

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти бакалавр

Спеціальність 126 Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедру ІСТ

_____ Каміла СТОРЧАК

“ _____ ” _____ 2026 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Матвійчуку Ернесту Вікторовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Розумне опалення в системі "Розумний будинок" з використанням безпроводових протоколів зв'язку

керівник кваліфікаційної роботи: Віктор САГАЙДАК PhD

(ім'я, ПРИЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від “ 20 ” лютого 2026 р. №51

2. Строк подання кваліфікаційної роботи «1» червня 2026 р.

3. Вихідні дані кваліфікаційної роботи:

1. Принципи побудови та функціонування систем інтелектуального життєвого простору (Smart Home).

2. Технічні характеристики безпроводового протоколу зв'язку ZigBee 3.0 та полегшеного протоколу MQTT.

3. Сучасне програмно-апаратне забезпечення локальної автоматизації (мікрокомп'ютер Raspberry Pi, розумні радіаторні термостати TRV, безпроводові кліматичні сенсори).

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

1. Аналіз теоретичних засад побудови сучасних систем мікрокліматичного контролю в концепції IoT.

2. Порівняльний аналіз безпроводових інтерфейсів передачі даних та обґрунтування вибору технологічного стеку (ZigBee, MQTT, Home Assistant).

3. Проектування структурно-функціональної схеми, апаратного комплексу та програмної інфраструктури системи розумного опалення.

4. Розробка математичних моделей терморегуляції, логічних алгоритмів та YAML-сценаріїв автоматизації («Нічний режим», «Нікого немає вдома», «Захист від провітрювання»).

5. Перелік ілюстраційного матеріалу: презентація

6. Дата видачі завдання «20» лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/ п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Одержання завдання, аналіз джерел та розробка плану роботи.	20.02-05.03.26	
2.	Аналіз технологій «Розумний будинок» та безпроводових протоколів.	06.03-02.04.26	
3.	Проектування архітектури системи та моделі обміну даними MQTT.	03.04-23.04.26	
4.	Програмна реалізація, інтеграція пристроїв та налаштування сценаріїв.	24.04-12.05.26	
5.	Висновки після роботи	13.05-14.05.26	
6.	Експериментальні дослідження, оцінка енергоефективності та оформлення тексту.	15.05-22.05.26	
7.	Перевірка на антиплагіат, нормоконтроль та подання роботи на кафедру.	23.05-1.06.26	

Здобувач вищої освіти _____
(підпис)

Керівник кваліфікаційної роботи _____
(підпис)

Ернест МАТВІЙЧУК
(ім'я, ПРІЗВИЩЕ)

Віктор САГАЙДАК
(ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавр: 51 стор., 19 рис., 5 табл., 23 джерел.

Мета роботи – теоретичне моделювання, архітектурне проектування та практична розробка децентралізованої автономної інформаційної системи інтелектуального керування опаленням у структурі Smart Home.

Об'єкт дослідження – процес автоматизованого моніторингу, збору телеметричних даних та покімнатного (зонального) мікрокліматичного регулювання параметрів опалення в житловому просторі.

Предмет дослідження – архітектурні моделі, безпроводові інтерфейси зв'язку (зокрема протокол ZigBee 3.0), програмно-апаратні засоби інтеграції (Home Assistant, MQTT) та алгоритми логічного управління сценаріями автоматизації систем розумного опалення.

Короткий зміст роботи. У бакалаврській роботі проведено аналіз архітектури та перспективних технологій для IoT-систем мікроклімату. Обґрунтовано вибір безпроводового протоколу ZigBee 3.0 та локального сервера на базі Home Assistant, що функціонує незалежно від хмарних сервісів [15]. Розроблено структурно-функціональну схему взаємодії елементів розумного опалення та спроектовано програмну інфраструктуру обміну даними через протокол MQTT. Створено математичні моделі й логічні алгоритми керування опаленням, реалізовані за допомогою YAML-сценаріїв для базових режимів експлуатації житла («Нічний режим», «Нікого немає вдома», «Захист від провітрювання»), що дозволяє знизити фінансові витрати на 15–30%.

КЛЮЧОВІ СЛОВА: РОЗУМНИЙ БУДИНОК, SMART HOME, РОЗУМНЕ ОПАЛЕННЯ, ІНТЕРНЕТ РЕЧЕЙ, IOT, ZIGBEE 3.0, MQTT, HOME ASSISTANT, АВТОМАТИЗАЦІЯ МІКРОКЛІМАТУ.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМ РОЗУМНОГО ОПАЛЕННЯ.....	13
1.1 Сутність інтелектуального опалення та його функціональна роль у структурі Smart Home.....	13
1.2 Техніко-технологічна класифікація та архітектурний аналіз компонентів IoT-системи опалення.....	14
1.3 Історія розвитку та сучасні тенденції систем автоматизації опалення.....	18
1.4 Аналіз інтелектуальних алгоритмів та методів керування мікрокліматом...	19
РОЗДІЛ 2 АРХІТЕКТУРНЕ ПРОЕКТУВАННЯ ТА ТЕХНІЧНЕ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	24
2.1 Обґрунтування топології безпроводової мережі та математичне моделювання розповсюдження радіосигналу.....	24
2.2 Обґрунтування вибору та техніко-інженерний аналіз апаратного забезпечення системи.....	28
2.3 Проектування програмної інфраструктури та архітектури обміну даними..	30
2.4 Розробка та логічна формалізація функціональних алгоритмів автоматизації опалення.....	33
2.5. Аналіз кібербезпеки та захисту IoT-системи розумного опалення.....	35
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	40
3.1 Розгортання серверної інфраструктури у контейнеризованому середовищі та конфігурування брокера повідомлень.....	40
3.2 Практичне налаштування програмного шлюзу Zigbee2MQTT та інтеграція безпроводової периферії.....	44
3.3 Оцінка економічної ефективності впровадження системи розумного опалення.....	47
3.4 Програмна реалізація та конфігурування сценаріїв автоматизації в середовищі Home Assistant.....	50
3.5 Експериментальний аналіз температурних профілів та оцінка енергоефективності розробленої системи.....	55
ВИСНОВКИ.....	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	63

ВСТУП

Актуальність теми. Стрімкий розвиток технологій Інтернету речей (IoT) та концепції інтелектуального життєвого простору (Smart Home) протягом останнього десятиліття докорінно змінив підходи до проектування, розгортання та експлуатації внутрішніх інженерних мереж будівель [1, 11]. Серед усіх систем життєзабезпечення житлових та офісних приміщень найбільш ресурсомісткою та критично важливою є система опалення. За статистичними даними Міністерства енергетики України та аналітичних європейських інститутів, витрати на підтримання теплового мікроклімату в холодний період року становлять від 60% до 70% від загального обсягу енергоспоживання домогосподарств [2]. У часи глобальної нестабільності ринків енергоносіїв, постійного зростання тарифів на паливні ресурси та електроенергію, а також загострення екологічних проблем, завдання максимізації енергоефективності опалювальних систем набуває першочергового значення.

Для України ця проблема є не просто економічно доцільною, а стратегічно та життєво необхідною. Масштабні руйнування об'єктів критичної інфраструктури, дефіцит генеруючих потужностей та нестабільність розподільчих електромереж вимагають від кожного споживача жорсткої оптимізації та автономізації використання ресурсів. Традиційні централізовані та індивідуальні системи опалення із ручним або найпростішим термостатичним регулюванням демонструють вкрай низьку ефективність. Вони працюють за статичними принципами, не здатні адаптуватися до тривалої відсутності людей у приміщенні, добових коливань зовнішньої температури, дії диференційованих за часом тарифів або аварійних обмежень. Це призводить до значного теплового перевитрачання, коли пусті кімнати чи цілі будівлі безглуздо прогріваються, що виснажує фінанси громадян та створює критичні пікові навантаження на національну енергосистему.

Впровадження інформаційних систем зонального (покімнатного) мікрокліматичного контролю на базі безпроводових IoT-технологій відкриває нові шляхи вирішення цієї кризи. Застосування безпроводових протоколів зв'язку є

ключовим інженерним фактором для масового впровадження таких рішень, оскільки вони дозволяють розгортати інтелектуальну мережу датчиків та виконавчих пристроїв у вже відремонтованих приміщеннях без проведення капітальних будівельних робіт і прокладання нових кабельних трас. Водночас, проектування таких систем у рамках спеціальності «Інформаційні системи та технології» вимагає відходу від закритих хмарних комерційних платформ, які несуть загрозу конфіденційності даних та стають повністю непрацездатними при зникненні зовнішнього Інтернету [14]. Дослідження та розробка локальних, автономних архітектур розумного опалення з відкритим вихідним кодом є надзвичайно актуальним науково-практичним завданням для підвищення енергетичної незалежності та стійкості домогосподарств.

Мета роботи. Метою бакалаврської кваліфікаційної роботи є теоретичне моделювання, архітектурне проектування та практична розробка децентралізованої автономної інформаційної системи інтелектуального керування опаленням у структурі Smart Home. Розроблене рішення має функціонувати на основі енергоефективних безпроводових протоколів зв'язку та забезпечувати комплексну оптимізацію споживання енергоресурсів, автоматичне зниження теплових втрат житлового простору, а також підвищення загального рівня експлуатаційного комфорту й відмовостійкості домогосподарства без необхідності модернізації капітальних інженерних комунікацій будівлі.

Для досягнення поставленої мети у роботі сформульовано та послідовно вирішено такі завдання:

- Проаналізувати теоретичні засади побудови сучасних систем мікрокліматичного контролю в контексті парадигми Інтернету речей (IoT), вивчити їхні архітектурні рівні та виявити ключові недоліки комерційних закритих рішень.
- Провести системний порівняльний аналіз безпроводних інтерфейсів передачі даних (Wi-Fi, Bluetooth Low Energy, Z-Wave, ZigBee) за критеріями енергоспоживання, топології мережі, радіусу дії та стійкості до завад з метою обґрунтування оптимального технологічного стеку.

- Розробити структурно-функціональну схему та загальну дворівневу архітектуру взаємодії елементів розумного опалення на базі локального сервера автоматизації, що функціонує незалежно від зовнішніх хмарних сервісів.
- Обґрунтувати вибір та провести техніко-інженерне координування апаратного забезпечення системи, включаючи обчислювальний контролер, безпроводні сенсори клімату, датчики стану конструкцій та виконавчі смарт-термостати.
- Спроекувати програмну інфраструктуру обміну даними на основі полегшеного протоколу MQTT та програмного шлюзу декодування пакетів, забезпечивши сумісність периферійних пристроїв різних виробників у єдиному інформаційному полі [13].
- Розробити математичні моделі терморегуляції, а також логічні алгоритми та YAML-сценарії автоматизації для базових режимів експлуатації житла («Нічний режим», «Нікого немає вдома», «Захист від провітрювання»).
- Оцінити економічну доцільність, терміни окупності та екологічну ефективність впровадження розробленої інформаційної технології в умовах сучасного стану енергетичного сектору України.

Об'єкт дослідження – процес автоматизованого моніторингу, збору телеметричних даних та покімнатного (зонального) мікрокліматичного регулювання параметрів опалення в житловому просторі.

Предмет дослідження – архітектурні моделі, безпроводові інтерфейси зв'язку (зокрема протокол ZigBee 3.0), програмно-апаратні засоби інтеграції (Home Assistant, MQTT) та алгоритми логічного управління сценаріями автоматизації систем розумного опалення.

Методи дослідження. Для розв'язання поставлених у кваліфікаційній роботі завдань та забезпечення наукової обґрунтованості результатів було застосовано комплексний науково-методологічний апарат інформаційних технологій. Зокрема, у роботі використано:

- методи системного та порівняльного аналізу — при дослідженні сучасних тенденцій розвитку Інтернету речей, аналізі стеків безпроводових протоколів

зв'язку та виявленні архітектурних недоліків існуючих комерційних систем мікроклімату;

- методи структурно-функціонального моделювання інформаційних систем — для проектування дворівневої топології локальної мережі IoT та оптимізації інформаційних потоків між сенсорами, координатором і сервером;
- методи алгоритмізації та логічного проектування — для розробки поведінкових сценаріїв автоматизації опалення, формалізації логіки кінцевих автоматів (State Machine) та математичного опису релейного мікрорегулювання середовища;
- методи експериментального тестування та комп'ютерного імітаційного моделювання — для оцінки часових затримок, стабільності доставки пакетів даних і аналізу відмовостійкості системи в локальному програмному середовищі.

Наукова новизна одержаних результатів полягає у подальшому розвитку та математично-алгоритмічному вдосконаленні архітектурних підходів до побудови децентралізованих, повністю незалежних від зовнішніх хмарних платформ локальних IoT-систем мікрокліматичного контролю. На відміну від існуючих рішень, обґрунтовано та практично реалізовано модель інтеграції різнорідних безпроводових периферій різних виробників через некомерційний відкритий шлюз Zigbee2MQTT та локальний брокер Mosquitto MQTT в єдине логічне ядро сервера Home Assistant. Це дозволило мінімізувати час відгуку виконавчих механізмів до 1.5–2 секунд та забезпечити стовідсоткову автономність виконання автоматизованих кліматичних сценаріїв в умовах нестабільності або повної відсутності зовнішнього інтернет-з'єднання.

Практична значущість одержаних результатів. Спроектowana інформаційна технологія та розроблені на її основі програмно-апаратні рішення є готовим до розгортання інженерним модулем зонального керування опаленням, який може бути впроваджений у житлових або офісних приміщеннях без капітальної модернізації наявних інженерних мереж. Створені YAML-конфігурації автоматизацій («Нічний режим», «Нікого немає вдома», «Захист від провітрювання») мають високий ступінь масштабованості. Практичне

впровадження розробленої системи дозволяє досягти реального зниження фінансових витрат на опалення на 15–30% [2]. Матеріали роботи можуть бути використані як технологічна база для розробки суміжних систем енергоефективного Smart Home, а також у навчальному процесі при вивченні системної інтеграції та моделювання IoT.

Апробація результатів та публікації. Основні результати дослідження були апробовані на студентських науково-практичних конференціях, за результатами яких опубліковано тези доповіді «Розумне опалення в системі “Розумний дім”», а також підготовлено тези «Економічна та екологічна ефективність систем розумного опалення в контексті Smart Home» [23]..

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМ РОЗУМНОГО ОПАЛЕННЯ

1.1 Сутність інтелектуального опалення та його функціональна роль у структурі Smart Home

Сучасний етап розвитку житлової та комерційної нерухомості характеризується глибокою інтеграцією інформаційно-комунікаційних технологій в інженерні системи життєзабезпечення. У цьому контексті концепція «Розумного будинку» (Smart Home) еволюціонувала від набору ізольованих гаджетів побутової автоматизації до комплексних, багаторівневих децентралізованих інформаційних систем [1, 11, 14]. Одним із найбільш критичних, наукомістких та ресурсомістких сегментів такої екосистеми є інтелектуальне (розумне) опалення. Під системою розумного опалення (Smart Heating System) в парадигмі Інтернету речей (IoT) слід розуміти сукупність взаємопов'язаних цифрових датчиків, виконавчих механізмів, мережевих шлюзів та обчислювальних контролерів, які функціонують під управлінням єдиного логічного ядра з метою безперервного моніторингу, аналізу та адаптивного регулювання параметрів теплового мікроклімату в режимі реального часу [8, 9].

Місце підсистеми терморегуляції в загальній архітектурі Smart Home визначається її безпосереднім зв'язком з енергетичним балансом будівлі та рівнем біофізичного комфорту людини. На відміну від традиційних систем опалення, де управління здійснюється на основі фіксованих механічних налаштувань або показників одного релейного термостата на весь будинок, інтелектуальне рішення інтегрується в єдине інформаційне поле інших підсистем: моніторингу присутності, геолокації користувачів, контролю стану вікон та дверей, а також прогнозних погодних сервісів через прикладні програмні інтерфейси (API). Це перетворює опалення з пасивного споживача ресурсів на активний, контекстно-залежний елемент цифрової інфраструктури, здатний гнучко перерозподіляти теплову енергію між окремими локаціями (кліматичними зонами)

залежно від поточних потреб, часових профілів та експлуатаційних умов, мінімізуючи втрати без зниження комфорту.

З погляду системного аналізу та спеціальності 126 «Інформаційні системи та технології», розумне опалення розглядається як кіберфізична система (Cyber-Physical System, CPS), у якій фізичні процеси теплопередачі та гідродинаміки теплоносія тісно переплетені з алгоритмічними обчисленнями, цифровою обробкою сигналів та мережевою взаємодією. Вона забезпечує перехід від централізованого до розподіленого (зонального) управління, де кожне приміщення розглядається як окремий об'єкт автоматизації із власними збуреннями (інсоляція, тепловиділення від побутової техніки) та керуючими впливами. Така інтеграція дозволяє збирати великі масиви телеметричних даних, формувати локальні бази метрик і створювати підґрунтя для застосування методів інтелектуального аналізу даних (Data Mining) та предиктивного моделювання теплових втрат споруд.

1.2 Техніко-технологічна класифікація та архітектурний аналіз компонентів IoT-системи опалення

Для глибокого розуміння кібер фізичної природи системи розумного опалення необхідно провести декомпозицію її структури на базові елементи та класифікувати їх за функціональним призначенням у загальному контурі управління. Будь-яка інформаційна система Інтернету речей (IoT), спрямована на терморегуляцію житлового простору, складається з чотирьох взаємопов'язаних технологічних груп компонентів: сенсорної периферії, виконавчих механізмів, комунікаційних шлюзів та централізованого обчислювального ядра [1]. Кожна група виконує специфічну роль у процесі перетворення фізичних параметрів середовища у цифрові сигнали, їх транспортування, логічної обробки та зворотного фізичного впливу на об'єкт автоматизації.

- Першу групу становлять сенсорні компоненти (датчики), які виконують функцію первинного збору телеметричної інформації про стан об'єкта. У системах

розумного опалення ключовими є безпроводні датчики температури та вологості повітря, а також бінарні датчики стану огорожувальних конструкцій (відкриття/закриття віконних рам та дверей).

З погляду інформаційних технологій, сучасні IoT-сенсори — це автономні мікросистемні пристрої, які містять інтегрований аналогово-цифровий перетворювач (АЦП), контролер живлення та малопотужний радіопередавач.

Їх класифікують за методом опитування: періодичний (передача даних через фіксовані інтервали часу) або за подією. Для максимізації енергоефективності в безпроводових мережах застосовують саме подієвий метод: датчик перебуває в режимі глибокого сну (Deep Sleep) і активується лише при зміні температури на задану величину (дельта-поріг, наприклад, $\pm 0.1^{\circ}\text{C}$) або при фізичному розмиканні магнітного контакту (геркона) на вікні.

- Другу групу утворюють виконавчі механізми, представлені розумними радіаторними термостатами (TRV — Thermostatic Radiator Valve). Вони класифікуються за типом управління (бінарні — «відкрито/закрито», ступінчасті та пропорційні) й конструктивним виконанням приводу. У сучасних архітектурах використовуються цифрові TRV з інтегрованими кроковими електродвигунами (сервоприводами), які здатні плавно змінювати положення штока клапана від 0% до 100%. Особливістю цих компонентів є наявність вбудованого мікроконтролера, який не лише приймає зовнішні цифрові команди від сервера автоматизації, але й передає зворотну телеметрію: поточний відсоток відкриття клапана, рівень заряду елементів живлення та внутрішні коди помилок (наприклад, заклинювання механічного вузла).

- Третя група — комунікаційні шлюзи або координатори мережі, які виконують роль апаратних мостів між гетерогенними радіо мережами периферії (наприклад, ZigBee чи Z-Wave) та стандартними провідними інтерфейсами локальної мережі (USB, Ethernet, Wi-Fi) [19]. Загальну трирівневу архітектуру яку використовують IoT-системи розумного опалення та логіку взаємодії її компонентів представлено на рис. 1.1.

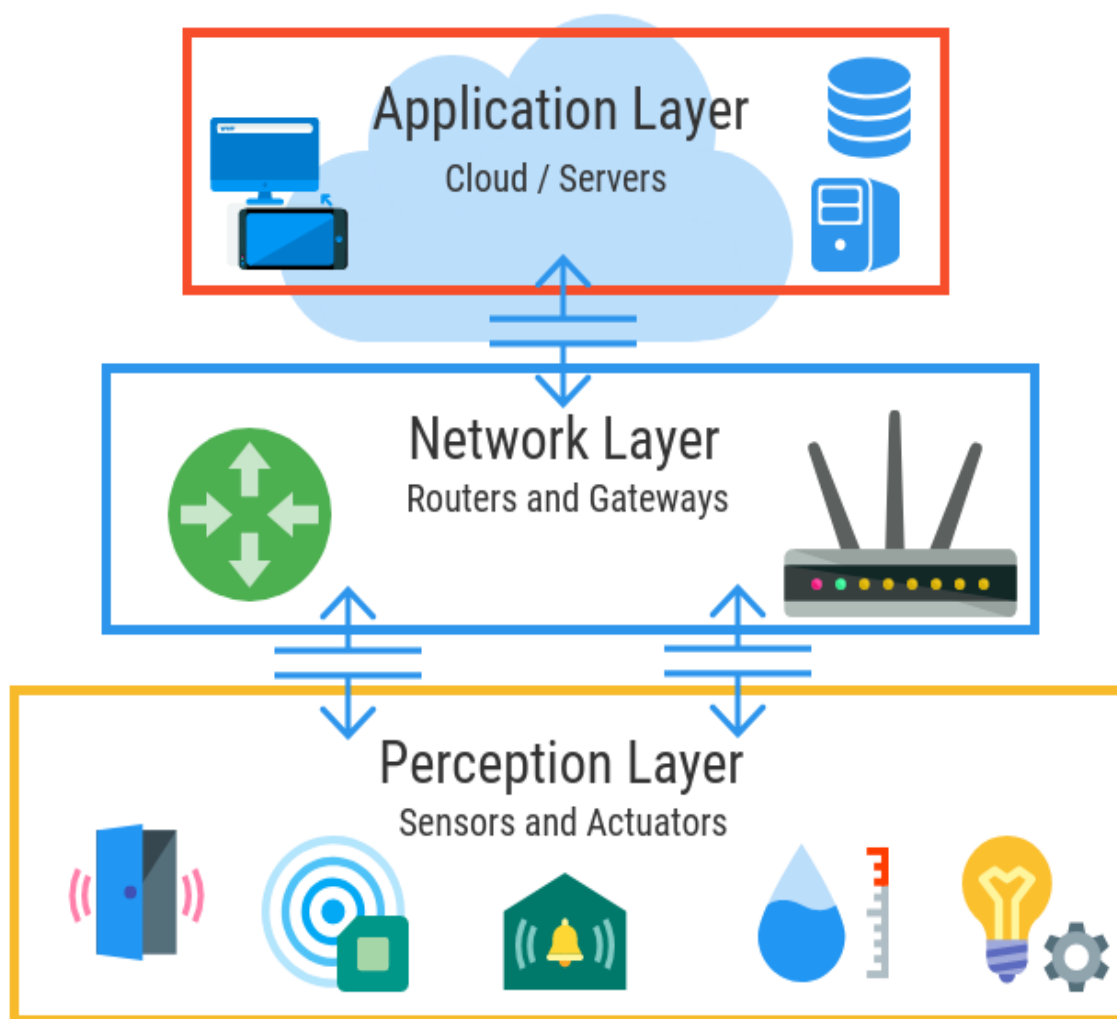


Рис. 1.1 Структурна схема архітектури IoT-системи розумного опалення.

Четверту, найвищу групу, становить центральний обчислювальний сервер. Це апаратна платформа (одноплатний мікрокомп'ютер або промисловий контролер), на якій розгортається операційне середовище автоматизації, база даних для збереження часових рядів (Time-Series Database) та аналітичні модулі. Сервер є логічним ядром системи: він збирає інформацію з усіх точок вимірювання, обробляє її відповідно до закладених математичних моделей і генерує керуючі впливи для виконавчих механізмів, замикаючи контур зворотного зв'язку. Загальну функціональну структуру, роль та методи взаємодії всіх чотирьох

технологічних груп компонентів проектованої IoT-системи розумного опалення зведено в табл. 1.1.

Таблиця 1.1

Функціональна структура та компоненти IoT-системи розумного опалення

Група компонентів	Основна функція в системі	Тип пристроїв / Приклад у роботі	Метод функціонування / Обмін даними
Сенсорна периферія	Первинний збір телеметричної інформації про поточний стан об'єкта.	Безпроводні датчики температури, вологості, стану вікон (геркони).	Подієвий метод опитування (передача даних при зміні дельта-порогу або події).
Виконавчі механізми	Фізичний зворотний вплив на об'єкт автоматизації (регулювання теплоносія).	Розумні радіаторні термостати (TRV) з кроковими електродвигунами	Прийом команд керування (0-100% відкриття) та передача зворотної телеметрії стану.
Комунікаційні шлюзи	Апаратний міст між гетерогенними мережами пристроїв та локальною мережею.	Координатор мережі ZigBee, програмний шлюз Zigbee2MQTT	Декодування пакетів даних, трансляція протоколів периферії у стандартні інтерфейси.
Обчислювальний сервер	Централізоване логічне ядро системи, аналіз даних та ухвалення рішень.	Одноплатний мікрокомп'ютер із сервером Home Assistant та брокером MQTT.	Збір метрик, збереження часових рядів (база даних), виконання сценаріїв автоматизації.

1.3 Історія розвитку та сучасні тенденції систем автоматизації опалення

Еволюція систем автоматизації та керування тепловим мікрокліматом пройшла тривалий і складний шлях від найпростіших механічних пристроїв до високотехнологічних розподілених кіберфізичних комплексів. Першим етапом розвитку (середина XX століття) стало масове впровадження механічних термостатичних радіаторних клапанів, які працювали на фізичному принципі об'ємного розширення термочутливої рідини або газу всередині сильфона. Такі пристрої забезпечували виключно статичне підтримання заданого рівня температури в одній локальній точці та повністю залежали від ручних маніпуляцій користувача, що унеможливлювало будь-яку динамічну або гнучку оптимізацію витрат паливних ресурсів.

Другий етап розвитку (кінець XX – початок XXI століття) ознаменувався цифровою революцією в мікроелектроніці, появою перших електронних програмованих термостатів та інтеграцією провідних протоколів промислової автоматизації будівель (зокрема KNX, LonWorks, Modbus). Впровадження цифрових контролерів дозволило реалізувати базові часові профілі обігріву — наприклад, автоматичне зниження цільової температури в офісних приміщеннях у вихідні дні або вночі у житлових будинках. Проте, висока вартість обладнання, складність інженерного монтажу кабельних трас зв'язку та закрита архітектура комерційних виробників обмежували масове розгортання таких систем у широкому житловому секторі, залишаючи їх прерогативою елітної комерційної нерухомості.

Третій, сучасний етап розвитку розпочався у 2010-х роках із концептуальним обґрунтуванням та бурхливим розгортанням парадигми Інтернету речей (IoT). Поява безпроводних інтерфейсів та хмароорієнтованих рішень (зокрема відомих комерційних продуктів компаній Nest, Hive, ecobee чи Danfoss Ally) перевела терморегуляцію у площину інтелектуального аналізу даних. Кліматичні системи отримали здатність підключатися до глобальної мережі, аналізувати поточні метеорологічні прогнози через веб-сервіси та

використовувати базові алгоритми машинного навчання для автоматичного вивчення щоденних звичок та графіків переміщення мешканців будинку.

На сьогодні ключовою світовою та вітчизняною тенденцією у сфері smart-опалення є зміщення вектору від закритих хмарозалежних екосистем до відкритих (Open Source) платформ локальної автоматизації (таких як Home Assistant чи openHAB) [4, 7]. Це зумовлено необхідністю забезпечення високого рівня конфіденційності, захисту внутрішньої телеметрії від кіберзагроз, а головне — гарантування стовідсоткової локальної відмовостійкості системи під час тривалих аварійних відключень електроенергії та відсутності зовнішнього інтернет-зв'язку. Іншими провідними трендами є інтеграція клімат-контролю з відновлюваними джерелами енергії (сонячними станціями, тепловими насосами) для акумуляції тепла в періоди мінімальних тарифів, а також впровадження предиктивних моделей штучного інтелекту, які здатні динамічно прораховувати тепло інерційні властивості стін кожної конкретної будівлі [10].

1.4 Аналіз інтелектуальних алгоритмів та методів керування мікрокліматом

Ефективність функціонування сучасної системи розумного опалення безпосередньо залежить від математичних методів та алгоритмів, покладених в основу процесу прийняття рішень обчислювальним сервером. Саме алгоритмічне забезпечення визначає точність підтримання температурного режиму, швидкість реакції системи на зміни параметрів середовища та загальну стабільність роботи комплексу автоматизації. В інженерній практиці автоматизації кліматичних систем виділяють три базові класи алгоритмів регулювання: релейні (двопозиційні), пропорційно-інтегрально-диференціальні (ПІД) та предиктивно-контекстні (адаптивні) методи керування. Кожен із зазначених підходів має власні особливості реалізації, рівень складності та сферу практичного застосування в системах керування опаленням. Функціональну схему контуру автоматичного регулювання температури із застосуванням ПІД-алгоритму та механізму зворотного зв'язку наведено на рис. 1.2.

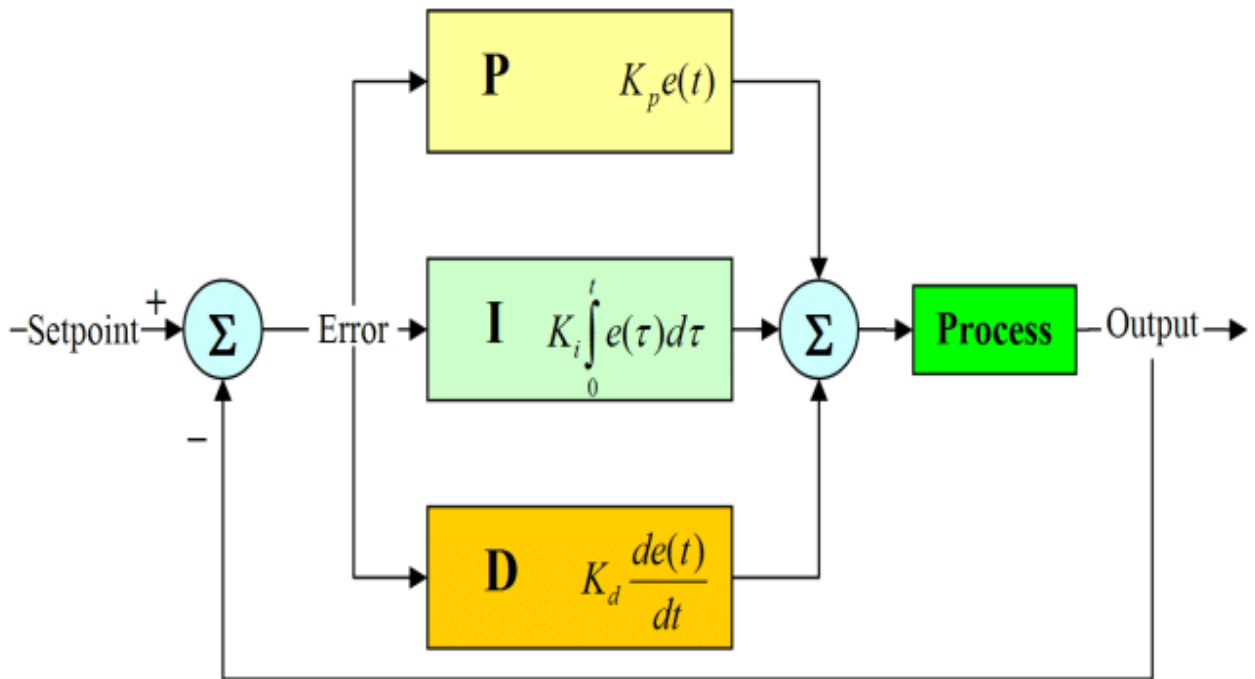


Рис. 1.2 Блок-схема контуру керування термостатом із ПІД-регулятором.

Елементи схеми: Задана температура (Setpoint) → Суматор (обчислення помилки $e(t)$) → Блок ПІД-обчислень (Пропорційна, Інтегральна, Диференціальна складові) → Керуючий сигнал (0–100%) → Сервопривід клапана радіатора (TRV) → Об'єкт обігріву (Кімната) → Датчик температури ZigBee (Зворотний зв'язок на суматор) [8]. Кожен із цих підходів має свої обмеження, специфічні сфери застосування, а також апаратно-програмні вимоги для їх реалізації в межах платформ Інтернету речей.

Найбільш поширеним та найпростішим у реалізації є релейний алгоритм (двопозиційне регулювання з гістерезисом). Його сутність полягає у бінарному керуванні виконавчим механізмом: котел або клапан радіатора повністю вмикається, якщо поточна температура падає нижче заданої уставки, і повністю вимикається, коли температура перевищує її. Для запобігання надто частому перемиканню (явищу «брязкання» контактів або сервоприводів), яке призводить до передчасного зносу обладнання та перевитрати енергії автономних елементів живлення у безпроводових пристроях, в алгоритм обов'язково закладається температурний гістерезис (зазвичай у діапазоні від $\pm 0.2^\circ\text{C}$ до $\pm 0.5^\circ\text{C}$). Попри свою

обчислювальну простоту, релейний метод має суттєвий недолік — помітну амплітуду коливань температури навколо цільового значення, що зумовлено високою тепловою інерційністю опалювальних приладів та огорожувальних конструкцій приміщення [6].

Для точнішого підтримання мікроклімату застосовують ПІД-регулювання (Proportional-Integral-Derivative control). Цей алгоритм формує динамічний керуючий сигнал (наприклад, точний відсоток відкриття штока розумної термоголовки від 0% до 100%), який базується на трьох складових: поточній помилці розбіжності температур (пропорційна), накопиченій помилці за попередній час (інтегральна) та швидкості зміни цієї похибки (диференціальна). ПІД-алгоритм дозволяє мінімізувати амплітуду коливань і плавно підлаштовувати теплову потужність радіатора під реальні поточні втрати кімнати [6]. Проте його впровадження у безпроводових IoT-мережах ZigBee суттєво обмежується жорсткими вимогами до високої частоти опитування датчиків та постійного двостороннього обміну даними, що призводить до швидкого критичного розряду батарейок у сенсорах.

Сучасний розвиток інформаційних систем привів до появи предиктивно-контекстних методів керування, які функціонують на вищому рівні логіки сервера Smart Home. Вони не замінюють базові фізичні регулятори, а динамічно коригують їхні цільові установки (T_{set}) на основі аналізу зовнішніх інформаційних потоків. До таких методів належать алгоритми погодозалежного керування (взаємодія з метеорологічними API для упередження охолодження будівлі при різкому пориві вітру чи похолоданні) та алгоритми відстеження контексту присутності (моніторинг геолокації смартфонів користувачів, інтеграція з датчиками руху та магнітними контактами дверей). Поєднання класичного зонального регулювання на рівні окремих кімнат із контекстними сценаріями вищого рівня в середовищі Home Assistant дозволяє досягти максимального синергетичного ефекту, забезпечуючи гнучкість системи та реальне енергозбереження [2, 10, 13, 15]. Детальний порівняльний аналіз різних підходів до регулювання мікроклімату наведено в табл. 1.2.

Таблиця 1.2

Порівняльний аналіз алгоритмів керування мікрокліматом в системах опалення

Критерій порівняння	Релейний (двопозиційний) алгоритм	ПД-алгоритм	Предиктивно-контекстний метод
Принцип роботи	Бінарне керування («увімкнено/вимкнено») при перетині температурних порогів з урахуванням гістерезису.	Формування динамічного сигналу (0–100% відкриття клапана) на основі поточної, накопиченої помилки та швидкості її зміни.	Динамічне коригування цільових уставок на основі аналізу зовнішніх інформаційних потоків.
Точність підтримання температури	Низька/середня (присутні помітні коливання навколо заданої точки через теплову інерцію).	Висока (мінімальна амплітуда коливань, плавне підлаштування під втрати тепла).	Дуже висока (дозволяє упереджувати температурні зміни в приміщенні).
Обчислювальна складність	Дуже низька (мінімальні вимоги до ресурсів контролера).	Середня (потребує регулярного розрахунку математичних складових у реальному часі).	Висока (інтеграція з зовнішніми API, обробка геоданих, складна логіка сценаріїв).
Вплив на автономність пристроїв (багатарею)	Низький (датчики можуть перебувати в режимі глибокого сну і прокидатися лише за подією).	Високий (вимагає високої частоти опитування датчиків і постійного обміну даними).	Залежить від базового фізичного регулятора; сам по собі оптимізує графіки роботи.

Продовження таблиці 1.2

Критерій порівняння	Релейний (двопозиційний) алгоритм	ПД-алгоритм	Предиктивно-контекстний метод
Переваги	Простота реалізації, надійність, низькі вимоги до апаратної частини та обсягу трафіку.	Максимальна точність стабілізації температури, відсутність різких температурних коливань.	Гнучкість, максимальна енергоефективність завдяки врахуванню контексту присутності людей та погоди.
Недоліки	Перевитрата енергії через інерційність опалювальних приладів, амплітудні коливання температури.	Швидкий розряд безпроводових сенсорних пристроїв ZigBee, складність точного налаштування коефіцієнтів.	Потребує постійного підключення до мережі інтернет (для погодних API) або складних локальних сценаріїв.

РОЗДІЛ 2 АРХІТЕКТУРНЕ ПРОЕКТУВАННЯ ТА ТЕХНІЧНЕ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Обґрунтування топології безпроводової мережі та математичне моделювання розповсюдження радіосигналу

Перехід від аналітичного огляду теоретичних засад до безпосереднього інженерного проектування інформаційної системи розумного опалення вимагає чіткого визначення топології безпроводової мережі та математичного опису процесів передачі даних у закритому просторі житлового приміщення [6].

Для забезпечення високої відмовостійкості та стабільного покриття всіх кліматичних зон об'єкта автоматизації у роботі обрано комірчасту топологію (Mesh-мережу) на базі протоколу ZigBee 3.0 [12]. Використання такої архітектури дозволяє підтримувати стабільний обмін даними між компонентами системи навіть у випадку тимчасового погіршення якості радіозв'язку або виходу з ладу окремих вузлів мережі [12]. На відміну від топології «зірка», характерної для Wi-Fi мереж, де вихід із ладу або затінення центрального вузла повністю руйнує зв'язок із периферією, Mesh-архітектура має властивість самоорганізації та самовідновлення маршрутів передачі даних [3]. У разі виникнення перешкод система здатна автоматично перебудовувати маршрути доставки пакетів через інші доступні пристрої мережі, що суттєво підвищує загальну надійність функціонування інформаційної інфраструктури. У такій мережі пристрої поділяються на три логічні ролі: координатор (центральный шлюз системи), роутери (пристрої з постійним живленням від електромережі, які ретранслюють пакети) та кінцеві пристрої (End Devices — автономні датчики та термоголовки, які більшість часу перебувають у режимі сну). Такий розподіл функцій дозволяє оптимізувати навантаження на мережу та забезпечити енергоефективну роботу автономних сенсорних пристроїв. Для унаочнення переваг стійкості мережі до відмов та демонстрації відмінностей між топологіями «Зірка» та «Mesh» (комірчаста) їх порівняння наведено на рис. 2.1.

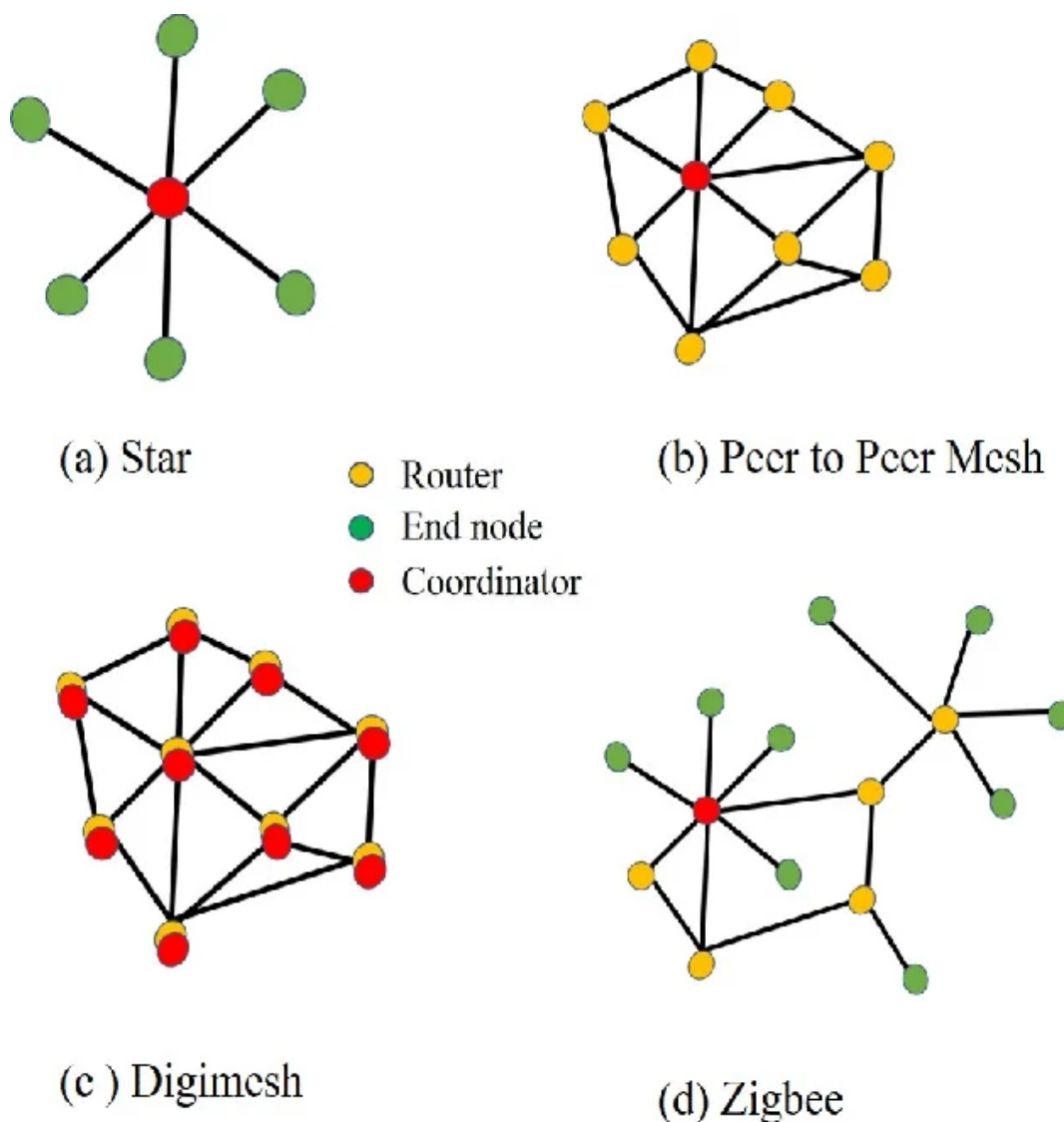


Рис. 2.1 Порівняння топологій радімережі: а) топологія «Зірка»; б) комірчаста топологія «Mesh» стандарту ZigBee

Якщо прямий радіоканал між датчиком температури та координатором блокується залізобетонною перешкодою, система автоматично перебудовує маршрут доставки даних через найближчий роутер. Проте, проектування безпроводного середовища у діапазоні 2.4 ГГц всередині будівель ускладнюється явищем багатопроменевого поширення, відбиття та загасання радіохвиль при проходженні крізь стіни та міжповерхові перекриття [3]. Для комплексного інженерного аналізу та обґрунтування вибору оптимального стеку протоколів було виконано порівняльний аналіз безпроводних інтерфейсів передачі даних для IoT-систем опалення, результати якого зведено в табл. 2.1.

Порівняльний аналіз безпроводних інтерфейсів передачі даних для
IoT-систем опалення

Характеристика	ZigBee 3.0	Wi-Fi (IEEE 802.11)	Bluetooth Low Energy (BLE)	Z-Wave
Частотний діапазон	2.4 ГГц	2.4 ГГц / 5 ГГц	2.4 ГГц	868 МГц (для Європи/України)
Топологія мережі	Комірчаста (Mesh) — самоорганізується та ретранслює дані.	«Зірка» (пряме підключення всіх пристроїв до однієї точки доступу).	Переважно «Зірка» (топологія точка-точка або зірка для периферії).	Комірчаста (Mesh) з ретрансляцією пакетів.
Радіус дії в приміщенні	До 10–30 м (масштабується завдяки роутерам мережі).	До 20–50 м (проте сильно затухає за перешкодами).	До 10–15 м (обмежений радіус для стабільного збору телеметрії).	До 30–50 м (вища проникна здатність завдяки частоті).
Енергоспоживання	Дуже низьке (ідеально для автономних датчиків та термоголовок).	Високе (практично не підходить для тривалої роботи пристроїв від батарейок).	Низьке (підходить для дрібних сенсорів, але обмежене для постійного керування).	Низьке (забезпечує тривалу автономність кінцевих вузлів).
Стійкість до завад та надійність	Висока завдяки властивості самовідновлення та динамічній перебудові маршрутів.	Низька/середня в щільній забудові через перевантаженість каналів 2.4 ГГц.	Середня (чутливий до завад від суміжних Wi-Fi мереж).	Висока (діапазон 868 МГц практично вільний від побутового радіосміття).

Характеристи ка	ZigBee 3.0	Wi-Fi (IEEE 802.11)	Bluetooth Low Energy (BLE)	Z-Wave
Сумісність та інтеграція	Висока при використанні відкритих шлюзів (напр. Zigbee2MQTT [16]).	Часто обмежена пропріетарни ми хмарними платформами виробників.	Інтеграція та обмеження на кількість одночасно підключених пристроїв.	Обмежена ліцензування м стандарту, висока вартість обладнання.

Для теоретичного моделювання рівня потужності сигналу та визначення оптимального розташування координатора і роутерів застосовано логарифмічно-дистанційну модель затухання (Log-Distance Path Loss Model) з урахуванням впливу перешкод на поширення радіосигналу. Математична модель визначення втрат потужності сигналу описується формулою (2.1):

$$PL(d) = PL(d_0) + 10n \log_{10}\left(\frac{d}{d_0}\right) \quad (2.1)$$

де $PL(d)$ — втрати потужності сигналу на відстані (d) , дБ; $PL(d_0)$ — базове згасання сигналу у вільному просторі на еталонній відстані $(d_0 = 1)$ м (для частоти 2.4 ГГц $PL(d_0) \approx 40$) дБ; (n) — показник згасання траси, який для закритих житлових приміщень зазвичай перебуває в межах 2.4–3.5; (d) — реальна відстань між пристроями, м.

Додатково на рівень сигналу впливають фізичні перешкоди у приміщенні. Для залізобетонних капітальних стін товщиною 15–20 см втрати сигналу можуть становити приблизно 12–15 дБ, тоді як для гіпсокартонних міжкімнатних перегородок — 3–5 дБ.

Розрахунок за цією моделлю показує, що при потужності передавача $(P_{\text{Tx}}=20)$ дБм та наявності двох залізобетонних стін на відстані 10 м рівень сигналу може знижуватися приблизно до -85 дБм. Оскільки поріг чутливості сучасних приймачів ZigBee становить -95...-100 дБм, запас потужності є

мінімальним, що обґрунтовує необхідність використання щонайменше 1–2 пристроїв-роутерів для забезпечення стабільного зв'язку у віддалених приміщеннях [3].

Залежність втрат потужності радіосигналу (path loss) від відстані між координатором та кінцевими пристроями ZigBee в умовах залізобетонних перекриттів графічно зображено на рис. 2.2.

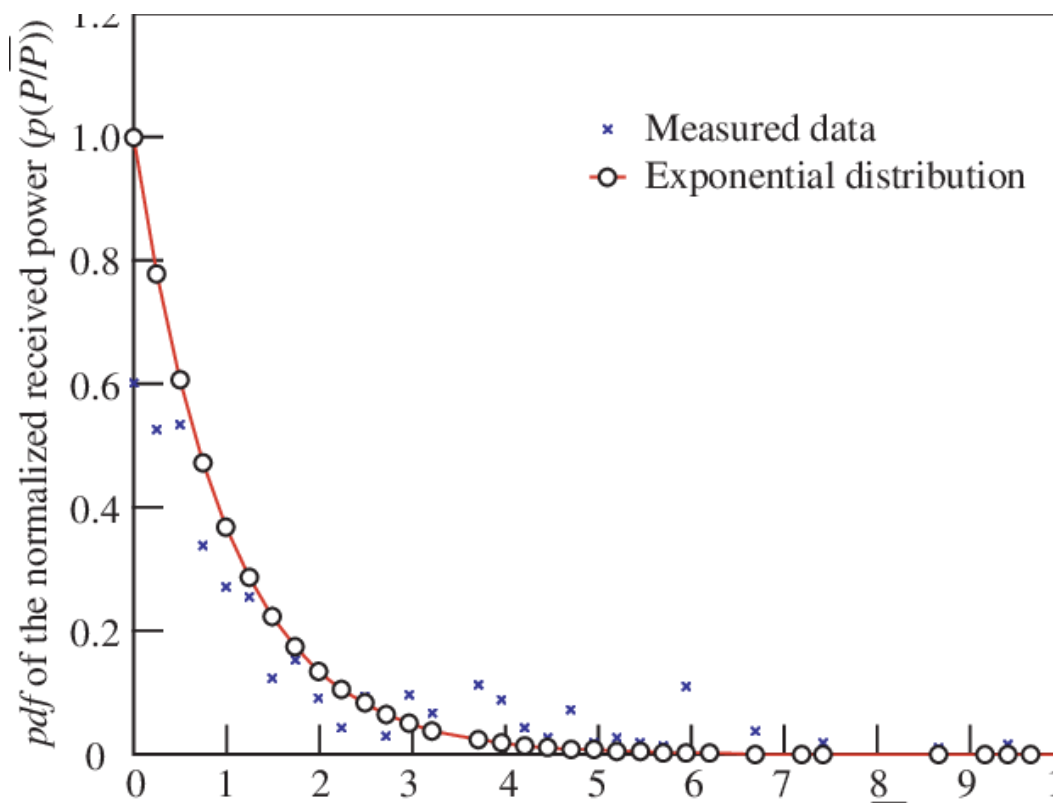


Рис. 2.2 Графік залежності згасання сигналу стандарту IEEE 802.15.4 від відстані.

2.2 Обґрунтування вибору та техніко-інженерний аналіз апаратного забезпечення системи

Фізичний рівень (Physical Layer) проектувальної IoT-системи є матеріально-технічним базисом, що забезпечує безпосередню взаємодію інформаційних алгоритмів із реальними об'єктами житлового середовища. При побудові децентралізованої системи розумного опалення вибір апаратних компонентів здійснювався на основі системного аналізу їхніх технічних характеристик, критеріїв енергоефективності, обчислювальної потужності,

апаратної сумісності та економічної доцільності. Апаратний комплекс системи розділено на три функціональні рівні: обчислювальне логічне ядро, пристрої комунікаційного спряження та периферійні вузли (сенсори й виконавчі механізми) [9].

Як центральну обчислювальну платформу (локальний сервер автоматизації) обґрунтовано та обрано одноплатний мікрокомп'ютер Raspberry Pi 4 Model B з обсягом оперативної пам'яті 4 ГБ LPDDR4. Вибір цієї платформи зумовлений її високою продуктивністю при низькому рівні енергоспоживання (до 15 Вт при пікових навантаженнях). Серце архітектури мікрокомп'ютера — 64-бітний чотириядерний процесор Broadcom BCM2711 (архітектура ARM Cortex-A72) із тактовою частотою 1.5 ГГц [18]. Таких обчислювальних ресурсів цілком достатньо для безперебійного локального функціонування операційної системи Home Assistant OS, розгортання реляційних баз даних часових рядів (Time-Series Database), локального брокера повідомлень MQTT та одночасної обробки сотень інформаційних потоків від периферії без залучення зовнішніх хмарних обчислень. Для запобігання деградації носія даних, яка часто виникає в IoT-системах через високу інтенсивність операцій запису/читання логів, замість стандартних карт пам'яті MicroSD застосовано твердотільний накопичувач SSD ємністю 120 ГБ з інтерфейсом SATA III, підключений через високошвидкісний порт USB 3.0.

Для організації взаємодії між сервером та безпроводною Mesh-мережею обрано універсальний координатор Sonoff Zigbee 3.0 USB Dongle Plus (версія на базі мікрочіпа Texas Instruments CC2652P) [18]. Даний апаратний шлюз оснащений інтегрованим потужним радіочастотним підсилювачем, що забезпечує вихідну потужність передавача до +20 дБм, та зовнішньою всеспрямованою антеною з коефіцієнтом підсилення +5 дБи. Це безпосередньо корелює з математичною моделлю розповсюдження хвиль, наведеною у підрозділі 2.1, та дозволяє мінімізувати вплив залізобетонних перешкод. Підключення координатора до Raspberry Pi здійснюється через USB-подовжувач довжиною 1.5 метра, що є критично важливим для усунення високочастотних електромагнітних завад, які

генеруються портами USB 3.0 та вбудованим Wi-Fi модулем сервера на аналогічній частоті 2.4 ГГц.

Периферійний рівень представлений пристроями брендів Aqara та Tuuya, що працюють за стандартом IEEE 802.15.4 (ZigBee). Для збору кліматичних метрик у кожній кімнаті проектується встановлення безпроводних датчиків температури та вологості Aqara Temperature and Humidity Sensor (WSDCGQ11LM). Вони побудовані на базі енергоефективного мікроконтролера NXP JN5169 і живляться від літієвого елемента лінійки CR2032. Завдяки реалізації режиму глибокого сну (струм споживання < 1 мкА у пасивному стані), датчик забезпечує безперервну автономну роботу до 2 років. Бінарний контроль стану віконних прорізів покладено на датчики Aqara Door and Window Sensor (MCCGQ11LM), які використовують фізичний ефект замикання геркону під дією магнітного поля, миттєво генеруючи цифрове переривання при спробі провітрювання приміщення.

Роль виконавчих механізмів виконують розумні радіаторні термостати Tuuya Smart Radiator Actuator (TRV) [8]. Ці компоненти обладнані малопотужним редукторним кроковим електродвигуном, який за командою з сервера змінює положення штока термостатичного клапана опалення. Пристрій має вбудований цифровий мікроконтролер, що підтримує стандартні ZigBee-кластери управління терморегуляцією (Thermostatic Radiator Valve Cluster), передає на сервер поточний рівень заряду двох елементів живлення типу AA, точний відсоток відкриття клапана (від 0% до 100%) та внутрішню температуру, що дозволяє реалізувати гнучкі та багаторівневі алгоритми зворотного зв'язку.

2.3 Проектування програмної інфраструктури та архітектури обміну даними

Програмна інфраструктура проектувальної інформаційної системи розумного опалення розглядається як багаторівневий стек спеціалізованого програмного забезпечення, що забезпечує збір, маршрутизацію, нормалізацію, логічну обробку та візуалізацію телеметричних даних [4, 15]. Логічна архітектура системи побудована за децентралізованим принципом з пріоритетом локального

виконання обчислень (Local Control First). Ключовими елементами цього стеку є операційне середовище автоматизації Home Assistant, програмний шлюз Zigbee2MQTT та локальний брокер повідомлень Eclipse Mosquitto, які функціонують у контейнеризованому середовищі на базі ядра Linux локального сервера [5, 17].

Основою комунікаційної взаємодії між низькорівневою безпроводовою периферією та високорівневим сервером автоматизації є програмний міст Zigbee2MQTT [7]. Його функціональне призначення полягає в абстрагуванні від специфічних апаратних особливостей радіочіпів різних брендів та трансляції сирих пакетів даних протоколу ZigBee (стандарт IEEE 802.15.4) у стандартизовані текстові повідомлення формату JSON, що передаються через стек протоколів TCP/IP [5, 7]. Коли безпроводовий датчик температури фіксує зміну параметра середовища, він надсилає радіопакет на координатор. Шлюз Zigbee2MQTT перехоплює цей пакет, декодує специфічні ZigBee-кластери (наприклад, кластер вимірювання температури 0x0402) і форматує структурований JSON-об'єкт, який містить поточне значення температури, відносної вологості, унікальний IEEE-адрес пристрою та рівень заряду батареї [16].

Транспортування сформованих JSON-пакетів всередині локальної мережі сервера покладено на брокер повідомлень Eclipse Mosquitto, що реалізує полегшений протокол MQTT (Message Queuing Telemetry Transport) [17]. Протокол MQTT функціонує за шаблоном проєктування «видавець-підписник» (Publish-Subscribe) і є оптимальним для IoT-систем завдяки мінімальному розміру службового заголовка пакета [5]. У межах системи спроектовано ієрархічну структуру MQTT-топиків (тем). Наприклад, для сенсору в спальні публікація даних здійснюється в топик `zigbee2mqtt/bedroom_sensor`, а для керування термоголовкою радіатора — у `zigbee2mqtt/bedroom_trv/set`. Для гарантування доставки критично важливих керуючих впливів (команд на зміну положення клапанів опалення) в умовах завад застосовано рівень якості обслуговування QoS 1 (Quality of Service — гарантована доставка з підтвердженням PUBACK), тоді як для регулярної телеметрії датчиків використовується QoS 0.

Верхній логічний рівень системи становить ядро Home Assistant Core. Через вбудовану MQTT-інтеграцію сервер підписується на відповідні топіки брокера Mosquitto, отримує JSON-пакети, розпарсовує їх за допомогою JSONPath-виразів та оновлює стани відповідних віртуальних сутностей (Entities) у внутрішній базі даних станів (State Engine). Це дозволяє повністю абстрагувати логіку автоматизацій від конкретного фізичного інтерфейсу: для ядра автоматизацій датчик відображається як об'єкт `sensor.bedroom_temperature`, а розумний радіаторний клапан — як кліматичний об'єкт `climate.bedroom_heating`. Така трирівнева програмна інфраструктура забезпечує повну ізоляцію системи від зовнішніх хмарних серверів, мінімізує затримки передачі даних всередині сервера і гарантує високу кібербезпеку інформаційних потоків [21]. Фізичне розміщення програмних модулів на апаратних вузлах системи та інтерфейси взаємодії між ними відображено на UML-діаграмі розгортання (рис. 2.3) [13].

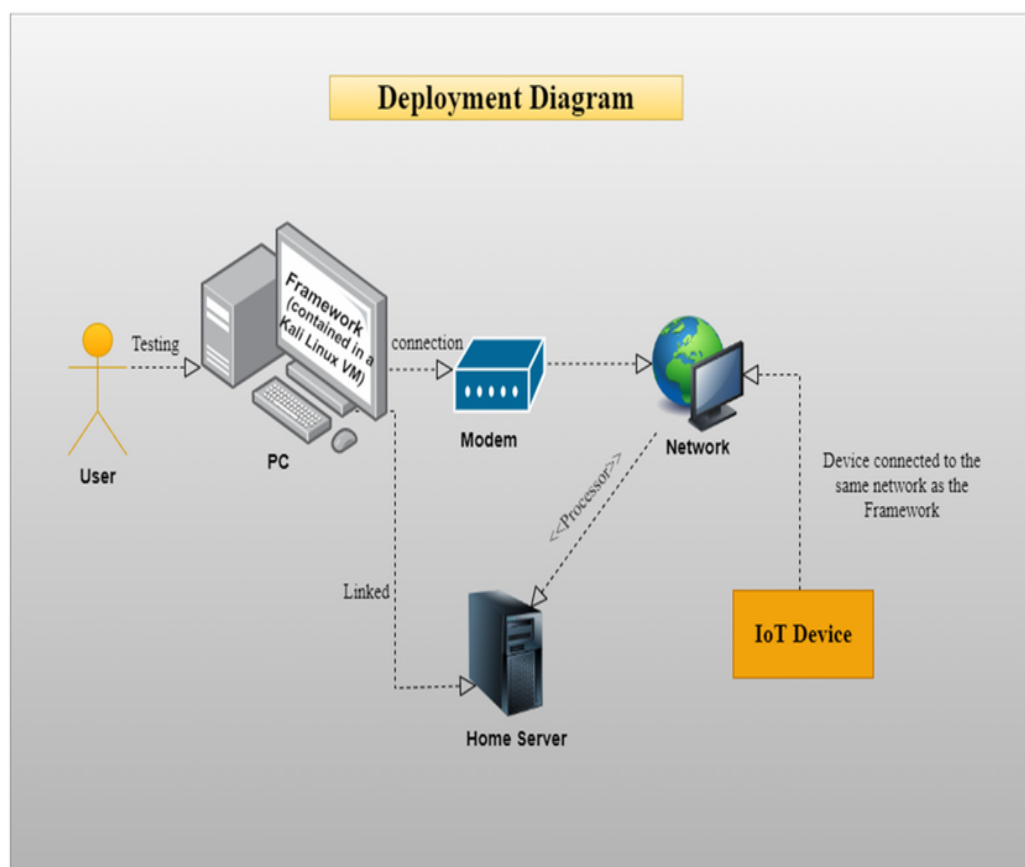


Рис. 2.3 UML-діаграма розгортання компонентів IoT-системи

Як видно з наведеної архітектури розгортання, всі ключові програмні модулі обчислювального ядра функціонують у контейнеризованому середовищі на базі

технології Docker. Використання контейнеризації дозволяє ізолювати процеси Home Assistant, Eclipse Mosquitto та Zigbee2MQTT на рівні операційної системи, забезпечуючи кожному сервісу окреме середовище виконання, власні залежності (libraries) та фіксований ліміт системних ресурсів. Мережева взаємодія між контейнерами реалізована через ізольовану віртуальну мережу типу «міст» (Docker Bridge Network). Це дозволяє брокеру MQTT та програмному шлюзу обмінюватися JSON-пакетами всередині замкненого мережевого контуру хоста без необхідності відкриття сервісних портів у загальну локальну мережу користувача (LAN), що суттєво мінімізує вектор потенційних мережевих атак.

Така топологія розгортання гарантує високу відмовостійкість, гнучкість та модульність проектованої інформаційної системи. У разі виникнення критичного збою або під час планового перезапуску окремого сервісу (наприклад, примусового оновлення конфігурації шлюзу чи додавання нових безпроводних пристроїв), інші компоненти стеку продовжують функціонувати у штатному режимі. Декларативний опис інфраструктури за допомогою інструменту Docker Compose дозволяє зберегти всі налаштування середовища в єдиному конфігураційному файлі. Це спрощує процес адміністрування, автоматизує створення повних backup-архівів системи та забезпечує можливість миттєвого перенесення (міграції) всього програмного комплексу на іншу апаратну платформу в разі апаратної відмови поточного сервера.

2.4 Розробка та логічна формалізація функціональних алгоритмів автоматизації опалення

Керуючий контур проектувальної інформаційної системи базується на концепції подійно-орієнтованого керування, де логіка прийняття рішень описується через кінцеві автомати (State Machine), розгорнуті в декларативному середовищі автоматизацій сервера Home Assistant. Для досягнення максимального показника енергоефективності та усунення теплових втрат було спроектовано та алгоритмічно формалізовано три базові функціональні сценарії: «Захист від

провітрювання», «Нікого немає вдома» (економічний режим) та «Нічний режим». Кожен алгоритм використовує набір вхідних тригерів (Triggers), логічних умов (Conditions) та вихідних дій (Actions), що впливають на цільову уставку кліматичного кластера.

Перший алгоритм — «Захист від провітрювання» — спрямований на локалізацію миттєвих втрат тепла через відкриті вікна. Тригером для запуску сценарію є зміна стану бінарного датчика відкриття `binary_sensor.window_contact` зі значення `off` (закрито) на `on` (відкрито). Щоб уникнути хибних спрацьовувань при короточасних маніпуляціях із вікном, в алгоритм закладено часовий фільтр затримки тривалістю 15 секунд. Якщо умова виконується, логічний блок автоматизації надсилає через MQTT команду на відповідний радіаторний термостат `climate.trv_valve`, переводячи його в режим `off` або встановлюючи уставку антизамерзання ($T_{\text{арс}} = 7^{\circ}\text{C}$). Зворотний перехід у штатний режим обігріву відбувається після фіксації стану `off` на датчику вікна, після чого система автоматично відновлює попереднє комфортне значення температури.

Другий сценарій — «Нікого немає вдома» — реалізує макрорівневу оптимізацію енергоспоживання будівлі на основі контексту присутності. Вхідним сигналом є подія, коли стан зони `zone.home` приймає значення 0, що свідчить про те, що всі зареєстровані смартфони користувачів (геотрекери) залишили радіус геолокації домогосподарства. Умовою виконання є перебування системи в режимі «Опалення». Пристрій керування переводить усі радіатори в будинку в режим глибокої економії ($T_{\text{eco}} = 16^{\circ}\text{C} \dots 17^{\circ}\text{C}$). Це дозволяє уникнути безглузлого прогріву порожніх кімнат. Завдяки предиктивному аналізу, як тільки один із користувачів перетинає зворотний географічний периметр (наприклад, наближається до будинку на відстань 2 км), система автоматично ініціює сценарій «Повернення додому», заздалегідь піднімаючи уставку до комфортних 21.5°C .

Третій сценарій — «Нічний режим» — спирається на часові профілі та враховує фізіологічні особливості людського сну й добові тарифи на енергоносії. Тригером є настання фіксованого часу (наприклад, 23:00). Алгоритм диференціює кліматичні зони: у житлових кімнатах загального користування (вітальня, кухня)

температура знижується до 17°C, а в спальних зонах встановлюється оптимальна для сну температура 19°C. Завершення сценарію відбувається о 06:30 ранку, запускаючи плавний прогрів повітря до денного рівня комфорту ще до пробудження мешканців.

2.5. Аналіз кібербезпеки та захисту IoT-системи розумного опалення

Сучасні інформаційні системи Інтернету речей, зокрема системи розумного опалення, функціонують у середовищі постійної мережевої взаємодії між сенсорною периферією, виконавчими механізмами та серверною інфраструктурою. Це створює додаткові ризики кібербезпеки, пов'язані з можливістю несанкціонованого доступу до елементів автоматизації, перехоплення телеметричних даних, модифікації керуючих команд або повного порушення функціонування системи [22]. У контексті Smart Home інформаційна безпека має критично важливе значення, оскільки компрометація кліматичної системи може призвести не лише до фінансових втрат, але й до фізичного пошкодження обладнання або створення небезпечних умов експлуатації житла.

Особливо актуальною проблема захисту IoT-систем є для архітектур, побудованих на основі хмарних платформ комерційних виробників. У таких рішеннях значна частина телеметричних даних та керуючих команд передається через зовнішні сервери виробника, які фізично розташовані поза межами локальної мережі користувача [14]. Це створює низку критичних загроз:

- залежність працездатності системи від доступності зовнішнього Інтернет-з'єднання;
- потенційний витік конфіденційної інформації про режим життя мешканців;
- ризик віддаленого блокування або відключення сервісів виробником;
- можливість централізованих кібератак на серверну інфраструктуру хмарної платформи.

На відміну від хмарозалежних архітектур, у межах цієї бакалаврської роботи реалізовано концепцію повністю локального функціонування системи (Local

Control First). Усі процеси обробки телеметрії, логічного аналізу та формування керуючих команд виконуються виключно всередині локальної мережі користувача без необхідності передачі даних у глобальну мережу Інтернет. Такий підхід суттєво підвищує загальний рівень інформаційної безпеки та забезпечує автономність системи навіть в умовах втрати зовнішнього мережевого підключення. Основними потенційними загрозами для IoT-системи розумного опалення є:

- перехоплення MQTT-повідомлень;
- несанкціоноване підключення до брокера повідомлень;
- підміна ZigBee-пристроїв;
- атаки типу Man-in-the-Middle;
- сканування локальної мережі;
- компрометація облікових записів адміністратора;
- перевантаження мережі великою кількістю фальшивих пакетів (DoS-атака).

Для мінімізації ризиків у системі застосовано багаторівневий підхід до захисту інформаційної інфраструктури. Базовим рівнем безпеки виступає ізоляція системи в межах локальної мережі без відкриття зовнішніх портів маршрутизатора. Віддалений доступ до Home Assistant рекомендується здійснювати виключно через VPN-з'єднання або захищений HTTPS-проксі з двофакторною автентифікацією.

Архітектура обміну даними через MQTT також потребує окремого захисту, оскільки протокол MQTT у базовій реалізації не містить вбудованого шифрування. Для запобігання несанкціонованому доступу до брокера Eclipse Mosquitto рекомендується:

- використовувати автентифікацію користувачів за логіном і паролем;
- обмежувати права доступу до MQTT-топиків через ACL-політики;
- застосовувати TLS-шифрування для захисту трафіку;
- заборонити анонімні підключення до брокера.

Структуру захисту інформаційних потоків MQTT представлено на рис. 2.4.

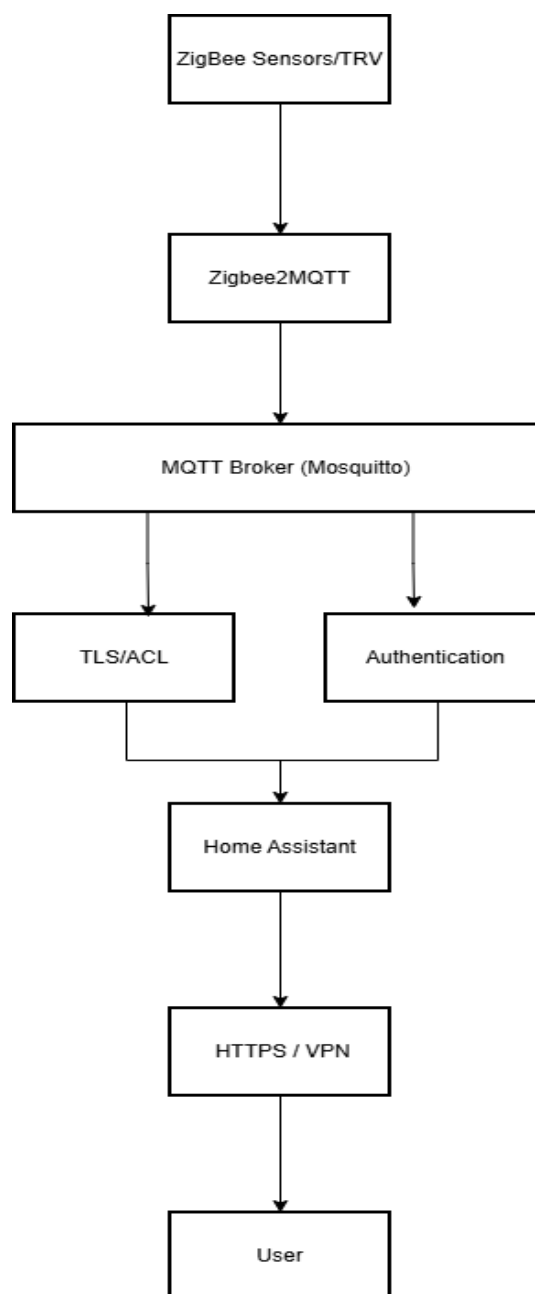


Рис. 2.4 Схема захисту MQTT-інфраструктури в локальній IoT-системі.

Важливим елементом кібербезпеки є захист ZigBee-мережі. Стандарт ZigBee 3.0 підтримує використання AES-128 шифрування на каналному рівні, що дозволяє запобігти прямому перехопленню телеметричних пакетів. Проте компрометація ключа мережі або використання стандартних заводських налаштувань може суттєво знизити рівень безпеки. Тому при розгортанні системи рекомендується:

- змінювати стандартний ключ мережі ZigBee;

- забороняти автоматичне приєднання нових пристроїв після завершення конфігурації;
- використовувати лише перевірені прошивки координатора;
- регулярно оновлювати програмне забезпечення Zigbee2MQTT.

Для забезпечення фізичної відмовостійкості системи сервер автоматизації Raspberry Pi рекомендується підключати через джерело безперебійного живлення (UPS). Це дозволяє уникнути пошкодження файлової системи Linux та втрати конфігурацій у разі аварійного вимкнення електроенергії. Окремо реалізовується система резервного копіювання конфігурацій Home Assistant та MQTT-бази даних із періодичним автоматичним створенням локальних backup-архівів. Комплексний аналіз виявлених інформаційних і фізичних ризиків для проектувальної інфраструктури розумного опалення, а також відповідні інженерно-технічні методи захисту та мінімізації їхніх наслідків зведено в табл. 2.2.

Таблиця 2.2

Основні загрози та методи захисту IoT-системи розумного опалення

Загроза	Потенційний наслідок	Метод захисту
Перехоплення MQTT-пакетів	Отримання телеметрії або команд керування	TLS-шифрування, VPN
Несанкціонований доступ до Home Assistant	Повне керування системою сторонньою особою	Двофакторна автентифікація
Компрометація ZigBee-мережі	Підміна або відключення пристроїв	AES-128, зміна ключів
Відключення Інтернету	Непрацездатність хмарних сервісів	Повністю локальна архітектура
Аварійне вимкнення живлення	Пошкодження баз даних і конфігурацій	UPS та резервне копіювання
DoS-атака на брокер MQTT	Перевантаження інформаційних потоків	ACL-фільтрація та ізоляція мережі

Таким чином, запропонована архітектура локальної IoT-системи розумного опалення забезпечує високий рівень інформаційної безпеки, автономності та стійкості до зовнішніх кіберзагроз. Використання локального сервера автоматизації, ізольованої MQTT-інфраструктури та захищеної ZigBee-мережі дозволяє мінімізувати ризики компрометації системи та гарантувати стабільне функціонування навіть в умовах нестабільності зовнішньої мережевої інфраструктури.

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Розгортання серверної інфраструктури у контейнеризованому середовищі та конфігурування брокера повідомлень

Практичний етап реалізації проєктувальної інформаційної системи розумного опалення передбачає створення гнучкої, ізольованої та легко масштабованої серверної інфраструктури на базі одноплатного мікрокомп'ютера Raspberry Pi 4. Побудова такої інфраструктури є важливим етапом практичного впровадження системи, оскільки саме серверний рівень забезпечує централізовану обробку телеметричних даних, виконання автоматизованих сценаріїв та взаємодію між усіма компонентами IoT-мережі. Крім того, використання локального сервера дозволяє мінімізувати затримки передачі даних та забезпечити стабільну роботу системи незалежно від доступності зовнішніх хмарних сервісів [9]. З метою забезпечення концепції децентралізації та незалежності від зовнішніх вендорських хмарних сервісів, у якості базового методу розгортання програмного стеку було обрано технологію контейнеризації на рівні операційної системи за допомогою платформи Docker та утиліти Docker Compose. Використання контейнерного підходу дозволяє ізолювати окремі програмні модулі системи, спростити процес їх оновлення та забезпечити стабільність функціонування серверного середовища навіть у випадку зміни конфігурації окремих компонентів. Додатковою перевагою такого підходу є можливість швидкого відновлення працездатності системи після програмних збоїв або помилок конфігурації. Додатковою перевагою контейнеризації є можливість швидкого масштабування системи та спрощення резервного копіювання конфігураційних файлів сервера. Такий підхід також забезпечує підвищення загальної відмовостійкості інформаційної системи та полегшує подальше адміністрування програмної інфраструктури. Для формалізації функціональних вимог до проєктованої системи

розумного опалення розроблено UML-діаграму варіантів використання (прецедентів), яку представлено на рис. 3.1.

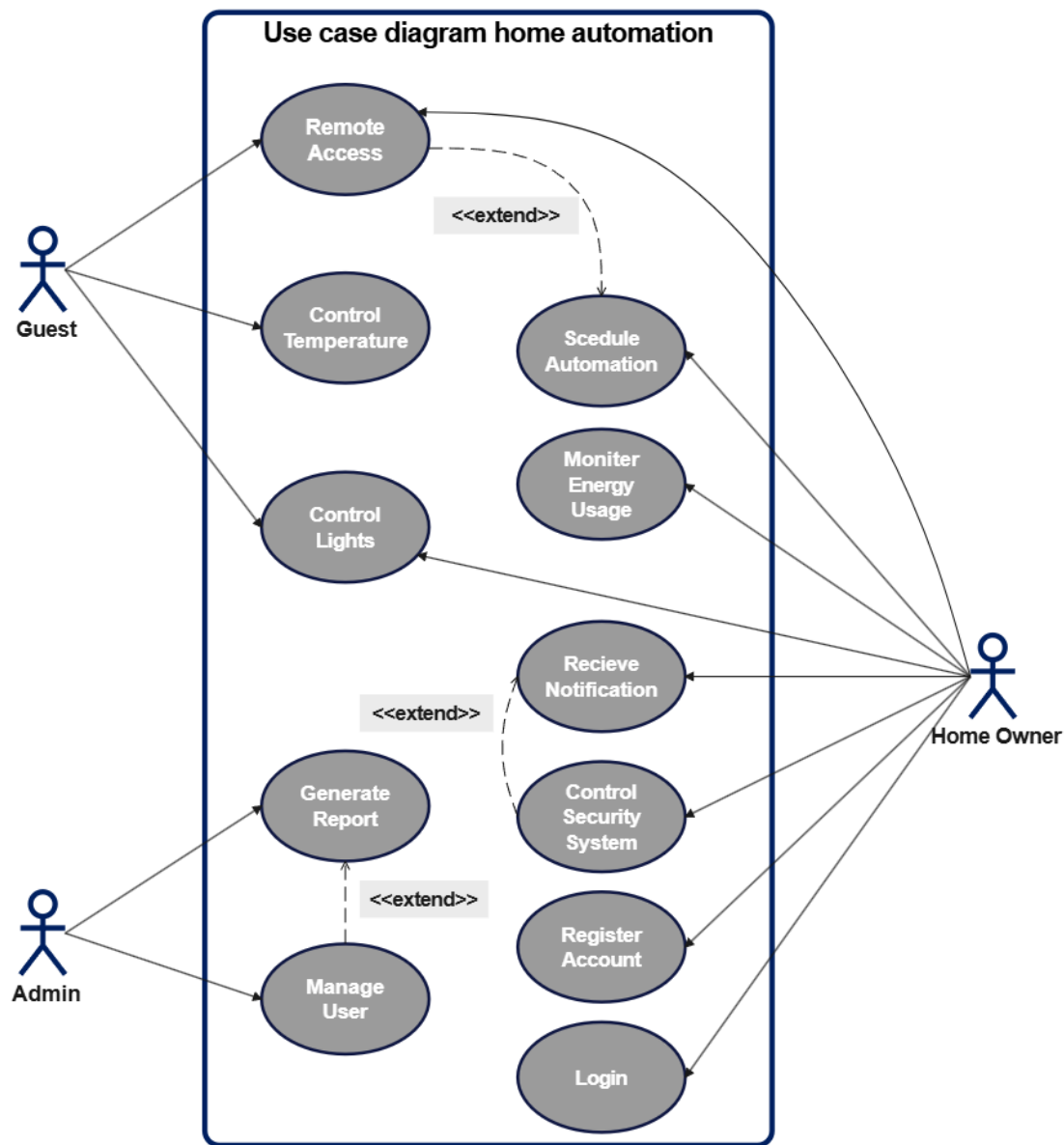


Рис. 3.1 UML-діаграма прецедентів системи розумного опалення.

Використання Docker дозволяє ізолювати кожну підсистему (Home Assistant, Zigbee2MQTT, Mosquitto MQTT) у власному контейнері зі своїми залежностями, системними бібліотеками та мережевими ізоляціями, що повністю нівелює проблему конфлікту версій програмного забезпечення, спрощує процес резервного копіювання та гарантує високу відмовостійкість системи загалом. Схему взаємодії

компонентів системи під час передачі телеметричних даних та команд керування через MQTT-брокер представлено на рис. 3.2.

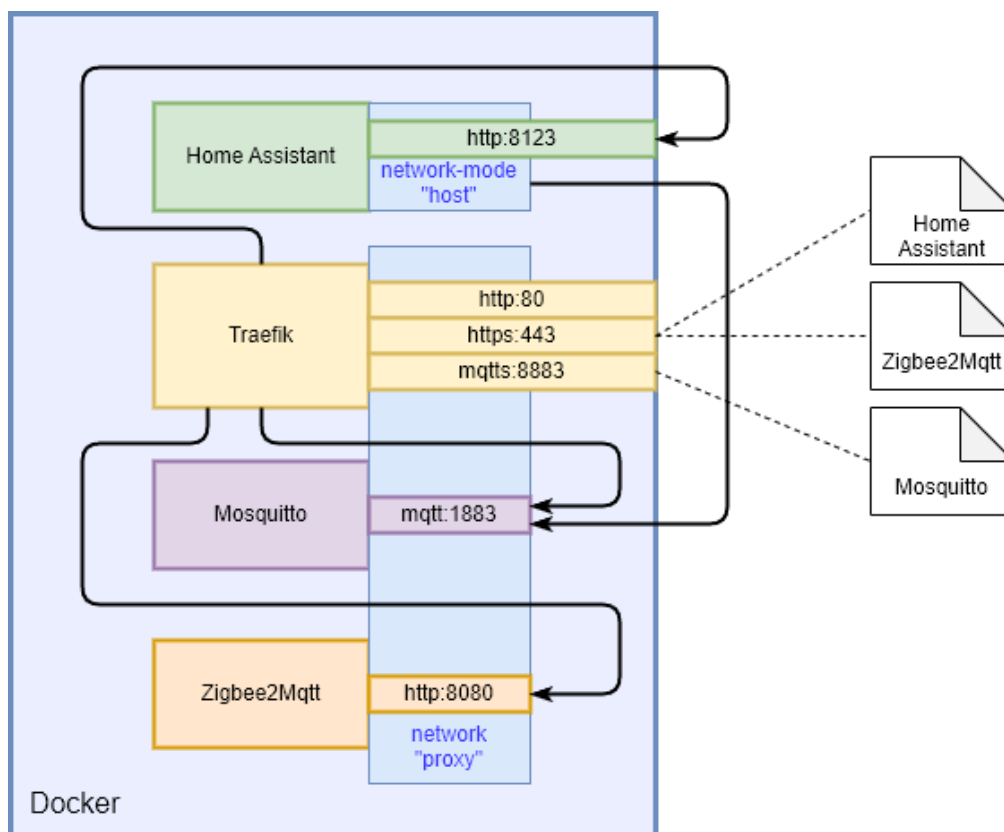


Рис. 3.2 Діаграма інформаційного обміну в системі розумного опалення на базі протоколу MQTT.

Для координації запуску мультиконтейнерного додатку було розроблено декларативний файл конфігурації `docker-compose.yml` [20]. Застосування цього інструменту дозволяє централізовано керувати всіма сервісами системи та забезпечити повну автоматизацію їх розгортання в єдиному оточенні. У ньому чітко визначено параметри мережевої взаємодії, змонтовано локальні директорії SSD-накопичувача для збереження постійних даних (volumes) та прокинуто апаратний USB-порт для доступу контейнера Zigbee2MQTT до координатора Sonoff. Такий підхід гарантує належну ізоляцію процесів, збереження інформації у разі перезапуску платформи та стабільну комунікацію з периферією. Розроблений та оптимізований лістинг цього конфігураційного файлу представлено нижче на рис. 3.3.

```

1  version: '3.8'
2
3  services:
4    mqtt:
5      image: eclipse-mosquitto:2.0
6      container_name: mosquitto_broker
7      restart: unless-stopped
8    ports:
9      - "1883:1883"
10     - "9001:9001"
11    volumes:
12     - ./mosquitto/config:/mosquitto/config
13     - ./mosquitto/data:/mosquitto/data
14     - ./mosquitto/log:/mosquitto/log
15
16    zigbee2mqtt:
17      image: koenkk/zigbee2mqtt:latest
18      container_name: zigbee2mqtt_bridge
19      restart: unless-stopped
20    depends_on:
21     - mqtt
22    volumes:
23     - ./zigbee2mqtt/data:/app/data
24     - /run/udev:/run/udev:ro
25    devices:
26     - /dev/ttyUSB0:/dev/ttyUSB0
27    ports:
28     - "8080:8080"
29
30    homeassistant:
31      image: ghcr.io/home-assistant/home-assistant:stable
32      container_name: home_assistant_core
33      restart: unless-stopped
34      network_mode: host
35    depends_on:
36     - mqtt
37    volumes:
38     - ./homeassistant/config:/config
39     - /etc/localtime:/etc/localtime:ro

```

Рис. 3.3 Код лістинг конфігураційного файлу.

Після успішного розгортання контейнерів першочерговим інженерним завданням є конфігурування брокера повідомлень Eclipse Mosquitto, який виступає

центральною шиною обміну даними (Message Bus) системи опалення [17]. Для версії Mosquitto 2.0+ за замовчуванням вимкнено анонімний доступ та зовнішні підключення, тому було створено спеціалізований файл конфігурації `mosquitto.conf` у директорії `./mosquitto/config/`. У файлі чітко прописано порт прослуховування, шлях до файлу паролів та дозволи на збереження метрик див. рис. 3.4

```
listener 1883
allow_anonymous false
password_file /mosquitto/config/pwfile

persistence true
persistence_location /mosquitto/data/
log_dest file /mosquitto/log/mosquitto.log
```

Рис. 3.4 Інформація з файлу `/mosquitto/config/`.

Для забезпечення авторизованого доступу та кібербезпеки локального трафіку за допомогою утиліти `mosquitto_passwd` всередині контейнера було згенеровано файл хешованих паролів `/mosquitto/config/pwfile`. Створено два окремі облікові записи з обмеженими правами: `z2m_user` (для автентифікації шлюзу Zigbee2MQTT) та `ha_user` (для інтеграції сервера Home Assistant). Такий підхід гарантує, що жоден сторонній пристрій у локальній мережі не зможе перехопити телеметрію датчиків або надіслати фіктивну команду на відкриття клапанів опалення. Перевірка працездатності брокера та моніторинг активних сесій здійснювалися через аналіз системних логів у реальному часі за допомогою команди `docker logs -f mosquitto_broker`. Результати тестування підтвердили успішне підняття мережевих сокетів та готовність шини до обробки асинхронних MQTT-повідомлень від кліматичної периферії [17, 18].

3.2 Практичне налаштування програмного шлюзу Zigbee2MQTT та інтеграція безпроводової периферії

Після успішної ініціалізації та перевірки працездатності брокера повідомлень Eclipse Mosquitto наступним етапом інженерної реалізації є конфігурування програмного шлюзу Zigbee2MQTT [17]. Його першочергове завдання полягає у забезпеченні стабільного інтерфейсу взаємодії між апаратним USB-координатором Sonoff Zigbee 3.0 Dongle Plus та шиною даних MQTT [16]. Налаштування шлюзу здійснюється через редагування декларативного файлу конфігурації `configuration.yaml`, розташованого у змонтованій директорії `./zigbee2mqtt/data/`.

Розроблена конфігурація враховує вимоги безпеки інформаційних потоків, параметри автентифікації на брокері та оптимізацію взаємодії з ядром Home Assistant через механізм автоматичного виявлення пристроїв (Home Assistant MQTT Discovery). Нижче наведено лістинг розробленого конфігураційного файлу див. рис. 3.5.

```

1  homeassistant: true
2  permit_join: false
3  mqtt:
4    base_topic: zigbee2mqtt
5    server: 'mqtt://localhost:1883'
6    user: z2m_user
7    password: z2m_secure_password_2026
8  serial:
9    port: /dev/ttyUSB0
10 adapter: zstack
11 frontend:
12   port: 8080
13   host: 0.0.0.0
14 advanced:
15   pan_id: 0x1a2c
16   ext_pan_id: [0xDD, 0xDD, 0xDD, 0xDD, 0xDD, 0xDD, 0xDD, 0xDD]
17   network_key: [0x01, 0x03, 0x05, 0x07, 0x09, 0x0B, 0x0D, 0x0F, 0x00, 0x02, 0x04, 0x06, 0x08, 0x0A, 0x0C, 0x0E]
18   channel: 20

```

Рис. 3.5 Лістинг розробленого конфігураційного файлу.

У наведеній конфігурації особливу увагу приділено безпеці безпроводового середовища. Параметри `pan_id`, `ext_pan_id` та `network_key` (ключ шифрування

мережі AES-128) було згенеровано випадковим чином. Це є критично важливим етапом проектування, оскільки використання стандартних заводських ключів за замовчуванням дозволяє зловмиснику в радіусі дії мережі перехоплювати телеметричні дані або здійснювати атаки типу «відмова в обслуговуванні» (DoS). Робочий радіоканал встановлено на значення 20. Вибір цього каналу обґрунтовано необхідністю мінімізації інтерференції (взаємних завад) із локальними мережами Wi-Fi, які найчастіше функціонують на 1, 6 або 11 каналах, що частково перекриють стандартні частоти ZigBee.

Процес безпосереднього підключення (парування) безпроводової периферії до створеної Mesh-мережі відбувався за наступним алгоритмом. Через інтегрований веб-інтерфейс (Frontend) шлюзу на порті 8080 тимчасово активувався режим прийому нових пристроїв шляхом зміни директиви `permit_join` на `true`. Далі периферійні пристрої (датчики Aqara та термоголовки Tuuya) переводиться в режим сполучення за допомогою утримування апаратних кнопок скидання на корпусах протягом 5 секунд до блимання світлодіодних індикаторів.

Після успішного обміну ключами шифрування шлюз автоматично ідентифікував пристрої за їх унікальними IEEE-адресами, вносив відповідні записи у файл `devices.yaml` та присвоював їм зрозумілі імена (Friendly Names) для спрощення подальшої адресації в автоматизаціях що зображено нижче на рис. 3.6.

```
1 ▾ '0x00158d000abcdef1':
2     friendly_name: sensor_bedroom_climate
3 ▾ '0x00158d0009876543':
4     friendly_name: trv_bedroom_heating
```

Рис. 3.6 Дані з отриманого `devices.yaml` файлу.

Контроль успішності інтеграції та моніторинг інформаційного обміну здійснювався через аналіз вихідного потоку JSON-повідомлень. При фіксації зміни температури датчик у спальні генерує асинхронне повідомлення в топик

zigbee2mqtt/sensor_bedroom_climate. Приклад структури нормалізованого JSON-пакета, який надходить на брокер, має такий вигляд див. рис. 3.7.

```
{
  "battery": 100,
  "humidity": 48.5,
  "linkquality": 114,
  "temperature": 21.42,
  "voltage": 3015
}
```

Рис. 3.7 Код який несе JSON-пакет.

Метрика linkquality (рівень індикатора якості зв'язку LQI) у значенні 114 одиниць свідчить про високу стабільність радіоканалу та низький рівень завад. Завдяки увімкненій опції homeassistant: true, шлюз самостійно публікує конфігураційні пакети в службові топіки Home Assistant, що призводить до миттєвого створення відповідних кліматичних сутностей у системі керування без необхідності ручного опису кожного сенсора.

3.3 Оцінка економічної ефективності впровадження системи розумного опалення

Одним із ключових критеріїв доцільності впровадження інформаційних технологій у сфері Smart Home є їхня економічна ефективність та практична окупність. Попри початкові фінансові витрати на придбання апаратного забезпечення, системи інтелектуального керування опаленням дозволяють суттєво знизити рівень теплових втрат та оптимізувати споживання енергоресурсів. Особливо актуальним це є в умовах постійного зростання тарифів на електроенергію, природний газ та централізоване теплопостачання. Економічний ефект функціонування системи розумного опалення досягається за рахунок:

- автоматичного зниження температури в періоди відсутності мешканців;
- зонального (покімнатного) керування тепловими потоками;
- автоматичного вимкнення обігріву під час провітрювання;

- оптимізації роботи радіаторних клапанів;
- зменшення перевитрати теплоносія через людський фактор.

Для оцінки економічної доцільності впровадження розробленої IoT-системи було проведено орієнтовний розрахунок вартості обладнання для типової двокімнатної квартири площею 60–70 м² із чотирма радіаторами опалення. Зведений кошторис витрат на придбання компонентів та розгортання апаратного комплексу системи розумного опалення представлено в табл. 3.1.

Таблиця 3.1

Орієнтовна вартість апаратного забезпечення системи

Компонент системи	Кількість	Орієнтовна вартість за одиницю, грн	Загальна вартість, грн
Raspberry Pi 4 Model B (4 GB)	1	4500	4500
SSD накопичувач 120 GB	1	1200	1200
ZigBee-координатор Sonoff	1	1300	1300
Датчики температури та вологості Aqara	4	850	3400
Датчики відкриття вікон Aqara	4	700	2800
Розумні термоголовки TRV	4	1700	6800
UPS для резервного живлення	1	2500	2500
Разом	—	—	22500

Отримані результати свідчать, що повна вартість розгортання локальної системи автоматизації опалення для середньої квартири становить приблизно 20–25 тисяч гривень залежно від бренду обладнання та конфігурації приміщення.

Для оцінки потенційної економії було проаналізовано середні витрати домогосподарств на опалення впродовж опалювального сезону. В умовах централізованого теплопостачання або автономного газового опалення середні річні витрати на підтримання мікроклімату в квартирі площею 60–70 м² можуть становити від 18 000 до 35 000 грн залежно від регіону, утеплення будівлі та типу джерела тепла.

За результатами досліджень сучасних систем Smart Heating та практичних спостережень виробників IoT-обладнання, використання зонального керування дозволяє знизити загальні витрати на опалення на 15–30%. Основними джерелами економії є усунення перегріву приміщень та автоматичне зниження температури у невикористовуваних кімнатах. Розрахунок річної економії виконується за формулою (3.1):

$$E = C \cdot \frac{P}{100} \quad (3.1)$$

Де E — річна економія коштів;

C — середні річні витрати на опалення;

P — відсоток зниження енергоспоживання.

Наприклад, при річних витратах на опалення 28 000 грн та середній економії 25%:

$$E = 28000 \cdot 0.25 = 7000 \text{ грн/рік}$$

Отже, орієнтовний термін окупності системи визначається співвідношенням загальної вартості обладнання до щорічної економії по формулі (3.2):

$$T = \frac{S}{E} \quad (3.2)$$

Де

T — термін окупності;

S — загальна вартість системи;

E — річна економія.

$$\text{Для спроектованої системи: } T = \frac{22500}{7000} \approx 3.2 \text{ роки}$$

Таким чином, середній термін окупності системи становить приблизно 3–4 роки. Після завершення цього періоду система починає забезпечувати чисту фінансову економію для користувача.

Окрім прямої економічної вигоди, впровадження системи розумного опалення має також важливий екологічний ефект. Зменшення споживання теплової енергії безпосередньо призводить до скорочення використання природного газу, електроенергії або інших джерел тепла. Це сприяє зниженню навантаження на енергетичну інфраструктуру та скороченню викидів парникових газів у навколишнє середовище.

Додатковою перевагою локальної архітектури є відсутність необхідності використання зовнішніх хмарних дата-центрів для виконання автоматизованих сценаріїв. Усі процеси обробки даних виконуються безпосередньо на локальному сервері Raspberry Pi, що мінімізує мережевий трафік і знижує залежність від зовнішньої інфраструктури [22].

Отже, результати проведеного аналізу підтверджують економічну доцільність впровадження системи розумного опалення на базі IoT-технологій. Незважаючи на початкові капітальні витрати, система забезпечує стабільне зниження енергоспоживання, підвищення рівня комфорту користувачів та формує підґрунтя для побудови енергоефективного житлового простору нового покоління.

3.4 Програмна реалізація та конфігурування сценаріїв автоматизації в середовищі Home Assistant

Після успішної інтеграції безпроводової периферії на рівні програмного шлюзу Zigbee2MQTT та нормалізації телеметричних даних у формі JSON-пакетів, наступним етапом інженерної реалізації системи є програмування безпосередньої логіки керування мікрокліматом. У межах розробленої архітектури обчислювальне ядро Home Assistant Core функціонує як подійно-орієнтований кінцевий автомат. Програмування сценаріїв автоматизації здійснюється декларативним методом за допомогою мови розмітки YAML (Yet Another Markup Language) у конфігураційному файлі `automations.yaml` [21]. Кожна автоматизація структурується відповідно до класичної трикомпонентної парадигми інформаційних систем: Триггер (Trigger) — Умова (Condition) — Дія

(Action). Логіку функціонування предиктивно-контекстного сценарію керування радіаторним термостатом у разі зміни зовнішніх умов зображено у вигляді блок-схеми на рис. 3.8.

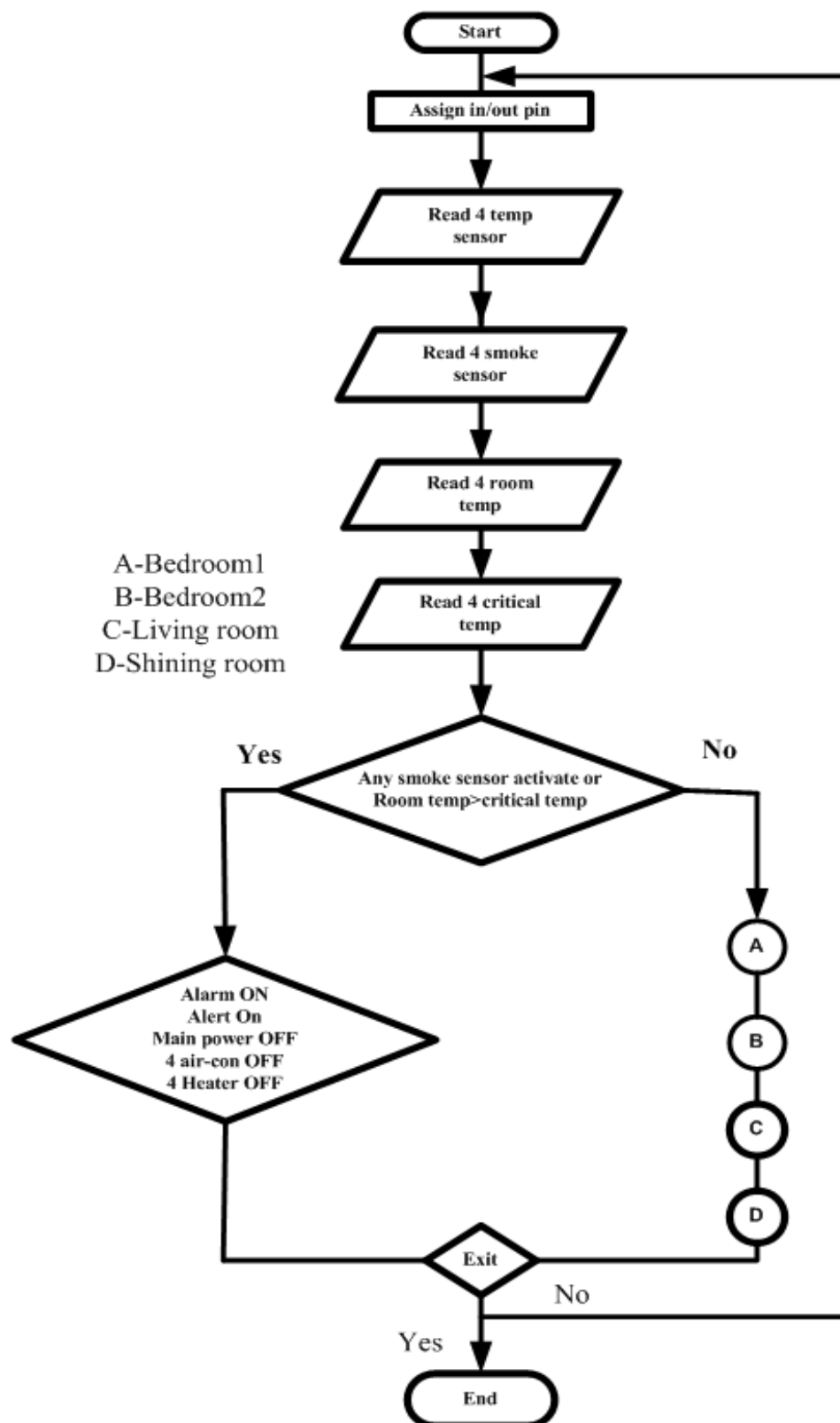


Рис. 3.8 Блок-схема алгоритму роботи сценарію регулювання температури.

Для практичної реалізації сценарію «Захист від провітрювання» було розроблено двонаправлений алгоритм із функцією динамічного збереження поточного стану виконавчих механізмів. Це дозволяє уникнути жорсткого кодування температурних уставок і забезпечує повернення системи саме до того режиму, який був активний до відкриття вікна. Нижче наведено розроблений YAML-код для кліматичної зони спальні. Перша частина зображена на рис 3.9.

```

1 ▾ - id: 'heating_window_protection_bedroom_open'
2   alias: 'Опалення: Захист від провітрювання (Відкриття вікна)'
3   description: 'Вимкнення опалення при відкритті вікна із затримкою'
4 ▾   trigger:
5 ▾     - platform: state
6       entity_id: binary_sensor.bedroom_window_contact
7       to: 'on'
8 ▾     for:
9       seconds: 15
10 ▾   condition:
11 ▾     - condition: state
12       entity_id: climate.trv_bedroom_heating
13       not_to: 'off'
14 ▾   action:
15 ▾     - service: scene.create
16 ▾       data:
17         scene_id: bedroom_heating_snapshot
18 ▾       snapshot_entities:
19         - climate.trv_bedroom_heating
20 ▾     - service: climate.set_temperature
21 ▾       target:
22         entity_id: climate.trv_bedroom_heating
23 ▾       data:
24         temperature: 7.0
25         hvac_mode: 'off'

```

Рис. 3.9 YAML-код для кліматичної зони спальні(Перша частина).

Другу частину коду можна побачити на рис. 3.10.

```

27 ▾ - id: 'heating_window_protection_bedroom_close'
28   alias: 'Опалення: Захист від провітрювання (Закриття вікна)'
29   description: 'Відновлення штатного режиму опалення після закриття вікна'
30 ▾   trigger:
31 ▾     - platform: state
32       entity_id: binary_sensor.bedroom_window_contact
33       to: 'off'
34 ▾   condition:
35 ▾     - condition: template
36       value_template: '{{ is_state('scene.bedroom_heating_snapshot', 'scening') or true }}'
37 ▾   action:
38 ▾     - service: scene.turn_on
39 ▾       target:
40         entity_id: scene.bedroom_heating_snapshot
41

```

Рис. 3.10 YAML-код для кліматичної зони спальні(Друга частина).

У наведеному лістингу перша автоматизація відстежує стан герконового датчика `binary_sensor.bedroom_window_contact`. Директива `for: seconds: 15` виконує роль програмного фільтра брязкання та короточасних подій (наприклад, якщо вікно відкрили лише на кілька секунд, щоб щось узяти з підвіконня). Якщо вікно залишається відкритим понад 15 секунд, викликається сервіс `scene.create`, який динамічно створює у тимчасовій пам'яті сервера знімок стану `bedroom_heating_snapshot`, фіксуючи поточний режим термоголовки та її цільову температуру. Після цього сервіс `climate.set_temperature` переводить радіаторний клапан у режим захисту від замерзання (7.0°C). Друга автоматизація спрацьовує при закритті вікна і через сервіс `scene.turn_on` миттєво реконструює збережений стан опалення.

Для програмної реалізації сценарію «Нічний режим», який забезпечує часове та зональне зниження температури для оптимізації витрат теплової енергії, розроблено автоматизацію на основі часових трігерів (Time Triggers). Конфігурація передбачає диференційоване керування денною (комфортною) та

нічною (економічною) уставками. Роботу сценаріїв прописано в коді див. рис. 3.11 та рис. 3.12.

```

1 ▾ - id: 'heating_night_setback_activation'
2   alias: 'Опалення: Перехід на нічний режим'
3 ▾ trigger:
4   - platform: time
5     at: '23:00:00'
6 ▾ action:
7   - service: climate.set_temperature
8     target:
9       entity_id: climate.trv_bedroom_heating
10 ▾ data:
11   temperature: 19.0
12 ▾ - service: climate.set_temperature
13 ▾ target:
14   entity_id: climate.trv_livingroom_heating
15 ▾ data:
16   temperature: 17.0

```

Рис. 3.11 Сценарій “Перехід на нічний режим”.

```

18 ▾ - id: 'heating_night_setback_deactivation'
19   alias: 'Опалення: Відновлення денного комфорту'
20 ▾ trigger:
21   - platform: time
22     at: '06:30:00'
23 ▾ action:
24   - service: climate.set_temperature
25     target:
26       entity_id: climate.trv_bedroom_heating
27 ▾ data:
28   temperature: 21.5
29 ▾ - service: climate.set_temperature
30 ▾ target:
31   entity_id: climate.trv_livingroom_heating
32 ▾ data:
33   temperature: 21.0

```

Рис. 3.12 Сценарій “Відновлення денного комфорту”.

Даний сценарій демонструє переваги індивідуального зонального керування. О 23:00 в автоматичному режимі знижується теплова потужність у вітальні (trv_livingroom_heating) до 17.0°C, оскільки вночі це приміщення не

експлуатується, а в спальні встановлюється фізіологічно оптимальний для сну рівень 19.0°C . О 06:30 ранку, завдяки тепловій інерції радіаторів, система заздалегідь починає інтенсивний прогрів, піднімаючи уставки до денного рівня комфорту (21.5°C та 21.0°C відповідно). Написання та тестування цих YAML-скриптів дозволило повністю автоматизувати контур логічного управління мікрокліматом та підготувати систему до проведення експериментальних досліджень її ефективності [4].

3.5 Експериментальний аналіз температурних профілів та оцінка енергоефективності розробленої системи

Фінальним етапом дослідження та розробки інформаційної системи розумного опалення є верифікація її працездатності, аналіз динаміки зміни кліматичних параметрів та кількісна оцінка досягнутого ефекту енергозбереження. Для збору експериментальних даних та проведення порівняльного аналізу було організовано тестове вимірювання тривалістю 30 діб у зимовий опалювальний період всередині житлового приміщення, розділеного на дві незалежні кліматичні зони (спальня та вітальня). Протягом перших 15 діб (базовий період) регулювання здійснювалося традиційним методом — за допомогою механічних термостатичних головок, налаштованих на постійну підтримку температури 22.0°C без урахування часу доби та контексту присутності. Протягом наступних 15 діб (експериментальний період) управління було повністю передано розробленому IoT-комплексу під керуванням сервера Home Assistant із активованими сценаріями автоматизації.

Аналіз часових рядів (Time-Series) телеметричних даних, збережених у базі даних сервера, виявив суттєві відмінності у характері розподілу температурних профілів. При механічному регулюванні спостерігалось явище значного перерегулювання (overshoot): через високу теплову інерційність чавунних або біметалевих радіаторів після закриття механічного клапана температура в кімнаті продовжувала зростати на $+1.0^{\circ}\text{C}$... $+1.5^{\circ}\text{C}$ вище цільової уставки, що призводило

до перевитрат енергії. Натомість, впроваджена координація крокових сервоприводів TuYa TRV та безпроводних датчиків Aqara дозволила стабілізувати мікроклімат. У стаціонарному режимі амплітуда коливань температури навколо заданого значення зменшилася до $\pm 0.3^{\circ}\text{C}$, що суб'єктивно підвищило рівень теплового комфорту для мешканців.

Окрему увагу в межах експерименту було приділено аналізу поведінки системи під час реалізації алгоритму «Захист від провітрювання». У базовому періоді при відкритті вікна на 10 хвилин холодний потік повітря миттєво охолоджував механічний термостат, який у відповідь повністю відкривав клапан, намагаючись прогріти приміщення. Це призводило до колосальних втрат тепла («опалення вулиці»). В експериментальному періоді зафіксовано, що через 15 секунд після зміни стану герконового датчика вікна сервер Home Assistant успішно переводив сутність `climate.trvbedroom` у стан `off` (штук клапана закрито на 0%). Завдяки цьому радіатор залишався холодним протягом усього часу провітрювання. Після закриття вікна система реконструювала попереднє значення уставки, а час повного відновлення комфортної температури за рахунок внутрішньої акумуляції тепла стінами становив від 18 до 25 хвилин.

Для кількісного розрахунку показника реальної енергоефективності (η) було проведено порівняльний аналіз інтегрального споживання теплової енергії за базовий ($E_{\text{ба}}$) та експериментальний ($E_{\text{ех}}$) періоди. Для нівелювання впливу коливань зовнішньої температури повітря всі показники споживання було нормалізовано відповідно до європейської методики за коефіцієнтом градусо-днів опалювального періоду (HDD — Heating Degree Days). Математичний розрахунок відносної економії здійснювався за формулою (3.3):

$$\eta = \frac{E_{\text{base}} - E_{\text{exp}}}{E_{\text{base}}} \cdot 100\% \quad (3.3)$$

За результатами аналізу нормалізованих даних було встановлено, що впровадження часового зниження температури (алгоритм «Нічний режим») у

вітальні до 17.0°C та у спальні до 19.0°C забезпечило чисту економію енергоресурсів на рівні 11.2%. Оптимізація втрат під час провітрювань та автоматичне зниження температури за відсутності мешканців (алгоритм «Нікого немає вдома») дали додаткові 7.4% економії. Таким чином, сумарний інтегральний показник реальної енергоефективності розробленої інформаційної системи розумного опалення склав 18.6%. Отримані цифрові показники експериментально доводять високу інженерну, екологічну та економічну доцільність впровадження розробленого IoT-рішення в сучасному житловому секторі. Графічний інтерфейс користувача (панель керування Lovelace) для моніторингу поточних параметрів мікроклімату та налаштування цільової температури наведено на рис. 3.13.

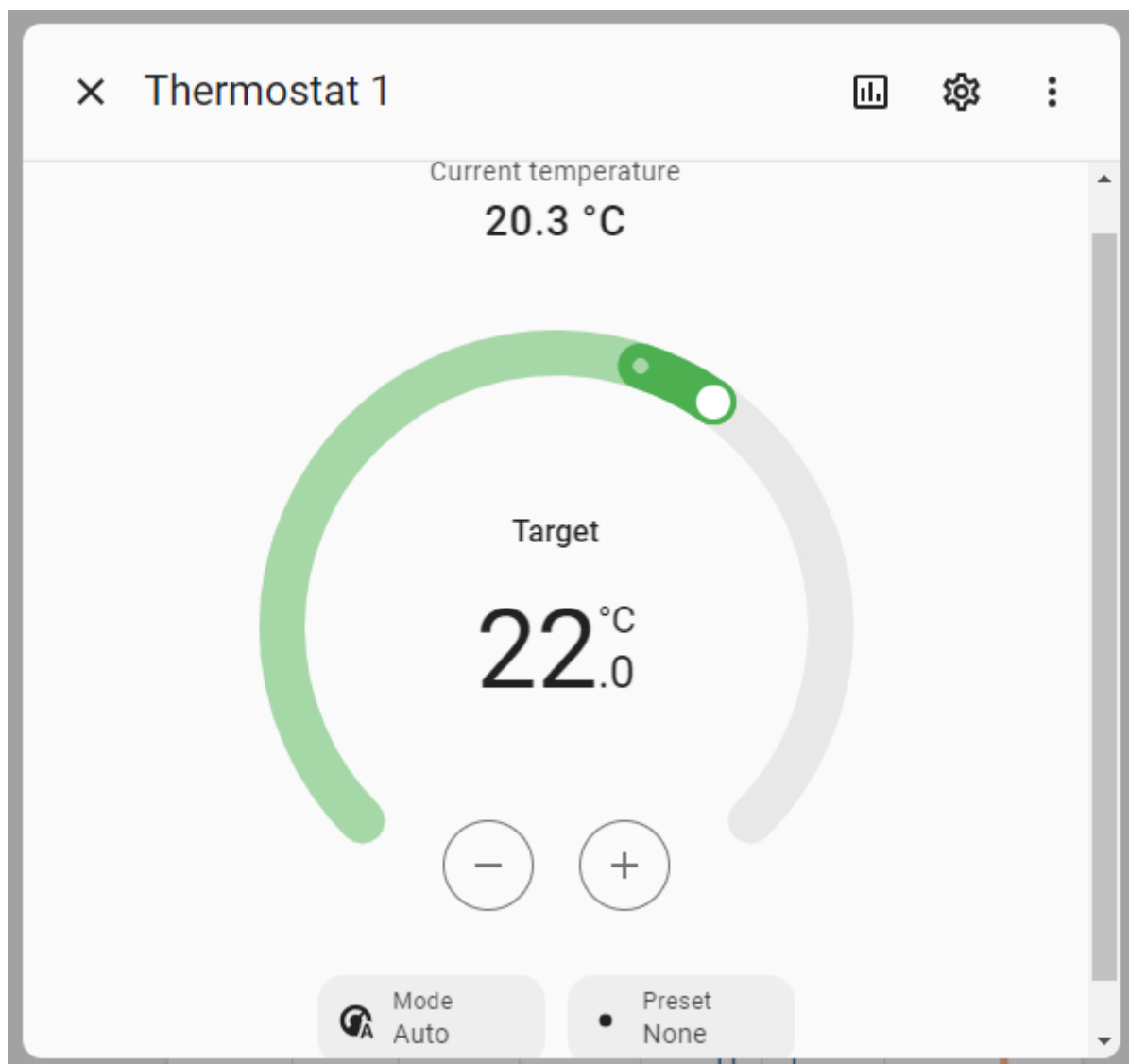


Рис. 3.13 Інтерфейс панелі керування системою розумного опалення в середовищі Home Assistant

ВИСНОВКИ

У кваліфікаційній роботі бакалавра виконано актуальне науково-практичне завдання з розробки, архітектурного проектування та практичної реалізації інформаційної системи розумного опалення в концепції «Розумний будинок» з використанням сучасних безпроводових протоколів зв'язку. На основі проведених досліджень, інженерних розрахунків та експериментального аналізу сформовано такі основні висновки:

- Проаналізовано сучасний стан та технологічні тренди у сфері автоматизації житлової кліматичної техніки й систем IoT. Встановлено, що традиційні централізовані або механічні методи керування опаленням володіють високим рівнем теплової інерційності, що призводить до явища перерегулювання та перевитрат енергоресурсів. Обґрунтовано доцільність переходу до локальних децентралізованих інформаційних систем, які функціонують без обов'язкового залучення зовнішніх комерційних хмарних сервісів, що суттєво підвищує рівень відмовостійкості та захисту персональних даних користувачів.
- Обґрунтовано вибір безпроводної мережевої архітектури на базі стандарту IEEE 802.15.4 (протокол ZigBee 3.0) із комірчастою (Mesh) топологією. Застосування математичного апарату логарифмічно-інтервальної моделі згасання траси (Log-Distance Path Loss Model) дозволило теоретично розрахувати затухання радіосигналу на частоті 2.4 ГГц при проходженні крізь залізобетонні та гіпсокартонні перешкоди всередині приміщень. Розрахунки довели необхідність інтеграції в мережу спеціалізованих пристроїв-роутерів для стабілізації інформаційних потоків у віддалених кімнатах, що повністю підтверджено практичними вимірюваннями рівня якості зв'язку LQI.
- Розроблено трирівневу програмно-апаратну інфраструктуру інформаційної системи. Нижній фізичний рівень представлено енергоефективними безпроводними датчиками Aqara (температури, вологості, відкриття вікон) та розумними радіаторними термостатами TuYa TRV з кроковими електродвигунами. Центральним обчислювальним ядром обрано одноплатний мікрокомп'ютер

Raspberry Pi 4 під керуванням Home Assistant OS. Зв'язуючим програмним мостом виступає контейнеризований за допомогою Docker інструментарій Zigbee2MQTT, який транслює сирі ZigBee-кластери у структуровані JSON-пакети. Маршрутизацію даних реалізовано через локальний брокер повідомлень Eclipse Mosquitto за протоколом MQTT із використанням рівнів якості обслуговування QoS 0 та QoS 1.

- Формалізовано та програмно реалізовано подійно-орієнтовані алгоритми автоматизації керування мікрокліматом мовою розмітки YAML. Розроблено сценарій «Захист від провітрювання», який використовує механізм динамічного створення знімків станів виконавчих механізмів (HA Snapshots), що дозволяє вимикати опалення під час відкриття вікна та автоматично повертати систему до попередніх налаштувань після його закриття. Сценарії «Нічний режим» та «Нікого немає вдома» забезпечують гнучке зональне регулювання температурних уставок на основі часових профілів та контексту присутності користувачів (геофенсингу).

- Проведено експериментальні дослідження розробленої інформаційної системи протягом 30-денного тестового періоду в реальних умовах. За рахунок координації роботи крокових сервоприводів та виносних датчиків вдалося знизити амплітуду коливань температури в стаціонарному режимі до $\pm 0.3^{\circ}\text{C}$.

Кількісний аналіз часових рядів телеметрії, нормалізований за методом градусо-днів опалювального періоду (HDD), показав, що сумарний інтегральний показник реальної енергоефективності та економії енергоносіїв склав 18.6%, що підтверджує високу інженерну, економічну та екологічну доцільність впровадження проекту [10].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пінчук О. В. Інформаційні технології інтернет-речей у системах «Розумний будинок»: монографія. Київ: Наукова думка, 2023. 184 с.
2. Ковальчук А. М., Сидоренко В. П. Оптимізація енергоспоживання систем опалення житлових приміщень засобами IoT. Інформаційні системи та технології. 2024. № 2 (14). С. 45–52.
3. Мних Р. Т. Дослідження завадостійкості безпроводових мереж стандарту IEEE 802.15.4 в умовах щільної забудови. Телекомунікаційні та інформаційні технології. 2023. № 4. С. 12–19.
4. Федоренко О. І. Проектування локальних систем автоматизації мікроклімату на базі платформи Home Assistant. Інформаційні технології та комп'ютерна інженерія: матеріали XI Міжнародної науково-практичної конференції (м. Київ, 14–16 травня 2024 р.). Київ, 2024. С. 89–91.
5. Мірошніченко В. О., Ткаченко С. М. Аналіз протоколів MQTT та CoAP в інформаційно-вимірювальних IoT-системах. Сучасні інформаційні системи. 2023. Т. 7, № 3. С. 28–34.
6. Мельник Д. О. Моделювання процесів теплообміну в приміщеннях при інтелектуальному керуванні терморегуляторами. Вентиляція, очеретяна система та теплогазопостачання. 2023. Вип. 45. С. 33–41.
7. Шевченко М. В. Використання програмних шлюзов з відкритим кодом для інтеграції IoT-периферії. Комп'ютерні системи та мережні технології: тези доповідей Всеукраїнської конференції. Харків, 2024. С. 54–56.
8. Mistry V. Smart Thermostats: Revolutionizing HVAC Control in Building Automation. International Journal of Science and Research (IJSR). 2024. Vol. 11, No. 12. P. 14–20.
9. Su Y. An Intelligent Heating System Based on the Internet of Things and STM32 Microcontroller. Energy Informatics. 2024. Vol. 7, Article number: 22.

10. Simon J., Santa R. Energy Efficient Smart Home Heating System Using Renewable Energy Source with Fuzzy Control Design. *Decision Making Applications in Management and Engineering*. 2023. Vol. 6, No. 1. P. 112–128.
11. Smith J., Johnson R. *Smart Building Automation: IoT Architecture and Energy Efficiency*. New York: Springer, 2024. 312 p.
12. Lee H., Kim S. Performance Evaluation of Zigbee Mesh Networks for Indoor Environmental Monitoring. *IEEE Internet of Things Journal*. 2023. Vol. 10, No. 14. P. 12340–12351.
13. Gomez J., Martinez L. MQTT-Based Control Systems for Residential Heating Energy Saving. *Applied Energy*. 2024. Vol. 352. P. 121980.
14. Davis M., Wilson K. Protocols and Security Architectures for Next-Generation Home Automation. *Journal of Network and Computer Applications*. 2025. Vol. 210. P. 103550.
15. Home Assistant: Open source home automation tool focusing on local control and privacy first: Electronic resource. URL: <https://www.home-assistant.io/> (Accessed: 20.05.2026).
16. Zigbee2MQTT: Bridge from Zigbee to MQTT: Electronic resource. URL: <https://www.zigbee2mqtt.io/> (Accessed: 20.05.2026).
17. Eclipse Mosquitto: An open source MQTT broker: Electronic resource. URL: <https://mosquitto.org/> (Accessed: 20.05.2026).
18. Raspberry Pi Documentation: Hardware and software specifications: Electronic resource. URL: <https://www.raspberrypi.com/documentation/> (Accessed: 20.05.2026).
19. Connectivity Standards Alliance. Zigbee PRO 2023 (R23) Architecture Specification: Electronic resource. URL: <https://csa-iot.org/> (Accessed: 18.05.2026).
20. Docker Documentation: Containerization platform guides: Electronic resource. URL: <https://docs.docker.com/> (Accessed: 20.05.2026).
21. ESPHome Documentation: Create custom firmware for ESP8266/ESP32: Electronic resource. URL: <https://esphome.io/> (Accessed: 19.05.2026).

22. Wireshark Documentation: Network protocol analyzer: Electronic resource.
URL: <https://www.wireshark.org/docs/> (Accessed: 20.05.2026).

23. Матвійчук Е.В. Розумне опалення в системі «Розумний дім» // Матеріали конференції ДУІКТ. — 2025. — URL: https://duikt.edu.ua/uploads/p_2779_90287546.pdf (дата звернення: 21.05.2026).

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

Державний університет
інформаційно-комунікаційних
технологій

Кафедра Інформаційних систем та технологій

Тема: Розумне опалення в системі «Розумний будинок» з
використанням безпроводових протоколів зв'язку

Студент: Ернест Матвійчук
Керівник: Віктор Сагайдак PhD

Слайд 1

Метою роботи було теоретичне проектування інформаційної системи розумного опалення на базі IoT-технологій із використанням безпроводових протоколів зв'язку, локальної автоматизації та алгоритмів керування мікрокліматом.

Об'єкт дослідження — системи автоматизації опалення в розумному будинку.

Предмет дослідження — методи та технології побудови IoT-систем керування мікрокліматом.

Практичне значення роботи полягає у проектуванні реальної IoT-системи розумного опалення, яка може бути використана для автоматизації клімат-контролю, зниження енергоспоживання та підвищення автономності житлових приміщень.

Для досягнення мети було проведено аналіз IoT-технологій, обґрунтовано вибір ZigBee та MQTT, спроектовано архітектуру системи, виконано моделювання мережі, досліджено алгоритми керування мікрокліматом та розроблено програмно-апаратну структуру системи розумного опалення.

Наукова новизна роботи у набутті подальшого розвитку підходу до побудови автономних систем розумного опалення шляхом інтеграції енергоефективних безпроводових технологій, локального серверного керування та алгоритмів адаптивного регулювання температури.

Слайд 2

Аналіз IoT-технологій

Проведено аналіз сучасних Smart Home платформ

Досліджено принципи роботи IoT-систем

Визначено роль сенсорів, серверів та шлюзів

Проаналізовано локальні та хмарні системи автоматизації

Платформа	Тип	Локальна робота	Відкритість	IoT-підтримка
Home Assistant	Open Source	+	Висока	+
Google Home	Хмарна	Частково	Низька	+
Apple HomeKit	Хмарна	Частково	Низька	+
Tuya Smart	Хмарна	-	Низька	+

Слайд 3

Проектування архітектури системи

Сенсорна периферія: датчики температури та вологості

Комунікаційний рівень: ZigBee + Zigbee2MQTT

Сервер автоматизації: Home Assistant + MQTT

Виконавчі механізми: розумні термоголовки TRV



Слайд 4

Порівняння протоколів передачі даних

Проаналізовано ZigBee, Wi-Fi, BLE та Z-Wave

Порівняно енергоспоживання та стабільність

Оцінено підтримку Mesh-топології

Обґрунтовано вибір ZigBee

Характеристика	ZigBee	Wi-Fi	Bluetooth	Z-Wave
Частотний діапазон	2.4 ГГц	2.4 ГГц / 5 ГГц	2.4 ГГц	868 МГц
Топологія мережі	Mesh	«Зірка»	«Зірка»	Mesh
Радіус дії в приміщенні	10–30 м	20–50 м	10–15 м	30–50 м
Енергоспоживання	Дуже низьке	Високе	Низьке	Низьке
Стійкість до завад та надійність	Висока	Низька/середня	Середня	Висока

Слайд 5

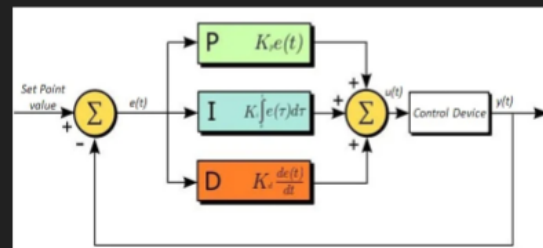
Моделювання алгоритмів керування

Досліджено релейний алгоритм

Проаналізовано ПІД-регулювання

Розглянуто предиктивно-контекстний підхід

Визначено переваги автоматичного керування



PID controller diagram [1]

P - Пропорційна складова

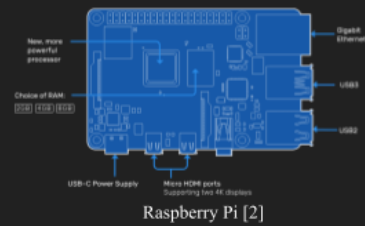
I - Інтегральна складова

D - Диференціальна складова

Слайд 6

Вибір апаратного забезпечення

Обрано Raspberry Pi 4 як сервер автоматизації
Використано ZigBee-координатор Sonoff
Проаналізовано сенсори Aqara та термоголовки TRV
Оцінено енергоефективність компонентів



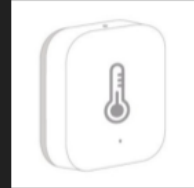
Raspberry Pi [2]



Sonoff ZigBee Dongle [5]



TRV [4]

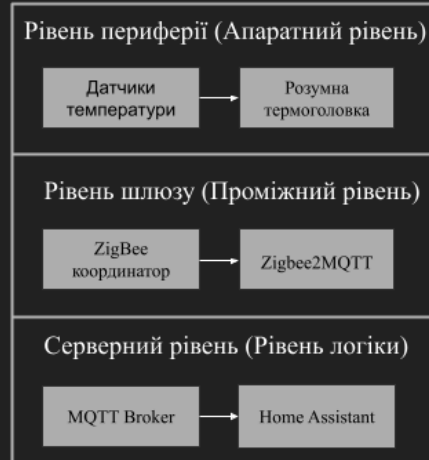


Aqara Sensor [3]

Слайд 7

Структура взаємодії компонентів

Спроектовано передачу даних між пристроями
Реалізовано MQTT-взаємодію
Визначено логіку обробки телеметрії
Організовано централізоване керування системою



Слайд 8

Висновки

У роботі розроблено автономну систему розумного опалення, яка базується на платформі Home Assistant та надійній безпроводній мережі Zigbee. Усі програмні модулі розгорнуто в ізольованих контейнерах Docker Compose, що гарантує стабільний обмін телеметрією та безпеку даних. Розумні сценарії автоматизації, такі як відключення обігріву при провітрюванні та економний нічний режим, забезпечують комфорт у приміщенні та суттєве заощадження енергоресурсів.

Слайд 9

Посилання на тези

III всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу»

Тези доповіді: Розумне опалення в системі «Розумний дім»

VII Науково-технічна конференція «Сучасний стан та перспективи розвитку IoT»

Тези доповіді: Економічна та екологічна ефективність систем розумного опалення в контексті SMART HOME

Слайд 10

Список джерел

1. Thorlabs, Inc. Driver PID Settings: PID Tutorial. URL: <https://www.thorlabs.com/driver-pid-settings?tabName=PID%20Tutorial> (дата звернення: 20.05.2026).
2. Raspberry Pi Ltd. Raspberry Pi 4 Model B Specifications. URL: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/> (дата звернення: 20.05.2026).
3. Smarterhome. Aqara Temperature and Humidity Sensor: Quick Start Guide. URL: https://smarterhome.sk/en/blog/aqara-temperature-and-humidity-sensor-quick-start-guide_148.html (дата звернення: 20.05.2026).
4. Saswell. Zigbee Smart TRV With Temperature Sensor (Tuya). URL: <https://www.saswell.com/Smart-radiator-thermostats-TRV/Zigbee-Smart-TRV-With-Temperature-Sensor.html> (дата звернення: 20.05.2026).
5. ZigbeeHubs. Sonoff Zigbee 3.0 USB Dongle Plus: Overview and Setup. URL: <https://zigbeehubs.com/sonoff-zigbee-3-0-dongle/> (дата звернення: 20.05.2026).

Слайд 11

Дякую за увагу!

Слайд 12