

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та кібербезпеки

(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА  
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

ВБУДОВАНА СИСТЕМА КЕРУВАННЯ ПОРТАТИВНИМИ  
ЕЛЕМЕНТАМИ ІНТЕРНЕТУ РЕЧЕЙ

BUILT-IN CONTROL SYSTEM FOR INTERNET OF THINGS  
PORTABLE ELEMENTS

спеціальність 123 Комп'ютерна інженерія  
(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія  
(назва освітньої програми)

Виконав: здобувач вищої освіти  
групи КІ-42  
Бельський Богдан Олександрович

(підпис)

Керівник:  
к.т.н., доцент  
Костючко Сергій Миколайович

(підпис)

Кваліфікаційну роботу  
допущено до захисту  
«    »      червня      2024 р.  
Гарант освітньої програми:  
к.т.н., доцент  
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та кібербезпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

проф. Н.Черняшук

« 10 » 01 2024 р.

ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

*Бельському Богдану Олександровичу*

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Вбудована система керування портативними елементами інтернету речей

Керівник роботи к.т.н., доцент Костючко Сергій Миколайович

затвержені наказом закладу вищої освіти від «30» грудня 2023 року № 459/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 11.06.2024р.

3. Вихідні дані до роботи Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області та різні інтернет-ресурси технічного спрямування

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

Вступ

Аналіз сучасних елементів та систем інтернету речей. Огляд літератури із досліджуваної проблеми

Теоретичні основи роботи з системами та елементами інтернету речей

Розробка системи керування портативними елементами інтернету речей

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Приклад роботи системи інтернету речей

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз проблеми за темою роботи та постановка завдань дослідження</i>	<i>Костючко С.М., доцент</i>		
<i>Теоретичне дослідження та практична реалізація</i>	<i>Костючко С.М., доцент</i>		
<i>Практична реалізація об'єкта проектування</i>	<i>Костючко С.М., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>	_____ %		
<i>Академічна доброчесність</i>	<i>Міскевич О.І., асистент</i>		

7. Дата видачі завдання 10.01.2024 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Розділ 1. Теоретичні основи інтернету речей. Огляд літератури із досліджуваної проблеми</i>	до 15.02.2024 р.	Виконано
2.	<i>Розділ 2. Аналіз сучасних елементів інтернету речей</i>	до 15.03.2024 р.	Виконано
3.	<i>Розділ 3. Реалізація та тестування вбудованої системи керування</i>	до 04.05.2024 р.	Виконано
4.	<i>Висновки та пропозиції</i>	до 07.05.2025 р.	Виконано
5.	<i>Формування списку використаних джерел</i>	до 10.05.2024 р.	Виконано
6.	<i>Формування додатків</i>	до 15.05.2024 р.	Виконано
7.	<i>Оформлення ілюстративного матеріалу</i>	до 20.05.2024 р.	Виконано
8.	<i>Нормоконтроль</i>	до 01.06.2024 р.	Виконано
9.	<i>Інструментальна перевірка на академічний плагіат</i>	до 04.06.2024 р.	Виконано
10.	<i>Представлення кваліфікаційної роботи бакалавра до захисту</i>	до 11.06.2024 р.	Виконано

Здобувач вищої освіти

\_\_\_\_\_ (підпис)

Бельський Б.О.

\_\_\_\_\_ (прізвище, ініціали)

Керівник кваліфікаційної роботи

\_\_\_\_\_ (підпис)

Костючко С.М.

\_\_\_\_\_ (прізвище, ініціали)

## АНОТАЦІЯ

Бельський Б.О. Вбудована система керування портативними елементами інтернету речей. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2024. 43 с.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел.

Перший розділ присвячено огляду предметної області Інтернету речей (IoT), де розглядаються основні поняття та концепції, пов'язані з цією технологією. IoT охоплює мережу фізичних об'єктів, які мають вбудовані датчики, програмне забезпечення та інші технології для підключення та обміну даними з іншими пристроями та системами через Інтернет. У цьому розділі докладно аналізується архітектура IoT та його основні компоненти. Також розглядаються можливі сфери застосування IoT.

В другому розділі розглянуто класифікацію IoT пристроїв та аналізу можливих проблем і викликів, які виникають під час їх розробки. У цьому розділі детально розглядаються різні типи IoT пристроїв. Обговорюються проблеми безпеки та конфіденційності, відсутність єдиних стандартів, а також складнощі, пов'язані з обмеженими ресурсами пристроїв, такими як обчислювальна потужність і енергоефективність.

Третій розділ присвячено опису процесу розробки системи керування портативними елементами та визначенню технологічного стеку під час розробки. Також, у цьому розділі проведено та описано тестування даного програмного забезпечення.

Ключові слова: інтернет речей, Internet of Things, портативні девайси, девайси, система керування.

## ANNOTATION

Belskyi B.O. Built-in control system for internet of things portable elements. Manuscript.

Qualifying work of a bachelor of EP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2024. 43 c.

Qualification work consists of an introduction, three sections, conclusions, a references.

The first section is dedicated to an overview of the subject area of the Internet of Things (IoT), where the main concepts and concepts related to this technology are considered. IoT encompasses a network of physical objects that have built-in sensors, software, and other technologies to connect and share data with other devices and systems over the Internet. This chapter analyzes the IoT architecture and its main components in detail. Possible areas of application of IoT are also considered in this section.

In the second section, the classification of devices and the analysis of possible problems and challenges that arise during their development was carried out. This section takes a detailed look at the different types of IoT devices. Security and privacy issues, the lack of uniform standards, and the challenges of limited device resources such as computing power and energy efficiency are discussed.

The third section is devoted to the description of the development process of the portable element control system and the re-purposing of the technology stack during development. Also, testing of this software is conducted and described in this section.

Keywords: Internet of Things, portable devices, devices, control system.

## ЗМІСТ

ВСТУП .....	7
РОЗДІЛ 1 ТЕОРИТИЧНІ ОСНОВИ ІНТЕРНЕТУ РЕЧЕЙ .....	8
1.1 Поняття інтернету речей і архітектура IoT-систем .....	8
1.2 Основні компоненти і технології зв'язку в IoT .....	12
1.3 Сфери застосування IoT .....	15
РОЗДІЛ 2 АНАЛІЗ СУЧАСНИХ ЕЛЕМЕНТІВ ІНТЕРНЕТУ РЕЧЕЙ .....	19
2.1 Класифікація пристроїв IoT .....	19
2.2 Проблеми і виклики про розробці IoT пристроїв .....	21
2.3 Енергоспоживання та автономність портативних пристроїв .....	25
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВБУДОВАНОЇ СИСТЕМИ КЕРУВАННЯ.....	29
3.1 Вибір апаратної платформи для вбудованої системи .....	29
3.2 Розробка програмного забезпечення для керування пристроями.....	31
3.3 Тестування та налагодження системи.....	38
ВИСНОВКИ.....	40
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	42

## ВСТУП

Актуальність теми. Інтернет речей (IoT) стрімко розвивається і стає невід'ємною частиною сучасного світу, забезпечуючи взаємодію між фізичними об'єктами через інтернет. Особливе місце в цій екосистемі займають портативні елементи IoT, які забезпечують мобільність та зручність використання у повсякденному житті. Вбудовані системи керування такими пристроями є ключовим компонентом, що забезпечує їх функціонування, ефективність та безпеку. Актуальність даної теми зумовлена зростаючою потребою у розробці надійних, енергоефективних та безпечних рішень для управління портативними елементами IoT.

Метою роботи є розробка та впровадження вбудованої системи керування портативними елементами інтернету речей, що забезпечить їх ефективне функціонування та взаємодію.

Об'єкт дослідження – портативні елементи інтернету речей та їх функціонування.

Предмет дослідження – вбудована система керування портативними елементами інтернету речей.

Завдання, які необхідно виконати:

- Реалізувати прототип вбудованої системи керування портативними IoT пристроями.
- Розробити програмне забезпечення для управління та моніторингу стану портативних IoT пристроїв.
- Дослідити ефективність та надійність запропонованої системи.
- Візуалізувати результати моніторингу та керування портативними елементами IoT.
- Спроекувати архітектуру вбудованої системи керування з урахуванням безпеки та енергоефективності.
- Запропонувати рекомендації щодо вдосконалення та подальшого розвитку системи.

## РОЗДІЛ 1

### ТЕОРИТИЧНІ ОСНОВИ ІНТЕРНЕТУ РЕЧЕЙ

#### 1.1 Поняття інтернету речей і архітектура IoT-систем

У сучасному світі стрімкого технологічного прогресу та цифрових інновацій, інтернет речей представляє собою одну з найбільш значущих технологічних інновацій останніх десятиліть, яка має потенціал змінити наше розуміння взаємодії з навколишнім світом. IoT об'єднує мільярди фізичних об'єктів через Інтернет, дозволяючи їм збирати, обробляти та обмінюватися даними в реальному часі. Це створює нові можливості для автоматизації та оптимізації різних процесів, покращуючи ефективність і комфорт у багатьох аспектах нашого життя.

Інтернет речей (internet of things, IoT) – це «мережа взаємопов'язаних пристроїв, які з'єднуються й обмінюються даними з іншими пристроями і хмарою. Пристрої IoT, як правило, оснащені такими технологіями, як датчики та програмне забезпечення, і можуть включати механічні та цифрові машини та споживчі об'єкти» [1].

Також, як інтернет речей можна розуміти «як назв для сукупної колекції мережевих пристроїв, за винятком традиційних комп'ютерів, таких як ноутбуки та сервери» [2]. Типи мережевих підключень можуть включати Wi-Fi з'єднання, Bluetooth з'єднання, and near-field communication (NFC).

Основна концепція Інтернету речей полягає в інтеграції сенсорів, актуаторів, мережевих протоколів та програмного забезпечення для створення комплексних систем, здатних самостійно приймати рішення на основі аналізу отриманих даних. Це означає, що звичайні предмети, такі як побутова техніка, транспортні засоби, виробниче обладнання та навіть інфраструктурні об'єкти, можуть стати розумними, забезпечуючи новий рівень взаємодії та функціональності [3]. На рисунку 1.1 зображено приклад IoT системи.

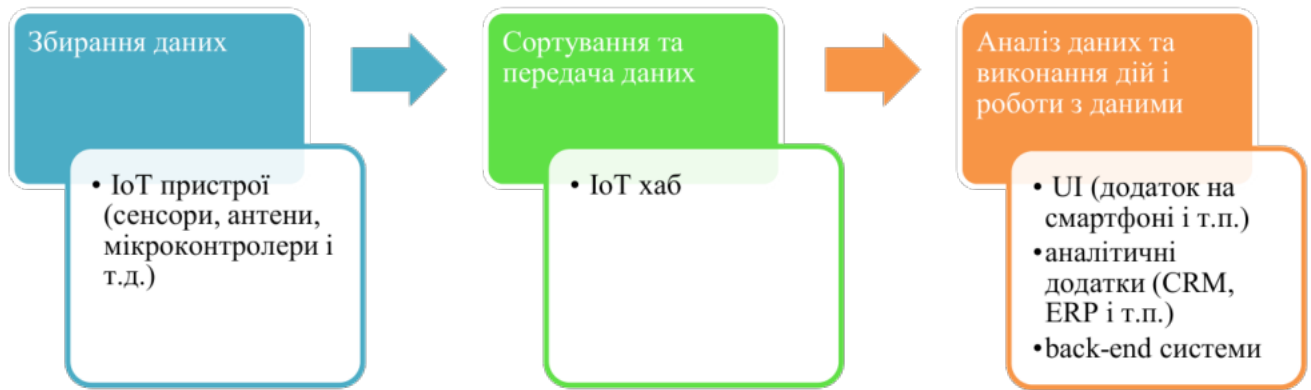


Рисунок 1.1 – Приклад ІоТ системи

До основних характеристик ІоТ систем можна віднести [4]:

- масштабованість: системи Інтернету речей можуть підтримувати все більшу кількість пристроїв. Незалежно від того, чи це розумний дім із кількома гаджетами чи великомасштабна промислова операція з тисячами датчиків, мережа має безперешкодно обробляти зростаючий потік даних;
- ідентифікація: кожному пристрою в мережі ІоТ потрібен унікальний ідентифікатор, щоб відрізнити його від інших. Це забезпечує правильну маршрутизацію даних і безпечний зв'язок між пристроями;
- обмін даними: пристрої збирають дані з навколишнього середовища (температура, тиск тощо) і передають їх іншим пристроям або хмарним платформам для подальшого аналізу та прийняття рішень;
- динамічність і самоадаптація: системи ІоТ не статичні. Вони можуть адаптуватися до мінливих умов і потреб користувачів. Наприклад, інтелектуальні термостати можуть регулювати кімнатну температуру залежно від кількості людей або погоди;
- безпека: оскільки в мережі передається величезна кількість даних, безпека має першочергове значення. Для захисту пристроїв і даних від кібератак і несанкціонованого доступу потрібні жорсткі заходи;
- сумісність: в ідеалі пристрої ІоТ від різних виробників повинні мати можливість безперерійно спілкуватися та працювати разом. Це забезпечує більшу гнучкість та інтеграцію в екосистему ІоТ;

- самостійність: багато пристроїв IoT оснащені вбудованими процесорами та програмним забезпеченням, які дозволяють їм самостійно збирати та аналізувати дані. Цей локальний інтелект дозволяє їм приймати базові рішення та виконувати автоматизовані дії без постійного зовнішнього контролю;

- підключення: це основа IoT. Для передачі та отримання даних пристроям потрібні надійні та безпечні з'єднання з Інтернетом або іншими мережами. Загальні протоколи включають Wi-Fi, Bluetooth і стільникові мережі.

Інтернет речей (IoT) - це концепція, яка передбачає мільярди взаємопов'язаних пристроїв, які обмінюються даними та формують те, як ми працюємо та живемо. У центрі лежить архітектура IoT, яку також можна назвати структурою, що складається з кількох елементів, які працюють разом, щоб забезпечити ефективну обробку інформації, що переміщується через фізичний рівень до рівня додатків, з'єднуючи датчики, програми, а також фізичні пристрої в одній консолідованій системі [5].

Зазвичай архітектура систем інтернету речей включає 4 шари:

- рівень сенсорів та сенсорних мереж;
- рівень шлюзів та мереж;
- сервісний рівень;
- рівень додатків.

Рівень сенсорів та сенсорних мереж – є основою систем IoT. Він включає як датчики, так і виконавчі механізми, які отримують і обробляють дані із зовнішніх джерел. Цей рівень відіграє вирішальну роль у отриманні необробленої інформації з фізичного середовища, щоб забезпечити вхідні дані для аналізу на вищих рівнях у межах архітектури. Наприклад, розгортання датчика на автомобільній складальній лінії можна використовувати для оцінки контролю якості за допомогою роботизованих функцій із ретрансляцією його вихідних даних на інші рівні для цілей обробки.

Без цього рівня, присутнього в будь-якій мережі Інтернету речей, ці системи не мали б здатності збирати життєво важливі знання, необхідні для взаємодії з реальністю, таким чином обмежуючи їх зовні, а також зменшуючи внутрішній потенціал.

Рівень шлюзів та мереж – необхідний для забезпечення переміщення цифрових даних між усіма елементами структури IoT. Він використовує такі технології, як HTTP, MQTT і AMQP, щоб полегшити передачу з однієї програми/пристрою на іншу. Не тільки це, цей самий рівень зв'язку може забезпечити безпечне середовище, коли мова йде про такі функції, як приватні мобільні мережі 5G, які забезпечують повний контроль над передачею інформації.

Це включає в себе кілька компонентів: інтернет-шлюзи, які дозволяють спілкуватися поза локальними системами, інтранет-порти, що забезпечують інтернет-шлюз і точки доступу в цих локальних системах, мережеві шлюзи, що з'єднують кілька пристроїв одночасно за допомогою мостових функцій, і, нарешті, системи збору даних (DAS), що дозволяють збирати показання датчиків через програми в системі.

Сервісний рівень – обробляє та аналізує зібрані дані, щоб допомогти компаніям приймати рішення, а також оптимізувати свою діяльність. Він може обробляти необроблену інформацію з систем IoT за допомогою алгоритмів машинного навчання, щоб зберегти корисні деталі, які використовуються для автоматичного прийняття рішень.

Гранична аналітика разом із методами ШІ також є частиною цього етапу, який допомагає відфільтрувати нерелевантну або непридатну інформацію про пристрій перед аналізом. Успішне впровадження цих заходів означає ефективні результати в реальному житті, оскільки це покращує якість розуміння, зібраного з отриманих необроблених даних, для ефективних дій відповідно до того, що потрібно під рукою – важливий крок до більш оптимізованих результатів, досягнутих системами Інтернету речей за допомогою зібраних даних.

Рівень додатків – служить життєво важливою платформою для взаємодії людей із системою та доступу до певних послуг. Це дозволяє користувачам зручно використовувати свої пристрої, чи то за допомогою мобільних додатків, чи за допомогою централізованих інформаційних панелей на комп'ютерах. Прикладом цього є випадки, коли хтось використовує програму, спеціально розроблену для розумних будинків. Вони можуть активувати кавоварку просто за допомогою функції натискання кнопок програми. Коротше кажучи, ці програми гарантують,

що люди мають безперервний досвід під час використання своїх систем IoT, щоб вони могли отримати від них максимальну користь.

## 1.2 Основні компоненти і технології зв'язку в IoT

Відповідно до складної архітектури систем, кожен з перелічених чотирьох шарів має свою велику кількість основних компонентів, їх у свою чергу ми можемо поділити на кілька великих груп: девайси, технології зв'язку, хмарні рішення, обробка даних, користувальницькі інтерфейси та безпека [6].

Під девайсами, зазвичай, мають на увазі, такі пристрої як датчики, всякі розумні пристрої, як от смартфон чи робот пилосос та інші. Вони розгортаються на фізичному рівні архітектури IoT, яка виступає основою системи. Головні функції цих речей і пристроїв полягають у сприйнятті та зборі даних із середовища та впровадженні будь-якого необхідного механізму керування для контролю необхідних змінних у середовищі [7].

Датчики – «це елементи які сприймають вхідний сигнал (вхідну величину, яка контролюється наприклад тиск, температура, струм, напруга та тощо) і перетворюють його в величину зручну для передачі, зберігання, використання у системі керування» [8].

До найпопулярніших датчиків відносяться:

- датчики температури: ці датчики вимірюють коливання температури в зоні вимірювання. Існують різні типи датчиків температури залежно від технології вимірювання температури, такі як термопари, термістори, ІЧ-датчики та резисторні датчики температури (RTD);

- датчики наближення: ці датчики виявляють наявність чи відсутність об'єкта поблизу або його властивості. Існують різні типи датчиків наближення, наприклад індуктивні датчики, ємнісні датчики, ультразвукові датчики та фотоелектричні датчики;

- датчики виявлення руху: ці датчики виявляють будь-який фізичний рух у певній зоні. Ультразвукові датчики, пасивні інфрачервоні датчики (PIR) і мікрохвильові датчики є одними з датчиків, які зазвичай використовуються;

- оптичні датчики: ці датчики здатні вловлювати фотони світла та перетворювати їх на сигнали для інтерпретації важливої інформації про навколишнє середовище;
- датчики тиску: ці датчики виявляють будь-яку зміну тиску в зоні вимірювання;
- гіроскопи та акселерометри: ці датчики здатні вимірювати будь-який обертальний рух і прискорення;
- хімічні та газові датчики: ці датчики виявляють наявність певних хімічних речовин і газів у навколишньому середовищі. Існують різні типи та сімейства цих датчиків залежно від вимог;
- датчики вологості: вони вимірюють вологість у навколишньому середовищі.

Наступною великою групою компонентів є технології зв'язку. Ця група є однією з ключових для коректної роботи IoT систем, адже вона забезпечує спілкування різноманітних пристроїв. Здебільшого ці технології зв'язку можна розділити на сім великих груп [9]:

- Bluetooth – створює бездротові з'єднання малого радіусу дії між пристроями IoT і смартфонами або концентраторами. Чудово працює в ситуаціях із низьким енергоспоживанням, надсилаючи пакети даних, які ідеально підходять для датчиків і переносних пристроїв. Однак його обмежений діапазон робить його непридатним для великомасштабних мереж IoT.

- Стільникові мережі – забезпечують широке покриття для пристроїв IoT, таких як дистанційні датчики або промислові машини. Вони використовують вежі стільникового зв'язку для передачі пакетів даних на великі відстані, пропонуючи надійний доступ до Інтернету практично будь-де. Однак стільникове підключення IoT може бути дорогим для пристроїв, які часто надсилають невеликі обсяги даних.

- Ethernet: пропонує дротове високошвидкісне підключення для потужних пристроїв IoT. Він безпечно передає пакети даних по кабелях, що ідеально підходить для додатків, які потребують постійного потоку даних або контролю в режимі реального часу. Однак дроти обмежують гнучкість і масштабованість порівняно з бездротовими варіантами.

- LPWAN (Low-Power Wide-Area Network), такі як LoRaWAN і NB-IoT, спеціалізуються на підключенні величезної кількості пристроїв Інтернету речей на величезних територіях. Вони відмінно справляються з рідкісним надсиланням невеликих пакетів даних, щоб максимізувати термін служби батареї віддалених датчиків і лічильників. Однак вони пропонують нижчу швидкість передачі даних порівняно з іншими варіантами.

- Меш-протоколи, такі як Zigbee, створюють самовідновлювані мережі, де пристрої передають дані один одному. Це розширює діапазон і обходить перешкоди, що робить його ідеальним для розумних будинків і автоматизації будівель. Однак сітчасті мережі можуть стати складними з багатьма пристроями і можуть потребувати центрального концентратора для зв'язку.

- RFID використовує радіохвилі для бездротової ідентифікації та відстеження об'єктів. Теги на предметах передають дані зчитувачам, коли вони близькі, що робить його ідеальним для управління запасами та контролю доступу. Однак RFID забезпечує обмежений обмін даними та потребує зчитувач у безпосередній близькості для зв'язку.

- Wi-Fi забезпечує підключення малої дальності з високою пропускнуою здатністю для пристроїв IoT у локальній зоні. Він використовує існуючі маршрутизатори Wi-Fi, що робить його зручним для домашньої автоматизації та розумних пристроїв. Однак радіус дії Wi-Fi може бути обмеженим, а для постійно підключених пристроїв може бути проблема з розрядкою акумулятора.

Варто зазначити, що вибір конкретної технології повинен залежати і від інших факторів, таких як, безпека, якщо девайси та програми працюють з конфіденційними даними, вартість розгортання та обслуговування, масштабованість чи простота використання.

У таблиці 1.1 наведено порівняння найпопулярніших технологій зв'язку в системах інтернету речей.

Таблиця 1.1 – Порівняння найпопулярніших технологій зв'язку в системах інтернету речей

Технологія	Діапазон дії	Споживання енергії	Швидкість передачі даних
2G (застаріла)	Широкий	Помірне	Низька
3G	Широкий	Помірне	Помірна
4G	Широкий	Помірне	Висока
5G	Широкий	Від середнього до високого	Дуже висока
Bluetooth	Низький	Низькіе	Низька
Стільникові мережі (NB-IoT, LTE-M)	Широкий	Дуже низьке	Від низької до середньої
LoRaWAN	Широкий	Дуже низьке	Дуже низька
Супутник (GNSS)	Глобальний	Має зовнішнє живлення	Помірна
PLC (Power Line Communication)	Середній	Низьке	Помірна
Ethernet	Обмежений (провідний)	Низьке	Висока
Меш-протоколи (Zigbee)	Розширений	Низьке	Помірний
RFID	Короткий	Пасивні теги	Дуже низька
Wi-Fi	Короткий	Помірне	Висока

### 1.3 Сфери застосування IoT

Не заважаючи на свою об'ємність у налаштуванні та розробці, інтернет речей має безліч можливостей для застосування. Його унікальність полягає у тому, що можна розробити та налаштувати все відповідно до наших потреб та постійно модифікувати чи підтримувати стан актуальним згідно них. Саме це надає системам IoT таку популярність у різноманітних сферах.

Однією з найбільш очевидних – буде розумний будинок. Адже, будь який тип житла, неважливо чи це квартира чи будинок, може бути оснащений різноманітними девайсами відповідно до наших щоденних потреб. Створюючи систему IoT самостійно, ви є інтегратором, який збирає ці розумні «речі», підключає їх до Інтернету, а також проектує та описує ваші потреби. Створення системи Інтернету речей, щоб зробити квартиру чи будинок розумними та здатними відповідати нашим потребам, потребує базових навичок IT та загальних знань.

У більшості систем Інтернету речей, а також у будинках із підтримкою таких систем датчики та виконавчі механізми утворюють повне локальне рішення. Для

контролю та взаємодії з системою можна використовувати спеціальні контролери або спеціальні програми на смартфонах і планшетах [10].

Наступною можливою сферою застосування буде сфера охорони здоров'я. Завдяки інтернету речей та використовуючи сучасні технології, ми можемо оточити себе чи датчиками, чи девайсами які постійно збиратимуть дані про стан нашого здоров'я та його аналізувати. Також, зібрані та посортовані дані можна надати відповідним спеціалістам, які на основі них можуть зробити висновки та надати рекомендації [10].

Ще однією сферою застосування можуть бути розумні міста. І, хоча, це звучить ніби фантастика, але в сучасних умовах це досить реально. Системи керування трафіком, розумне освітлення на вулицях, моніторинг забруднень та звітування цього, вирішення проблем з паркінгом, менеджмент сміття та води – це все може бути створене за рахунок інтернету речей [11].

Однією з найочевидніших сфер, певно є сфера носимих девайсів. Смарт годинники, навушники та інші пристрої, що супроводжують нас кожного дня також є чи можуть стати частиною інтернету речей. Багато людей розповідали історії, коли такі девайси врятували їхнє життя, а лікарі чи доглядачі можуть знати коли показники пацієнта є небезпечними для здоров'я. Розумні пристрої майбутнього, такі як розумні годинники та браслети для фітнесу, будуть оптимізовані для виконання більшої кількості функцій і підключення до інших розумних пристроїв IoT у розумному домі та інших додатків. Сполучення з додатками для смартфонів дозволить цим смарт-носним пристроям ініціювати більше завдань і миттєво отримувати сповіщення [11].

Однією з цікавих сфер застосування є сільське господарство. IoT системи допомагають збирати дані про стан здоров'я великої рогатої худоби, щоб вона могла витягнутись і запобігти захворюванню великої кількості худоби. Також, Ця технологія надає дані та розуміння фермерам, за допомогою яких вони можуть отримувати кращу віддачу від інвестицій. Відстеження вологості ґрунту та поживних речовин, контроль використання води для росту рослин і визначення індивідуальних добрив – це все прості способи використання IoT у сфері сільського господарства [12].

Також, системи IoT можуть бути корисними у логістиці, адже вони доможуть завдяки своїм можливостям відстеження товарів і обміну інформацією. Використовуючи IoT-технології, компанії можуть в режимі реального часу отримувати дані про місцезнаходження вантажів, що дозволяє оперативно реагувати на будь-які зміни або затримки. Крім того, ці системи допомагають оптимізувати управління запасами, забезпечуючи точну інформацію про кількість товарів на складі. Завдяки цьому підприємства можуть більш ефективно планувати поставки, уникати дефіциту або надлишків продукції і, в результаті, підвищувати свою загальну ефективність [13].

Застосування технологій Інтернету речей у сфері виробництва сприятиме спрощенню робочих процесів та створенню можливостей для впровадження нових IoT-рішень у виробничих циклах. Ці системи допоможуть з оптимізацією потоків, адже можна буде максимально ефективно керувати виробничими потоками, забезпечуючи безперервний і злагоджений процес роботи, завдяки збору та аналізу даних у реальному часі, можна ідентифікувати вузькі місця у виробничому процесі та вчасно приймати рішення для їх усунення; управлінням запасами в режимі реального часу, за допомогою IoT системи можна постійно моніторити наявність матеріалів та готової продукції на складах, це дозволяє уникати ситуацій, коли товарів не вистачає або навпаки, є їхній надлишок, що сприяє зменшенню витрат та оптимізації складських запасів; відстеженням активів, це допоможе зменшити втрати, забезпечити ефективне використання ресурсів та планувати технічне обслуговування, запобігаючи несправностям; безпекою працівників: IoT технології можуть значно підвищити рівень безпеки на виробництві, системи відстеження та контролю за допомогою датчиків можуть попереджати про небезпечні умови або інциденти, забезпечуючи швидке реагування на надзвичайні ситуації. Також це допомагає в моніторингу здоров'я та добробуту працівників, знижуючи ризик травматизму [14].

У сфері продажів, IoT системи змінять досвід покупок і дозволяють роздрібним торговцям краще взаємодіяти з клієнтами. Технології Інтернету речей, такі як маяки, RFID та розумні касові системи, революціонізують галузь роздрібною торгівлі. Технологія визначення місця розташування дозволяє роздрібним

торговцям надсилати персоналізовані пропозиції та сповіщення на смартфони клієнтів залежно від їхнього розташування в магазині. Технологія RFID забезпечує ефективне управління запасами та відстеження, зменшуючи брак і покращуючи видимість ланцюга поставок. Розумні системи оформлення замовлення на основі Інтернету речей спрощують процес оплати та підвищують задоволеність клієнтів. Ці системи використовують RFID або інші технології ідентифікації для автоматичного списання загальної суми з рахунку клієнта, усуваючи необхідність у традиційних чергах на касі [15].

Інтернет речей (IoT) у сфері енергетики, або Smart Energy, забезпечує ефективніше управління енергоресурсами, знижує витрати, підвищує надійність енергопостачання та сприяє інтеграції відновлюваних джерел енергії. Основними напрямками застосування є розумні лічильники, які дозволяють в реальному часі відслідковувати споживання електроенергії, води та газу, а також дистанційно передавати дані постачальникам послуг. Системи енергоменеджменту автоматично регулюють енергоспоживання залежно від потреб та тарифів, оптимізуючи витрати. Важливим є також впровадження інтелектуальних мереж, які автоматизують управління потоками енергії, виявляють та усувають несправності, і збалансовують навантаження. Інтеграція відновлювальних джерел енергії, таких як сонячні та вітрові електростанції, і децентралізована генерація також значно поліпшуються завдяки IoT. Акумуляторні системи зберігання енергії дозволяють ефективно використовувати надлишкову енергію, а інтелектуальні сервіси надають споживачам персоналізовані рекомендації щодо зниження енергоспоживання. Моніторинг та діагностика стану обладнання за допомогою сенсорів і аналізу даних дозволяють вчасно проводити технічне обслуговування та планувати модернізацію. Впровадження IoT у сфері енергетики сприяє підвищенню ефективності використання енергії, розвитку відновлювальних джерел, зменшенню впливу на довкілля та покращенню якості обслуговування споживачів.

## РОЗДІЛ 2

### АНАЛІЗ СУЧАСНИХ ЕЛЕМЕНТІВ ІНТЕРНЕТУ РЕЧЕЙ

#### 2.1 Класифікація пристроїв IoT

Через велику різноманітність, пристрої в системах інтернету речей, можна класифікувати за великою кількістю ознак та підходів. Наприклад, за сферою застосування ці пристрої можна поділити на [16]:

- Розумний будинок: Ці пристрої використовуються для автоматизації та керування домашніми приладами та системами, такими як освітлення, опалення, безпека та розваги.

- Носимі пристрої: Ці пристрої носять на тілі і збирають дані про здоров'я та фізичну активність. Прикладами носимих пристроїв IoT є фітнес-трекери, смарт-годинники та розумні окуляри.

- Індустріальний Інтернет речей (IIoT): Ці пристрої використовуються в промислових налаштуваннях для моніторингу та керування машинами, процесами та інфраструктурою. Прикладами IIoT-пристроїв є датчики вібрації, датчики температури та системи керування двигунами.

- Смарт-місто: Ці пристрої використовуються для збору даних про міську інфраструктуру та навколишнє середовище, а також для покращення ефективності та якості життя жителів міста. Прикладами пристроїв Smart City IoT є датчики паркування, датчики якості повітря та розумні ліхтарі.

- Смарт-сільське господарство: Ці пристрої використовуються для моніторингу та керування умовами вирощування, такими як вологість ґрунту, поживні речовини та здоров'я рослин. Прикладами пристроїв IoT для розумного сільського господарства є датчики вологості ґрунту, датчики зрошення та системи моніторингу врожаю.

За призначенням:

- Споживчий.
- Промисловий.
- Комерційний.

За функціональністю пристрої поділяють на [17]:

- Шлюзи: Ці пристрої служать мостом між IoT-пристроями та Інтернетом. Вони збирають дані з датчиків, обробляють їх і надсилають до хмарних платформ або інших систем. Прикладами шлюзів IoT є розумні колонки та роутери.

- Датчики: Ці пристрої збирають дані з навколишнього середовища, такі як температура, вологість, тиск або рух. Прикладами датчиків IoT є термостати, фітнес-трекери та розумні лічильники.

- Виконавчі пристрої: Ці пристрої можуть керувати фізичними об'єктами або системами. Прикладами виконавчих пристроїв IoT є розумні термостати, розумні лампочки та розумні дверні замки.

За протоколами зв'язку:

- HTTP.
- MQTT.
- CoAP.

За рівнем безпеки:

- Низький: пристрої які мають низький рівень безпеки і можуть бути легко скомпрометовані.

- Середній: пристрої які мають деякі функції безпеки, але вони все ще можуть бути вразливими до атак.

- Високий: пристрої які мають надійні функції безпеки, що ускладнює їх компрометацію.

За форм-фактором:

- Настільні.
- Настінні.
- Носимі.
- Вбудовані.

За обчислювальними можливостями:

- Обмежені: пристрої, що можуть виконувати лише прості завдання.
- Потужні: пристрої, що можуть виконувати складні завдання.

За типом даних:

- Датчики: збирають дані про навколишнє середовище, такі як температура, вологість, тиск або рух.

- Лічильники: підраховують події, такі як кількість кроків, кількість пройдених відстаней або кількість використаної води.

- Сигнали: генерують сигнали, коли виявляють певну подію, наприклад, відкриття дверей або вікна.

За способом живлення:

- Мережеві.
- Батарейні.
- Сонячні.
- Самозарядні.

За типом мережі:

- Дротові: Ці пристрої підключаються до Інтернету за допомогою кабелів Ethernet.

- Бездротові: Ці пристрої підключаються до Інтернету за допомогою бездротових технологій, таких як Wi-Fi, Bluetooth або Zigbee.

- Мережеві: Ці пристрої підключаються до Інтернету за допомогою мобільних мереж, таких як 3G, 4G або 5G.

Оскільки, є дуже багато різноманітних пристроїв то і кількість класифікацій є безмежною. З часом, деякі з них перестануть бути актуальним, а також, обов'язково з'являться нові. Також, важливо зазначити, що конкретний вид класифікацій залежить від конкретних потреб і не обмежується переліченими вище.

## 2.2 Проблеми і виклики про розробці IoT пристроїв

Незважаючи на те, що IoT пристрої досить корисні, як для комерційного, так і для особистого використання, все ще є проблеми коли справа доходить до складності даних пристроїв IoT, керування та підключення. До найпоширеніших проблем можна віднести: складність даних, неефективне управління пристроєм, відсутня гнучкість у випадку власної розробки, безпека [18].

Пристрої IoT постійно збирають великі об'єми даних та відправляють їх на сервер. Їх різноманітність, яка надходить, може призвести до створення складних

наборів даних, які необхідно уніфікувати, перш ніж їх можна буде проаналізувати для отримання інформації.

Також, для масштабної IoT системи вам знадобиться здатність керувати тисячами, сотнями тисяч і навіть мільйонами пристроїв. Реєстрація та керування цими пристроями протягом усього життєвого циклу, від моніторингу та обслуговування до виходу з експлуатації, потенційно може стати великою проблемою за відсутності коректно спроектованої та реалізованої IoT системи.

Витрати на створення IoT системи вимагають значних інвестицій як часу, так і ресурсів для розробки та підтримки. Домашня IoT система може добре вирішити одну конкретну проблему, але, ймовірно, вводить технічну складність, яка ускладнює запуск інших ініціатив, що розширюватимуть її функціонал. Крім того, багато IoT систем не розроблені для масштабування, тобто вони досягають обмеження кількості пристроїв, які вони можуть підключити, або обсягу даних, які вони можуть збирати, обробляти та зберігати.

Вартно зазначити, що за даними багатьох досліджень, майже три чверті проєктів IoT систем [19] не вважатимуться безпечними для користування, адже без надійного захисту будь який девайс з IoT системи буде надзвичайно вразливим, що полегшить роботу для хакерів.

Якщо говорити не зі сторони експлуатації, а зі сторони розробки то виклики полягатимуть у, першу чергу, безпеці, витратах часу на розробку, надійності обладнання, легкості інтеграцій з зовнішніми розробками, протоколами зв'язку між пристроями, навичках розробника, тестуванні та архітектурі і дизайні. Розглянемо кожен з них детальніше.

Основною проблемою як і при експлуатації, так і розробці є безпека. Це надзвичайно важливий нюанс, адже розробники які зазвичай розробляють такі системи не є експертами у галузі кібербезпеки та мають лише її базове розуміння, що призводить до багатьох уразливостей в коді та прошивці девайсів. В першу чергу, нестача шифрування, що призводить до можливості втручання сторонніх осіб в процес обміну даними та можливе їхнє викрадення чи підміну. Наступним важливим питанням є можливості доступу до девайсів, адже, якщо девайси не

захищені належним чином, то це може призвести до надання доступу до всієї мережі.

Наступною проблемою як і при експлуатації, так і розробці є безпека, є затрати часу. Оскільки, IoT системи насправді є складними і досить об'ємними для архітектури і розробки є багато невидимих витрат. Вартість оновлень, обслуговування, проектування, заміни застарілих пристроїв і необхідних наборів технічних навичок може накопичуватися дуже швидко, а знайти команду з необхідним досвідом не так вже і просто.

Важливо пам'ятати, що будь яка IoT система залежить від девайсів, що в ній використовуються, але і тут не все так просто. Адже їх якість, технічне обслуговування, надійність і ефективність надзвичайно важливі. Візьмемо, наприклад, датчики. Багато людей вважають, що дешеві датчики легкодоступні та ефективні. Однак довговічні та надійні датчики рідко бувають недорогими. Якщо технічне обслуговування датчиків, які використовуються для делікатних операцій, таких як датчики газу або рН, не є регулярним і ретельним, дані, отримані від них, можуть бути ненадійними.

Датчики можуть збирати та передавати дані лише за умови надійного підключення, а це не завжди так. Особливо, коли їх розміщено у віддалених районах, підключення може стати справжньою проблемою. Коли вам потрібно підключити кілька хмарних серверів, фізичних пристроїв і програм, зв'язок дуже важливий для успіху вашого проекту. Локальне підключення не менш важливо для пристроїв IoT. Наприклад, якщо ваші пристрої підключені через Bluetooth, вони не працюватимуть належним чином на з'єднаннях на великій відстані.

Однією з важливих частин розробки IoT систем є забезпечення того, що вона може бути інтегрована з різними платформами, операційними системами, хмарними службами та навіть застарілими системами. Невиконання цього може призвести до того, що продукт буде незручним для використання, і він може не витримати прогресу в технології та бізнес-практики.

Знайти людей, які є експертами в IoT і розуміють вимоги до безпеки, апаратного забезпечення та користувачів, надзвичайно складно. Це

багатодоменний досвід, який також супроводжується подальшою спеціалізацією в галузі та масштабі.

Разом із кваліфікованими фахівцями зріс і попит на обладнання. Прагнучи задовольнити цей попит, виробники часто поспішають із процесом, що призводить до відсутності тестування та оновлень. Пристрої, які не пройшли ретельну перевірку, можуть становити загрозу безпеці та часто виявляються ненадійними.

Оскільки IoT стає звичним явищем у всьому світі, зручність використання стає ключовою проблемою. Інтерфейс, який дозволяє працювати з кількома пристроями одночасно, має бути розроблений так само добре, як і система в цілому. Якщо не брати до уваги досвід користувача, ефективність ваших пристроїв і обробки мало що зміниться.

Однак зручність використання передбачає не лише інтерфейс користувача. Проблеми користувача, такі як час автономної роботи та підключення, не менш важливі для загального досвіду користувача. Ці проблеми можуть призвести до створення поганого продукту, який не буде дуже корисним для кінцевих користувачів.

Енергоефективність є однією з основних проблем у розробці IoT пристроїв. Багато з цих пристроїв працюють від батарей або використовують обмежені джерела живлення, що вимагає максимальної ефективності у споживанні енергії. Високий рівень енергоефективності необхідний для продовження терміну служби пристроїв та зменшення частоти заміни батарей, що особливо важливо для пристроїв, встановлених у важкодоступних місцях. Розробники стикаються з викликом зменшення енергоспоживання без втрати функціональності та продуктивності пристроїв. Це включає оптимізацію апаратного забезпечення, розробку енергоефективних алгоритмів та протоколів передачі даних, а також використання новітніх технологій, таких як енергоефективні мікроконтролери та сенсори. Крім того, важливо враховувати вплив режимів очікування та сну для мінімізації енергоспоживання, коли пристрій не активно виконує свої основні функції. Всі ці заходи спрямовані на те, щоб забезпечити довговічність і стабільну роботу IoT пристроїв у реальних умовах експлуатації.

Здебільшого цих проблем можна уникнути, дотримуючись кількох простих порад:

- перевірте технологічних партнерів, яким ви доручаєте роботу над проектом. Шукайте: членів команди з досвідом у всьому стеку, сильним портфоліо, можливостями автоматизації, керування пристроями та технічними ноу-хау;

- враховуйте час і бюджет на рішення з покращення безпеки IoT системи;
- дослідіть якість, довговічність і надійність пристроїв, у які ви інвестуєте.

Хоча доступно багато дешевих альтернатив, у довгостроковій перспективі вони можуть обійтися вам дорожче;

- переконайтеся, що оновлення та обслуговування є частими та регулярними;

- досліджуйте аналоги на ринку для того, щоб побудувати зрозумілий інтерфейс користувача, якщо аналогів немає – проводьте опитування потенційних груп користувачів.

Також важливо не забувати про відповідне тестування розроблених чи існуючих рішень, адже тестування є критично важливим етапом у розробці IoT систем. Воно допомагає виявити потенційні проблеми та недоліки на ранніх етапах, забезпечуючи надійність та стабільність роботи системи. Ретельне тестування дозволяє перевірити функціональність, продуктивність, безпеку та взаємодію між компонентами системи, а також їх стійкість до пікових навантажень та несприятливих умов. Без належного тестування IoT система може працювати з перебоями, давати некоректні дані або ставати вразливою до кіберзагроз, що в кінцевому результаті може призвести до значних фінансових втрат та втрати довіри користувачів.

### **2.3 Енергоспоживання та автономність портативних пристроїв**

Портативні пристрої, такі як мобільні телефони, планшети та ноутбуки, є невід'ємною частиною сучасного життя. Однак, однією з основних проблем, які вони стикаються, є обмежена тривалість роботи від одного заряду батареї. Ефективне управління енергоспоживанням та підвищення автономності цих

пристроїв стають ключовими завданнями для вирішення цієї проблеми. У цьому підрозділі буде проведено аналіз основних факторів, що впливають на енергоспоживання портативних пристроїв та їхню автономність [20].

Фактори, що впливають на енергоспоживання та автономність портативних пристроїв:

- Потужність батареї: Обсяг та ефективність батареї безпосередньо впливають на час роботи пристрою від одного заряду. Розвиток нових технологій акумуляторів, які б забезпечували більшу ємність та швидше зарядження, є важливим напрямком досліджень.

- Енергоефективність компонентів: Процесори, дисплеї, сенсори та інші компоненти пристроїв різняться за своєю енергоефективністю. Використання менш енергоємних компонентів або розробка ефективних методів управління енергією може значно збільшити автономність пристрою.

- Використання енергоефективних комунікаційних протоколів: Вибір оптимальних протоколів для передачі даних може значно впливати на енергоспоживання пристрою. Наприклад, використання Bluetooth Low Energy (BLE) замість традиційного Bluetooth може дозволити значно зменшити споживання енергії при збереженні можливостей зв'язку.

- Програмне забезпечення: Правильна оптимізація програмного забезпечення щодо енергоспоживання може суттєво зменшити витрату енергії. Використання ефективних алгоритмів сну, обмеження фонових процесів та оптимізація роботи додатків можуть позитивно позначитися на тривалості роботи пристрою.

- Оптимізація синхронізації та передачі даних: Синхронізація та передача даних є важливими функціями портативних пристроїв, але вони можуть споживати значну кількість енергії, особливо при роботі в бездротових мережах. Ефективне керування цими процесами, таке як зменшення частоти синхронізації, використання оптимальних протоколів передачі даних та управлінням потоками даних, може значно зменшити споживання енергії та збільшити тривалість роботи пристрою.

- Оптимізація використання сенсорів та периферійних пристроїв: Багато сучасних портативних пристроїв оснащені різними сенсорами та периферійними пристроями, такими як геолокаційні модулі, камери, датчики руху тощо. Ефективне керування цими пристроями та оптимізація їхнього використання можуть допомогти зменшити споживання енергії. Наприклад, вмикання сенсорів лише тоді, коли вони дійсно потрібні для роботи додатків, або регулювання частоти їх оновлення може значно зменшити енергоспоживання.

- Вплив зовнішніх чинників: Фактори навколишнього середовища, такі як температура та якість сигналу мережі, можуть також впливати на тривалість автономної роботи пристрою. Наприклад, в холодну погоду акумулятор може швидше розряджатися.

- Вплив апаратних та програмних оновлень: Оновлення апаратного та програмного забезпечення пристрою можуть включати в себе вдосконалені алгоритми керування енергією та оптимізацію роботи пристрою, що може позитивно позначитися на його автономності.

Далі можна очікувати подальший розвиток інтегрованих енергоефективних рішень, які враховуватимуть усі аспекти пристрою, від апаратної складової до програмного забезпечення. Інноваційні підходи до керування енергією та використання відновлюваних джерел енергії, таких як сонячні панелі чи кінетичні зарядники, також будуть набувати все більшого значення [21].

Крім того, розвиток мережі 5G може вплинути на споживання енергії портативних пристроїв. Вдосконалені мережеві технології можуть допомогти зменшити споживання енергії під час передачі даних, що відкриє нові можливості для тривалого використання пристроїв без підзарядки.

Також, індустрія продовжуватиме дослідження та впровадження нових матеріалів для батарей та компонентів пристроїв, що дозволить підвищити енергетичну щільність та тривалість роботи пристроїв. Наприклад, деякі дослідження спрямовані на розробку літєвих батарей з вищою енергетичною щільністю, що дозволяє збільшити їхню ємність при збереженні або навіть зменшити їхню масу. Крім того, активно вивчаються альтернативні типи батарей,

такі як суперконденсатори, які можуть забезпечувати велику кількість циклів заряджання-розряджання та мають потенціал для швидкого заряджання.

Додатково, можливості дистанційного заряджання пристроїв набувають все більшої популярності. Технології, такі як бездротове заряджання за стандартами Qi, дозволяють заряджати пристрої безпосередньо з відповідно підготовленої поверхні, що забезпечує більшу зручність для користувачів та дозволяє продовжити автономність пристроїв.

Ще однією перспективою є розвиток технологій, які дозволяють ефективно використовувати енергію, що втрачається або раніше була маловикористана. Наприклад, концепція термоелектричних пристроїв, які виробляють електроенергію з теплової енергії, може знайти застосування в портативних пристроях. Такі пристрої можуть використовувати тепло, що виділяється від пристрою чи навколишнього середовища, для заряджання вбудованих акумуляторів, підвищуючи їхню автономність.

Крім того, розвиток технологій інтернету речей (IoT) може вплинути на енергоефективність портативних пристроїв шляхом оптимізації їх взаємодії з іншими пристроями та середовищем. Алгоритми розумного управління енергією можуть допомагати пристроям автоматично реагувати на зміни в умовах навколишнього середовища та робити оптимальні рішення з точки зору енергозбереження.

У майбутньому можна очікувати подальшого зростання ефективності та автономності портативних пристроїв завдяки поєднанню цих технологічних тенденцій та постійному пошуку нових інновацій.

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВБУДОВАНОЇ СИСТЕМИ КЕРУВАННЯ

#### 3.1 Вибір апаратної платформи для вбудованої системи

Вибір апаратної платформи є критичним етапом розробки вбудованої системи, оскільки саме апаратна платформа визначає можливості та обмеження системи. У цьому розділі розглянемо критерії вибору апаратної платформи для проекту на основі ESP32 та BM160, а також обґрунтуємо наш вибір.

При виборі апаратної платформи для вбудованої системи необхідно враховувати кілька основних критеріїв:

1. обчислювальна потужність – платформа повинна мати достатню обчислювальну потужність для виконання всіх необхідних завдань.
2. Енергоефективність – важливо оцінити енергоспоживання платформи, особливо для автономних систем, що працюють від батарей.
3. Інтерфейси вводу/виводу - наявність необхідних інтерфейсів для підключення сенсорів, виконавчих пристроїв та інших компонентів.
4. Підтримка програмного забезпечення – наявність бібліотек, драйверів та прикладів коду, що спрощують розробку.
5. Вартість – вартість компонентів повинна бути прийнятною в межах бюджету проекту.
6. Розмір і форма – фізичні розміри плати повинні відповідати вимогам проекту.

Для нашої вбудованої системи була обрана ESP32S-S2 з наступних причин:

- ESP32 оснащена однокристальним процесором Xtensa LX7 з тактовою частотою до 240 МГц і має до 320 КБ SRAM, що забезпечує достатню потужність для виконання складних обчислювальних завдань та обробки даних з сенсорів.
- ESP32 має кілька режимів енергозбереження, включаючи режим глибокого сну, що дозволяє зменшити енергоспоживання до декількох мікроамперів, що є критично важливим для автономних систем.

- ESP32 підтримує різноманітні інтерфейси, такі як GPIO, I2C, SPI, UART, що дозволяє легко підключати широкий спектр сенсорів та інших периферійних пристроїв. BMI160 підключається через інтерфейс I2C або SPI, обидва з яких підтримуються ESP32.

- ESP32 має велику спільноту розробників та широкий вибір бібліотек, що спрощує процес розробки. Платформа підтримується Arduino IDE, що робить розробку інтуїтивно зрозумілою навіть для новачків.

- ESP32 є досить доступною за ціною, що робить її привабливим варіантом для освітніх проектів та прототипів.

- ESP32 доступна в різних форм-факторах, що дозволяє вибрати плату, яка найкраще відповідає вимогам нашого проекту.

BMI160 був обраний як інерціальний вимірювальний модуль (IMU) для нашої системи з наступних причин:

- BMI160 забезпечує високу точність вимірювань прискорення та кутових швидкостей, що є критично важливим для задач орієнтації та стабілізації.

- Сенсор підтримує як I2C, так і SPI інтерфейси, що забезпечує гнучкість при підключенні до різних мікроконтролерів. У нашому випадку, він буде підключений до ESP32 через I2C інтерфейс.

- BMI160 має низьке енергоспоживання, що дозволяє використовувати його в автономних системах без значного збільшення споживання енергії.

- Компактні розміри сенсора дозволяють використовувати його в малогабаритних пристроях.

Вибір апаратної платформи для вбудованої системи є важливим етапом, що впливає на всі наступні етапи розробки. Враховуючи всі перелічені критерії, ESP32 була обрана як оптимальний мікроконтролер для нашої системи, а BMI160 — як інерціальний вимірювальний модуль. Ці компоненти забезпечують необхідну обчислювальну потужність, енергоефективність, гнучкість підключення та підтримку програмного забезпечення, що дозволяє створити ефективну та надійну вбудовану систему.

### 3.2 Розробка програмного забезпечення для керування пристроями

Архітектура програмного забезпечення для нашого пристрою включає кілька основних компонентів:

1. Системний модуль: відповідає за ініціалізацію та конфігурацію апаратних компонентів, керування енергоспоживанням та забезпечення основних системних функцій.
2. Модуль керування сенсорами: відповідає за взаємодію з BMI160, збір та попередню обробку даних з сенсора.
3. Модуль розпізнавання жестів: аналізує дані з сенсора для розпізнавання жестів кубика.
4. Комунікаційний модуль: забезпечує обмін даними з зовнішніми пристроями через інтерфейси, такі як Wi-Fi або Bluetooth.
5. Інтерфейс користувача: дозволяє користувачам взаємодіяти з системою через командний інтерфейс або мобільний додаток.

Розробка ПЗ починається з ініціалізації та конфігурації апаратних компонентів, зокрема ESP32 та BMI160. Налаштування ESP32 включає ініціалізацію периферійних інтерфейсів (GPIO, I2C, SPI, UART), налаштування Wi-Fi та/або Bluetooth для підключення до мережі або мобільного додатку, а також встановлення режиму енергозбереження. Ініціалізація BMI160 включає встановлення з'єднання через I2C, конфігурацію параметрів сенсора (частота вимірювань, режими роботи) та перевірку працездатності сенсора.

Основним завданням модуля керування сенсорами є збір даних з BMI160 та їх попередня обробка. Збір даних включає періодичне зчитування даних з акселерометра та гіроскопа, перевірку коректності даних та обробку можливих помилок зчитування. Попередня обробка даних включає фільтрацію сигналів для зменшення шумів, компенсацію зміщень та калібрування сенсора.

Модуль розпізнавання жестів аналізує зібрані дані для виявлення конкретних жестів та орієнтацій кубу. Аналіз рухів включає визначення типів рухів, таких як поворот, трясіння, обертання та нахил, а також визначення напрямку та швидкості руху. Алгоритми розпізнавання використовуються для розпізнавання шаблонів

рухів та їх інтерпретації, зокрема застосування фільтра Калмана або інших алгоритмів для покращення точності розпізнавання. Інтеграція з розумним домом дозволяє визначати дії, які відповідають розпізнаним жестам (наприклад, перемикання світла, регулювання температури тощо), та відправляти команди до центрального контролера або безпосередньо до керованих пристроїв.

Комунікаційний модуль забезпечує обмін даними між системою та зовнішніми пристроями, а також взаємодію з користувачем. Передача даних включає відправлення зібраних даних та команд до центрального контролера системи розумного дому через Wi-Fi або Bluetooth, а також обробку команд від користувача.

Для підключення до мережі Wi-Fi використовується API для роботи з Wi-Fi (рисунок 3.1). Основна логіка підключення реалізована в класі WifiClient, який включає наступні кроки:

1. ініціалізація TCP/IP стеку за допомогою esp\_netif\_init().
2. Ініціалізація модулю Wi-Fi за допомогою esp\_wifi\_init().
3. Налаштування конфігурації Wi-Fi (SSID та пароль) за допомогою esp\_wifi\_set\_config() і з'єднання з мережею.

```

esp_err_t WifiClient::init() {
    ESP_ERROR_CHECK(esp_netif_init());
    ESP_ERROR_CHECK(esp_event_loop_create_default());
    esp_netif_create_default_wifi_sta();

    wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT();
    ESP_ERROR_CHECK(esp_wifi_init(&cfg));
    ESP_ERROR_CHECK(esp_event_handler_instance_register(
        WIFI_EVENT, ESP_EVENT_ANY_ID, &WifiClient::event_handler, this, NULL));
    ESP_ERROR_CHECK(esp_event_handler_instance_register(
        IP_EVENT, IP_EVENT_STA_GOT_IP, &WifiClient::event_handler, this, NULL));

    wifi_config_t wifi_config = {};
    strncpy((char*)wifi_config.sta.ssid, ssid, sizeof(wifi_config.sta.ssid) - 1);
    strncpy((char*)wifi_config.sta.password, password, sizeof(wifi_config.sta.password) - 1);
    wifi_config.sta.ssid[sizeof(wifi_config.sta.ssid) - 1] = '\0'; // Ensure null-terminated string
    wifi_config.sta.password[sizeof(wifi_config.sta.password) - 1] = '\0'; // Ensure null-terminated string

    ESP_ERROR_CHECK(esp_wifi_set_mode(WIFI_MODE_STA));
    ESP_ERROR_CHECK(esp_wifi_set_config(WIFI_IF_STA, &wifi_config));

    ESP_ERROR_CHECK(esp_wifi_start());

    ESP_ERROR_CHECK(esp_wifi_connect());
    return ESP_OK;
}

```

Рисунок 3.1 – Метод init() класу WifiClient

Для налаштування та використання MQTT (Message Queuing Telemetry Transport) використовується клас MqttClient. Цей код відповідає за налаштування

та ініціалізацію клієнта для забезпечення комунікації з брокером MQTT (рисунок 3.2).

```

MqttClient::MqttClient(const char* mqtt_host, const char* mqtt_port, const char* mqtt_username, const char* mqtt_password)
: mqtt_host(mqtt_host), mqtt_port(mqtt_port), mqtt_username(mqtt_username), mqtt_password(mqtt_password) {
    int portNum = atoi(mqtt_port);
    esp_log_level_set(TAG, ESP_LOG_INFO);
    esp_mqtt_client_config_t mqtt_cfg = {};
    mqtt_cfg.broker.address.uri = mqtt_host,
    mqtt_cfg.broker.address.port = portNum,
    mqtt_cfg.credentials.username = mqtt_username,
    mqtt_cfg.credentials.authentication.password = mqtt_password,

    client = esp_mqtt_client_init(&mqtt_cfg);
    esp_mqtt_client_register_event(client, (esp_mqtt_event_id_t)ESP_EVENT_ANY_ID, mqtt_event_handler, this);
    esp_mqtt_client_start(client);
}

```

Рисунок 3.2 – Конструктор класу MqttClient

Ось опис його кроків:

1. Перетворення порту у ціле число: Спочатку рядок, який представляє номер порту MQTT брокера, перетворюється у ціле число за допомогою функції `atoi()`. Це потрібно для подальшого використання цього значення при ініціалізації клієнта MQTT.

2. Встановлення рівня журналювання: Встановлення рівня журналювання для клієнта MQTT за допомогою функції `esp_log_level_set()`. У цьому випадку використовується рівень журналювання `ESP_LOG_INFO`.

3. Налаштування конфігурації клієнта MQTT: Створюється конфігурація для клієнта за допомогою структури `esp_mqtt_client_config_t`. В цій конфігурації вказуються адреса та порт MQTT брокера, а також облікові дані для аутентифікації (ім'я користувача та пароль), якщо вони є.

4. Ініціалізація клієнта MQTT: За допомогою функції `esp_mqtt_client_init()` створюється новий клієнт з заданою конфігурацією.

5. Реєстрація обробників подій для клієнта: За допомогою функції `esp_mqtt_client_register_event()` реєструються обробники подій для клієнта MQTT. Це дозволяє визначати та обробляти різні події, які виникають під час роботи з MQTT.

6. Запуск клієнта MQTT: Функція `esp_mqtt_client_start()` запускає клієнта MQTT, розпочинаючи процес підключення до брокера.

Клас `Imu` призначений для роботи з інерціальними датчиками `BMI160` та для обробки їхніх даних. Основні функції цього класу включають:

1. Конструктор `Imu`: При створенні об'єкта класу `Imu` вказується адреса датчика, порт `I2C`, а також пini `SDA` та `SCL`. В конструкторі ініціалізуються основні параметри датчика, такі як `ID`, інтерфейс, час затримки тощо (рисунок 3.3).

```
Imu::Imu(uint8_t i2c_address, i2c_port_t i2c_port, gpio_num_t sda_pin, gpio_num_t scl_pin)
{
    sensor.id = i2c_address;
    sensor.intf = BMI160_I2C_INTF;
    sensor.read = user_i2c_read;
    sensor.write = user_i2c_write;
    sensor.delay_ms = user_delay_ms;
    accel_sensitivity = 16384.0;
    gyro_sensitivity = 131.2;
    last_time = TimeToSec();
    i2c_master_init(i2c_port, sda_pin, scl_pin);
}
```

Рисунок 3.3 – Конструктор класу `Imu`

2. Метод `init()`: Ініціалізує датчик `BMI160` та налаштовує його параметри. Встановлюється режим роботи датчика, діапазони і динаміка прискорення та обертання (рисунок 3.4).

```
esp_err_t Imu::init()
{
    int8_t rslt = bmi160_init(&sensor);
    if (rslt != BMI160_OK) {
        ESP_LOGE(TAG, "Failed to initialize BMI160: %d", rslt);
        return ESP_FAIL;
    }

    sensor.accel_cfg.odr = BMI160_ACCEL_ODR_100HZ;
    sensor.accel_cfg.range = BMI160_ACCEL_RANGE_2G;
    sensor.accel_cfg.bw = BMI160_ACCEL_BW_NORMAL_AVG4;
    sensor.accel_cfg.power = BMI160_ACCEL_NORMAL_MODE;

    sensor.gyro_cfg.odr = BMI160_GYRO_ODR_100HZ;
    sensor.gyro_cfg.range = BMI160_GYRO_RANGE_250_DPS;
    sensor.gyro_cfg.bw = BMI160_GYRO_BW_NORMAL_MODE;
    sensor.gyro_cfg.power = BMI160_GYRO_NORMAL_MODE;
    sensor.delay_ms = user_delay_ms;
    rslt = bmi160_set_sens_conf(&sensor);
    if (rslt != BMI160_OK) {
        ESP_LOGE(TAG, "Failed to set BMI160 sensor configuration: %d", rslt);
        return ESP_FAIL;
    }
    ESP_LOGI(TAG, "BMI160 sensor initialized successfully");
    return ESP_OK;
}
```

Рисунок 3.4 – Метод `init()` класу `Imu`

3. Метод `update()`: Зчитує дані з датчика BMI160 і виконує оновлення показань. Обробляються дані про прискорення та обертання, а також визначається поворот куба (рисунок 3.5).

```
void Imu::update()
{
    bmi160_sensor_data accel;
    bmi160_sensor_data gyro;

    int8_t rslt = bmi160_get_sensor_data(BMI160_ACCEL_SEL | BMI160_GYRO_SEL, &accel, &gyro, &sensor);
    if (rslt == BMI160_OK) {
        float dt = (TimeToSec() - last_time);
        last_time = TimeToSec();

        ax = (float)accel.x / accel_sensitivity;
        ay = (float)accel.y / accel_sensitivity;
        az = (float)accel.z / accel_sensitivity;
        gx = (float)gyro.x / gyro_sensitivity;
        gy = (float)gyro.y / gyro_sensitivity;
        gz = (float)gyro.z / gyro_sensitivity;

        madgwick.updateIMU(gx, gy, gz, ax, ay, az, dt);
        static uint64_t rotationUpdateTime = esp_timer_get_time();
        if ((esp_timer_get_time() - rotationUpdateTime) >= 1000*2000)
        {
            update_cube_rotation();
            rotationUpdateTime = esp_timer_get_time();
        }
    } else {
        ESP_LOGE(TAG, "Failed to get sensor data: %d", rslt);
    }
}
```

Рисунок 3.5 – Метод `update()` класу `Imu`

4. Методи `getRoll()`, `getPitch()`, `getYaw()`: Повертають значення кутів кочення, тангажу та рискання на основі оброблених даних з датчика.

5. Методи `i2c_master_init()`, `user_i2c_read()`, `user_i2c_write()`, `user_delay_ms()`, `TimeToSec()`: Допоміжні методи для роботи з шиною I2C та для обробки часу.

6. Методи `update_cube_rotation()` та `getRotation()`: Метод `update_cube_rotation()` розраховує поворот куба на основі отриманих даних про прискорення, а метод `getRotation()` повертає цей поворот та обнуляє його (рисунок 3.6)

```

void Imu::update_cube_rotation() {
    static int current_face = 0;
    int new_face = 0;

    if (fabs(ax) > fabs(ay) && fabs(ax) > fabs(az)) {
        new_face = (ax > 0) ? 1 : 2;
    } else if (fabs(ay) > fabs(ax) && fabs(ay) > fabs(az)) {
        new_face = (ay > 0) ? 3 : 4;
    } else {
        new_face = (az > 0) ? 5 : 6;
    }

    if (current_face == 0) {
        current_face = new_face;
    }
    if (new_face != current_face) {
        if ((current_face == 1 && new_face == 2) || (current_face == 2 && new_face == 1) ||
            (current_face == 3 && new_face == 4) || (current_face == 4 && new_face == 3) ||
            (current_face == 5 && new_face == 6) || (current_face == 6 && new_face == 5)) {
            rotation = 180;
        } else {
            rotation = 90;
        }
        current_face = new_face;
    }
}

```

Рисунок 3.6 - Метод update\_cube\_rotation() класу Imu

У головній функції (рисунок 3.7) програми спочатку створюється об'єкт класу Imu для роботи з датчиком BMI160, вказуючи його адресу, порт I2C та піни SDA та SCL. Після цього викликається метод init() для ініціалізації датчика. Якщо ініціалізація не вдалася, виводиться помилка та програма припиняє свою роботу.

Після ініціалізації датчика BMI160 викликається ініціалізація NVS (Non-Volatile Storage) для зберігання параметрів конфігурації. Якщо ініціалізація NVS виявиться невдалим через нестачу вільних сторінок або знайдену нову версію, вона буде очищена і спроба ініціалізації буде повторена. Якщо під час ініціалізації виникають помилки, програма завершується з помилкою.

Далі створюються об'єкти класів WifiClient та MqttClient для підключення до мережі Wi-Fi та відправки даних по протоколу MQTT відповідно. Якщо ініціалізація Wi-Fi або MQTT виявиться невдалим, виводиться відповідне повідомлення та програма припиняє свою роботу.

У циклі while(true) викликається метод update() об'єкта imu для оновлення даних з датчика. Після цього перевіряється значення повороту куба, яке отримано від методу getRotation(). Якщо значення не дорівнює нулю, створюються дані для

відправлення по протоколу MQTT: тема і дані, що містять інформацію про поворот куба. Дані надсилаються за допомогою методу `publish()` об'єкта `mqttClient`, а також виводиться повідомлення про поворот куба в журнал.

```
extern "C" void app_main()
{
    Imu imu(BMI160_SENSOR_ADDR, I2C_NUM_0, (gpio_num_t)21, (gpio_num_t)22);

    if (imu.init() != ESP_OK) {
        ESP_LOGE("MAIN", "Failed to initialize IMU");
        return;
    }
    esp_err_t ret = nvs_flash_init();
    if (ret == ESP_ERR_NVS_NO_FREE_PAGES || ret == ESP_ERR_NVS_NEW_VERSION_FOUND) {
        ESP_ERROR_CHECK(nvs_flash_erase());
        ret = nvs_flash_init();
    }
    ESP_ERROR_CHECK(ret);
    WifiClient wifiClient("AirPortMyHome", "passwordismyhome");
    if (wifiClient.init() != ESP_OK) {
        ESP_LOGE("MAIN", "Failed to initialize WiFi");
        return;
    }

    MqttClient mqttClient("mqtt://test.mosquitto.org", "1883", NULL, NULL);

    while (true) {
        imu.update();

        uint8_t rotation = imu.getRotation();
        if (rotation != 0) {
            char topic[50];
            char data[10];
            snprintf(topic, sizeof(topic), "cube/rotation");
            snprintf(data, sizeof(data), "%d", rotation);

            mqttClient.publish(topic, data);

            ESP_LOGI("cube", "Rotated %d degrees. Published to MQTT.", rotation);
        }
    }
}
```

Рисунок 3.7 – Функція `app_main()`

### 3.3 Тестування та налагодження системи

Після розробки програмного забезпечення та підключення до мережі Wi-Fi та MQTT брокера, система пройшла ряд тестів та процедур налагодження.

Під час тестування перевірялася коректність зчитування даних з інерціального датчика BMP160, правильність визначення поворотів куба та відправка відповідних даних на MQTT брокер. Усі основні функції системи були протестовані та перевірені на працездатність.

В ході тестування були виявлені деякі помилки та недоліки у роботі системи, такі як некоректне зчитування даних з датчика або проблеми зі з'єднанням до мережі Wi-Fi та MQTT брокера. Ці помилки були вирішені шляхом аналізу та виправлення програмного коду.

Після виправлення помилок програмне забезпечення було оптимізоване для покращення продуктивності та ефективності роботи системи. Було внесено зміни в алгоритми обробки даних, зменшено споживання ресурсів та підвищено швидкість роботи.

Після завершення налагодження система була протестована на різних умовах реального середовища. Було перевірено стабільність та надійність роботи системи, а також її здатність до коректної реакції на різні сценарії використання.

Під час тестування системи також звернули увагу на взаємодію з зовнішніми пристроями та іншими системами. Зокрема, перевірялася сумісність інтерфейсів, таких як MQTT, з різними клієнтами та брокерами, а також реакція системи на вхідні дані та події зовнішніх пристроїв.

Додатково, було проведено тестування системи на відновлення після втрати зв'язку з мережею Wi-Fi або MQTT брокером. Це включало в себе сценарії відновлення після тимчасової втрати зв'язку або відключення пристрою від живлення.

Налаштування параметрів та конфігурація системи також були важливою частиною тестування. Перевірялися різні комбінації налаштувань для оптимальної роботи системи в різних умовах середовища.

Усі виявлені проблеми та питання були документовані і вирішені перед релізом системи. Тестування було завершено успішно, і система готова до ефективного використання в реальних умовах експлуатації.

У випадку виникнення нових вимог або змін в середовищі, система готова до масштабування та адаптації. Тестування та налагодження системи виконались з урахуванням цих можливостей, щоб забезпечити стабільну та ефективну роботу в майбутньому.

Додатково було проведено тестування інтеграції системи з іншими компонентами та сервісами. Це включало в себе перевірку сумісності з іншими пристроями, з якими система може взаємодіяти, а також інтеграцію зі сторонніми сервісами, наприклад, іншими платформами IoT або системами керування будинком.

Також в ході тестування було оцінено ефективність та продуктивність системи у реальних умовах експлуатації. Було виміряно час реакції системи на зміни у вхідних даних та навантаження, а також перевірено її здатність до обробки великого обсягу даних без втрати продуктивності.

Були виявлені та виправлені всі виявлені проблеми, щоб забезпечити найвищу якість та надійність роботи системи в майбутньому.

## ВИСНОВКИ

Дослідження на тему вбудованої системи керування портативними елементами Інтернету речей (IoT) показало, що ефективне та надійне керування портативними IoT пристроями вимагає комплексного підходу до проектування, розробки і впровадження системи. Важливими аспектами є вибір надійного апаратного забезпечення, забезпечення безперебійного підключення, реалізація заходів безпеки, а також регулярне технічне обслуговування. Особливу увагу слід приділити тестуванню всіх компонентів системи для виявлення потенційних проблем і забезпечення їх безперебійної роботи.

Вбудовані системи керування повинні бути адаптивними і здатними інтегруватися з різноманітними мережевими технологіями для забезпечення стабільного зв'язку навіть у віддалених районах. Крім того, необхідно впроваджувати ефективні методи обробки та аналізу даних, щоб забезпечити точність та актуальність отримуваної інформації.

В ході дослідження було реалізовано прототип вбудованої системи керування портативними IoT пристроями.

Розроблено програмне забезпечення для управління та моніторингу стану портативних IoT пристроїв.

Досліджено ефективність та надійність запропонованої системи.

Візуалізовано результати моніторингу та керування портативними елементами IoT.

Спроектовано архітектуру вбудованої системи керування з урахуванням безпеки та енергоефективності.

Запропоновано рекомендації щодо вдосконалення та подальшого розвитку системи.

На основі дослідження можемо зробити висновок, що успішне впровадження вбудованої системи керування портативними елементами IoT сприятиме підвищенню ефективності та функціональності цих пристроїв, забезпечуючи їх надійність і довговічність. Це, в свою чергу, дозволить максимально використати

потенціал IoT технологій у різних сферах, від розумного дому до промислових додатків.

Розроблену систему можна використовувати для ефективного керування та моніторингу портативних елементів Інтернету речей (IoT) у різних сферах. Зокрема, у розумних будинках система може керувати такими пристроями, як розумні термостати, освітлення, безпекові камери та побутові прилади, забезпечуючи підвищений комфорт і безпеку мешканців. У промисловому секторі система може застосовуватись для моніторингу стану обладнання, керування роботами та дронами, оптимізації виробничих процесів, а також для запобігання аваріям через своєчасне виявлення несправностей. В аграрній галузі система дозволяє автоматизувати процеси зрошення, моніторинг стану ґрунтів і посівів, що сприяє підвищенню врожайності та ефективності використання ресурсів. Крім того, система може бути використана у сфері охорони здоров'я для віддаленого моніторингу стану пацієнтів, управління медичними приладами та збору даних для покращення якості медичних послуг. Загалом, розроблена вбудована система керування портативними елементами IoT є універсальним інструментом, що підвищує ефективність, надійність і функціональність пристроїв у різних галузях, сприяючи розвитку технологічно просунутого та більш зручного середовища.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is iot (internet of things) and how does it work? *Tech Target | IoT Agenda*. URL: <https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT> (дата звернення: 28.04.2024).
2. Kenton W. What is the internet of things (iot)? How it works and benefits. *Investopedia*. URL: <https://www.investopedia.com/terms/i/internet-things.asp> (дата звернення: 28.04.2024).
3. Understanding IoT: How the Internet of Things connects our world. *Resco*. URL: <https://www.resco.net/learning/internet-of-things/> (дата звернення: 28.04.2024).
4. Introduction to internet of things (iot) *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/introduction-to-internet-of-things-iot-set-1/> (дата звернення: 28.04.2024).
5. Unpacking iot architecture: layers and components explained *Device Authority*. URL: <https://deviceauthority.com/unpacking-iot-architecture-layers-and-components-explained/> (дата звернення: 28.04.2024).
6. Digitalux. What are the key components of an iot ecosystem? *LinkedIn*. URL: <https://www.linkedin.com/pulse/what-key-components-iot-ecosystem-digitaluxpk-pporf/> (дата звернення: 28.04.2024).
7. The 7 main components of the IoT ecosystem *Tesswave*. URL: <https://www.tesswave.com/components-of-the-iot-ecosystem/> (дата звернення: 28.04.2024).
8. Датчики. Класифікація датчиків. Використання датчиків *Corelamps*. URL: <https://corelamps.com/elektromontazhne-obladnannia/datchyky/> (дата звернення: 28.04.2024).
9. IoT connectivity: types & how to choose the right technology. *Xyte*. URL: <https://www.xyte.io/blog/iot-connectivity> (дата звернення: 29.04.2024).
10. Important areas for IoT. *MinnaLearn*. URL: <https://courses.minnalearn.com/en/courses/emerging-technologies/the-internet-of-things/important-areas-for-iot/> (дата звернення: 29.04.2024).

11. Applications of internet of things (iot). *RF Page*.  
URL: <https://www.rfpage.com/applications-of-internet-of-things-iot/> (дата звернення: 29.04.2024).

12. Chauhan Y. Application areas of iot! *Medium*.  
URL: <https://yogeshchauhan09.medium.com/application-areas-of-iot-24c784223b00> (дата звернення: 29.04.2024).

13. Internet of things applications *javatpoint*  
URL: <https://www.javatpoint.com/internet-of-things-applications> (дата звернення: 29.04.2024).

14. IoT applications | 8 most important area of application in 2021. *EDUCBA*.  
URL: <https://www.educba.com/iot-applications/> (дата звернення: 29.04.2024).

15. Applications of iot: all you need to know. *SabPaisa*  
URL: <https://sabpaisa.in/blog/applications-of-iot/> (дата звернення: 29.04.2024).

16. Duggal N. What are iot devices? Definition, types, and 5 most popular ones for 2024 *Simplilearn* URL: <https://www.simplilearn.com/iot-devices-article> (дата звернення: 29.04.2024).

17. 23 iot devices connecting the world *Built In*.  
URL: <https://builtin.com/articles/iot-devices> (дата звернення: 29.04.2024).

18. What is an iot device? | learn how iot devices work *Software AG*.  
URL: [https://www.softwareag.com/en\\_corporate/resources/iot/article/iot-device.html](https://www.softwareag.com/en_corporate/resources/iot/article/iot-device.html) (дата звернення: 29.04.2024).

19. Why iot projects fail & how to beat the odds *Software AG*.  
URL: [https://www.softwareag.com/en\\_corporate/resources/iot/wp/why-iot-projects-fail.html](https://www.softwareag.com/en_corporate/resources/iot/wp/why-iot-projects-fail.html) (дата звернення: 30.04.2024).

20. ICTs and energy efficiency. *ITU*.  
URL: [https://www.itu.int/en/action/environment-and-climate-change/Pages/energy\\_efficiency-BAK.aspx](https://www.itu.int/en/action/environment-and-climate-change/Pages/energy_efficiency-BAK.aspx) (дата звернення: 30.04.2024).

21. The future of battery technology. *S&P Global*  
URL: <https://www.spglobal.com/esg/s1/topic/the-future-of-battery-technology.html> (дата звернення: 30.04.2024).