

## КВАЛІФІКАЦІЙНА РОБОТА

Кіберфізична система «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина

Назва теми

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Шифр КвРКІ 022056.22.02.12 ПЗ

Виконав здобувач IV курсу, група КІ2-22-2

  
Підпис

Владислав МАРКЕВИЧ

Ініціали, прізвище

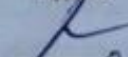
Керівник канд. тех. наук, доцент  
Науковий ступінь, учене звання

  
Підпис

Катерина БЕРЕЗЬКА

Ініціали, прізвище

Нормоконтролер канд. фіз.-мат. наук, доц.  
Науковий ступінь, учене звання

  
Підпис

Тетяна КИСІЛЬ

Ініціали, прізвище

До захисту допускаю:  
завідувач кафедри КІС

  
Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

«01» червня 2026 р.

дата

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Завідувачка кафедри КІС

Ольга ПАВЛОВА

“ 10 ” 01 2026 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Маркевичу Владиславу Анатолійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кіберфізична система «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина

Керівник проекту (роботи) Березька Катерина Миколаївна, к.т.н., доцент

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

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

Кіберфізична система «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина та постановка задачі щодо її удосконалення

Проектування системи обробки інформації у кіберфізичній системі «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина

Програмно-апаратна реалізація кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) \_\_\_\_\_

Архітектура ПЗ проекту

Схема електрична принципова

Монтажна схема


6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

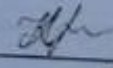
7. Дата видачі завдання « 10 » 01 2026 р.

**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	10.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2026	виконано
4	Робота над розділом 2 – вибір компонентів для проектування системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина	01.04.2026	виконано
5	Робота над розділом 3 – проектування системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина	29.04.2026	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2026	виконано
7	Попередній захист ВКР	26.05.2026	виконано
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач  Підпис

Владислав МАРКЕВИЧ  
Імя, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи  Підпис

Катерина БЕРЕЗЬКА  
Імя, 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 022056.22.02.12 ПЗ	Пояснювальна записка	58		
			<u>Графічні матеріали</u>			
2		КВРКІ 022056.22.02.12 Е8	Архітектура ПЗ проекту	1		
3		КВРКІ 022056.22.02.12 Е8	Схема електрична принципова	1		
4		КВРКІ 022056.22.02.12 Е8	Монтажна схема	1		

КВРКІ 022056.22.02.12 ВП

Зм	Арж	№ докум	Під пис	Дата
Розробив		Маркевич		
Перевір.		Березька		
Н. конпр.		Кисіль		
Затв.		Павлова		01.06

Відомість проекту

Літера	Аркуш	Аркушів
У	1	1
ХНУ, КІ2-22-2		

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Кіберфізична система «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина».

Автор роботи: Владислав МАРКЕВИЧ.

Керівник роботи: Катерина БЕРЕЗЬКА.

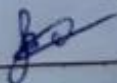
Пояснювальна записка: 58 с., 17 рис., 1 табл., 3 дод., 54 джерел.

Графічна частина: 3 креслення.

БАЗА ДАНИХ, ЗБІР ДАНИХ, ІНТЕРНЕТ РЕЧЕЙ, КІБЕРФІЗИЧНА СИСТЕМА, МОНІТОРИНГ, ОПТИМІЗАЦІЯ МАРШРУТІВ, РЕАЛЬНИЙ ЧАС, РОЗУМНІ КОНТЕЙНЕРИ, СЕНСОРИ.

Кваліфікаційна робота бакалавра присвячена розробці та дослідженню кіберфізичної системи моніторингу рівня заповненості сміттєвих контейнерів у режимі реального часу. Актуальність теми зумовлена необхідністю оптимізації процесів поводження з твердими побутовими відходами, зниження логістичних витрат комунальних служб та покращення екологічного стану населених пунктів. Використання технологій Інтернету речей та сенсорних мереж дозволяє своєчасно отримувати дані про стан накопичувачів, значно підвищуючи ефективність.

Метою роботи є проектування, реалізація та тестування апаратно-програмного комплексу для збору, передавання, оброблення й візуалізації даних про стан сміттєвих контейнерів. Для досягнення поставленої мети було виконано аналіз сучасних підходів до систем розумного управління відходами, обрано сенсорну та мікроконтролерну базу, розроблено структурну схему системи. Також спроектовано програмне забезпечення для обробки даних і користувацький інтерфейс, що формує інформаційне підґрунтя для прийняття рішень щодо своєчасного обслуговування контейнерів та оптимізації маршрутів їх вивезення.



Підпис здобувача





30.05.2026

Дата

# ЗМІСТ

Вступ.....	4
1 Кіберфізична система «розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина .....	6
1.1 Аналіз структурних і функціональних особливостей кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина .....	6
1.2 Аналіз програмно-апаратного забезпечення обробки інформації в кіберфізичній системі «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина .....	8
1.3 Постановка задачі .....	14
1.4 Висновки до першого розділу.....	15
2 Проектування системи обробки інформації у кіберфізичній системі «розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. програмна частина .....	16
2.1 Визначення мети та встановлення вимог до проектованої кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина .....	16
2.2 Загальна структура апаратної реалізації кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина.....	20
2.3 Схема кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина .....	24
2.4 Монтажна схема кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина.....	29

КвРКІ.022056.22.02.12 ПЗ

Зм.	Арк.	№докум.	Підпис	Дата		Літера	Аркуш	Аркушів
Виконав		МАРКЕВИЧ В.			Підсистема керування освітленням кіберфізичної системи «Розумний будинок». Пояснювальна записка	у	2	58
Перевір.		БЕРЕЗЬКА К.				ХНУ КІ2-22-2		
Н.контр.		Тетяна КИСЛЬ		07.06				
Затвер.		Ольга ПАВЛОВА						

2.5 Аналіз обраних рішень .....	33
2.5.1 Аналіз обраних апаратних рішень та їх технічних характеристик .....	34
2.5.2. Аналіз обраних програмних рішень.....	40
2.6 Висновки до другого розділу .....	42
3 Програмно-апаратна реалізація кіберфізичної системи «розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. програмна частина.....	43
3.1 Огляд використовуваних програмних бібліотек та їх функціональне призначення .....	43
3.2 Розробка програмного коду для моніторингу збору відходів у режимі реального часу.....	46
3.3 Розробка веб-інтерфейсу та візуалізація даних у режимі реального часу .....	57
3.4 Висновки до третього розділу.....	60
Висновки .....	61
Перелік джерел посилань .....	63
Додаток А Копія креслення «Архітектура ПЗ проекту».....	71
Додаток Б копія креслення «схема електрична принципова» .....	72
Додаток В копія креслення «монтажна схема».....	73

## ВСТУП

Умови стрімкої урбанізації та зростання обсягів твердих побутових відходів висувають підвищені вимоги до організації систем їх збору та утилізації. Традиційні підходи до обслуговування сміттєвих контейнерів, що базуються на фіксованих графіках вивезення, часто є неефективними, оскільки не враховують реальний рівень їх заповнення. Це призводить як до переповнення контейнерів і погіршення санітарного стану територій, так і до нераціонального використання ресурсів комунальних служб.

У цьому контексті доцільним є використання кіберфізичних систем та технологій Інтернету речей. Такі системи дозволяють поєднати фізичні об'єкти, зокрема сміттєві контейнери, із цифровими сервісами шляхом використання сенсорів, засобів бездротового зв'язку та програмного забезпечення для збору й аналізу даних. Такі системи дозволяють поєднати фізичні об'єкти (сміттєві контейнери) із цифровими сервісами шляхом використання сенсорів, засобів бездротового зв'язку та програмного забезпечення для збору й аналізу даних. Це відкриває можливість здійснювати моніторинг стану контейнерів у режимі реального часу, оперативно реагувати на їх заповнення та оптимізувати логістику вивезення відходів.

Використання розумних сміттєвих контейнерів сприяє підвищенню ефективності роботи міських служб, зменшенню витрат на транспортування, зниженню негативного впливу на довкілля та покращенню якості життя населення. Крім того, такі рішення є важливою складовою концепції «розумного міста», що передбачає інтеграцію сучасних інформаційних технологій у міську інфраструктуру.

Таким чином, розробка кіберфізичної системи моніторингу збору відходів є актуальним і практично значущим завданням, спрямованим на підвищення ефективності управління міським середовищем та впровадження сучасних підходів у сфері поводження з відходами

					КвРКІ.022056.22.02.12 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

Метою дипломної роботи є розробка, реалізація та тестування працездатної кіберфізичної системи «розумних» сміттєвих контейнерів, що забезпечує автоматизоване визначення рівня заповнення та передачу отриманих даних до інформаційної системи з подальшим відображенням для користувачів.

Об'єктом дослідження є кіберфізична система моніторингу збору відходів, у межах якої відбувається взаємодія сенсорних пристроїв, обчислювальних модулів для відстеження стану контейнерів та підтримки процесів вивезенням сміття.

Предметом дослідження виступають технічні й програмні засоби, методи збору, передачі, обробки та візуалізації даних, що формують інформаційне підґрунтя для прийняття рішень щодо своєчасного обслуговування контейнерів, а також принципи побудови такої системи й підвищення її ефективності.

					КВРКІ.022056.22.02.12 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

# 1 КІБЕРФІЗИЧНА СИСТЕМА «РОЗУМНІ СМІТТЄВІ КОНТЕЙНЕРИ» ДЛЯ МОНІТОРИНГУ ЗБОРУ ВІДХОДІВ У РЕЖИМІ РЕАЛЬНОГО ЧАСУ. ПРОГРАМНА ЧАСТИНА

1.1 Аналіз структурних і функціональних особливостей кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина

Проблема ефективної організації процесів поводження з відходами є однією з важливих для сучасних урбанізованих територій [1]. Постійне зростання обсягів сміття в поєднанні з обмеженими ресурсами комунальних служб ускладнює забезпечення належного рівня обслуговування [2]. Нерівномірне заповнення контейнерів у різних районах міста призводить до ситуацій, коли частина з них переповнюється, тоді як інші обслуговуються передчасно. Це негативно впливає як на санітарний стан, так і на ефективність використання техніки та людських ресурсів [3].

Суттєвим недоліком традиційних підходів є їх статичність: графіки вивезення формуються заздалегідь і не враховують реального стану контейнерів [4]. У таких умовах відсутній механізм оперативного реагування на зміну навантаження, що особливо помітно у періоди пікової активності або в густонаселених районах. Як наслідок, виникають додаткові витрати, погіршується екологічна ситуація та знижується якість надання послуг [5].

Застосування кіберфізичних систем дозволяє перейти від заздалегідь визначених сценаріїв до адаптивного управління [6]. Оснащення контейнерів сенсорними модулями дає змогу отримувати актуальну інформацію про їх стан, яка передається до центральних систем обробки [7]. Надалі ці дані можуть використовуватись для аналізу, прогнозування та підтримки прийняття рішень щодо обслуговування інфраструктури збору відходів.

Важливою перевагою такого підходу є можливість побудови гнучкої логістики. Замість фіксованих маршрутів формується динамічна схема руху

					КвРКІ.022056.22.02.12 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

спеціалізованого транспорту, що враховує фактичну потребу у вивезенні сміття [8]. Це дозволяє зменшити кількість зайвих рейсів, скоротити час обслуговування та підвищити загальну продуктивність системи [9].

Окремо слід відзначити роль таких рішень у розвитку цифрової інфраструктури міста [10]. Інтеграція даних про стан контейнерів з іншими інформаційними системами відкриває можливості для комплексного аналізу та довгострокового планування [11]. Зокрема, це може бути корисним для визначення оптимальних місць розташування контейнерів, оцінки навантаження на окремі райони або вдосконалення схем обслуговування.

Важливим аспектом є також підвищення прозорості процесів у сфері поводження з відходами. Використання цифрових технологій дозволяє відстежувати виконання робіт, контролювати своєчасність обслуговування контейнерів та оцінювати ефективність діяльності комунальних підприємств [12]. Це створює передумови для більш якісного управління та зменшує вплив людського фактора.

Крім того, накопичення великого обсягу даних відкриває можливості для застосування сучасних методів аналітики [13]. На основі історичної інформації можна виявляти закономірності у заповненні контейнерів, прогнозувати навантаження та планувати розвиток інфраструктури. Це особливо важливо для довгострокового управління міськими ресурсами та підвищення ефективності прийняття управлінських рішень [14].

Практичне впровадження кіберфізичних систем у сфері збору відходів забезпечує ряд відчутних переваг [15]:

- підвищення своєчасності обслуговування контейнерів та зменшення кількості випадків їх переповнення;
- раціональніше використання транспортних і паливних ресурсів;
- скорочення витрат на експлуатацію техніки;
- зменшення негативного впливу на довкілля;

					КВРКІ.022056.22.02.12 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

– можливість накопичення статистичних даних для подальшого аналізу та прогнозування.

У підсумку, використання кіберфізичних систем у сфері управління відходами дозволяє не лише вирішити окремі операційні проблеми, а й створює основу для переходу до більш ефективної, гнучкої та технологічно розвиненої моделі функціонування міської інфраструктури.

1.2 Аналіз програмно-апаратного забезпечення обробки інформації в кіберфізичній системі «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина

Реалізація кіберфізичних систем у сфері моніторингу збору відходів базується на поєднанні апаратних компонентів, засобів передачі даних що формують єдину інформаційну інфраструктуру [16]. Залежно від масштабу та вимог до системи, її архітектура може відрізнятися, проте у більшості випадків вона включає рівень збору даних, рівень передачі та рівень обробки й візуалізації [17]. Головною метою впровадження таких систем є перехід від традиційних, статичних графіків вивезення сміття до динамічних моделей обслуговування. Завдяки аналізу телеметричних даних у реальному часі, комунальні служби отримують можливість суттєво оптимізувати логістичні маршрути спецтехніки, зменшити експлуатаційні витрати автопарку та знизити рівень шкідливих викидів у навколишнє середовище.

На фізичному рівні основну роль відіграють сенсорні пристрої на фізичному рівні використовуються сенсорні пристрої, найбільш поширеним рішенням є ультразвукові датчики, які визначають рівень заповнення контейнера шляхом вимірювання відстані до поверхні відходів. Приклад використання ультразвукового датчика у сміттєвому контейнері наведено на рисунку 1.1.

					КВРКІ.022056.22.02.12 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		



Також у сучасних системах моніторингу відходів застосовуються інфрачервоні та оптичні сенсори, які фіксують наявність об'єктів у певній зоні [21]. Інфрачервоні датчики дозволяють визначати присутність сміття або факт відкриття кришки контейнера за допомогою випромінювання та відбиття інфрачервоного світла. Оптичні сенсори, у свою чергу, використовують світлові промені для виявлення об'єктів і можуть допомагати уточнювати показники заповнення, коли ультразвукові або вагові сенсори дають неточні дані через нерівномірне розташування відходів [22].

Крім того, у сучасних рішеннях часто інтегрують магнітні та ємнісні сенсори. Магнітні датчики дозволяють відстежувати відкриття кришки контейнера, а ємнісні визначають рівень заповнення за зміною електричного поля навколо відходів [23]. Використання цих технологій підвищує точність системи і забезпечує додаткові дані.

Для підвищення надійності та точності у більш складних системах використовуються комбіновані сенсорні модулі, які об'єднують декілька принципів вимірювання в одному пристрої [24]. Такі модулі дозволяють одночасно визначати рівень заповнення контейнера, його масу, факт відкриття кришки та наявність сторонніх об'єктів. Це значно ймовірність помилок, спричинених впливом погодних умов, нерівномірним розташуванням відходів або механічними коливаннями контейнера. Приклад комбінованого сенсорного модуля наведено на рисунку 1.3.

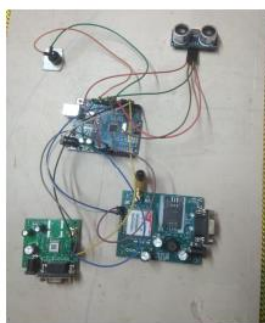


Figure 1



Figure 2

Рисунок 1.3 – Комбінований сенсор для моніторингу стану контейнера [25]

									КВРКІ.022056.22.02.12 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата						

Окрему категорію в сучасних системах моніторингу відходів становлять рішення на основі комп'ютерного зору [26]. Використання камер дає змогу не лише визначати рівень заповнення контейнера, а й оцінювати тип відходів, їх розподіл всередині контейнера та загальний стан самого контейнера. Такі дані дозволяють більш точно планувати маршрути вивезення відходів, визначати потребу у сортуванні та навіть виявляти неправомірні дії, наприклад, скидання небезпечних предметів у звичайні контейнери [27]. Зокрема, системи відеофіксації дозволяють проводити базовий морфологічний аналіз сміття, розпізнаючи вторсировину (пластик, папір, скло), що є важливим для контролю якості роздільного збору.

Комп'ютерне бачення часто використовує алгоритми обробки зображень та штучного інтелекту для автоматичного розпізнавання об'єктів і аналізу їх характеристик. Це забезпечує високий рівень автоматизації процесу та дозволяє інтегрувати систему з іншими інформаційними платформами для прийняття рішень у режимі реального часу.

Наприклад, можна автоматично відправляти повідомлення операторам про перевищення допустимого рівня заповнення або про наявність сторонніх предметів. У такому випадку аналіз зображення відбувається безпосередньо на локальному мікрокомп'ютері, а на сервер передається не важкий відеопотік, а лише готові метадані.

Проте подібні системи є значно дорожчими порівняно з класичними сенсорними рішеннями. Вони потребують більшої обчислювальної потужності для обробки відеопотоків, а також стабільного та швидкого каналу передачі даних. Через це застосування камер у системах моніторингу відходів переважно обмежується великими міськими проєктами, дослідницькими ініціативами або спеціальними випадками, де необхідно отримати розширену аналітику. Приклад використання відеокамер у системах моніторингу відходів наведено на рисунку 1.4.



контейнера. Це дозволяє зменшити обсяг переданої інформації та підвищити енергоефективність системи [32].

Подальша обробка передбачає агрегацію даних із великої кількості пристроїв, їх збереження та аналіз. Для цього використовуються серверні рішення або хмарні платформи, що забезпечують масштабованість і доступ до інформації в режимі реального часу.

Для ефективної взаємодії користувачів із системою моніторингу збору відходів застосовуються веб-інтерфейси та мобільні додатки. Вони дозволяють у режимі реального часу відображати актуальний стан контейнерів, відстежувати їх заповненість та визначати пріоритетні маршрути для вивезення сміття. Такі інтерфейси забезпечують зручне відображення даних, що дозволяє приймати оперативні рішення без необхідності безпосередньої присутності на місці.

Крім базових функцій моніторингу, користувацькі інтерфейси включають можливість формування маршрутів вивезення, оптимізованих за показниками заповненості контейнерів та географічним розташуванням. Це дозволяє значно підвищити ефективність роботи транспортних служб та зменшити витрати на логістику. Мобільні додатки забезпечують доступ до системи з будь-якої точки міста, що особливо важливо для інспекторів та операторів, які перебувають у полі. Для операторів доступні розширені функції, зокрема перегляд статистики за певні періоди, аналіз ефективності роботи системи та налаштування параметрів сенсорів або сповіщень. Такі можливості дозволяють своєчасно виявляти проблеми, коригувати алгоритми маршрутизації та підтримувати стабільну роботу системи на високому рівні.

Таким чином, сучасні апаратні та програмні рішення забезпечують широкі можливості для побудови кіберфізичних систем моніторингу збору відходів. Різноманітність технологій дозволяє адаптувати систему під конкретні умови використання, забезпечуючи баланс між вартістю, точністю та функціональністю.

					КВРКІ.022056.22.02.12 ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

### 1.3 Постановка задачі

Головною метою дослідження даної кваліфікаційної роботи є розробка функціонального прототипу кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу з можливістю автоматичної фіксації рівня заповнення контейнерів, передачі даних, їх обробки та візуалізації.

Для досягнення цієї мети необхідно виконати наступні етапи дослідження та розробки:

- визначення функціональних та нефункціональних вимог до системи, включаючи точність сенсорів, частоту передачі даних, автономність та способи взаємодії з користувачами;
- розробка загальної архітектури системи, що охоплює рівні збору, передачі, обробки та візуалізації даних;
- вибір апаратних компонентів, зокрема мікроконтролера, сенсорів рівня заповненості, модулів бездротового зв'язку та елементів живлення;
- розробка принципів електричних та монтажних схем пристрою для коректного підключення обраних компонентів;
- вибір середовища програмування, необхідних бібліотек, протоколів передачі даних та хмарної платформи для їх збереження й обробки;
- реалізація програмного забезпечення для керування сенсорами, обробки сигналів та відображення інформації через користувацький інтерфейс;
- створення готового прототипу системи, здатного в реальному часі фіксувати стан контейнерів, передавати дані на сервер та відображати їх оператору.

Таким чином, дотримуючись вищезазначених етапів, можна виконати поставлену задачу та досягти мети кваліфікаційної роботи створення ефективної кіберфізичної системи моніторингу збору відходів у режимі реального часу.

					КВРКІ.022056.22.02.12 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

#### 1.4 Висновки до першого розділу

У даному розділі було проведено комплексний аналіз сучасного стану розробок у сфері моніторингу збору відходів із застосуванням кіберфізичних систем. Було обґрунтовано актуальність використання КФС для автоматизації процесу контролю рівня заповнення сміттєвих контейнерів, що дозволяє підвищити ефективність роботи систем збору відходів, зменшити витрати на логістику та покращити санітарний стан міського середовища.

Був проведений всебічний аналіз можливих типів сенсорів, які використовуються для фіксації рівня заповнення контейнерів: ультразвукових, вагових, інфрачервоних, оптичних, комбінованих та камерних модулів. Для кожного типу розглянуто особливості збору, передачі та обробки даних, а також визначено переваги та обмеження застосування в різних умовах експлуатації.

Загалом аналіз показав, що сучасні апаратні та програмні рішення дозволяють будувати ефективні, надійні та масштабовані кіберфізичні системи моніторингу відходів. Розуміння сильних та слабких сторін різних підходів створює базу для обґрунтованого вибору технічних і програмних компонентів при розробці власного прототипу «Розумного сміттевого контейнера», що дозволить забезпечити автоматичну фіксацію даних у режимі реального часу та ефективне управління процесом збору відходів.

					КвРКІ.022056.22.02.12 ПЗ	Арк. 15
Зм.	Арк.	№ докум.	Підпис	Дата		

## **2 ПРОЄКТУВАННЯ СИСТЕМИ ОБРОБКИ ІНФОРМАЦІЇ У КІБЕРФІЗИЧНІЙ СИСТЕМІ «РОЗУМНІ СМІТТЄВІ КОНТЕЙНЕРИ» ДЛЯ МОНІТОРИНГУ ЗБОРУ ВІДХОДІВ У РЕЖИМІ РЕАЛЬНОГО ЧАСУ. ПРОГРАМНА ЧАСТИНА**

2.1 Визначення мети та встановлення вимог до проєктованої кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина

Будь-який інженерний процес створення складних програмно-апаратних комплексів починається з чіткого формулювання кінцевої мети. Для розробки кіберфізичної системи управління збором відходів цей етап є важливим, оскільки він задає вектор розвитку архітектури, впливає на вибір технологічного стека та математичних алгоритмів. Розмиті орієнтири або їх зміна в процесі імплементації неминуче призводять до розростання технічного боргу, збільшення витрат часу та неефективності готового продукту.

Слід зазначити, що у контексті кіберфізичних систем формулювання мети виходить за межі класичного програмного проєктування, оскільки охоплює не лише програмну складову, але й фізичні компоненти системи датчики, канали передачі даних та виконавчі механізми. Таким чином, мета повинна враховувати особливості взаємодії між цифровим і фізичним середовищем, зокрема затримки передачі даних, можливі похибки вимірювань та обмеження апаратного забезпечення.

Крім того, важливо враховувати зовнішні фактори, які можуть впливати на функціонування системи, такі як погодні умови, щільність забудови міста або нестабільність мережевого покриття. Ігнорування цих аспектів на етапі постановки задачі може призвести до часткової або повної непридатності системи в реальних умовах експлуатації.

Основним завданням даного проєкту є проєктування та розробка програмного рішення для автоматизації аудиту твердих побутових відходів.

					КВРКІ.022056.22.02.12 ПЗ	Арк. 16
Зм.	Арк.	№ докум.	Підпис	Дата		

Головна ідея полягає в тому, щоб на основі даних про наповненість сміттєвих баків будувати такі логістичні маршрути, які б повністю виключали візити спецтехніки до порожніх або напівпорожніх контейнерів. Це дозволить суттєво знизити витрати паливно-мастильних матеріалів, оптимізувати робочий час екіпажів та розвантажити дорожній рух у межах міста.

Важливою складовою поставленої мети є також екологічний аспект. Оптимізація маршрутів збору відходів сприяє зменшенню викидів вуглекислого газу за рахунок скорочення пробігу спецтехніки. Таким чином, впровадження системи має не лише економічний, але й екологічний ефект, що відповідає сучасним концепціям сталого розвитку міської інфраструктури.

Окрім цього, система може виступати як інструмент для прийняття управлінських рішень. На основі накопичених даних можливо визначати проблемні райони з підвищеним рівнем накопичення відходів, оптимізувати розміщення контейнерів та планувати їх обслуговування з урахуванням сезонних або поведінкових змін населення.

Спираючись на затверджену мету, проводиться декомпозиція завдання на конкретні технічні критерії. Вони встановлюють межі можливостей системи, специфіку обробки даних та умови її надійної експлуатації. Крім того, виявлення логічних помилок на стадії формування вимог обходиться значно дешевше, ніж рефакторинг коду на етапі запуску. Традиційно ці критерії класифікують на функціональні та нефункціональні вимоги.

Процес формування вимог також передбачає визначення зацікавлених сторін (stakeholders), до яких належать комунальні служби, органи місцевого самоврядування та кінцеві користувачі системи диспетчери. Кожна з цих груп має власні очікування від системи, що повинні бути враховані для забезпечення її практичної цінності.

Також доцільно враховувати можливість подальшої інтеграції системи з іншими міськими інформаційними платформами, такими як системи «розумного

міста» (Smart City), що дозволить розширити функціональність та підвищити ефективність використання даних.

До функціональних вимог розроблюваної системи моніторингу відходів належать:

– безперервний збір телеметрії, програмний комплекс має своєчасно приймати показники від ультразвукових датчиків (або спеціалізованого модуля симуляції), актуалізуючи дані про відсоток заповненості кожного бака;

– інтелектуальна фільтрація, алгоритми системи повинні самостійно відбирати для подальшого обслуговування виключно ті контейнери, рівень накопичення сміття в яких перевищує встановлений критичний поріг (наприклад, понад 80%);

– динамічна маршрутизація, система повинна розраховувати найкоротший та найвигідніший шлях об'їзду проблемних локацій, поєднуючи математичні моделі оптимізації (алгоритми найближчого сусіда та 2-opt) з картографічними сервісами для врахування реальної дорожньої мережі;

– візуальний моніторинг: обов'язковою є наявність диспетчерського веб-інтерфейсу, який відображатиме контейнери на інтерактивній карті за допомогою кольорової індикації (залежно від їх статусу) та дозволить переглядати геометрію побудованих маршрутів;

– збір аналітики, система повинна фіксувати історію заповнень та маршрутів для формування статистичної бази даних.

Окрему увагу слід приділити точності та достовірності отриманих даних, оскільки саме вони є основою для подальших обчислень і прийняття рішень. Використання неякісних або некоректно відкаліброваних датчиків може призвести до побудови неефективних маршрутів та зниження загальної продуктивності системи.

Крім того, функціональні можливості системи можуть бути розширені за рахунок впровадження механізмів прогнозування. Застосування методів машинного навчання дозволить передбачати рівень заповненості контейнерів на

					КВРКІ.022056.22.02.12 ПЗ	Арк. 18
Зм.	Арк.	№ докум.	Підпис	Дата		

основі історичних даних, що відкриває можливість переходу від реактивної до проактивної моделі управління.

Серед основних нефункціональних вимог до кіберфізичної системи виділяють наступні:

– висока продуктивність обчислень, генерація логістичних маршрутів має відбуватися швидко, незважаючи на складність математичних операцій, а дані на диспетчерській панелі повинні оновлюватися без видимих затримок;

– відмовостійкість архітектури, додаток повинен гарантувати стабільну роботу та продовжувати генерувати базові маршрути об'їзду навіть у разі тимчасової недоступності зовнішніх навігаційних сервісів (використання резервних алгоритмів обчислення);

– масштабованість рішення, структура бази даних та модульний підхід до написання коду повинні забезпечувати можливість легкого додавання нових контейнерів або розширення зони обслуговування без необхідності глобального переписування ядра системи;

– ергономічність інтерфейсу, подання географічної та аналітичної інформації має бути структурованим, інтуїтивно зрозумілим і не перевантажувати користувача зайвими візуальними елементами.

Важливим нефункціональним аспектом є також безпека даних. Оскільки система працює з мережею підключених пристроїв, вона може бути потенційною цілью для кіберзагроз. Тому необхідно передбачити механізми шифрування переданих даних, автентифікації пристроїв та захисту від несанкціонованого доступу.

Не менш значущою є підтримка та супровід системи після її впровадження. Архітектура повинна передбачати можливість оновлення програмного забезпечення, діагностики несправностей та швидкого реагування на інциденти без повного зупинення роботи сервісу.

З урахуванням усіх вищезазначених вимог формується цілісне бачення майбутньої системи як складного багаторівневого рішення, що поєднує апаратні

засоби збору даних, програмні модулі обробки інформації та інтерфейси взаємодії з користувачем. Такий підхід дозволяє забезпечити гнучкість, адаптивність та високу ефективність функціонування системи в умовах реального міського середовища.

Виконання зазначених вимог дасть змогу створити робочий прототип системи, який може бути використаний для моніторингу стану сміттєвих контейнерів і підтримки роботи комунальних служб. Застосування такого рішення дозволить автоматизувати частину процесів збору відходів, зменшити кількість зайвих виїздів спецтехніки та підвищити обґрунтованість рішень щодо планування маршрутів.

2.2 Загальна структура апаратної реалізації кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина

Система моніторингу збору відходів є класичним прикладом кіберфізичної системи, яка інтегрує апаратну частину (сенсорні вузли на контейнерах), програмне забезпечення (серверна логіка маршрутизації) та веб-технології для автоматизації логістичних процесів комунальних служб. Структура цієї системи поєднує мікроконтролери та датчики вимірювання відстані з програмними рішеннями на базі REST API та інтерактивними користувацькими інтерфейсами.

При проектуванні структури системи було враховано вимоги: енергоефективність апаратних вузлів, надійність передачі даних, а також здатність до масштабування (додавання нових контейнерів без зміни архітектури). Структура спроектована так, щоб відповідати концепції «Розумного міста» (Smart City), залишаючись при цьому економічно доцільною для масового впровадження. Окрему увагу при проектуванні було приділено питанням модульності. Кожен програмний та апаратний компонент розглядається як незалежний об'єкт із чітко визначеним інтерфейсом взаємодії,

					КВРКІ.022056.22.02.12 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

що дозволяє спростити тестування, обслуговування та подальше розширення інфраструктури.

На початковому етапі розробки формується загальне бачення інформаційних потоків між фізичним середовищем та логічним рівнем обробки даних. Схематичне зображення загальної структури проєктованої системи наведено на рисунку 2.1.

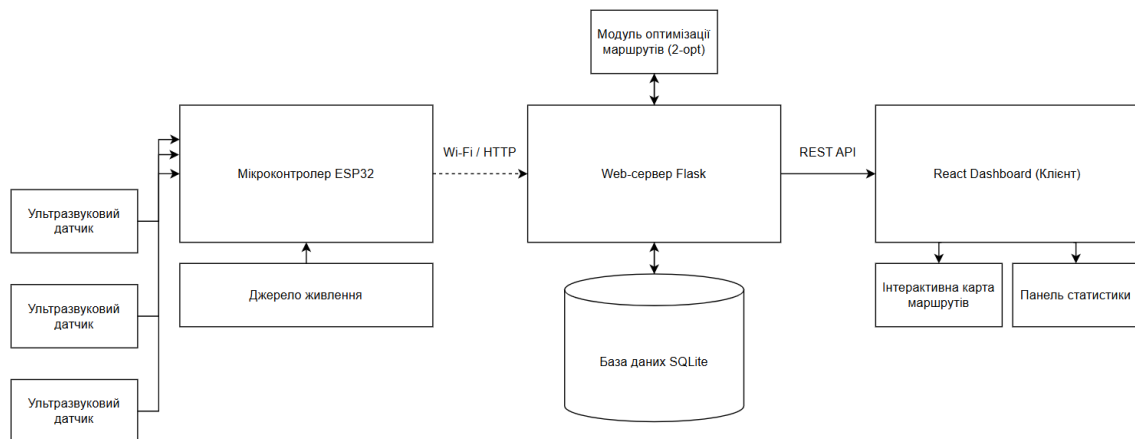


Рисунок 2.1 – Загальна структура проєктованої кіберфізичної системи

Інформаційні потоки в системі мають чітко визначений вектор: від фізичних пристроїв збору даних (сенсорів) до центрального обчислювального сервера і далі до користувацького веб-інтерфейсу. Аналізуючи цю базову архітектуру, можна виділити первинні складові: ультразвукові датчики для оцінки заповненості баків, мікроконтролери з підтримкою бездротових мереж, автономні джерела живлення та серверну частину з базою даних. Водночас можливий і зворотний канал зв'язку, який дозволяє віддалено налаштовувати параметри роботи сенсорних вузлів.

Проте, для глибокого розуміння того, як саме функціонує розроблений програмно-апаратний комплекс, загальну схему необхідно декомпонувати. Кіберфізична система даного типу будується за ієрархічною моделлю, де кожен рівень інкапсулює у собі певну логіку обробки інформації. Розширена



Зібрані сирі імпульси передаються на Edge-рівень (керування та зв'язок). Обчислювальним центром на цьому етапі є мікроконтролер NodeMCU V1.0 на базі чіпа ESP8266. Його завдання полягає не лише у зборі даних, а й у їх первинній фільтрації (відсіюванні хибних показників). Однією з функцій мікроконтролера є керування режимом глибокого енергозбереження (Deep Sleep). Пристрій "прокидається" лише за заданим розкладом, формує стандартизований JSON-пакет зі статусом бака та ініціює його відправку.

Далі інформація переходить на інтеграційний рівень (дані та сервіси). Передача телеметрії здійснюється через бездротову мережу Wi-Fi за протоколом HTTP (PUT запити). Основним компонентом цього рівня є реляційна база даних SQLite 3, обрана за її легковаговість та високу швидкість обробки транзакцій. У ній структуровано зберігається основна інформація: таблиця containers (фіксує поточний статус та географічні координати баків), history (накопичує історію заповнень для подальшої аналітики) та routes (архівує згенеровані логістичні маршрути).

База даних використовується на серверному рівні для збереження поточного стану контейнерів, історії заповнення та сформованих маршрутів. Серверну частину реалізовано мовою Python із використанням мікрофреймворку Flask. На цьому рівні виконуються приймання даних, їх обробка та розрахунок маршрутів. Окрім REST API для прийому даних від IoT-симулятора чи реальних контролерів, сервер виконує складні математичні обчислення. Саме тут працює алгоритм маршрутизації, який поєднує метод «найближчого сусіда» (Nearest Neighbor) із локальною оптимізацією 2-opt для побудови шляхів об'їзду без перетинів. Крім того, на цьому рівні функціонує модуль предиктивної аналітики, який на основі методів лінійної регресії розраховує швидкість накопичення відходів і прогнозує час досягнення критичної позначки у 80%.

Завершує ієрархію клієнтський рівень (презентаційний). Він є єдиною точкою взаємодії користувача (диспетчера або водія комунальної техніки) із системою. Цей рівень реалізовано у вигляді односторінкового веб-додатка (React

НМІ). Через API-запити фронтенд отримує оброблену інформацію та візуалізує її. Основний акцент зроблено на інтерактивній картографії (бібліотека Leaflet), де заповненість баків відображається кольоровими маркерами, а оптимальні маршрути виводяться у вигляді поліліній. Додатково диспетчер отримує доступ до панелі аналітики та статистики аудиту відходів у режимі реального часу.

Отже, запропонована багаторівнева структура кіберфізичної системи є оптимальною для вирішення задач логістики відходів. Чіткий розподіл обов'язків від безпосереднього збору первинних даних мікроконтролером до ресурсоємних математичних обчислень на сервері мінімізує затримки передачі інформації, гарантує стабільну роботу системи в умовах нестабільного зв'язку та забезпечує легкість в обслуговуванні і масштабуванні міської інфраструктури.

### 2.3 Схема кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина

Принципова електрична схема є базовим технічним документом, який відображає всі основні електричні з'єднання та фізичну взаємодію між апаратними компонентами системи. Вона дозволяє зрозуміти, яким чином передаються керуючі та інформаційні сигнали, а також як організовано автономне живлення апаратного вузла, що розміщується безпосередньо на сміттєвому контейнері.

Принципова схема не лише відображає топологію електричних з'єднань, але й виступає основою для подальшого проектування друкованої плати (PCB), вибору компонентів та аналізу енергетичних характеристик пристрою. Вона дозволяє інженеру оцінити коректність підключення ще до фізичної реалізації, що значно знижує ризик помилок на етапі виготовлення прототипу.

Крім того, на основі принципової схеми здійснюється моделювання роботи системи, включаючи аналіз споживання струму, перевірку стабільності напруги

					КВРКІ.022056.22.02.12 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

живлення та виявлення потенційних конфліктів між компонентами. Це особливо важливо для автономних IoT-пристроїв.

Головним обчислювальним компонентом сенсорного вузла є мікроконтролер NodeMCU на базі чіпа ESP8266 (на схемі позначений як U1). До нього підключено ультразвуковий датчик вимірювання відстані HC-SR04 (U2), який відповідає за фіксацію рівня заповненості бака. Оскільки сміттєві контейнери не мають доступу до стаціонарної електромережі, головною вимогою до схеми є забезпечення автономного живлення та жорстка оптимізація енергоспоживання.

Вибір платформи NodeMCU (ESP8266) обумовлений оптимальним співвідношенням вартості, функціональних можливостей та енергоспоживання. Даний мікроконтролер має вбудований Wi-Fi модуль, що дозволяє уникнути використання додаткових комунікаційних компонентів, зменшуючи складність схеми та її енергетичні витрати.

Окрім цього, ESP8266 підтримує широкий спектр програмних бібліотек і середовищ розробки (Arduino IDE, MicroPython), що спрощує процес розробки, тестування та подальшої модифікації прошивки пристрою. Це є важливим фактором у контексті швидкого прототипування кіберфізичних систем.

Для живлення системи використовується автономне джерело (BT1) літій-іонний акумулятор формату 18650. Позитивний контакт акумулятора підключено безпосередньо до входу 5V мікроконтролера (пін 15), а негативний до загальної шини GND (пін 14). Вбудований на платі NodeMCU лінійний стабілізатор напруги знижує вхідну напругу до 3.3V, які необхідні для стабільної роботи чіпа ESP8266.

Водночас пряме підключення акумулятора до входу 5V потребує врахування особливостей роботи літій-іонних елементів. Напруга повністю зарядженого акумулятора може досягати 4.2V, а при розряді знижуватись до приблизно 3.0V. Це може впливати на стабільність роботи як самого мікроконтролера, так і периферійних пристроїв.

					КВРКІ.022056.22.02.12 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

Для підвищення надійності системи доцільно використовувати додаткові модулі керування живленням, такі як DC-DC перетворювачі або контролери заряду/розряду (наприклад, TP4056), які забезпечують стабілізацію напруги та захист акумулятора від глибокого розряду або перезаряду.

Ультразвуковий датчик HC-SR04 живиться від тієї ж лінії 5V, оскільки для генерації потужного ультразвукового імпульсу йому потрібна підвищена напруга. Підключення датчика реалізовано наступним чином:

- контакт VCC (пін 1 датчика) з'єднаний з лінією живлення 5V (пін 15 мікроконтролера);
- контакт GND (пін 4 датчика) з'єднаний із загальною шиною GND (пін 17 мікроконтролера);
- контакт TRIG (пін 2 датчика), що відповідає за генерацію вимірювального імпульсу, підключений до цифрового виводу IO5 мікроконтролера (пін 29);
- контакт ECHO (пін 3 датчика), який приймає відбитий сигнал, підключений до цифрового виводу IO4 (пін 28).

Особливу увагу необхідно звернути на рівні логічних сигналів між датчиком HC-SR04 та мікроконтролером ESP8266. Оскільки датчик працює з логічним рівнем 5V, а ESP8266 з 3.3V, пряме підключення сигналу ECHO може призвести до пошкодження мікроконтролера.

Для уникнення цього ризику рекомендується використовувати діляник напруги (резистивний подільник) або логічний перетворювач рівнів (level shifter), який знижує рівень сигналу з 5V до безпечних 3.3V. Це підвищує довговічність пристрою та запобігає апаратним несправностям.

Для забезпечення тривалої роботи від акумулятора, мікроконтролер більшу частину часу перебуває в режимі глибокого сну (Deep Sleep), мінімізуючи споживання струму. Щоб ESP8266 міг самостійно «прокидатися» за вбудованим таймером для проведення чергового вимірювання, на електричній схемі реалізовано апаратну петлю: цифровий вивід IO16 (пін 30) з'єднано з піном

апаратного скидання RST (пін 1). Завдяки цьому, після закінчення заданого інтервалу часу, таймер реального часу надсилає сигнал пробудження, система зчитує показники датчика HC-SR04, відправляє дані на сервер через Wi-Fi та знову переходить у режим енергозбереження.

Режим глибокого сну є основним механізмом енергозбереження у даній системі. У цьому режимі споживання струму мікроконтролера може знижуватись до одиниць мікроамперів, що дозволяє значно продовжити час автономної роботи пристрою.

Однак використання Deep Sleep накладає певні обмеження, зокрема повне перезавантаження мікроконтролера після кожного пробудження. Це означає, що програма повинна бути оптимізована для швидкого запуску та виконання всіх необхідних операцій (зчитування даних, підключення до Wi-Fi, передача інформації) за короткий проміжок часу.

Усі описані підключення компонентів, лінії передачі даних та схема автономного живлення зображені на рисунку 2.3.

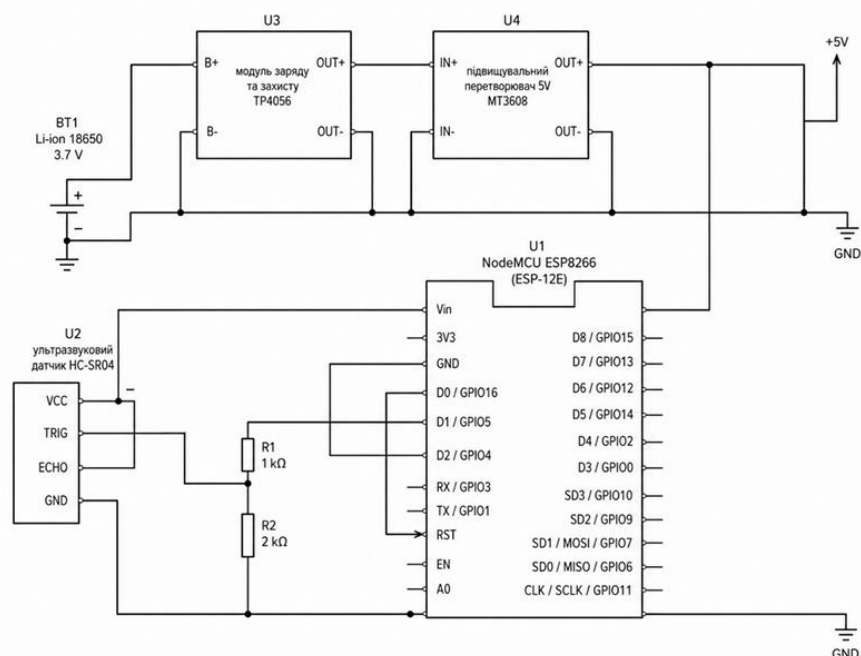


Рисунок 2.3 – Схема електрична принципова кіберфізичної системи моніторингу збору відходів

Зм.	Арк.	№ докум.	Підпис	Дата

На рисунку 2.3 компоненти системи об'єднані спільною шиною живлення та заземлення, що забезпечує коректну роботу електричного кола. При цьому особливу увагу слід приділити якості з'єднань та мінімізації паразитних опорів, які можуть впливати на стабільність роботи пристрою.

Розміщення компонентів на платі також відіграє важливу роль. Наприклад, датчик HC-SR04 повинен бути фізично віддалений від джерел електромагнітних завад, а лінії живлення максимально короткими для зменшення втрат напруги.

Крім того, доцільно передбачити використання фільтруючих конденсаторів (декаплінг-конденсаторів) між лініями живлення та землею. Вони згладжують пульсації напруги та забезпечують стабільну роботу мікроконтролера під час пікових навантажень, наприклад, при передачі даних через Wi-Fi.

Зважаючи на умови експлуатації (відкрите середовище, вплив вологи, пилу та температурних коливань), необхідно передбачити додаткові заходи захисту апаратного вузла. До них належать використання герметичних корпусів із класом захисту не нижче IP65, застосування захисних покриттів для друкованих плат (конформне покриття), а також встановлення захисту від перенапруги. Такі заходи дозволяють значно підвищити надійність системи та зменшити ймовірність виходу з ладу компонентів у процесі тривалої експлуатації.

Додатково варто відзначити, що запропонована електрична схема є достатньо універсальною та може бути модифікована залежно від конкретних умов використання. Наприклад, у разі необхідності збільшення дальності передачі даних може бути інтегровано GSM або LoRa модулі, що потребуватиме незначних змін у схемі живлення та комунікаційних інтерфейсах. Також можливе розширення функціоналу за рахунок підключення додаткових сенсорів, таких як датчики температури, вологості або газів, що дозволить створити більш комплексну систему моніторингу стану навколишнього середовища.

Запропоноване схемотехнічне рішення відзначається високою надійністю та мінімальною кількістю потенційних точок відмови, що є важливим для IoT-

пристроїв, які експлуатуються у складних зовнішніх умовах. Завдяки простоті підключення компонентів, такий апаратний вузол є економічно вигідним у масовому виробництві, легко монтується на будь-які типи сміттєвих баків і не потребує складного технічного обслуговування. Зрештою, саме ця компактна апаратна база виступає первинним джерелом даних для всієї кіберфізичної системи, перетворюючи фізичну відстань до відходів у цифрові метрики, які серверна частина використовує для генерації оптимальних логістичних маршрутів.

2.4 Монтажна схема кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина

Під час проєктування кіберфізичної системи важливим кроком є створення монтажно-схематичного зображення, на якому показані всі електронні компоненти, що використовуються в системі, зокрема мікроконтролери, датчики, джерела живлення тощо. На схемі ці компоненти зображені у вигляді реальних модулів або їхніх спрощених аналогів. Вона надає чітке уявлення про фізичне з'єднання компонентів, їхнє реальне розташування та спосіб підключення.

Монтажна схема, на відміну від принципової, орієнтована не стільки на аналіз електричних процесів, скільки на практичну реалізацію пристрою. Вона демонструє реальне фізичне розташування компонентів, довжину та прокладання провідників, а також дозволяє оцінити зручність складання та обслуговування системи.

Особливо важливою монтажна схема є на етапі прототипування, коли інженер має можливість швидко змінювати конфігурацію з'єднань, тестувати різні варіанти підключення та оптимізувати розташування компонентів з урахуванням ергономіки та мінімізації перешкод.

					КВРКІ.022056.22.02.12 ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

Головною метою монтажної схеми є надання зрозумілого візуального плану, який дозволяє уникнути помилок під час складання. Завдяки монтажній схемі можна швидко зібрати пристрій і протестувати його. У разі виявлення проблеми схема допомагає легко відслідкувати підключення і знайти місце помилки або несправності.

Крім цього, монтажна схема дозволяє оцінити потенційні проблеми, пов'язані з наведеннями, паразитними ємностями та електромагнітними завадами. Неправильне прокладання провідників або надмірна довжина з'єднань може призвести до нестабільної роботи датчиків або мікроконтролера.

Також вона є важливим документом для командної роботи, оскільки забезпечує однакове розуміння структури пристрою всіма учасниками проекту, включаючи розробників апаратної частини, програмістів та технічний персонал.

При проектуванні монтажної схеми апаратного вузла розумного сміттєвого контейнера використана стандартна безпайкова макетна плата (Breadboard), яка дозволяє створити загальні шини живлення та зручно об'єднати всі компоненти. Для забезпечення автономної роботи системи використовується літій-іонний акумулятор формату 18650 (на схемі представлений блоком живлення), позитивний і негативний контакти якого підключені відповідно до нижніх шин живлення (+) та (-) макетної плати.

Використання безпайкової макетної плати має ряд переваг, зокрема можливість багаторазового використання та швидкої зміни конфігурації з'єднань без застосування паяльного обладнання. Це значно пришвидшує процес розробки та знижує ризик пошкодження компонентів.

Однак варто враховувати, що breadboard має і певні обмеження зокрема, менш надійні контакти та більший опір з'єднань у порівнянні з паяними платами. Тому така реалізація використовується виключно на етапі прототипування, тоді як для фінального продукту необхідно розробити друковану плату.

Від цих шин отримує робочу напругу головний обчислювальний елемент системи мікроконтролер NodeMCU V1.0 (піни Vin та GND), а також

ультразвуковий датчик вимірювання відстані HC-SR04 (піни VCC та GND). Для передачі керуючих та інформаційних сигналів контакти датчика з'єднані з мікроконтролером: пін генерації ультразвукового імпульсу Trig підключено до цифрового виводу мікроконтролера D1 (GPIO5), а пін прийому відбитого сигналу Echo до виводу D2 (GPIO4).

Особливістю цієї монтажної схеми є апаратна реалізація режиму глибокого енергозбереження (Deep Sleep): цифровий пін D0 (GPIO16) з'єднано окремим провідником із піном апаратного скидання RST. Завдяки цьому мікроконтролер має змогу автоматично пробуджуватися за таймером для проведення вимірювань рівня відходів, що важливо для тривалої роботи пристрою від акумулятора. Монтажу схему апаратного вузла кіберфізичної системи моніторингу збору відходів відображено на рисунку 2.4.

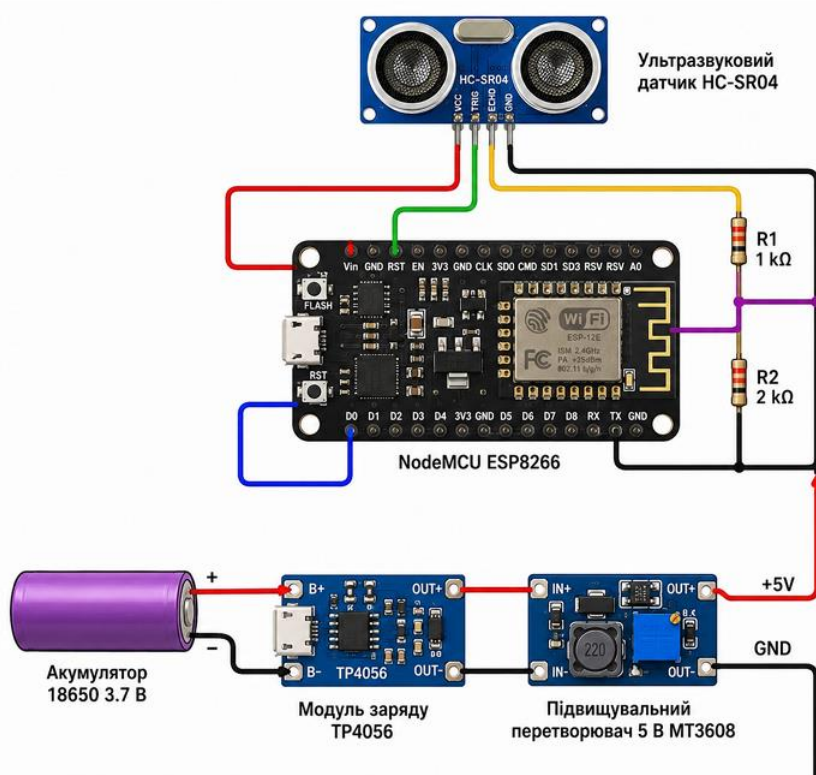


Рисунок 2.4 – Монтажна схема апаратного вузла кіберфізичної системи моніторингу збору відходів

Зм.	Арк.	№ докум.	Підпис	Дата

Компоненти розташовані таким чином, щоб мінімізувати довжину сигнальних ліній та уникнути їх перетину. Це дозволяє знизити рівень електромагнітних завад та підвищити стабільність передачі сигналів між датчиком і мікроконтролером.

Крім того, при розробці монтажною схеми враховано зручність подальшого фізичного розміщення пристрою всередині контейнера. Компактне компонування дозволяє інтегрувати електроніку в захисний корпус без значного збільшення габаритів. Окрему увагу приділено організації живлення використання спільних шин забезпечує простоту підключення та зменшує кількість провідників, що, у свою чергу, знижує ризик помилок під час складання.

Важливим етапом після створення монтажною схеми є її практична перевірка шляхом складання прототипу та проведення серії тестувань. Це дозволяє виявити можливі недоліки, такі як нестабільні контакти, неправильні з'єднання або некоректна робота окремих компонентів.

Результати тестування використовуються для подальшої оптимізації схеми перед переходом до етапу розробки друкованої плати, що є завершальним кроком у створенні апаратної частини системи.

Розроблене візуальне компонування монтажною схеми відзначається ергономічністю та логічною структурованістю фізичних з'єднань. Використання безпайкової макетної плати на етапі прототипування забезпечує високу гнучкість розробки, дозволяючи швидко вносити апаратні зміни та проводити діагностику окремих модулів без ризику їх пошкодження. Запропонований варіант фізичної збірки гарантує безперешкодну перевірку працездатності алгоритмів зчитування даних і переходу в режим сну перед етапом створення фінальної друкованої плати та остаточним інтегруванням сенсорного вузла у конструкцію сміттевого контейнера.

## 2.5 Аналіз обраних рішень

Завершальним етапом проєктування апаратного забезпечення кіберфізичної системи моніторингу збору відходів є комплексне оцінювання обраної елементної бази, її апаратної сумісності та застосованих архітектурних підходів. Враховуючи специфіку експлуатації розумних сміттєвих контейнерів зокрема, роботу в умовах змінного середовища та відсутність доступу до стаціонарної електромережі обґрунтований вибір мікроконтролера і сенсорів стає критичним фактором для забезпечення довговічності та відмовостійкості автономного вузла.

Аналіз обраних рішень охоплює не лише технічні характеристики окремих компонентів, але й оцінку їх взаємодії в межах єдиної системи. Важливим аспектом є узгодженість параметрів, таких як рівні напруги, швидкість передачі даних та енергоспоживання. Крім того, оцінюється економічна доцільність впровадження системи, що включає вартість компонентів, витрати на монтаж та обслуговування, а також потенційний ефект від оптимізації логістичних процесів.

З іншого боку, коректно спроектовані програмні та логічні рішення гарантують безперебійну взаємодію вимірювальних приладів з обчислювальним ядром. Вони дозволяють ефективно управляти режимами енергоспоживання (наприклад, через циклічне використання режиму Deep Sleep) та забезпечують надійну бездротову маршрутизацію зібраних метрик на віддалений сервер для подальшої аналітики.

Особливу увагу приділено питанням масштабованості системи. Обрані технічні рішення повинні забезпечувати можливість розгортання великої кількості сенсорних вузлів без значного ускладнення інфраструктури. Також враховано сумісність із сучасними стандартами IoT, що дозволяє інтегрувати систему з іншими інформаційними платформами у межах концепції «розумного міста».

					КвРКІ.022056.22.02.12 ПЗ	Арк. 33
Зм.	Арк.	№ докум.	Підпис	Дата		

Проведення такого аналізу дає змогу об'єктивно перевірити, наскільки імплементовані технічні рішення задовольняють початкові функціональні та нефункціональні вимоги до системи. Крім того, це дозволяє виявити "вузькі місця" прототипу та окреслити перспективи для подальшого масштабування або модернізації розробленого IoT-пристрою.

### 2.5.1 Аналіз обраних апаратних рішень та їх технічних характеристик

Для забезпечення повноцінного та безперебійного функціонування проектованої кіберфізичної системи моніторингу збору відходів, відповідно до поставлених вимог, був сформований наступний перелік базових апаратних компонентів: мікроконтролер як головний обчислювальний та комунікаційний центр, ультразвуковий далекомір для безконтактного вимірювання рівня заповненості контейнера, а також автономне джерело живлення для забезпечення незалежної роботи вузла.

Вибір мікроконтролера є одним із важливих етапів апаратного проектування, оскільки він визначає архітектуру всієї системи та накладає обмеження на вибір периферії. Головною метою був пошук платформи, яка поєднує енергоефективність, достатню обчислювальну потужність та інтегровані мережеві можливості. Виходячи з цього, до мікроконтролера були висунуті наступні критерії:

- наявність вбудованого Wi-Fi модуля, важливо для прямої передачі метрик на хмарний сервер без використання додаткових мережевих шлюзів;
- апаратна підтримка режимів енергозбереження (Deep Sleep), обов'язкова умова для максимізації терміну служби пристрою від одного заряду акумулятора;
- достатня кількість цифрових виводів (GPIO), для підключення вимірювального датчика та організації апаратної петлі пробудження;

– низька вартість та доступність, для забезпечення економічної доцільності масового впровадження системи на об'єктах міської інфраструктури.

Додатковим критерієм вибору є підтримка стабільної роботи в умовах змінних температур та підвищеної вологості, що є типовими для середовища експлуатації сміттєвих контейнерів. Надійність компонентів у таких умовах безпосередньо впливає на довговічність системи.

У результаті детального порівняльного аналізу наявних на сучасному ринку апаратних рішень, оптимальним вибором для реалізації сенсорного вузла системи моніторингу відходів стала відлагоджувальна плата NodeMCU V1.0, побудована на базі високоінтегрованого Wi-Fi чіпа ESP8266 [33]. Рішення зупинитися саме на готовому форматі відлагоджувальної плати (development board), а не на використанні базового мікромодуля (наприклад, ESP-12E) у чистому вигляді, обґрунтоване значним спрощенням процесу апаратного прототипування та підвищенням загальної надійності системи. Така висока ступінь інтеграції дозволяє уникнути розробки додаткових проміжних плат для узгодження живлення, забезпечуючи при цьому компактність кінцевого пристрою, що монтується на сміттєвому контейнері. Зовнішній вигляд обраної мікроконтролерної плати відображено на рисунку 2.5.

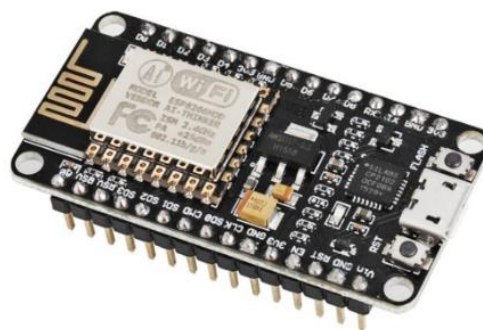


Рисунок 2.5 – Плата NodeMCU ESP8266 [51]

Незважаючи на свої переваги, ESP8266 має і певні обмеження, зокрема відносно невелику кількість GPIO та відсутність апаратної підтримки деяких інтерфейсів. Проте для задачі моніторингу одного сенсора ці обмеження не є критичними. У разі потреби розширення функціоналу (наприклад, підключення декількох датчиків) можливим є перехід на більш потужні платформи, такі як ESP32, що забезпечують більшу кількість периферійних інтерфейсів.

Дана платформа повністю задовольняє висунуті вимоги та є поширеним форматом для створення бюджетних IoT-рішень. Обчислювальним ядром ESP8266 виступає високопродуктивний 32-розрядний мікропроцесор Tensilica Xtensa L106. Його базова тактова частота становить 80 МГц із можливістю програмного розгону до 160 МГц. Обсяг інтегрованої оперативної пам'яті складає 64 КБ для інструкцій та 96 КБ для даних, а об'єм зовнішньої флеш-пам'яті на платі NodeMCU зазвичай становить 4 МБ, чого з великим запасом вистачає для зберігання мікропрограми.

Інтегрований радіомодуль підтримує стандарти бездротового зв'язку IEEE 802.11 b/g/n у діапазоні 2,4 ГГц. Контролер здатен функціонувати у двох режимах: як точка доступу (SoftAP) та як клієнтська станція (Station). У рамках даного проєкту використовується режим Station, що дозволяє пристрою авторизуватися у наявній Wi-Fi мережі та встановлювати TCP/IP з'єднання з віддаленим сервером.

Для зручності програмування та налагодження плата NodeMCU оснащена мікросхемою перетворювача інтерфейсів USB-UART (найчастіше CP2102 або CH340). Цей чіп дозволяє підключати мікроконтролер безпосередньо до USB-порту комп'ютера без використання зовнішніх програматорів, автоматично організовуючи віртуальний COM-порт.

Плата має розширену колодку контактів, серед яких доступні виводи загального призначення (GPIO). Для реалізації нашої системи залучені цифрові пінні D1 (GPIO5) та D2 (GPIO4) для комунікації з датчиком, а також пін D0

					КВРКІ.022056.22.02.12 ПЗ	Арк. 36
Зм.	Арк.	№ докум.	Підпис	Дата		



інтеграції з мікроконтролером. Зовнішній вигляд датчика та розташування його основних елементів наведено на рисунку 2.7.



Рисунок 2.7 – Ультразвуковий датчик відстані HC-SR04 [53]

Попри високу точність, датчик HC-SR04 має обмеження у використанні в умовах сильного забруднення або наявності перешкод, які можуть частково поглинати або розсіювати ультразвукові хвилі. Це необхідно враховувати при інтерпретації отриманих даних. Для підвищення достовірності вимірювань може застосовуватись програмна фільтрація або комбінування даних із декількох вимірювань.

Принцип дії HC-SR04 базується на явищі ехолокації. Мікроконтролер подає короткий імпульс (не менше 10 мкс) на вхід Trig датчика. У відповідь п'єзоелектричний випромінювач генерує пакет з восьми ультразвукових хвиль на частоті 40 кГц. Коли ці хвилі досягають поверхні відходів, вони відбиваються і повертаються до приймача. Датчик фіксує цей момент і генерує на виводі Echo логічну одиницю, тривалість якої прямо пропорційна часу подорожі звукової хвилі. Мікроконтролер вимірює цю тривалість і, знаючи швидкість звуку в повітрі (приблизно 343 м/с), обчислює точну відстань до поверхні сміття.

Зм.	Арк.	№ докум.	Підпис	Дата

Короткі технічні характеристики датчика HC-SR04:

- робоча напруга: 5 В постійного струму;
- споживаний струм у режимі очікування: < 2 мА;
- ефективний кут вимірювання: 15 градусів;
- діапазон вимірюваних відстаней: від 2 см до 400 см;
- точність вимірювання: до 0.3 см;
- габарити: 45 x 20 x 15 мм.

Оскільки кіберфізична система монтується безпосередньо на сміттєвих контейнерах, які не мають доступу до стаціонарної електромережі, важливим є забезпечення надійного автономного живлення. Як джерело енергії було обрано промисловий літій-іонний (Li-Ion) акумулятор формату 18650 [36]. Назва цього формату безпосередньо відображає його фізичні габарити: акумулятор має форму циліндра з діаметром 18 мм та довжиною 65 мм. Цей тип елементів живлення став індустріальним стандартом для автономної електроніки та IoT-систем завдяки високій щільності енергії, здатності віддавати стабільний струм при різних температурних режимах, відсутності вираженого "ефекту пам'яті" та низькому рівню саморозряду. Зовнішній вигляд акумулятора формату 18650 зображено на рисунку 2.8.



Рисунок 2.8 – Літій-іонний акумулятор формату 18650 [54]

Номинальна напруга одного осередку 18650 становить 3.7 В, а при повному заряді досягає 4.2 В. Цієї напруги, поданої на вхідний пін VIN, цілком достатньо для коректної роботи лінійного стабілізатора на платі NodeMCU. Типова ємність таких акумуляторів варіюється від 2000 до 3500 мА·год. Враховуючи, що абсолютну більшість часу мікроконтролер перебуватиме у режимі Deep Sleep (споживаючи лише десятки мікроампер струму), одного заряду такого акумулятора вистачить на тривалий термін автономної експлуатації системи без необхідності технічного обслуговування.

Важливим аспектом використання літій-іонних акумуляторів є забезпечення їх безпечної експлуатації. Для цього необхідно застосовувати спеціалізовані плати захисту (BMS), які контролюють рівень заряду, запобігають короткому замиканню та перегріву.

Крім того, необхідно враховувати природний процес деградації акумулятора, що призводить до зменшення його ємності з часом. Це може впливати на тривалість автономної роботи системи та потребує періодичної заміни джерела живлення.

### 2.5.2. Аналіз обраних програмних рішень

Під час розробки програмної частини кіберфізичної системи моніторингу збору відходів, важливим етапом є вибір оптимального стеку технологій. Від архітектурних рішень залежить не лише стабільність прийому телеметрії від апаратних вузлів, але й ефективність роботи складних математичних алгоритмів оптимізації маршрутів та прогнозування [37].

Під час вибору програмних інструментів були сформовані наступні критерії:

- модульність та масштабованість, система повинна мати чітко розділену клієнт-серверну архітектуру (Backend та Frontend) для незалежного тестування та оновлення модулів;

- математична продуктивність, серверна частина має швидко обробляти часові ряди та виконувати ресурсоємні алгоритми просторової оптимізації (розв'язання задачі комівояжера);
- реактивність інтерфейсу, диспетчерський веб-додаток повинен миттєво оновлювати інформацію на карті без перезавантаження сторінки при надходженні нових даних від контейнерів;
- простота інтеграції, наявність зрозумілого програмного інтерфейсу (API) для зв'язку між IoT-датчиками, базою даних та клієнтським додатком.

Після аналізу доступних технологій розробку було вирішено базувати на класичній мікросервісній архітектурі з використанням Python для серверної логіки та React.js для клієнтського інтерфейсу [38].

Основним інструментом для реалізації бекенду було обрано мову програмування Python та мікрофреймворк Flask [39]. Вибір Python є беззаперечним лідером для задач Data Science та алгоритмічної оптимізації [40]. Це дозволило ефективно реалізувати бізнес-логіку системи: алгоритми маршрутизації сміттєвозів (Nearest Neighbor у поєднанні з локальною оптимізацією 2-opt) та модуль предиктивної аналітики (лінійна інтерполяція для прогнозування часу заповнення баків) [41].

Використання фреймворку Flask із застосуванням патернів Application Factory та Blueprints забезпечило створення чіткої структури REST API додатку, де бізнес-логіка (services) повністю ізольована від маршрутизаторів запитів (routes) [42]. Для збереження даних (координат, історії заповнення, згенерованих маршрутів) інтегровано реляційну базу даних SQLite 3. Її вибір обґрунтований легкістю розгортання та відсутністю потреби в окремому сервері баз даних, при цьому вона забезпечує достатню швидкодію для роботи з індексованими часовими рядами [43].

Для розробки інтерактивного веб-інтерфейсу диспетчера використано бібліотеку React.js у поєднанні зі швидкісним збирачем модулів Vite [44]. React дозволяє побудувати реактивний користувацький інтерфейс на основі

ізолюваних компонентів (Header, Map, Statistics) [45]. Важливою перевагою стало використання користувацьких хуків (Custom Hooks, зокрема useContainers, useRoute), які автоматизують процес опитування серверного API та оновлення стану додатку в режимі реального часу [46].

Для візуалізації геопросторових даних інтегровано JavaScript-бібліотеку Leaflet (через обгортку react-leaflet) [47]. Вона дозволяє динамічно відобразити координати контейнерів на інтерактивній мапі міста за допомогою кольорових маркерів (залежно від рівня заповненості) та малювати оптимізовані полілінії маршрутів для комунальної техніки [48].

На відміну від систем, що покладаються на брокери повідомлень типу MQTT, зв'язок між апаратним (IoT) рівнем та сервером у даному проєкті побудовано на базі синхронної REST API архітектури [49]. Апаратний вузол (робота якого на етапі розробки імітується програмним симулятором simulation.py) передає телеметричні дані про рівень заповнення бака за допомогою стандартних HTTP PUT запитів. Таке рішення значно спрощує архітектуру системи, зменшує кількість проміжних вузлів та дозволяє серверу миттєво валідувати вхідні дані перед їх записом до бази [50].

## 2.6 Висновки до другого розділу

Під час апаратної реалізації системи було визначено її функціональні вимоги та розроблено загальну структурну концепцію (сенсорний, серверний і клієнтський рівні), орієнтовану на енергоефективність та оптимізацію логістики збору відходів. Створено принципову й монтажну схеми, що наочно відображають взаємодію мікроконтролера NodeMCU, датчика HC-SR04 та елемента живлення з підтримкою режиму глибокого сну (Deep Sleep). Загальний аналіз обраного апаратного та програмного забезпечення підтвердив його надійність і повну відповідність цілям проєкту.

					КвРКІ.022056.22.02.12 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

### 3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ «РОЗУМНІ СМІТТЄВІ КОНТЕЙНЕРИ» ДЛЯ МОНІТОРИНГУ ЗБОРУ ВІДХОДІВ У РЕЖИМІ РЕАЛЬНОГО ЧАСУ. ПРОГРАМНА ЧАСТИНА

3.1 Огляд використовуваних програмних бібліотек та їх функціональне призначення

Під час написання програмного коду для кіберфізичної системи моніторингу збору відходів було використано низку програмних бібліотек та фреймворків, що забезпечують коректну роботу серверної логіки, обробку просторових даних та побудову інтерактивного диспетчерського інтерфейсу. Правильний вибір технологічного стеку дозволяє ефективно реалізувати архітектуру системи, маршрутизацію телеметрії, складні математичні обчислення логістичних маршрутів та їх візуалізацію на карті. Оскільки програмна частина має чіткий поділ на серверну (Backend) та клієнтську (Frontend) частини, використовувані інструменти класифіковано за їхніми рівнями абстракції.

Для реалізації серверної частини (Backend) основним інструментом є мікрофреймворк Flask. Ця бібліотека забезпечує розгортання ядра веб-додатка за патерном Application Factory та дозволяє зручно організувати REST API через механізм Blueprints (маршрутизатори запитів). Flask приймає телеметрію від апаратних вузлів через HTTP-запити та передає її на обробку ізольованим модулям бізнес-логіки.

Для взаємодії з базою даних використовується інтегрована бібліотека sqlite3. Вона забезпечує виконання CRUD-операцій у реляційній базі даних waste\_system.db. Використання саме цієї бібліотеки дозволяє системі швидко зберігати історію заповненості баків, оновлювати актуальні статуси та генерувати логістичні маршрути без необхідності розгортання окремого важкого

					КВРКІ.022056.22.02.12 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		

сервера баз даних, зберігаючи при цьому достатню швидкість завдяки внутрішнім індексам.

У підсистемі генерації даних (модуль `simulation.py`), яка імітує роботу фізичних мікроконтролерів, активно задіяна бібліотека `requests`. Її функціональне призначення полягає у формуванні та відправці HTTP PUT-запитів до API-сервера у форматі JSON. Це дозволяє повноцінно симулювати передачу показників з ультразвукових датчиків у режимі реального часу.

Для програмної реалізації алгоритмів маршрутизації (Nearest Neighbor, 2-орт оптимізація) та предиктивної аналітики (лінійна регресія) були використані стандартні бібліотеки обчислень `math` та `datetime`. Бібліотека `math` забезпечує точні математичні інструменти для обчислення реальних географічних відстаней між контейнерами за формулою Гаверсина (елементи сферичної тригонометрії). Бібліотека `datetime` є важливою для розрахунку часових дельт при обчисленні швидкості заповнення бака (Fill Rate) та подальшої екстраполяції часу до його переповнення.

Для забезпечення безпечної крос-доменної взаємодії між клієнтською (React) та серверною (Flask) частинами системи була інтегрована бібліотека `Flask-CORS`. Оскільки архітектура проєкту передбачає роботу фронтенду та бекенду на різних портах під час розробки та розгортання, ця бібліотека налаштовує механізм Cross-Origin Resource Sharing, дозволяючи браузеру безпечно виконувати REST API запити до сервера. Паралельно з цим, для гнучкого управління конфігураціями та захисту конфіденційних даних середовища (наприклад, параметрів бази даних чи секретних ключів) застосовано бібліотеку `python-dotenv`, яка автоматично завантажує змінні середовища з конфігураційного файлу.

Окремої уваги заслуговує програмна реалізація модуля предиктивної аналітики (`prediction.py`). Для ефективної обробки масивів телеметричних даних та виконання багатовимірних математичних операцій над часовими рядами залучено бібліотеку `NumPy`. Вона дозволяє векторизувати обчислення історії

заповнень баків, що значно пришвидшує роботу алгоритму лінійної регресії. Цей математичний апарат лежить в основі прогнозування: система вираховує швидкість накопичення відходів (Fill Rate) та екстраполює ці дані для точного визначення часу досягнення порогу заповненості у 80%.

Для вирішення логістичної задачі комівояжера (TSP) та програмної реалізації алгоритму 2-opt у модулі оптимізації маршрутів (route\_optimizer.py) активно використовуються вбудовані модулі Python, зокрема itertools та copy. Бібліотека itertools забезпечує генерацію оптимальних комбінацій для жадібного алгоритму «найближчого сусіда» (Nearest Neighbor), а модуль copy використовується для глибокого копіювання структур даних під час ітеративного пошуку локальних оптимумів. Це дозволяє уникнути мутацій вихідних даних маршруту та суттєво знижує споживання оперативної пам'яті під час перерахунку складних багатоточкових маршрутів об'їзду.

Клієнтська частина (Frontend) базується на бібліотеці React. Це основний інструмент для побудови реактивного користувацького інтерфейсу (SPA), який дозволяє структурувати диспетчерську панель у вигляді незалежних компонентів (Header, Map, Controls, Statistics). Важливою перевагою цієї бібліотеки є підтримка користувацьких хуків (Custom Hooks, зокрема useContainers, useRoute), що автоматизують періодичне опитування серверного API та гарантують оновлення стану застосунку в режимі реального часу без перезавантаження сторінки.

Для візуалізації геопросторових даних невід'ємним інструментом стала JavaScript-бібліотека Leaflet та її React-обгортка react-leaflet. Вони забезпечують рендеринг інтерактивної карти міста, прив'язку контейнерів до географічних координат (широти та довготи), маркування їх відповідними кольорами залежно від статусу критичності, а також відмальовування геометричних поліліній для згенерованих маршрутів смітєвозів.

Для забезпечення швидкого процесу збирання та розробки фронтенд-модуля було використано інструмент Vite. На відміну від класичних бандлерів,

він гарантує миттєве оновлення коду в браузері (Hot Module Replacement) під час розробки та ефективну мініфікацію кінцевих CSS/JS файлів перед розгортанням у робоче середовище.

На клієнтському рівні для організації асинхронної взаємодії з REST API сервером застосовується бібліотека Axios (або надійно налаштований Fetch API, інкапсульований у модулі `services/api.js`). Цей інструмент дозволяє зручно формувати HTTP-запити до серверних маршрутизаторів, обробляти JSON-відповіді та автоматично перехоплювати помилки мережі (наприклад, у разі тимчасової втрати зв'язку із сервером). Використання єдиного сервісного шару для API-комунікації робить код компонентів чистішим і стійкішим до змін у серверній архітектурі.

Для стилізації диспетчерського веб-інтерфейсу задіяно підхід CSS Modules. На відміну від класичного використання глобальних таблиць стилів, ця технологія дозволяє локалізувати CSS-класи в межах конкретних React-компонентів (наприклад, `Map.module.css` або `Controls.module.css`). Це повністю виключає ризик конфліктів імен класів при масштабуванні інтерфейсу, полегшує підтримку коду та дозволяє ефективно впроваджувати сучасні візуальні ефекти, такі як адаптивний дизайн (Responsive Design) та елементи Glassmorphism.

Таким чином, використання зазначеного набору сучасних програмних бібліотек повністю закриває потреби розробки кіберфізичної системи моніторингу відходів, забезпечуючи її високу продуктивність, обчислювальну точність алгоритмів та інтуїтивну зручність диспетчерського керування.

### 3.2 Розробка програмного коду для моніторингу збору відходів у режимі реального часу

Розробка програмного коду для кіберфізичної системи це один з найважливіших та найскладніших етапів проєктування, адже саме тут формується основна бізнес-логіка роботи системи, здійснюється інтеграція

апаратних протоколів передачі даних із серверними базами даних та реалізуються математичні алгоритми оптимізації. Процес написання коду розділений на ізольовані модулі: конфігурація середовища, серверне ядро (Backend) та модуль генерації телеметрії.

Написання програмного коду серверної частини починається із підключення необхідних бібліотек мови Python, які були детально проаналізовані у попередньому підрозділі. Після імпорту базових пакетів (таких як `os`, `json`, `requests` для роботи з мережею та `Flask`, `sqlite3` для веб-ядра), відбувається ініціалізація та задання значень для системним константам. Для зручного та безпечного управління налаштуваннями додатку в різних середовищах (розробка, тестування, робоче середовище) створюється окремий клас конфігурації у файлі `config.py`. Фрагмент коду з ініціалізацією необхідних параметрів виглядає наступним чином:

```
import os

class Config:
    SECRET_KEY = os.environ.get('SECRET_KEY') or 'super-secret-key'
    DATABASE_URI =
os.path.join(os.path.abspath(os.path.dirname(__file__)),
'waste_system.db')
    API_BASE_URL = 'http://localhost:5001/api'
    SIMULATION_INTERVAL = 10
    CRITICAL_FILL_LEVEL = 80
```

У даному фрагменті константа `SECRET_KEY` використовується для криптографічного захисту сесій Flask-дodatка; система спочатку намагається отримати її із захищених змінних середовища операційної системи (`os.environ.get`), а у разі їх відсутності використовує базовий рядок. Змінна `DATABASE_URI` динамічно формує абсолютний шлях до файлу реляційної бази даних SQLite (`waste_system.db`), що гарантує коректну роботу системи незалежно від директорії запуску скрипта. Змінна `API_BASE_URL` містить базову адресу серверного інтерфейсу (REST API), до якого будуть звертатися апаратні вузли для передачі телеметрії. Числові константи `SIMULATION_INTERVAL` та

CRITICAL\_FILL\_LEVEL задають паузу між циклами отримання даних (10 секунд) та порогове значення (80%), при досягненні якого бак класифікується як "критичний" і передається алгоритмам маршрутизації.

Після налаштування конфігурацій відбувається ініціалізація головного веб-додатка за допомогою архітектурного патерну Application Factory. У файлі app.py створюється функція create\_app(), яка інкапсулює логіку створення екземпляра сервера. Блок-схему алгоритму ініціалізації веб-додатка та створення бази даних відображено на рисунку 3.1.

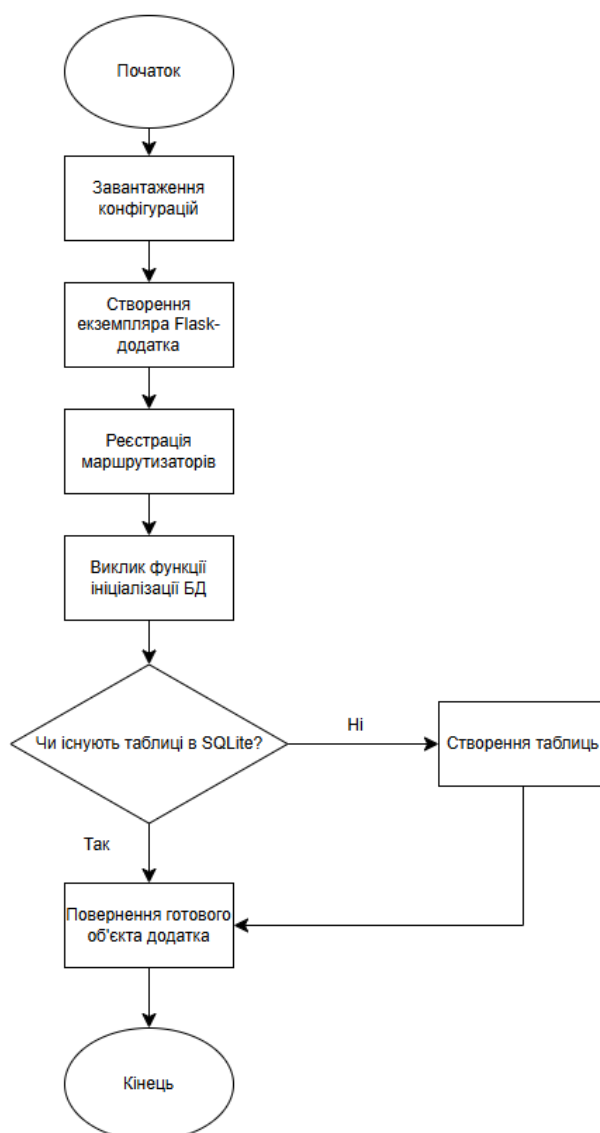


Рисунок 3.1 – Блок-схема функції ініціалізації веб-додатка та бази даних

Алгоритм, зображений на рисунку 3.1, ілюструє повний життєвий цикл ініціалізації серверної частини кіберфізичної системи (Backend) від моменту запуску головного скрипта до переходу в стан готовності обробки мережових запитів. Процес розпочинається із завантаження системних конфігурацій. На цьому етапі програма зчитує змінні середовища, встановлює параметри безпеки (наприклад, SECRET\_KEY для захисту сесій), визначає абсолютні шляхи до файлів бази даних та задає глобальні константи, такі як порогові значення критичного заповнення сміттєвих баків.

Наступним кроком є створення базового екземпляра Flask-додатка. Використання патерну Application Factory дозволяє інкапсулювати налаштування сервера, що робить систему більш гнучкою та підготовленою до масштабування чи модульного тестування. Далі відбувається реєстрація маршрутизаторів (Blueprints). Цей етап забезпечує архітектурний поділ програмних інтерфейсів: створюються окремі логічні зони для обробки REST API-запитів від апаратних IoT-модулів (прийом телеметрії) та відокремлені маршрути для обслуговування запитів від диспетчерського веб-інтерфейсу (віддача аналітики та побудованих маршрутів).

Після успішного налаштування мережових параметрів алгоритм переходить до підготовки підсистеми зберігання даних, викликаючи функцію ініціалізації бази даних. Оскільки система використовує легковагову реляційну базу SQLite, програма повинна переконатися в цілісності структур даних перед початком запису. Для цього алгоритм перевіряє логічну умову: чи існують вже необхідні таблиці у файлі бази даних. У випадку запуску системи на новому середовищі або після очищення даних система автоматично виконує серію DDL-скриптів для створення структури таблиць: containers (для зберігання географічних координат та поточного статусу), history та routes (для збереження згенерованих логістичних шляхів). Якщо ж таблиці вже існують (відгалуження «Так»), що є стандартною ситуацією при перезапуску сервера, цей ресурсоємний крок пропускається для збереження накопиченої історичної телеметрії.

Оскільки на етапі прототипування та тестування логістичних алгоритмів (зокрема 2-орт) потрібна наявність безперервного потоку вхідних телеметричних даних від великої кількості точок, паралельно з сервером запускається розроблений IoT-симулятор (скрипт simulation.py). Він повністю імітує мережеву поведінку фізичних мікроконтролерів ESP8266, самостійно генеруючи показники рівня заповненості та відправляючи їх на сервер. Логіку роботи цього модуля наведено у вигляді блок-схеми на рисунку 3.2.

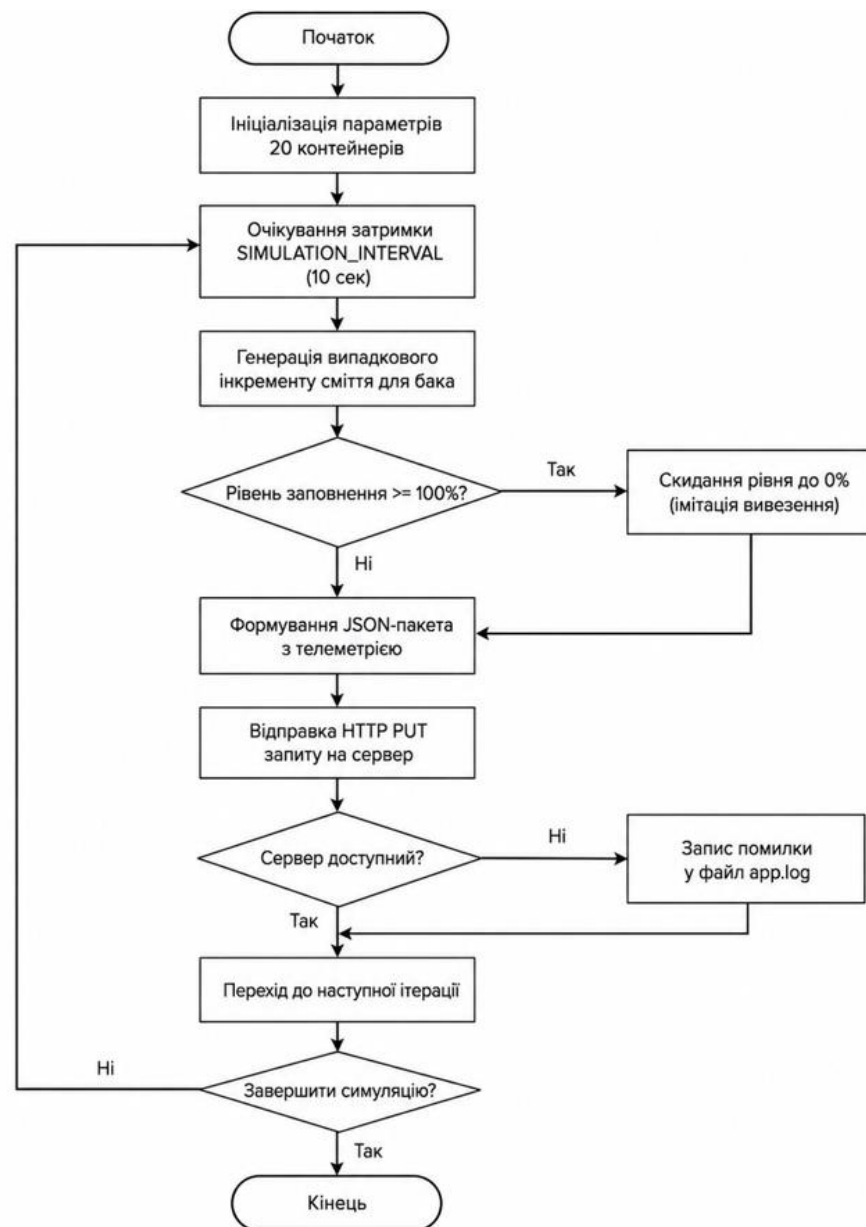


Рисунок 3.2 – Блок-схема функції роботи IoT-симулятора

Відповідно до розробленої блок-схеми, наведеної на рисунку 3.2, програмна реалізація модульного IoT-симулятора базується на концепції нескінченного циклу. Завдання скрипта забезпечити розробників та тестувальників безперервним потоком реалістичних телеметричних даних без необхідності розгортання фізичної мережі ультразвукових датчиків на контейнерах.

Робота алгоритму починається з одноразової ініціалізації початкових параметрів для масиву з 20 віртуальних контейнерів. На цьому етапі кожному об'єкту присвоюється унікальний ідентифікатор, географічні координати (широта та довгота на реальній карті міста) та базовий нульовий рівень заповненості. Після цього алгоритм входить у головний робочий цикл і переходить у режим системного очікування, тривалість якого жорстко регламентується константою SIMULATION\_INTERVAL (наприклад, 10 секунд). Ця штучна затримка необхідна для запобігання перевантаженню локального сервера надлишковими запитами (DDoS-ефект) та імітації реальних часових проміжків між відправкою даних фізичними мікроконтролерами.

Після завершення паузи алгоритм ініціює обхід масиву контейнерів. Для кожного окремого бака за допомогою математичних функцій генерується випадковий інкремент (приріст) об'єму сміття. Цей крок математично моделює реальну, непередбачувану поведінку мешканців міста щодо утилізації побутових відходів. Перед формуванням пакета даних система здійснює важливу логічну перевірку: чи досяг поточний рівень заповнення контейнера максимальної позначки у 100% або більше. У разі позитивної відповіді (відгалуження «Так»), алгоритм програмно імітує подію приїзду комунальної спецтехніки та спорожнення бака, примусово скидаючи змінну рівня заповненості до 0%. Якщо ж контейнер ще має вільний об'єм, цей крок очищення ігнорується.

Наступним етапом є консолідація згенерованих даних. Алгоритм формує структурований JSON-пакет, який містить ідентифікатор контейнера, актуальну мітку часу (timestamp) та оновлений відсоток заповненості. Використовуючи

бібліотеку для роботи з HTTP-протоколом, симулятор ініціює відправку сформованого пакета на сервер за допомогою методу PUT. Завершальним блоком поточної ітерації є перевірка стану мережевого з'єднання та відповіді сервера. Якщо сервер з технічних причин недоступний і повертає помилку або не відповідає на запит (відгалуження «Ні»), алгоритм перехоплює цей виняток і виконує запис відповідного повідомлення про збій у локальний системний файл `app.log`. Такий підхід гарантує відмовостійкість самого симулятора: він не припиняє аварійно свою роботу, а продовжує виконання (відгалуження «Так» та повернення на початок циклу), очікуючи відновлення зв'язку із сервером у наступних ітераціях.

Для забезпечення інтелектуальної обробки зібраної телеметрії основним компонентом серверної частини є модуль `route_optimizer.py`. Саме він містить математичні алгоритми для вирішення задачі маршрутизації сміттєвозів. Головною точкою входу для генерації логістичного шляху є функція `optimize_route()`. На початку свого виконання функція звертається до бази даних і формує масив виключно з тих контейнерів, рівень заповнення яких досяг або перевищив критичний поріг. Якщо таких контейнерів не знайдено, функція повертає повідомлення про відсутність потреби у виїзді спецтехніки.

Якщо проблемні локації існують, алгоритм розпочинає побудову базового маршруту за допомогою "жадібного" методу найближчого сусіда (Nearest Neighbor). Для цього використовується допоміжна функція `calculate_distance(lat1, lon1, lat2, lon2)`, яка обчислює реальну географічну відстань між двома точками на поверхні Землі з використанням сферичної тригонометрії (формула Гаверсина). Функція приймає координати (широту та довготу) і повертає дистанцію у кілометрах. Алгоритм найближчого сусіда ітеративно будує шлях: стартуючи з бази (депо), він на кожному кроці шукає найближчий невідвіданий контейнер і додає його до маршруту, поки всі критичні точки не будуть пройдені.

					КВРКІ.022056.22.02.12 ПЗ	Арк. 52
Зм.	Арк.	№ докум.	Підпис	Дата		

Хоча алгоритм найближчого сусіда працює швидко, він часто створює маршрути з самоперетинами, які не є глобально оптимальними. Тому наступним етапом є виклик функції `apply_2opt_optimization()`, яка реалізує локальну оптимізацію (алгоритм 2-opt). Ця функція приймає базовий маршрут і вкладеними циклами перевіряє кожну пару ребер (відрізків шляху). Якщо перехресне з'єднання (swap) двох вершин зменшує загальну довжину маршруту, функція виконує розвертання цієї частини шляху.

Для предиктивної аналітики системи та попередження переповнень використовується модуль `prediction.py`. Головна функція цього модуля `calculate_fill_rate_and_predict(container_id)` отримує з бази даних `history` останні записи (часові ряди) для конкретного бака. На основі цих даних розраховується середня швидкість накопичення відходів (Fill Rate). Зрештою, функція екстраполює поточний графік і повертає орієнтовний час (timestamp), коли бак досягне 80% заповненості. Це дозволяє диспетчеру включати баки до маршруту ще до того, як вони фізично переповняться.

Формування JSON-відповіді вимагає суворого дотримання узгоджених структур даних між серверною та клієнтською частинами. Згенерований пакет містить не лише впорядкований масив географічних координат (широта та довгота) для відмальовування поліліній на карті Leaflet, але й важливі метадані: загальну дистанцію об'їзду в кілометрах, орієнтовний час виконання маршруту та перелік ідентифікаторів контейнерів, що підлягають обслуговуванню.

Такий підхід до побудови REST API архітектури гарантує високу реактивність диспетчерського інтерфейсу. Отримавши серіалізований масив даних, React-клієнт за допомогою асинхронних запитів (через сервіс `api.js`) миттєво оновлює стан (state) компонентів. Це призводить до автоматичного перемальовування інтерфейсу без необхідності повного перезавантаження веб-сторінки: на інтерактивній мапі одразу з'являється новий оптимальний маршрут, а в аналітичній панелі оновлюються показники очікуваного пробігу комунальної

спецтехніки. Блок-схему логіки роботи алгоритму генерації маршруту наведено на рисунку 3.3.

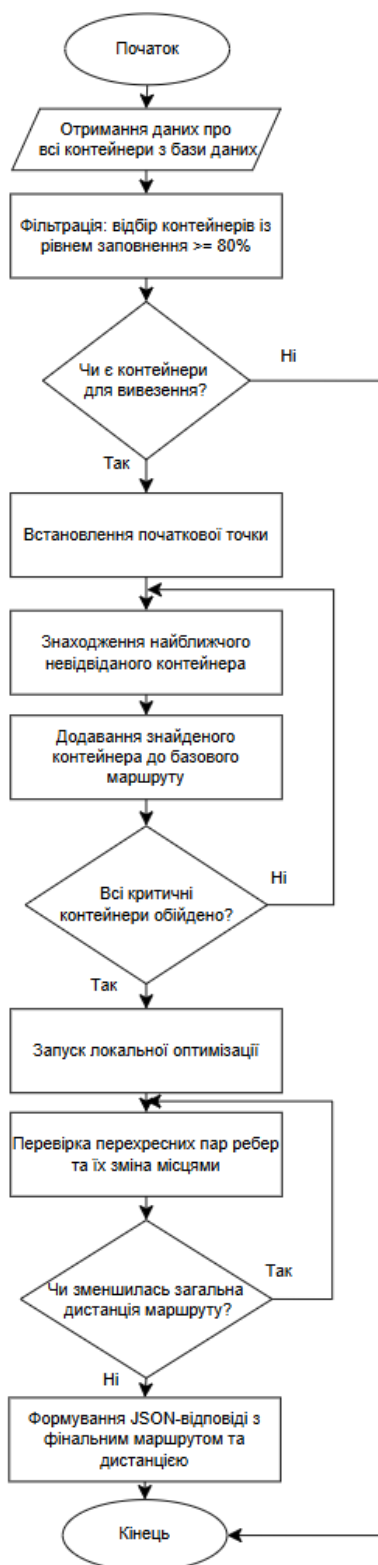


Рисунок 3.3 – Блок-схема алгоритму генерації оптимізованого маршруту

Відповідно до рисунку 3.3, алгоритм генерації оптимізованого маршруту демонструє комплексний двоетапний підхід до вирішення класичної логістичної задачі комівояжера (TSP).

Процес обчислень розпочинається із системного запиту до бази даних для отримання актуальної телеметрії всіх підключених контейнерів. Отриманий масив даних підлягає фільтрації за заданим порогом: алгоритм відбирає виключно ті баки, рівень заповнення яких досяг або перевищив встановлений критичний поріг. Після цього перевіряється перша логічна умова: чи існують взагалі контейнери, що потребують негайного обслуговування. Якщо масив порожній, алгоритм достроково завершує свою роботу, оскільки потреби у виїзді спецтехніки немає, що запобігає зайвим витратам пального.

У разі наявності критичних локацій (відгалуження «Так»), система встановлює географічні координати початкової точки (депо комунальної служби) і переходить до першого етапу маршрутизації застосування «жадібного» алгоритму найближчого сусіда (Nearest Neighbor). Алгоритм входить у цикл, де на кожній ітерації здійснюється пошук найближчого невідвіданого контейнера шляхом розрахунку реальної географічної відстані за допомогою сферичної тригонометрії (формула Гаверсина). Знайдена найближча точка додається до масиву базового маршруту. Цей цикл повторюється доти, доки не будуть обійдені усі відфільтровані контейнери.

Оскільки «жадібний» підхід формує швидкий, але не завжди глобально оптимальний шлях (часто утворюючи самоперетини та петлі на карті), алгоритм ініціює другий, більш ресурсоємний етап локальну оптимізацію методом 2-opt. У рамках цього другого циклу система ітеративно перевіряє перехресні пари ребер (відрізків маршруту) та виконує їх гіпотетичну зміну місцями (swar-операцію). Після кожної такої перестановки алгоритм перевіряє умову: чи призвела вона до зменшення загальної дистанції маршруту. Якщо шлях став коротшим (відгалуження «Так»), оновлена конфігурація зберігається як поточна найкраща, і цикл перевірки запускається заново для пошуку подальших покращень.

Процес оптимізації зупиняється тоді, коли жодна перестановка ребер більше не дає зменшення кілометражу (відгалуження «Ні»). Це означає, що алгоритм досяг локального мінімуму довжини шляху. На фінальному етапі відбувається серіалізація даних: формування структурованої JSON-відповіді, яка містить упорядкований масив географічних координат фінального маршруту та загальну дистанцію об'їзду. Після цього сформований пакет даних передається через REST API на клієнтський рівень для подальшої візуалізації на інтерактивній мапі диспетчера. Усі розроблені серверні функції та коротка інформація про їх призначення у системі подано у таблиці 3.1.

Таблиця 3.1 – Опис основних функцій та методів програмного забезпечення

Назва функції / методу	Модуль	Призначення
create_app()	app.py	Ініціалізація Flask-сервера та реєстрація API-маршрутів
init_db()	database.py	Створення та перевірка цілісності таблиць SQLite
calculate_distance()	route_optimizer.py	Обчислення відстані між точками за формулою Гаверсина
optimize_route()	route_optimizer.py	Побудова базового маршруту алгоритмом найближчого сусіда
apply_2opt_optimization()	route_optimizer.py	Локальна оптимізація маршруту для усунення перетинів
calculate_fill_rate()	prediction.py	Обчислення швидкості накопичення відходів методом лінійної регресії
get_optimized_route()	routes_api.py	REST API контролер для передачі згенерованого маршруту на Frontend

У ході етапу розробки програмного коду для кіберфізичної системи моніторингу збору відходів було успішно реалізовано повноцінну серверну інфраструктуру: обробку вхідних HTTP-запитів від IoT-симулятора, взаємодію з реляційною базою даних та реалізацію складних математичних алгоритмів. Поєднання жадібних логістичних алгоритмів із методами локальної оптимізації та предиктивного моделювання забезпечило виконання головної мети проєкту створення ефективного інструменту для автоматизації аудиту відходів та суттєвої мінімізації витрат пального комунальними службами.

### 3.3 Розробка веб-інтерфейсу та візуалізація даних у режимі реального часу

Важливим аспектом будь-якої кіберфізичної системи, орієнтованої на сектор комунального господарства, є наявність зручного, інформативного та швидкого інтерфейсу користувача. Отримані сервером телеметричні дані про стан сміттєвих контейнерів не мають практичної цінності без їх правильної інтерпретації та візуалізації для кінцевого оператора (диспетчера). З метою забезпечення віддаленого контролю та управління логістичними процесами було розроблено спеціалізований веб-додаток (Frontend) на базі бібліотеки React.js.

На відміну від спрощених систем, що використовують готові мобільні MQTT-панелі, розроблений диспетчерський інтерфейс є повноцінним односторінковим застосунком (SPA), який працює у вікні будь-якого сучасного веб-браузера. Архітектура клієнтської частини побудована за модульним принципом і складається з ізольованих компонентів (Header.jsx, Map.jsx, Controls.jsx, Statistics.jsx), кожен з яких відповідає за відображення конкретної частини бізнес-логіки та має власні інкапсульовані стилі (CSS Modules).

Головним функціональним елементом створеної інформаційної панелі є інтерактивна географічна карта, реалізована за допомогою бібліотеки Leaflet (компонент Map.jsx). Цей модуль отримує від сервера масив JSON-об'єктів, що містять актуальні дані про всі 20 контейнерів міста: їхні унікальні

					КВРКІ.022056.22.02.12 ПЗ	Арк. 57
Зм.	Арк.	№ докум.	Підпис	Дата		

ідентифікатори, координати (широту та довготу) і поточний відсоток заповненості. Для забезпечення інтуїтивно зрозумілої візуалізації в системі імплементовано алгоритм кольорового маркування:

- зелений маркер, рівень заповненості бака становить менше 50% (контейнер не потребує уваги);
- жовтий маркер, рівень заповненості становить від 50% до 80% (нормальний робочий стан);
- червоний маркер, рівень заповненості перевищує 80% (критичний стан, необхідне негайне включення до маршруту вивезення).

Зовнішній вигляд головної панелі диспетчера та відображення статусу контейнерів на інтерактивній карті наведено на рисунку 3.4.

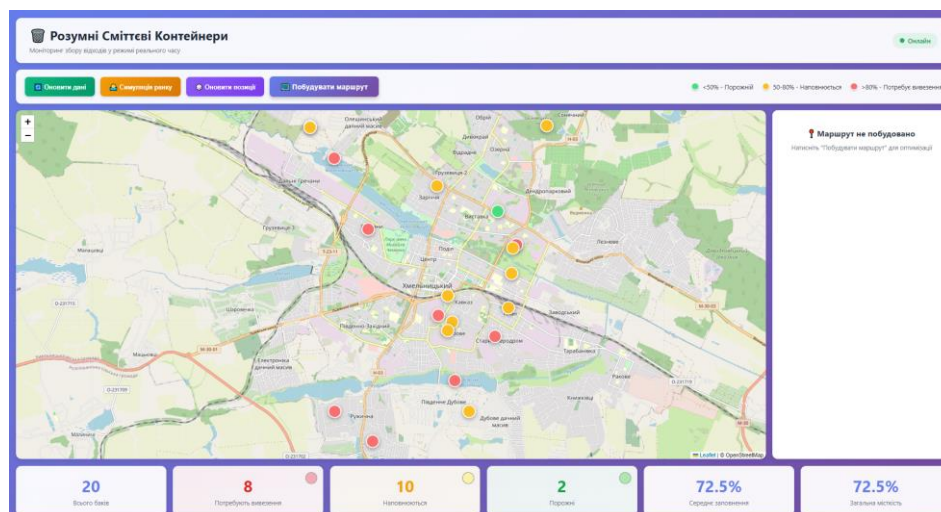


Рисунок 3.4 – Головний екран диспетчерського веб-інтерфейсу

Для підтримки даних в актуальному стані без необхідності ручного оновлення сторінки користувачем, у програмі реалізовано механізм періодичного (асинхронного) опитування серверного REST API. За цей процес відповідають спеціально написані користувацькі хуки (Custom Hooks), зокрема useContainers та useStatistics. Вони ініціюють HTTP GET запити до бекенду, отримують нові дані від IoT-симулятора та реактивно оновлюють стан (State) компонентів додатка. Завдяки цьому диспетчер бачить, як контейнери "наповнюються" в режимі реального часу, а статистичні показники на дашборді



### 3.4 Висновки до третього розділу

У процесі роботи було виконано повний цикл програмної розробки кіберфізичної системи моніторингу збору відходів на базі клієнт-серверної архітектури з використанням Python/Flask та React.js. На серверному рівні реалізовано взаємодію з базою даних SQLite, розроблено IoT-симулятор для генерації телеметричних даних та створено програмні засоби для обробки інформації про стан сміттєвих контейнерів.

Для побудови логістичних маршрутів у системі використано алгоритми просторової оптимізації, зокрема метод найближчого сусіда та алгоритм 2-opt для подальшого покращення отриманого маршруту. У серверній частині передбачено модуль предиктивної аналітики, який дає змогу прогнозувати орієнтовний час досягнення контейнером критичного рівня заповнення. Що дозволяє використовувати систему не лише для поточного моніторингу, а й для попереднього планування роботи комунальних служб.

Для взаємодії користувача із системою створено інтерактивний диспетчерський веб-інтерфейс із використанням бібліотеки Leaflet. Через REST API клієнтська частина отримує актуальні дані із сервера та відображає їх на карті у вигляді маркерів із кольоровою індикацією стану контейнерів. Такий підхід спрощує сприйняття інформації диспетчером і дає змогу швидко визначати контейнери, які потребують обслуговування. Крім того, у веб-інтерфейсі реалізовано відображення сформованих маршрутів, що дозволяє оцінити послідовність об'їзду контейнерів і зменшити кількість зайвих виїздів спецтехніки.

У результаті розроблена система забезпечує автоматизований збір, обробку та візуалізацію даних про стан контейнерів, а також може бути використана як програмна основа для подальшого розвитку рішень у сфері моніторингу та оптимізації збору відходів.

					КВРКІ.022056.22.02.12 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень було розроблено кіберфізичну систему моніторингу збору відходів у режимі реального часу, яка забезпечує автоматичний контроль заповненості сміттєвих контейнерів, предиктивну аналітику, генерацію оптимальних логістичних маршрутів та зручну візуалізацію для диспетчера через сучасний веб-інтерфейс.

У першому розділі проведено комплексний аналіз сучасного стану розвитку кіберфізичних систем у сфері комунального господарства та концепції «Розумного міста» (Smart City). Було обґрунтовано актуальність впровадження таких систем як ефективного інструменту для розв'язання проблеми нерациональної логістики вивезення твердих побутових відходів. Розглянуто принципи функціонування КФС, їхні переваги, а також здійснено детальний аналіз основних типів сенсорів для фіксації рівня відходів (ультразвукових, інфрачервоних, вагових). Проаналізовано способи передачі телеметрії, особливості алгоритмів маршрутизації та підходи до побудови диспетчерських інтерфейсів. У результаті сформовано цілісне бачення технічних можливостей реалізації таких систем, визначено критерії вибору апаратних і програмних засобів, а також проведено постановку задачі для розробки власного комплексу.

У другому розділі було зосереджено увагу на проектуванні та апаратній реалізації сенсорного вузла системи. Сформульовано функціональні та нефункціональні вимоги до КФС моніторингу відходів, визначено її структуру та проведено вибір компонентів відповідно до цілей дослідження. Розроблено загальну архітектуру, побудовано електричну принципову та монтажну схеми, які відобразили логіку взаємодії мікроконтролера NodeMCU, ультразвукового датчика HC-SR04 та автономного джерела живлення. Окремо розглянуто питання енергоефективності обґрунтовано використання режиму глибокого сну (Deep Sleep) для максимізації терміну роботи пристрою від акумулятора формату

					КВРКІ.022056.22.02.12 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

18650. У результаті створено надійну апаратну платформу, здатну безперервно збирати первинні дані у складних вуличних умовах.

У третьому розділі детально описано програмну реалізацію системи, яка базується на сучасній клієнт-серверній архітектурі. Здійснено розробку бекенду на мові Python (фреймворк Flask) із використанням реляційної бази даних SQLite. Особливу увагу приділено розробці математичного апарату: успішно реалізовано алгоритми просторової оптимізації маршрутів сміттєвозів (метод найближчого сусіда у поєднанні з алгоритмом локальної оптимізації 2-opt) та модуль предиктивної аналітики (лінійна регресія) для прогнозування переповнення контейнерів. Для тестування логіки створено програмний IoT-симулятор генерації телеметрії. Крім того, розроблено диспетчерський веб-інтерфейс на базі React.js та бібліотеки Leaflet, який забезпечує інтерактивну візуалізацію геопросторових даних, кольорове маркування статусу баків та відмальовування оптимальних маршрутів у режимі реального часу.

Загалом, результати виконаної роботи підтверджують ефективність застосування кіберфізичних систем для оптимізації логістики комунальних підприємств. Розроблений комплекс демонструє здатність автоматично відбирати контейнери, що потребують обслуговування, генерувати найкоротші шляхи об'їзду та надавати зручну аналітику. Отримані результати можуть бути використані для подальшого впровадження системи в реальних міських умовах, що дозволить суттєво зменшити витрати паливно-мастильних матеріалів, знизити викиди CO<sub>2</sub> та підвищити загальну якість обслуговування міської інфраструктури.

					КвРКІ.022056.22.02.12 ПЗ	Арк.
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. United Nations Environment Programme. Global Waste Management Outlook 2024: Beyond an Age of Waste: Turning Rubbish into a Resource. Nairobi : UNEP, 2024. URL: <https://www.unep.org/resources/global-waste-management-outlook-2024> (дата звернення: 05.05.2026).

2. World Bank Group. What a Waste 3.0: Global Solid Waste Management Data and Trends. Washington : World Bank Group, 2026. URL: <https://www.worldbank.org/en/publication/what-a-waste> (дата звернення: 05.05.2026).

3. Choudhary A., Jana P. K., Kumar M. Internet of Things: a comprehensive overview, architectures, applications, simulation tools, challenges and future directions. *Discover Internet of Things*. 2024. Vol. 4. Art. 31. DOI: 10.1007/s43926-024-00084-3. URL: <https://link.springer.com/article/10.1007/s43926-024-00084-3> (дата звернення: 05.05.2026).

4. Dauda A., Flauzac O., Nolot F. A Survey on IoT Application Architectures. *Sensors*. 2024. Vol. 24, № 16. Art. 5320. DOI: 10.3390/s24165320. URL: <https://www.mdpi.com/1424-8220/24/16/5320> (дата звернення: 05.05.2026).

5. Omrany H., Al-Obaidi K. M., Hossain M. et al. IoT-enabled smart cities: a hybrid systematic analysis of key research areas, challenges, and recommendations for future direction. *Discover Cities*. 2024. Vol. 1. Art. 2. DOI: 10.1007/s44327-024-00002-w. URL: <https://link.springer.com/article/10.1007/s44327-024-00002-w> (дата звернення: 05.05.2026).

6. Mansour M., Gamal A., Ahmed A. I. et al. Internet of Things: A Comprehensive Overview on Protocols, Architectures, Technologies, Simulation Tools, and Future Directions. *Energies*. 2023. Vol. 16, № 8. Art. 3465. DOI: 10.3390/en16083465. URL: <https://www.mdpi.com/1996-1073/16/8/3465> (дата звернення: 05.05.2026).

					КВРКІ.022056.22.02.12 ПЗ	Арк. 63
Зм.	Арк.	№ докум.	Підпис	Дата		

7. World Bank Group. Clean Cities, Bright Futures: Accelerating Investment and Reforms in Solid Waste Management in Developing Countries. 2025. URL: <https://www.worldbank.org/en/results/2025/04/30/clean-cities-bright-futures-accelerating-investment-and-reforms-in-solid-waste-management-in-developing-countries> (дата звернення: 05.05.2026).

8. Roy A., Manna A., Kim J., Moon I. IoT-based smart bin allocation and vehicle routing in solid waste management: A case study in South Korea. *Computers & Industrial Engineering*. 2022. Vol. 175. Art. 108457. DOI: 10.1016/j.cie.2022.108457. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0360835222004910> (дата звернення: 05.05.2026).

9. ISO/IEC 30141:2024. Internet of Things (IoT) – Reference Architecture. Geneva : ISO, 2024. URL: <https://www.iso.org/standard/88800.html> (дата звернення: 05.05.2026).

10. Vishnu S., Jino Ramson S. R., Senith S. et al. IoT-Enabled Solid Waste Management in Smart Cities. *Smart Cities*. 2021. Vol. 4, № 3. P. 1004–1017. DOI: 10.3390/smartcities4030053. URL: <https://www.mdpi.com/2624-6511/4/3/53> (дата звернення: 05.05.2026).

11. Sosunova I., Porras J. IoT-Enabled Smart Waste Management Systems for Smart Cities: A Systematic Review. *IEEE Access*. 2022. Vol. 10. P. 73326–73363. DOI: 10.1109/ACCESS.2022.3188308. URL: <https://ieeexplore.ieee.org/document/9825942> (дата звернення: 05.05.2026).

12. Sony M. Design of cyber physical system architecture for industry 4.0 through lean six sigma: conceptual foundations and research issues. *Production & Manufacturing Research*. 2020. Vol. 8, № 1. P. 158–181. DOI: 10.1080/21693277.2020.1774814. URL: <https://www.tandfonline.com/doi/full/10.1080/21693277.2020.1774814> (дата звернення: 05.05.2026).

13. Arthur M. P., Shoba S., Pandey A. A survey of smart dustbin systems using the IoT and deep learning. *Artificial Intelligence Review*. 2024. Vol. 57. Art. 56. DOI: 10.1007/s10462-023-10646-6. URL:

					КВРКІ.022056.22.02.12 ПЗ	Арк. 64
Зм.	Арк.	№ докум.	Підпис	Дата		

<https://link.springer.com/article/10.1007/s10462-023-10646-6> (дата звернення: 05.05.2026).

14. Kannan D., Khademolqorani S., Janatyan N., Alavi S. Smart waste management 4.0: The transition from a systematic review to an integrated framework. *Waste Management*. 2024. Vol. 174. P. 1–14. DOI: 10.1016/j.wasman.2023.08.041. URL: <https://www.sciencedirect.com/science/article/pii/S0956053X23005573> (дата звернення: 05.05.2026).

15. Ali B., Wang Y., Ullah I. et al. Internet of Things-Assisted Vehicle Route Optimization for Solid Waste Management. *Applied Sciences*. 2024. Vol. 14, № 1. Art. 287. DOI: 10.3390/app14010287. URL: <https://www.mdpi.com/2076-3417/14/1/287> (дата звернення: 05.05.2026).

16. Biswas R., Jana S., Pal K., Ghosh A. Smart Waste Management System using IoT (Internet of Things). *International Journal of Advanced Research in Science, Communication and Technology*. 2025. DOI: 10.48175/IJARST-28235. URL: <https://doi.org/10.48175/IJARST-28235> (дата звернення: 05.05.2026).

17. Rifat R. I., Haque F. B., Jahan T., Zahir Z.-B. IoT Based Smart Garbage Management System. *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*. 2022. DOI: 10.1109/IC3IoT53935.2022.9767993. URL: <https://doi.org/10.1109/IC3IoT53935.2022.9767993> (дата звернення: 05.05.2026).

18. Sensoneo. Ultrasonic Bin Fill-Level Sensor. URL: <https://www.sensoneo.com/products/ultrasonic-bin-sensor/> (дата звернення: 05.05.2026).

19. Devendra Kumar R. N., Dhaniya R. D., Poobesh Gupta M., Soundar S., Thamaraiselvan S. IOT-Based Garbage Monitoring System Using Weight Sensing. *International Journal for Science and Advance Research in Technology*. 2023. Vol. 9, № 3. P. 355–358. URL: <https://ijsart.com/public/storage/paper/pdf/IJSARTV9I360236.pdf> (дата звернення: 05.05.2026).

					КВРКІ.022056.22.02.12 ПЗ	Арк. 65
Зм.	Арк.	№ докум.	Підпис	Дата		

20. Likotiko E., Matsuda Y., Yasumoto K. Smart garbage bin: Garbage growth behavior prediction. *DPSWS* 2020. 2020. URL: [https://yukimat.jp/data/pdf/paper/SmartGarbageBin\\_c\\_202011\\_eunice\\_DPSWS.pdf](https://yukimat.jp/data/pdf/paper/SmartGarbageBin_c_202011_eunice_DPSWS.pdf) (дата звернення: 05.05.2026).

21. Zha Y., Zhou M., Yang J., Yang Y. A Design of Intelligent Garbage Bin System. *Journal of Physics: Conference Series*. 2021. Vol. 1972. Art. 012056. DOI: 10.1088/1742-6596/1972/1/012056. URL: <https://iopscience.iop.org/article/10.1088/1742-6596/1972/1/012056> (дата звернення: 05.05.2026).

22. Vesga Ferreira J. C., Perez Waltero H. E., Vesga Barrera J. A. Design of a Waste Classification System Using a Low Experimental Cost Capacitive Sensor and Machine Learning Algorithms. *Applied Sciences*. 2025. Vol. 15, № 3. Art. 1565. DOI: 10.3390/app15031565. URL: <https://www.mdpi.com/2076-3417/15/3/1565> (дата звернення: 05.05.2026).

23. Mohammed S. T. et al. Smart Electronic Bin Using IoT. *International Journal of Advanced Trends in Computer Science and Engineering*. 2021. Vol. 10, № 3. URL: <https://www.warse.org/IJATCSE/static/pdf/file/ijatcse951032021.pdf> (дата звернення: 05.05.2026).

24. Maturi S. M. et al. Smart Dustbins: Real-Time Monitoring and Optimization for Smart Waste Management. *Engineering, Technology & Applied Science Research*. 2025. URL: <https://etasr.com/index.php/ETASR/article/view/8562> (дата звернення: 05.05.2026).

25. Zoumpoulis P. et al. Smart bins for enhanced resource recovery and waste management. *Cities*. 2024. Art. 105276. DOI: 10.1016/j.cities.2024.105276. URL: <https://www.sciencedirect.com/science/article/pii/S0264275124003640> (дата звернення: 05.05.2026).

26. Agnew C., Mewada D., Grua E. M. et al. Detecting the overfilled status of domestic and commercial bins using computer vision. *Intelligent Systems with Applications*. 2023. Vol. 18. Art. 200229. DOI: 10.1016/j.iswa.2023.200229. URL:

<https://www.sciencedirect.com/science/article/pii/S2667305323000546> (дата звернення: 05.05.2026).

27. Wang C., Qin J., Qu C. et al. A smart municipal waste management system based on deep-learning and Internet of Things. *Waste Management*. 2021. Vol. 135. P. 20–29. DOI: 10.1016/j.wasman.2021.08.028. URL: <https://www.sciencedirect.com/science/article/pii/S0956053X21004621> (дата звернення: 05.05.2026).

28. Fang B., Yu J., Chen Z. et al. Artificial intelligence for waste management in smart cities: a review. *Environmental Chemistry Letters*. 2023. Vol. 21. P. 1959–1989. DOI: 10.1007/s10311-023-01604-3. URL: <https://link.springer.com/article/10.1007/s10311-023-01604-3> (дата звернення: 05.05.2026).

29. Hussain I., Elomri A., Kerbache L., El Omri A. Smart city solutions: Comparative analysis of waste management models in IoT-enabled environments using multiagent simulation. *Sustainable Cities and Society*. 2024. Vol. 103. Art. 105247. DOI: 10.1016/j.scs.2024.105247. URL: <https://www.sciencedirect.com/science/article/pii/S2210670724000763> (дата звернення: 05.05.2026).

30. Subero A. USART, SPI, I2C, and Communication Protocols. *Programming PIC Microcontrollers with XC8*. Berkeley : Apress, 2024. P. 297–366. DOI: 10.1007/979-8-8688-0467-0\_8. URL: [https://link.springer.com/chapter/10.1007/979-8-8688-0467-0\\_8](https://link.springer.com/chapter/10.1007/979-8-8688-0467-0_8) (дата звернення: 05.05.2026).

31. Inthasuth T., Kaewjumras Y., Somwong S., Boonsong W. Comparative analysis of ZigBee, LoRa, and NB-IoT in a smart building: advantages, limitations, and integration possibilities. *International Journal of Reconfigurable and Embedded Systems*. 2025. Vol. 14, № 1. P. 165–175. DOI: 10.11591/ijres.v14.i1.pp165-175. URL: <https://ijres.iaescore.com/index.php/IJRES/article/view/21041> (дата звернення: 05.05.2026).

32. Rojek I., Prokopowicz P., Piechowiak M. et al. The Impact of Data Analytics Based on Internet of Things, Edge Computing, and Artificial Intelligence on Energy

Efficiency in Smart Environment. *Applied Sciences*. 2026. Vol. 16, № 1. Art. 225. DOI: 10.3390/app16010225. URL: <https://www.mdpi.com/2076-3417/16/1/225> (дата звернення: 05.05.2026).

33. Afkarien A. K., Astawa I. G. P., Nadziroh F. Development of IoT-Based Smart Waste Management Systems for Organic and Non-Organic Waste in Smart Cities. *The Indonesian Journal of Computer Science*. 2025. Vol. 14, № 2. P. 2479–2487. DOI: 10.33022/ijcs.v14i2.4606. URL: <https://www.ijcs.net/ijcs/index.php/ijcs/article/view/4606> (дата звернення: 05.05.2026).

34. Telicko J., Jakovics A. Power efficient wireless monitoring system based on ESP8266. *2022 IEEE 63th International Scientific Conference on Power and Electrical Engineering. IEEE*, 2022. DOI: 10.1109/RTUCON56726.2022.9978873. URL: <https://ieeexplore.ieee.org/document/9978873> (дата звернення: 05.05.2026).

35. Djalilov A., Sobirov E., Nazarov O., Urolov S., Gayipov I. Study on automatic water level detection process using ultrasonic sensor. *IOP Conference Series: Earth and Environmental Science*. 2023. Vol. 1142. Art. 012020. DOI: 10.1088/1755-1315/1142/1/012020. URL: <https://iopscience.iop.org/article/10.1088/1755-1315/1142/1/012020> (дата звернення: 05.05.2026).

36. Pahkala M. Power management solutions for cellular IoT systems utilizing power constrained energy sources. Oulu : University of Oulu, 2025. URL: <https://oulurepo.oulu.fi/handle/10024/54560> (дата звернення: 05.05.2026).

37. Zhu L. Research on Innovative Software Architecture Design and System Performance Improvement Based on Intelligent Optimization Algorithm. IEEE, 2024. URL: <https://ieeexplore.ieee.org/document/11439249> (дата звернення: 05.05.2026).

38. Development of microservices-based APIs for smart energy systems. Porto : Instituto Superior de Engenharia do Porto, 2025. URL: <http://hdl.handle.net/10400.22/31061> (дата звернення: 05.05.2026).

39. Flask Documentation. Pallets Projects. URL: <https://flask.palletsprojects.com/> (дата звернення: 05.05.2026).

					КВРКІ.022056.22.02.12 ПЗ	Арк. 68
Зм.	Арк.	№ докум.	Підпис	Дата		

40. Wan G. AI Inspired Algorithms for Several Combinatorial Optimization Problems in Data Science. Dallas : University of Texas at Dallas, 2021. URL: <https://utd-ir.tdl.org/items/17579dfd-3bd2-4a34-a47d-2e0fa74127ff> (дата звернення: 05.05.2026).

41. Agárdi A., Kovács L., Bányai T. Using Time Series and Classification in Vehicle Routing Problem. *International Journal of Performability Engineering*. 2021. Vol. 17, № 1. P. 14–25. DOI: 10.23940/ijpe.21.01.p2.1425. URL: <https://www.ijpe-online.com/EN/10.23940/ijpe.21.01.p2.1425> (дата звернення: 05.05.2026).

42. Bogo M., Soldani J., Neri D., Brogi A. Component-aware orchestration of cloud-based enterprise applications, from TOSCA to Docker and Kubernetes. *Software: Practice and Experience*. 2020. DOI: 10.1002/spe.2848. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2848> (дата звернення: 05.05.2026).

43. Daqouri L. Performance Analysis of Time Series Databases for IoT Applications. Uppsala : Uppsala University, 2024. URL: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1947085> (дата звернення: 05.05.2026).

44. React Documentation. Meta Open Source. URL: <https://react.dev/> (дата звернення: 05.05.2026).

45. Leaflet Documentation. Leaflet. URL: <https://leafletjs.com/reference.html> (дата звернення: 05.05.2026).

46. Withana Gamage G. Web Application development project for vehicle parking reservation system in student apartments. Theseus, 2025. URL: <https://www.theseus.fi/handle/10024/893663> (дата звернення: 05.05.2026).

47. Karampakakis P. et al. A Web-Based Application for Smart City Data Analysis and Visualization. *Future Internet*. 2025. Vol. 17, № 5. Art. 217. DOI: 10.3390/fi17050217. URL: <https://www.mdpi.com/1999-5903/17/5/217> (дата звернення: 05.05.2026).

48. Madasanal K., Gudnavar A., Nagarmunoli S., Madagouda B., Naregal K. Efficient Smart Public Transit System Using RFID and IoT integration with MERN

					КВРКІ.022056.22.02.12 ПЗ	Арк. 69
Зм.	Арк.	№ докум.	Підпис	Дата		

web applications. IEEE, 2024. URL: <https://ieeexplore.ieee.org/document/10940803> (дата звернення: 05.05.2026).

49. Nemlaha E., Střelec P., Horák T., Kováč S., Tanuška P. Suitability of MQTT and REST Communication Protocols for AIoT or IIoT Devices Based on ESP32 S3. *Software Engineering Application in Systems Design*. Cham : Springer, 2023. P. 225–233. DOI: 10.1007/978-3-031-21435-6\_19. URL: [https://link.springer.com/chapter/10.1007/978-3-031-21435-6\\_19](https://link.springer.com/chapter/10.1007/978-3-031-21435-6_19) (дата звернення: 05.05.2026).

50. Bader S. R., Maleshkova M. SOLIOT: Decentralized Data Control and Interactions for IoT. *Future Internet*. 2020. Vol. 12, № 6. Art. 105. DOI: 10.3390/fi12060105. URL: <https://www.mdpi.com/1999-5903/12/6/105> (дата звернення: 05.05.2026).

51. NodeMCU ESP8266 specifications, overview and setting up. Make-It.ca. URL: <https://www.make-it.ca/nodemcu-details-specifications/> (дата звернення: 05.05.2026).

52. ESP8266 pinout reference. Last Minute Engineers. URL: <https://lastminuteengineers.com/esp8266-pinout-reference/> (дата звернення: 05.05.2026).

53. HC-SR04 ultrasonic sensor module: pinout, features and datasheet. Components101. 2021. URL: <https://components101.com/sensors/ultrasonic-sensor-working-pinout-datasheet> (дата звернення: 05.05.2026).

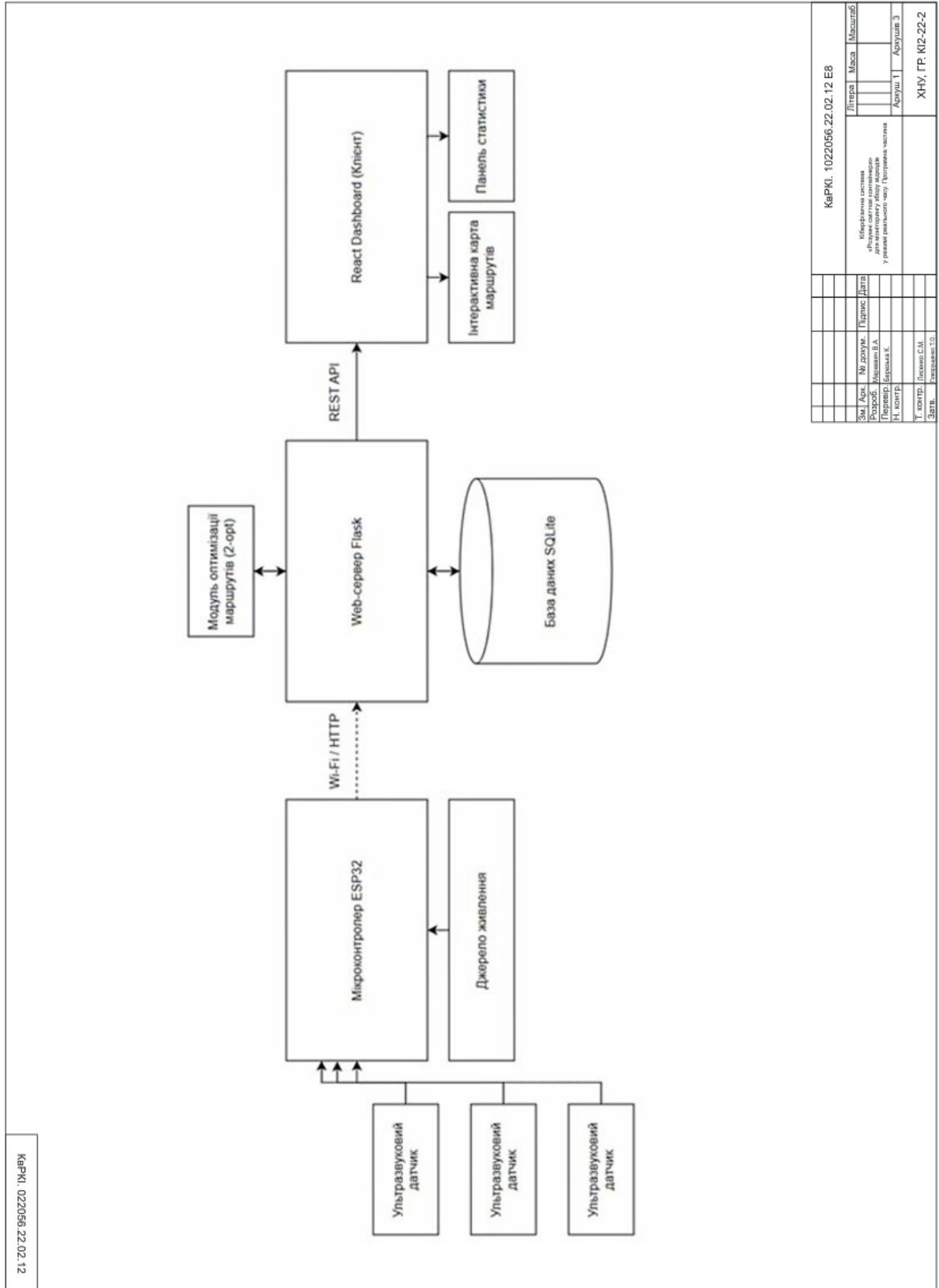
54. Li-ion battery technology specification: ICR18650 3000mAh 3.7V. PKCELL. 2020. URL: [https://www.rcscomponents.kiev.ua/datasheets/ICR18650\\_3\\_7V\\_3000mAh-pkcell.pdf](https://www.rcscomponents.kiev.ua/datasheets/ICR18650_3_7V_3000mAh-pkcell.pdf) (дата звернення: 05.05.2026).

					КВРКІ.022056.22.02.12 ПЗ	Арк. 70
Зм.	Арк.	№ докум.	Підпис	Дата		

# ДОДАТОК А

(обов'язковий)

Копія креслення «Архітектура ПЗ проекту»



Зм.	Дат.	№ докум.	Підпис	Дата	Літера	Місяць	Місяць
Розроб.		Маслова В.А.					
Н. контр.		Борозняк К.					
Т. контр.		Резанко С.М.					
Затв.		Резанко С.М.					

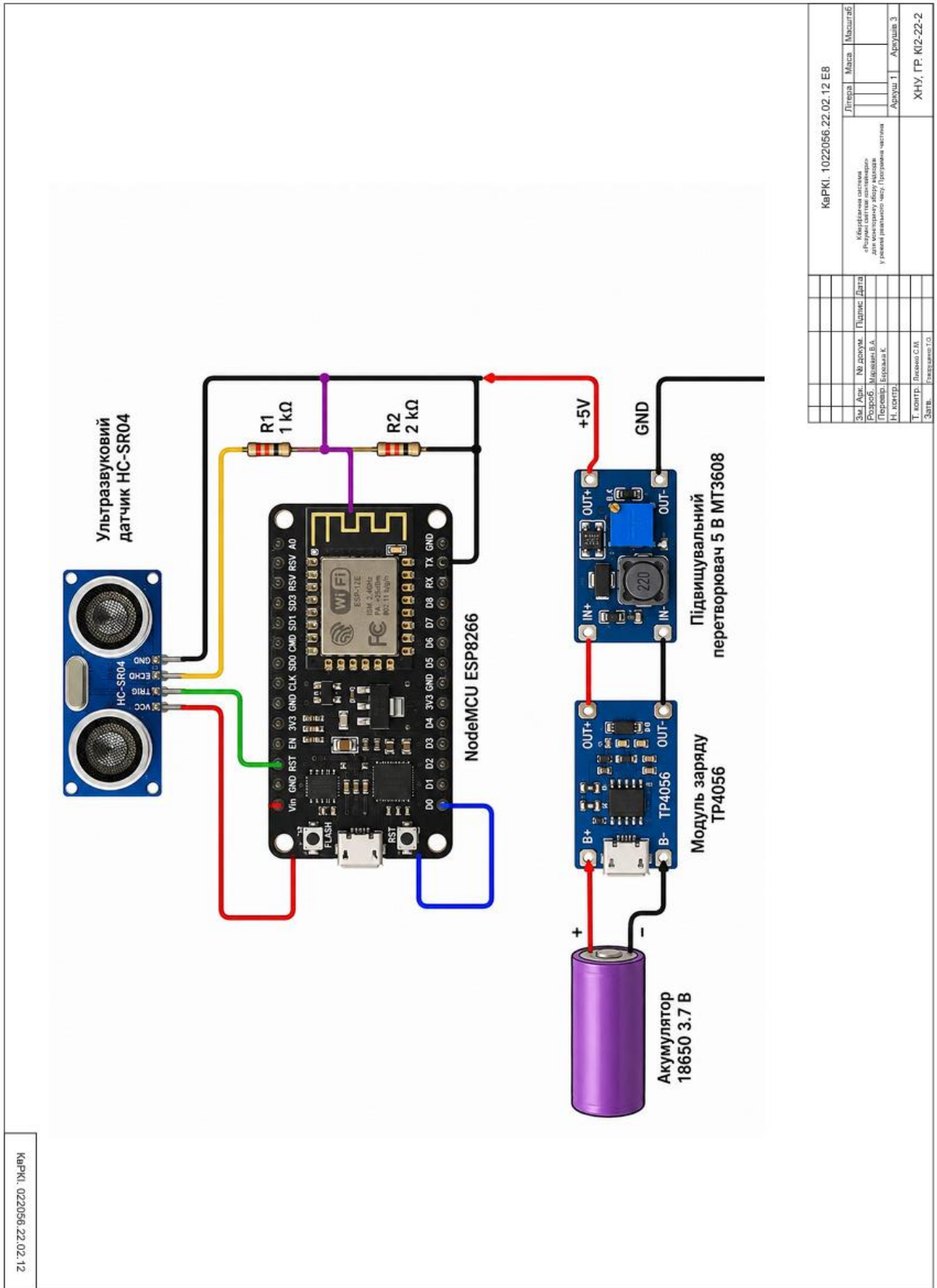
Кубернаторська система  
Державна система управління  
у сфері розвитку місту / Регіональна частина

Автори: 1) Арсучай 3



# ДОДАТОК В (обов'язковий)

Копія креслення «Монтажна схема»



## 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:** Владислав МАРКЕВИЧ

**Співавтор:**

**Назва:** Кіберфізична система «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина

**Експерт:** Катерина БЕРЕЗЬКА

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

**Коефіцієнт подібності 1:** 7.72%

**Коефіцієнт подібності 2:** 2.73%

**Мікропробіли:** 4

**Заміна букв:** 0

**Інтервали:** 0

**Блі знаки:** 0

**Дата створення звіту:** 2026-05-28 08:03:40.0

**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 процедурам. Таким чином робота не приймається.

**Обґрунтування:**

2026-05-28

Дата



Доцент Андрій Нічепорук

експерт

**Anti-Plagiarism (<http://ap.km.ua>) v-15.701****Максимальне співпадіння з одним документом 2.0%****Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 12%**

ID: 272537 Назва: БКР Кіберфізична система «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина Додано в БД: 2026-05-28 Автора: Владислав МАРКЕВИЧ Керівник: Катерина БЕРЕЗЬКА Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	102998	703	4913 (5%)	57 (8%)

## Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Маркевич Владислав Анатолійович

Тема: Кіберфізична система «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу. Програмна частина

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 58

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розробка програмної частини кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу заповненості контейнерів у режимі реального часу. У роботі обґрунтовано вибір компонентів, розроблено схеми системи та реалізовано програмне забезпечення для збору, оброблення й візуалізації даних.

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

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі кваліфікаційної роботи проведено аналіз структурних і функціональних особливостей кіберфізичних систем моніторингу збору відходів, розглянуто сучасні підходи до використання сенсорних пристроїв, засобів передавання даних, веб-інтерфейсів та інформаційних систем для контролю стану сміттєвих контейнерів, виконано постановку задачі. У другому розділі виконано проекткування системи обробки інформації у кіберфізичній системі «Розумні сміттєві контейнери». Визначено функціональні та нефункціональні вимоги до системи, розглянуто загальну структуру апаратної реалізації, обґрунтовано вибір мікроконтролера, датчика рівня заповнення, джерела живлення та допоміжних компонентів. Розроблено структурну схему програмного забезпечення, електричну принципову та монтажну схеми, що відображають взаємодію основних елементів системи. У третьому розділі здійснено програмно-апаратну реалізацію системи, розглянуто використані програмні бібліотеки

та їх функціональне призначення, розроблено програмний код для отримання й передавання даних про рівень заповнення контейнерів, реалізовано серверну частину, базу даних і веб-інтерфейс для диспетчерського моніторингу. Основною частиною роботи є реалізація інтерактивної візуалізації стану контейнерів, обробка телеметрії та підтримка логіки побудови маршрутів для обслуговування контейнерів. У роботі використано сучасні програмні засоби, підходи клієнт-серверної взаємодії, технології Інтернету речей та методи організації обміну даними між апаратним і програмним рівнями системи.

4. Позитивні сторони роботи: практична цінність та комплексна реалізація апаратної і програмної частин системи.

5. Негативні сторони роботи: доцільно доповнити роботу ширшим аналізом результатів маршрутизації.

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 технічному рівні.

8. Інші зауваження: \_\_\_\_\_

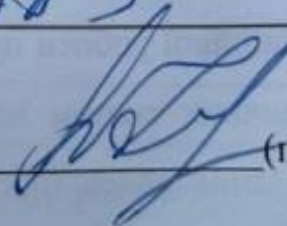
9. Оцінка дипломної роботи: добре (75/С).

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

К.Г.Н., доц. Корич Ю.В.

доцент кафедри ІТЗ

"28" травня 2026 р.

 (підпис)

Зав. кафедри КПС  
д-р. філософії Ользі ПАВЛОВІЙ

Владислав МАРКЕВИЧ

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-22-2

### ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповішений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

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

1 травня 2026 року



## РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

### КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Інформаційна система моніторингу стану здоров'я пацієнтів із оптимізацією планування завдань

Автор Владислав МАРКЕВИЧ

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія

Науковий керівник: к.т.н., доцент Катерина БЕРЕЗЬКА

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	

#### Підтвердження:

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

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.
- 4) значна частина знайденого плагіату відноситься до списку використаних джерел

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 7,72%; та системою Anti-Plagiarism складає 2,73%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

01.06.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи

  
Підпис

  
Підпис

  
Підпис

Ольга ПАВЛОВА  
Ім'я, ПРІЗВИЩЕ

Андрій НІЧЕПОРУК  
Ім'я, ПРІЗВИЩЕ

Катерина БЕРЕЗЬКА  
Ім'я, ПРІЗВИЩЕ