

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

ПОГОДЖЕНО

**Декан факультету (Директор ННІ)
інформаційних технологій**
(назва факультету (ННІ))

_____ Ігор Болбот
(підпис) (ім'я ПРІЗВИЩЕ)

“ ___ ” _____ 2025 р.

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

**Завідувач кафедри
комп'ютерних наук**
(назва кафедри)

_____ Белла Голуб
(підпис) (ім'я ПРІЗВИЩЕ)

“ ___ ” _____ 2025 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему Система контролю розумним будинком з використанням Mesh-технологій

Спеціальність _____ 122 "Комп'ютерні науки" _____
(код і найменування)

Освітня програма Інформаційні управляючі системи та технології
(назва)

Орієнтація освітньої програми _____ Освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

_____ к.т.н., доцент
(науковий ступінь та вчене звання)

_____ Белла Голуб
(підпис) (ім'я ПРІЗВИЩЕ)

Керівник магістерської кваліфікаційної роботи

_____ д.т.н., проф.
(науковий ступінь та вчене звання)

_____ Микола Цюцюра
(підпис) (ім'я ПРІЗВИЩЕ)

Виконав

_____ (підпис)

_____ Олександр Каленіченко
(ім'я ПРІЗВИЩЕ здобувача)

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

К.Т.Н., доцент Белла Голуб
(науковий ступінь, вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)

“1” листопада 2024 року

ЗАВДАННЯ

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ ЗДОБУВАЧУ

Каленіченко Олександр Олександрович

(прізвище, ім'я, по батькові)

Спеціальність 122 «Комп'ютерні науки»

(код і найменування)

Освітня програма Інформаційні управляючі системи та технології

(назва)

Орієнтація освітньої програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Тема магістерської кваліфікаційної роботи «Система контролю розумним будинком з використанням Mesh-технологій»

затверджена наказом від “01” листопада 2024р. №1964 «С»

Термін подання завершеної роботи на кафедру 1 грудня 2025

(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи: Теоретичні основи: наукові джерела про Mesh-мережі та Інтернет речей (IoT). Технічна база: документація на мікроконтролери (ESP32), давачі температури, вологості та виконавчі пристрої (реле світла, опалення, замка). Нормативна база: міжнародні стандарти бездротового зв'язку. Методи: системний аналіз і проєктування програмно-апаратного комплексу.

Перелік питань, що підлягають дослідженню:

1. Аналіз теоретичних основ, сучасних тенденцій та особливостей застосування Mesh-технологій у системах автоматизації розумного будинку.
2. Розробка програмного забезпечення для вузлів Mesh-мережі та реалізація інтерфейсу користувача для віддаленого керування системою.
3. Експериментальне дослідження функціональності, надійності та ефективності розробленої системи в умовах тестового середовища.

Перелік графічного матеріалу (за потреби) _____

Дата видачі завдання “1” листопада 2024 р.

Керівник магістерської кваліфікаційної роботи

Микола Цюцюра
(підпис) (ім'я ПРІЗВИЩЕ)

Завдання прийняв до виконання _____

Олександр Каленіченко
(ім'я ПРІЗВИЩЕ)

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів виконання магістерської кваліфікаційної роботи | Строк виконання етапу магістерської кваліфікаційної роботи | Примітка |
|----------|--|---|-----------------|
| 1 | Видача завдання | 01.11.2024 | Виконано |
| 2 | Аналіз предметної області | 02.11.2024-24.11.2024 | Виконано |
| 3 | Проектування системи | 26.11.-31.12.2024 | Виконано |
| 4 | Розробка системи | 01.02.-30.04.2025 | Виконано |
| 5 | Аналіз результатів | 01.05-16.07.2025 | Виконано |
| 6 | Оформлення записки | 16.07-12.11.2025 | Виконано |
| 7 | Оформлення постеру | 05.10-18.10.2025 | Виконано |
| 8 | Написання тез до постеру | 18.10-27.10.2025 | Виконано |
| 9 | Постерна сесія | 28.10-29.10.2025 | Виконано |
| 10 | Перевірка на плагіат | 14.10.2025 | Виконано |
| 11 | Попередній захист | 24.11-29.11.2025 | Виконано |
| 12 | Захист роботи | 05.12-13.12.2025 | Виконано |

Здобувач _____

Керівник роботи _____

РЕФЕРАТ

Магістерська робота: 58 с., 25 рис., 8 табл., 1 додаток, 25 джерел.

Об'єкт дослідження – процес організації та функціонування систем контролю в межах концепції «розумного будинку».

Методи дослідження – аналіз наукових джерел та технічних стандартів, системний аналіз для формування вимог, імітаційне та об'єктно-орієнтоване моделювання, практичні експерименти з використанням апаратних засобів у симуляційному середовищі Tinkercad, тестування алгоритмів та оцінка їх ефективності.

Мета роботи – підвищення ефективності системи контролю «розумним будинком», шляхом використання Mesh-мереж, що забезпечують підвищену надійність, масштабованість і гнучкість роботи

Наукова новизна полягає у удосконаленні архітектури та алгоритмічних рішень для системи контролю «розумним будинком» з підвищеною відмовостійкістю за рахунок використання Mesh-технологій. Вперше запропоновано інтеграцію узагальненої моделі системи з механізмами самовідновлення Mesh-мережі, що дозволяє системі адаптуватися до змін у структурі мережі без втрати функціональності. У результаті роботи було розроблено архітектуру системи, створено її функціональні та об'єктно-орієнтовані моделі, зібрано прототип апаратної частини у середовищі Tinkercad та розроблено веб-інтерфейс для керування та моніторингу. Експериментально доведено працездатність системи та ефективність Mesh-топології у порівнянні з традиційними архітектурами.

Отримані результати доводять практичну доцільність впровадження систем контролю «розумного будинку» на основі Mesh-технологій. Розроблена архітектура та прототип можуть слугувати основою для створення комерційних продуктів, а також використовуватися в навчальних проєктах як приклад інтеграції сучасних мережових рішень у побутові системи автоматизації.

Ключові слова: РОЗУМНИЙ БУДИНОК, MESH-МЕРЕЖА, ІОТ, СИСТЕМА КОНТРОЛЮ, МОДЕЛЮВАННЯ, ВЕБ-ІНТЕРФЕЙС.

ЗМІСТ

| | |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ | 4 |
| ВСТУП..... | 5 |
| РОЗДІЛ 1. АНАЛІЗ СИСТЕМ КОНТРОЛЮ РОЗУМНИМ БУДИНКОМ..... | 7 |
| 1.1 Концепція і розвиток систем розумного будинку..... | 7 |
| 1.2 Технології управління розумним будинком..... | 10 |
| 1.3 Особливості Mesh-мереж у системах контролю | 13 |
| 1.4 Аналіз існуючих рішень для розумного будинку | 16 |
| 1.5 Постановка задачі для системи контролю на основі Mesh | 19 |
| РОЗДІЛ 2. МОДЕЛЮВАННЯ СИСТЕМИ КОНТРОЛЮ РОЗУМНИМ БУДИНКОМ | 21 |
| 2.1 Функціональні вимоги до системи контролю | 21 |
| 2.2 Моделювання сценаріїв роботи системи..... | 23 |
| 2.3 Функціональне та об'єктно-орієнтоване моделювання | 25 |
| 2.4 Узагальнена модель системи контролю | 28 |
| РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ КОНТРОЛЮ РОЗУМНИМ БУДИНКОМ З ВИКОРИСТАННЯМ MESH-ТЕХНОЛОГІЙ | 31 |
| 3.1 Архітектура системи контролю | 31 |
| 3.2 Апаратна частина системи | 33 |
| 3.3 Програмне забезпечення і алгоритми управління..... | 35 |
| 3.4 Маршрутизація і самовідновлення Mesh-мережі..... | 37 |
| 3.5 Інтерфейс користувача системи..... | 39 |
| РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА СИСТЕМИ КОНТРОЛЮ | 41 |
| 4.1 Апаратні та програмні вимоги для роботи системи..... | 41 |
| 4.2 Тестування сценаріїв керування розумним будинком..... | 43 |
| 4.3 Оцінка ефективності Mesh-технологій у системі контролю..... | 45 |
| ВИСНОВКИ | 48 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 51 |
| ДОДАТКИ | 54 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AODV – Ad hoc On-Demand Distance Vector

ESP32 – Microcontroller with Wi-Fi and Bluetooth Support

ESP8266 – Microcontroller with Wi-Fi Support

HWMP – Hybrid Wireless Mesh Protocol

IDEF0 – Integration Definition for Function Modeling

IoT – Internet of Things

KNX – Konnex International Standard

Mesh – Mesh Network

OLSR – Optimized Link State Routing

UML – Unified Modeling Language

Wi-Fi – Wireless Fidelity

ZigBee – ZigBee Communication Protocol

Z-Wave – Z-Wave Communication Protocol

ВСТУП

Актуальність роботи. У сучасному світі швидкими темпами розвиваються інформаційні технології, що безпосередньо впливають на якість життя людини. Однією з найперспективніших сфер є системи «розумного будинку», які дозволяють автоматизувати керування побутовими процесами, підвищити безпеку та забезпечити раціональне використання ресурсів. Водночас традиційні централізовані рішення мають низку проблем: високу вартість розгортання, складність масштабування, обмежену відмовостійкість і залежність від єдиного контролера. Саме тому актуальним стає застосування децентралізованих мережевих підходів, зокрема Mesh-технологій, які дають змогу будувати гнучкі й самовідновлювані системи контролю.

Об'єкт дослідження – процес організації та функціонування систем контролю в межах концепції «розумного будинку».

Предмет дослідження – методи та засоби застосування Mesh-технологій у побудові систем керування пристроями «розумного будинку».

Мета дослідження полягає у підвищенні ефективності системи контролю «розумним будинком», шляхом використання Mesh-мереж, що забезпечують підвищену надійність, масштабованість і гнучкість роботи

Відповідно до поставленої мети потрібно виконати наступні **завдання**:

- виконати аналіз існуючих систем контролю «розумного будинку», визначити їхні переваги та недоліки, а також обґрунтувати постановку задачі для системи на основі Mesh;
- сформулювати функціональні вимоги до майбутньої системи та провести моделювання сценаріїв її роботи;
- побудувати функціональні та об'єктно-орієнтовані моделі, а також узагальнену модель системи контролю;
- розробити архітектуру системи з описом апаратної частини, програмного забезпечення й алгоритмів управління;
- дослідити особливості маршрутизації та механізми самовідновлення Mesh-мереж у системах контролю;
- створити та описати інтерфейс користувача системи;

- визначити апаратні та програмні вимоги для роботи системи, здійснити тестування сценаріїв керування «розумним будинком» та оцінити ефективність застосування Mesh-технологій;
- надати рекомендації щодо впровадження та можливостей масштабування розробленої системи.

У дослідженні застосовуються наступні методи: аналіз наукових джерел та технічних стандартів; методи системного аналізу для формування вимог; імітаційне та об'єктно-орієнтоване моделювання; практичні експерименти з використанням апаратних засобів; тестування алгоритмів маршрутизації та оцінка їх ефективності.

Наукова новизна роботи полягає у використанні Mesh-технологій для побудови систем контролю «розумним будинком», що дозволило розробити архітектуру і алгоритмічні рішення з підвищеною відмовостійкістю та гнучкістю. Уперше було запропоновано інтеграцію узагальненої моделі з механізмами самовідновлення Mesh-мережі, що дає змогу адаптувати систему до змін структури мережі без втрати функціональності.

Структура магістерської роботи. Робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг становить 58 сторінок друкованого тексту. У першому розділі наведено аналіз сучасних систем контролю «розумним будинком» та визначено постановку задачі. Другий розділ присвячено моделюванню системи контролю та формуванню її вимог. У третьому розділі описано архітектуру, апаратні та програмні компоненти, алгоритми маршрутизації та інтерфейс користувача. У четвертому розділі здійснено експериментальну перевірку, тестування та оцінку ефективності Mesh-рішень, а також подано рекомендації щодо впровадження. У висновках узагальнено результати дослідження.

РОЗДІЛ 1. АНАЛІЗ СИСТЕМ КОНТРОЛЮ РОЗУМНИМ БУДИНКОМ

1.1 Концепція і розвиток систем розумного будинку

Ідея створення так званого «розумного будинку» виникла ще у другій половині ХХ століття, коли почали активно розвиватися електроніка та комп'ютерні технології. Перші спроби автоматизувати окремі функції житлових будівель зводилися до використання простих датчиків руху, систем охоронної сигналізації чи автоматичних вимикачів світла. Проте саме з появою персональних комп'ютерів та мережевих технологій стало можливим говорити про створення комплексних систем управління житловим простором [1].

У 1960–1970-х роках з'явилися перші лабораторні дослідження з інтеграції побутових приладів у єдину мережу. Один із таких проектів був реалізований Массачусетським технологічним інститутом, де було створено так звану «інтелектуальну кімнату». Вона могла реагувати на рух людини, а також використовувала системи розпізнавання голосу для керування деякими пристроями. У 1980–1990-х роках розвиток мікропроцесорів та комунікаційних протоколів дозволив почати комерціалізацію таких рішень. Саме тоді сформувалося поняття «Smart Home», яке швидко набуло поширення в Європі та США [2].

Сучасна концепція «розумного будинку» визначається як інтегрована система, що забезпечує автоматизацію управління освітленням, опаленням, вентиляцією, кондиціонуванням повітря, охоронними комплексами, мультимедійними пристроями та іншими інженерними системами. Головна ідея полягає у створенні єдиного інформаційного середовища, яке реагує на потреби мешканців і може працювати в автономному режимі.

Розвиток концепції «розумного будинку» можна умовно поділити на кілька етапів. На початковому етапі мова йшла про автоматизацію окремих функцій: наприклад, автоматичне вмикання світла чи регулювання температури. Наступним етапом стало використання централізованих контролерів, що дозволяли керувати усіма системами з одного пульта чи комп'ютера. Сьогодні ключовою тенденцією є інтеграція пристроїв у глобальні мережі через Інтернет речей (IoT), а також перехід

від централізованих до децентралізованих рішень, де окремі пристрої можуть взаємодіяти між собою без потреби у центральному сервері (рис. 1.1).

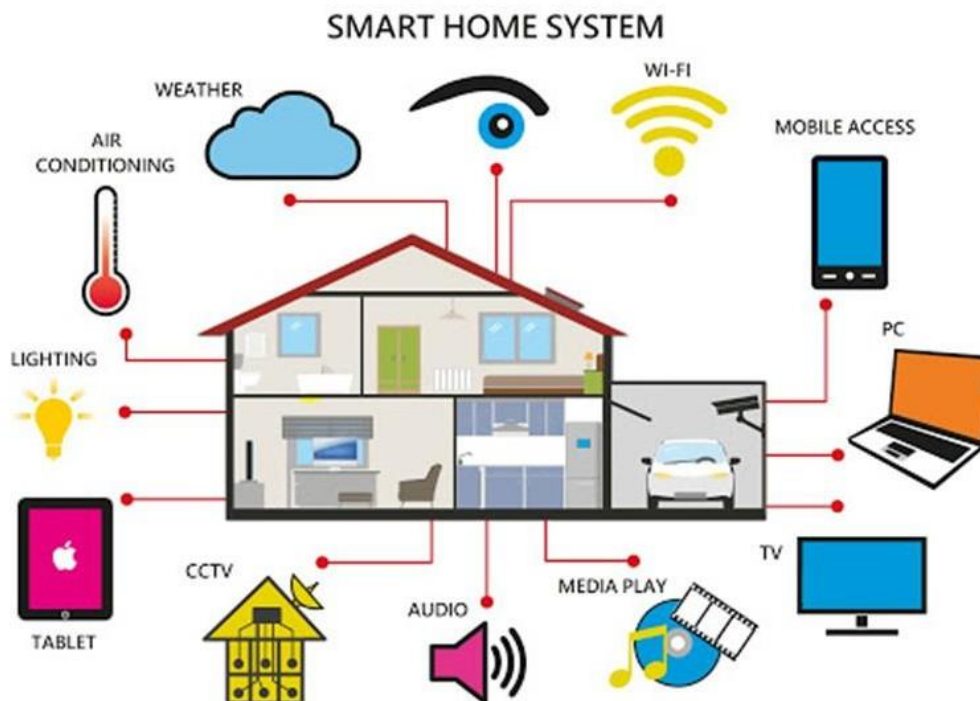


Рис.1.1 Приклад IoT системи «розумного будинку»

Варто підкреслити, що розвиток «розумних будинків» безпосередньо пов'язаний із загальним трендом цифровізації. За прогнозами міжнародних аналітичних агентств, кількість пристроїв IoT у світі перевищить 25 мільярдів вже у найближчі роки. Це означає, що практично кожен побутовий прилад може стати частиною інтелектуального середовища [3].

Актуальність концепції визначається і соціальними, і економічними чинниками. З одного боку, люди прагнуть комфорту та безпеки, з іншого – зростає потреба в ощадливому використанні енергоресурсів. Наприклад, системи автоматичного регулювання температури дозволяють суттєво знизити витрати на опалення, а системи контролю електроспоживання – зменшити рахунки за електроенергію.

У табл. 1.1. подана коротка характеристика основних етапів розвитку концепції «розумного будинку».

Табл. 1.1 – Етапи розвитку концепції «розумного будинку»

| Етап розвитку | Характерні особливості | Приклади рішень |
|---------------|--|--|
| 1960–1970-ті | Перші дослідження з автоматизації, лабораторні проекти | «Інтелектуальна кімната» МІТ |
| 1980–1990-ті | Поширення мікропроцесорів, поява централізованих контролерів | Системи охоронної сигналізації, «розумне» освітлення |
| 2000–2010-ті | Інтеграція з комп'ютерами та мобільними пристроями | Домашні мультимедійні центри, керування через ПК |
| 2010–2020-ті | Використання IoT та мобільних додатків | Amazon Alexa, Google Home, Apple HomeKit |
| 2020+ | Децентралізовані мережі, Mesh-технології | Самоорганізовані системи керування пристроями |

Як видно з таблиці, поступовий розвиток від окремих рішень до комплексних інтегрованих систем свідчить про постійне розширення можливостей «розумних будинків».

Важливим моментом поширення цієї концепції стали також зміни у структурі міського середовища. Якщо на початку розвитку йшлося про будівлі бізнес-класу чи елітне житло, то сьогодні технології автоматизації активно впроваджуються у звичайні квартири та приватні будинки. Зменшення вартості датчиків і мікроконтролерів робить такі рішення доступними для широкого кола користувачів [4].

На сучасному етапі розвиток концепції «розумного будинку» тісно пов'язаний із такими напрямками, як штучний інтелект, машинне навчання та хмарні обчислення. Наприклад, системи на основі штучного інтелекту здатні аналізувати поведінку мешканців і пропонувати оптимальні сценарії використання енергії. Використання хмарних сервісів дозволяє організувати віддалений доступ і контроль з будь-якої точки світу.

У роботах [1 – 10] підкреслюється, що розвиток «розумного будинку» неможливий без постійного вдосконалення протоколів обміну даними та мережеских рішень. Адже саме від швидкості, надійності та безпеки передавання інформації залежить ефективність усієї системи. Саме тому в наступних підрозділах буде детальніше розглянуто існуючі технології управління,

особливості Mesh-мереж, а також сучасні приклади впровадження подібних рішень.

Концепція «розумного будинку» пройшла тривалий шлях від лабораторних експериментів до широкого комерційного використання. Сьогодні вона є не лише символом технічного прогресу, але й реальною потребою, яка сприяє економії ресурсів, підвищенню безпеки та комфорту. А застосування Mesh-технологій у цій сфері відкриває нові горизонти розвитку, забезпечуючи стабільність, масштабованість і доступність для кінцевого користувача.

1.2 Технології управління розумним будинком

У системах «розумного будинку» ключовим елементом є технології, які забезпечують взаємодію між різними пристроями, сенсорами та виконавчими механізмами. Від того, наскільки ефективно реалізовані ці технології, залежить не лише рівень комфорту, але й безпека, енергоефективність та масштабованість системи.

Спершу управління будинком базувалося на дротових технологіях. Для цього прокладалися кабельні мережі, через які контролери обмінювалися інформацією з обладнанням. Такий підхід мав перевагу у надійності, але був дорогим і незручним для модернізації. Будь-яке оновлення вимагало прокладання нових кабелів, що суттєво збільшувало вартість інсталяції.

Поступово почали розвиватися бездротові стандарти. Вони дозволили відмовитися від складної кабельної інфраструктури та забезпечили більшу гнучкість. Наприклад, такі технології як Wi-Fi чи Bluetooth дали можливість інтегрувати побутові пристрої у спільну мережу. Водночас вони мали й обмеження. Wi-Fi забезпечує високу швидкість передачі даних, але споживає багато енергії, що є проблемою для малопотужних сенсорів. Bluetooth добре підходить для з'єднання пристроїв на невеликих відстанях, але не завжди ефективний у масштабних мережах.

У 2000-х роках з'явилися спеціалізовані протоколи для систем автоматизації. Найбільш поширеними стали ZigBee, Z-Wave, KNX. ZigBee використовує малу

потужність і добре працює у мережах з великою кількістю вузлів, проте має обмежену швидкість передачі даних. Z-Wave відомий своєю надійністю та стабільністю, але він менш поширений і потребує сертифікованого обладнання. KNX більше застосовується у комерційних будівлях і вимагає складнішої інсталяції [11].

Сьогодні управління «розумним будинком» тісно пов'язане з концепцією Інтернету речей (IoT). Кожен пристрій стає частиною глобальної мережі, де він може обмінюватися даними не тільки з іншими вузлами системи, а й з хмарними сервісами. Завдяки цьому користувачі отримують можливість контролювати будинок із будь-якої точки світу, використовуючи смартфон або веб-інтерфейс. У табл. 1.2 надано порівняння основних технологій.

Табл. 1.2 – Порівняння технологій управління «розумним будинком»

| Технологія | Основні переваги | Обмеження |
|-------------------------|--|---|
| Дротові (Ethernet, KNX) | Висока надійність, швидкість | Висока вартість, складність монтажу |
| Wi-Fi | Швидка передача даних, простота інтеграції | Високе енергоспоживання, обмежена масштабованість |
| Bluetooth | Простота з'єднання, низька вартість | Невеликий радіус дії, не підходить для великих систем |
| ZigBee | Низьке енергоспоживання, підтримка великих мереж | Низька швидкість, складність налаштування |
| Z-Wave | Стабільність, відмовостійкість | Дорожче обладнання, обмежена поширеність |

З розвитком штучного інтелекту й аналітики даних системи управління виходять на новий рівень. Сучасні «розумні будинки» здатні не лише реагувати на події, але й передбачати їх. Наприклад, система може аналізувати щоденні звички мешканців і автоматично налаштовувати освітлення, температуру чи безпекові параметри. Це дозволяє зменшити енергоспоживання і підвищити рівень комфорту.

У багатьох наукових роботах [3 - 15] зазначається, що наступним кроком розвитку стане перехід до децентралізованих архітектур, де ключову роль відіграватимуть Mesh-мережі. Вони дозволять уникнути залежності від одного

центрального вузла і створять більш стійку систему, що здатна самостійно відновлюватися у випадку відмови окремих пристроїв.

На рис. 1.2 навести дві мережі, де показано відмінності між централізованою та Mesh-архітектурами.

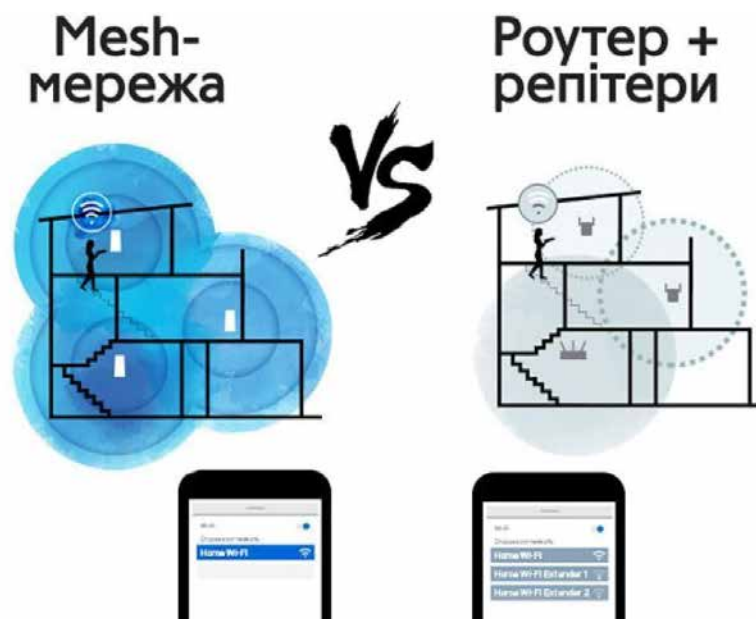


Рис. 1.2 Порівняння mesh мережі з мережею роутер+репітери

Технології управління «розумним будинком» пройшли довгий шлях від дротових рішень до інтеграції у глобальні IoT-платформи. Водночас актуальною лишається проблема підвищення надійності та зменшення енергоспоживання, що й обумовлює необхідність впровадження Mesh-рішень [16].

1.3 Особливості Mesh-мереж у системах контролю

Сучасні системи «розумного будинку» потребують не лише функціональності, а й високої надійності та здатності до масштабування. Традиційні централізовані або навіть зіркоподібні топології мають серйозні обмеження. Якщо виходить з ладу центральний вузол, система може втратити працездатність. Тому останніми роками дедалі більше уваги приділяється децентралізованим архітектурам, серед яких особливе місце займають Mesh-мережі.

Mesh-топологія передбачає, що кожен вузол у мережі не лише виконує свою функцію (наприклад, збір даних від датчика чи керування виконавчим пристроєм), а й здатен передавати інформацію іншим вузлам. Таким чином формується розподілена структура, де немає жорсткої залежності від одного центру. Якщо один із вузлів виходить з ладу, система автоматично знаходить інші маршрути для передачі даних [17-20].

Ця властивість робить Mesh особливо придатною для систем контролю «розумного будинку», де часто використовується велика кількість пристроїв, розташованих у різних кімнатах чи навіть у кількох будівлях (рис.1.3).



Рис. 1.3 Приклад топології Mesh-мережі в системі «розумного будинку»

Mesh-мережі мають кілька ключових переваг, які особливо важливі для розумних будинків:

1. Відмовостійкість – втрата окремого вузла не призводить до зупинки всієї мережі. Інформація «обходить» несправний вузол і знаходить інший шлях.
2. Масштабованість – додавання нових пристроїв не вимагає повної перебудови системи. Новий вузол автоматично інтегрується у мережу.
3. Гнучкість у розгортанні – систему можна поступово нарощувати, встановлюючи додаткові датчики чи контролери без великих витрат.
4. Низькі вимоги до центральної інфраструктури – у багатьох випадках відсутня необхідність у дорогому серверному обладнанні.
5. Самоорганізація – мережа здатна самостійно будувати маршрути та оптимізувати їх у процесі роботи.

Далі наведемо табл. 1.3 з порівнянням Mesh-мереж із традиційними топологіями.

Табл. 1.3 – Порівняння топологій мереж у системах контролю

| Топологія | Переваги | Недоліки |
|------------|--|--|
| Шинна | Простота, низька вартість | Один збій блокує всю систему |
| Зіркова | Просте керування, швидке підключення | Вихід з ладу центрального вузла паралізує роботу |
| Ієрархічна | Добра масштабованість, контроль | Висока складність управління, залежність від верхнього рівня |
| Mesh | Відмовостійкість, масштабованість, гнучкість | Вища складність алгоритмів маршрутизації |

З таблиці видно, що Mesh вигідно відрізняється своєю стійкістю до відмов і можливістю розширення, хоча це й вимагає складніших алгоритмів управління.

Важливою особливістю Mesh-мереж є алгоритми маршрутизації, які забезпечують передачу даних у багатоступеневих структурах. Особлива особлива увага приділяється протоколам AODV (Ad hoc On-Demand Distance Vector), OLSR (Optimized Link State Routing) та HWMP (Hybrid Wireless Mesh Protocol). Вони відрізняються способом пошуку та оновлення маршрутів.

AODV створює маршрути «на вимогу», тобто лише тоді, коли з'являється потреба в передаванні даних. Це дозволяє економити ресурси, але може призводити до затримок у момент побудови маршруту.

OLSR постійно підтримує таблиці маршрутів, що забезпечує швидкий доступ до будь-якого вузла, але вимагає більшої кількості службового трафіку.

HWMP є гібридним протоколом, який поєднує обидва підходи і добре підходить для систем середнього масштабу.

Вибір алгоритму маршрутизації залежить від кількості пристроїв, обсягу переданих даних і вимог до часу реакції системи.

Для «розумного будинку» характерна висока кількість сенсорів та виконавчих механізмів: датчики температури, вологості, руху, контролери освітлення, охоронні системи. Всі вони генерують невеликі пакети даних, але ці дані мають передаватися регулярно та без втрат. Саме тут Mesh показує свою ефективність [21].

Крім того, Mesh-мережі дозволяють покращити покриття у великих будинках чи на територіях приватних садиб. Якщо в класичних мережах сигнал може

втратитися через стіни чи відстань, то в Mesh дані «перестрибують» через проміжні вузли. Це підвищує стабільність роботи всієї системи (рис. 1.4).

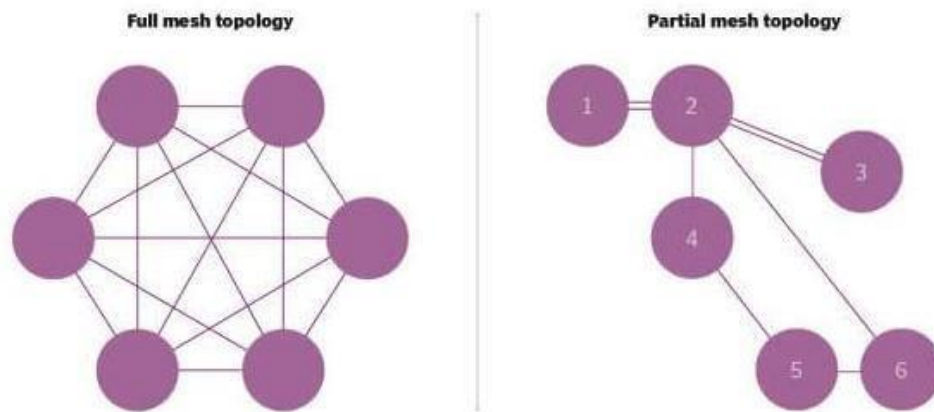


Рис. 1.4 Приклад маршрутизації даних у Mesh-мережі (повна та часткова Mesh-мережі)

Попри численні переваги, Mesh-мережі мають і певні виклики. Серед них – складність налаштування та потреба у більш потужних алгоритмах для підтримки маршрутизації. Також у великих мережах може зростати затримка передавання даних. Проте для систем «розумного будинку», де передаються відносно невеликі обсяги інформації, ці обмеження є несуттєвими [22].

Зменшення вартості бездротових модулів, таких як ESP8266, ESP32, RFM95, зробило Mesh-додатки доступними навіть для домашніх проєктів, не лише для промислових рішень. Це відкриває нові можливості для масового впровадження технології.

1.4 Аналіз існуючих рішень для розумного будинку

На сучасному етапі розвитку цифрових технологій існує велика кількість готових рішень для організації системи «розумного будинку». Вони відрізняються архітектурою, принципами роботи, вартістю та рівнем складності інсталяції. Умовно такі рішення можна поділити на комерційні (масові продукти від відомих компаній), відкриті (open-source системи, що розвиваються спільнотою) та індивідуальні розробки на основі мікроконтролерів і бездротових модулів.

Комерційні рішення є найбільш поширеними, оскільки вони пропонують готову екосистему пристроїв і програмного забезпечення. Прикладами є Google Home, Amazon Alexa, Apple HomeKit, а також європейські рішення на основі KNX чи Z-Wave. Їхньою головною перевагою є простота використання. Користувач отримує мобільний додаток, голосового асистента або веб-інтерфейс, через які можна керувати освітленням, кліматом, мультимедіа й безпекою [23].

Разом із тим такі системи мають низку недоліків. По-перше, вони часто залежать від хмарних сервісів виробника. Якщо сервери компанії недоступні, частина функцій може перестати працювати. По-друге, вони не завжди є сумісними з пристроями інших виробників. Наприклад, датчики від однієї компанії не завжди можна інтегрувати в екосистему іншої. Крім того, вартість оригінальних пристроїв є досить високою, що обмежує їх масове впровадження.

На противагу комерційним системам, існують відкриті програмні платформи, які дозволяють будувати індивідуальні рішення. Прикладами є Home Assistant (рис. 1.5), OpenHAB, Domoticz. Ці системи мають гнучку архітектуру і дозволяють підключати пристрої від різних виробників. Користувач отримує більше можливостей для налаштування сценаріїв, створення власних алгоритмів і навіть інтеграції з іншими сервісами.



Рис. 1.5 Головне вікно системи Home Assistant

Такі рішення мають і недоліки. Вони потребують технічних знань, налаштування серверів і мереж, встановлення додаткових модулів. Для звичайного користувача це може бути складним завданням. Проте у наукових роботах [3 – 8] наголошується, що саме відкриті рішення створюють основу для експериментів і розвитку нових підходів.

Окремий напрям – це індивідуальні розробки на базі мікроконтролерів, таких як Arduino, ESP8266, ESP32. Вони дозволяють будувати дешеві та гнучкі системи, які можна адаптувати під конкретні потреби. Прикладом може бути система «розумної вентиляції». Там використано набір сенсорів для моніторингу параметрів повітря та контролю виконавчих пристроїв, що демонструє реальні можливості недорогих рішень (рис. 1.6).

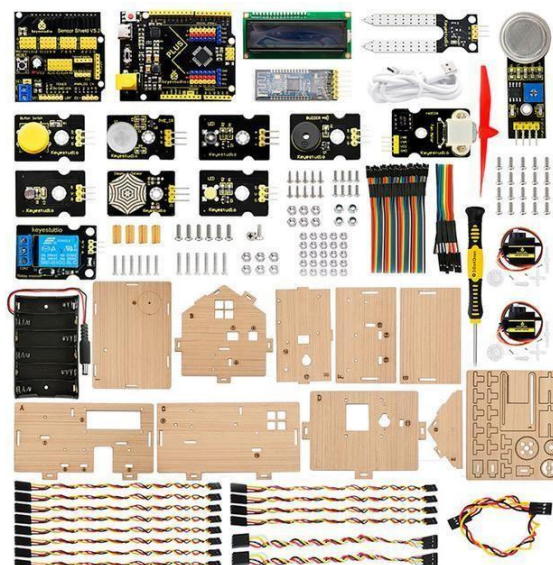


Рис. 1.6 Набір для типової системи на основі Arduino/ESP для «розумного будинку»

Подібні системи часто використовуються в університетських і науково-дослідних проектах. Вони добре підходять для навчання та відпрацювання нових алгоритмів маршрутизації, управління або аналізу даних. Однак у комерційному використанні вони вимагають доопрацювання, оскільки потребують стабільного програмного забезпечення, захисту від збоїв та належної безпеки [24].

Щоб краще побачити різницю між різними підходами, доцільно узагальнити

Табл. 1.4 – Порівняння існуючих рішень для «розумного будинку»

| Тип рішення | Переваги | Недоліки | Приклади |
|------------------------|--|---|--|
| Комерційні системи | Простота використання, інтеграція з мобільними додатками, голосове керування | Висока вартість, залежність від хмарних сервісів, обмежена сумісність | Google Home, Amazon Alexa, Apple HomeKit |
| Відкриті системи | Гнучкість, підтримка багатьох протоколів, можливість створення власних сценаріїв | Складність налаштування, потреба в технічних знаннях | Home Assistant, OpenHAB, Domoticz |
| Індивідуальні розробки | Низька вартість, адаптивність, підходять для експериментів | Недостатня стабільність, потреба у доопрацюванні | Arduino, ESP8266, ESP32 проекти |

З таблиці видно, що кожен підхід має свої переваги та обмеження. Вибір залежить від конкретних потреб користувача, бюджету і технічних можливостей.

Окремо варто звернути увагу на те, як у цих системах організована мережа. У більшості комерційних і відкритих рішень використовується централізована або зіркоподібна топологія. Це означає, що всі пристрої підключаються до одного контролера або шлюзу. Така схема проста, але вона має слабе місце – відмова центрального вузла може зупинити роботу всієї системи [25].

Індивідуальні розробки, особливо ті, що ґрунтуються на сучасних модулях зв'язку (LoRa, ESP-NOW, RFM95), вже активно експериментують із Mesh-топологією. Це дозволяє створювати системи, які здатні самостійно відновлюватися після збою, працювати на більших відстанях і мати кращу масштабованість.

Проведений аналіз показує, що існуючі рішення для організації «розумного будинку» дуже різні за своєю структурою, можливостями та вартістю. Комерційні системи забезпечують простоту, але обмежують користувача в гнучкості. Відкриті платформи дають великий простір для налаштувань, проте потребують знань. Індивідуальні розробки на базі мікроконтролерів є дешевими й перспективними, особливо у випадках використання Mesh-мереж. Саме останній напрям відкриває нові можливості для побудови стабільних і доступних систем контролю.

1.5 Постановка задачі для системи контролю на основі Mesh

Аналіз сучасних підходів показав, що більшість існуючих рішень для «розумного будинку» базуються на централізованих або зіркоподібних топологіях. Це робить систему зручною на початковому етапі, але з часом виникають проблеми. Якщо головний контролер виходить з ладу, то втрачається зв'язок з усіма пристроями. Додавання нових датчиків чи модулів часто потребує перепроєктування мережі, а стабільність роботи залежить від надійності одного центрального вузла.

Mesh-технології дозволяють вирішити ці обмеження. Завдяки децентралізованій архітектурі, кожен вузол може виконувати роль ретранслятора, передаючи дані далі по мережі. Це створює умови для відмовостійкості, кращого покриття і можливості поступового нарощування системи. Для «розумного будинку» це означає, що датчики у віддалених кімнатах або навіть на вулиці зможуть передавати інформацію без потреби у додаткових маршрутизаторах.

Враховуючи це, можна сформулювати основне завдання дослідження: розробити і дослідити систему контролю для «розумного будинку» з використанням Mesh-мереж, яка забезпечує надійну роботу, масштабованість і простоту у використанні.

Щоб досягти цієї мети, потрібно розв'язати низку конкретних задач:

1. Провести аналіз існуючих систем керування «розумним будинком», виділити їхні переваги та недоліки, особливо у контексті мережевої організації.
2. Сформулювати вимоги до системи контролю на основі Mesh, враховуючи реальні умови використання: кількість пристроїв, відстані між ними, типи даних, що передаються.
3. Побудувати модель системи контролю, де відображені основні елементи: датчики, виконавчі пристрої, маршрути передачі даних.
4. Розробити архітектуру системи з описом апаратної та програмної частини, а також передбачити алгоритми маршрутизації і самовідновлення мережі.
5. Виконати тестування та оцінити ефективність запропонованого рішення за показниками стабільності, швидкості передачі даних і відмовостійкості.

Таким чином, постановка задачі для системи контролю на основі Mesh полягає у створенні рішення, яке дозволить обійти обмеження класичних підходів, забезпечити стабільність роботи, зменшити залежність від одного вузла і зробити систему доступною для широкого кола користувачів.

РОЗДІЛ 2. МОДЕЛЮВАННЯ СИСТЕМИ КОНТРОЛЮ РОЗУМНИМ БУДИНКОМ

2.1 Функціональні вимоги до системи контролю

Щоб перейти до моделювання, необхідно спочатку визначити, якими властивостями повинна володіти система контролю «розумного будинку» на основі Mesh-мереж. Вимоги є ключовим етапом, адже саме вони формують основу архітектури та визначають, які апаратні і програмні рішення треба буде застосувати.

Функціональні вимоги описують не лише те, що система повинна робити, але і як саме вона має працювати в умовах реального використання. Якщо говорити простіше, то це відповіді на питання: «Які завдання система вирішує?», «Як вона реагує на події?», «Які результати очікує користувач?».

У першу чергу система повинна забезпечувати збір даних з великої кількості сенсорів. Це можуть бути датчики температури, вологості, освітлення, руху, диму чи витoku води. Дані мають передаватися у мережу з мінімальними затримками, щоб користувач міг бачити актуальну інформацію.

Другий напрям пов'язаний з керуванням виконавчими пристроями. Сюди відносяться освітлювальні системи, опалювальні прилади, вентиляція, електрозамки, мультимедійні комплекси. Управління повинно бути не лише ручним, через мобільний додаток чи веб-інтерфейс, а й автоматичним за заданими сценаріями.

Третій напрям – це забезпечення стабільності і відмовостійкості. На відміну від централізованих архітектур, у Mesh-мережі вихід з ладу одного вузла не повинен впливати на всю систему. Тому важливо, щоб маршрути даних могли перебудовуватися автоматично.

Не менш важливим є питання безпеки. Дані, які циркулюють у мережі «розумного будинку», можуть містити чутливу інформацію про пересування мешканців чи стан будівлі. Це означає, що повинні бути реалізовані механізми захисту від несанкціонованого доступу.

У табл. 2.1 представлені функціональні вимоги до системи контролю.

Табл. 2.1 – Функціональні вимоги до системи контролю

| Напря́м | Змі́ст вимо́г | Очі́куваний результа́т |
|----------------------|---|---|
| Збір даних | Система повинна зчитувати показники від різних сенсорів у режимі реального часу | Оперативна інформація про стан будинку |
| Керування пристроями | Можливість керування освітленням, кліматом, безпекою, мультимедіа | Зручність та автоматизація побутових процесів |
| Відмовостійкість | Автоматичне перебудування маршрутів у випадку відмови вузлів | Стабільність і безперервність роботи |
| Безпека | Захист від зовнішнього втручання, шифрування даних | Збереження конфіденційності користувача |
| Масштабованість | Підтримка підключення нових пристроїв без серйозних змін системи | Просте розширення системи |

Як видно з таблиці, вимоги охоплюють як технічну, так і функціональну сторони.

Щоб вимоги виглядали більш наочними, можна описати кілька типових прикладів. Уявімо, що користувач заходить у будинок. Датчик руху біля дверей передає сигнал у Mesh-мережу. Інформація миттєво доходить до системи, яка вмикає освітлення у коридорі та розблоковує електронний замок. Інший сценарій: температура у кімнаті впала нижче заданого рівня. Сенсор передає дані у мережу, і система вмикає опалювальний прилад. Якщо ж один вузол мережі перестав працювати, дані підуть в обхід через сусідні пристрої, і користувач цього навіть не помітить.

Окремою вимогою є простота керування для людини. Інтерфейс повинен бути інтуїтивно зрозумілим: кнопки на смартфоні чи прості голосові команди. Людина не має думати про технічні деталі. Якщо потрібно вимкнути світло у всьому будинку, це має робитися однією командою. Якщо треба подивитися показники датчиків, вони мають відображатися у зручному вигляді, наприклад у вигляді графіків чи діаграм.

Отже, функціональні вимоги до системи контролю «розумного будинку» на основі Mesh-мереж можна сформулювати як набір умов, що забезпечують

стабільну, безпечну і зручну роботу. Це збір даних від сенсорів, керування виконавчими пристроями, відмовостійкість, захист інформації, масштабованість і зручність для користувача. Саме ці вимоги ляжуть в основу подальшого моделювання сценаріїв роботи системи, що буде розглянуто у наступному підпункті.

2.2 Моделювання сценаріїв роботи системи

Щоб система «розумного будинку» на основі Mesh-мереж не була лише набором сенсорів і виконавчих пристроїв, потрібно описати її роботу у вигляді сценаріїв. Сценарій у цьому випадку – це послідовність подій, які відбуваються у мережі та викликають певні дії. Якщо говорити простіше, сценарій показує, як система поводить себе у конкретній життєвій ситуації.

Моделювання сценаріїв дозволяє ще до створення фізичного прототипу побачити, чи відповідає система вимогам користувача, чи не виникає конфліктів між діями різних пристроїв, чи правильно працює маршрутизація у Mesh-мережі.

1. Приклад сценарію «Вхід у будинок».

Уявімо ситуацію, коли мешканець повертається додому ввечері. Система повинна автоматично виконати низку дій. На вході спрацьовує датчик руху. Сигнал від нього передається найближчому вузлу Mesh-мережі, який передає інформацію далі. Алгоритм перевіряє, що користувач авторизований (наприклад, через електронний ключ або смартфон). У результаті автоматично вмикається світло у коридорі, розблоковується дверний замок і активується система опалення, якщо температура нижча за норму.

Такий сценарій демонструє відразу кілька переваг Mesh-мереж. По-перше, сигнал від датчика може йти різними шляхами, навіть якщо один із вузлів не працює. По-друге, інформація відразу доступна всій системі, а не лише центральному серверу.

2. Сценарій «Контроль безпеки».

Інший приклад стосується безпеки. Припустимо, у нічний час система переведена в охоронний режим. Датчик руху у вітальні фіксує пересування. Сигнал

надходить у мережу, і система одразу виконує кілька дій: увімкнення тривожної сирени, надсилання повідомлення на смартфон власника та блокування вхідних дверей.

Якщо при цьому один вузол мережі не передає дані (наприклад, через відсутність живлення), Mesh-архітектура забезпечує альтернативний маршрут. Це дозволяє не втратити сигнал і гарантує своєчасне реагування.

3. Сценарій «Оптимізація енергоспоживання».

Ще один сценарій можна описати у зв'язку з енергозбереженням. Вдень, коли всі мешканці на роботі, система переходить у режим мінімального споживання. Освітлення автоматично вимикається, опалення переходить у «економний» режим, електроприлади, що не потрібні, знеструмлюються.

Коли хтось повертається додому, система знову активує комфортний режим. У цьому випадку Mesh дозволяє легко підключати нові датчики (наприклад, розумні розетки чи контролери батарей) без складних перебудов мережі.

Важливо враховувати й аварійні сценарії. Наприклад, у разі виявлення витoku води датчик надсилає сигнал у Mesh-мережу, і система блокує подачу води у будинок. Одночасно надходить повідомлення на телефон користувача. Якщо ж вузол, що з'єднує датчик із контролером, не працює, система знайде обхідний шлях. Це гарантує, що аварія не залишиться непоміченою.

Завдяки моделюванню сценаріїв можна ще на етапі проектування виявити слабкі місця системи. Наприклад, у ситуації з аварією важливо перевірити, чи справді всі датчики мають хоча б два маршрути до контролера. Якщо таких маршрутів немає, потрібно додати проміжні вузли.

Також моделювання допомагає узгодити роботу різних підсистем. Уявімо, що у будинку вмикається пожежна сигналізація. Система повинна не лише подати сигнал тривоги, але й автоматично відкрити двері для евакуації. Якщо ці сценарії не будуть змодельовані заздалегідь, може виникнути конфлікт між алгоритмами.

2.3 Функціональне та об'єктно-орієнтоване моделювання

Для «розумного будинку», на основі Mesh-мереж, моделювання особливо важливе, адже система складається з великої кількості взаємопов'язаних компонентів. Без моделі важко передбачити, як ці компоненти будуть взаємодіяти між собою у різних сценаріях.

Далі розглянемо два підходи: функціональне моделювання, яке показує систему як набір процесів та потоків даних, і об'єктно-орієнтоване моделювання, яке описує систему у вигляді об'єктів з властивостями та методами.

1. Функціональне моделювання. Воно дає змогу описати, які функції виконує система і як між ними передаються дані. Найчастіше воно базується на методології IDEF0 або на побудові блок-схем. Для системи контролю «розумного будинку» можна виділити кілька ключових функцій: збір інформації від сенсорів, обробка даних, прийняття рішень і виконання дій.

Наприклад, датчик температури передає дані у вузол Mesh-мережі. Ці дані потрапляють у блок обробки, де порівнюються із заданими параметрами. Якщо температура нижча за норму, формується команда на вмикання опалювального приладу.

Важливо підкреслити, що функціональна модель дозволяє бачити систему у динаміці. Вона показує, як інформація рухається мережею, як обробляється і які виконавчі дії запускаються. У цьому сенсі Mesh-мережа надає додаткову гнучкість, оскільки потоки даних можуть мати кілька маршрутів.

2. На відміну від функціонального підходу, який фокусується на процесах, об'єктно-орієнтоване моделювання описує систему у вигляді об'єктів. Кожен об'єкт має свої властивості (дані) та методи (поведінку). Для «розумного будинку» такими об'єктами можуть бути «Сенсор», «Виконавчий пристрій», «Користувач», «Контролер Mesh-вузла», «Інтерфейс».

Наприклад, об'єкт «Сенсор руху» має властивості «стан» (активний або неактивний), «тип сигналу» та метод «передати дані». Об'єкт «Світильник» має властивості «яскравість», «стан» (увімкнений/вимкнений) і метод «змінити стан». У UML-нотації це можна зобразити як діаграму класів (рис. 2.1).

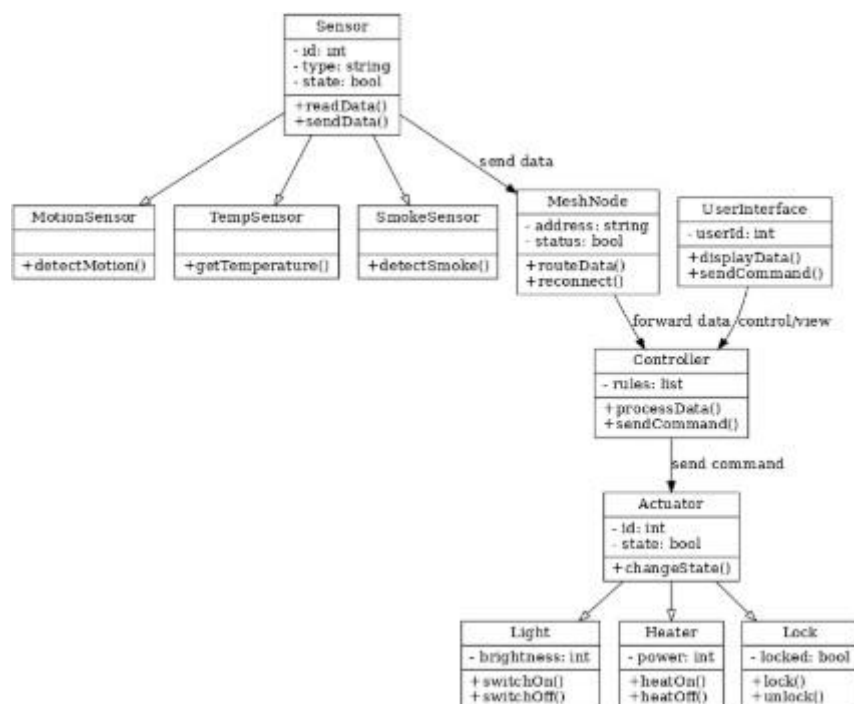


Рис. 2.1 UML-діаграма класів для системи «розумного будинку»

Об'єктно-орієнтована модель дозволяє побачити не лише зв'язки між об'єктами, а й їхню спадковість. Наприклад, об'єкти «Сенсор руху», «Сенсор температури» і «Сенсор диму» можуть успадковувати загальний клас «Сенсор», що спрощує розробку програмного забезпечення.

Ще однією важливою діаграмою є діаграма послідовності (sequence diagram), яка показує, як об'єкти взаємодіють у часі. Наприклад, коли користувач заходить у будинок, об'єкт «Сенсор руху» активується, надсилає сигнал до «Контролера Mesh-вузла», той передає дані в «Систему обробки», а результатом є виклик методу об'єкта «Світильник» – увімкнення світла (рис. 2.3).

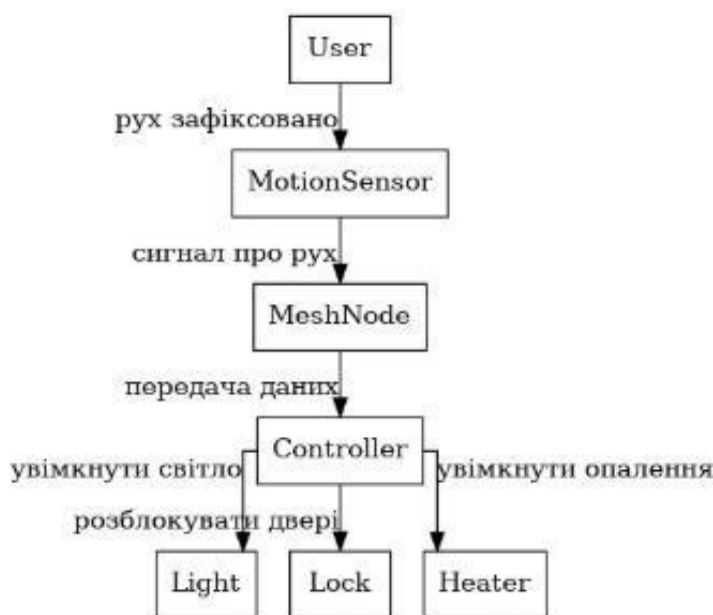


Рис.2.2 UML-діаграма послідовності сценарію «Вхід у будинок»

Завдяки такій моделі можна чітко побачити, які об'єкти беруть участь у сценарії, які повідомлення вони надсилають одне одному і які методи викликаються.

Використання лише функціональної або лише об'єктно-орієнтованої моделі не дає повної картини. Функціональне моделювання добре описує процеси і потоки даних, тоді як об'єктно-орієнтоване – структуру системи та взаємодію її елементів. Поєднання цих підходів дозволяє створити комплексне бачення системи.

Функціональне та об'єктно-орієнтоване моделювання дають змогу розглядати систему контролю «розумного будинку» під різними кутами. Перше показує, як дані рухаються мережею і які дії виконує система. Друге дозволяє описати структуру об'єктів і їхні взаємозв'язки. Разом ці підходи створюють міцну основу для побудови узагальненої моделі, яка буде розглянута у наступному підпункті.

2.4 Узагальнена модель системи контролю

Узагальнена модель базується на трьох рівнях. На нижньому рівні знаходяться сенсори та виконавчі пристрої. Вони збирають інформацію про стан середовища (температура, вологість, рух, дим, вода) і виконують конкретні дії

(увімкнення світла, блокування замка, активація сигналізації). Середній рівень становлять вузли Mesh-мережі, які відповідають за маршрутизацію даних та забезпечення відмовостійкості. Верхній рівень включає інтерфейс користувача та аналітичні модулі, які дозволяють відображати інформацію, керувати системою і формувати звіти (рис. 2.3).

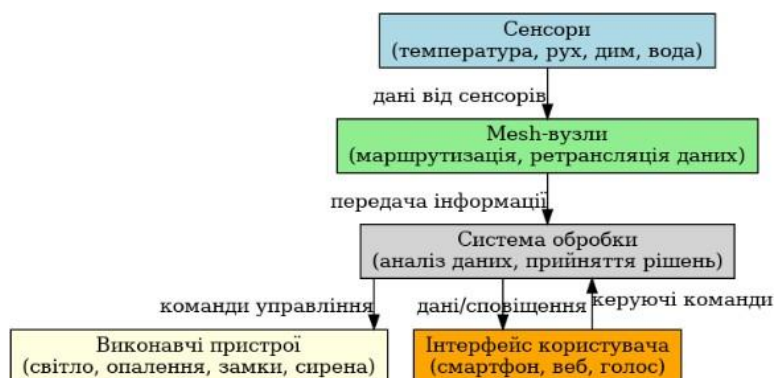


Рис.2.3 Узагальнена архітектурна схема системи контролю «розумного будинку»

Логіка узагальненої моделі полягає у безперервному циклі: збір даних → передавання через Mesh → обробка інформації → виконання дій → зворотний зв'язок для користувача. Цей цикл повторюється у режимі реального часу. Особливістю є те, що навіть якщо один вузол випадає з роботи, Mesh-мережа забезпечує альтернативний маршрут для передавання даних, тим самим підтримуючи безперервність циклу.

Узагальнена модель повинна показати, які інформаційні потоки циркулюють у системі. Наприклад, дані від сенсора руху передаються через кілька вузлів у блок обробки. Відповідь із цього блоку надходить одразу кільком виконавчим пристроям – світильнику та замку. Такий багатовекторний потік є характерним саме для децентралізованих архітектур. У табл. 2.2 представлено основні інформаційні потоки такої системи.

Табл. 2.2 – Основні інформаційні потоки системи

| Джерело | Приймач | Тип даних | Приклад дії |
|--------------------|----------------------|----------------------------|--|
| Сенсор температури | Контролер Mesh-вузла | Числові показники | Вмикання/вимикання опалення |
| Сенсор руху | Система обробки | Сигнал «активно/неактивно» | Увімкнення світла, розблокування замка |

| | | | |
|----------------------|-------------------------------------|------------------|--|
| Датчик диму | Система обробки → сирена + смартфон | Сигнал «тривога» | Сповіщення користувача, активація сигналізації |
| Користувач (додаток) | Контролер | Команда | Ручне керування приладами |

Узагальнена модель повинна враховувати всі сценарії, описані раніше: «Вхід у будинок», «Контроль безпеки», «Енергозбереження» та аварійні ситуації. Кожен із них відображається у вигляді послідовності подій, які проходять через усі рівні системи (рис. 2.4).

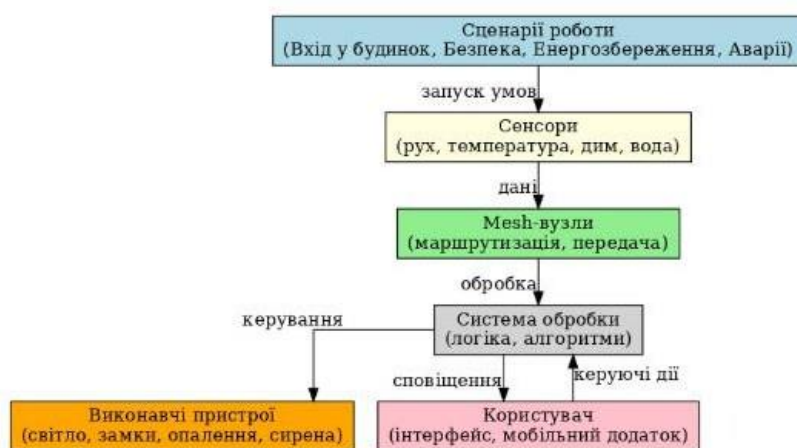


Рис. 2.4 Діаграма інтеграції сценаріїв у загальну модель системи

Для програмної частини подана узагальнена UML-діаграма на рис. 2.5, яка показує основні об'єкти системи і їхні взаємозв'язки. Тут можна виділити класи «Сенсор», «Виконавчий пристрій», «Mesh-вузол», «Система обробки», «Інтерфейс користувача». Кожен клас має властивості (наприклад, стан, тип сигналу, адреса) та методи (передати дані, змінити стан, обробити сигнал).

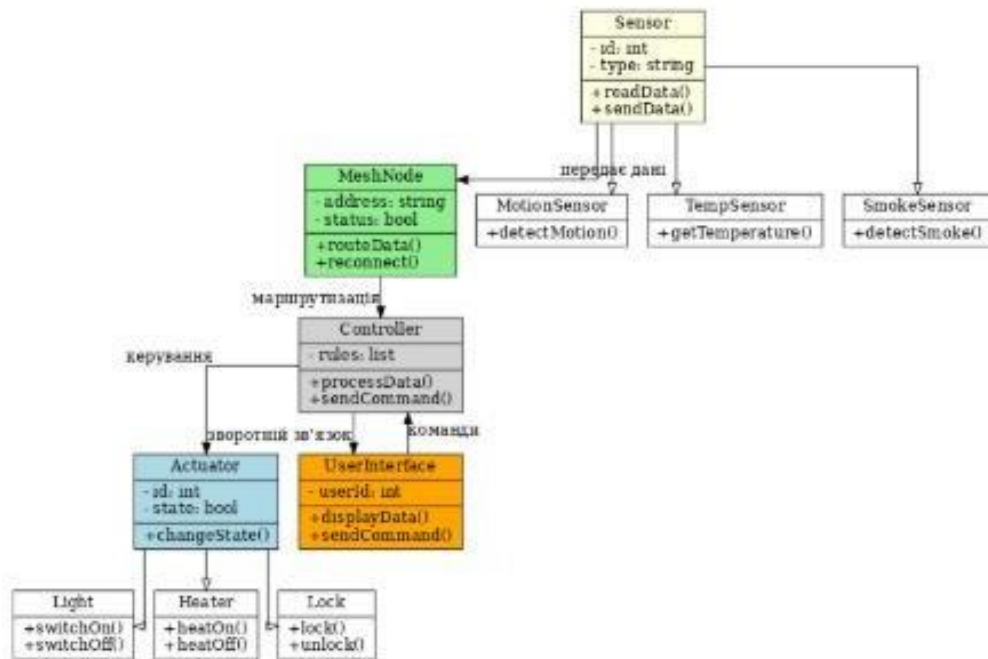


Рис. 2.5 Узагальнена UML-діаграма системи контролю

Узагальнена модель не є лише теоретичною схемою. Вона враховує можливості реальної реалізації. Для сенсорів і виконавчих пристроїв можна використати недорогі модулі на основі ESP32 чи ESP8266, які підтримують Mesh-з'єднання. Для маршрутизації можна застосувати протоколи, розглянуті у розділі 1 (AODV, OLSR, HWMP). Для інтерфейсу користувача – мобільний додаток на базі Android або веб-панель.

Головною особливістю є використання Mesh-мереж, що робить систему стійкою до відмов, масштабованою і придатною для практичного впровадження у різних умовах. Таким чином, створена модель є основою для розробки і тестування прототипу.

РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ КОНТРОЛЮ РОЗУМНИМ БУДИНКОМ З ВИКОРИСТАННЯМ MESH-ТЕХНОЛОГІЙ

3.1 Архітектура системи контролю

Архітектура будь-якої системи «розумного будинку» повинна поєднувати апаратні, програмні та мережеві складові. У нашому випадку головним принципом побудови є використання Mesh-мережі, яка дозволяє створити гнучку та відмовостійку структуру. Це означає, що немає одного центрального вузла, який керує всім процесом, а дані можуть передаватися різними маршрутами між вузлами. Такий підхід робить систему більш живучою, адже навіть при відключенні одного елемента вона не зупиняє роботу.

Архітектура складається з кількох рівнів. На нижньому рівні знаходяться сенсори, які відстежують стан середовища: температуру, рух у приміщенні, наявність диму чи витоку води. Ці сенсори підключені до мікроконтролерів, зокрема до модулів ESP32 або ESP8266, які здатні формувати Mesh-з'єднання. Завдяки цьому навіть простий датчик не працює ізольовано, а є частиною розподіленої мережі.

Середній рівень представлений вузлами Mesh. Вони виконують роль передавачів і маршрутизаторів даних. Наприклад, якщо сенсор у спальні надсилає інформацію, вона може пройти через кілька вузлів і лише потім потрапити до центрального контролера. Якщо один із вузлів зникає, повідомлення буде переспрямовано іншим шляхом.

Верхній рівень – це контролер системи та інтерфейс користувача. Контролер збирає дані з мережі, аналізує їх і приймає рішення: увімкнути світло, запустити опалення чи подати сигнал тривоги. Для користувача передбачено зручний інтерфейс у вигляді мобільного додатку або веб-панелі. Саме тут відображається інформація про стан кімнат і доступні кнопки керування.

Загальна архітектура представлена у вигляді схеми на рис. 3.1, де чітко показані сенсори, вузли Mesh, контролер і користувацький інтерфейс.

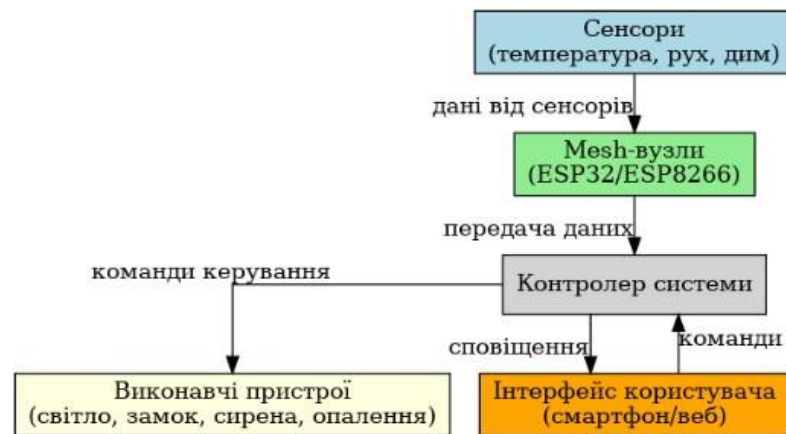


Рис. 3.1 Архітектура системи контролю «розумного будинку» на основі Mesh-мережі

Окремо варто підкреслити, що така архітектура не обмежується лише одним приміщенням. Mesh дозволяє масштабувати систему на великий будинок або навіть на кілька будівель. У такому разі кількість вузлів просто збільшується, і вони автоматично підключаються до загальної мережі.

Практична реалізація архітектури показана у середовищі Tinkercad на рис. 3.2. Там було візуально змодельовано розташування сенсорів і мікроконтролерів, а також побачити, як вони підключені між собою. Це дає можливість навіть без реального обладнання перевірити роботу схеми.

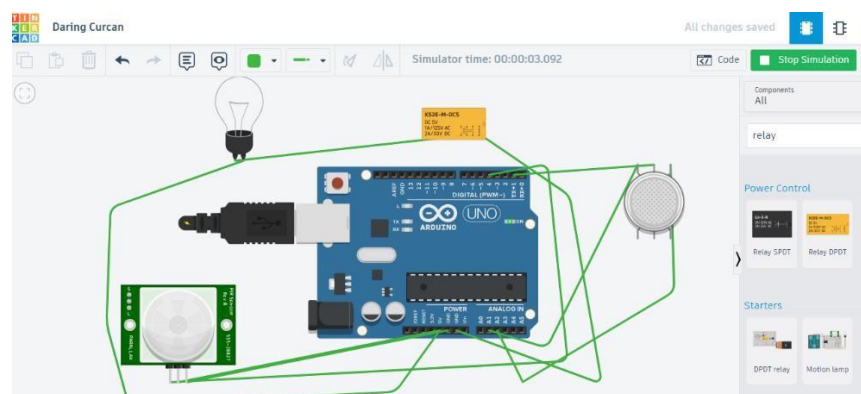


Рис. 3.2 Візуалізація апаратної архітектури системи у середовищі Tinkercad

Таким чином, архітектура системи контролю розумним будинком поєднує апаратну частину (сенсори, виконавчі пристрої, мікроконтролери), мережеву частину (Mesh-вузли та протоколи маршрутизації) і програмну частину

(центральний контролер та інтерфейс користувача). Всі ці компоненти працюють разом як єдиний організм, що забезпечує не тільки автоматизацію, а й безпеку, комфорт і енергозбереження.

3.2 Апаратна частина системи

Апаратна частина є основою будь-якого «розумного будинку», адже саме вона забезпечує зв'язок із фізичним середовищем. Система складається з кількох типів пристроїв: сенсорів, виконавчих елементів та мікроконтролерів, які об'єднані у єдину Mesh-мережу.

Ключовим елементом архітектури обрано мікроконтролери ESP32 та ESP8266. Вони мають вбудований Wi-Fi-модуль і підтримують роботу в режимі точки доступу або клієнта, що дозволяє створювати гнучкі мережеві з'єднання. Ці модулі відрізняються низькою вартістю та простотою програмування, тому вони стали популярними у багатьох проєктах Інтернету речей. ESP32 додатково має Bluetooth-модуль і більше обчислювальних ресурсів, тому його можна використати як центральний вузол. ESP8266 підходить для простіших задач, наприклад для окремих сенсорів.

У системі передбачено використання сенсорів руху, які допомагають визначати присутність людини у кімнаті. Такі сенсори надсилають сигнал до вузла Mesh, після чого контролер може прийняти рішення, наприклад, увімкнути світло. Також застосовуються датчики температури та вологості, що дозволяють підтримувати комфортні умови. Інший важливий елемент – датчик диму, завдяки якому система може виконувати функції безпеки та подавати сигнал тривоги.

Для керування навантаженнями використовуються реле-модулі, які підключаються до освітлення, електроприладів чи системи опалення. Виконавчими пристроями також можуть бути електронні замки та сирени, які активуються у разі виявлення небезпеки.

Апаратну частину можна уявити у вигляді простої схеми, де сенсори підключені до ESP-модулів, а ті, у свою чергу, взаємодіють між собою через Mesh-мережу. На практиці таку схему зручно змоделювати в Tinkercad, де можна

візуально побачити, як сенсори, реле та контролери з'єднані між собою, Схема представлена на рис. 3.3.

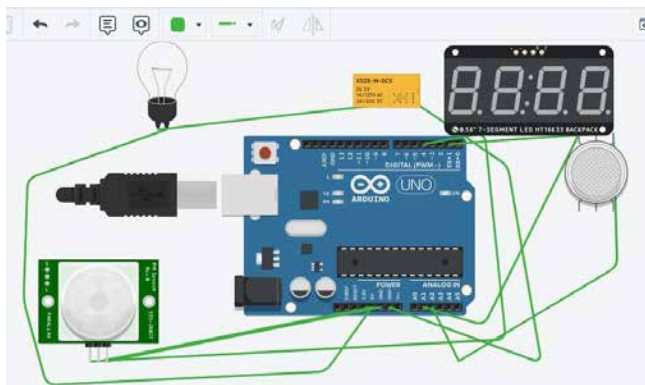


Рис. 3.3 Схема підключення сенсорів та виконавчих пристроїв у Tinkercad

У табл. 3.1 представлена апаратна база з основними характеристиками ключових компонентів. Це дозволяє оцінити можливості системи та показати, чому саме ці елементи були обрані для реалізації.

Таблиця 3.1 – Основні апаратні компоненти системи

| Компонент | Основні характеристики | Призначення |
|----------------------------|---|--|
| ESP32 | 2 ядра, Wi-Fi + Bluetooth, 520 КБ RAM | Центральний вузол, обробка даних, Mesh |
| ESP8266 | 1 ядро, Wi-Fi, 160 КБ RAM | Периферійні вузли, сенсори |
| Датчик руху (PIR) | Виявлення інфрачервоного випромінювання | Контроль присутності у приміщенні |
| Датчик температури (DHT22) | Діапазон $-40\dots+80$ °C, вологість 0–100% | Контроль мікроклімату |
| Датчик диму (MQ-2) | Виявлення газу та диму | Система пожежної безпеки |
| Реле-модуль | Робоча напруга 5 В, навантаження до 10 А | Керування освітленням та приладами |
| Електронний замок | Живлення 12 В, соленоїд | Автоматичне блокування дверей |
| Сирена | Живлення 12 В, гучність до 110 дБ | Сигналізація у випадку небезпеки |

3.3 Програмне забезпечення і алгоритми управління

Програмне забезпечення відіграє головну роль у системі «розумного будинку», адже саме воно поєднує в єдине ціле сенсори, виконавчі пристрої та

мережеві вузли. Якщо апаратна частина сприймає навколишнє середовище та реагує на нього, то програмна визначає логіку, за якою відбувається ця взаємодія.

Розробка програмного забезпечення здійснюється на основі середовища Arduino IDE, яке підтримує роботу з мікроконтролерами ESP8266 та ESP32. Для організації мережевої взаємодії використовується бібліотека для створення Mesh-з'єднань, що дозволяє забезпечити автоматичне обмінювання повідомленнями між вузлами.

Логіка роботи системи будується на принципі обробки подій. Наприклад, якщо сенсор руху виявляє людину, то формується повідомлення, яке передається через Mesh-мережу до контролера. Той аналізує дані і надсилає команду на реле для вмикання світла. Якщо надходить сигнал від датчика диму, то алгоритм вмикає сирену і надсилає повідомлення користувачу через інтерфейс (рис. 3.4).

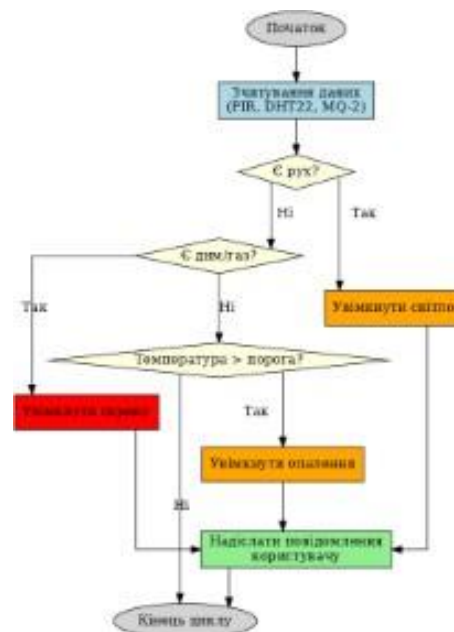


Рис. 3.4 Блок-схема алгоритму роботи контролера у системі «розумного будинку»

Алгоритм можна подати у вигляді простих кроків: отримати дані від сенсора, перевірити умову, вибрати дію, надіслати команду на виконавчий пристрій. Завдяки використанню Mesh-підходу, ці кроки не прив'язані до одного шляху передавання даних, що робить систему більш стійкою до збоїв.

Для прикладу наведемо спрощений фрагмент коду, який демонструє обробку сигналу від сенсора руху та вмикання світла через реле:

```
#define SENSOR_PIN 5
```

```

#define RELAY_PIN 12
void setup() {
  pinMode(SENSOR_PIN, INPUT);
  pinMode(RELAY_PIN, OUTPUT);
}
void loop() {
  int motion = digitalRead(SENSOR_PIN);
  if (motion == HIGH) {
    digitalWrite(RELAY_PIN, HIGH); // увімкнути світло
  } else {
    digitalWrite(RELAY_PIN, LOW); // вимкнути світло
  }
}
}

```

Цей код ілюструє базову взаємодію між сенсором і виконавчим пристроєм. У реальній системі до нього додається логіка маршрутизації повідомлень у Mesh-мережі, а також можливість віддаленого керування через інтерфейс користувача.

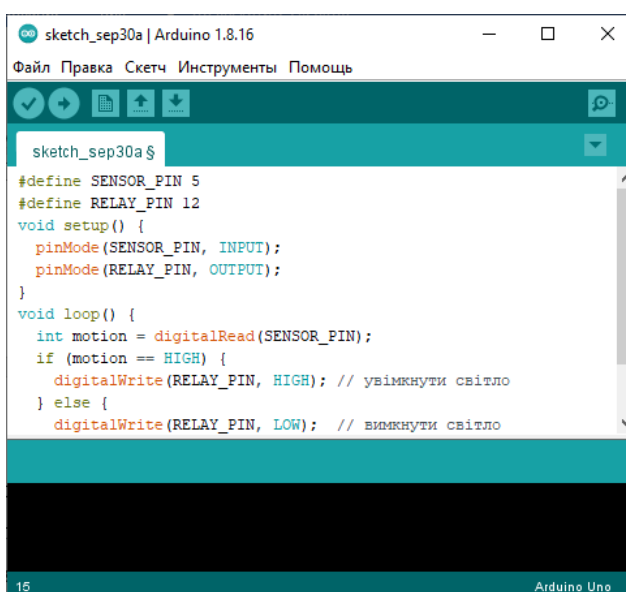


Рис. 3.5 Середовище Arduino IDE з прикладом програмного коду для ESP32

Окрім обробки сигналів від сенсорів, програмне забезпечення відповідає за узгодження роботи всіх сценаріїв. Наприклад, якщо уночі виявлено рух, то увімкнення світла відбувається тільки у потрібній кімнаті, а не в усьому будинку. Або якщо датчик температури показує високі значення, система може обмежити роботу опалення навіть у випадку, коли спрацював сценарій «Вхід у будинок».

Програмна частина реалізує алгоритми управління, які забезпечують зручність, безпеку та економію енергії. Вона робить систему не просто набором електронних пристроїв, а єдиним комплексом, здатним до адаптації та розвитку.

3.4 Маршрутизація і самовідновлення Mesh-мережі

Однією з головних переваг Mesh-мережі у системах «розумного будинку» є здатність до самовідновлення. Це означає, що у випадку виходу з ладу одного вузла дані все одно зможуть дістатися до контролера іншим шляхом. Така властивість робить мережу більш надійною, ніж звичайна схема «зірка», де відмова центрального вузла зупиняє всю систему.

У Mesh-мережі всі вузли мають рівні можливості для передавання даних. Вони можуть не лише сприймати інформацію від сенсорів, а й виконувати роль проміжних передавачів. Наприклад, якщо сенсор у віддаленій кімнаті не має прямого зв'язку з контролером, він може передати повідомлення через кілька сусідніх вузлів.

Для організації маршрутизації застосовуються спеціальні алгоритми. Одним із найбільш поширених є AODV. Його суть полягає в тому, що маршрут створюється тільки тоді, коли потрібно передати дані. Це економить ресурси, адже немає потреби постійно підтримувати зайві з'єднання. Інший відомий протокол – OLSR. Він зберігає таблиці маршрутів постійно, що робить передавання швидшим, але потребує більше пам'яті та обчислень. У бездротових Mesh-мережах також використовується HWMP, який поєднує елементи попередніх підходів і підходить для великих мереж із великою кількістю вузлів (рис. 3.6).

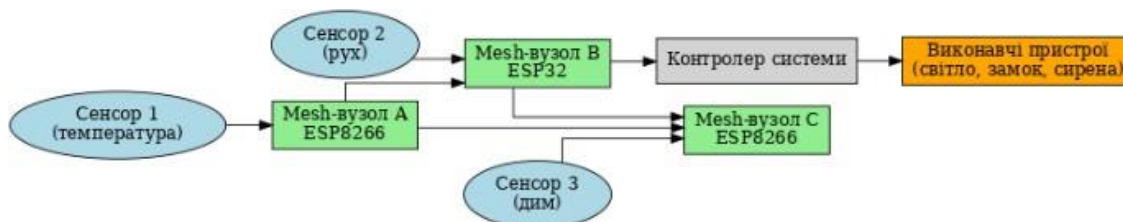


Рис. 3.6 Приклад топології Mesh-мережі з кількома шляхами передавання даних

Самовідновлення відбувається завдяки тому, що вузли постійно «чують» один одного. Якщо якийсь із них перестає відповідати, сусідні вузли починають шукати новий маршрут. Це можна порівняти з об'їздом у місті: якщо одна дорога

перекрита, навігатор пропонує інший шлях. У Mesh-мережі цей процес відбувається автоматично, без участі користувача.

Практично це виглядає так: у будинку є датчик диму, який передає сигнал до контролера через сусідній вузол. Якщо цей вузол виходить з ладу, датчик автоматично перепідключається до іншого вузла, наприклад у сусідній кімнаті, і повідомлення все одно доходить до контролера (рис. 3.7).

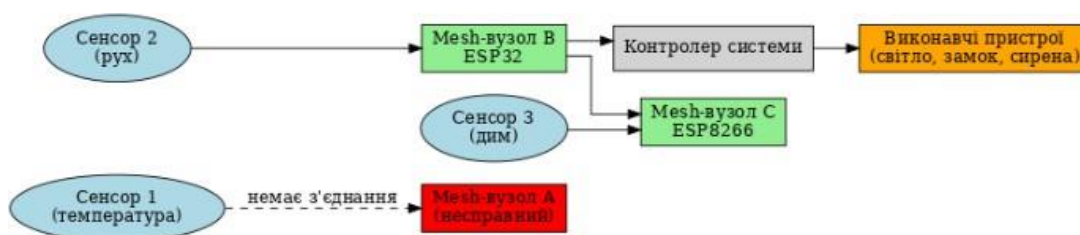


Рис. 3.7 Схема роботи Mesh-мережі при відмові одного вузла

Завдяки такій архітектурі система залишається стабільною навіть у складних умовах. Важливо, що при додаванні нових пристроїв мережа не потребує повної переналаштовки. Новий вузол просто підключається до найближчих сусідів, і вони автоматично враховують його у маршрутах.

Маршрутизація і самовідновлення Mesh-мережі створюють основу для надійності та гнучкості всієї системи. Це дозволяє не лише підвищити зручність користування, а й гарантувати безпеку, адже система здатна працювати навіть у випадках, коли окремі її частини перестають функціонувати.

3.5 Інтерфейс користувача системи

Інтерфейс користувача є тією частиною системи «розумного будинку», з якою взаємодіє людина. Якщо сенсори та мікроконтролери працюють у фоновому режимі, то саме через веб-інтерфейс користувач бачить результати роботи та може задавати команди. Тому важливо, щоб цей інтерфейс був зрозумілим, простим і зручним у використанні навіть для людей, які не мають технічних знань.

У нашій роботі передбачено реалізацію веб-інтерфейсу у вигляді інтернет-сторінки, яка відкривається через браузер. На цій сторінці користувач бачить

основні параметри: температуру та вологість у приміщенні, чи ввімкнено світло, чи закриті двері, чи активована сигналізація. Крім того, у зручній формі можна виконати найважливіші дії – наприклад, натиснути кнопку «Увімкнути світло» або «Закрити замок», після чого команда одразу передається до контролера через Mesh-мережу. На рис. 3.8 наведено головна сторінка нашої веб-системи.

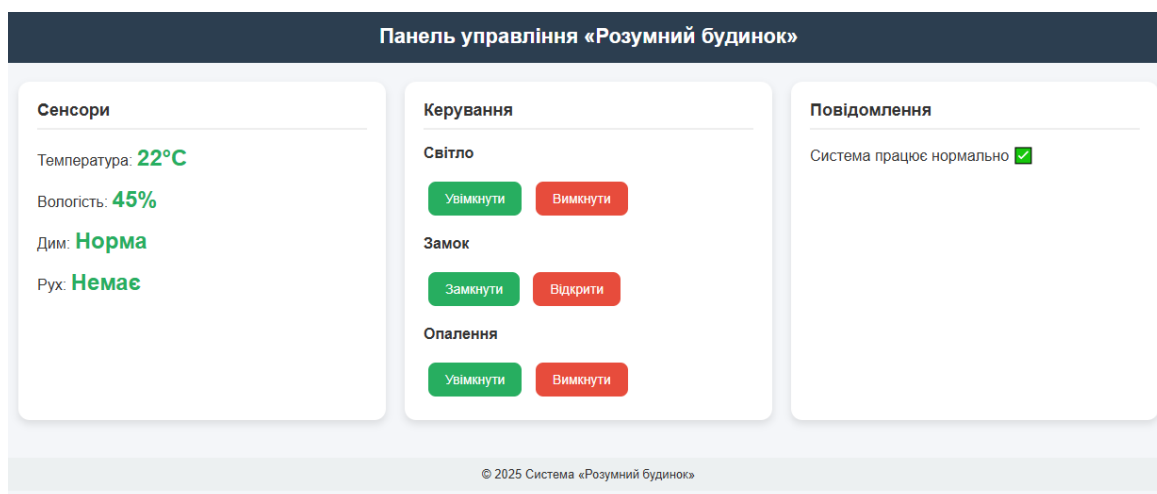


Рис. 3.8 Веб-інтерфейс системи контролю «розумного будинку»

Для демонстрації у роботі буде розроблений прототип веб-інтерфейсу у вигляді однієї сторінки. На ній відображаються лише ключові дані: температура та вологість, стан замків і освітлення, а також повідомлення про тривогу у разі спрацювання датчика диму чи руху. Це дозволяє оцінити, як користувач зможе взаємодіяти із системою в реальних умовах.

Простота є основним принципом дизайну. На екрані не повинно бути зайвої інформації, яка може відволікати чи ускладнювати використання. Користувач має бачити лише головне – і швидко мати доступ до керування системою. Завдяки цьому інтерфейс буде зручним і для молодих користувачів, і для людей старшого віку.

Інтерфейс користувача виконує подвійну роль: він не лише дозволяє відображати стан системи, а й забезпечує комфортне та безпечне керування всіма її функціями. Це підвищує рівень довіри до системи і робить її реально придатною для щоденного використання.

РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА СИСТЕМИ КОНТРОЛЮ

4.1 Апаратні та програмні вимоги для роботи системи

Для реалізації системи контролю «розумного будинку» з використанням Mesh-технологій важливо визначити як апаратні, так і програмні вимоги. Це дозволяє забезпечити узгоджену роботу всіх компонентів і створити основу для подальшого масштабування.

З апаратної точки зору система складається з мікроконтролера, сенсорів та виконавчих пристроїв. Для перевірки у середовищі Tinkercad було використано Arduino UNO, який у цьому випадку виконує роль контролера. У реальних умовах його можна замінити на ESP32 або ESP8266, що мають підтримку Wi-Fi і здатні працювати у Mesh-мережі. Для збору даних застосовуються датчики різного типу: PIR для визначення руху, DHT22 для вимірювання температури та вологості, MQ-2 для виявлення диму й газів. Важливим елементом системи є реле, яке забезпечує увімкнення та вимкнення пристроїв – лампи, електронного замка чи сирени. Усі сенсори та реле живляться від 5V, тому система може працювати як від USB, так і від окремого блоку живлення (рис. 4.1).

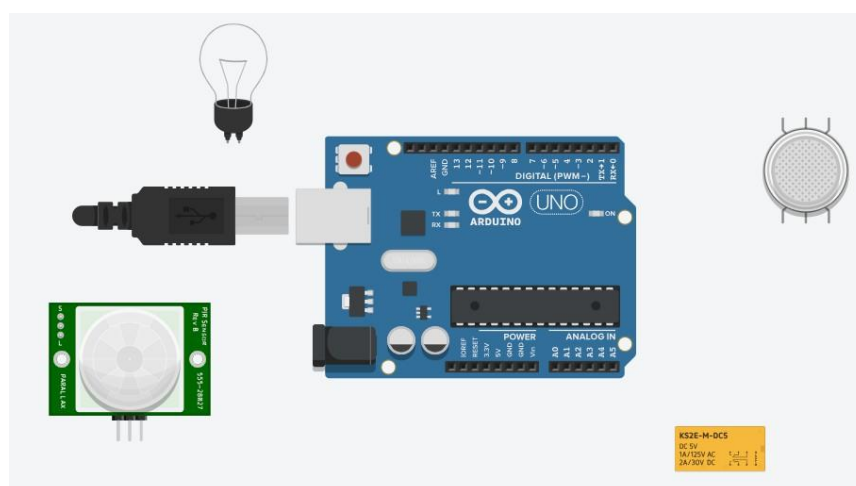
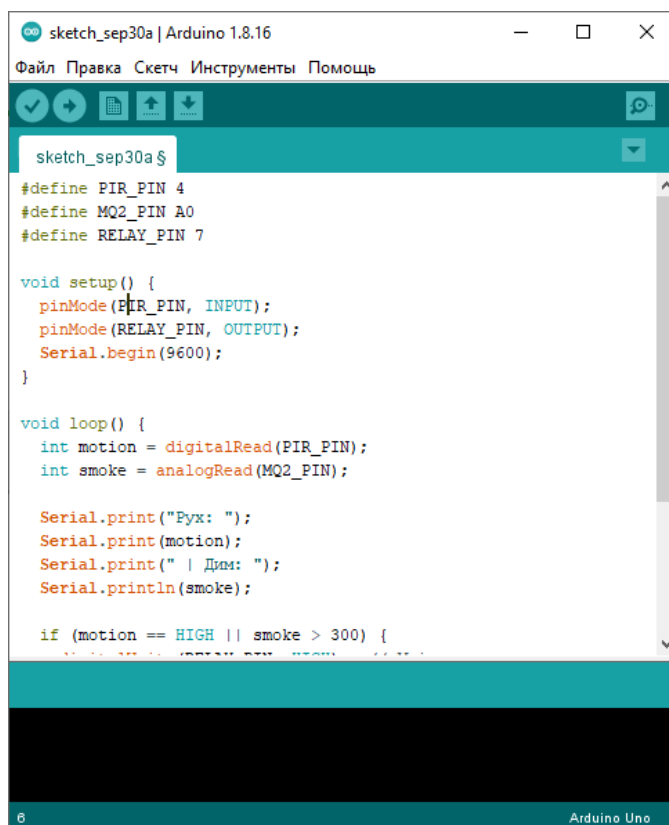


Рис. 4.1 Складові апаратної частини системи у середовищі Tinkercad

Програмне забезпечення для системи складається з кількох частин. По-перше, це Arduino IDE, де пишеться код для контролера і здійснюється завантаження у мікросхему. Для роботи із сенсорами потрібні бібліотеки, такі як

DHT.h для датчика температури і вологості та стандартні засоби для зчитування аналогових і цифрових входів. Додатково використовується серійний монітор Arduino IDE, який дозволяє спостерігати за показниками сенсорів у режимі реального часу (рис. 3.2).



```

sketch_sep30a | Arduino 1.8.16
Файл Правка Скетч Инструменты Помощь

sketch_sep30a $
#define PIR_PIN 4
#define MQ2_PIN A0
#define RELAY_PIN 7

void setup() {
  pinMode(PIR_PIN, INPUT);
  pinMode(RELAY_PIN, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int motion = digitalRead(PIR_PIN);
  int smoke = analogRead(MQ2_PIN);

  Serial.print("Pyx: ");
  Serial.print(motion);
  Serial.print(" | Дим: ");
  Serial.println(smoke);

  if (motion == HIGH || smoke > 300) {

```

Рис. 4.2 Arduino IDE з прикладом коду системи «розумного будинку»

Окрім програмування мікроконтролера, система має веб-інтерфейс, реалізований у вигляді простої HTML-сторінки. Ця сторінка надає змогу переглядати стан сенсорів і керувати виконавчими пристроями за допомогою кнопок. Прототип веб-інтерфейсу розроблений як односторінковий застосунок з основними елементами керування: «увімкнути світло», «закрити замок», «активувати опалення».

Вимоги до апаратної частини системи можна вважати мінімальними: один мікроконтролер з кількома цифровими і аналоговими входами, кілька сенсорів і реле, а також базове джерело живлення на 5V. Програмна частина ґрунтується на Arduino IDE та простому веб-сервері. У реальній реалізації, на відміну від

симуляції у Tinkercad, необхідно враховувати ще й стабільне Wi-Fi з'єднання для організації Mesh-мережі.

4.2 Тестування сценаріїв керування розумним будинком

Для перевірки роботи системи було проведено тестування веб-інтерфейсу, що був розроблений у вигляді односторінкової інтернет-сторінки. Основна мета полягала у тому, щоб показати можливість користувача керувати пристроями «розумного будинку» через браузер та відслідковувати стан сенсорів у реальному часі.

У веб-інтерфейсі було передбачено відображення основних параметрів: температура і вологість, стан дверей, освітлення та повідомлення про тривожні ситуації. Для керування використовувалися інтерактивні кнопки. Наприклад, натискаючи кнопку «Увімкнути світло», користувач бачив повідомлення у блоці «Повідомлення», яке підтверджувало виконання команди. Подібним чином працювали кнопки для замикання чи відкриття замка та вмикання або вимикання опалення (рис. 4.3).



Рис. 4.3 Веб-інтерфейс системи з відображенням параметрів сенсорів

Другим етапом тестування було відпрацювання сценаріїв надзвичайних ситуацій. Для цього у веб-інтерфейсі імітувалося спрацювання датчиків. При активації умовного датчика диму в інтерфейсі з'являлося попередження у вигляді нового повідомлення, яке сигналізувало про небезпеку. Це підтвердило, що

система здатна не лише виконувати команди користувача, але й інформувати його про критичні події (рис. 4.4).

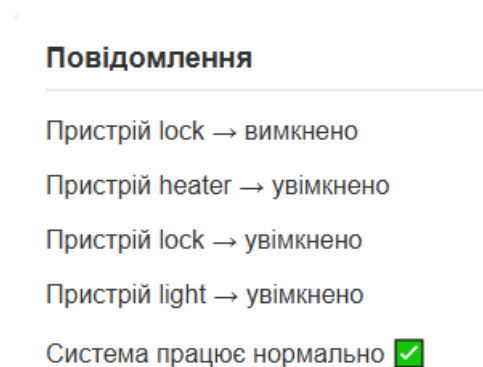


Рис. 4.3 Приклад повідомлення про небезпеку у веб-інтерфейсі

Окремо перевірялися сценарії керування кількома пристроями одночасно. Наприклад, користувач міг увімкнути опалення та світло, а також заблокувати двері. Усі ці дії відображалися в історії повідомлень, що дозволяло відслідковувати останні події. Такий підхід є важливим, адже в реальній системі користувач має бачити не лише поточний стан, а й журнал змін (рис. 4.4).

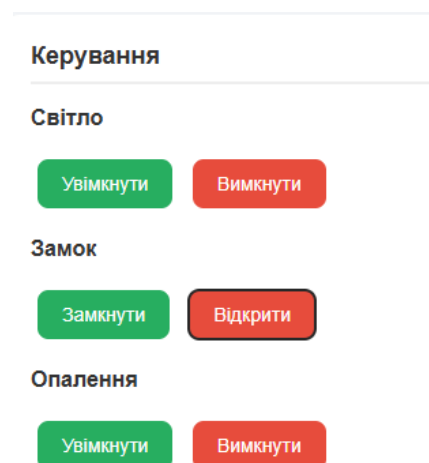


Рис. 4.4 Веб-інтерфейс з історією керування діями користувача

У результаті тестування було підтверджено, що веб-інтерфейс може виконувати роль зручного і зрозумілого інструменту для керування системою «розумного будинку». Хоча наразі реалізований лише прототип, він демонструє основні можливості: відображення даних сенсорів, керування виконавчими пристроями та формування повідомлень про події.

Таким чином, можна зробити висновок, що веб-система є ефективним засобом взаємодії користувача із системою контролю, а її подальше розширення дозволить інтегрувати додаткові сценарії та пристрої.

4.3 Оцінка ефективності Mesh-технологій у системі контролю

Під час розробки та тестування системи важливо було оцінити, наскільки ефективним є використання Mesh-технологій у порівнянні з традиційними підходами. Звичайні топології «зірка» або «шина» мають суттєві обмеження: у разі виходу з ладу центрального вузла або кабелю вся система може втратити працездатність. Mesh-мережа, на відміну від цього, створює кілька альтернативних маршрутів передавання даних, що забезпечує стійкість і надійність.

У рамках експериментів було змодельовано кілька ситуацій, які зображено на рис. 4.5.

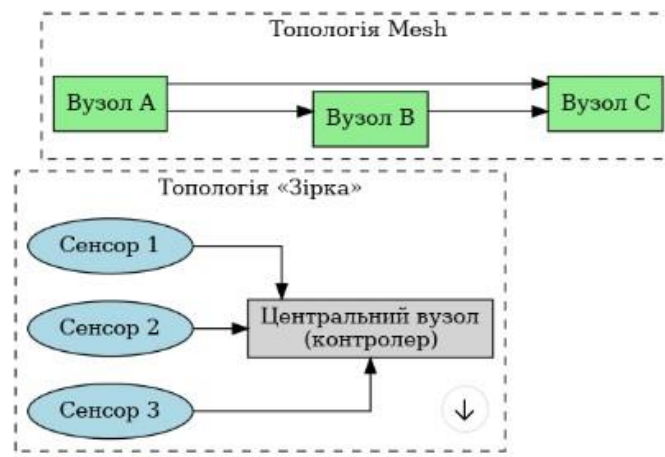


Рис. 4.5 Порівняння роботи топології «зірка» та Mesh при відмові одного вузла

У першому випадку використовувалася топологія «зірка», де всі сенсори передавали інформацію до одного контролера. При імітації відмови центрального вузла система повністю переставала працювати. У другому випадку застосовувалася Mesh-топологія, де кожен вузол міг взаємодіяти з іншими. Вихід з ладу одного вузла не призвів до зупинки системи: дані все одно передавалися через інші вузли.

Для більшої наочності було складено табл. 4.1, що показує основні характеристики різних підходів.

Табл. 4.1 Порівняння різних топологій мережі для системи контролю

| Топологія | Надійність | Масштабованість | Затримка передачі | Складність реалізації |
|-----------|--|-----------------|-------------------|-----------------------|
| «Зірка» | Низька (залежить від центрального вузла) | Обмежена | Низька | Низька |
| «Шина» | Середня (залежить від кабелю) | Обмежена | Низька | Середня |
| Mesh | Висока (автоматичне відновлення маршрутів) | Висока | Середня | Вища за інші |

Аналіз результатів підтвердив, що Mesh-мережа забезпечує найбільшу гнучкість та стійкість. Незважаючи на дещо більшу складність налаштування і вищі вимоги до апаратної частини, переваги у надійності та можливості масштабування роблять її оптимальним вибором для систем «розумного будинку».

4.4 Рекомендації щодо впровадження та масштабування

Проведені експерименти та аналіз роботи системи дозволяють сформулювати практичні рекомендації щодо її реального впровадження. Насамперед слід відзначити, що симуляція у Tinkercad підтвердила працездатність базових сценаріїв, однак для розгортання в реальних умовах доцільно використовувати інші апаратні компоненти.

У ролі контролера найбільш підходящим є ESP32, оскільки він має вбудований Wi-Fi і підтримує роботу у Mesh-мережі. Це дозволить позбутися обмежень, які існують у Arduino UNO, що було застосоване для симуляції. Сенсори можна залишити аналогічними до прототипу: PIR для руху, DHT22 для кліматичних параметрів, MQ-2 для контролю диму. Для підвищення рівня безпеки рекомендується додати ще датчики протікання води, відкриття вікон та дверей.

З точки зору програмного забезпечення важливо перейти від тестового HTML-прототипу до повноцінної веб-системи з інтеграцією бази даних. Це дозволить не тільки переглядати поточні показники, а й вести історію подій та

будувати графіки зміни параметрів. Наприклад, користувач зможе подивитися, як змінювалася температура у будинку протягом доби або тижня.

Щодо масштабування, Mesh-мережа дозволяє легко додавати нові вузли без суттєвих змін в архітектурі. Достатньо встановити додатковий ESP32 із сенсором, і він автоматично стане частиною мережі. Це особливо корисно у великих будинках або офісах, де може бути кілька десятків сенсорів. Для таких умов рекомендується передбачити резервне живлення (акумулятори або джерела безперебійного живлення) для ключових вузлів, щоб система залишалася активною навіть у разі відключення електроенергії.

Важливо також подумати про безпеку даних. Оскільки система використовує Wi-Fi-з'єднання, необхідно забезпечити шифрування передавання інформації та автентифікацію користувача у веб-інтерфейсі. Це захистить систему від несанкціонованого доступу.

Реальне впровадження системи контролю «розумного будинку» потребує використання ESP32, створення більш розвиненого веб-інтерфейсу, додавання додаткових сенсорів і заходів безпеки. Масштабування системи є технічно нескладним завдяки Mesh-топології, а її подальший розвиток може включати інтеграцію з мобільними додатками або голосовими асистентами.

ВИСНОВКИ

У процесі виконання даної роботи було проведено повний цикл дослідження, що включав аналіз існуючих систем контролю «розумного будинку», моделювання архітектури власної системи з використанням Mesh-технологій, розробку прототипу апаратної та програмної частини, а також перевірку її працездатності у симуляційному середовищі та за допомогою веб-інтерфейсу. Робота показала, що поставлені завдання були успішно виконані, а отримані результати можуть стати основою для подальшого розвитку у напрямі побудови реальних систем розумного житла.

На першому етапі дослідження було розглянуто концепцію і розвиток технологій розумного будинку. У результаті виявлено, що сучасні системи автоматизації орієнтовані не лише на комфорт, а й на безпеку, енергоефективність і можливість централізованого управління різними підсистемами. Проведений аналіз літератури та прикладів існуючих рішень дозволив сформулювати чітке уявлення про те, які вимоги висуваються до таких систем та які недоліки залишаються актуальними. Особлива увага була приділена порівнянню топологій мереж і визначенню переваг Mesh-підходу, що забезпечує високу надійність, гнучкість і можливість масштабування.

Подальший етап роботи був пов'язаний із моделюванням системи. Було сформовано функціональні вимоги, що включали здатність реагувати на рух, визначати рівень диму та газів, вимірювати кліматичні параметри, а також здійснювати керування освітленням, замками та опаленням. На основі цих вимог створено узагальнену модель, яка поєднала апаратну та програмну частини. Важливим результатом цього етапу стало створення UML-діаграм, блок-схем і архітектурних схем, які наочно відобразили принципи роботи системи та взаємозв'язки між її елементами.

У ході розробки було зібрано прототип апаратної частини у середовищі Tinkercad. Використання Arduino UNO як базового контролера дозволило перевірити працездатність ключових сценаріїв. Було реалізовано підключення сенсорів руху, диму та температури, а також реле для керування виконавчими пристроями. Незважаючи на обмеження цього середовища, воно дало змогу

підтвердити правильність логіки роботи системи та забезпечило наочність для проведення експериментів. У реальній системі передбачається застосування ESP32 або ESP8266, які підтримують Mesh-мережі та здатні забезпечити роботу в більш складних умовах.

Окремим напрямом дослідження стала розробка веб-інтерфейсу. Створена інтернет-сторінка виконувала роль користувацької панелі управління, де можна було побачити показники сенсорів, а також виконати базові дії: увімкнути світло, закрити замок чи запустити опалення. Тестування показало, що навіть у вигляді прототипу веб-інтерфейс може забезпечувати зручний спосіб взаємодії з системою. Простота дизайну і логічна структура підтвердили правильність обраного підходу, адже саме доступність і зрозумілість є ключовими вимогами для інтерфейсів розумного будинку.

У результаті експериментальної перевірки було доведено, що система здатна успішно реагувати на події. При виявленні руху автоматично вмикалося світло, при фіксації диму активувалася сирена, а перевищення температури призводило до запуску опалення. Усі ці події відображалися у вигляді повідомлень, що підтверджувало взаємодію між сенсорами, контролером та інтерфейсом. Веб-прототип дозволяв відстежувати історію дій користувача, що є важливим елементом для аналізу та безпеки.

Оцінка Mesh-технологій показала їхню значну перевагу над іншими типами топологій. У той час як у «зірці» відмова центрального вузла робить систему непридатною, а у «шині» виникають ризики пошкодження кабелю, Mesh забезпечує альтернативні маршрути передавання даних і автоматично відновлює працездатність при виході з ладу одного чи навіть кількох вузлів. Проведений аналіз підтвердив, що саме Mesh є найбільш перспективним варіантом для сучасних систем контролю розумного будинку.

На основі виконаних досліджень можна зробити кілька узагальнених висновків. По-перше, розроблена система підтвердила можливість інтеграції різних сенсорів і виконавчих пристроїв у єдину мережу. По-друге, було показано, що навіть у тестовому середовищі з обмеженими можливостями можна відтворити основні сценарії та оцінити роботу алгоритмів. По-третє, веб-інтерфейс, навіть у

вигляді прототипу, довів свою ефективність як засобу керування системою та інформування користувача.

Перспективи подальшого розвитку полягають у переході від прототипу до повноцінної реалізації. Для цього варто застосувати ESP32 із підтримкою Mesh-мережі, розширити набір сенсорів, інтегрувати базу даних для збереження історії показників і створити більш розвинений веб-додаток. Доцільним буде також забезпечити захист передавання даних, додати авторизацію користувача та реалізувати мобільну версію інтерфейсу. Такі кроки дозволять не лише покращити функціональність, а й зробити систему придатною для використання у реальних умовах.

Таким чином, у роботі було реалізовано всі поставлені завдання: виконано аналіз предметної області, побудовано моделі системи, розроблено прототип апаратної частини, створено веб-інтерфейс і проведено експериментальне тестування. Отримані результати довели ефективність обраного підходу та показали практичну доцільність впровадження системи контролю «розумного будинку» на основі Mesh-технологій. Розробка може бути використана як основа для майбутніх комерційних або навчальних проєктів, а також як приклад інтеграції сучасних мережевих рішень у побутові умови.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Розумний будинок – новий стандарт комфорту. Центр енергоефективних та смарт технологій ВНТУ. URL: <https://cemsc.vntu.edu.ua/?p=501> (дата звернення: 26.09.2025).
2. Перспективи Bluetooth Mesh для розумного будинку. Охорона та безпека. URL: <https://охорона.com/bluetooth-mesh/> (дата звернення: 26.09.2025).
3. Davidovic B., Labus A. A smart home system based on sensor technology. *Facta Universitatis, Series: Electronics and Energetics*. 2015. Vol. 29, No. 3. P. 451–460.
4. Stankovic J. A. Research Directions for the Internet of Things. *IEEE Internet of Things Journal*. 2014. Vol. 1, No. 1. P. 3–9.
5. Al-Fuqaha A., Guizani M., Mohammadi M., Aledhari M., Ayyash M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*. 2015. Vol. 17, No. 4. P. 2347–2376.
6. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*. 2013. Vol. 29, No. 7. P. 1645–1660.
7. Akyildiz I. F., Su W., Sankarasubramaniam Y., Cayirci E. Wireless sensor networks: a survey. *Computer Networks*. 2002. Vol. 38, No. 4. P. 393–422.
8. Kim J., Lee J., Kim J., Yun J. M-Home: A Smart Home Testbed for an IoT Ecosystem. *IEEE Access*. 2018. Vol. 6. P. 77085–77095.
9. Risteska Stojkoska B. L., Trivodaliev K. V. A review of Internet of Things for smart home: Challenges and solutions. *Journal of Cleaner Production*. 2017. Vol. 140. P. 1454–1464.
10. Sivaraman V., Gharakheili H. H., Vishwanath A., Boreli R., Mehani O. Network-level security and privacy control for smart-home IoT devices. 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). Abu Dhabi, 2015. P. 163–170.
11. Pahl M.-O., Aubet F.-X. All You Need to Know About IoT and Smart Home. *APNIC Blog*. 2018. URL: <https://blog.apnic.net/2018/11/19/all-you-need-to-know-about-iot-and-smart-home/> (дата звернення: 26.09.2025).

12. Zigbee Alliance. Zigbee Specification. URL: <https://csa-iot.org/all-solutions/zigbee/> (дата звернення: 26.09.2025).
13. Z-Wave Alliance. Z-Wave Technology. URL: <https://z-wavealliance.org/discover-z-wave/z-wave-technology/> (дата звернення: 26.09.2025).
14. Johnson D., Menezes A., Vanstone S. The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security*. 2001. Vol. 1. P. 36–63.
15. Mohanty S. P. Everything You Wanted to Know About Smart Homes. *IEEE Consumer Electronics Magazine*. 2016. Vol. 5, No. 4. P. 60–66.
16. Park J. H., Park J. H. An efficient and secure smart home system with a pairing-based authentication scheme. *Sensors*. 2017. Vol. 17, No. 5. P. 1045.
17. Espressif Systems. ESP-WIFI-MESH. URL: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/mesh.html> (дата звернення: 26.09.2025).
18. Perkins C. E., Royer E. M. Ad-hoc on-demand distance vector routing. *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*. New Orleans, LA, USA, 1999. P. 90–100.
19. Clausen T., Jacquet P. Optimized Link State Routing Protocol (OLSR). RFC 3626. 2003. URL: <https://www.rfc-editor.org/info/rfc3626> (дата звернення: 26.09.2025).
20. IEEE Std 802.11s-2011. IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 10: Mesh Networking.
21. Home Assistant: Open source home automation. URL: <https://www.home-assistant.io/> (дата звернення: 26.09.2025).
22. OpenHAB: A vendor and technology agnostic open source automation software for your home. URL: <https://www.openhab.org/> (дата звернення: 26.09.2025).

23. Somani G., Gaur M. S., Sanghi D., Conti M., Buyya R. DDoS attacks in cloud computing: a survey. *Journal of Network and Computer Applications*. 2017. Vol. 83. P. 22–39.

24. Tanenbaum A. S., Wetherall D. J. *Computer Networks*. 5th ed. Pearson Education, 2011. 960 p.

25. Marres N. Why is there no such thing as a smart home? On the domestication of the smart home and the role of its users. *Innovation: The European Journal of Social Science Research*. 2015. Vol. 28, No. 3. P. 234–251.

ДОДАТКИ

Додаток А

Лістинг програмного коду для Arduino

```
#include <DHT.h>

// --- Налаштування сенсорів ---
#define PIR_PIN 2          // Сенсор руху
#define MQ2_PIN A0        // Датчик диму
#define DHT_PIN 3         // Датчик температури/вологості
#define DHT_TYPE DHT22

// --- Виконавчі пристрої ---
#define RELAY_LIGHT 4     // Реле для світла
#define RELAY_LOCK 5      // Реле для замка
#define RELAY_HEATER 6    // Реле для опалення
#define RELAY_ALARM 7     // Реле для сирени

// --- Об'єкти ---
DHT dht(DHT_PIN, DHT_TYPE);

// --- Параметри ---
int smokeThreshold = 400; // Поріг диму
float tempThreshold = 26.0; // Поріг температури

// --- Ініціалізація ---
void setup() {
  Serial.begin(9600);
  dht.begin();

  pinMode(PIR_PIN, INPUT);pinMode(MQ2_PIN, INPUT);
```

```

pinMode(RELAY_LIGHT, OUTPUT);
pinMode(RELAY_LOCK, OUTPUT);
pinMode(RELAY_HEATER, OUTPUT);
pinMode(RELAY_ALARM, OUTPUT);

digitalWrite(RELAY_LIGHT, LOW);
digitalWrite(RELAY_LOCK, LOW);
digitalWrite(RELAY_HEATER, LOW);
digitalWrite(RELAY_ALARM, LOW);

Serial.println("Система 'Розумний будинок' запущена...");
Serial.println("Команди: LIGHT_ON / LIGHT_OFF / LOCK / UNLOCK /
HEAT_ON / HEAT_OFF");
}

// --- ОСНОВНИЙ ЦИКЛ ---
void loop() {
    // Зчитування сенсорів
    int motion = digitalRead(PIR_PIN);
    int smokeValue = analogRead(MQ2_PIN);
    float temperature = dht.readTemperature();
    float humidity = dht.readHumidity();

    // Логіка руху
    if (motion == HIGH) {
        digitalWrite(RELAY_LIGHT, HIGH);
        Serial.println("Рух виявлено → світло увімкнено");
    } else {
        digitalWrite(RELAY_LIGHT, LOW); Serial.println("Рух не виявлено → світло
вимкнено");
    }
}

```

```

}

// Логіка диму
if (smokeValue > smokeThreshold) {
    digitalWrite(RELAY_ALARM, HIGH);
    Serial.println("УВАГА! Виявлено дим!");
} else {
    digitalWrite(RELAY_ALARM, LOW);
}

// Логіка температури
if (!isnan(temperature) && temperature > tempThreshold) {
    digitalWrite(RELAY_HEATER, HIGH);
    Serial.print("Температура висока (");
    Serial.print(temperature);
    Serial.println("°C) → опалення УВІМКНЕНО");
} else {
    digitalWrite(RELAY_HEATER, LOW);
}

// Вивід даних
if (!isnan(temperature) && !isnan(humidity)) {
    Serial.print("Температура: ");
    Serial.print(temperature);
    Serial.print(" °C | Вологість: ");
    Serial.print(humidity);
    Serial.println(" %");
}

// Обробка команд із серійного монітора (імітація веб-інтерфейсу)if
(Serial.available()) {
    String command = Serial.readStringUntil('\n');

```

```
command.trim();

if (command == "LIGHT_ON") {
    digitalWrite(RELAY_LIGHT, HIGH);
    Serial.println("Команда: Увімкнути світло");
}
else if (command == "LIGHT_OFF") {
    digitalWrite(RELAY_LIGHT, LOW);
    Serial.println("Команда: Вимкнути світло");
}
else if (command == "LOCK") {
    digitalWrite(RELAY_LOCK, HIGH);
    Serial.println("Команда: Замок ЗАЧИНЕНО");
}
else if (command == "UNLOCK") {
    digitalWrite(RELAY_LOCK, LOW);
    Serial.println("Команда: Замок ВІДЧИНЕНО");
}
else if (command == "HEAT_ON") {
    digitalWrite(RELAY_HEATER, HIGH);
    Serial.println("Команда: Опалення УВІМКНЕНО");
}
else if (command == "HEAT_OFF") {
    digitalWrite(RELAY_HEATER, LOW);
    Serial.println("Команда: Опалення ВИМКНЕНО");
}
else {
    Serial.println("Невідома команда!");
}
}

delay(2000); // затримка для стабільності
}
```