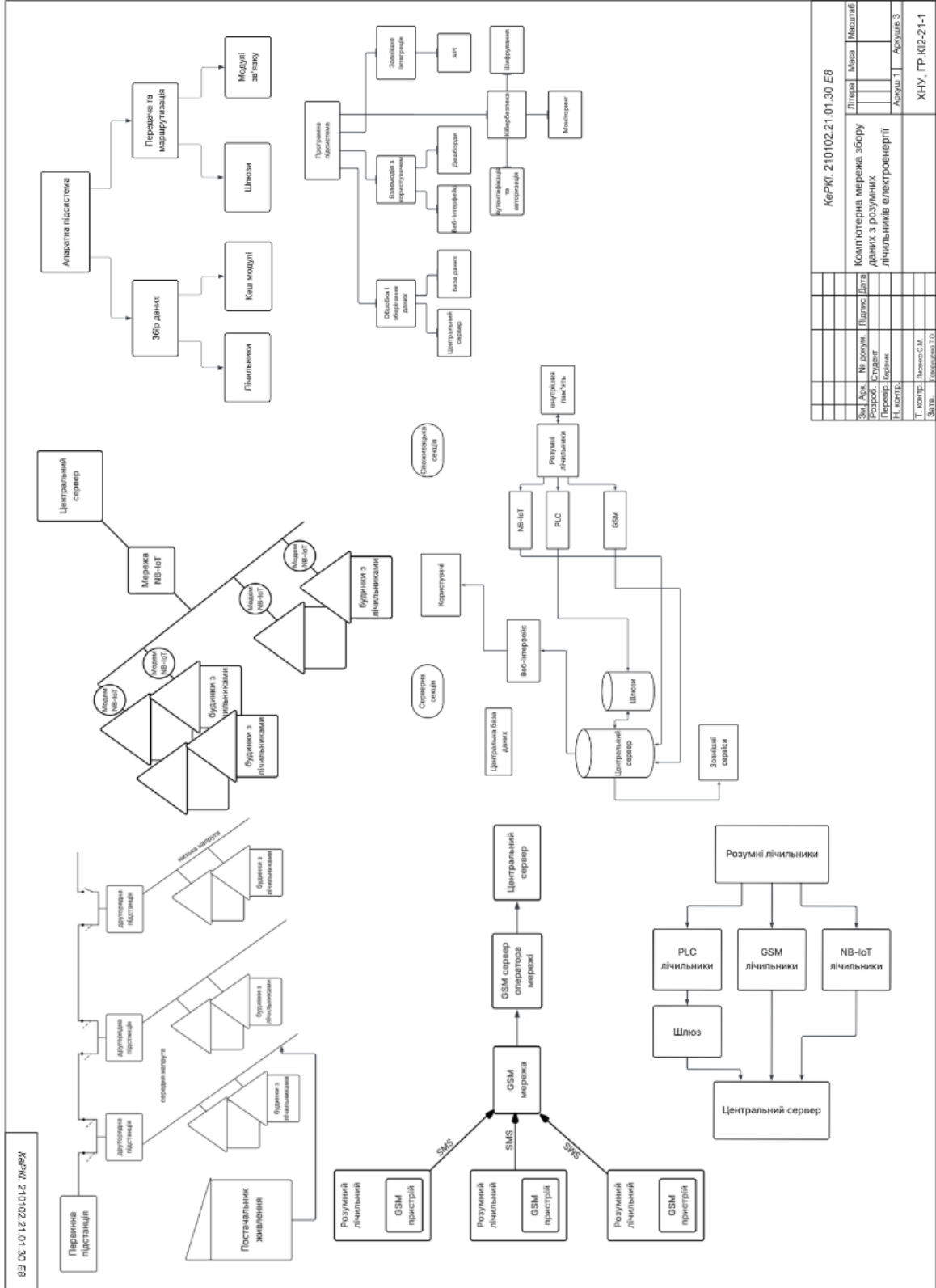
40. PFB Transformer in Narrowband PLC Coupling Circuit Designs for Smart Energy Infrastructure by Bourns. URL: [https://www.bourns.com/docs/technical-documents/technical-library/inductive-components/application-notes/bourns\\_pfb\\_transformer\\_in\\_narrowband\\_plc\\_coupling\\_circuit\\_designs\\_appnote.pdf?sfvrsn=27550f6\\_12](https://www.bourns.com/docs/technical-documents/technical-library/inductive-components/application-notes/bourns_pfb_transformer_in_narrowband_plc_coupling_circuit_designs_appnote.pdf?sfvrsn=27550f6_12)(дата звернення 06.06.2025)

41. Arduino & Serial Peripheral Interface (SPI). URL: <https://docs.arduino.cc/learn/communication/spi/>(дата звернення 06.06.2025)

					КВРКІ 210102.21.01.30 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

# Додаток А (обов'язковий)

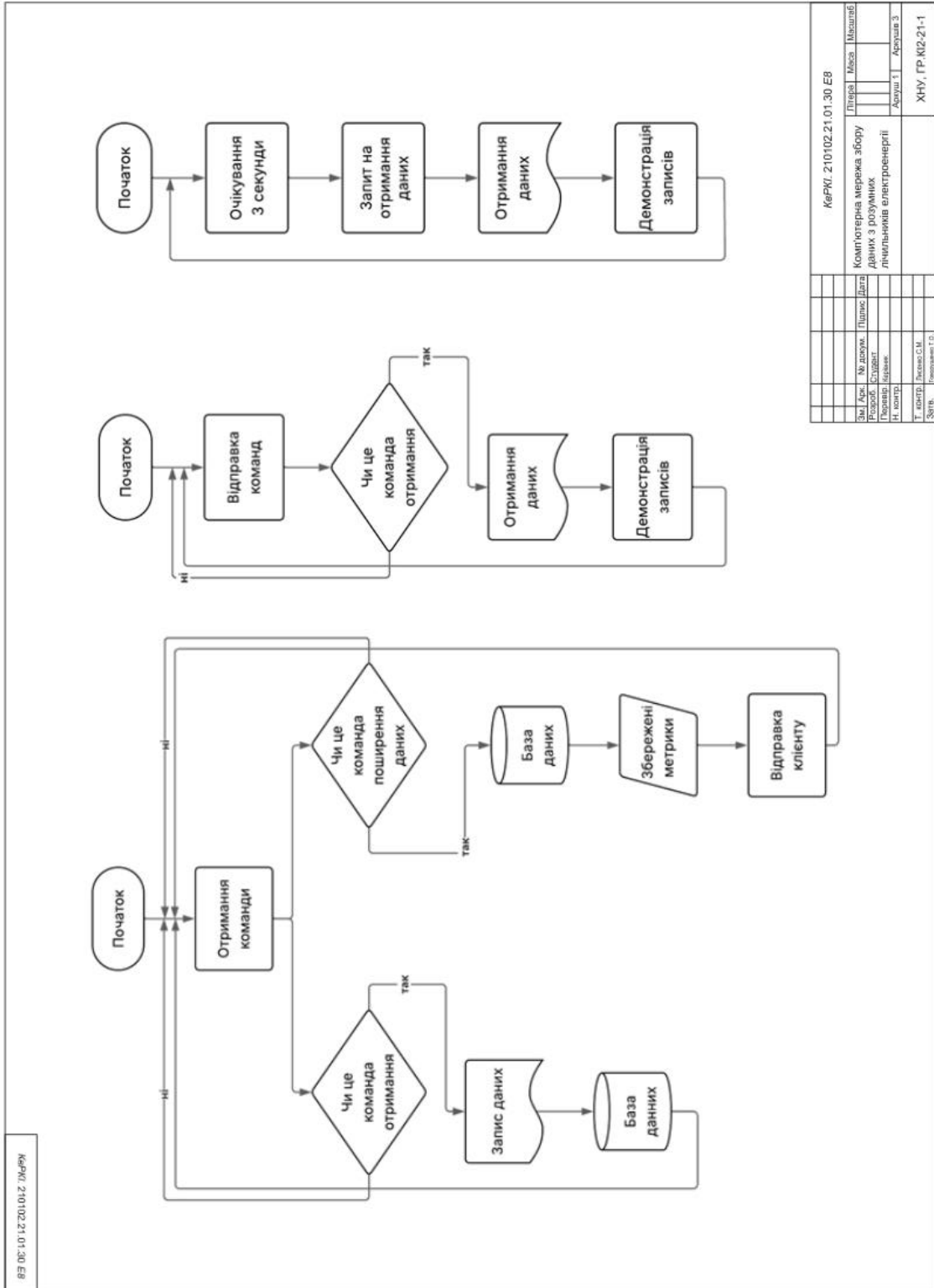
## КОПІЯ КРЕСЛЕННЯ «АРХІТЕКТУРА МЕРЕЖ ЗВ'ЯЗКУ ПРОЄКТУ»



КерМІ. 210102.21.01.30.ЕВ			
Літера	Міра	Масштаб	
Комп'ютерна мережа збору даних з розумних лічильників електроенергії			
Арх.1	Арх.уа.3		
ХНУ, ГР.КІЗ-21-1			

## Додаток Б (обов'язковий)

# КОПІЯ КРЕСЛЕННЯ «АРХІТЕКТУРА ПЗ ДЛЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ»



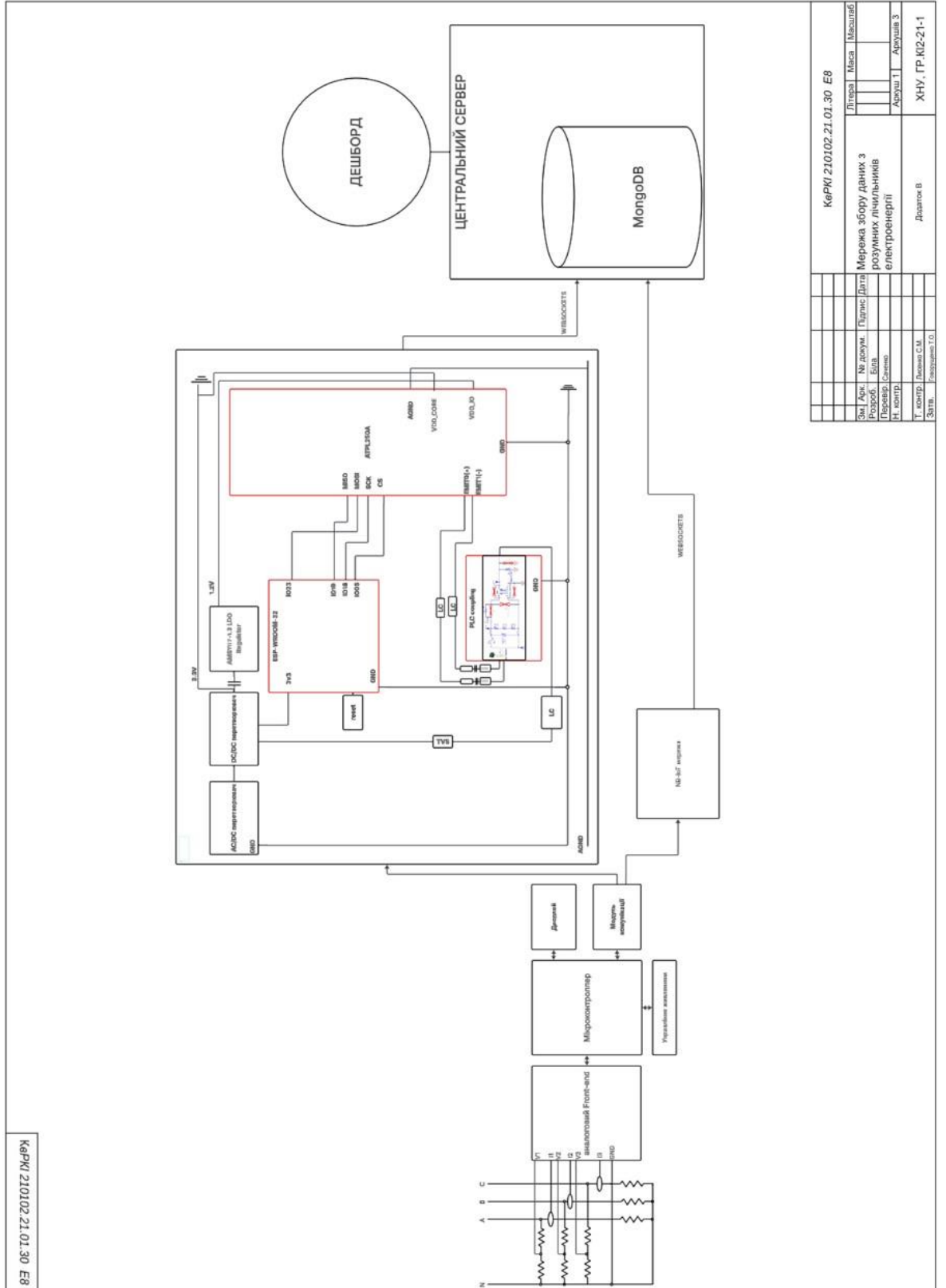
КерФКІ. 210102.21.01.30.ЕВ

КерФКІ. 210102.21.01.30.ЕВ			
Від. Бюро:	М. Дубоусь	Підпис:	Дата:
Розроб.:	С. Савченко	Перевір.:	М. Савченко
Н. контр.:	М. Савченко	Апробув. 1:	Апробув. 3:
Т. констр.:	Л. Савченко	С. М. Савченко	ХНУ, ГР. КІЗ-21-1
Зміст:	Савченко І.О.		
Комп'ютерна мережа збору даних з розумних лічильників електроенергії			
Літера:	Мова:	Місяць:	Рік:



# Додаток Г (обов'язковий)

## КОПІЯ КРЕСЛЕННЯ «СХЕМАТИЧНЕ КРЕСЛЕННЯ ПРОЕКТУ»



КерПК 21.01.02.21.01.30 E8		Літера	Маса	Масштаб
Вид. Бук.	№ докум.	Підпис	Дата	
Розроб.	Екз.			
Перевір.	Стор.			
Н. контр.				
Т. контр.	Листів			
Заст.	Архивів			
	Додаток В			
	ХНУ, ГР.К02-21-1			

КерПК 21.01.02.21.01.30 E8

## Додаток Д (обов'язковий)

### КОПІЯ КРЕСЛЕННЯ «ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРОЕКТУ»

```

import express from 'express'
import { WebSocketServer } from 'ws'
import mongoose from 'mongoose'
import http from 'http'

await mongoose.connect(`mongodb+srv://<username>:<password>@cluster0.mongodb.net/meters?retryWrites=true&w=majority`)
console.log('Connected to MongoDB')

const meterSchema = new mongoose.Schema({
  meterId: String,
  timestamp: Date,
  value: Number,
})

const MeterData = mongoose.model('MeterData', meterSchema)

const app = express()
const server = http.createServer(app)
const ws = new WebSocketServer({ server })

ws.on('connection', (ws) => {
  console.log('New client connected')

  ws.on('message', async (message) => {
    try {
      const parsed = JSON.parse(message)

      if (parsed.type === 'meter_data') {
        const { meterId, value } = parsed.data

        await MeterData.create({
          meterId,
          value,
          timestamp: new Date(),
        })

        ws.send(JSON.stringify({ type: 'status', status: 'saved' }))
      }

      if (parsed.type === 'get_data') {
        const data = await MeterData.find({ meterId: parsed.meterId }).sort({ timestamp: -1 }).limit(50)
        ws.send(JSON.stringify({ type: 'data_response', data }))
      }

    } catch (err) {
      console.error('Error handling message:', err)
      ws.send(JSON.stringify({ type: 'error', message: 'Invalid format or server error' }))
    }
  })

  ws.on('close', () => {
    console.log('Client disconnected')
  })
})

const PORT = process.env.PORT || 3000
server.listen(PORT, () => {
  console.log('WebSocket server running on port ${PORT}')
})

```

```

import axios from 'axios'

const SERVER_URL = 'http://localhost:3000'
const meterId = 'METER_001'

async function fetchData() {
  try {
    const response = await axios.get(`${SERVER_URL}/api/meter/${meterId}`)
    console.table(response.data.map(x => ({
      Time: new Date(x.timestamp).toLocaleString(),
      Value: x.value
    })))
  } catch (e) {
    console.error('HTTP error:', e.message)
  }
}

// Можна викликати вручну або раз на N секунд:
fetchData()

// setInterval(fetchData, 10000) // <- автозаяв кожні 10 сек

```

КерН 210102.21.01.30 Е8

КерН 210102.21.01.30 Е8		Літера	Місяц	Місяць
Зм. Акт.	№ докум.	Підпис	Дата	
Розроб.	Біла			
Перевір.	Сиченя			
Н. контр.				
Т. контр.	Дісько С.М.			
Зам.	Корчак І.О.			
Мережа збору даних з розумних лічильників електроенергії		Август 1		Август 3
Додаток Г		ХНУ, ГР. КІЗ-21-1		

## Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Анна БіЛА

**Співавтор:**

**Назва:** Біла\_Комп'ютерна мережа збору даних з розумних лічильників електроенергії

**Експерт:**

**Підрозділ:** Кафедра комп'ютерної інженерії та інформаційних систем

**Коефіцієнт подібності 1:**4.8%

**Коефіцієнт подібності 2:**2.6%

**Мікропробіли:** 22

**Заміна букв:** 20

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2025-06-20 04:13:55.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2025-06-20

Дата



Доцент Андрій Нічепорук

експерт

**Anti-Plagiarism (UA) v-15.281 Educational****The maximum coincidence with one document 1.0%**

Dictionaries check: en\_US, ru\_RU, ua\_UA. Errors in the documents: 15%

ID: 247085 Title: БКР Комп'ютерна мережа збору даних з розумних лічильників електроенергії Added in a DB: 2025-06-19 Authors: Анна БІЛА Heads: Світлана САЧЕНКО Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	68425	570	585 (1%)	9 (2%)

## Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Біла Анна Андріївна

Тема: Комп'ютерна мережа збору даних з розумних лічильників електроенергії

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 4 Кількість сторінок записки 57

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розробка то моделювання комп'ютерної мережі збору даних з розумних лічильників електроенергії

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

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі було описано предметну область, проаналізовано проблеми і рішення та проведено аналіз наявних можливостей. В другому розділі було обрано типи зв'язку мережі, детально досліджено їх роботу, було складено схеми роботи усієї мережі; визначено апаратну та програмну підсистеми програмно-технічного засобу; розроблено алгоритм роботи мережі та передачі даних; визначено потрібні технічні елементи для реалізації; проаналізовано системні вимоги та вимоги до захисту даних; в третьому розділі описано роботу кожного елемента системи; реалізовано сервер та базу даних; складено креслення PLC-шлюзу, лічильника та програмної частини; реалізовано веб-інтерфейс та його взаємодія з сервером.

---

---

---

---

4. Позитивні сторони роботи: достатня практична цінність роботи, приділено увагу програмній реалізації.

5. Негативні сторони роботи: недостатньо приділено уваги зображенню апаратних креслень.

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

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

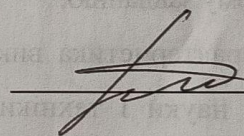
8. Інші зауваження: \_\_\_\_\_

9. Оцінка дипломної роботи: задовільно/Е/3,00

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_

*Валерій Русман Олександрович, доц. кафедри КЦ*

“\_19\_” \_\_\_\_06\_\_\_\_ 2025 р.

 (підпис)

Завідувачу кафедри КПС  
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Білої Анни Андріївни

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-21-1

### ЗАЯВА

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

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

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

18.06 2025 року

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА  
ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Комп'ютерна мережа збору даних з розумних лічильників електроенергії

Автор Анна БІЛА

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти перший (бакалаврський) рівень

Спеціальність 123 – Комп'ютерна інженерія

Науковий керівник: к.т.н., доцент Світлана САЧЕНКО

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	Відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	Не виявлено

Підтвердження:

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

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 4,75% і адресується до 19 першоджерела; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КПС



Світлана САЧЕНКО

Андрій НІЧЕПОРУК

Ольга ПАВЛОВА