

КВАЛІФІКАЦІЙНА РОБОТА

Кіберфізична система "Розумні сміттєві контейнери" для моніторингу збору відходів у режимі реального часу. Апаратна частина

Назва теми

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

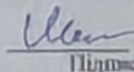
Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Шифр: КВРКІ.022114.22.02.11 ПЗ

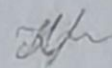
Виконав здобувач IV курсу група K12-22-2


Підпис

Богдан ШЕВЧУК

Ініціали, прізвище

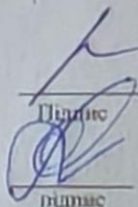
Керівник,..... к.т.н., доцент



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
«07» червня 2026 р.

Ольга ПАВЛОВА

Ініціали, прізвище

Дата

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

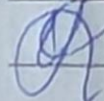
Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

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

Шевчуку Богдану Володимировичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кіберфізична система «Розумні сміттєві контейнери для моніторингу збору відходів у режимі реального часу» Апаратна 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 10.01.2026 р. № 11

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) дослідження предметної області, вибір компонентів для проектування програмно – технічного засобу, реалізація програмно – технічного засобу, апаратного прототипу, серверного програмного забезпечення, користувацького інтерфейсу та кіберфізичної си

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

Архітектура ПЗ проекту

Архітектура ПЗ для кіберфізичної системи

Апаратне забезпечення проекту

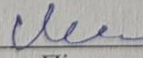
6. Консультанти розділів кваліфікаційної роботи

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

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

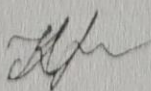
КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму 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 програмно-технічного засобу	01.04.2026	виконано
5	Робота над розділом 3 – реалізація програмно-технічного засобу, реалізація апаратного прототипу, серверного програмного забезпечення, користувацького інтерфейсу та кіберфізичної системи.	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 2026року	

Здобувач 
Підпис

Богдан ШЕВЧУК
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.022114.22.02.11 ПЗ	Пояснювальна записка	72		
			<u>Графічні матеріали</u>			
2		КВРКІ.022114.22.02.11 Е8	Архітектура ПЗ проєкту	1		
3		КВРКІ.022114.22.02.11 Е8	Архітектура ПЗ для кіберфізичної системи	1		
		КВРКІ.022114.22.2.11 Е8	Апаратне забезпечення проєкту	1		

КВРКІ.022114.22.02.11 ВП

Зм	Арк	№ докум	Підпис	Дата
Розробив		Шевчук	<i>Шевчук</i>	
Перевір.		Березька	<i>Березька</i>	
Н. контр.		Кисіль	<i>Кисіль</i>	
Зав.		Павлова	<i>Павлова</i>	01.06

Відомість проєкту

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» Апаратна частина (Сміттєві контейнери обладнані ультразвуковими датчиками рівня заповнення. Аудит відходів в реальному часі. Система аналізує ці дані щоранку буде оптимальний маршрут для сміттєвозів. Що б вони заїжджали лише до повних баків. Це розвантажує трафік та економить пальне).

Автор роботи: Богдан ШЕВЧУК

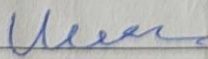
Керівник роботи: Катерина БЕРЕЗЬКА.

Пояснювальна записка: 72 с., 17 рис., 11 табл., 3 дод., 50 джерел.

Графічна частина:

У кваліфікаційній роботі розглянуто розробку кіберфізичної системи «Розумні сміттєві контейнери для моніторингу збору відходів у режимі реального часу» Апаратна частина (Сміттєві контейнери обладнані ультразвуковими датчиками рівня заповнення. Аудит відходів в реальному часі. Система аналізує ці дані щоранку буде оптимальний маршрут для сміттєвозів, що б вони заїжджали лише до повних баків. Це розвантажує трафік та економить пальне).

Метою роботи є створення програмно-технічного засобу для автоматизованого контролю рівня заповнення контейнерів, передачі телеметричних даних на сервер та їх відображення у користувацькому інтерфейсі. У роботі виконано аналіз існуючих систем моніторингу відходів, сформовано функціональні вимоги та спроектовано архітектуру кіберфізичної системи. Результатом роботи є спроектована кіберфізична система, яка забезпечує моніторинг стану сміттєвих контейнерів у режимі реального часу, своєчасне виявлення критичного заповнення та підвищення ефективності організації збору відходів.



Підпис здобувача





30.05.2026

Дата

ЗМІСТ

Вступ.....	4
1 Дослідження предметної області та постановка задачі.....	6
1.1 Змістовний аналіз предметної області та функціональних особливостей.....	6
1.2 Аналіз наявного програмно–апаратного забезпечення предметної області.....	9
1.3 Визначення вимог до кіберфізичної системи та постановка задачі розробки.....	12
1.4 Висновки до першого розділу.....	15
2 Проектування програмно – технічного засобу.....	17
2.1 Розробка загальної архітектури кіберфізичної системи	17
2.2 Проектування програмного забезпечення кіберфізичної системи.....	19
2.3 Проектування бази даних та серверної частини.....	23
2.4 Висновки до другого розділу.....	29
3 Реалізація програмно – технічного засобу.....	32
3.1 Реалізація апаратного прототипу сенсорного вузла.....	32
3.2 Реалізація вбудованого програмного забезпечення.....	40
3.3 Реалізація серверного програмного забезпечення.....	47
3.4 Реалізація користувацького інтерфейсу.....	54
3.5 Тестування кіберфізичної системи.....	58
3.6 Висновок до третього розділу.....	61
Висновки.....	64

КВРКІ.022114.22.02.11 ПЗ

Зм.	Арк.	№докум.	Підпис	Дата				
Виконав		Богдан ШЕВЧУК			Кіберфізична система «Розумні сміттєві контейнери для моніторингу збору відходів у режимі реального часу» Апаратна частина	Літера	Арквпш	Арквпшів
Перевір.		Катерина БЕРЕЗЬКА				у	2	72
Н.контр.		Тетяна КИСІЛЬ			ХНУ КІ2-22-2			
Затвер.		Ольга ПАВЛОВА		01/09/20				

Перелік джерел посилань.....	66
Додаток А «Архітектура ПЗ проєкту».....	70
Додаток Б «Архітектура ПЗ для кіберфізичної системи».....	71
Додаток В Архітектурне забезпечення проєкту.....	72

					КвРКІ.022114.22.02.11 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		3

ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій, Інтернету речей (IoT), кіберфізичних систем та засобів бездротового зв'язку особливої актуальності набуває автоматизація процесів управління об'єктами міської інфраструктури. Концепція «розумного міста» передбачає впровадження інтелектуальних технологій для підвищення ефективності функціонування комунальних служб, оптимізації використання ресурсів та покращення якості життя населення. Одним із важливих напрямів цифрової трансформації міського середовища є модернізація систем збору, транспортування та контролю твердих побутових відходів.

Зростання чисельності населення, процеси урбанізації та збільшення обсягів споживання призводять до постійного зростання кількості побутових відходів. У зв'язку з цим комунальні служби стикаються з необхідністю підвищення ефективності управління процесами збору сміття та контролю за станом контейнерних майданчиків. Традиційні системи обслуговування контейнерів переважно базуються на заздалегідь визначених графіках вивезення відходів без урахування фактичного рівня заповнення контейнерів. Такий підхід часто призводить до переповнення контейнерів, нераціонального використання транспортних ресурсів, збільшення витрат на обслуговування та погіршення санітарного стану навколишнього середовища.

Ефективним шляхом вирішення зазначених проблем є впровадження кіберфізичних систем моніторингу, які забезпечують автоматизований збір даних із фізичних об'єктів, їх передачу через мережі зв'язку, подальшу обробку на серверній стороні та відображення результатів у зручному для користувача вигляді. Використання сенсорних пристроїв, мікроконтролерних платформ та сучасних програмних технологій дозволяє здійснювати безперервний контроль стану контейнерів і своєчасно реагувати на критичні ситуації.

					КвРКІ.022114.22.02.11 ПЗ	Арк.
						4
Зм.	Арк.	№докум.	Підпис	Дата		

Особливу роль у побудові таких систем відіграють технології Інтернету речей, які забезпечують взаємодію між фізичними пристроями та інформаційними сервісами. Поєднання сенсорів, бездротових каналів передачі даних, серверної інфраструктури та веборієнтованих інтерфейсів створює основу для реалізації інтелектуальних систем управління відходами, здатних функціонувати в режимі реального часу та підтримувати прийняття управлінських рішень. У зв'язку з цим розробка кіберфізичної системи «Розумні сміттєві контейнери» є актуальним завданням у галузі комп'ютерної інженерії та відповідає сучасним тенденціям розвитку технологій автоматизації, Інтернету речей та цифровізації міської інфраструктури.

Метою кваліфікаційної роботи є проектування та програмно-апаратна реалізація кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу.

Об'єктом дослідження є процес автоматизованого моніторингу контейнерів для збору твердих побутових відходів.

Предметом дослідження є методи, програмно-апаратні засоби та технології побудови кіберфізичних систем моніторингу в режимі реального часу.

Практичне значення роботи полягає у створенні програмно-апаратного прототипу кіберфізичної системи, який забезпечує автоматичне визначення рівня заповнення контейнерів, передачу телеметричних даних на сервер та відображення актуальної інформації у користувацькому інтерфейсі. Результати роботи можуть бути використані під час створення або модернізації систем управління відходами на рівні окремих підприємств чи муніципальних служб.

Сфера застосування розробки охоплює системи автоматизації міської інфраструктури, підприємства житлово-комунального господарства, муніципальні інформаційні системи, IoT-платформи та інші програмно-технічні комплекси, що потребують організації дистанційного моніторингу об'єктів у режимі реального часу.

					КвРКІ.022114.22.02.11 ПЗ	Арк.
						5
Зм.	Арк.	№докум.	Підпис	Дата		

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

На сучасному етапі розвитку інформаційних технологій особлива увага приділяється автоматизації процесів управління об'єктами міської інфраструктури. Одним із ключових напрямів цифрової трансформації міського середовища є впровадження інтелектуальних систем моніторингу, управління та аналізу стану комунальних об'єктів у режимі реального часу. До таких об'єктів належать системи освітлення, транспортна інфраструктура, системи безпеки, енергомережі, а також системи збору та утилізації побутових відходів.

Розвиток концепції «розумного міста» передбачає широке використання сучасних інформаційно-комунікаційних технологій для підвищення ефективності роботи міських служб, оптимізації використання ресурсів та покращення якості життя населення. У межах даної концепції особливе значення мають системи автоматизованого збору та аналізу даних, які дозволяють оперативно отримувати інформацію про стан інфраструктурних об'єктів і приймати обґрунтовані управлінські рішення.

Зростання чисельності населення, урбанізація та збільшення обсягів споживання ресурсів призводять до постійного збільшення кількості твердих побутових відходів. Це створює додаткове навантаження на міську інфраструктуру та вимагає впровадження сучасних технологічних рішень для ефективного управління процесами збору, транспортування та утилізації сміття. Водночас підвищуються вимоги до екологічної безпеки, санітарного стану населених пунктів та раціонального використання матеріальних і транспортних ресурсів.

					КвРКІ.022114.22.02.11 ПЗ	Арк.
						6
Зм.	Арк.	№докум.	Підпис	Дата		

Традиційна організація збору відходів (рис. 1.1) переважно базується на фіксованих графіках обслуговування смітєвих контейнерів незалежно від їх фактичного стану заповнення. Такий підхід є простим в організації, проте не враховує реальні потреби окремих районів і не дозволяє оперативно реагувати на зміну інтенсивності накопичення відходів. У результаті одні контейнери можуть тривалий час залишатися частково порожніми, тоді як інші швидко переповнюються ще до прибуття спеціалізованого транспорту.



Рисунок 1.1 – Традиційна організація збору відходів

Подібна організація роботи супроводжується низкою суттєвих недоліків. До них належать переповнення контейнерів у періоди підвищеного навантаження, несвоєчасне вивезення відходів, неефективне використання транспортних ресурсів, збільшення витрат пального та експлуатаційних витрат комунальних підприємств. Крім того, накопичення сміття навколо переповнених контейнерів негативно впливає на санітарно-епідеміологічний стан територій, погіршує екологічну ситуацію та створює дискомфорт для мешканців.

У зв'язку з цим виникає потреба у впровадженні інтелектуальних систем моніторингу, здатних забезпечувати безперервний контроль стану контейнерів

та автоматизоване отримання інформації про їх рівень заповнення. Використання сучасних сенсорних технологій, засобів бездротового зв'язку та програмних платформ для обробки даних дозволяє перейти від планового до адаптивного підходу в організації збору відходів. Це сприяє підвищенню ефективності роботи комунальних служб, оптимізації маршрутів спецтехніки та зменшенню витрат на обслуговування системи збору сміття.

Такий підхід має ряд суттєвих недоліків, серед яких: переповнення контейнерів у періоди підвищеного навантаження, несвоєчасне вивезення відходів, неефективне використання транспортних ресурсів, підвищені витрати пального та експлуатаційних ресурсів, погіршення санітарного стану навколишнього середовища, складність оперативного реагування на аварійні ситуації.

Для вирішення зазначених проблем активно впроваджуються інтелектуальні системи управління відходами, що базуються на технологіях.

Інтернету речей, хмарних обчисленнях, бездротових мережах передачі даних та методах автоматизованого аналізу інформації.

Одним із перспективних напрямів у даній сфері є застосування кіберфізичних систем. Кіберфізична система являє собою інтегровану сукупність апаратних компонентів, програмного забезпечення, сенсорних пристроїв, комунікаційних модулів та аналітичних сервісів, що забезпечують взаємодію між фізичними об'єктами та цифровим середовищем.

У контексті моніторингу збору відходів кіберфізична система забезпечує автоматизований контроль параметрів сміттевого контейнера, збір інформації з сенсорів, передачу даних на сервер обробки, аналіз отриманих показників та відображення актуального стану контейнерів у користувацькому інтерфейсі.

Основними функціональними компонентами такої системи є: фізичний об'єкт моніторингу – контейнер для збору відходів, сенсорний модуль, призначений для вимірювання рівня заповнення контейнера, обчислювальний модуль, який виконує локальну обробку даних, комунікаційний модуль для

					КвРКІ.022114.22.02.11 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		8

передачі інформації через бездротові канали зв'язку, серверна інфраструктура для накопичення, обробки та аналізу даних, клієнтський інтерфейс для візуалізації інформації та взаємодії з оператором системи.

Структурно система моніторингу збору відходів може бути представлена у вигляді багаторівневої архітектури, яка включає:

- сенсорний рівень – збір даних про стан контейнера;
- мережевий рівень – передача телеметричної інформації;
- рівень обробки даних – серверна обробка та збереження інформації;
- рівень представлення – візуалізація інформації для користувача.

Функціонування системи передбачає циклічне виконання таких процесів: зчитування показників із сенсорів, первинна обробка отриманих даних, передача інформації через бездротову мережу, збереження даних у базі даних, аналіз отриманої інформації, формування сповіщень про критичні події, відображення актуального стану контейнерів.

Таким чином, проведений аналіз предметної області показав, що існуючі методи організації збору відходів мають ряд суттєвих обмежень, які можуть бути усунені шляхом впровадження кіберфізичних систем моніторингу.

Використання сучасних сенсорних технологій, мікроконтролерних платформ та засобів бездротового зв'язку дозволяє реалізувати інтелектуальну систему, здатну забезпечити безперервний контроль стану сміттєвих контейнерів та підвищити ефективність управління процесами збору відходів.

1.2 Аналіз наявного програмно–апаратного забезпечення предметної області

Активний розвиток технологій Інтернету речей (IoT), вбудованих систем і бездротових комунікацій сприяв появі інтелектуальних систем моніторингу збору твердих побутових відходів. Сучасні програмно–апаратні комплекси дозволяють автоматично контролювати стан контейнерів, визначати рівень їх

					КвРКІ.022114.22.02.11 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		9

заповнення, збирати статистичні дані та оптимізувати маршрути обслуговування.

Для формування вимог до розроблюваної системи було проведено аналіз існуючих комерційних та дослідницьких рішень(табл. 1.1).

Одним із відомих рішень є Enevo, яка використовує ультразвукові сенсори та хмарну обробку даних для моніторингу заповнення контейнерів і планування маршрутів. Перевагами системи є висока точність і розвинена аналітика, недоліками – висока вартість впровадження та залежність від хмарної інфраструктури.

Іншим поширеним рішенням є Bigbelly, що поєднує функції моніторингу та механічного ущільнення відходів. Система забезпечує високу автономність і зменшує частоту обслуговування, однак характеризується складною конструкцією, підвищеним енергоспоживанням і значною вартістю обслуговування.

Перспективним рішенням також є Ecube Labs, яке використовує бездротові сенсорні вузли, підтримує GPS–моніторинг та інтеграцію з платформами «розумного міста». Основними перевагами є модульність і масштабованість, недоліками – складність початкового налаштування та залежність від стабільного мережевого покриття.

Крім промислових рішень, поширеними є навчальні та дослідницькі проекти на базі Arduino та ESP32. Такі системи зазвичай використовують ультразвукові датчики, модулі Wi-Fi, GSM або LoRa та серверні платформи для обробки даних.

Перевагами таких рішень є низька вартість, гнучкість та простота модифікації, а недоліками – обмежена промислова надійність, потреба в додатковому налаштуванні безпеки та нижча масштабованість.

Таблиця 1.1 – Результати проведеного аналізу.

					КвРКІ.022114.22.02.11 ПЗ	Арк. 10
Зм.	Арк.	№докум.	Підпис	Дата		

Система	Тип сенсорів	Канал зв'язку	Аналітика	Мобільний інтерфейс	Переваги	Недоліки
Enevo	Ультразвукові	GSM	Так	Так	Висока точність, прогнозування	Висока вартість
Bigbelly	Ультразвукові	GSM/Wi-Fi	Так	Так	Автономність, пресування	Складність конструкції
Ecube Labs	Ультразвукові	LoRa/GSM	Так	Так	Масштабованість	Складність налаштування
Arduino/ESP-рішення	Ультразвукові	Wi-Fi/LoRa	Частково	Частково	Низька вартість, гнучкість	Обмежена надійність

Проведений аналіз існуючих програмно-апаратних засобів показав, що сучасні системи моніторингу збору відходів здатні ефективно вирішувати задачі автоматичного контролю заповнення контейнерів, збору статистичних даних та оптимізації логістичних процесів. Водночас більшість промислових рішень характеризується високою вартістю впровадження, складністю масштабування або залежністю від закритих хмарних платформ.

Таким чином, актуальним є розроблення власної кіберфізичної системи моніторингу збору відходів, яка поєднуватиме функціональність сучасних комерційних рішень із доступністю, гнучкістю налаштування та можливістю подальшої модернізації відповідно до конкретних умов експлуатації.

1.3 Визначення вимог до кіберфізичної системи та постановка задачі розробки

У результаті проведеного аналізу предметної області та існуючих програмно-апаратних рішень встановлено, що актуальним напрямом розвитку систем моніторингу збору відходів є створення доступних, масштабованих та енергоефективних кіберфізичних систем, здатних забезпечувати автоматичний збір, передачу, обробку та візуалізацію інформації про стан контейнерів у режимі реального часу.

Сучасні тенденції розвитку технологій Інтернету речей, бездротових мереж передачі даних та хмарних сервісів створюють передумови для впровадження інтелектуальних систем управління відходами, які дозволяють підвищити ефективність роботи комунальних служб, зменшити експлуатаційні витрати та покращити санітарний стан міського середовища. Особливо важливим є забезпечення своєчасного отримання інформації про стан контейнерів, що дозволяє оперативно реагувати на їх переповнення та уникати виникнення проблемних ситуацій.

Основним призначенням розроблюваної системи є автоматизація процесу контролю стану сміттєвих контейнерів та забезпечення оперативного отримання достовірної інформації про рівень їх заповнення, технічний стан і критичні події. Запропонована система повинна забезпечувати безперервний моніторинг об'єктів контролю, накопичення історичних даних та підтримку прийняття рішень щодо організації процесу збору відходів.

На основі отриманих результатів аналізу сформовано основні вимоги до кіберфізичної системи «Розумні сміттєві контейнери». Визначені вимоги охоплюють функціональні можливості системи, особливості її програмно-апаратної реалізації, вимоги до надійності роботи, енергоефективності, масштабованості та безпеки передачі даних.

					КьРКІ.022114.22.02.11 ПЗ	Арк. 12
Зм.	Арк.	№докум.	Підпис	Дата		

Розроблювана система повинна забезпечувати виконання таких функцій (рис. 1.2): автоматичне вимірювання рівня заповнення контейнера у задані часові інтервали, збір даних із сенсорних модулів у режимі реального часу, локальну обробку отриманої інформації на рівні мікроконтролера, передачу телеметричних даних на сервер через бездротовий канал зв'язку, збереження отриманих даних у базі даних, відображення поточного стану контейнерів у графічному інтерфейсі користувача, автоматичне формування повідомлень при досягненні критичного рівня заповнення, ведення історії вимірювань та статистичних показників, можливість перегляду місцезнаходження контейнерів на карті, а також підтримку одночасної роботи з декількома контейнерами.

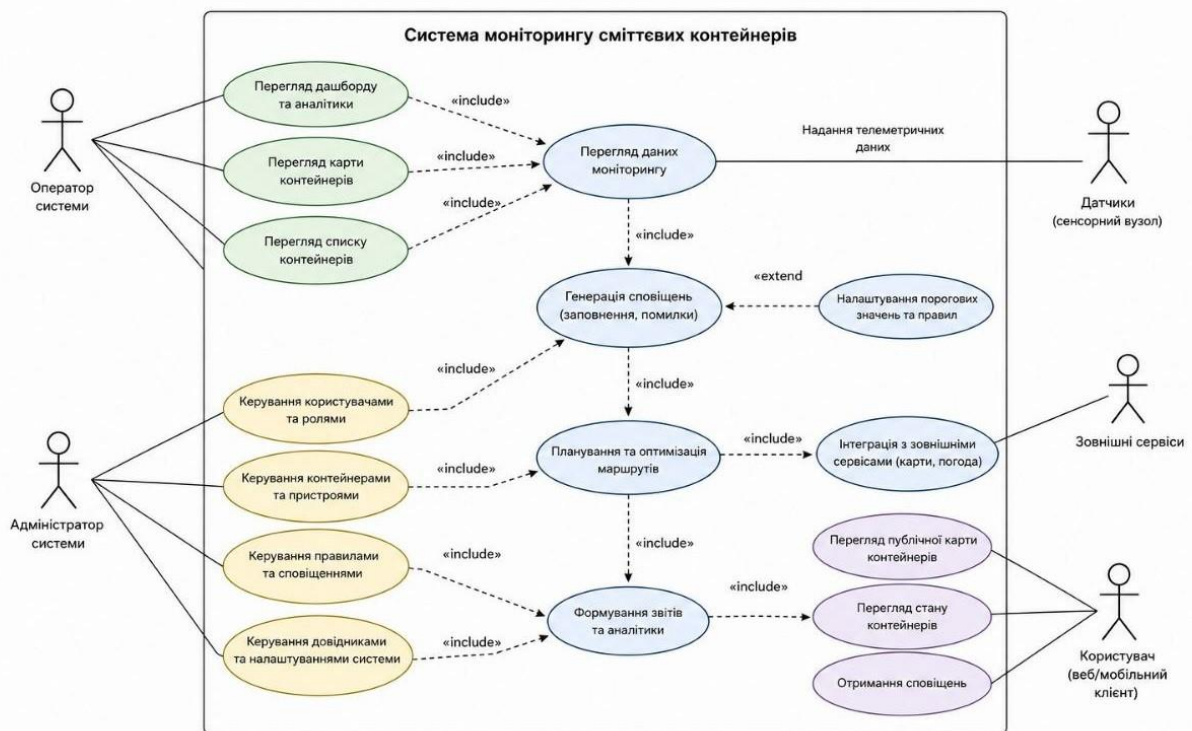


Рисунок 1.2 – Функції забезпечення системи

Крім основних функцій моніторингу, система повинна забезпечувати можливість подальшого розширення функціоналу шляхом підключення додаткових сенсорів, збільшення кількості контрольованих контейнерів та інтеграції з іншими інформаційними системами міської інфраструктури. Такий

підхід дозволяє створити універсальну платформу для побудови сучасних рішень у сфері управління відходами та реалізації концепції «розумного міста».

Таким чином, сформовані вимоги визначають основні характеристики майбутньої кіберфізичної системи та створюють основу для її подальшого проектування, програмно-апаратної реалізації й експериментального дослідження ефективності роботи.

Окрім функціональних характеристик, система повинна відповідати ряду нефункціональних вимог: надійність (система повинна забезпечувати стабільну роботу в умовах безперервної експлуатації та зберігати працездатність при тимчасовій втраті зв'язку), масштабованість (архітектура системи повинна дозволяти підключення нових контейнерів без суттєвої зміни програмного забезпечення або серверної інфраструктури), енергоефективність (сенсорні вузли повинні забезпечувати мінімальне енергоспоживання для тривалої автономної роботи), відмово стійкість (при виникненні помилок передачі даних система повинна підтримувати повторне надсилання інформації або тимчасове локальне збереження даних) безпека(передача даних між фізичними пристроями та сервером повинна бути захищеною від несанкціонованого доступу), зручність використання (інтерфейс користувача повинен забезпечувати зрозуміле відображення інформації та швидкий доступ до актуальних показників системи).

Апаратна частина системи повинна включати: мікроконтролерний модуль для збору та обробки інформації, сенсор заповнення контейнера, модуль бездротової передачі даних, автономне джерело живлення; захищений корпус для зовнішньої експлуатації.

Для реалізації системи доцільним є використання мікроконтролерної платформи на базі Espressif Systems серії ESP, що забезпечує достатню обчислювальну потужність, інтегровані засоби бездротового зв'язку та низьке енергоспоживання.

Програмне забезпечення системи повинно забезпечувати: зчитування показників сенсорів, обробку та фільтрацію даних, формування пакетів

					КвРКІ.022114.22.02.11 ПЗ	Арк. 14
Зм.	Арк.	№докум.	Підпис	Дата		

телеметрії, передачу інформації на сервер, приймання, обробку та збереження даних на серверній стороні, візуалізацію інформації у вебінтерфейсі або мобільному застосунку.

Відповідно до сформованих вимог, у межах кваліфікаційної роботи необхідно вирішити задачу розробки кіберфізичної системи «Розумні сміттєві контейнери», яка забезпечуватиме автоматизований моніторинг процесу збору відходів у режимі реального часу.

Для досягнення поставленої задачі необхідно: розробити структурну та функціональну архітектуру системи, обрати апаратні компоненти сенсорного вузла, розробити алгоритми збору, обробки та передачі даних, реалізувати вбудоване програмне забезпечення мікроконтролера, реалізувати серверну частину системи та базу даних, розробити користувацький інтерфейс моніторингу, провести тестування системи та оцінити ефективність її роботи.

Таким чином, сформовані функціональні, нефункціональні та технічні вимоги визначають основу для подальшого проєктування, програмно–апаратної реалізації та тестування кіберфізичної системи моніторингу збору відходів у режимі реального часу.

1.1 Висновки до першого розділу

У першому розділі кваліфікаційної роботи було проведено дослідження предметної області, пов'язаної з автоматизацією процесів моніторингу збору твердих побутових відходів, а також виконано аналіз сучасних підходів до побудови інтелектуальних систем управління об'єктами міської інфраструктури.

У ході виконання змістовного аналізу предметної області встановлено, що традиційні методи організації збору відходів, які базуються на фіксованих графіках обслуговування контейнерів, характеризуються низькою адаптивністю до реального стану заповнення контейнерів, що призводить до неефективного

використання ресурсів, збільшення експлуатаційних витрат та погіршення санітарного стану міського середовища.

Проведений аналіз існуючих програмно–апаратних рішень у сфері моніторингу збору відходів показав, що сучасні системи здатні забезпечувати автоматичний збір телеметричних даних, дистанційний контроль стану контейнерів, аналітичну обробку інформації та оптимізацію логістичних процесів. Водночас більшість наявних комерційних рішень характеризується високою вартістю впровадження, складністю інтеграції або залежністю від закритої хмарної інфраструктури.

На основі результатів проведеного аналізу було сформовано функціональні, нефункціональні та технічні вимоги до розроблюваної кіберфізичної системи «Розумні сміттєві контейнери», визначено основні задачі розробки, а також окреслено ключові характеристики майбутнього програмно–апаратного комплексу.

Отримані результати дослідження підтвердили доцільність розробки власної кіберфізичної системи моніторингу збору відходів, яка поєднуватиме доступність, масштабованість, енергоефективність та можливість роботи в режимі реального часу.

На основі сформованих вимог у наступному розділі буде виконано проектування архітектури системи, обґрунтовано вибір апаратних компонентів, розроблено структурну модель та визначено програмні засоби реалізації системи.

					КвРКІ.022114.22.02.11 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		16

2 ПРОЄКТУВАННЯ ПРОГРАМНО–ТЕХНІЧНОГО ЗАСОБУ

2.1 Розробка загальної архітектури кіберфізичної системи

На основі проведеного аналізу предметної області та сформованих функціональних вимог було розроблено загальну архітектуру кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу.

Система призначена для автоматизованого контролю стану контейнерів, визначення рівня їх заповнення, збору телеметричних даних, передачі інформації на сервер, її подальшої обробки, збереження та візуалізації. Реалізація такої системи дозволяє підвищити ефективність організації збору відходів, оптимізувати логістичні витрати та забезпечити своєчасне реагування на критичні ситуації.

Кіберфізична система реалізована за багаторівневою архітектурою, що забезпечує модульність, масштабованість та можливість подальшої модернізації. Загальну структуру системи (рис. 2.1)

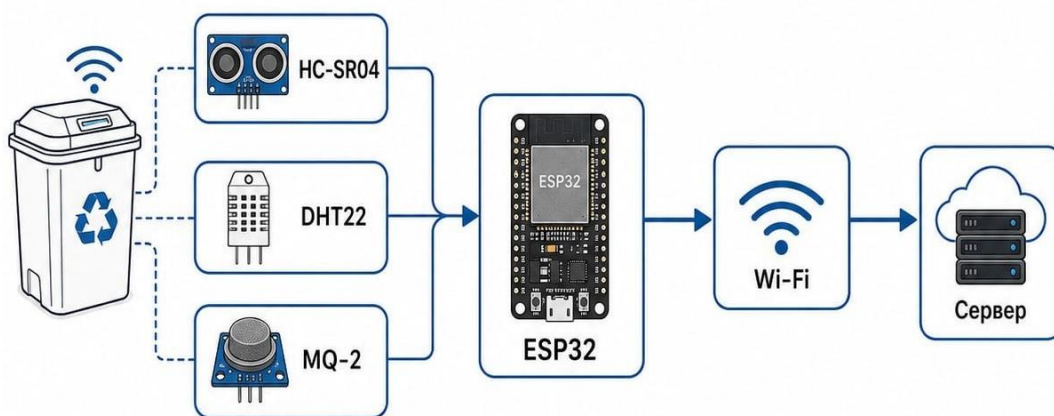


Рисунок 2.1. – Загальна структура системи

Архітектура включає такі основні рівні:

- сенсорний рівень;
- рівень локальної обробки даних;

Зм.	Арк.	№докум.	Підпис	Дата

- комунікаційний рівень;
- серверний рівень;
- рівень представлення інформації.

Сенсорний рівень забезпечує безпосередню взаємодію з фізичним середовищем та збір первинних даних про стан контейнера.

До його складу входять: ультразвуковий датчик HC-SR04 для визначення рівня заповнення, датчик температури та вологості DHT22, газовий сенсор MQ-2, модуль контролю живлення.

Основною функцією даного рівня є перетворення фізичних параметрів у цифрові сигнали для подальшої обробки.

Цей рівень відповідає за зчитування, перевірку та попередню обробку даних, отриманих від сенсорів.

Центральним елементом є мікроконтролер ESP32, який виконує: ініціалізацію сенсорів, зчитування показників, фільтрацію похибок, розрахунок рівня заповнення контейнера, формування телеметричного пакета, контроль енергоспоживання.

Після обробки сформовані дані передаються на комунікаційний рівень.

Комунікаційний рівень забезпечує передачу телеметричних даних між сенсорним вузлом і сервером.

У межах проєкту використовується бездротове з'єднання Wi-Fi, яке підтримується платформою ESP32. Для обміну даними можуть використовуватись протоколи HTTP або MQTT.

Серверний рівень є центральною частиною системи та забезпечує прийом, обробку, аналіз і збереження телеметричних даних.

Сервер виконує перевірку отриманих повідомлень, збереження історії вимірювань та виявлення критичних станів контейнерів.

Даний рівень забезпечує взаємодію користувача із системою за допомогою вебінтерфейсу.

Основні функції: відображення рівня заповнення контейнерів, візуалізація контейнерів на карті, перегляд історії вимірювань, отримання системних сповіщень, формування статистичних звітів.

Такої логіки взаємодії компонентів(рис. 2.2) було обрано дотримуватись для зручності і функціонування системи.



Рисунок 2.2 – Логіка взаємодії компонентів

Таким чином, розроблена архітектура забезпечує повний цикл роботи системи: від збору фізичних параметрів до їх обробки, збереження та відображення користувачу в режимі реального часу. Використання модульного підходу забезпечує гнучкість, масштабованість та можливість подальшого розвитку системи.

2.2 Проектування програмного забезпечення кіберфізичної системи

Для забезпечення функціонування кіберфізичної системи «Розумні сміттєві контейнери» було виконано проектування програмного забезпечення, яке забезпечує збір телеметричних даних, їх обробку, передачу на сервер,

збереження, аналіз та візуалізацію. Програмне забезпечення побудовано за модульним принципом, що забезпечує масштабованість, гнучкість та спрощує подальшу модернізацію системи.

Відповідно до архітектури системи програмне забезпечення складається з трьох рівнів(рис. 2.3):вбудоване програмне забезпечення сенсорного вузла, серверне програмне забезпечення, програмне забезпечення користувацького інтерфейсу.

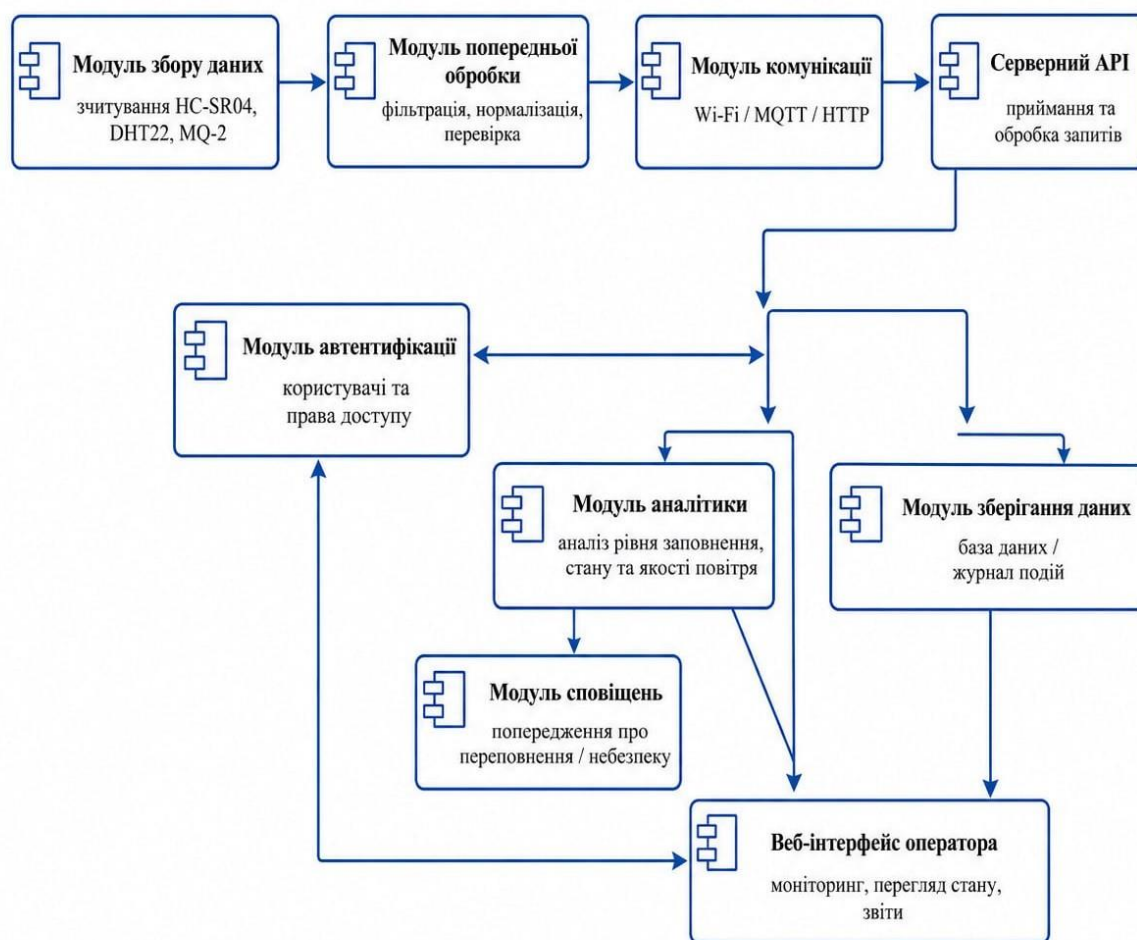


Рисунок 2.3 – Архітектура програмного забезпечення

Вбудоване програмне забезпечення реалізується на мікроконтролері ESP32 та забезпечує взаємодію з сенсорними модулями.

Основні функції: ініціалізація датчиків, зчитування показників, фільтрація похибок, розрахунок рівня заповнення, формування телеметричного пакета,

передача даних через Wi-Fi, контроль живлення, перехід у режим енергозбереження.

Робота модуля організована циклічно: вимірювання, обробка, передача, очікування.

Рівень заповнення контейнера визначається за формулою:

$$P = \frac{H}{H - D} * 100\%$$

де:

P – рівень заповнення, %;

H – висота контейнера;

D – виміряна відстань.

Для класифікації стану використовуються такі порогові значення:

Від 0 – 50% – нормальний стан контейнера;

Від 51 – 80% – частково заповнений стан контейнера;

Більше 80% – критичне заповнення контейнера

Для передачі даних використовується формат JSON.

Проектування серверного програмного забезпечення

Серверне програмне забезпечення забезпечує прийом, перевірку, збереження та аналіз телеметричних даних.

До його складу входять: модуль прийому даних, модуль валідації, модуль обробки, модуль збереження, модуль аналітики, модуль сповіщень, модуль автентифікації, API(табл.. 2.1).

Сервер аналізує отримані показники, визначає стан контейнерів, формує події та передає дані користувачькому інтерфейсу.

Таблиця 2.1 – Взаємодії компонентів за допомогою REAST API

Метод	Маршрут	Призначення
POST	/api/measurements	Прийом телеметричних даних від сенсорного вузла
GET	/api/containers	Отримання списку контейнерів
GET	/api/containers/{id}	Отримання детальної інформації про контейнери
GET	/api/measurements {containers_id}	Отримання історії вимірювань
GET	/api/events	Отримання списку подій
POST	/api/auth/login	Авторизація користувача
GET	/api/statistics	Отримання статистичних даних

Користувацький інтерфейс реалізується у вигляді вебзастосунку та забезпечує моніторинг стану контейнерів.

Основні функції: перегляд списку контейнерів, відображення контейнерів на карті.

Контейнери відображаються кольоровими маркерами:

- зелений – нормальний стан;
- помаранчевий – підвищене заповнення;
- червоний – критичний стан;
- сірий – відсутній зв'язок.

Логіка обробки подій

Система автоматично формує події при перевищенні порогових значень(табл. 2.2)

Таблиця 2.2 – Приклади умов створення подій:

Умова	Тип події	Пріоритет
Рівень заповнення понад 80%	Переповнення контейнера	Високий
Рівень газу перевищує допустиму межу	Небезпечний рівень газу	Високий
Температура перевищує норму	Підвищена температура	Середній
Заряд батареї нижче 20%	Низький заряд пристрою	Середній
Потрібне очищення або перевірки	Обслуговування	Низький/Середній
Данні не надходять тривалий час	Втрата зв'язку	Високий

Для підвищення стабільності роботи передбачено: повторне підключення до мережі, повторна передача даних, локальне збереження при втраті зв'язку, журналювання подій, контроль активності пристроїв.

У результаті проєктування програмного забезпечення визначено основні програмні модулі, що забезпечують повний цикл роботи системи: збір даних, передачу, обробку, збереження та візуалізацію. Модульна архітектура забезпечує надійність, масштабованість та можливість подальшого розвитку системи.

2.3 Проєктування бази даних та серверної інфраструктури

Для забезпечення стабільної роботи кіберфізичної системи «Розумні сміттєві контейнери» необхідною є серверна інфраструктура, яка забезпечує прийом телеметричних даних від сенсорних вузлів, їх перевірку, збереження, обробку та відображення у користувацькому інтерфейсі.

Таблиця 2.3 – Зереження інформації про контейнери.

Поле	Тип даних	Опис
Container_id	INT, PK	Унікальний ідентифікатор контейнера
Location	VARCHAR	Графічне або текстове місцезнаходження
Address	VARCHAR	Адреса розташування контейнера
Type	VARCHAR	Тип контейнера
Status	VARCHAR	Поточний стан контейнера
Created_at	DATETIME	Дата додавання контейнера до системи

Таблиця Devices призначена для збереження інформації про сенсорні вузли, встановлені на контейнерах(табл. 2.4).

Таблиця 2.4 – Структура таблиць Devices:

Поле	Тип даних	Опис
Device_id	INT, PK	Унікальний ідентифікатор пристрою
Container_id	INT, FK	Ідентифікатор контейнера
Esp32_uid	VARCHAR	Унікальний ідентифікатор ESP32
Install_date	DATETIME	Дата встановлення пристрою
Battery_level	INT	Рівень заряду пристрою
Status	VARCHAR	Поточний стан пристрою

Таблиця Measurements призначена для збереження телеметричних даних, які надходять від сенсорних вузлів(табл.. 2.5). Вона є однією з найбільш активно

Таблиця Events(табл. 2.7) призначена для збереження системних подій, які виникають у процесі роботи системи.

Таблиця 2.7 – Структура таблиці Events:

Поле	Тип даних	Опис
Event_id	INT, PK	Унікальний ідентифікатор події
User_id	INT, FK	Ідентифікатор користувача
Container_id	INT, FK	Ідентифікатор контейнера
Event_type	VARCHAR	Тип події
Description	TEXT	Опис події
Priority	VARCHAR	Пріоритет події
Status	VARCHAR	Стан обробки події
Created_at	DATETIME	Дата та час створення події

Для візуального представлення структури використовується ER–діаграма.

Таблиця 2.8 – Основні зв'язки між таблицями:

Зв'язок	Тип зв'язку	Опис
Containers – Devices	1:N	Один контейнер може мати декілька пристроїв
Devices – Measurements	1:N	Один пристрій може створювати багато вимірювань
Users – Events	1:N	Один користувач може бути пов'язаний з багатьма подіями
Containers – Events	1:N	Один контейнер може мати багато подій

Для обміну даними між сенсорними вузлами, сервером і користувацьким інтерфейсом використовується API(табл. 2.9).

Таблиця 2.9 – Основні API–запити системи:

Метод	Маршрут	Призначення
POST	/api/measurements	Передача телеметричних даних від сенсорного вузла
GET	/api/containers	Отримання списку контейнерів
GET	/api/containers/{id}	Отримання детальної інформації про контейнер
GET	/api/measurements/{device_id}	Отримання історії вимірювань пристрою
GET	/api/events	Отримання списку подій
POST	/api/events	Створення нової події
PUT	/api/events/{id}	Оновлення статусу події
POST	/api/auth/login	Авторизація користувача
GET	/api/statistics	Отримання статистичних даних

Після отримання телеметричного повідомлення сервер виконує: прийом даних, перевірку структури, ідентифікацію пристрою, перевірку допустимих значень, запис у базу даних, оновлення стану контейнера, перевірку порогових значень, створення події при критичному стані.

Наприклад, при заповненні контейнера понад 80 % сервер створює подію типу «переповнення».

Для безпеки системи реалізовано: автентифікацію користувачів, розмежування прав доступу(табл. 2.10), збереження паролів у вигляді хешів, токени доступу, журналювання дій, перевірку ідентифікаторів пристроїв.

представлення даних. Такий підхід забезпечує модульність, масштабованість і можливість подальшого розвитку системи.

Для реалізації сенсорного вузла обрано мікроконтролер ESP32, який забезпечує достатню обчислювальну продуктивність, підтримку Wi-Fi та низьке енергоспоживання. Для контролю рівня заповнення контейнера використано ультразвуковий датчик HC-SR04, а для моніторингу параметрів середовища – датчики DHT22 та MQ-2. Передбачено автономне живлення та використання режиму енергозбереження Deep Sleep.

Програмне забезпечення системи спроектовано за модульним принципом і поділено на три основні частини: вбудоване програмне забезпечення сенсорного вузла, серверне програмне забезпечення та користувацький інтерфейс. Розроблено логіку циклічної роботи сенсорного вузла, яка включає зчитування показників, їх обробку, формування телеметричного пакета та передачу даних на сервер.

Серверну частину системи спроектовано у вигляді набору взаємопов'язаних модулів, що забезпечують прийом, перевірку, збереження, аналіз телеметричних даних та формування сповіщень. Для передачі інформації між компонентами системи використовується формат JSON.

У межах інформаційного забезпечення спроектовано структуру бази даних, яка включає таблиці контейнерів, пристроїв, вимірювань, користувачів і подій. Для збереження структурованої інформації запропоновано використання PostgreSQL, що забезпечує надійність, цілісність даних, підтримку зв'язків між таблицями та можливість масштабування.

Також спроектовано серверну інфраструктуру, яка включає API, базу даних, модуль аналітики, модуль сповіщень та систему резервного копіювання. Це дозволяє централізовано обробляти дані від великої кількості сенсорних вузлів та забезпечувати стабільну роботу системи.

Користувацький інтерфейс запропоновано реалізувати у веборієнтованому форматі. Визначено основні функціональні модулі: авторизація, інформаційна

панель, карта контейнерів, сторінка детального моніторингу, статистика, сповіщення та адміністрування. Використання кольорової індикації дозволяє оператору швидко оцінювати поточний стан контейнерів і оперативно реагувати на критичні ситуації.

Таким чином, у другому розділі сформовано повну проєктну основу для подальшої програмно-апаратної реалізації, інтеграції компонентів та проведення експериментального тестування системи.

					КвРКІ.022114.22.02.11 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		31

3 РЕАЛІЗАЦІЯ ПРОГРАМНО – ТЕХНІЧНОГО ЗАСОБУ

3.1 Реалізація апаратного прототипу сенсорного вузла

На основі проєктних рішень, розглянутих у другому розділі, було виконано реалізацію апаратного прототипу сенсорного вузла кіберфізичної системи «Розумні сміттєві контейнери». Основним призначенням сенсорного вузла є збирання даних про стан сміттєвого контейнера, їх попередня обробка та передача на серверну частину системи.

Апаратний прототип складається з мікроконтролерного модуля ESP32, сенсора рівня заповнення HC-SR04, датчика температури та вологості DHT22, газового сенсора MQ-2, модуля живлення та захисного корпусу. Така конфігурація компонентів дозволяє забезпечити багатопараметричний моніторинг стану контейнера та підвищити загальну точність оцінки його стану.

Загальний вигляд апаратного прототипу сенсорного вузла(рис.3.1)

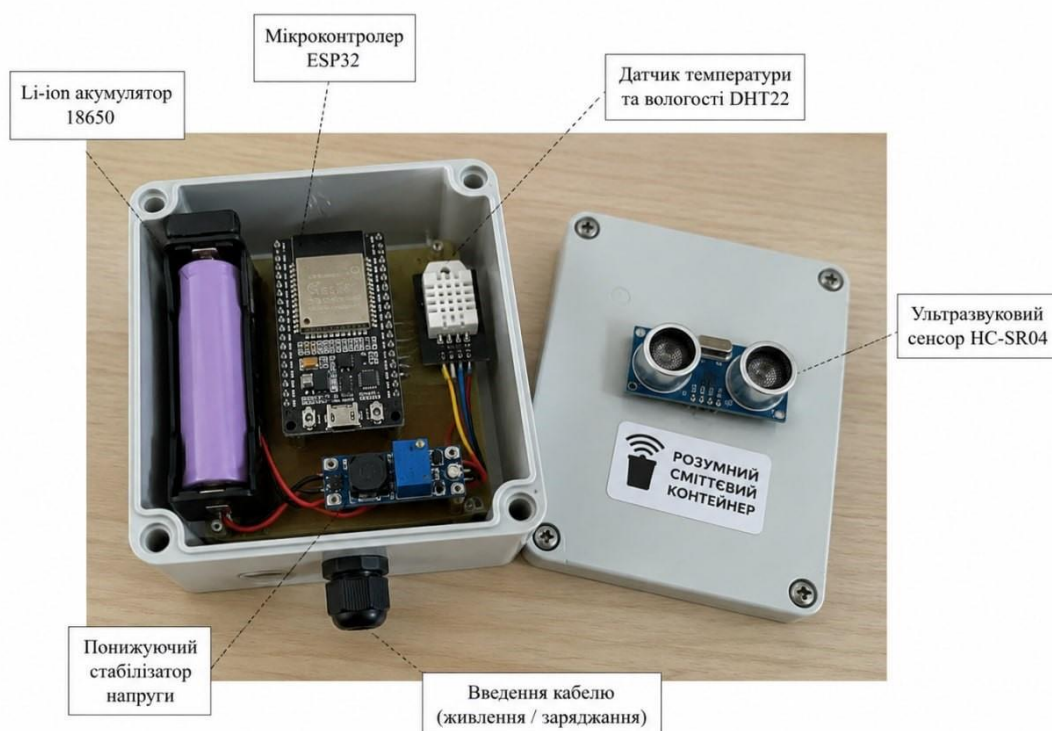


Рисунок 3.1 – Загальний вигляд апаратного прототипу
Центральним елементом прототипу є мікроконтролер ESP32. Він виконує функції керування всіма підключеними сенсорами, забезпечує зчитування

вимірювань, первинну обробку даних, фільтрацію можливих похибок та передачу інформації через Wi-Fi-з'єднання на сервер. Вибір даного модуля є обґрунтованим завдяки його високій продуктивності, енергоефективності та вбудованій підтримці бездротового зв'язку.

Для визначення рівня заповнення контейнера використовується ультразвуковий сенсор HC-SR04. Його встановлення у верхній частині контейнера дозволяє вимірювати відстань до поверхні відходів без фізичного контакту. На основі отриманих значень у програмному забезпеченні обчислюється відсотковий рівень заповнення, що дозволяє визначати стан контейнера у режимі реального часу.

Датчик DHT22 використовується для вимірювання температури та вологості всередині або навколо контейнера. Отримані дані можуть застосовуватися для аналізу умов експлуатації, виявлення аномальних змін мікроклімату та додаткового контролю середовища.

Газовий сенсор MQ-2 призначений для виявлення диму, горючих газів або потенційно небезпечних випарів. Його використання підвищує рівень безпеки системи, оскільки дозволяє оперативно фіксувати можливі аварійні або небезпечні ситуації.

Для живлення апаратного прототипу передбачено використання автономного джерела живлення. У складі вузла застосовується літій-іонний акумулятор, стабілізатор напруги та елементи захисту від перенапруги та глибокого розряду. Така конфігурація забезпечує стабільну роботу пристрою навіть у випадку відсутності зовнішнього електроживлення, що є критично важливим для системи, яка працює в польових умовах.

Підключення сенсорів до мікроконтролера ESP32 здійснюється через цифрові та аналогові входи/виходи. Усі компоненти мають спільну лінію землі, що є обов'язковою умовою коректної роботи схеми.

Ультразвуковий сенсор HC-SR04 підключається до ESP32 за допомогою контактів VCC, TRIG, ECHO та GND. Контакт TRIG використовується для

запуску ультразвукового імпульсу, а контакт ЕЧО – для отримання сигналу у відповідь. На основі часу проходження сигналу мікроконтролер визначає відстань до поверхні відходів.

Датчик DHT22 підключається до одного з цифрових GPIO–виводів ESP32. Через цей вивід передаються значення температури та вологості. Газовий сенсор MQ–2 може підключатися до аналогового входу мікроконтролера, що дозволяє отримувати числове значення рівня газу або диму. Загальна схема підключення сенсорів до мікроконтролера ESP32(рис. 3.2)

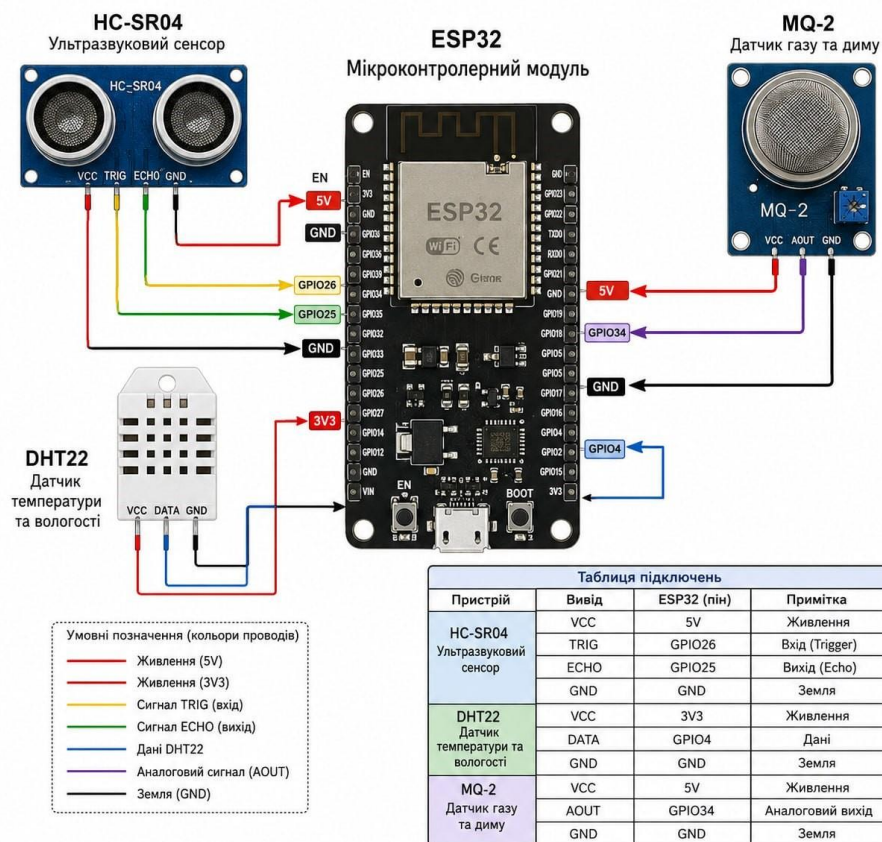


Рисунок 3.2 – Схема підключення сенсорів до мікроконтролера ESP32

У цій схемі доцільно об'єднати підключення всіх основних сенсорів: HC–SR04, DHT22 та MQ–2. Так рисунок буде більш корисним для дипломної роботи, оскільки одразу показуватиме повну апаратну структуру сенсорного вузла.

Компонент: Контакт сенсора(Контакт ESP32), призначення (HC–SR04, VCC, 5V), живлення сенсора (HC–SR04, TRIG, GPIO 26), запуск вимірювання(HC–SR04, ECHO, GPIO 25), прийом сигналу(HC–SR04, GND,

GND), земля(DHT22, VCC, 3.3V), живлення(DHT22, DATA, GPIO 4), передача даних(DHT22, GND, GND), земля(MQ-2, VCC, 5V), живлення(MQ-2, AOUT, GPIO 34), аналоговий сигнал(MQ-2, GND, GND).

Під час підключення апаратних компонентів сенсорного вузла особливу увагу необхідно приділяти узгодженню електричних параметрів окремих модулів системи. Одним із важливих аспектів є забезпечення сумісності робочих напруг між мікроконтролером та підключеними сенсорами. Мікроконтролер ESP32 працює з логічним рівнем 3,3 В, тоді як окремі периферійні пристрої можуть використовувати сигнали з рівнем 5 В. Безпосереднє підключення таких сигналів до входів мікроконтролера може призвести до некоректної роботи або навіть пошкодження електронних компонентів.

Для запобігання подібним ситуаціям доцільно використовувати дільники напруги, перетворювачі логічних рівнів або інші засоби узгодження сигналів. Застосування таких рішень дозволяє забезпечити надійний обмін даними між компонентами системи та підвищити загальну безпеку експлуатації електронного обладнання.

Важливим елементом розроблюваного сенсорного вузла є система живлення. Від її правильного проєктування значною мірою залежить стабільність функціонування пристрою, точність вимірювань та тривалість автономної роботи. Оскільки сенсорний вузол передбачається використовувати в умовах зовнішнього розміщення, де відсутнє стаціонарне джерело електроживлення, доцільним є застосування автономного джерела живлення на базі літій-іонного акумулятора.

У межах розробленого прототипу система живлення виконує декілька важливих функцій: забезпечує подачу живлення на мікроконтролер ESP32, ультразвуковий датчик HC-SR04, датчик температури та вологості DHT22, газовий сенсор MQ-2, здійснює стабілізацію напруги, захист від перевантаження та короткого замикання, а також забезпечує можливість заряджання акумулятора і контролю рівня його заряду. Реалізація таких функцій дозволяє підвищити

надійність роботи пристрою та зменшити ризик виникнення аварійних ситуацій під час експлуатації.

Для коректної роботи системи необхідно враховувати, що ESP32 потребує стабільної напруги живлення 3,3 В, тоді як окремі сенсори, зокрема ультразвуковий датчик HC-SR04, можуть працювати від напруги 5 В. У зв'язку з цим у структурі системи живлення передбачено використання стабілізаторів та перетворювачів напруги, які забезпечують необхідні параметри електроживлення для кожного компонента. Такий підхід дозволяє уникнути нестабільної роботи сенсорів та забезпечити коректне функціонування всієї системи в цілому.

Однією з важливих вимог до кіберфізичних систем моніторингу є енергоефективність. Оскільки сенсорний вузол може працювати від акумулятора протягом тривалого часу, доцільним є використання програмних механізмів зниження енергоспоживання. Для цього у програмному забезпеченні мікроконтролера може використовуватися режим енергозбереження Deep Sleep, який підтримується платформою ESP32.

У даному режимі після виконання циклу вимірювання та передачі телеметричних даних мікроконтролер переходить у стан мінімального енергоспоживання. Через заздалегідь визначений проміжок часу пристрій автоматично активується, повторює цикл збору інформації та знову переходить у режим сну. Використання такого підходу дозволяє суттєво зменшити витрати енергії та збільшити тривалість автономної роботи сенсорного вузла без необхідності частого обслуговування або заміни джерела живлення.

Не менш важливим етапом розробки є забезпечення захисту апаратних компонентів від впливу зовнішнього середовища. Оскільки система призначена для встановлення на сміттєвих контейнерах, вона повинна працювати в умовах підвищеної вологості, запиленості, перепадів температур та можливих механічних навантажень. З цією метою всі електронні компоненти розміщуються у спеціальному захисному корпусі.

					КвРКІ.022114.22.02.11 ПЗ	Арк.
						36
Зм.	Арк.	№докум.	Підпис	Дата		

Корпус виконує функцію фізичного захисту електроніки від пилу, атмосферних опадів, механічних пошкоджень і випадкового контакту з відходами. Крім того, він забезпечує зручне розміщення всіх елементів системи та спрощує процес монтажу і подальшого технічного обслуговування пристрою.

Під час проєктування конструкції корпусу необхідно враховувати особливості роботи кожного сенсора. Розташування компонентів повинно забезпечувати надійне кріплення мікроконтролера та датчиків, захист провідників і контактних з'єднань, можливість підключення зовнішнього живлення або заряджання акумулятора, а також зручний доступ до вузлів системи у випадку проведення профілактичних чи ремонтних робіт.

Ультразвуковий сенсор HC-SR04 доцільно встановлювати у верхній частині корпусу або на кришці контейнера таким чином, щоб його робоча зона була спрямована вертикально вниз у внутрішній простір контейнера. Таке розташування забезпечує найбільш точне вимірювання відстані до поверхні відходів і, відповідно, коректне визначення рівня заповнення контейнера.

Датчик температури та вологості DHT22 повинен мати вільний доступ до навколишнього повітря для отримання достовірних показників. Водночас необхідно забезпечити його захист від прямого потрапляння дощу, снігу та конденсату. Аналогічні вимоги висуваються і до газового сенсора MQ-2, який для коректної роботи потребує постійного контакту з повітряним середовищем. Тому в конструкції корпусу доцільно передбачити вентиляційні отвори, закриті захисною сіткою або спеціальними фільтрувальними елементами.

Мікроконтролер ESP32, акумулятор та елементи системи живлення рекомендується розміщувати у внутрішній частині корпусу в максимально захищеній зоні. Таке розташування зменшує ризик механічних пошкоджень, впливу вологи та забруднення, а також сприяє підвищенню загальної надійності роботи сенсорного вузла протягом тривалого періоду експлуатації. Схему розміщення основних апаратних компонентів усередині захисного корпусу (рис

3.3), взаємне розташування мікроконтролера, сенсорів, елементів живлення та допоміжних модулів системи

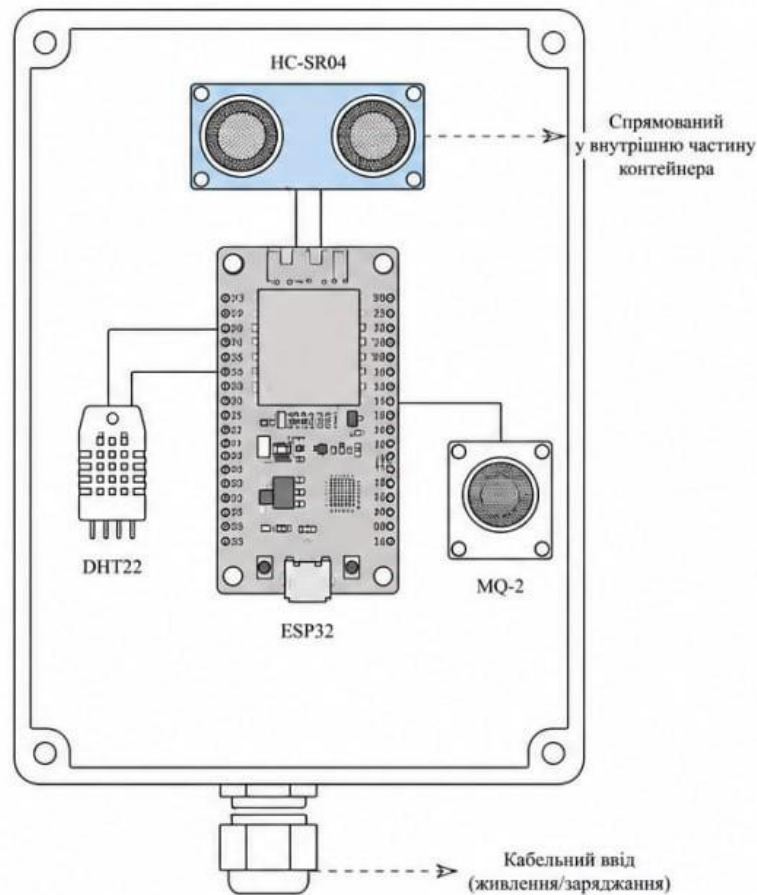


Рисунок 3.3 – Розміщення апаратних компонентів у захисному корпусі

Після подачі живлення мікроконтролер ESP32 виконує ініціалізацію підключених сенсорів і перевіряє їх готовність до роботи. Далі система переходить до зчитування показників.

Спочатку ультразвуковий сенсор HC–SR04 вимірює відстань до поверхні відходів. На основі отриманого значення розраховується рівень заповнення контейнера. Після цього зчитуються показники температури, вологості та рівня газу або диму.

Зм.	Арк.	№докум.	Підпис	Дата

Отримані дані обробляються мікроконтролером. У разі потреби виконується фільтрація значень або усереднення кількох вимірювань. Після обробки формується телеметричний пакет, який містить ідентифікатор пристрою, рівень заповнення, температуру, вологість, рівень газу, стан батареї та час вимірювання.

Після формування пакета ESP32 встановлює Wi-Fi-з'єднання та передає дані на серверну частину системи. Якщо передача успішна, пристрій може перейти у режим енергозбереження до наступного циклу вимірювання.

Загальний алгоритм роботи апаратного прототипу можна подати так:

1. подача живлення на сенсорний вузол.
2. Запуск мікроконтролера ESP32.
3. Ініціалізація сенсорів.
4. Зчитування рівня заповнення контейнера.
5. Зчитування температури, вологості та рівня газу.
6. Обробка отриманих даних.
7. Формування телеметричного повідомлення.
8. Передача даних на сервер.
9. Перехід у режим енергозбереження.
10. Повторення циклу через заданий інтервал часу.

У результаті реалізації апаратного прототипу сенсорного вузла було зібрано пристрій, який забезпечує вимірювання основних параметрів стану смітцевого контейнера. До складу прототипу входять мікроконтролер ESP32, ультразвуковий сенсор HC-SR04, датчик температури та вологості DHT22, газовий сенсор MQ-2, система живлення та захисний корпус.

Запропонована апаратна реалізація дозволяє здійснювати моніторинг рівня заповнення контейнера, контролювати додаткові параметри середовища та передавати отримані дані на сервер. Розміщення компонентів у захисному корпусі забезпечує можливість подальшого використання пристрою в умовах зовнішньої експлуатації.

Для зменшення перевантаження розділу графічними матеріалами доцільно залишити три основні рисунки: загальний вигляд прототипу, схему підключення сенсорів до ESP32 та розміщення компонентів у корпусі. Інші технічні деталі можна описати текстом або подати у вигляді короткої таблиці підключень.

3.2 Реалізація вбудованого програмного забезпечення

Після завершення реалізації апаратної частини кіберфізичної системи було виконано розробку вбудованого програмного забезпечення, призначеного для керування сенсорним вузлом, збору телеметричних даних, їх локальної обробки та передачі на серверну частину системи.

Вбудоване програмне забезпечення є важливою складовою сенсорного вузла, оскільки саме воно забезпечує взаємодію між мікроконтролером ESP32, підключеними сенсорами та серверною інфраструктурою. Програмна частина відповідає за запуск пристрою, ініціалізацію апаратних модулів, зчитування показників, перевірку коректності даних, формування телеметричного повідомлення та його передачу через бездротову мережу.

Основною метою розробки вбудованого програмного забезпечення є забезпечення стабільної та автономної роботи сенсорного вузла в умовах реальної експлуатації. Програма повинна автоматично виконувати вимірювання, визначати стан контейнера, передавати дані на сервер і за потреби переходити в енергоощадний режим.

Для реалізації програмного забезпечення було використано середовище Arduino IDE, яке підтримує програмування мікроконтролерної платформи ESP32. Це середовище є зручним для розробки прототипів, оскільки має простий інтерфейс, велику кількість готових бібліотек та підтримує роботу з Wi-Fi, сенсорами й протоколами передачі даних.

Для роботи з апаратними компонентами у програмі використовуються бібліотеки для: керування GPIO-виводами ESP32, роботи з датчиком температури та вологості DHT22, зчитування аналогових значень із газового

підключення до бездротової мережі Wi-Fi та передачу телеметричної інформації на серверну частину системи. Модуль контролю помилок відповідає за виявлення збоїв під час роботи окремих компонентів і забезпечує коректне реагування на нестандартні ситуації. Модуль керування енергоспоживанням реалізує механізми економії енергії, що є особливо важливим для автономної роботи сенсорного вузла.

Після подачі живлення мікроконтролер ESP32 виконує запуск системи та послідовну ініціалізацію всіх програмних модулів. На цьому етапі відбувається налаштування апаратних ресурсів мікроконтролера, перевірка працездатності підключених сенсорів, запуск мережевих служб та підготовка внутрішніх структур даних, необхідних для подальшої роботи пристрою.

Під час ініціалізації виконуються такі операції: налаштування GPIO-виводів мікроконтролера, запуск послідовного інтерфейсу для налагодження, ініціалізація датчиків HC-SR04, DHT22 та MQ-2, перевірка коректності їх підключення, запуск Wi-Fi-модуля та підключення до заданої мережі, створення змінних для збереження результатів вимірювань, а також встановлення часових інтервалів між циклами опитування сенсорів.

Етап ініціалізації є одним із найважливіших у роботі системи, оскільки саме на ньому перевіряється готовність усіх програмних і апаратних компонентів до подальшої експлуатації. У разі виявлення несправностей або відсутності зв'язку з окремими модулями система повинна зафіксувати помилку, повідомити про неї через службові механізми та виконати повторну спробу підключення. Такий підхід дозволяє підвищити відмовостійкість програмного забезпечення та забезпечити стабільність роботи сенсорного вузла в реальних умовах експлуатації.

Після завершення ініціалізації запускається модуль збору даних, який забезпечує циклічне опитування підключених сенсорів і формування набору поточних показників.

Процес збору даних включає зчитування відстані за допомогою ультразвукового сенсора HC-SR04, отримання показників температури та вологості з датчика DHT22, зчитування аналогового сигналу з газового сенсора MQ-2, перевірку коректності отриманих значень та передачу результатів до модуля обробки даних.

Отримані показники температури та вологості використовуються для моніторингу умов експлуатації контейнера та можуть застосовуватися під час подальшого аналізу роботи системи. Дані з газового сенсора MQ-2 дозволяють виявляти підвищену концентрацію диму або горючих газів, що може свідчити про виникнення небезпечних ситуацій.

Основним параметром, який визначається сенсорним вузлом, є рівень заповнення контейнера. Для його розрахунку використовується ультразвуковий сенсор HC-SR04, який вимірює відстань від місця встановлення датчика до поверхні відходів. На основі отриманого значення та попередньо заданої висоти контейнера програмне забезпечення обчислює відсоток його заповнення.

Після визначення рівня заповнення виконується автоматична класифікація стану контейнера. Для цього використовуються такі порогові значення:

- до 50 % — нормальний стан;
- від 51 % до 80 % — попереджувальний стан;
- понад 80 % — критичний стан.

Якщо рівень заповнення не перевищує 50 %, контейнер вважається таким, що функціонує у штатному режимі та не потребує обслуговування. При досягненні рівня заповнення від 51 % до 80 % система формує попереджувальний статус, що свідчить про необхідність планування вивезення відходів найближчим часом. У випадку перевищення позначки 80 % контейнер отримує критичний статус, а серверна частина системи може автоматично створювати події або надсилати сповіщення оператору для оперативного реагування.

Загальний алгоритм роботи вбудованого програмного забезпечення (рис.3.5), де відображено послідовність виконання основних етапів функціонування сенсорного вузла — від ініціалізації обладнання та збору даних до їх обробки, передачі на сервер і переходу мікроконтролера в режим енергозбереження.

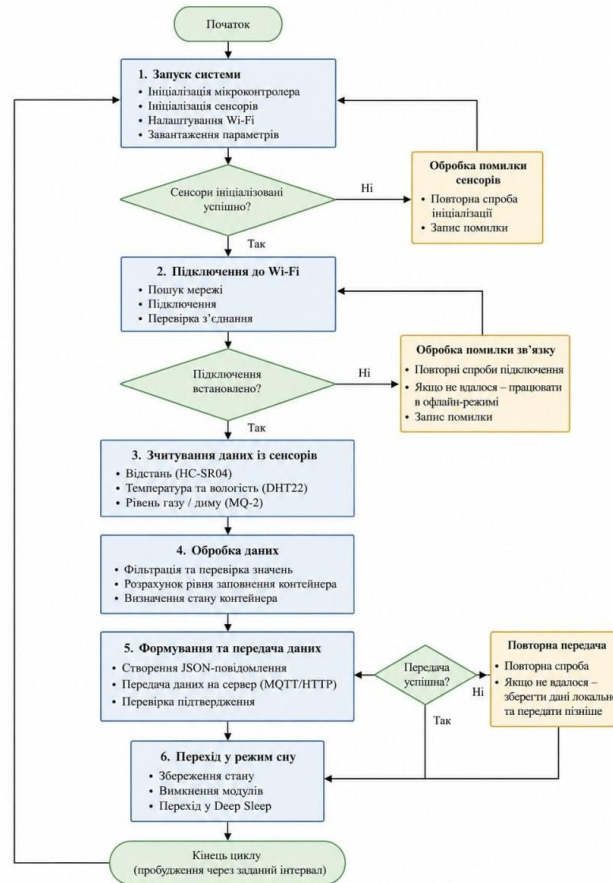


Рисунок 3.5 – Загальний алгоритм роботи вбудованого програмного забезпечення

На цьому рисунку доцільно об'єднати всі основні етапи: запуск системи, ініціалізацію, перевірку датчиків, зчитування показників, розрахунок рівня заповнення, формування JSON-повідомлення, передачу даних на сервер, обробку помилок та перехід у режим сну.

Після обробки отриманої інформації вбудоване програмне забезпечення формує телеметричний пакет і передає його на сервер. Для передачі даних використовується Wi-Fi-модуль, вбудований у ESP32.

Перед передачею формується JSON-повідомлення, яке містить основні параметри стану контейнера: ідентифікатор пристрою, ідентифікатор

контейнера, час вимірювання, рівень заповнення, температура, вологість, рівень газу, рівень заряду батареї, поточний статус контейнера.

Приклад структури повідомлення

```
{  
  "device_id": "ESP32_001",  
  "container_id": "K-001",  
  "fill_level": 82,  
  "temperature": 24.6,  
  "humidity": 58,  
  "gas_level": 130,  
  "battery": 76,  
  "status": "critical"  
}
```

Такий формат є зручним для подальшої обробки на сервері, оскільки всі параметри передаються у структурованому вигляді. Сервер може легко перевірити отримані поля, зберегти їх у базі даних та оновити стан контейнера у користувацькому інтерфейсі.

Для передачі телеметрії може використовуватися протокол MQTT або HTTP. У межах розробки прототипу доцільним є використання MQTT, оскільки він добре підходить для IoT-систем, має невелике навантаження на мережу та дозволяє ефективно передавати короткі повідомлення від пристроїв.

Для забезпечення стабільної роботи сенсорного вузла було передбачено механізми контролю та обробки помилок. Це необхідно, оскільки пристрій може працювати в умовах нестабільного живлення, слабкого Wi-Fi-сигналу або періодичних збоїв у роботі датчиків.

Система виконує такі дії у разі виникнення помилок: повторне підключення до Wi-Fi при втраті зв'язку, повторну передачу телеметричних даних у разі помилки доставки, повторне зчитування показників у разі некоректних значень, ігнорування явно помилкових вимірювань, фіксацію

помилки у службовому повідомленні, перезапуск окремих програмних модулів у разі потреби.

Наприклад, якщо Wi-Fi-з'єднання відсутнє, мікроконтролер виконує кілька спроб повторного підключення. Якщо підключення не відновлюється, пристрій може тимчасово пропустити передачу даних або зберегти їх до наступного циклу. Такий підхід дозволяє уникнути повної зупинки роботи системи.

Якщо датчик повертає некоректне значення, наприклад відстань дорівнює нулю або перевищує фізично можливу висоту контейнера, програма повторює вимірювання. Якщо після повторного зчитування значення залишається некоректним, система може передати на сервер статус помилки сенсора.

Оскільки сенсорний вузол може працювати від автономного джерела живлення, важливим етапом є реалізація механізму зменшення енергоспоживання. Для цього в ESP32 може використовуватися режим Deep Sleep.

Принцип роботи енергоощадного режиму полягає в тому, що мікроконтролер не працює постійно, а активується через певні проміжки часу. Після пробудження він виконує вимірювання, передає дані на сервер і знову переходить у режим сну.

Такий цикл роботи має вигляд:

1. Пробудження мікроконтролера.
2. Запуск сенсорів.
3. Зчитування показників.
4. Обробка даних.
5. Передача телеметрії.
6. Перехід у Deep Sleep.
7. Повторне пробудження через заданий інтервал.

Використання такого підходу дозволяє суттєво зменшити споживання енергії та продовжити час автономної роботи сенсорного вузла. Для системи

моніторингу сміттєвих контейнерів немає потреби виконувати вимірювання щосекунди, тому інтервал у кілька хвилин є достатнім для отримання актуальної інформації.

У результаті реалізації вбудованого програмного забезпечення було створено програмну логіку роботи сенсорного вузла на базі мікроконтролера ESP32. Розроблене програмне забезпечення забезпечує ініціалізацію апаратних компонентів, зчитування даних із сенсорів, обчислення рівня заповнення контейнера, формування телеметричного повідомлення та передачу інформації на сервер. Програмна структура побудована за модульним принципом, що спрощує її подальше вдосконалення та обслуговування. Окремо реалізовано модулі збору даних, обробки показників, передачі інформації, контролю помилок та керування енергоспоживанням.

3.3 Реалізація серверного програмного забезпечення

Після розробки вбудованого програмного забезпечення сенсорного вузла було реалізовано серверну частину кіберфізичної системи «Розумні сміттєві контейнери». Серверне програмне забезпечення забезпечує прийом телеметричних даних від фізичних пристроїв, їх перевірку, обробку, збереження у базі даних та передачу до користувацького інтерфейсу.

Серверна частина є центральним елементом програмної архітектури, оскільки забезпечує взаємодію між сенсорними вузлами, базою даних та вебінтерфейсом оператора. Дані, отримані від ESP32, проходять перевірку, зберігаються та використовуються для формування поточного стану контейнерів.

Для реалізації серверної частини використано Node.js, Express.js та PostgreSQL, що забезпечують обробку мережових запитів, створення API та надійне збереження даних.

Загальну архітектуру серверного програмного забезпечення (рис 3.6)

Модульний підхід дозволяє розділити логіку серверної частини на окремі відповідальні компоненти. Наприклад, один модуль відповідає лише за прийом MQTT-повідомлень, інший – за перевірку та обробку телеметрії, третій – за збереження інформації у базі даних.

Такий підхід є зручним для подальшої модернізації системи. У разі необхідності можна замінити спосіб передачі даних, додати нові API-маршрути або розширити модуль сповіщень без повної зміни всієї серверної логіки.
Реалізація модуля прийому телеметричних даних

Для прийому телеметрії від сенсорних вузлів було реалізовано MQTT-сервіс. Протокол MQTT є зручним для IoT-систем, оскільки забезпечує легкий асинхронний обмін повідомленнями між пристроями та сервером.

Сенсорний вузол ESP32 після зчитування та обробки показників формує JSON-повідомлення і публікує його у відповідний MQTT-канал. Сервер підписується на цей канал і отримує всі нові повідомлення, що надходять від пристроїв.

Під час запуску сервер виконує такі дії: підключення до MQTT-брокера, підписку на канал прийому телеметрії, очікування нових повідомлень, прийом JSON-пакетів, передачу отриманих даних до модуля обробки

Телеметричне повідомлення містить основні параметри стану контейнера: ідентифікатор пристрою, ідентифікатор контейнера, рівень заповнення, температуру, вологість, рівень газу або диму, рівень заряду батареї, час вимірювання.

Приклад структури повідомлення:

```
{  
  "device_id": "ESP32_001",  
  "container_id": "K-001",  
  "fill_level": 84,  
  "temperature": 25.4,  
}
```

					КвРКІ.022114.22.02.11 ПЗ	Арк.
						49
Зм.	Арк.	№докум.	Підпис	Дата		

```
"humidity": 61,  
"gas_level": 140,  
"battery": 78,  
"timestamp": "2026-05-10T13:35:00"  
}
```

Використання JSON-формату дозволяє легко передавати структуровані дані між сенсорним вузлом і сервером. Сервер може швидко розібрати повідомлення, перевірити значення та зберегти їх у базі даних.

Після отримання телеметричного пакета сервер виконує його первинну обробку. На цьому етапі перевіряється коректність структури повідомлення, наявність необхідних полів і допустимість отриманих значень.

Модуль обробки телеметрії виконує такі функції: перевірку цілісності пакета, перевірку наявності обов'язкових параметрів, перевірку типів отриманих значень, перевірку допустимих діапазонів параметрів, визначення поточного статусу контейнера, формування системних подій у разі критичних значень.

Наприклад, рівень заповнення контейнера повинен знаходитися в межах від 0 % до 100 %. Якщо сервер отримує значення, яке виходить за ці межі, таке повідомлення може бути позначене як некоректне. Аналогічно перевіряються температура, вологість, рівень газу та заряд батареї.

Після перевірки даних система визначає стан контейнера. Якщо рівень заповнення не перевищує 50 %, контейнер отримує нормальний статус. Якщо рівень знаходиться в межах від 51 % до 80 %, контейнер позначається як такий, що має підвищене заповнення. Якщо рівень перевищує 80 %, система формує критичний статус. Алгоритм обробки телеметричних даних на сервері (рис. 3.7) працює за таким принципом.

					КвРКІ.022114.22.02.11 ПЗ	Арк.
						50
Зм.	Арк.	№докум.	Підпис	Дата		

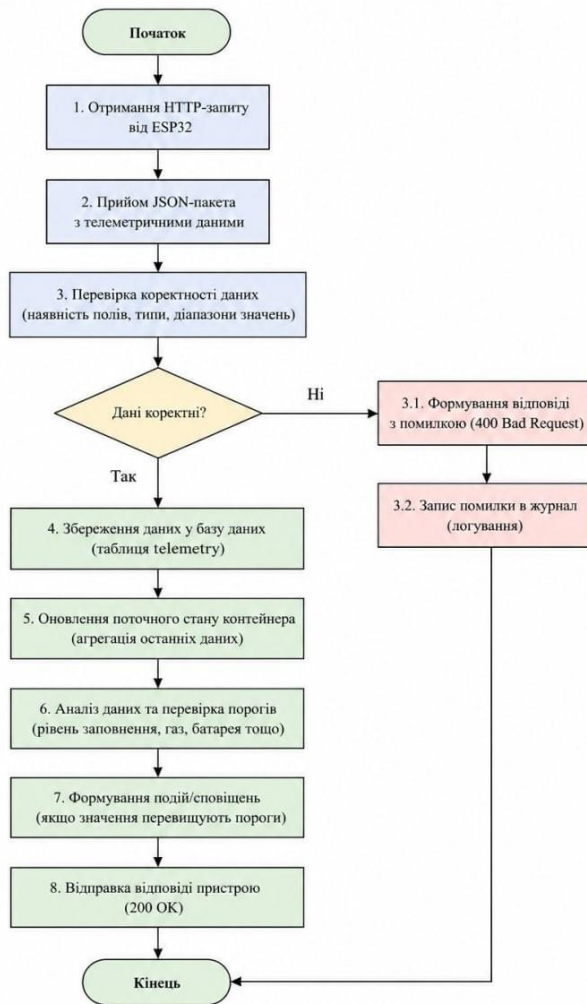


Рисунок 3.7 – Алгоритм обробки телеметричних даних сервера

У разі критичного заповнення сервер автоматично формує системну подію. Наприклад, якщо контейнер заповнений на 90 %, у базі даних створюється запис про критичний стан, а користувацький інтерфейс відображає відповідне сповіщення.

Для централізованого збереження інформації було реалізовано підключення серверного застосунку до бази даних PostgreSQL. База даних використовується для збереження відомостей про контейнери, сенсорні пристрої, телеметричні вимірювання, користувачів і системні події.

Після отримання та перевірки даних сервер виконує такі операції: запис нового вимірювання у таблицю Measurements, оновлення поточного стану контейнера, оновлення інформації про сенсорний пристрій, збереження

У системі можуть бути передбачені ролі адміністратора, оператора та технічного спеціаліста. Адміністратор має доступ до налаштувань системи, оператор переглядає стан контейнерів і події, а технічний спеціаліст отримує інформацію про пристрої, які потребують обслуговування.

Розмежування прав доступу дозволяє підвищити безпеку серверної частини та уникнути несанкціонованої зміни даних.

Для оперативного інформування операторів було реалізовано механізм автоматичного формування системних сповіщень. Сповіщення створюються у випадках, коли певні параметри контейнера перевищують встановлені порогові значення або виникають помилки в роботі пристрою.

Події формуються у таких випадках: критичне заповнення контейнера, втрата зв'язку з пристроєм, низький рівень заряду батареї, підвищений рівень газу або диму, перевищення допустимої температури, некоректні показники сенсорів.

Після формування подія зберігається у базі даних і стає доступною для перегляду у користувацькому інтерфейсі. Для кожної події може зберігатися її тип, опис, час створення, пріоритет і статус обробки.

Наприклад, якщо рівень заповнення контейнера перевищує 80 %, сервер створює подію з високим пріоритетом. Якщо пристрій не передає дані протягом тривалого часу, система може створити подію про втрату зв'язку.

Після реалізації основних програмних модулів було виконано тестовий запуск серверного застосунку. Метою тестування була перевірка працездатності серверної частини, коректності прийому даних, взаємодії з базою даних та роботи API.

У ході тестування перевірялися: стабільність підключення до MQTT-брокера, коректність прийому JSON-повідомлень, перевірка структури телеметричних пакетів, правильність визначення статусу контейнера, запис даних у PostgreSQL, формування системних подій, працездатність REST API-запитів, отримання даних клієнтським інтерфейсом.

Під час тестового запуску сервер успішно приймав телеметричні повідомлення, обробляв їх і зберігав у базі даних. Також було перевірено, що у випадку перевищення критичного рівня заповнення система автоматично формує відповідну подію.

У результаті реалізації серверного програмного забезпечення було створено центральну програмну частину кіберфізичної системи, яка забезпечує прийом, обробку, збереження та передачу даних між сенсорними вузлами і користувацьким інтерфейсом.

Серверна частина реалізована з використанням Node.js, Express.js, MQTT та PostgreSQL. Такий набір технологій дозволяє забезпечити стабільну роботу системи, обробку телеметричних повідомлень, збереження історії вимірювань, формування системних подій і надання даних клієнтським застосункам.

3.4 Реалізація користувацького інтерфейсу

Після реалізації серверного програмного забезпечення було створено користувацький інтерфейс кіберфізичної системи «Розумні сміттєві контейнери». Його основне призначення – забезпечити оператору зручний доступ до інформації про стан контейнерів, перегляд телеметричних показників, системних подій та результатів моніторингу в режимі реального часу.

Інтерфейс використовується для демонстрації основних функцій системи: перегляду контейнерів, аналізу їхнього стану, імітації надходження телеметричних даних та відображення системних подій.

Інтерфейс побудовано у форматі інформаційної панелі, де основні елементи розміщені на одному екрані. Це дозволяє оператору швидко оцінити загальний стан системи без переходу між великою кількістю сторінок.

До складу інтерфейсу входять такі блоки: верхня панель системи, головна інформаційна панель, блок статистичних показників, список контейнерів, панель детальної інформації, карта розташування контейнерів, журнал системних подій, кнопка генерації тестових даних.

температура, вологість, рівень газу, заряд батареї та статус з'єднання. Рівень заповнення подається у вигляді числового значення та графічної шкали.

Якщо контейнер переходить у критичний стан, індикатор змінює колір на червоний, що дозволяє оператору швидко помітити проблему.

Реалізація карти контейнерів

Для наочного відображення розташування контейнерів реалізовано карту–схему. Контейнери позначені маркерами, колір яких відповідає їхньому поточному стану.

До журналу можуть потрапляти такі події: критичне заповнення контейнера, підвищений рівень газу або диму, низький заряд батареї, втрата зв'язку із сенсорним вузлом.

Кожна подія містить час виникнення, ідентифікатор контейнера, короткий опис і рівень пріоритету.

Імітація отримання телеметричних даних

Для демонстрації роботи системи в інтерфейсі реалізовано кнопку «Генерувати дані»(рис. 3.10). Після її натискання система імітує надходження нових показників від сенсорних вузлів.

Такий механізм дозволяє показати роботу системи навіть без фізичного підключення ESP32. Це зручно для демонстрації на захисті, оскільки можна наочно показати зміну показників і автоматичне формування подій.

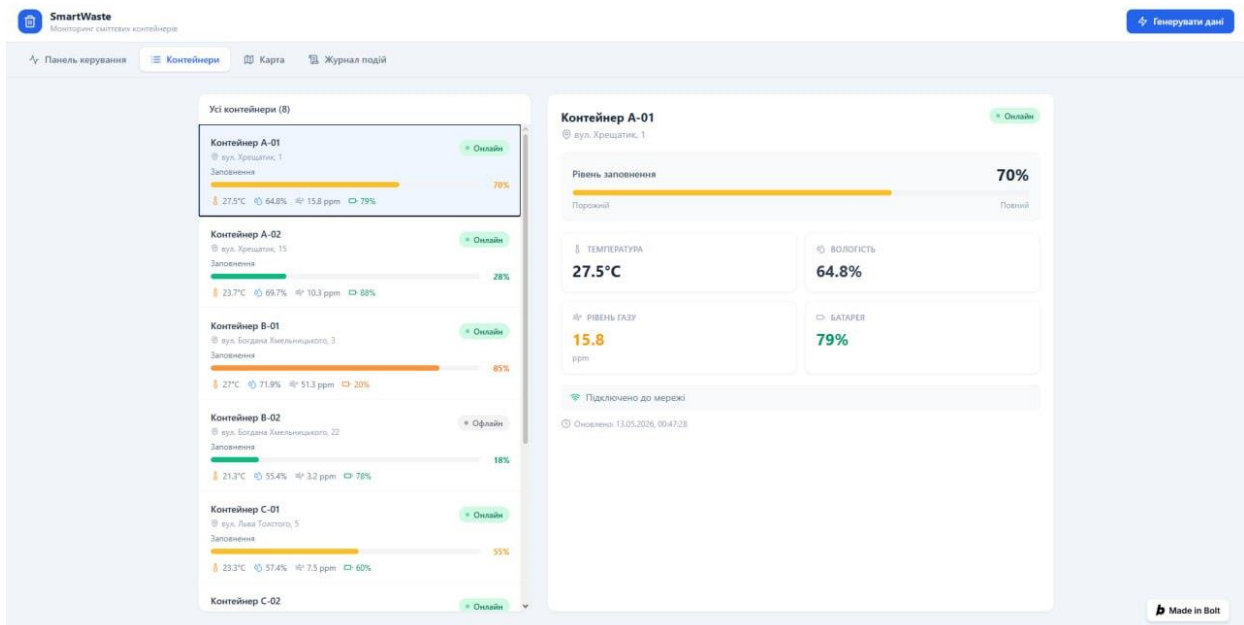


Рисунок 3.10 – Демонстраційний вебпрототип системи моніторингу контейнерів

У результаті реалізації користувацького інтерфейсу було створено демонстраційний вебпрототип системи моніторингу розумних сміттєвих контейнерів. Інтерфейс забезпечує перегляд стану контейнерів, рівня заповнення, температури, вологості, рівня газу, заряду батареї та статусу з'єднання.

Також реалізовано карту розташування контейнерів, список об'єктів, панель детального моніторингу та журнал системних подій. Механізм генерації тестових даних дозволяє продемонструвати роботу системи в режимі реального часу та підтверджує працездатність розробленого програмного прототипу.

3.5 Тестування кіберфізичної системи

Після реалізації апаратної частини, вбудованого програмного забезпечення, серверної частини та користувацького інтерфейсу було виконано тестування кіберфізичної системи «Розумні сміттєві контейнери». Метою тестування є перевірка працездатності основних компонентів системи,

коректності обробки телеметричних даних та правильності відображення інформації у вебінтерфейсі.

Тестування проводилося поетапно, оскільки система складається з кількох взаємопов'язаних частин: сенсорного вузла, серверного програмного забезпечення, бази даних та користувацького інтерфейсу.

Основними завданнями тестування були: перевірка коректності зчитування сенсорних даних, перевірка розрахунку рівня заповнення контейнера, перевірка формування телеметричного повідомлення, перевірка передачі даних на сервер, перевірка обробки критичних значень, перевірка роботи користувацького інтерфейсу, перевірка генерації системних подій.

На першому етапі було перевірено роботу сенсорного вузла на базі мікроконтролера ESP32. Основна увага приділялася правильності зчитування показників із підключених датчиків.

Перевірялися такі компоненти: ультразвуковий сенсор HC-SR04, датчик температури та вологості DHT22, газовий сенсор MQ-2, модуль Wi-Fi-з'єднання, система живлення.

Під час тестування ультразвукового сенсора перевірялося визначення відстані до умовної поверхні відходів. На основі отриманої відстані програмне забезпечення обчислювало рівень заповнення контейнера за формулою:

Також було перевірено, що система правильно класифікує стан контейнера:

- до 50 % – нормальний стан;
- від 51 % до 80 % – попереджувальний стан;
- понад 80 % – критичний стан.

Результати тестування показали, що логіка визначення рівня заповнення працює коректно, а зміна вхідних даних призводить до відповідної зміни статусу контейнера.

На наступному етапі перевірялася передача телеметричних даних від сенсорного вузла до серверної частини. Телеметричне повідомлення формується у форматі JSON і містить основні параметри стану контейнера.

До складу повідомлення входять: ідентифікатор пристрою, ідентифікатор контейнера, рівень заповнення, температура, вологість, рівень газу, заряд батареї, поточний статус контейнера.

Приклад тестового повідомлення:

```
{  
  "device_id": "ESP32_001",  
  "container_id": "K-001",  
  "fill_level": 84,  
  "temperature": 25.4,  
  "humidity": 61,  
  "gas_level": 140,  
  "battery": 78,  
  "status": "critical"  
}
```

Під час тестування було перевірено, що сервер приймає повідомлення, перевіряє його структуру, обробляє отримані значення та передає їх для подальшого збереження й відображення.

Серверна частина перевірялася на здатність приймати, обробляти та зберігати телеметричні дані. Основна увага приділялася правильності обробки JSON-повідомлень і формуванню системних подій.

Під час тестування перевірялися такі функції: прийом телеметричних даних, перевірка коректності отриманого пакета, визначення статусу контейнера, запис даних у базу, формування подій при критичних значеннях, передача даних до користувацького інтерфейсу.

Якщо рівень заповнення контейнера перевищував 80 %, система автоматично формувала подію про критичне заповнення. Якщо рівень газу перевищував допустиме значення, створювалося повідомлення про потенційно небезпечний стан контейнера. У разі втрати зв'язку з пристроєм система відображала відповідний статус.

У процесі тестування перевірялося: відображення списку контейнерів, відображення рівня заповнення, відображення температури, вологості та рівня газу, відображення заряду батареї, зміна статусу контейнера, робота карти розташування контейнерів, формування записів у журналі подій, робота кнопки генерації тестових даних.

Для імітації надходження телеметричних даних було використано кнопку «Генерувати дані». Після її натискання значення показників змінювалися, а система автоматично оновлювала статуси контейнерів та журнал подій. Проведено тестування вебінтерфейсу системи моніторингу(рис. 3.11)

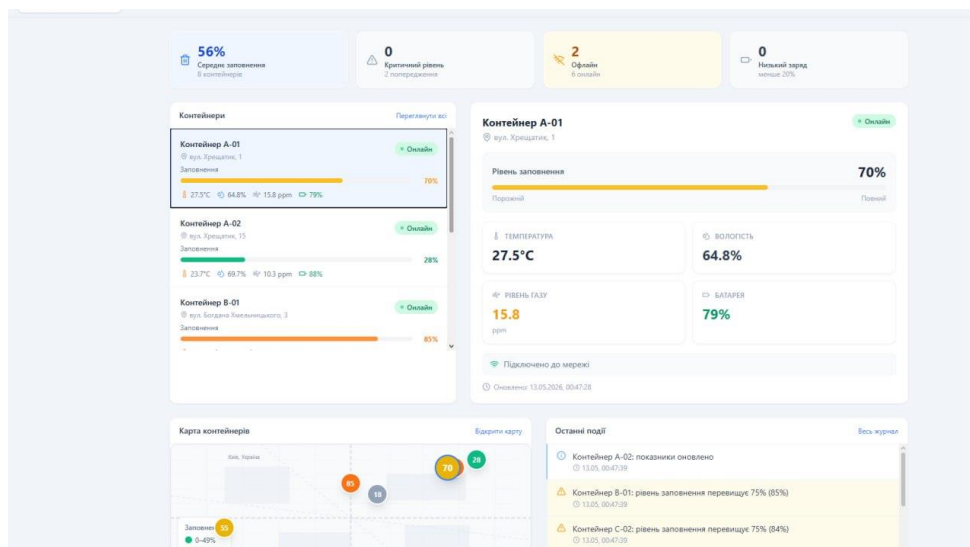


Рисунок 3.11 – Тестування вебінтерфейсу системи моніторингу

Результати тестування показали, що розроблена система виконує основні функції, передбачені технічним завданням. Сенсорний вузол забезпечує отримання показників, програмна частина виконує їх обробку, сервер приймає

телеметричні дані, а користувацький інтерфейс відображає стан контейнерів у зручному вигляді.

Проведене тестування підтвердило працездатність розробленого програмно–технічного засобу. Система коректно реагує на зміну вхідних параметрів, визначає стан контейнера, формує події та відображає результати у вебінтерфейсі.

У ході роботи було виконано реалізацію основних компонентів кіберфізичної системи «Розумні сміттєві контейнери для моніторингу збору відходів у режимі реального часу».

Було реалізовано апаратний прототип сенсорного вузла, до складу якого входять мікроконтролер ESP32, ультразвуковий сенсор HC–SR04, датчик температури та вологості DHT22, газовий сенсор MQ–2, система живлення та захисний корпус. Запропонована апаратна реалізація дозволяє отримувати основні параметри стану контейнера та забезпечує можливість подальшої передачі даних на сервер.

3.6. Висновки до третього розділу

У третьому розділі кваліфікаційної роботи було виконано реалізацію програмно–технічного засобу кіберфізичної системи «Розумні сміттєві контейнери для моніторингу збору відходів у режимі реального часу» Апаратна частина. На основі проєктних рішень, розроблених у другому розділі, було реалізовано основні складові системи: апаратний прототип сенсорного вузла, вбудоване програмне забезпечення, серверну частину, користувацький вебінтерфейс та проведено тестування працездатності системи.

У процесі реалізації апаратної частини було сформовано прототип сенсорного вузла на базі мікроконтролерного модуля ESP32. До складу вузла входять ультразвуковий сенсор HC–SR04 для визначення рівня заповнення контейнера, датчик температури та вологості DHT22, газовий сенсор MQ–2,

					КвРКІ.022114.22.02.11 ПЗ	Арк. 61
Зм.	Арк.	№докум.	Підпис	Дата		

система живлення та захисний корпус. Таке апаратне рішення дозволяє отримувати основні параметри стану контейнера та передавати їх для подальшої обробки.

Було реалізовано вбудоване програмне забезпечення сенсорного вузла, яке забезпечує ініціалізацію апаратних компонентів, зчитування показників із сенсорів, обчислення рівня заповнення контейнера, визначення його поточного статусу, формування телеметричного повідомлення та передачу даних на сервер. Для підвищення стабільності роботи передбачено обробку можливих помилок, повторне зчитування некоректних значень та використання енергоощадного режиму.

У межах реалізації серверної частини було створено програмну логіку прийому, перевірки, обробки та збереження телеметричних даних. Серверна частина забезпечує обробку отриманих показників, визначення критичних станів, формування системних подій та передачу актуальної інформації до клієнтського інтерфейсу через REST API.

Окремо було реалізовано користувацький вебінтерфейс системи моніторингу. Інтерфейс дозволяє переглядати список контейнерів, рівень їх заповнення, температуру, вологість, рівень газу, заряд батареї, статус з'єднання та журнал подій. Також реалізовано карту розташування контейнерів і панель детального моніторингу вибраного об'єкта. Для демонстрації роботи системи було створено вебпрототип, доступний за посиланням <https://react-uut4.bolt.host>.

Проведене тестування підтвердило працездатність основних функцій системи. Було перевірено зчитування сенсорних даних, розрахунок рівня заповнення контейнера, формування телеметричних повідомлень, обробку критичних значень, генерацію системних подій та коректне відображення інформації у вебінтерфейсі. Результати тестування показали, що система правильно реагує на зміну вхідних параметрів і дозволяє оператору швидко оцінювати стан контейнерів.

Таким чином, у третьому розділі було реалізовано демонстраційний програмно–технічний прототип кіберфізичної системи, який підтверджує можливість практичного застосування запропонованого рішення. Розроблена система забезпечує автоматизований моніторинг стану сміттєвих контейнерів, своєчасне виявлення критичних ситуацій та наочне представлення даних для користувача в режимі реального часу.

					КвРКІ.022114.22.02.11 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		63

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне науково–практичне завдання, пов’язане з проєктуванням та програмно–апаратною реалізацією кіберфізичної системи «Розумні сміттєві контейнери» для моніторингу збору відходів у режимі реального часу.

У процесі виконання роботи було проведено аналіз предметної області, досліджено особливості функціонування сучасних систем моніторингу збору твердих побутових відходів, проаналізовано існуючі програмно–апаратні рішення у сфері Інтернету речей, кіберфізичних систем та автоматизації міської інфраструктури. На основі проведеного аналізу визначено функціональні та технічні вимоги до розроблюваної системи.

У ході роботи було розроблено архітектуру кіберфізичної системи, яка включає сенсорний вузол, вбудоване програмне забезпечення, серверну інфраструктуру, базу даних та користувацький вебінтерфейс. Обґрунтовано вибір апаратної платформи на базі Espressif Systems ESP32, сенсорного обладнання для контролю рівня заповнення контейнера та параметрів навколишнього середовища, а також засобів бездротової передачі даних.

У результаті практичної реалізації було створено працездатний програмно–апаратний прототип кіберфізичної системи, що забезпечує автоматизований збір телеметричних даних, локальну обробку інформації, бездротову передачу даних на сервер, їх централізоване збереження, аналітичну обробку та візуалізацію в режимі реального часу.

Було реалізовано вбудоване програмне забезпечення сенсорного вузла, серверну частину системи, механізми взаємодії з базою даних, API для обміну інформацією, а також користувацький вебінтерфейс для моніторингу стану контейнерів, перегляду статистичних даних та отримання системних сповіщень.

Під час експериментального тестування підтверджено працездатність усіх основних функціональних модулів системи. Отримані результати показали

					КьРКІ.022114.22.02.11 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		64

достатню точність вимірювання рівня заповнення контейнерів, стабільність бездротової передачі телеметричних даних, коректність роботи серверного програмного забезпечення та ефективність візуалізації інформації у вебінтерфейсі.

Результати проведених досліджень і експериментів підтвердили відповідність розробленої системи поставленим технічним вимогам та можливість її практичного використання у сфері житлово–комунального господарства, муніципального управління, системах «розумного міста» та інших IoT–рішеннях.

Практичне значення одержаних результатів полягає у створенні масштабованого програмно–апаратного рішення, яке може бути адаптоване для впровадження в реальні системи моніторингу міської інфраструктури та подальшого розвитку кіберфізичних комплексів автоматизованого контролю.

					КвРКІ.022114.22.02.11 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		65

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Andrew S. TanenbaumDavid J. Wetherall Computer Networks (5th Edition), Prentice Hall, 2011. 960 p.
- 2.
3. Sommerville I. Software Engineering. 10th ed. Boston : Pearson, 2016. 816 p.
4. Pressman R. S. Software Engineering: A Practitioner's Approach. 8th ed. New York : McGraw-Hill Education, 2019. 976 p.
5. Silberschatz A., Korth H., Sudarshan S. Database System Concepts. 7th ed. New York : McGraw-Hill, 2019. 1376 p.
6. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to Algorithms. 3rd ed. Cambridge : MIT Press, 2020. 1312 p.
7. Marwedel P. Embedded System Design. Dordrecht : Springer, 2018. 425 p.
8. Kurose J. F., Ross K. W. Computer Networking: A Top-Down Approach. 7th ed. Boston : Pearson, 2017. 864 p.
9. Stallings W. Data and Computer Communications. 10th ed. Boston : Pearson, 2015. 912 p.
10. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston : Addison-Wesley, 1994. 395 p.
11. Dargie W., Poellabauer C. Fundamentals of Wireless Sensor Networks: Theory and Practice. Chichester : Wiley, 2017. 311 p.
12. Krishnamachari B. Networking Wireless Sensors. Cambridge : Cambridge University Press, 2018. 216 p.
13. Box G. E. P., Hunter J. S., Hunter W. G. Statistics for Experimenters. 2nd ed. Hoboken : Wiley, 2018. 672 p.
14. Smart City Technologies and Applications / ed. by H. Song, R. Srinivasan, T. Sookoor. Hoboken : Wiley, 2020. 520 p.

					КвРКІ.022114.22.02.11 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		66

15. Lee E. A. Cyber Physical Systems: Design Challenges. 11th IEEE *International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*. 2008. P. 363–369.

16. Rajkumar R., Lee I., Sha L., Stankovic J. Cyber-Physical Systems: The Next Computing Revolution. *Design Automation Conference*. 2010. P. 731–736.

17. ISO 25010:2011 Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE). Geneva : ISO, 2019. 34 p.

18. IEC 61508 Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems. Geneva : IEC, 2010. 412 p.

19. Smart Waste Management Using IoT. *International Journal of Engineering Research*. 2021. Vol. 10, № 4. P. 45–51.

20. IoT Based Smart Garbage Monitoring System. *IEEE Conference Proceedings*. 2022. P. 112–118.

21. Yick J., Mukherjee B., Ghosal D. *Wireless Sensor Network Survey*. *Computer Networks*. 2008. Vol. 52, № 12. P. 2292–2330.

22. Espressif Systems. ESP32 Technical Reference Manual. URL: <https://docs.espressif.com> (дата звернення: 12.04.2026).

23. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs/> (дата звернення: 12.04.2026).

24. Arduino Documentation. URL: <https://docs.arduino.cc> (дата звернення: 12.04.2026).

25. MQTT Version 3.1.1 Specification. URL: <https://mqtt.org> (дата звернення: 12.04.2026).

26. Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content RFC 7231. URL: <https://datatracker.ietf.org/doc/html/rfc7231> (дата звернення: 12.04.2026).

27. The JavaScript Guide. MDN Web Docs. URL: <https://developer.mozilla.org> (дата звернення: 24.04.2026).

28. React Documentation. URL: <https://react.dev> (дата звернення: 24.04.2026).

					КвРКІ.022114.22.02.11 ПЗ	Арк. 67
Зм.	Арк.	№докум.	Підпис	Дата		

29. Bootstrap Documentation. URL: <https://getbootstrap.com/docs> (дата звернення: 24.04.2026).
30. Node.js Documentation. URL: <https://nodejs.org/docs> (дата звернення: 24.04.2026).
31. OpenStreetMap API Documentation. URL: <https://wiki.openstreetmap.org> (дата звернення: 24.04.2026).
32. Docker Documentation. URL: <https://docs.docker.com> (дата звернення: 24.04.2026).
33. AWS IoT Core Documentation. URL: <https://docs.aws.amazon.com/iot> (дата звернення: 24.05.2026).
34. Microsoft Azure IoT Documentation. URL: <https://learn.microsoft.com/azure/iot> (дата звернення: 06.05.2026).
35. Google Cloud Documentation. URL: <https://cloud.google.com> (дата звернення: 08.05.2026).
36. Cisco Networking Academy. IoT Fundamentals. URL: <https://www.netacad.com> (дата звернення: 08.05.2026).
37. HiveMQ MQTT Essentials. URL: <https://www.hivemq.com> (дата звернення: 08.05.2026).
38. LoRaWAN Specification v1.0.4. URL: <https://lora-alliance.org> (дата звернення: 10.05.2026).
39. OWASP Foundation. Web Application Security Testing Guide. URL: <https://owasp.org> (дата звернення: 10.05.2026).
40. Enevo Smart Waste Solutions. URL: <https://www.enevo.com> (дата звернення: 11.05.2026).
41. Bigbelly Smart Waste & Recycling System. URL: <https://bigbelly.com> (дата звернення: 13.05.2026).
42. Ecube Labs Smart Waste Management Solutions. URL: <https://www.ecubelabs.com> (дата звернення: 13.05.2026).

					КвРКІ.022114.22.02.11 ПЗ	Арк. 68
Зм.	Арк.	№докум.	Підпис	Дата		

43. ThingsBoard IoT Platform. URL: <https://thingsboard.io> (дата звернення: 13.05.2026).

44. Firebase Documentation. URL: <https://firebase.google.com/docs> (дата звернення: 16.05.2026).

45. Grafana Documentation. URL: <https://grafana.com/docs> (дата звернення: 16.05.2026).

46. Prometheus Monitoring System Documentation. URL: <https://prometheus.io> (дата звернення: 16.05.2026).

47. PlatformIO Documentation. URL: <https://platformio.org> (дата звернення: 20.05.2026).

48. Visual Studio Code Documentation. URL: <https://code.visualstudio.com/docs> (дата звернення: 20.05.2026).

49. GitHub Documentation. URL: <https://docs.github.com> (дата звернення: 20.05.2026).

50. JSON Data Interchange Syntax RFC 8259. URL: <https://datatracker.ietf.org/doc/html/rfc8259> (дата звернення: 20.05.2026).

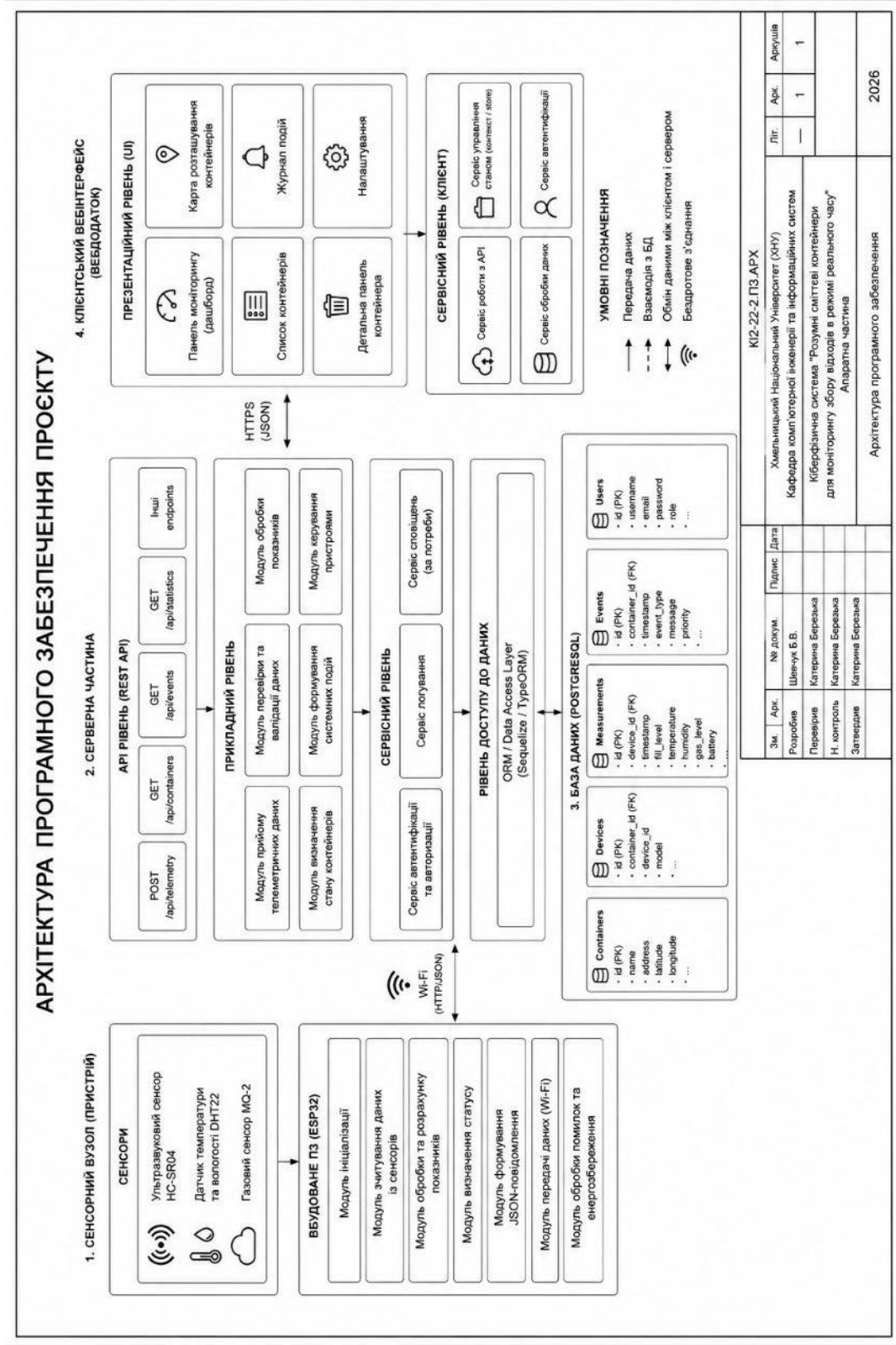
51. Linux Foundation. Internet of Things Resources. URL: <https://www.linuxfoundation.org> (дата звернення: 20.05.2026).

					КвРКІ.022114.22.02.11 ПЗ	Арк. 69
Зм.	Арк.	№докум.	Підпис	Дата		

ДОДАТОК А

(обов'язковий)

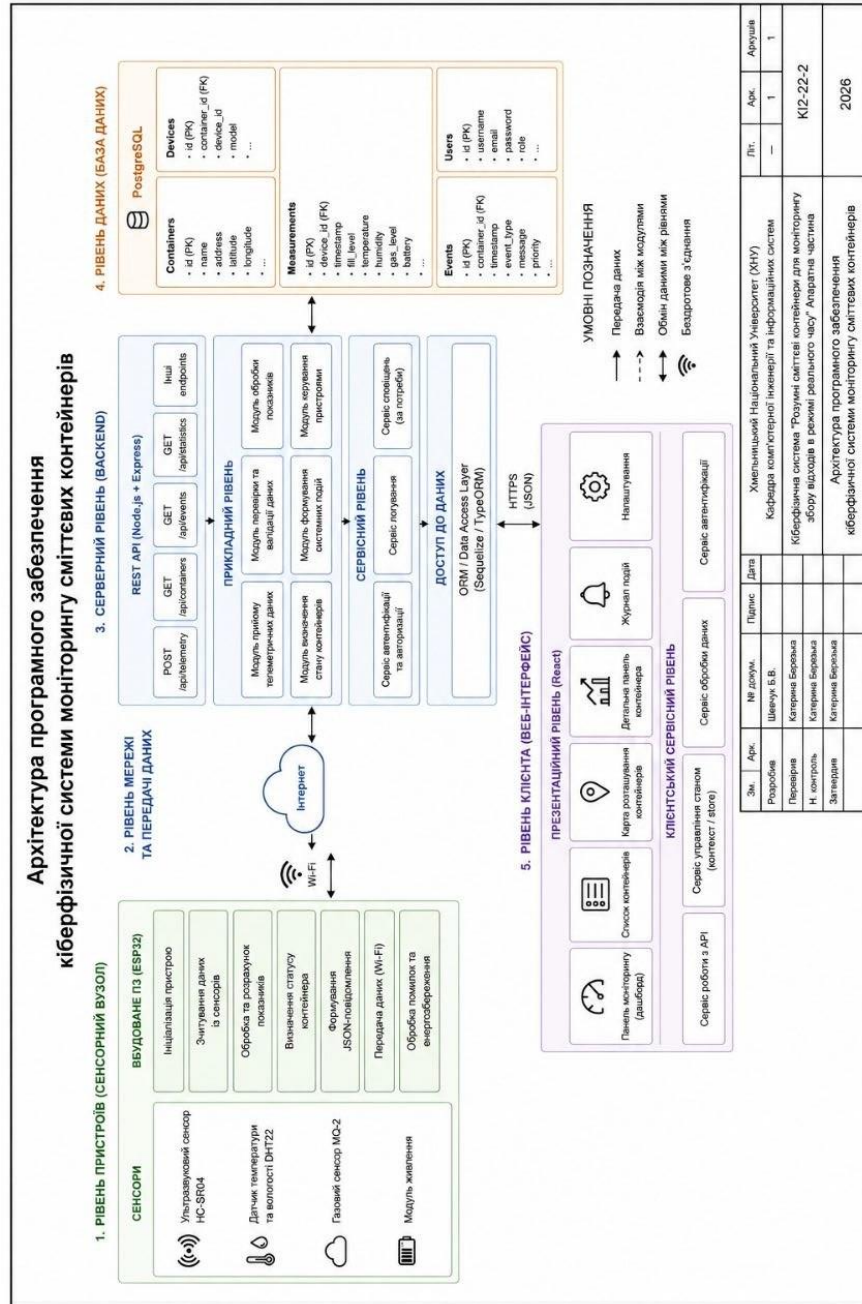
«Архітектура ПЗ проекту»



ДОДАТОК Б

(обов'язковий)

«Архітектура ПЗ для кіберфізичної системи»



ДОДАТОК В (обов'язковий)

«Апаратне забезпечення проекту»

АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ

Кіберфізична система моніторингу заповнення смітєвих контейнерів

1. СЕНСОРНИЙ ВУЗОЛ (ВСТАНОВЛЮЄТЬСЯ НА КОНТЕЙНЕРІ)

- Ультразвуковий сенсор HC-SR04**
Вимірювання відстані до рівня виходів
- Датчик температури та вологості DHT22**
Вимірювання температури та вологості всередині контейнера
- Газовий сенсор MQ-2**
Виявлення газу та неприємних запахів
- Акумулятор 18650 (3,7 В)**
Живлення пристрою

2. КЕРУЮЧИЙ МОДУЛЬ

- Мікроконтролер ESP32**
Обробка даних з сенсорів та передача інформації
- OLED дисплей 0,96" (SSD1306)**
Відображення поточного стану пристрою
- Wi-Fi модуль**
Передача даних на сервер

3. ЖИВЛЕННЯ

- Модуль зарядження Зарядження Li-Ion акумулятора TP4056**
- Параметричний DC-DC перетворювач (5В)**
Стабілізоване живлення для компонентів

4. КОРПУС ТА ВСТАНОВЛЕННЯ

- Вологозахисний корпус IP65**
Кріплення на кришку контейнера

5. ПЕРЕЛІК ОСНОВНИХ КОМПОНЕНТІВ

№	Компонент	Кількість	Призначення
1	ESP32 DevKit	1 шт.	Керування та передача даних
2	Ультразвуковий сенсор HC-SR04	1 шт.	Вимірювання рівня заповнення
3	Датчик DHT22	1 шт.	Вимірювання температури та вологості
4	Газовий сенсор MQ-2	1 шт.	Виявлення газу та запахів
5	OLED дисплей 0,96" (SSD1306)	1 шт.	Відображення інформації
6	Акумулятор 18650 (3,7 В)	2 шт.	Живлення пристрою
7	Модуль зарядження TP4056	1 шт.	Зарядження акумулятора
8	DC-DC перетворювач (5В)	1 шт.	Стабілізоване живлення 5В
9	Монтажний кронштейн	1 шт.	Кріплення пристрою на контейнер
10	Корпус пластиковий IP65	1 шт.	Захист електроніки від зовнішніх впливів

6. СТРУКТУРНА СХЕМА АПАРАТНОЇ ЧАСТИНИ

7. ОСНОВНІ ХАРАКТЕРИСТИКИ

- Напряг живлення: 3,7 В (акумулятор), вихід 5 В
- Споживання в режимі роботи: ~120–150 мА
- Споживання в режимі сну: ~10–30 мА
- Діапазон вимірювання відстані: 2 – 400 см (HC-SR04)
- Діапазон вимірювання температури: -40 ... +80 °C (DHT22)
- Діапазон вологості: 0 – 100 % RH (DHT22)
- Діапазон виявлення газу: 200 – 10000 ppm (MQ-2)
- Інтерфейс зв'язу: Wi-Fi (802.11 b/g/n)
- Ступінь захисту корпусу: IP65
- Робоча температура: -20 ... +60 °C
- Час автономної роботи: до 7 дб (залежить від режиму)

8. ПЕРЕЛІК ОСНОВНИХ КОМПОНЕНТІВ

Зм.	Авк.	№ докум.	Підпис	Дата
Розробив		Шевчук Б.В.		
Перевірив		Катерина Бурдава		
Н. контроль		Катерина Бурдава		
Затвердив		Катерина Бурдава		

9. ІНФОРМАЦІЯ ПРО ПРОЄКТ

Літ.	Авк.	Апробув.
—	1	1

Хмельницький Національний Університет (ХНУ)
Кафедра комп'ютерної інженерії та інформаційних систем

Кіберфізична система "Роумінг смітєвих контейнерів для моніторингу заповнення в режимі реального часу" Апаратна частина

КІ2-22-2

Апаратне забезпечення проєкту

2026

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: Богдан ШЕВЧУК

Співавтор:

Назва: Кіберфізична система "Розумні сміттєві контейнери" для моніторингу збору відходів у режимі реального часу. Апаратна частина

Експерт: Катерина Березька

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 5.76%

Коефіцієнт подібності 2: 0.5%

Мікропробіли: 3

Заміна букв: 0

Інтервали: 0

Блі знаки: 0

Дата створення звіту: 2026-06-12 11:19:23.0

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 укріття 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:

2026-06-12

Дата

Доцент Андрій Нічепорук

експерт

Anti-Plagiarism (<http://ap.km.ua>) v-15.701**Максимальне співпадіння з одним документом 1.0%****Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 11%**

ID: 274822 Назва: БКР Кіберфізична система "Розумні сміттєві контейнери" для моніторингу збору відходів у режимі реального часу. Апаратна частина Додано в БД: 2026-06-11 Автора: Богдан ШЕВЧУК Керівники: Катерина Березька Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	83547	628	3451 (4%)	57 (9%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Шевчук Богдан Володимирович

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

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 72

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є створення програмно – технічного засобу для автоматизованого контролю рівня заповнення контейнерів, передачі телеметричних даних на сервер та їх відображення в користувацькому інтерфейсі.
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: В першому розділі кваліфікаційної роботи проведено дослідження предметної області (проаналізовано види сучасних підходів до побудови інтелектуальних систем управління об'єктами міської інфраструктури). В другому розділі кваліфікаційної роботи було виконано проектування програмно – технічного засобу для реалізації кіберфізичної системи «Розумні сміттєві контейнери для моніторингу збору відходів в режимі реального часу», було розроблено багаторівневу архітектуру системи, що включає сенсорний, обчислювальний, комунікаційний, серверний рівні та рівень представлення даних. В третьому розділі кваліфікаційної роботи виконано реалізацію програмно – технічного засобу кіберфізичної системи, на основі проектних рішень у другому розділі було реалізовано основні складові системи: апаратний прототип сенсорного вузла, вбудоване програмне забезпечення, серверну частину, користувацький вебінтерфейс та проведено тестування працездатності системи. Було сформовано прототип сенсорного вузла на базі мікроконтролерного модуля ESP32. До його складу входять

ультра звуковий сенсор HV- SR04, датчик температури та вологості DHT22, газовий сенсор MQ – 2, система живлення та захисний корпус. Було реалізовано вбудоване програмне забезпечення сенсорного вузла, яке забезпечує ініціалізацію апаратних компонентів, зчитування показників із сенсорів, обчислення рівня заповнення контейнера, визначення його поточного статусу, формування телеметричного повідомлення та передачу даних на сервер. У межах серверної частини було реалізовано та створено програмну логіку прийому, перевірки, обробки та збереження телеметричних даних.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатня увага при створенні вебінтерфейсу, та дослідженню теми.

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

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

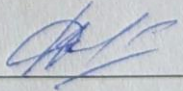
8. Інші зауваження: _____

9. Оцінка дипломної роботи: задовільна /65/E

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) К.Т.Н.,

доцент каф. АКІТ:Р Корецька Людмила
Олександрівна

"01" 06 2026 р.

 (підпис)

Зав. кафедри КПС
д-р. філософії Ользі ПАВЛОВІЙ

Богдан ШЕВЧУК

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 оповіщений (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 Кіберфізична система «Розумні сміттєві контейнери» для моніторингу збору відходів в режимі реального часу. Апаратна частина
 Автор Богдан ШЕВЧУК
 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

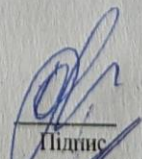
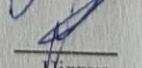
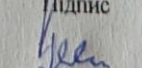
Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 5,76%; та системою Anti-Plagiarism складає 0,5%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

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