

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Навчально-науковий
інститут електроенергетики
(інститут)
Факультет інформаційних технологій
(факультет)
Кафедра інформаційних технологій та комп'ютерної інженерії
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня бакалавра

здобувача Маніна Богдана Максимовича
(ПІБ)

академічної групи 123-21-2
(шифр)

спеціальності 123 Комп'ютерна інженерія
(код і назва спеціальності)

за освітньо-професійною програмою 123 Комп'ютерна інженерія
(офіційна назва)

на тему “Інтеграція та імплементація елементів IoT житлового приміщення з комп'ютерною системою контролю енергоефективності житлового комплексу”
(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	Доц. Шедловський І.А.			
спеціальної частини	Доц. Шедловський І.А.			
розділу розробка корпоративної мереж	Доц. Шедловський І.А.			
Рецензент				
Нормоконтролер	Проф. Цвіркун Л.І.			

Дніпро
2025

ЗАТВЕРДЖЕНО:
завідувач кафедри
інформаційних технологій
та комп'ютерної інженерії
(повна назва)

_____ Гнатушенко В.В.
(підпис) (прізвище, ініціали)

“25” лютого 2025 року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавр

здобувача Маніна Б.М. академічної групи 123-21-2
(прізвище та ініціали) (шифр)

спеціальності 123 Комп'ютерна інженерія

за освітньо-професійною програмою 123 Комп'ютерна інженерія
(офіційна назва)

на тему “Інтеграція та імплементація елементів IoT житлового приміщення з комп'ютерною системою контролю енергоефективності житлового комплексу”

затверджену наказом ректора НТУ «Дніпровська політехніка» 05.05.2025 р. №336-с

Розділ	Зміст	Термін виконання
Стан питання та постановка завдання	На основі аналізу технічних джерел конкретизується предмет та мета роботи та виконується постановка завдання	14.05.2025
Розробка апаратної частини комп'ютерної системи	На основі вимог до системи виконується вибір сенсорів, контролерів і пристроїв керування, розробити структурну схему IoT-системи.	24.05.2025
Розробка корпоративної мережі	Розв'язується завдання з побудови комп'ютерної мережі житлового комплексу з реалізацією адресації, маршрутизації і засобів безпеки.	29.05.2025
Розробка програмного компоненту системи	Реалізується програмний компонент із розробкою автоматизацій, інтерфейсу та логіки керування.	02.06.2025

Завдання видано _____
(підпис керівника)

доц. Шедловський І.А.
(прізвище, ініціали)

Дата видачі 25.02.2025

Дата подання до екзаменаційної комісії 16.06.2025

Прийнято до виконання _____ Манін Б.М.

РЕФЕРАТ

Пояснювальна записка 102 с., 40 рис., 8 табл., 2 дод., 20 джерел.

ЕНЕРГОЕФЕКТИВНІСТЬ, ІоТ, ІМПЛЕМЕНТАЦІЯ, КОМП'ЮТЕРНА МЕРЕЖА, КІБЕРФІЗИЧНА СИСТЕМА

Об'єкт розробки – ІоТ система контролю енергоефективності окремого приміщення з функцією інтеграції в систему енергоефективності житлового комплексу та системи дистанційного онлайн контролю стану об'єктів приміщення та показників внутрішньої атмосфери та кліматичних параметрів.

Мета роботи – інтеграція ІоТ-елементів у систему контролю енергоефективності житлового комплексу.

Завдання роботи – Аналіз існуючих технологічних рішень, визначення оптимальних способів впровадження ІоТ системи та мережевих інструментів для забезпечення інтерактивного управління енергоспоживанням та підвищення енергоефективності житлового комплексу.

Здійснено розробку структуру ІоТ системи, обгрунтовано обрано основні функції контролю та управління елементами кіберфізичної системи. Відповідно до реалізованих функцій системи обрано сучасні технічні засоби. Відповідно до завдання розроблено проект комп'ютерної мережі та виконані її налаштування для забезпечення ефективності інформаційного обміну та безпеки.

Реалізовано повноцінний програмний компонент системи керування енергоефективністю житлового приміщення, що базується на платформі Home Assistant. Описані та протестовані типові сценарії автоматизації - керування освітленням за присутністю, температурне регулювання через термоголівки, а також контроль енергоспоживання побутових приладів.

ЗМІСТ

	Перелік умовних позначень, символів, одиниць, скорочень і термінів.....	6
	Вступ.....	7
1	Стан питання і постановка завдання.....	8
	1.1 Галузь застосування IoT системи контролю енергоефективності.....	8
	1.2 Характеристика і структура об'єкта впровадження.....	10
	1.3 Функціональні особливості IoT системи.....	13
	1.3.1 Огляд комп'ютерних систем управління енергоефективністю.....	13
	1.3.2 Технології Інтернету речей (IoT) у сфері енергоефективності.....	16
	1.3.3 Аналітичний огляд існуючих рішень інтеграції IoT в енергоменеджмент.....	17
	1.4 Завдання і мета роботи.....	19
	1.5 Висновки до розділу.....	20
2	Розробка апаратної частини комп'ютерної системи.....	21
	2.1 Технічні вимоги до комп'ютерної системи.....	21
	2.1.1 Вимоги до системи в цілому.....	21
	2.1.1.1 Структура і функціонування системи.....	21
	2.1.1.2 Чисельність і кваліфікація персоналу, що обслуговує систему і режим роботи.....	22
	2.1.1.3 Вимоги до надійності.....	22
	2.1.1.4 Вимоги безпеки.....	23
	2.1.1.5 Вимоги до експлуатації, технічного обслуговування, ремонту і збереження компонентів системи.....	23
	2.1.1.6 Вимоги до захисту інформації від несанкціонованого доступу.....	24
	2.1.1.7 Вимоги до патентної чистоти.....	24
	2.1.1.8 Вимоги до стандартизації й уніфікації.....	25
	2.1.2 Вимоги до видів забезпечення	25
	2.1.2.1 Інформаційне забезпечення системи	25
	2.1.2.2 Технічне забезпечення системи.....	26
	2.1.2.3 Вимоги до організаційного забезпечення.....	26
	2.1.2.4 Вимоги до складу нормативно-технічної документації системи.....	27
	2.2 Розробка структурної схеми IoT системи.....	27
	2.3 Характеристика технічних пристроїв що складають комп'ютерну мережу.....	28

2.3.1	Характеристика існуючих систем.....	28
2.3.2	Аналіз особливостей об'єкта оснащення IoT.....	33
2.3.3	Архітектура інтеграції IoT з енергоменеджмент- системою (EMS).....	37
2.4	Вибір датчиків та виконавчих пристроїв для підсистем.....	42
2.5	Розробка апаратної схеми IoT системи.....	44
2.6	Висновки до розділу.....	45
3	Розробка корпоративної мережі	47
3.1	Розрахунок адресного простору IoT-системи.....	48
3.2	Розробка логічної та фізичної топології.....	51
3.3	Налаштування та перевірка роботи комп'ютерної системи.....	53
3.3.1	Базове налаштування маршрутизації.....	54
3.3.2	Налаштування DHCP.....	56
3.3.3	Налаштування NAT.....	61
3.3.4	Організація віддаленого доступу до системи.....	61
3.3.5	Налаштування VLAN.....	67
3.4	Перевірка працездатності комп'ютерної мережі.....	68
3.5	Висновки до розділу.....	69
4	Розробка програмного компоненту системи	70
4.1	Вибір платформи та обґрунтування архітектури програмного компонента.....	70
4.1.1	Аналіз альтернативних архітектурних підходів.....	70
4.1.2	Обґрунтування вибору програмної платформи Home Assistant.....	72
4.1.3	Обґрунтування вибору апаратної платформи Raspberry Pi.....	73
4.1.4	Загальна архітектура програмного компонента.....	74
4.2	Вибір та обґрунтування технологій і протоколів зв'язку.....	77
4.3	Розробка архітектури баз даних.....	81
4.4	Реалізація логіки автоматизації на основі даних сенсорів	84
4.4.1	Інтерфейс системи Home Assistant.....	85
4.4.2	Структура та принципи створення автоматизацій в Home Assistant.....	87
4.4.3	Сценарії автоматизації	89
4.5	Побудова користувацького інтерфейсу та об'єднання логічної структури конфігурацій.....	93
4.6	Висновки до розділу.....	96
	Висновки.....	98
	Перелік джерел посилання.....	100
	Додаток А. Програмне забезпечення Yaml-конфігурації реалізації автоматизації.....	103
	Додаток Б. Програмне забезпечення налаштування мережі комп'ютерної системи.....	110

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ

IoT – інтернет речей, концепція об'єднання фізичних пристроїв у мережу для автоматизованого обміну даними та керування;

EMS – система керування енергоспоживанням (Energy Management System);

BMS – система керування будівлею (Building Management System);

LAN – локальна мережа;

DHCP – протокол динамічного розподілу адрес вузлам;

NAT – трансляція мережевих адрес (Network Address Translation);

VLAN – віртуальна локальна мережа;

VPN – віртуальна приватна мережа;

MQTT – протокол передачі телеметричних даних між пристроями;

Wi-Fi – бездротова технологія передавання даних у локальній мережі;

Zigbee – бездротовий протокол обміну даними для пристроїв розумного дому;

MCU – мікроконтролер (Microcontroller Unit);

Home Assistant – відкрите програмне середовище для автоматизації розумного дому;

ПК – персональний комп'ютер;

JSON – текстовий формат обміну даними між комп'ютерами;

HTTP – протокол передачі гіпертексту;

°C – градус Цельсія;

Вт (W) – одиниця вимірювання потужності;

кВт·год (kWh) – одиниця вимірювання спожитої енергії;

CO₂ – вуглекислий газ.

ВСТУП

У сучасному світі зростаючі вимоги до енергоефективності житлових приміщень стимулюють розвиток нових технологій, зокрема впровадження рішень на основі Інтернету речей (IoT). Зменшення енергоспоживання є пріоритетом як для власників житла, так і для будівельних компаній, що прагнуть створювати розумні житлові комплекси.

Інтеграція елементів IoT з комп'ютерними системами управління енергоефективністю дозволяє покращити контроль над споживанням енергії, а також забезпечити ефективне управління всіма ключовими аспектами функціонування житлового приміщення.

Метою даної кваліфікаційної роботи є інтеграція IoT-елементів у систему контролю енергоефективності житлового комплексу.

Завданням роботи є аналіз існуючих технологічних рішень, визначення оптимальних способів впровадження IoT системи та мережевих інструментів для забезпечення інтерактивного управління енергоспоживанням та підвищення енергоефективності житлового комплексу.

У процесі дослідження розглянуті методи збору, обробки та використання даних сенсорних пристроїв, архітектура взаємодії між компонентами системи, а також вимоги до безпеки та захисту інформації.

Запропоновані підходи дозволять енергоефективність у житлових приміщеннях та сформувані сучасні моделі управління енергоресурсами, що відповідають тенденціям цифровізації житлових комплексів.

Таким чином, дана кваліфікаційна робота сприятиме розвитку автоматизованих систем керування енергоефективністю, інтеграції IoT-рішень у повсякденне життя та створенню технологій, що забезпечують комфортне, безпечне та раціональне використання ресурсів у житлових приміщеннях.

1 СТАН ПИТАННЯ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Галузь застосування IoT системи контролю енергоефективності

Енергоефективність житлових приміщень визначається як сукупність технологічних і поведінкових заходів, спрямованих на зменшення споживання енергетичних ресурсів за умови збереження комфортних умов для мешканців. Цей підхід передбачає оптимальне використання енергії у процесах опалення, охолодження, вентиляції, освітлення та функціонування побутових систем, що забезпечує не лише економічну вигоду, а й позитивний екологічний ефект. Згідно з аналітичними дослідженнями, житловий сектор є одним із головних споживачів енергії у більшості країн світу, включаючи Україну. Внаслідок цього підвищення енергоефективності у житловому будівництві розглядається як один із пріоритетних напрямів зменшення навантаження на енергетичну систему держави, а також як засіб зниження викидів парникових газів і покращення екологічної ситуації.

Основу енергоефективності житла становлять декілька ключових принципів. Зокрема, важливу роль відіграє мінімізація теплових втрат через зовнішні огорожувальні конструкції — стіни, дах, підлогу, вікна та вентиляційні системи. Значну увагу приділяють використанню енергоощадних будівельних матеріалів, що забезпечують ефективну теплоізоляцію. Крім того, сучасні тенденції передбачають впровадження автоматизованих систем керування енергоспоживанням на основі технологій Інтернету речей (IoT), що дозволяє контролювати та оптимізувати витрати енергії в реальному часі. Також важливим напрямом є застосування відновлюваних джерел енергії, зокрема сонячних батарей, теплових насосів та інших альтернативних рішень.

Структура тепловтрат у житлових приміщеннях свідчить, що найбільше енергії втрачається через неякісну теплоізоляцію огорожувальних конструкцій,

неефективну вентиляцію та недостатню герметичність вікон і дверей. Відповідно, практичні заходи щодо підвищення енергоефективності передбачають проведення термомодернізації, встановлення енергоощадних вікон, впровадження розумних термостатів і сенсорних систем керування мікрокліматом, а також використання альтернативної енергії.

Залежно від рівня споживання енергоресурсів, будівлі класифікуються за класами енергоефективності, від найвищого класу А до найнижчого G. Ця класифікація, що базується на міжнародних стандартах, таких як ISO 50001 та ДБН В.1.2-11:2021, дозволяє об'єктивно оцінити енергетичний стан об'єктів. Наприклад, будівлі класу А характеризуються мінімальним енергоспоживанням і високою ефективністю, тоді як об'єкти класу G зазнають значних тепловтрат і потребують негайної модернізації. В Україні рівень енергоспоживання житлових будівель варіюється залежно від року побудови та впроваджених енергоощадних технологій. Старі будинки можуть споживати до 300 кВт·год/м² на рік, тоді як енергоефективні або пасивні будівлі обмежуються показниками 60 і навіть 15 кВт·год/м² відповідно. Найвищу ефективність демонструють будівлі з нульовим або навіть позитивним енергоспоживанням, які виробляють більше енергії, ніж споживають.

Енергоефективність має комплексний вплив на економічні, соціальні та екологічні аспекти життєдіяльності. Вона сприяє зниженню витрат на комунальні послуги, створенню стабільного та комфортного мікроклімату в приміщеннях, зменшенню викидів вуглекислого газу та стимулюванню інвестицій у новітні будівельні технології. Водночас вона підвищує ринкову вартість нерухомості, оскільки енергоощадні об'єкти користуються більшою популярністю серед потенційних покупців.

На міжнародному рівні активно розвиваються концепції будинків із нульовим енергоспоживанням, що функціонують винятково за рахунок відновлюваних джерел енергії. Також набуває поширення ідея "розумного енергоменеджменту",

що базується на поєднанні IoT-рішень, автоматизованого керування та кіберфізичних систем, які забезпечують високий рівень адаптивності й ефективності енергоспоживання.

У підсумку, енергоефективність житлових приміщень є одним із визначальних чинників сталого розвитку будівництва. Вона не лише знижує навантаження на енергетичні ресурси та довкілля, але й сприяє формуванню сучасного, технологічного та комфортного житлового середовища.

1.2 Характеристика і структура об'єкта впровадження

Житлові будинки можуть мати різні архітектурні рішення, які впливають на їхню енергоефективність. Вибір типу будівлі залежить від багатьох факторів: кліматичних умов, доступності ресурсів, технологічних можливостей та фінансових обмежень. Найпоширенішими є одноповерхові, двоповерхові та багатоквартирні будинки.

Одноповерхові будинки мають просту конструкцію, що робить їх доступними для будівництва та експлуатації. Вони зручні для сімей з дітьми та літніх людей, оскільки не потребують сходів. Однак через велику площу зовнішніх стін такі будівлі можуть мати значні тепловтрати, що потребує якісної теплоізоляції.

Двоповерхові будинки дозволяють ефективніше використовувати площу ділянки, розділяючи житлові та господарські зони. Вони мають меншу площу зовнішніх стін порівняно з одноповерховими, що сприяє зменшенню тепловтрат.

Багатоквартирні будинки мають централізовані системи опалення та водопостачання, що дозволяє ефективно керувати енергоефективністю на рівні всього комплексу. Завдяки спільному використанню ресурсів такі будівлі можуть бути більш енергоефективними, особливо якщо вони оснащені сучасними системами управління енергоспоживанням.

Для дослідження обрано приватний житловий будинок, оскільки саме в таких будівлях мешканці мають найбільший контроль над енергоспоживанням. Вони можуть самостійно впроваджувати технології автоматизації, інтегрувати альтернативні джерела енергії та адаптувати систему управління відповідно до власних потреб.

Енергоефективне планування приватного будинку передбачає правильне розташування приміщень, використання природного освітлення та вентиляції, а також застосування сучасних технологій для зменшення енергоспоживання.

Одним із ключових аспектів є орієнтація будинку. Якщо житлові кімнати розташовані на південній стороні, вони отримують більше сонячного світла, що дозволяє зменшити витрати на опалення взимку. Господарські приміщення, такі як ванні кімнати та комори, краще розміщувати на північній стороні, щоб вони слугували бар'єром проти холоду.

Житловий комплекс, для якого розробляється система моніторингу та управління енергоефективністю, являє собою групу одноповерхових будинків, об'єднаних спільною мережею живлення, зв'язку та управління. Кожен будинок має індивідуальні сенсори температури, освітлення, вологості, а також виконавчі пристрої (розумні лампи, термоголовки, реле тощо). Комунікація між пристроями здійснюється за допомогою бездротових технологій (Wi-Fi, Zigbee) та зведена до центрального вузла управління EMS (сервер Home Assistant), який аналізує дані, забезпечує автоматизацію та оптимізацію енергоспоживання. Також є центральний вузол BMS, який керує всіма підсистемами ЖК (освітлення, камери відеоспостереження) та збирає аналітичні дані з кожної підсистеми EMS будинків.

Схематичне представлення структури житлового комплексу та взаємодії його елементів наведено на рисунку 1.1.

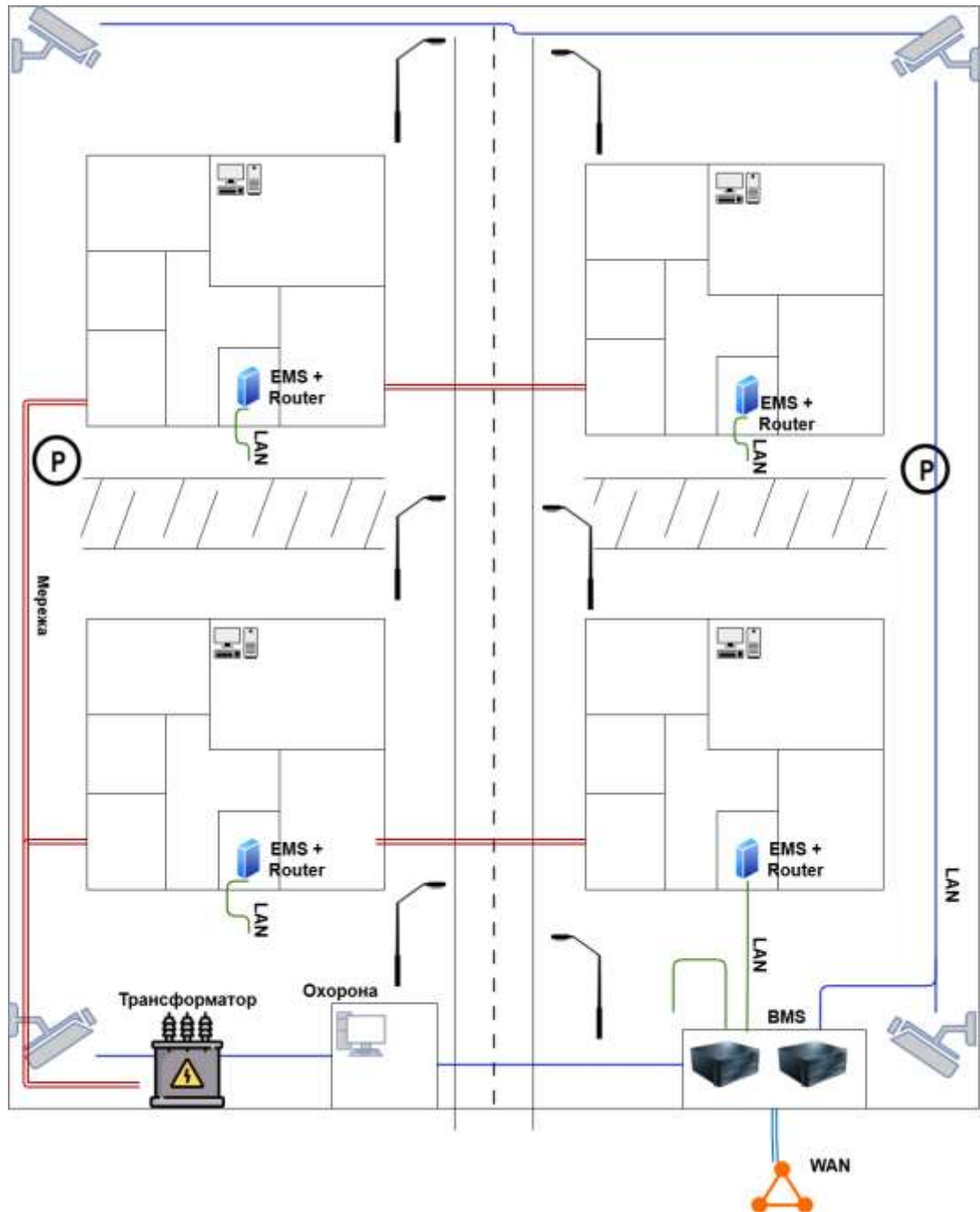


Рисунок 1.1 - Схематичне представлення структури житлового комплексу та взаємодії його елементів.

Інтернет речей відіграє ключову роль у створенні розумних будинків, де всі пристрої взаємодіють між собою. Наприклад, сенсорне освітлення автоматично вмикається та вимикається залежно від рівня освітленості та присутності людей, що значно знижує споживання електроенергії.

На рисунку 1.2 представлено загальну архітектуру інтернету речей.

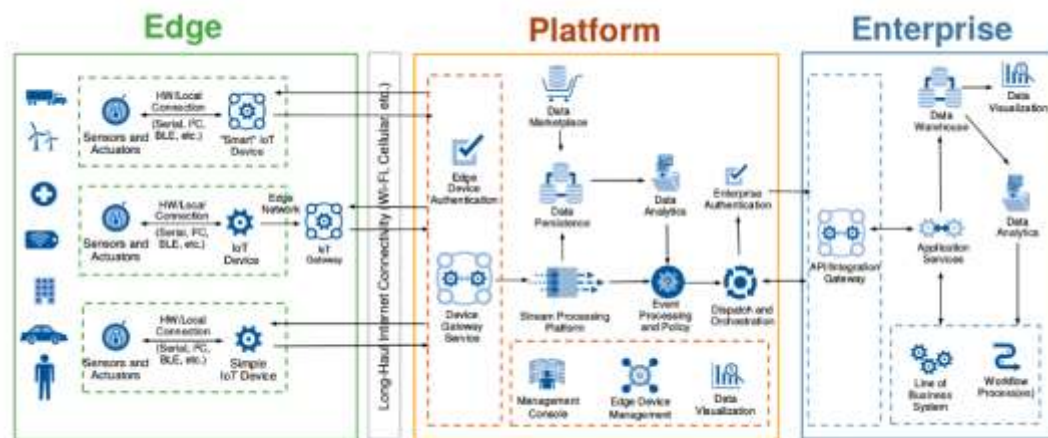


Рисунок 1.2 – Загальна архітектура інтернету речей.

Також активно використовуються системи моніторингу енергоспоживання, які аналізують дані про використання електроенергії та пропонують оптимальні сценарії її економії. Сучасні енергоефективні будинки активно використовують сонячні панелі, теплові насоси та системи рекуперації тепла. Завдяки IoT-рішенням ці технології можуть працювати в єдиній системі, автоматично регулюючи виробництво та споживання енергії.

1.3 Функціональні особливості IoT системи

1.3.1 Огляд комп'ютерних систем управління енергоефективністю

Системи управління енергоефективністю є невід'ємною частиною сучасного житлового та промислового будівництва. Вони дозволяють автоматизувати контроль енергоспоживання, мінімізувати зайві витрати та забезпечити ефективну взаємодію між усіма інженерними мережами будівлі. Головною метою таких систем є оптимізація використання електроенергії, води, теплової енергії та вентиляції, що дозволяє досягти балансу між комфортом мешканців та економічною доцільністю.

Системи енергоефективності можуть бути реалізовані на різних рівнях – від локального (контроль окремих будинків чи квартир) до централізованого (управління житловими комплексами, комерційними будівлями чи промисловими об'єктами). Залежно від поставлених завдань, доцільним є застосування Building Management System (BMS) або Energy Management System (EMS).

Building Management System (BMS) – це система управління будівлею, що контролює всі інженерні процеси, такі як опалення, вентиляція, кондиціонування повітря (HVAC), освітлення, безпека та управління доступом. Вона дозволяє централізовано керувати кожним елементом інфраструктури, оптимізуючи витрати енергії та забезпечуючи комфортні умови проживання чи роботи.

Energy Management System (EMS), у свою чергу, має ширший функціонал, оскільки її головною метою є глобальна оптимізація енергоспоживання. На відміну від BMS, EMS здатна аналізувати загальні витрати на енергію, прогнозувати споживання та інтегрувати відновлювані джерела енергії.

Вибір між BMS та EMS залежить від масштабів проекту та його цілей. Якщо необхідно контролювати інженерні системи окремої будівлі, то BMS є оптимальним рішенням. Вона забезпечує стабільну роботу житлового приміщення, автоматизуючи його ключові процеси. EMS доцільно використовувати у великих житлових комплексах або промислових підприємствах, де важливо не лише керувати окремими системами, а й оптимізувати загальне енергоспоживання, інтегрувати альтернативні джерела енергії та взаємодіяти з електромережею.

Сучасні житлові комплекси активно впроваджують автоматизовані системи управління енергоефективністю. Наприклад, у нових будівлях часто використовуються BMS, які дозволяють централізовано керувати всіма інженерними системами.

У приватних будинках мешканці все частіше встановлюють розумні термостати та системи моніторингу енергоспоживання, що дозволяє значно зменшити витрати на комунальні послуги.

Окрім цього, у великих комерційних комплексах впроваджуються гібридні платформи, які поєднують функціонал BMS та EMS для досягнення максимальної ефективності. Такі системи працюють у реальному часі, керуючи HVAC-системами, освітленням, безпекою, а також прогножуючи споживання енергії відповідно до поточних тарифів.

Тому, найбільш оптимальним рішенням для ЖК є інтеграція системи EMS в систему BMS, де вони мають розмеженні повноваження. EMS керує енергоефективністю в кожному окремому будинку та відправляє аналітику та дані до BMS, яка в свою чергу отримує аналітику з багатьох інших підсистем ЖК, таких як камери, система поливу, система безпеки, пожежна система та ін..

Тому можна зробити висновок, що EMS в даному випадку є частиною архітектури BMS та її підсистемою.

На рисунку 1.3 представлено ієрархічну архітектуру взаємодії BMS з іншими підсистемами.

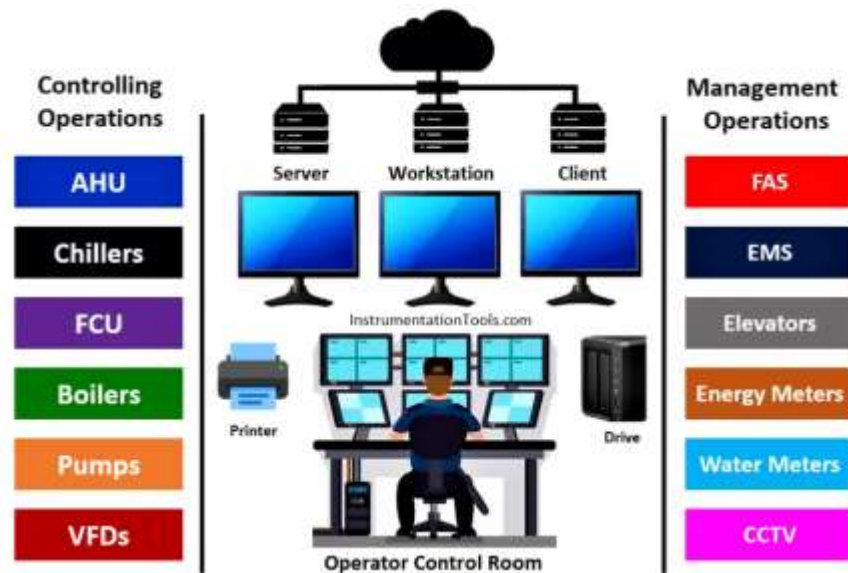


Рисунок 1.3 - Ієрархічна архітектура взаємодії BMS з іншими підсистемами.

1.3.2 Технології Інтернету речей (IoT) у сфері енергоефективності

Інтернет речей (IoT) – це технологія, яка дозволяє об'єднувати фізичні пристрої в єдину мережу для збору, аналізу та обміну даними. У сфері енергоефективності IoT відіграє ключову роль, оскільки дозволяє автоматизувати процеси управління енергоспоживанням, зменшити витрати ресурсів та підвищити загальну ефективність роботи житлових і промислових об'єктів.

Завдяки IoT пристрої можуть взаємодіяти між собою без втручання людини, що дозволяє створювати розумні будинки, де всі системи працюють у синхронізованому режимі. Наприклад, розумні термостати можуть автоматично регулювати температуру в приміщенні залежно від погодних умов, а сенсори освітлення – змінювати рівень яскравості відповідно до природного освітлення.

IoT у сфері енергоефективності базується на кількох ключових принципах, а саме: моніторинг та аналіз даних, автоматизація управління, оптимізація використання ресурсів

У житлових будинках IoT використовується для автоматизації систем опалення, вентиляції, кондиціонування повітря (HVAC), освітлення та безпеки. Наприклад, розумні термостати можуть навчатися поведінці мешканців і автоматично коригувати температуру в приміщенні, що дозволяє зменшити витрати на опалення.

Також активно застосовуються системи моніторингу енергоспоживання, які аналізують дані про використання електроенергії та пропонують оптимальні сценарії її економії. Наприклад, якщо система виявляє, що певні пристрої споживають надмірну кількість енергії, вона може автоматично вимкнути їх або перевести в енергоощадний режим.

У промислових об'єктах IoT дозволяє контролювати роботу обладнання, прогнозувати витрати енергії та запобігати аварійним ситуаціям. Наприклад,

розумні датчики можуть виявляти несправності у системах електропостачання та автоматично повідомляти про них операторів.

У міській інфраструктурі IoT використовується для створення розумних міст, де системи управління транспортом, освітленням та водопостачанням працюють у єдиній екосистемі. Наприклад, автоматизовані системи освітлення можуть змінювати рівень яскравості залежно від часу доби та погодних умов, що дозволяє значно зменшити витрати електроенергії.

1.3.3 Аналітичний огляд існуючих рішень інтеграції IoT в енергоменеджмент

Інтернет речей (IoT) вже активно використовується у сфері енергоефективності, і його інтеграція в системи управління енергоспоживанням дозволяє значно зменшити витрати ресурсів та підвищити загальну ефективність роботи житлових і промислових об'єктів. Одним із ключових напрямів застосування IoT є автоматизовані платформи управління енергією, які дозволяють централізовано контролювати всі аспекти енергоспоживання. Наприклад, платформа SmartHeat використовує IoT для оптимізації роботи теплових насосів, що дозволяє зменшити витрати енергії на опалення. Завдяки інтеграції датчиків температури та вологості система може автоматично регулювати потужність роботи насосів, адаптуючи їх до поточних умов.

Ще одним важливим напрямом є системи моніторингу енергоспоживання, які аналізують дані про використання електроенергії та пропонують оптимальні сценарії її економії. Наприклад, 2Smart Standalone – це IoT-платформа, яка збирає дані про енергоспоживання та автоматично коригує параметри роботи пристроїв. Вона дозволяє не лише відстежувати витрати електроенергії, а й прогнозувати пікові навантаження, що дає змогу уникати перевитрат та знижувати загальне навантаження на електромережу.

Крім того, IoT активно використовується у розумних електромережах, які дозволяють оптимізувати розподіл енергії між споживачами. Завдяки використанню розумних лічильників енергетичні компанії можуть аналізувати споживання електроенергії та коригувати тарифи відповідно до попиту. Це дозволяє не лише зменшити витрати для кінцевих споживачів, а й підвищити загальну стабільність енергосистеми.

У великих офісних центрах та житлових комплексах IoT використовується для автоматизації роботи HVAC-систем, термостатів та навіть жалюзі. Це дозволяє оптимізувати енергоспоживання залежно від рівня заповненості приміщень та погодних умов. Наприклад, у розумних комерційних будівлях IoT-технології дозволяють автоматизувати освітлення, вентиляцію та кондиціонування повітря. Завдяки інтеграції датчиків руху та температури система може автоматично регулювати параметри роботи обладнання, що значно зменшує витрати електроенергії.

У промислових об'єктах IoT дозволяє контролювати роботу обладнання, прогнозувати витрати енергії та запобігати аварійним ситуаціям. Наприклад, розумні датчики можуть виявляти несправності у системах електропостачання та автоматично повідомляти про них операторів. Це дозволяє уникати непередбачених збоїв у роботі обладнання та мінімізувати ризики простою виробничих ліній.

Також IoT використовується у системах управління промисловими процесами, де датчики контролюють рівень споживання енергії на кожному етапі виробництва. Це дозволяє не лише зменшити витрати, а й підвищити загальну продуктивність підприємства. Наприклад, у металургійній промисловості IoT-рішення дозволяють контролювати температуру плавлення металу та автоматично коригувати параметри роботи печей, що значно зменшує витрати енергії.

У міській інфраструктурі IoT використовується для створення розумних міст, де системи управління транспортом, освітленням та водопостачанням працюють у

єдиній екосистемі. Автоматизовані системи освітлення можуть змінювати рівень яскравості залежно від часу доби та погодних умов, що дозволяє значно зменшити витрати електроенергії. Також IoT використовується у розумних транспортних системах, де датчики контролюють рівень завантаженості доріг та автоматично коригують роботу світлофорів. Це дозволяє зменшити затори та оптимізувати витрати пального у громадському транспорті.

Ще одним важливим напрямом є системи управління водопостачанням, де IoT дозволяє контролювати рівень споживання води та запобігати витокам. Наприклад, у деяких містах вже впроваджено IoT-рішення, які дозволяють автоматично виявляти пошкодження трубопроводів та повідомляти про них комунальні служби.

1.4 Завдання і мета роботи

Метою даної кваліфікаційної роботи є інтеграція IoT-елементів у систему контролю енергоефективності житлового комплексу.

Завданням роботи є аналіз існуючих технологічних рішень, визначення оптимальних способів впровадження IoT системи та мережевих інструментів для забезпечення інтерактивного управління енергоспоживанням та підвищення енергоефективності житлового комплексу.

У роботі передбачено реалізацію архітектури взаємодії сенсорів, виконавчих пристроїв та керуючих сервісів, а також моделювання автоматизованої логіки енергозбереження з використанням відкритих платформ і протоколів. Основний акцент зроблено на проєктуванні технічного та програмного компонента системи, налагодженні міжмережевої взаємодії, створенні зручного інтерфейсу користувача та апробації автоматизацій у віртуальному середовищі.

Такий підхід забезпечує підвищення енергоефективності, зменшення експлуатаційних витрат і створює передумови для подальшого впровадження IoT-рішень у побутовому та промисловому секторах.

1.5 Висновки до розділу

У результаті проведеного аналізу в першому розділі було сформовано теоретичне підґрунтя для реалізації системи контролю енергоефективності житлового комплексу з використанням технологій Інтернету речей. Було встановлено, що сучасні підходи до підвищення енергоефективності потребують комплексної інтеграції сенсорних систем, виконавчих пристроїв та програмних платформ керування.

У межах розділу розглянуто основні принципи енергоощадного будівництва, а також структурні особливості житлових приміщень, що є об'єктами впровадження IoT-систем. Визначено, що для приватних одноповерхових будинків доцільним є використання локалізованої системи керування на основі відкритих апаратних та програмних рішень.

Також проведено огляд комп'ютерних систем керування енергоспоживанням, серед яких виокремлено системи BMS і EMS, визначено їх функціональні межі, архітектурну взаємодію та доцільність використання у житловому середовищі. Обґрунтовано роль EMS як підсистеми BMS.

Узагальнення проведеного теоретичного аналізу дозволило сформулювати мету та завдання дослідження, визначити архітектуру майбутньої системи, принципи її побудови та критерії ефективності. Одержані результати стали базою для подальшої розробки апаратної структури системи, її програмного компонента та побудови комп'ютерної мережі, яка забезпечує централізований моніторинг і керування енергоефективністю.

2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ

2.1 Технічні вимоги до комп'ютерної системи

2.1.1 Вимоги до системи в цілому

2.1.1.1 Структура і функціонування системи

Комп'ютерна система контролю енергоефективності житлового приміщення є інтегрованим програмно-апаратним рішенням, що поєднує інтелектуальні сенсорні пристрої, виконавчі елементи, локальну мережу та серверну платформу для обробки інформації та прийняття рішень. Функціональне ядро системи реалізується за допомогою програмного середовища Home Assistant, розгорнутого на мікрокомп'ютері Raspberry Pi з доступом до внутрішньої мережі будинку.

Архітектура системи побудована за принципом модульності та розподіленого керування. Комунікація між компонентами реалізується через бездротові протоколи Zigbee та Wi-Fi, що забезпечують масштабованість та гнучкість розташування обладнання в межах житлового об'єкта. Сенсорні елементи, розташовані у приміщеннях, здійснюють моніторинг температури, вологості, освітлення та присутності, а виконавчі пристрої – зокрема розумні розетки, термоголовки та реле – реалізують алгоритми керування. Передача даних та логіка автоматизації реалізуються через брокер MQTT, що функціонує локально.

Система забезпечує постійний контроль над параметрами мікроклімату і електроспоживанням, адаптуючись до зміни зовнішніх умов та поведінки мешканців, з метою досягнення максимальної енергоефективності без втрати комфорту.

2.1.1.2 Чисельність і кваліфікація персоналу, що обслуговує систему і режим роботи

З огляду на високий рівень автоматизації, система не вимагає постійної присутності обслуговуючого персоналу. У типових умовах достатньо одного фахівця, який виконує періодичний контроль працездатності обладнання та програмного забезпечення, а також реагує на повідомлення про несправності. Такий спеціаліст повинен мати базові знання з адміністрування Linux-систем, розуміти принципи роботи протоколів MQTT, Zigbee, TCP/IP, а також володіти навичками роботи з Home Assistant.

Обслуговування може здійснюватися за змішаним графіком, який включає щотижневі перевірки та щомісячне профілактичне оновлення системи, а також негайне реагування у випадку збоїв. Дистанційне керування допускається, що дозволяє оперативно втручатися в роботу системи без необхідності фізичної присутності.

2.1.1.3 Вимоги до надійності

Надійність системи визначається її здатністю функціонувати безперервно та безвідмовно в умовах змін зовнішнього середовища, нестабільного енергопостачання або часткової втрати зв'язку з окремими пристроями. Для цього передбачається локальна автономна логіка реагування, що дозволяє частково зберігати функціональність навіть у випадку недоступності центрального сервера. Обмін даними між компонентами реалізується з використанням MQTT, який є стійким до короткочасних збоїв зв'язку.

Задля підвищення надійності система має передбачати резервне живлення для серверної частини, можливість автоматичного перезапуску та збереження історичних даних на локальному носії. Критичні компоненти повинні функціонувати з мінімальним часом простою, що не перевищує кількох секунд.

Також має бути реалізовано журналювання подій з можливістю подальшого аналізу причин збоїв.

2.1.1.4 Вимоги безпеки

Система повинна відповідати основним вимогам електричної, пожежної та експлуатаційної безпеки. Усі елементи, що підключаються до електромережі, повинні бути сертифіковані та відповідати національним стандартам з електробезпеки. Встановлення елементів керування у вологих або важкодоступних зонах допускається лише за умови відповідного ступеня захисту (IP-захист не нижче IP44).

Під час експлуатації забороняється використання пошкоджених кабелів або пристроїв із порушеним корпусом. Система повинна мати можливість безпечного відключення живлення, а також бути захищеною від перегріву або перевантаження шляхом використання захисних механізмів у розумних реле чи автоматичних вимикачах. Наявність автоматичних запобіжників або обмежувачів навантаження є бажаною для зменшення ризиків виникнення аварій.

2.1.1.5 Вимоги до експлуатації, технічного обслуговування, ремонту і збереження компонентів системи

Система розрахована на тривалу експлуатацію в умовах житлового приміщення без потреби в щоденному технічному обслуговуванні. Рекомендується здійснювати профілактичну перевірку всіх ключових компонентів щонайменше раз на шість місяців. Такі перевірки повинні включати оновлення прошивок пристроїв, моніторинг стану акумуляторів (у випадку автономного живлення), тестування зв'язку та аналіз журналів Home Assistant.

У разі виявлення несправностей або аномальної поведінки пристроїв, обслуговуючий персонал повинен мати інструкцію з діагностики та типових

способів усунення помилок. Комплект запасних елементів, таких як сенсори або адаптери живлення, бажано зберігати в окремому приміщенні з контрольованим мікрокліматом. Усі компоненти мають бути марковані, задокументовані, та відповідно збережені до потреб подальшого обслуговування або заміни.

2.1.1.6 Вимоги до захисту інформації від несанкціонованого доступу

Захист даних, що передаються та зберігаються в системі, має бути реалізований як на фізичному, так і на програмному рівні. Основною умовою є заборона на прямий доступ до внутрішньої мережі ззовні без застосування методів шифрування та автентифікації. Усі підключення до сервера Home Assistant повинні бути захищені за допомогою паролів, протоколу HTTPS або VPN.

Журналювання доступу, реєстрація дій користувачів та моніторинг спроб несанкціонованого входу мають бути реалізовані через відповідні програмні модулі. Всі критичні налаштування системи повинні бути обмежені для редагування без спеціального рівня доступу. Дані, що зберігаються в системі, мають бути локальними і недоступними для сторонніх сервісів без згоди користувача.

2.1.1.7 Вимоги до патентної чистоти

Усі компоненти, використані під час проектування та впровадження системи, повинні мати ліцензії, що дозволяють їх вільне використання у некомерційних або освітніх цілях. Програмні платформи, зокрема Home Assistant, MQTT-брокери та бібліотеки інтеграцій, мають відкритий вихідний код та використовуються відповідно до ліцензій типу Apache, MIT або GNU GPL. Патентна чистота системи повинна забезпечуватися на території України та не порушувати прав інтелектуальної власності на апаратні або програмні компоненти, що були інтегровані до системи.

2.1.1.8 Вимоги до стандартизації й уніфікації

Система повинна будуватися з урахуванням вимог чинних стандартів на електронне, мережеве та інформаційне обладнання, що забезпечує її сумісність, масштабованість і технічну уніфікацію. Усі компоненти системи – як апаратні, так і програмні – повинні відповідати встановленим міжнародним і національним нормам, включно з IEEE 802.11 (Wi-Fi), IEEE 802.15.4 (Zigbee), MQTT Protocol Standard та ін.

Уніфікація компонентів дозволяє спростити процес модернізації, забезпечити взаємозамінність пристроїв, а також мінімізувати витрати на обслуговування. Крім того, дотримання принципів стандартизації дає змогу інтегрувати систему в ширші інфраструктурні рішення або адаптувати її під інші житлові приміщення без суттєвих змін у структурі.

2.1.2 Вимоги до видів забезпечення

2.1.2.1 Інформаційне забезпечення системи

Інформаційне забезпечення комп'ютерної системи контролю енергоефективності житлового приміщення охоплює процеси збору, збереження, передачі, обробки та візуалізації даних, необхідних для її повноцінного функціонування. Основу інформаційної моделі становлять параметри, що зчитуються з фізичних сенсорів, а саме: температура, вологість, рівень освітленості, стан розеток та споживання електроенергії.

Дані формуються у структурованому вигляді та зберігаються у форматах, сумісних із базами даних, що інтегруються в Home Assistant, зокрема у вигляді записів у MariaDB або SQLite. Передача даних між компонентами системи відбувається через брокер MQTT з використанням тем, що стандартизовано описують джерело, тип і час надходження даних. Це забезпечує логічну впорядкованість інформаційних потоків та їхню обробку в режимі реального часу.

Система також підтримує побудову графіків, звітів та журналів, що формуються на основі накопиченої інформації і відображаються на інтерактивних панелях управління. Важливою складовою інформаційного забезпечення є реалізація механізмів контролю цілісності даних, а також можливість їх резервного копіювання та відновлення в разі збоїв.

2.1.2.2 Технічне забезпечення системи

Технічне забезпечення системи включає сукупність апаратних засобів, які забезпечують виконання її функцій. Основу апаратної частини становить мікрокомп'ютер Raspberry Pi, на якому розгорнуто Home Assistant, що виконує функції обчислювального ядра. Також до складу технічного забезпечення входять датчики температури, вологості, освітленості, розумні розетки, керовані реле та інші IoT-пристрої, що використовують бездротові протоколи Zigbee або Wi-Fi.

Забезпечення безперебійної роботи реалізується через наявність стабілізованого живлення, а за потреби – резервного джерела енергії для критичних компонентів, зокрема сервера та мережевого обладнання. Для підключення пристроїв використовується маршрутизатор з підтримкою мультипротокольної взаємодії, а для координації Zigbee-пристроїв – відповідний USB-координатор. Функціональні характеристики технічних засобів повинні відповідати вимогам безперервної роботи у побутових умовах. Усі компоненти мають бути сумісними між собою як на фізичному, так і на логічному рівні, що дозволяє забезпечити надійну, стабільну та масштабовану систему.

2.1.2.3 Вимоги до організаційного забезпечення

Організаційне забезпечення системи передбачає упорядкування процесів її впровадження, підтримки та взаємодії з користувачами. Основною організаційною умовою є наявність чіткої схеми розміщення пристроїв у межах житлового

приміщення, а також документації, що описує логіку автоматизацій, сценарії керування та умови експлуатації. Користувачі системи повинні бути ознайомлені з основними принципами її роботи, а також мати інструкції з експлуатації у випадку стандартних сценаріїв або аварійних ситуацій. Персонал, який здійснює обслуговування, повинен мати доступ до технічної документації, журналів подій та звітів, що дозволяють оперативно виявити відхилення у роботі системи.

Організація експлуатації повинна базуватись на чітко визначених інтервалах перевірки та оновлення системи. У разі масштабування або модернізації система повинна легко адаптуватися до нових вимог без потреби кардинального перегляду організаційних процедур.

2.1.2.4 Вимоги до складу нормативно-технічної документації системи

До складу нормативно-технічної документації, що супроводжує комп'ютерну систему, входять технічні описи, інструкції з експлуатації, схеми взаємозв'язку компонентів, а також протоколи налаштування мережевих і бездротових з'єднань. Усі документи повинні бути оформлені відповідно до державних стандартів, а також мати версії для оновлень та внесення змін у разі модернізації системи.

Крім того, документація має містити відомості про сертифікацію окремих компонентів, умови гарантійного обслуговування, журнали обліку оновлень програмного забезпечення та шаблони для резервного копіювання. У випадку розширення системи на інші об'єкти або приміщення, нормативно-технічна документація повинна забезпечувати можливість повторного використання або адаптації без необхідності повної переробки проєкту.

2.2 Розробка структурної схеми IoT системи

На структурній схемі зображено логічну архітектуру IoT-системи, інтегрованої з системою енергоменеджменту (EMS). У верхній частині схеми

розміщено блок EMS, який відповідає за централізовану обробку даних та формування команд керування. Нижчі рівні представляють пристрої збору даних та керування – мікроконтролер ESP32, дротові та бездротові сенсори, а також виконавчі пристрої, що взаємодіють через ZigBee або інші інтерфейси. Така структура дозволяє реалізувати ефективний зворотний зв'язок між фізичним середовищем і системою прийняття рішень.

Схему представлено на рисунку 2.1



Рисунок 2.1 – Структурна схема IoT системи.

2.3 Характеристика технічних пристроїв що складають комп'ютерну мережу

2.3.1 Характеристика існуючих систем

У сучасному житловому будинку ключову роль відіграють інженерні системи, які забезпечують комфорт, безпеку та енергоефективність. Водночас, із розвитком технологій Інтернету речей (IoT), з'явилася можливість інтегрувати численні підсистеми житла в єдине кероване середовище. Таке об'єднання дозволяє

не тільки автоматизувати окремі процеси, але й оптимізувати споживання ресурсів, знизити витрати та підвищити загальну ефективність функціонування будинку.

Розглянемо основні підсистеми, що доцільно інтегрувати з IoT для досягнення цілей підвищення енергоефективності житлового простору.

Система освітлення

Автоматизація освітлення є одним із базових напрямів впровадження IoT у побуті. У типовому сценарії освітлення регулюється на основі показників присутності мешканців, часу доби або рівня природного освітлення. Для реалізації подібного функціоналу зазвичай використовуються датчики руху, сенсори освітленості та розумні реле. Застосування таких рішень дозволяє знизити споживання електроенергії в середньому на 15–25% у порівнянні з традиційною системою.

На рисунку 2.2 зображено типову схему інтеграції системи освітлення з контролером IoT.

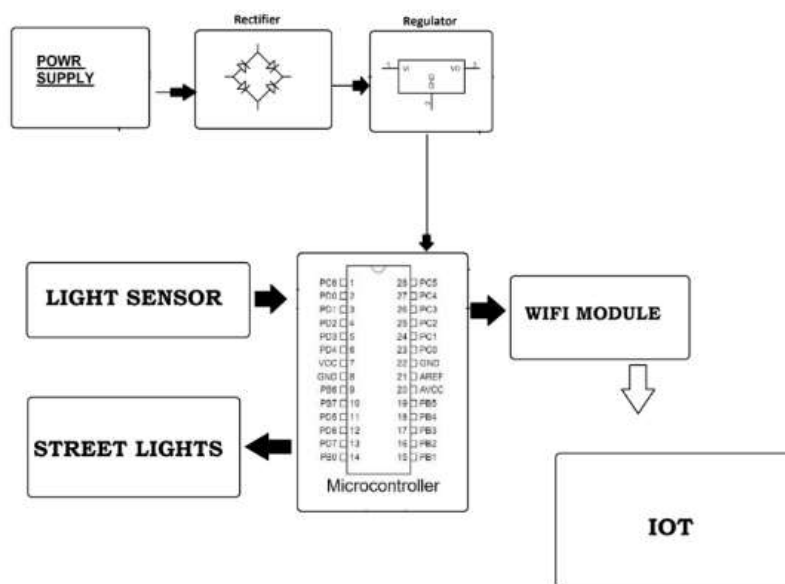


Рисунок 2.2 – Типова схема інтеграції системи освітлення з елементами IoT системи.

Система опалення та кондиціонування

Основними виконавчими елементами такої системи є термостати, клапани, датчики температури та вологості. Інформація від сенсорів надходить до центрального контролера, який на основі запрограмованої логіки або ШІ-моделей приймає рішення щодо вмикання/вимикання опалення або кондиціонера.

Це дозволяє досягти значного зниження теплових втрат, особливо в нічний час або при відсутності мешканців у будинку.

Система електроживлення та контроль споживання

Ще одним критичним компонентом є система електроживлення. Вона включає контроль напруги, струму, потужності окремих груп споживачів, виявлення аномальних навантажень та автоматичне відключення аварійних ліній.

Завдяки IoT-модулям (наприклад, PZEM-004T або Sonoff POW), споживання енергії можна моніторити в реальному часі. Ці дані аналізуються системою енергоменеджменту (EMS), що дозволяє виявляти перевантаження, неефективні пристрої або визначати графіки споживання.

Автоматизація вмикання/вимикання живлення дозволяє зменшити нічне або пікове навантаження на мережу. Крім того, користувач отримує змогу відстежувати витрати у зручному форматі – через мобільний застосунок або веб-інтерфейс.

Розетки та побутові електроприлади

Розумні розетки – ще один важливий елемент у структурі IoT-дому. Вони дозволяють дистанційно вмикати або вимикати підключені пристрої, контролювати їхнє енергоспоживання та навіть створювати автоматизовані сценарії (наприклад, вимкнення всіх розеток при виході з дому).

Завдяки таким функціям користувач може мінімізувати втрати енергії, спричинені пристроями у режимі "очікування" або при забутому увімкненому обладнанні (праска, обігрівач тощо).

Таким чином, усі зазначені підсистеми можуть бути ефективно інтегровані у єдину IoT-екосистему житлового приміщення. Їх об'єднання через енергоменеджмент-систему забезпечує автоматизований контроль, оптимізацію витрат ресурсів і підвищення загальної енергоефективності будинку.

У рамках даної кваліфікаційної роботи об'єктом дослідження було обрано житловий комплекс одноповерхових будинків, розташований у передмісті міста Дніпра. Основною причиною вибору саме цього об'єкта є його відповідність актуальним технічним, архітектурним та функціональним характеристикам, які дають змогу апробувати розроблену систему енергоефективного керування на базі IoT у реальних умовах. Для зручності моделювання, подальшу розробку та опис IoT системи буде проведено для одного з цих одноповерхових будинків.

Будинок має одноповерхову конструкцію, загальною площею 118 м². У його плані передбачено п'ять основних житлових приміщень: кухню-їдальню, вітальню, дві спальні та санвузол, а також технічне приміщення, де розміщується обладнання систем опалення та електроживлення. Додатково присутні господарське приміщення для зберігання речей, невелика тераса та підсобне приміщення (кладова). Для кожної з цих зон передбачається інтеграція певних IoT-рішень відповідно до їх призначення та характеру енергоспоживання.

Будинок побудований із використанням енергоефективних будівельних матеріалів: стіни з газоблоку з утепленням, вікна – п'ятикамерні склопакети з енергозберігаючим покриттям, дах утеплений мінеральною ватою. Таке конструктивне рішення створює передумови для ефективного використання системи керування мікрокліматом з урахуванням мінімізації тепловтрат.

Будівля має відкриту планувальну структуру з вільним розміщенням меблів, що дозволяє вільно встановлювати датчики, контролери та інші компоненти системи у стратегічно важливих точках. Електропроводка будинку виконана за

сучасними стандартами з наявністю окремого електрощита, що спрощує підключення виконавчих пристроїв для автоматизації електроживлення.

На рисунку 2.3 представлено загальне планування будинку з позначенням основних зон.



Рисунок 2.3 – План будинку з позначенням приміщень

У будинку реалізовано автономну систему опалення на базі електричного котла, підключеного до радіаторів опалення в кожному з приміщень.

Додатково встановлено теплу підлогу у ванній кімнаті та кухні. Система освітлення передбачає наявність стельових світильників із можливістю димерування у вітальні та спальні. Система електроживлення охоплює всі приміщення, кожна група виведена на окремий автомат у щитку.

Для розробки IoT-системи це дозволяє реалізувати контроль:

1. За температурою в кожному приміщенні окремо;
2. За споживанням електроенергії окремими групами споживачів;
3. За рівнем освітлення та активністю користувачів.

Крім того, будинок оснащено базовою охоронною сигналізацією та відеонаглядом, що надає змогу інтегрувати елементи безпеки в загальну систему моніторингу та керування.

Ще однією вагомою причиною вибору саме цього об'єкта є можливість вільного доступу до всіх приміщень, інженерних комунікацій та обладнання. Це дає змогу оперативно вносити зміни в конфігурацію системи, тестувати роботу сенсорів та виконавчих пристроїв, проводити знімання показників для подальшого аналізу.

Таким чином, обраний об'єкт є оптимальним з точки зору розміру, конфігурації, інженерної підготовки та технічної придатності для реалізації і тестування IoT-системи енергоефективності. Він дає змогу змоделювати умови повноцінного використання технологій Інтернету речей у контексті реального житлового приміщення, що дозволяє максимально наблизити результати проєкту до практичних потреб користувачів.

2.3.2 Аналіз особливостей об'єкта оснащення IoT

Планування житлового простору відіграє ключову роль у процесі розробки та впровадження енергоефективних систем керування на базі IoT. Від розташування приміщень, конфігурації інженерних мереж, наявності та доступності інфраструктури залежить ефективність збору даних, якість реагування системи на зміну параметрів середовища та зручність інтеграції технічних рішень.

Об'єктом розгляду є одноповерховий житловий будинок із загальною площею 118 м². Його просторове зонування та інженерне забезпечення дозволяють органічно впровадити елементи розумного будинку та налаштувати індивідуальне керування ключовими підсистемами.

Житловий будинок має класичне планування, яке включає наступні функціональні зони:

1. Кухня-їдальня – суміщене приміщення площею близько 20 м² з панорамним вікном, що забезпечує хороше природне освітлення. У зоні кухні встановлено побутову техніку: холодильник, варильну поверхню, духову шафу, витяжку, мікрохвильову піч, чайник. Усі пристрої підключені через розетки, винесені на окрему групу в щиті.
2. Вітальня – просторе приміщення площею близько 25 м², яке слугує центральним місцем відпочинку. Тут розташовані телевізор, освітлення (люстра та LED-підсвітка), а також елементи меблів із вбудованими розетками. Наявне панорамне вікно та вихід на терасу.
3. Спальня 1 та Спальня 2 – два ізольовані приміщення, кожне площею близько 12–14 м², облаштовані базовим освітленням, настільними лампами, розетками біля ліжок, вікнами з сонцезахисними ролетами.
4. Санвузол – суміщена ванна кімната з теплою підлогою, витяжним вентилятором, електричним бойлером, дзеркалом з підсвіткою.
5. Технічне приміщення – окрема кімната площею 6 м², у якій встановлено електричний котел, розподільчий електрощит, резервний акумулятор, а також розміщено мережеве обладнання.
6. Кладова та коридор – допоміжні приміщення, які з'єднують основні зони.
7. Загальна структура планування зображена на рисунку 2.3.

Опалення в будинку реалізовано на базі теплового насосу типу "повітря–вода" номінальною тепловою потужністю 9 кВт, встановленого на зовнішній стіні технічного приміщення. Тепловий насос забезпечує високий коефіцієнт перетворення електричної енергії в теплову ($COP \approx 3,5–4,0$ при температурі зовнішнього повітря до $+7^{\circ}C$), що дозволяє значно зменшити експлуатаційні витрати порівняно з традиційними електричними котлами.

Гідравлічна частина системи включає буферну ємність об'ємом 100 л, що забезпечує стабілізацію температури теплоносія, зменшує кількість пусків

компресора та дозволяє безперервно циркулювати теплоносію навіть у випадках часткового або повного перекриття терморегуляторів у кімнатах. Буфер також відіграє роль гідравлічного роздільника між контуром теплового насоса та внутрішньою розводкою.

Внутрішня система опалення виконана на основі алюмінієвих радіаторів, рівномірно розміщених по всьому периметру будинку. У кожному приміщенні встановлено від одного до трьох радіаторів з дистанційними електричними термоприводами, що керуються за допомогою протоколу ZigBee 3.0. Це дозволяє реалізувати зональне (кімнатне) управління опаленням, оптимізуючи споживання енергії та забезпечуючи комфорт користувачам.

Температура подачі теплоносія в систему опалення зазвичай не перевищує 45–50°C, що відповідає вимогам енергоефективної низькотемпературної системи. Завдяки збільшеній площі теплообміну радіаторів та високій чутливості терморегуляторів, навіть при такій температурі досягається необхідний рівень теплового комфорту в приміщеннях.

Окремо функціонує система "тепла підлога" у кухні, санвузлі та коридорі. Вона живиться від того ж теплового насоса через окремий змішувальний вузол з сервоприводом і регулятором температури подачі. Робоча температура в контурі теплої підлоги складає 30–35°C.

У кваліфікаційній роботі передбачається повна інтеграція системи опалення в IoT-мережу будинку. В рамках цієї модернізації будуть встановлені виконавчі реле, блоки дистанційного керування тепловим насосом, а також температурні сенсори в кожному приміщенні. Управління системою здійснюватиметься на основі заданих сценаріїв, температурних порогів, а також часових графіків, що дозволить забезпечити максимальну енергоефективність та автоматизацію опалення без втрати комфорту.

Електроживлення організовано через трифазну систему із загальним щитом на 24 автомати. Всі приміщення розділено на окремі групи: освітлення, розетки, побутова техніка, електроопалення.

У щиті передбачено вільні місця для встановлення модулів реле та датчиків струму, що забезпечує гнучкість подальшої інтеграції. Кабельна система прокладена під штукатуркою, використано кабель ВВГнг-LS відповідного перерізу. Земляний контур реалізовано відповідно до норм ПУЕ.

У кімнатах в середньому встановлено по 4–6 подвійних розеток, розміщених біля робочих зон, ліжок та меблів. У зоні кухні та ванної встановлено розетки із вологозахистом.

Освітлення реалізовано комбіновано – люстри, LED-стрічки, споти, локальне підсвічування. Усі лінії підключені за допомогою димерів, щоб мати можливість за допомогою ШІМ сигналів керувати рівнем освітлення. Фізичні вимикачі під'єднані до ESP32, таким чином унеможливується сценарій роботи освітлення, коли на контактах завжди присутня напруга. ESP32 зчитує положення фізичного вимикача та дані, що надходять з EMS. В стандартному режимі, ESP32, як виконавчий пристрій, керується даними отриманими з EMS. У випадку, коли EMS знаходиться в аварійному стані, тобто відбувається втрата зв'язку, ESP32 керується даними положення фізичного вимикача.

Головний електрощит розташований у технічному приміщенні. Там же встановлено УЗО, автоматичні вимикачі, реле напруги, контролер навантаження та резервне живлення (інвертор + акумулятор на 3 кВт*год).

Усередині щита передбачено місце для встановлення контролера системи, а також датчиків напруги, температури, струму.

Комунікаційні канали для IoT-пристроїв проводитимуться за тією ж схемою, що і електрика – через кабель-канали або гофротрубу. У разі потреби датчики будуть підключені по бездротових каналах (Wi-Fi або Zigbee).

Інтернет до будинку підведено через оптоволоконний кабель зі швидкістю до 100 Мбіт/с. Роутер встановлено в технічному приміщенні, поруч із щитом, для зручності підключення контролерів та шлюзів.

Таким чином, планування житла, його інженерні системи та мережі повністю відповідають вимогам для впровадження сучасної IoT-системи. Будинок має добре продуману інфраструктуру, що спрощує встановлення сенсорів, виконавчих пристроїв і контролерів. Завдяки наявності централізованого щита, окремих груп електроживлення та сучасного інтернет-зв'язку забезпечується повна технічна готовність об'єкта до впровадження розумної системи керування енергоефективністю.

2.3.3 Архітектура інтеграції IoT з енергоменеджмент-системою (EMS)

Інтеграція елементів Інтернету речей (IoT) з енергоменеджмент-системою (EMS) у житловому будинку має ключове значення для досягнення практичної енергоефективності. Така архітектура дозволяє системі не лише фіксувати енергоспоживання в реальному часі, а й активно реагувати на зміни та автоматизовано керувати інженерними підсистемами відповідно до заданих сценаріїв.

Основна мета цієї архітектури – досягнення максимальної автономності, зменшення енергетичних втрат і створення зручного інтерфейсу для користувача без зайвих проміжних систем або хмарних сервісів, які створюють додаткові ризики для безпеки або збільшують час відгуку системи.

IoT-система енергоефективного керування складається з трьох ключових рівнів: фізичний рівень(датчики, реле, виконавчі пристрої), передавальний рівень (бездротові та дротові протоколи зв'язку (Wi-Fi, Zigbee, MQTT)), Логічний рівень (EMS)(платформа обробки даних, логіка керування, інтерфейс користувача).

У цій роботі обрано реалізацію, яка поєднує локальну обробку даних та пряме керування без залежності від хмарних рішень.

В якості EMS центром обробки виступає Raspberry Pi 4B з встановленою платформою Home Assistant, яка дозволяє організувати повноцінну енергоменеджмент-систему на локальному рівні.

На рисунку 2.4 представлено загальну архітектуру системи EMS.

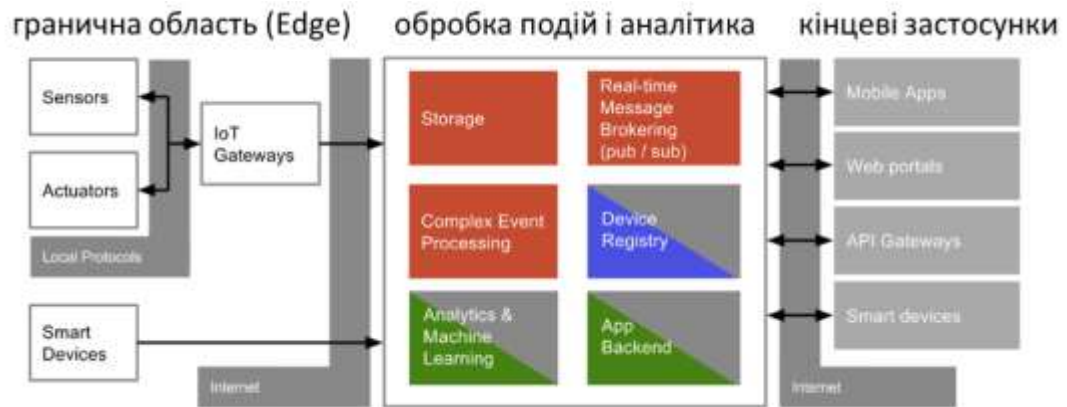


Рисунок 2.4 – Загальна архітектура системи EMS.

Передача даних від сенсорів до серверу є критичним етапом у роботі системи. У цьому проєкті використовуються два основні типи комунікації:

1. Wi-Fi – для сенсорів на базі ESP32, які передають дані безпосередньо на MQTT-брокер.
2. Zigbee – для енергоефективних пристроїв із низьким енергоспоживанням (термодатчики, сенсори руху), які з'єднуються через шлюз Sonoff ZBBridge.

Більшість датчиків, що використовуються, є пасивними – вони лише передають зафіксовані дані через мікроконтролер.

В якості мікроконтролера, що буде посередником між пасивним датчиком та EMS було обрано ESP32. Вибір був обумовлений тим, що ESP має багато логічних входів/виходів, має низьке енергоспоживання та може працювати в режимі прийомо-передачі, що дуже важливо, особливо при проектування підсистем керування побутовими приладами, тобто «розумними розетками».

Опишемо типовий принцип роботи пасивного датчика в парі з ESP32. Наприклад, датчик температури DHT22 зчитується через ESP32, який через MQTT надсилає значення температури на локальний сервер.

Ця логіка однакова для більшості сенсорів та представлена на рисунку 2.5

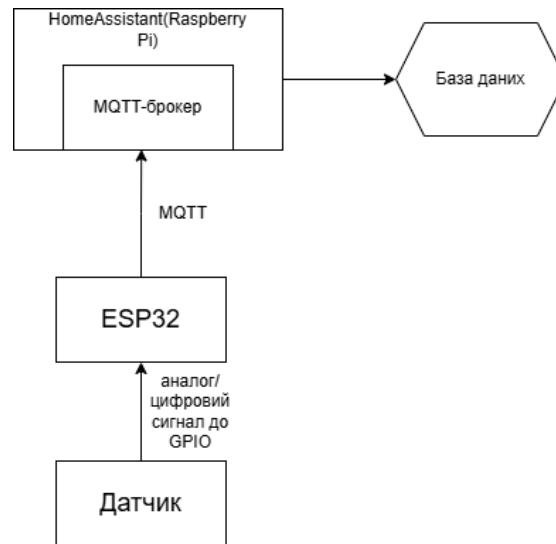


Рисунок 2.5 – Логіка роботи пасивного датчика в парі з ESP32.

- Датчики передають цифрові/аналогові значення на ESP;
- ESP формує MQTT-повідомлення у форматі JSON;
- Home Assistant зчитує повідомлення через вбудований MQTT-брокер (Mosquitto);
- Дані зберігаються у локальній базі даних (SQLite або MariaDB);
- Передача виконується або за графіком (кожні 10–60 секунд), або за подією (зміна стану);

На Raspberry Pi встановлено Home Assistant OS, Mosquitto (MQTT), Zigbee2MQTT (для шлюзу), MariaDB (для збереження історії). Це дозволяє повністю ізолювати систему від зовнішніх загроз і мінімізувати час відгуку при прийнятті рішень.

В таблиці 2.1 показано взаємодію EMS з основними системами будинку.

Таблиця 2.1 – Взаємодія EMS з основними системами будинку.

Опалення:	На основі даних, отриманих від температурних датчиків та присутності людей, EMS подає сигнал закрити/відкрити термостати на радіаторах та запустити рециркуляторний насос, який перекачує теплоносій з буферу, де ТН підтримує постійну температуру.
Освітлення:	При спрацюванні сенсора присутності в приміщенні, рівня освітленості та в залежності від часу доби EMS надсилає команду на виконуючий пристрій(ESP32) надіслати відповідний сигнал на диммер, що керує відповідним освітлювальним пристроєм.
Електроживлення:	«Розумні розетки» передають значення енергоспоживання на EMS, яке приймає рішення про ввімкнення/вимкнення електроживлення в залежності від сценарію роботи.
Безпека:	спрацювання сенсора відкриття дверей після 23:00 активує сирену або надсилає push-повідомлення.

Рішення приймаються за допомогою вбудованого механізму автоматизації у Home Assistant або через скрипти Node-RED для складніших логік.

Нижче наведено приклади реальних сценаріїв, реалізованих у системі:

Опалення:

- Якщо хоча б один радіатор відкритий (термоголовка відкривається) – запускається циркуляційний насос і при потребі тепловий насос, забезпечуючи нагрівання води в буфері.
- Якщо всі термоголовки закриті – циркуляційний насос вимикається для запобігання гідроудару, а тепловий насос переходить у режим підтримки температури буфера або вимикається при досягненні мінімальної температури.

- При зниженні температури в кімнаті нижче заданої користувачем – відкривається відповідний термодіапазон та активується опалення.
- При досягненні встановленої температури в кімнаті – відповідна термоголова закривається, а насос продовжує роботу лише за наявності відкритих клапанів.
- Якщо будинок залишає користувач (режим «Відсутність») – система плавно знижує температуру опалення та, за можливості, вимикає тепловий насос, підтримуючи мінімальний рівень тепла в буфері.
- У разі аварійного відключення теплового насоса – система сповіщає користувача через Telegram і переходить у режим роботи з буфером, знижуючи навантаження на радіатори.

Освітлення:

- Вмикання освітлення за допомогою фізичного вимикача або через дистанційне керування EMS.
- Якщо датчик присутності виявляє рух у кімнаті, а рівень природного освітлення нижчий за встановлений поріг – освітлення автоматично вмикається або підвищується за допомогою димера.
- При відсутності руху в кімнаті протягом заданого часу освітлення поступово вимикається або приглушується до мінімального рівня.
- Користувач може вручну регулювати рівень яскравості через систему EMS або фізичний димер.
- При виході з кімнати освітлення вимикається фізичним вимикачем, при цьому система EMS оновлює статус для коректної роботи автоматичних сценаріїв.
- Якщо EMS виходить з ладу, фізичні вимикачі зберігають можливість включення/вимикання освітлення в автономному режимі.

Під час реалізації архітектури було враховано такі виклики:

- Взаємна несумісність пристроїв: не всі датчики підтримують MQTT або Zigbee, тому деякі елементи потребують шлюзів або додаткового кодування прошивок (Tasmota, ESPHome).
- Залежність від локальної мережі: у разі її відмови автоматизації можуть не спрацювати, тому реалізовано резервне живлення.
- Складність початкового налаштування: системи типу Home Assistant потребують досвіду для правильного налаштування YAML, брокера, шлюзу.

Проте всі ці проблеми вирішуються у процесі інсталяції, що в цілому дозволяє вийти на стабільну, автономну роботу системи.

У результаті було розроблено архітектуру, яка базується на відкритих рішеннях і легко адаптується під нові типи сенсорів, сценарії чи будівлі. Вона повністю локальна, гнучка, економічно доцільна і технічно зріла для практичного використання в приватному секторі або малих житлових об'єктах. Її головна перевага – максимальна енергоефективність за мінімального втручання користувача.

2.4 Вибір датчиків та виконавчих пристроїв для підсистем

Для забезпечення функціональності розумної системи управління енергоефективністю було обрано низку датчиків і виконавчих пристроїв, які відповідають технічним вимогам, легко інтегруються в інфраструктуру на базі ESP32 та підтримують стандартизовані протоколи обміну даними.

У підсистемі освітлення застосовується пасивний інфрачервоний сенсор присутності PIR HC-SR501, що виявляє рух людини в приміщенні в межах кута до 120° і дальності до 7 метрів. Для адаптації освітлення до природного світла використовується цифровий датчик освітленості BH1750, що працює за інтерфейсом I²C та забезпечує точність у широкому діапазоні освітлення (від 1 до 65535 люкс). В якості виконавчого елемента керування яскравістю світлодіодних

світильників застосовується димер на MOSFET-модулі, керований ШІМ-сигналом (PWM) з мікроконтролера.

Для реалізації підсистеми опалення обрано цифрові температурні сенсори DS18B20, що мають інтерфейс 1-Wire, високу точність вимірювання температури (до $\pm 0,5$ °C) і можливість каскадного підключення до одного GPIO. Регулювання температури в кожному приміщенні здійснюється за допомогою термостатичних головок TuYa TRV, що підтримують зв'язок за протоколом Zigbee. Вони мають низьке енергоспоживання, живлення від батарейок, внутрішній дисплей і механізм керування клапаном радіатора.

Підсистема моніторингу енергоспоживання включає пристрої Sonoff POW R2 та PZEM-004T V3.0. Sonoff POW R2 – це Wi-Fi модуль з реле на 15A та вбудованим ватметром, що забезпечує моніторинг напруги, сили струму, потужності та накопиченої енергії, з можливістю передачі даних через MQTT. Модуль PZEM-004T, підключений через UART до ESP32, дозволяє здійснювати вимірювання електричних параметрів у різних частинах мережі безпосередньо, з високою точністю і незалежним живленням.

Центральним елементом усіх підсистем виступає мікроконтролер ESP32 DevKit V1, що має вбудований модуль Wi-Fi, підтримує до 34 GPIO, інтерфейси I²C, UART, PWM та інші необхідні для інтеграції сенсорів. ESP32 виконує роль вузла збору та обробки даних, забезпечує їх передачу до серверної частини системи управління за допомогою MQTT, а також здатен працювати автономно з локальною логікою реагування.

Таким чином, обране апаратне забезпечення дозволяє повністю реалізувати логіку автоматизованого контролю за освітленням, опаленням, енергоспоживанням із врахуванням вимог до енергоефективності, стабільності, масштабованості та інтеграції в локальну мережу будинку.

2.5 Розробка апаратної схеми IoT системи

У якості EMS було використано мікрокомп'ютер Raspberry Pi 4. Він під'єднується до роутеру за допомогою LAN кабелю. Було надано перевагу дротовому зв'язку для покращення показників часу, який витрачається на передачу сигналу та для більшої безпеки системи.

Всі пасивні датчики під'єднані до мікроконтроллера ESP32 до його портів GPIO. Також до мікроконтроллера під'єднані деякі виконавчі пристрої, такі як димер, насос системи опалення, тепловий насос. ESP32 під'єднана до локальної Wi-Fi мережі за допомогою роутеру. Вона може отримувати та відправляти дані до EMS в рамках цієї мережі бо має власну статичну IP адресу.

Схема апаратної реалізації представлена на рисунку 2.6.

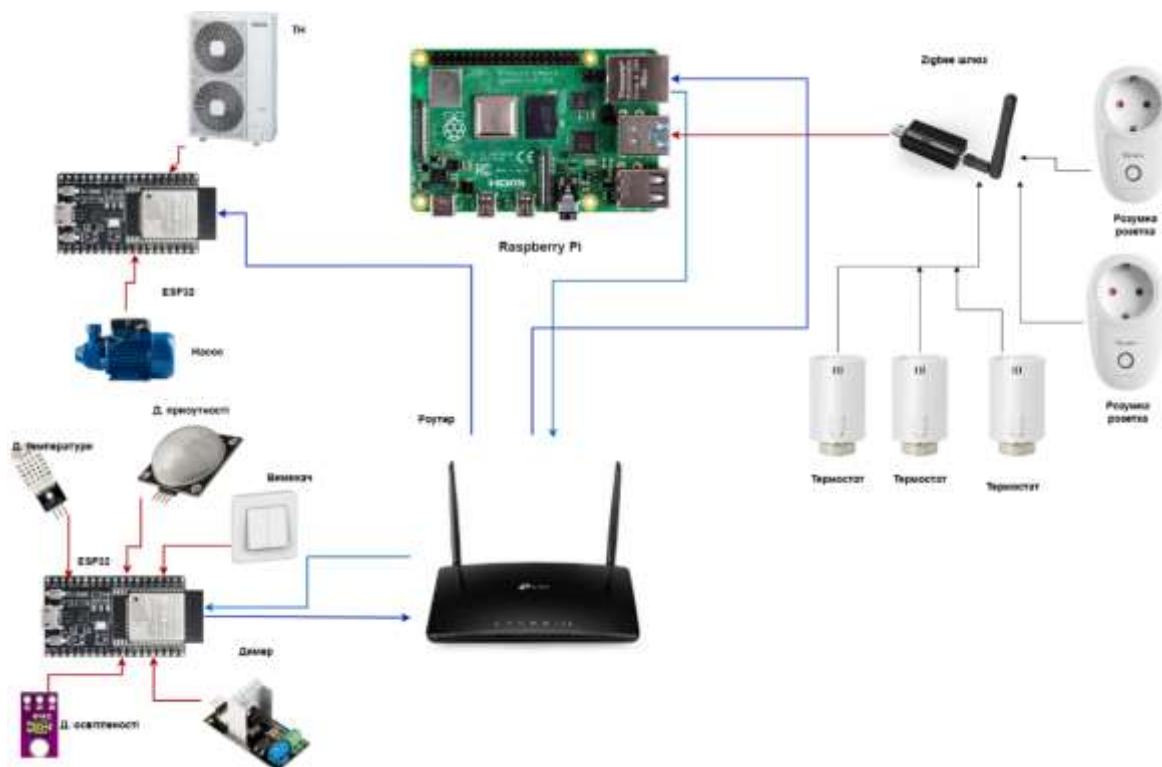


Рисунок 2.6 – Апаратна реалізація системи EMS.

Термостати є розумними виконавчими пристроями, які отримуючі сигнал від EMS можуть керувати температурою радіаторів, перекриваючи чи надаючи протік теплоносія. Розумні термостати працюють за протоколом Zigbee, тому не можуть обмінюватися даними з EMS в Wi-Fi мережі. Вони під'єднуються до EMS за допомогою Zigbee шлюза, який під'єднується до Raspberry Pi 4 за допомогою USB кабелю.

Розумні розетки, як і термостати не мають можливості до використання Wi-Fi протоколу і також під'єднанні бездротовим зв'язком до Zigbee шлюза.

Через обмежений функціонал програмного пакету Cisco Packet Tracer, у наступних розділах кваліфікаційної роботи використовувалася спрощена модель цієї архітектури, де «розумні датчики» виконують роль ESP32 разом з пасивним датчиком. Тобто датчики самі передають дані в локальну мережу.

Запропонована архітектура може бути реалізована як окреми курсовий проєкт чи частина іншої кваліфікаційної роботи. В наступних розділах в основному буде розглядатися спрощена версія цієї архітектури з урахуванням програмних можливостей середовища розробки.

2.6. Висновки до розділу

У другому розділі було здійснено комплексну розробку апаратної частини системи управління енергоефективністю житлового приміщення з використанням технологій Інтернету речей. Проведений аналіз показав, що ефективність функціонування такої системи безпосередньо залежить від правильності вибору архітектури, структурної організації компонентів, їх топологічного розміщення та способу інтеграції з енергоменеджмент-системою (EMS).

У результаті аналізу підсистем, що найбільше впливають на загальне енергоспоживання житлового об'єкта, було обґрунтовано доцільність інтеграції IoT-рішень у такі напрямки: освітлення, опалення, моніторинг енергоспоживання.

Для кожної з підсистем були визначені оптимальні типи сенсорів і виконавчих елементів, обрано апаратну платформу для реалізації керування та передачі даних, а також обґрунтовано вибір відповідних протоколів зв'язку.

Як об'єкт дослідження, розглянуто житловий комплекс, що складається з одноповерхових житлових будинків площею 118 м². Було детально проаналізовано планування приміщень, розташування основних інженерних вузлів, конфігурацію електромережі, локальної мережі та інтернет-з'єднання.

Особливу увагу приділено архітектурі інтеграції IoT з EMS. Було обґрунтовано вибір локальної моделі взаємодії компонентів із використанням мікроконтролерів ESP32, платформи Home Assistant на базі Raspberry Pi та MQTT-брокера Mosquitto.

Описано принцип роботи сенсорів у контексті IoT-архітектури: від збору аналогових та цифрових даних через ESP32 до передачі MQTT-повідомлень на локальний сервер. У межах реалізації було враховано як дротові, так і бездротові способи підключення, що дало змогу охопити як пристрої, розташовані у складнодоступних місцях, так і мобільні компоненти.

Окремим етапом було здійснено вибір типів сенсорів і виконавчих пристроїв відповідно до функціонального призначення підсистем. У підсистемі освітлення реалізовано керування за допомогою димера та сенсорів присутності/освітленості.

В опаленні використано термоголовки Tuua та температурні датчики DS18B20. У моніторингу енергоспоживання – модулі типу Sonoff POW та PZEM-004T.

Таким чином, у другому розділі було виконано повноцінну технічну реалізацію апаратної частини IoT-системи для енергоефективного управління житловим будинком. Запропонована архітектура та компонентна база можуть бути використані як прототип для впровадження у реальних житлових об'єктах, що підтверджує прикладну цінність виконаної роботи.

3 РОЗРОБКА КОРПОРАТИВНОЇ МЕРЕЖІ

У цьому розділі виконується побудова мережевої інфраструктури розумного житлового приміщення, яке є об'єктом розробленої IoT-системи енергоефективності. Метою є створення моделі структурованої комп'ютерної мережі, яка забезпечує стабільну, безпечну та розмежовану взаємодію між усіма компонентами системи: сенсорами, виконавчими пристроями, мікроконтролерами, сервером керування, а також клієнтськими пристроями користувачів.

Фізичною основою реалізованої IoT-системи у межах апаратної частини виступають мікроконтролери ESP32, підключені до сенсорів та виконавчих елементів, з передачею даних через Wi-Fi до серверної частини (на базі Raspberry Pi, з розгорнутим MQTT-брокером і платформою Home Assistant). Однак через технічні обмеження програмного середовища Cisco Packet Tracer, яке використовується для моделювання, у віртуальній мережі були застосовані узагальнені «розумні пристрої», які імітують функціональність реальних IoT-вузлів.

В моделі не використано фізичні підключення сенсорів до ESP32 (GPIO, UART, I²C), оскільки такі з'єднання не підтримуються у Cisco. Натомість сенсори замінені на віртуальні IoT-пристрої з можливістю безпосередньої передачі даних у мережу. Таким чином, функціональна логіка збережена: кожен віртуальний пристрій у Packet Tracer виконує роль вузла збору та передачі даних до центрального серверу через локальну мережу. У реальному проєкті цю функцію виконує мікроконтролер з підключеними сенсорами.

У рамках IoT-системи житлового комплексу передбачено створення окремої локальної мережі (LAN) у кожному будинку. Кожна з них, у свою чергу, розділена на логічні підмережі для серверів, користувачів та пристроїв Інтернету речей.

VLAN-сегментацію не передбачено, її функції виконує маршрутизація трьох окремих підмереж.

3.1 Розрахунок адресного простору IoT-системи

У таблиці 3.1 наведено перелік підмереж локальної мережі кожного з чотирьох будинків та приблизну кількість пристроїв у кожному з них.

Таблиця 3.1 – Сегменти локальної мережі будинку та кількість пристроїв.

Назва сегменту	Орієнтовна кількість пристроїв	Призначення
IoT	25	Розумні пристрої, сенсори, контролери, підключені до шлюзу
Користувачі	5	Клієнтські пристрої: ноутбуки, смартфони, планшети
Сервери	1	Центральний сервер з IoT-сервісами (HTTP, MQTT, Home Assistant емуляція)

Для кожної з підмереж будинку визначено адресний простір, шлюз за замовчуванням та діапазон доступних IP-адрес. Адресацію обрано із зони 192.168.0.0/24, що дозволяє уникнути конфліктів у глобальній мережі. IP-адреси пристроям можуть надаватись автоматично через DHCP або вручну (для серверів та ключових вузлів).

В таблиці 3.2 наведена схема IP-адресації загальної мережі.

Таблиця 3.2 - Схема IP-адресації загальної мережі.

Назва підмережі	К-сть вузлів	Номер мережі	Маска мережі	Початкова IP	Кінцева IP
LAN1 Сервер	10	192.168.1.0	/24	192.168.1.1	192.168.1.254
LAN1 HomeGateway	20	192.168.20.0	/24	192.168.20.1	192.168.20.254
LAN1 Користувачі	50	192.168.3.0	/24	192.168.3.1	192.168.3.254
LAN2 Сервер	10	192.168.4.0	/24	192.168.4.1	192.168.4.254

LAN2 HomeGateway	20	192.168.50.0	/24	192.168.50.1	192.168.5.254
LAN2 Користувачі	50	192.168.6.0	/24	192.168.6.1	192.168.6.254
LAN3 Сервер	10	192.168.7.0	/24	192.168.7.1	192.168.7.254
LAN3 HomeGateway	20	192.168.80.0	/24	192.168.80.1	192.168.8.254
LAN3 Користувачі	50	192.168.9.0	/24	192.168.9.1	192.168.9.254
LAN4 Сервер	10	192.168.10.0	/24	192.168.10.1	192.168.10.254
LAN4 HomeGateway	20	192.168.110.0	/24	192.168.110.1	192.168.11.254
LAN4 Користувачі	50	192.168.12.0	/24	192.168.12.1	192.168.12.254
WAN1	2	10.0.1.0	/30	10.0.1.1	10.0.1.2
WAN2	2	10.0.2.0	/30	10.0.2.1	10.0.2.2
WAN3	2	10.0.3.0	/30	10.0.3.1	10.0.3.2
WAN4	2	10.0.4.0	/30	10.0.4.1	10.0.4.2
LAN5 Сервер	10	192.168.100.0	/24	192.168.100.1	192.168.100.254
LAN5 Користувачі	10	192.168.101.0	/24	192.168.101.1	192.168.101.254
LAN5 HomeGateway	2	209.165.200.0	/30	209.165.200.1	209.165.200.2

За технічними вимогами до IoT системи були призначенні мережні адреси пристроїв мережі (див. таблиця 3.3).

Таблиця 3.3 – Схема адресації пристроїв мережі.

Пристрій	Інтерфейс	IP-адреса	Маска	Шлюз
Router R1	GigabitEthernet0/0	192.168.1.1	255.255.255.0	
	GigabitEthernet1/0	192.168.2.1	255.255.255.0	
	GigabitEthernet2/0	192.168.3.1	255.255.255.0	
	FastEthernet3/0	10.0.1.2	255.255.255.252	10.0.1.1
ServerIoT1	FastEthernet0/0	192.168.1.2	255.255.255.0	192.168.1.1
HomeGateway1	Internet	192.168.2.10	255.255.255.0	192.168.2.1
PC1	FastEthernet0/0	192.168.3.10	255.255.255.0	192.168.3.1
PC2	FastEthernet0/0	192.168.3.11	255.255.255.0	192.168.3.1
Laptop1	FastEthernet0/0	192.168.3.12	255.255.255.0	192.168.3.1
Router R2	GigabitEthernet0/0	192.168.4.1	255.255.255.0	
	GigabitEthernet1/0	192.168.5.1	255.255.255.0	
	GigabitEthernet2/0	192.168.6.1	255.255.255.0	
	FastEthernet3/0	10.0.2.2	255.255.255.252	10.0.2.1
ServerIoT2	FastEthernet0/0	192.168.4.2	255.255.255.0	192.168.4.1
HomeGateway2	Internet	192.168.5.10	255.255.255.0	192.168.5.1
PC1_2	FastEthernet0/0	192.168.6.10	255.255.255.0	192.168.6.1

PC2_2	FastEthernet0/0	192.168.6.11	255.255.255.0	192.168.6.1
Laptop1_2	FastEthernet0/0	192.168.6.12	255.255.255.0	192.168.6.1
Router R3	GigabitEthernet0/0	192.168.7.1	255.255.255.0	
	GigabitEthernet1/0	192.168.8.1	255.255.255.0	
	GigabitEthernet2/0	192.168.9.1	255.255.255.0	
	FastEthernet3/0	10.0.3.2	255.255.255.252	10.0.3.1
ServerIoT3	FastEthernet0/0	192.168.7.2	255.255.255.0	192.168.7.1
HomeGateway3	Internet	192.168.8.10	255.255.255.0	192.168.8.1
PC1_3	FastEthernet0/0	192.168.9.10	255.255.255.0	192.168.9.1
PC2_3	FastEthernet0/0	192.168.9.11	255.255.255.0	192.168.9.1
Laptop1_3	FastEthernet0/0	192.168.9.12	255.255.255.0	192.168.9.1
Router R4	GigabitEthernet0/0	192.168.10.1	255.255.255.0	
	GigabitEthernet1/0	192.168.11.1	255.255.255.0	
	GigabitEthernet2/0	192.168.12.1	255.255.255.0	
	FastEthernet3/0	10.0.4.2	255.255.255.252	10.0.4.1
ServerIoT4	FastEthernet0/0	192.168.10.2	255.255.255.0	192.168.10.1
HomeGateway4	Internet	192.168.11.10	255.255.255.0	192.168.11.1
PC1_4	FastEthernet0/0	192.168.12.10	255.255.255.0	192.168.12.1
PC2_4	FastEthernet0/0	192.168.12.11	255.255.255.0	192.168.12.1
Laptop1_4	FastEthernet0/0	192.168.12.12	255.255.255.0	192.168.12.1
RouterBMS	GigabitEthernet0/0	10.0.1.1	255.255.255.252	10.0.1.1
	GigabitEthernet1/0	10.0.2.1	255.255.255.252	10.0.2.1
	GigabitEthernet2/0	10.0.3.1	255.255.255.252	10.0.3.1
	GigabitEthernet3/0	10.0.4.1	255.255.255.252	10.0.4.1
	GigabitEthernet4/0	192.168.100.1	255.255.255.252	192.168.100.1
	GigabitEthernet5/0	192.168.101.1	255.255.255.252	192.168.101.1
	GigabitEthernet6/0	209.165.200.4	255.255.255.0	209.165.200.1
ServerBMS	FastEthernet0/0	192.168.100.2	255.255.255.0	192.168.100.1
PC0	FastEthernet0/0	192.168.101.10	255.255.255.0	192.168.101.1
HomeGatewayBMS	FastEthernet0/0	192.168.101.11	255.255.255.0	192.168.101.1

Точки доступу для IoT пристроїв HomeGateway розгортають всередині власну LAN мережу. В таблиці 3.4 наведено схему IP-адресації мереж HomeGateway.

Таблиця 3.4 - Схема IP-адресації мереж HomeGateway.

Назва підмережі	К-сть вузлів	Номер мережі	Маска мережі	Початкова IP	Кінцева IP
LAN HomeGateway1	22	192.168.20.0	/24	192.168.20.100	192.168.20.121
LAN HomeGateway2	22	192.168.50.0	/24	192.168.50.100	192.168.50.121
LAN HomeGateway3	22	192.168.80.0	/24	192.168.80.100	192.168.80.121

LAN HomeGateway4	22	192.168.110.0	/24	192.168.110.100	192.168.110.121
LAN HomeGatewayBMS	6	192.168.25.0	/24	192.168.25.100	192.168.110.105

3.2 Розробка логічної та фізичної топології

У програмі Cisco Packet Tracer була розроблена топологічна схема, що складається із 4 підмереж, а саме будинків, які в свою чергу мають по 3 підмережі, а саме з серверної частини, локальної мережі користувачів і IoT пристроїв.

Також в мережу входять: підмережа загального серверу BMS та об'єднана підмережа контролю і спостереження. Схему з'єднання контролера MCU, що імітує ESP32 з датчиками представлено на рисунку 3.1

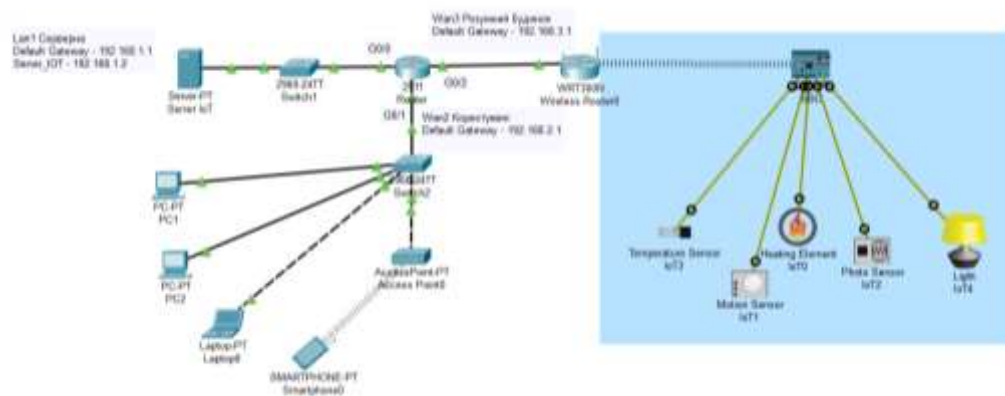


Рисунок 3.1 - З'єднання контролера MCU з датчиками.

Враховуючи обмеженість програмних компонентів Cisco Packet Tracer, реалізувати повноцінну логіку роботи системи за допомогою мікроконтроллера не вдалося. Тому було вирішено зімітувати логіку роботи зв'язки контроллер + датчик одним «розумним» пристроєм.

Топологію мережі першого будинку ЖК та топологію мережі його IoT пристроїв представлено на рисунках 3.2 – 3.3

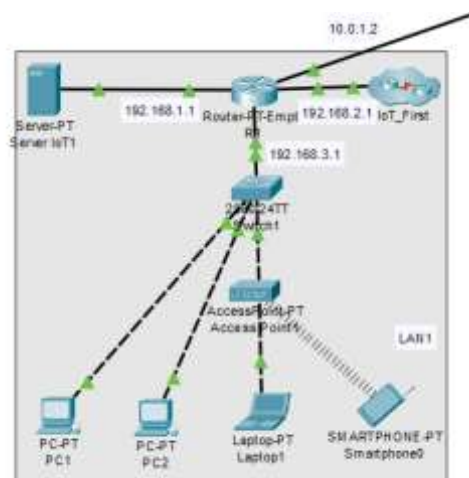


Рисунок 3.2 - Топологія мережі першого будинку.

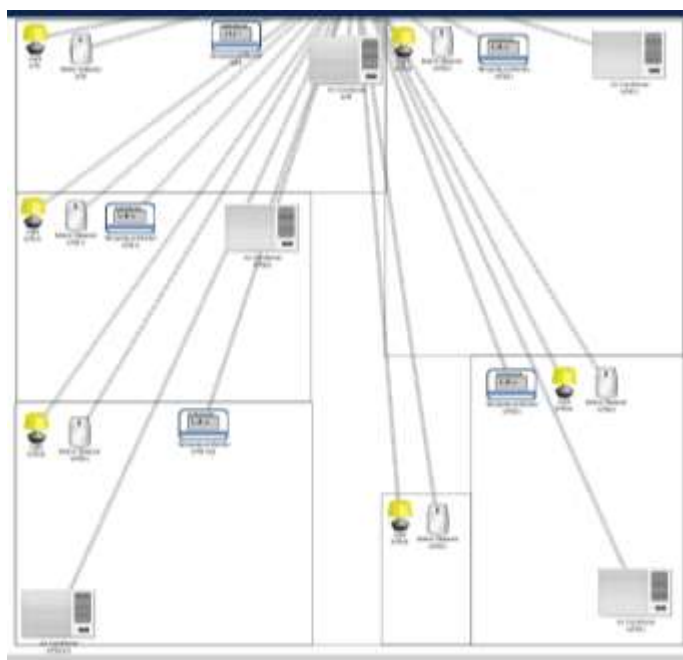


Рисунок 3.3 - Топологію мережі IoT пристроїв будинку.

Загальну топологію мережі зображено на рисунку 3.4

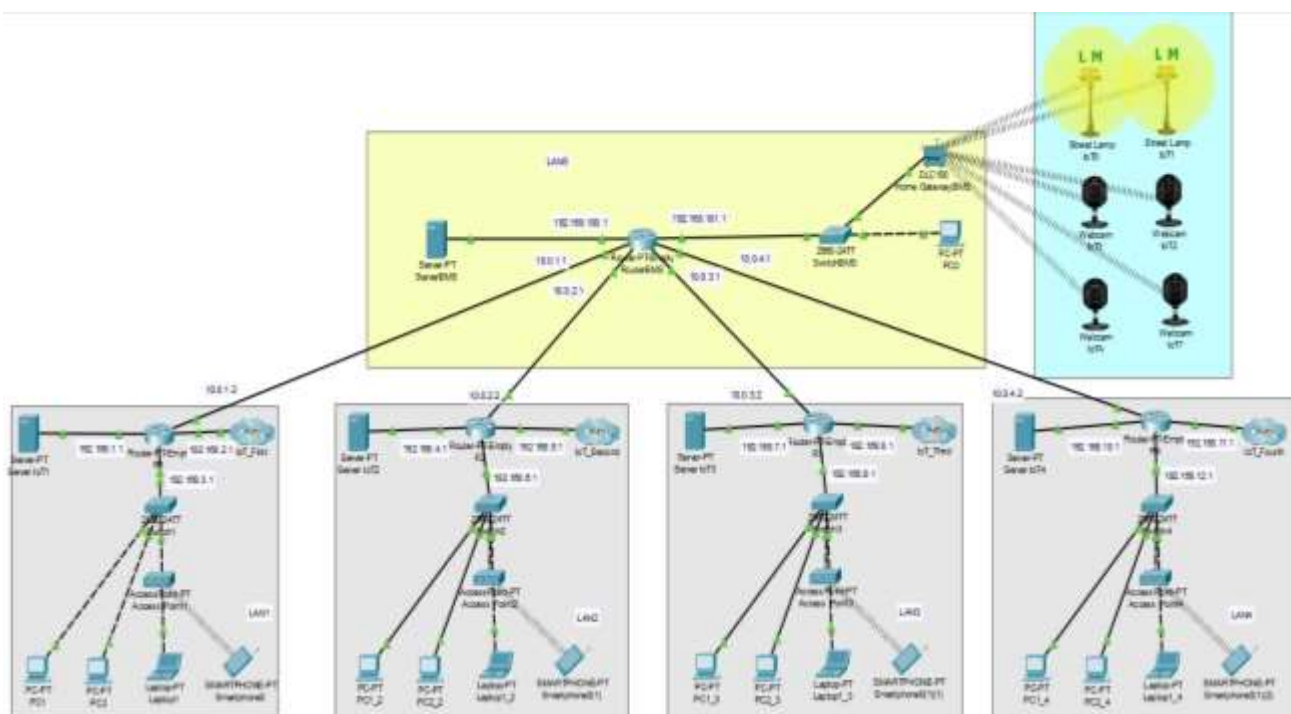


Рисунок 3.4 – Загальна топологія мережі.

3.3 Налаштування мережевих компонентів

На цьому етапі розробки системи здійснюється конфігурація ключових елементів мережевої інфраструктури, які забезпечують взаємодію всіх логічних сегментів корпоративної мережі IoT-системи, з урахуванням вимог до адресації, маршрутизації, безпеки та масштабованості.

Конфігураційні дії охоплюють базове IP-налаштування, організацію автоматичного присвоєння адрес через DHCP, реалізацію маршрутизації між підмережами, налаштування трансляції мережевих адрес (NAT), сегментацію мережі за допомогою VLAN, а також впровадження віддаленого доступу до серверної частини системи. Усі конфігурації виконуються з урахуванням моделі «Router-on-a-Stick» для підтримки міжмережевої маршрутизації та логічного поділу доменів.

3.3.1 Базове налаштування маршрутизації

У запропонованій структурі мережі кожного з будинків передбачено поділ загальної мережі на кілька ізольованих підмереж за функціональним призначенням: окрема підмережа для IoT-пристроїв, для клієнтських пристроїв користувачів та для серверного сегменту. Така логічна сегментація забезпечує базовий рівень ізоляції та підвищує безпеку системи. Для маршрутизації між підмережами використано класичну схему з багатопортовим маршрутизатором, де кожен фізичний або логічний інтерфейс маршрутизатора підключено до окремої підмережі.

На відміну від корпоративних систем з використанням VLAN, де взаємодія між віртуальними мережами реалізується через інкапсуляцію 802.1Q (Router-on-a-Stick), у даній реалізації кожна підмережа безпосередньо пов'язана з окремим інтерфейсом маршрутизатора, що спрощує налаштування та уможливорює наочну перевірку маршрутизації у середовищі Cisco Packet Tracer.

Віртуальний маршрутизатор в кожному будинку (R1, R2, R3, R4) має три активні інтерфейси, кожен з яких виконує роль шлюзу для відповідного сегменту мережі. У таблиці 3.5 наведено конфігураційні параметри інтерфейсів маршрутизатора для першого будинку.

Таблиця 3.5 - Конфігураційні параметри інтерфейсів маршрутизатора для першого будинку.

Сегмент мережі	Інтерфейс маршрутизатора	IP-адреса шлюзу	Маска підмережі
LAN IoT	G0/0	192.168.2.1	255.255.255.0
LAN Користувачі	G0/1	192.168.3.1	255.255.255.0
LAN Серверна зона	G0/2	192.168.1.1	255.255.255.0
WAN	F0/0	10.0.1.1	255.255.255.0

Кожен інтерфейс маршрутизатора було налаштовано на відповідну IP-адресу, що дозволяє здійснювати маршрутизацію між підмережами на мережевому рівні та між головним роутером мережі. Така схема, хоча й є менш гнучкою порівняно з

VLAN, дозволяє досягти еквівалентного рівня логічної ізоляції та значно спрощує візуалізацію роботи маршрутизації в навчальному середовищі.

На рисунку 3.5 можемо побачити приклад налаштування роутеру R1 в першому будинку.

```

Physical  Config  CLI  Attributes

Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#int g0/0
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# no shutdown

Router(config-if)#
Router(config-if)#int g1/0
Router(config-if)# ip address 192.168.2.1 255.255.255.0
Router(config-if)# no shutdown

Router(config-if)#
Router(config-if)#int g2/0
Router(config-if)# ip address 192.168.3.1 255.255.255.0
Router(config-if)# no shutdown

Router(config-if)#
Router(config-if)#int f3/0
Router(config-if)# ip address 10.0.1.2 255.255.255.252
Router(config-if)# no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
%LINK-5-CHANGED: Interface GigabitEthernet1/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changed state to up
%LINK-5-CHANGED: Interface GigabitEthernet2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet2/0, changed state to up
%LINK-5-CHANGED: Interface FastEthernet3/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet3/0, changed state to up
end
% Invalid input detected at '^' marker.

Router(config-if)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console
Writing memory
Building configuration...
[OK]
Router#

```

Рисунок 3.5 – Базове налаштування роутеру R1.

За таким принципом було налаштовані інші роутери в будинках мережі, згідно з таблицею IP-адресації мереж.

Інтерфейси головного роутера також було налаштовано згідно з таблицею IP-адресації мереж. Налаштування представлені на рисунку 3.6

Налаштування DHCP для головного роутеру представлено на рисунку 3.8

```

Physical Only Config Admin
IOS Command Line Interface

Router(config-router)# network 20.0.2.0 0.0.0.0 area 0
Router(config-router)# network 20.0.2.0 0.0.0.0 area 0
Router(config-router)# network 20.0.2.0 0.0.0.0 area 0
Router(config-router)#
Router(config-router)#exit
Router(config)#end
Router#show memory
4520-KIOSK_1: Configured from console by console
Building configuration...

[OK]
Router#
R1-0/2/0: 4096-KIOSK: Process 1, Src 2.2.2.1 on GigabitEthernet1/0 from 100000 to 1000,
Loading Done
R1-0/2/1: 4096-KIOSK: Process 1, Src 2.2.2.2 on GigabitEthernet1/0 from 100000 to 1000,
Loading Done
R1-0/2/2: 4096-KIOSK: Process 1, Src 2.2.2.3 on GigabitEthernet1/0 from 100000 to 1000,
Loading Done
R1-0/2/3: 4096-KIOSK: Process 1, Src 2.2.2.4 on GigabitEthernet1/0 from 100000 to 1000,
Loading Done

Poweroff 0
Enter configuration commands, one per line. Use Ctrl-Z to
Router(config)#ip dhcp excluded-address 192.168.101.1
Router(config)#ip dhcp pool 192.168.101.1
Router(dhcp-pool)#network 192.168.101.0 255.255.255.0
Router(dhcp-pool)#default-router 192.168.101.1
Router(dhcp-pool)#
Router(dhcp-pool)#exit
Router(config)#end
Router#
Router#
R101-KIOSK_1: Configured from console by console
Building configuration...

[OK]
Router#

```

Рисунок 3.8 – Налаштування DHCP роутеру RouterBMS.

Результат перевірки правильності налаштування DHCP представлено на рисунку 3.9

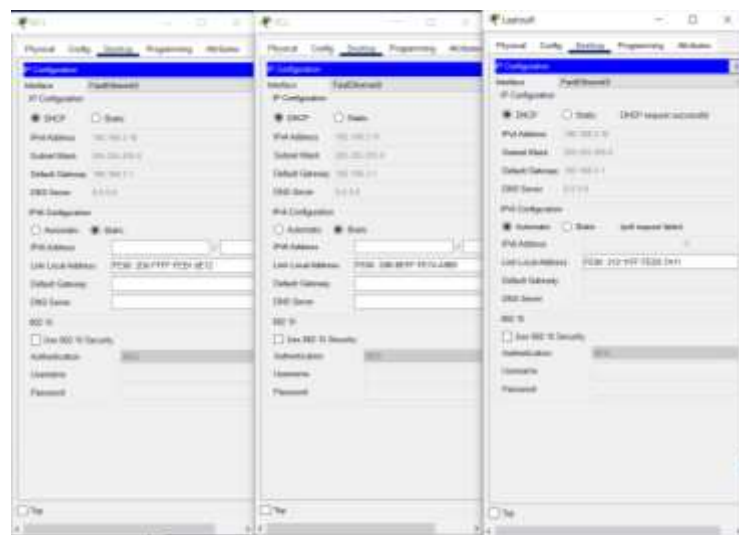


Рисунок 3.9 – Перевірка налаштування DHCP.

У маршрутизації існує два основних підходи: статичний і динамічний. Статична маршрутизація передбачає ручне внесення маршрутів

адміністратором, що забезпечує повний контроль, однак не дозволяє системі гнучко реагувати на зміни в мережевій структурі.

Динамічна маршрутизація, навпаки, використовує спеціальні протоколи (наприклад, RIP, OSPF, BGP), які автоматично будують таблиці маршрутів і адаптуються до змін. Такий підхід особливо ефективний у великих і розгалужених мережах, де потрібна масштабованість та автоматизація.

У роботі було обрано динамічну маршрутизацію з використанням протоколу OSPF (Open Shortest Path First), що забезпечує ефективне та швидке визначення оптимальних маршрутів. Серед основних переваг OSPF можна виділити:

- Високу швидкість збіжності;
- Підтримку мереж із масками змінної довжини (VLSM);
- Відсутність обмежень щодо кількості переходів;
- Раціональне використання пропускної здатності мережі;
- Визначення найкоротших маршрутів на основі метрики.

Нижче наведено рисунки 3.10-3.11 з налаштуваннями OSPF як на головному маршрутизаторі, так і на маршрутизаторі одного з будинків. На всіх інших маршрутизаторах було проведено аналогічні налаштування, з врахуванням IP-адресації.

З метою забезпечення контролю та моніторингу IoT-системи без необхідності використання зовнішніх хмарних платформ, було реалізовано механізм прямого IP-доступу до серверної частини мережі. Це дозволяє користувачеві керувати компонентами системи (зокрема розумними пристроями) через локальну мережу, або, при необхідності, через тунельований віддалений доступ (наприклад, VPN у подальшому розширенні).

У початковій моделі було передбачено пряме підключення плати MCU до сервера. Проте у середовищі Cisco Packet Tracer немає можливості реалізувати повноцінний двосторонній обмін даними між MCU та сервером з використанням призначених протоколів, що імітували б сценарії реальної взаємодії з сенсорами. У зв'язку з цим було прийнято архітектурне рішення реалізувати умовне розміщення MCU в окремому житловому сегменті, підключеному до Wi-Fi маршрутизатора, який виступає у ролі локального шлюзу (Home Gateway).

Таким чином, Wi-Fi маршрутизатор Home Gateway виконує функцію точки доступу до IoT-сегменту. До нього підключені «розумні» пристрої, які емулюють роботу ESP32 з пасивними датчиками.

Серверна частина (ServerIoT1) налаштована на прийом з'єднань від пристроїв у мережі 192.168.20.0/24, що підключені через Home Gateway. Передача даних здійснюється через IP-протокол, що дозволяє моделювати механізм керування побутовими приладами на основі вхідних параметрів з сенсорів.

На рисунку 3.13 можемо бачити налаштування SSID Home Gateway та типу Wi-Fi з'єднання, а також налаштування підмережі IoT.

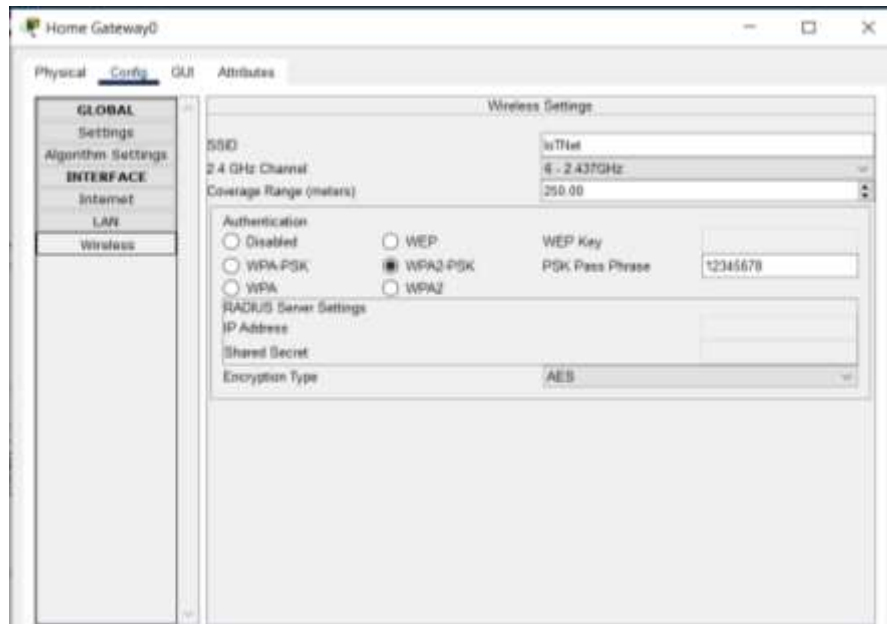


Рисунок 3.13 – Налаштування HomeGateway.

Розумним пристроям було надано доступ до серверу за ір-адресою. ІР-адресація розумних пристроїв відбувається за допомогою DHCP(рис. 3.14)

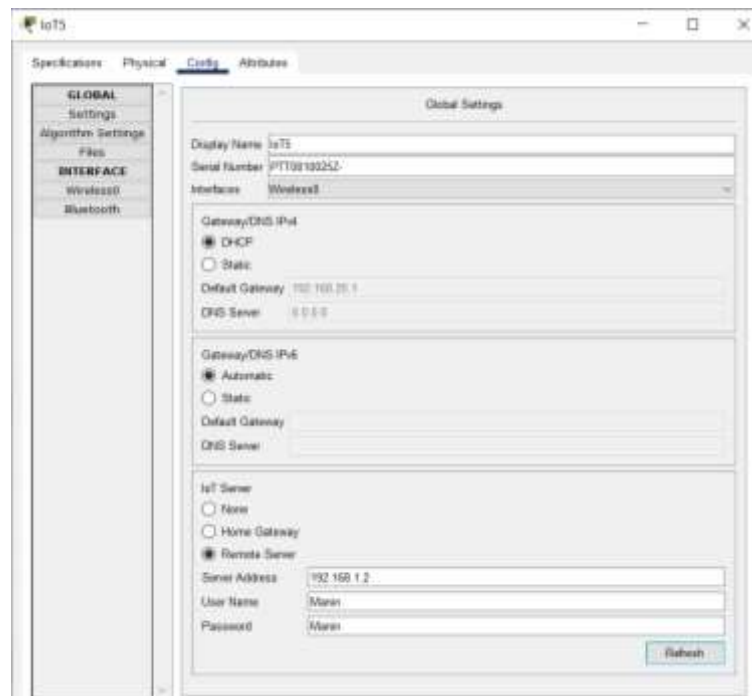


Рисунок 3.14 – Налаштування доступу IoT пристроїв.

Наступним етапом було налаштування серверної частини IoT-системи. На пристрої ServerIoT1 змодельовано роботу системного вузла, який приймає та опрацьовує інформацію від розумних пристроїв, а також надає інтерфейс для адміністрування системи користувачем.

Замість використання окремого мікроконтролера, роль керуючого елемента було покладено на локальний сервер (ServerIoT1), який виконує функції системного ядра — обробляє вхідні дані, забезпечує зберігання змінних, та надає користувачу інтерфейс для віддаленого керування розумними пристроями.

У середовищі Cisco Packet Tracer на Server1 було активовано одразу два сервіси(рис.):

- IoT Service — для створення змінних, до яких можуть підключатись датчики та пристрої,
- HTTP Service — для доступу до веб-інтерфейсу керування.

Користувач може під'єднатися до сервера через веб-браузер з будь-якого пристрою у внутрішній мережі, ввівши IP-адресу сервера(рис.3.15)



Рисунок 3.15 – Веб-інтерфейс локального IoT серверу.

Перед отриманням доступу до панелі керування сервер запитує облікові дані користувача. Це дозволяє змодельювати базовий механізм автентифікації, аналогічний до того, який реалізується у реальних розумних системах (наприклад, Home Assistant або інтерфейсах контролерів ESP32).

Користувач має пройти реєстрацію у вкладці на самому сервері. Там задається ім'я користувача, пароль і рівень доступу (наприклад, адміністратор або звичайний користувач).

Після реєстрації, на сервері з'являється обліковий запис користувача. На рисунку 3.16 наведено приклад облікового запису користувача серверу ServerBMS.

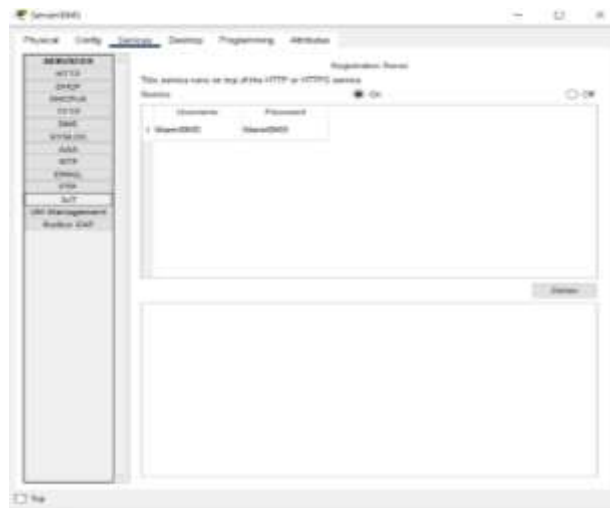


Рисунок 3.16 – Обліковий запис користувача.

Після авторизації відкривається вбудована HTML-сторінка, що надає доступ до панелі змінних, створених у сервісі IoT. Ці змінні можуть бути прив'язані до елементів керування (наприклад, вимикачів, температурних зон тощо), що дозволяє моделювати реакцію системи на дії користувача(рис.3.17).



Рисунок 3.17 – Перелік IoT пристроїв, зв'язаних з ServerIoT1.

Такі ж самі дії було проведено для кожного будинку. Також було налаштовано HomeGatewayBMS, що відповідає за освітлення та доступ до камер видеоспостереження, та ServerBMS, що відповідає за обробку даних, які надходять від IoT пристроїв. На рисунку 3.18 представлено Відображення IoT пристроїв на стороні ServerBMS.

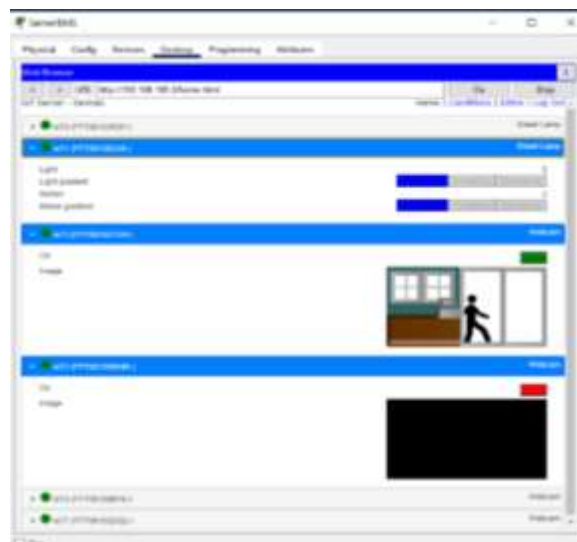


Рисунок 3.18 – Перелік IoT пристроїв, зв'язаних з ServerBMS.

3.3.5 Налаштування VLAN

У типовій корпоративній мережі сегментація трафіку реалізується за допомогою віртуальних локальних мереж (VLAN), що дозволяє створювати логічно ізольовані домени, забезпечуючи контроль доступу, підвищену безпеку та зменшення широкомовного трафіку. Однак у межах даного проєкту, який реалізовано у віртуальному середовищі Cisco Packet Tracer, від впровадження VLAN було свідомо відмовлено з урахуванням технічних обмежень симулятора та особливостей топології IoT-системи.

Замість VLAN було застосовано альтернативний підхід до логічної сегментації мережі шляхом виділення окремих підмереж IP-адресації для кожного функціонального сегменту. Мережа IoT-пристроїв, користувацький сегмент та серверна зона функціонують у межах власних підмереж, з'єднаних через маршрутизатор, який виконує міжмережеву маршрутизацію. Таким чином досягається розмежування мережевого трафіку без прямого використання VLAN.

Подібна реалізація не суперечить загальним принципам проектування локальних мереж, оскільки логічне розмежування трафіку може бути забезпечено не лише на каналному рівні (через VLAN), але й на мережевому рівні – через чітке структурування адресного простору та відповідну маршрутизацію.

Варто зазначити, що в умовах фізичного розгортання системи, зокрема при побудові мережі з використанням керованого комутаційного обладнання, впровадження VLAN може бути реалізоване на подальших етапах розгортання, не потребуючи значних змін у логіці системи. Отже, обраний підхід до сегментації забезпечує необхідну ізоляцію між вузлами системи та відповідає базовим принципам побудови безпечної інфраструктури Інтернету речей.

3.4 Перевірка працездатності комп'ютерної мережі

Для перевірки коректної роботи налаштованої системи було здійснено тестову відправку пакетів між головним маршрутизатором та маршрутизаторами в будинках з метою перевірки функціонування протоколу OSPF. Також виконано обмін пакетами між комп'ютерами, що знаходяться в різних внутрішніх підмережах, для перевірки роботи статичного NAT та маршрутизації в цілому. Результати перевірки наведено на рисунку 3.19.

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC1	Server IoT1	ICMP		0.000	N	0	(edit)	
	Successful	PC2	ServerBMS	ICMP		0.437	N	1	(edit)	
	Successful	PC1_2	Server IoT2	ICMP		0.619	N	2	(edit)	
	Successful	PC2_2	ServerBMS	ICMP		0.685	N	3	(edit)	

Рисунок 3.19 - Результат перевірки OSPF, NAT та маршрутизації.

За умовою, співробітник відділу безпеки повинен мати доступ до серверу ServerBMS, куди всі IoT пристрої ЖК надсилають свої дані, насамперед камери та світильники. Тому було протестовано доступ комп'ютера PC0 до сервера командою ping. Результати тестування представлені на рисунку 3.20.

```

Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.100.2

Pinging 192.168.100.2 with 32 bytes of data:

Reply from 192.168.100.2: bytes=32 time<1ms TTL=127
Reply from 192.168.100.2: bytes=32 time<1ms TTL=127
Reply from 192.168.100.2: bytes=32 time<1ms TTL=127
Reply from 192.168.100.2: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.100.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>

```

Рисунок 3.20 – Результати тестування доступу PC0 до ServerBMS.

3.5 Висновки до розділу

У цьому розділі було реалізовано повноцінну архітектуру корпоративної мережі IoT-системи для житлового комплексу. Було проведено розрахунок IP-адресного простору з логічною сегментацією за функціональним призначенням (сервери, користувачі, IoT), а також побудовано фізичну та логічну топології мережі.

За допомогою динамічного маршрутизатора OSPF забезпечено автоматизовану маршрутизацію між усіма підмережами, включаючи точки доступу в кожному будинку. Завдяки впровадженню NAT на головному маршрутизаторі, всі вузли мережі отримали змогу виходу до Інтернету через одну публічну IP-адресу. Результати перевірки підтвердили коректність функціонування маршрутизації, NAT, DHCP та мережевої взаємодії між усіма компонентами системи. Реалізована мережа є масштабованою, логічно ізольованою та готовою до подальшого впровадження засобів безпеки та управління.

4. РОЗРОБКА ПРОГРАМНОГО КОМПОНЕНТУ СИСТЕМИ

4.1 Вибір платформи та обґрунтування архітектури програмного компонента

Розробка сучасних систем інтернету речей (IoT), особливо тих, що орієнтовані на автоматизацію житлових приміщень та контроль енергоефективності, вимагає ретельного підходу до вибору апаратної та програмної платформи. Від коректності цього вибору залежить не лише функціональна повнота рішення, але й його масштабованість, надійність, безпека та легкість подальшої підтримки.

У контексті дипломної роботи, де поєднуються елементи IoT житлового приміщення з комп'ютерною системою контролю енергоефективності, особливого значення набуває можливість інтеграції різнорідних пристроїв, ефективна обробка даних з сенсорів та реалізація складної логіки автоматизації.

4.1.1 Аналіз альтернативних архітектурних підходів

При аналізі потенційних платформ для центрального вузла управління розумним будинком, були розглянуті кілька ключових архітектурних підходів, кожен з яких має свої переваги та недоліки.

Перший підхід полягає у використанні спеціалізованих комерційних хмарних систем, таких як Google Home, Amazon Alexa або Apple HomeKit. Ці платформи пропонують високий рівень інтеграції з відповідними екосистемами, простоту налаштування та зручний користувацький інтерфейс.

Вони ідеально підходять для масового споживача, який цінує простоту використання та не вимагає глибокої кастомізації. Однак, їх істотним недоліком є обмеженість у виборі підтримуваних пристроїв, що може створювати проблеми при використанні нестандартних або саморобних IoT-елементів.

Крім того, відсутність гнучкості у реалізації кастомних сценаріїв автоматизації обмежує можливості для складних, специфічних завдань, що є типовим для науково-дослідних та дипломних робіт. Залежність від хмарних сервісів також створює ризики при втраті інтернет-з'єднання, що може призвести до повного паралічу системи, та викликає суттєві питання щодо конфіденційності даних користувачів. З точки зору дослідницької роботи, комерційні рішення часто не дозволяють глибокого втручання у внутрішню логіку системи та її оптимізацію, що є неприпустимим.

Другий підхід передбачає розробку повністю власного програмного забезпечення з нуля. Такий шлях забезпечує максимальну гнучкість та повний контроль над кожним аспектом системи, від низькорівневої взаємодії з апаратним забезпеченням до розробки користувацького інтерфейсу та хмарних сервісів.

Це дозволяє реалізувати будь-які, навіть найскладніші та нетипові функціональні вимоги. Проте, такий шлях є надзвичайно ресурсомістким, вимагає значних часових витрат на розробку, тестування та налагодження, а також наявності великого обсягу знань у широкому спектрі галузей, включаючи програмування мікроконтролерів, мережеві технології, проектування баз даних, розробку API та веб-розробку. Для обсягу та часових рамок дипломної роботи створення повноцінної, надійної та безпечної системи з нуля є непрактичним та перевищує її мету.

Враховуючи вищезазначені обмеження та вимоги до гнучкості, масштабованості, відкритості та можливості глибокої кастомізації, оптимальним рішенням для реалізації поставлених завдань було обрано третій підхід – використання відкритої, гнучкої платформи з широкими можливостями інтеграції та підтримкою великої кількості пристроїв.

4.1.2 Обґрунтування вибору програмної платформи Home Assistant

Серед відкритих платформ автоматизації розумного будинку, які були розглянуті для даної роботи, виділяються OpenHAB, Domoticz та Home Assistant. Після детального аналізу їхніх функціональних можливостей, архітектурних особливостей, рівня підтримки спільноти, якості документації та зручності розгортання, вибір був зупинений на Home Assistant. Ця платформа є відкритим програмним забезпеченням для автоматизації розумного будинку, розробленим на мові програмування Python. Її архітектура спеціально спроектована для забезпечення високого рівня інтеграції з величезною кількістю існуючих IoT-пристроїв та сервісів, незалежно від їхнього виробника та протоколів зв'язку.

Ключові переваги Home Assistant, що зумовили його вибір для реалізації нашої роботи, включають:

Широка сумісність та інтеграційні можливості. Home Assistant підтримує понад 2000 офіційних інтеграцій, що дозволяє підключати практично будь-які сучасні IoT-пристрої, від елементарних датчиків та освітлювальних приладів до складних систем відеоспостереження, контролю доступу та клімат-контролю.

Гнучкість у реалізації автоматизації. Платформа надає потужний інструментарій для створення складних та багатоетапних сценаріїв автоматизації (так званих "автоматизацій", "скриптів" та "сцен") на основі даних, отриманих з сенсорів, а також інших умов та тригерів. Це дозволяє ефективно реалізувати комплексну логіку контролю енергоефективності, що є однією з центральних задач даної дипломної роботи.

Локальний контроль та високий рівень конфіденційності. Усі процеси обробки даних та управління пристроями здійснюються локально на сервері Home Assistant. Це значно підвищує безпеку системи та захищеність персональних даних користувача, оскільки інформація не передається на сторонні хмарні сервіси без явного дозволу. Відсутність постійної залежності від зовнішніх хмарних рішень

також забезпечує стабільність та надійність роботи системи, мінімізує затримки у виконанні команд та автоматизацій.

Активна спільнота та всебічна документація. Навколо Home Assistant сформувалася велика та надзвичайно активна спільнота користувачів та розробників. Це забезпечує широку підтримку, доступ до численних прикладів використання, готових плагінів, розширень та рішень типових проблем. Детальна та регулярно оновлювана офіційна документація спрощує процес налаштування системи, інтеграції нових пристроїв та вирішення потенційних несправностей, що є значною перевагою для освітніх та дослідницьких проектів.

4.1.3 Обґрунтування вибору апаратної платформи Raspberry Pi

Центральним компонентом апаратної платформи для розгортання Home Assistant та функціонування всієї системи автоматизації було обрано одноплатний комп'ютер Raspberry Pi. Цей вибір обумовлений рядом ключових факторів, які роблять його ідеальним рішенням для прототипування, розгортання та тривалої експлуатації систем розумного будинку, а саме:

Компактність та низьке енергоспоживання. Raspberry Pi є надзвичайно компактним пристроєм, що дозволяє легко інтегрувати його у будь-яке житлове середовище без значної потреби у просторі. Його низьке енергоспоживання є важливим фактором для цілодобової роботи системи, забезпечуючи економічність експлуатації.

Достатня продуктивність для поставлених завдань. Сучасні моделі Raspberry Pi, зокрема Raspberry Pi 4 Model B, володіють достатньою обчислювальною потужністю та обсягом оперативної пам'яті для стабільної та ефективної роботи Home Assistant.

Широкий спектр комунікаційних інтерфейсів. Наявність вбудованих модулів Wi-Fi та Bluetooth, порту Ethernet, декількох USB-портів, а також

програмованих GPIO-роз'ємів забезпечує високу гнучкість у підключенні різноманітних периферійних пристроїв. Це дозволяє легко встановлювати зв'язок з IoT-сенсорами та контролерами, розробленими в рамках розділу 2 даної роботи (на базі ESP32).

Популярність, потужна спільнота та розвинена екосистема. Як і Home Assistant, Raspberry Pi має величезну спільноту користувачів, активну підтримку розробників, багату документацію та велику кількість готових рішень, проектів та інструкцій. Це значно спрощує процес розробки, налагодження та усунення несправностей, оскільки більшість типових питань вже має готові відповіді або обговорювалася на форумах.

Гнучкість операційної системи та легкість розгортання. Raspberry Pi підтримує різні дистрибутиви Linux, що надає гнучкість у налаштуванні операційного середовища та встановленні додаткового програмного забезпечення. Існує також спеціалізований дистрибутив Home Assistant OS, який забезпечує максимально просту та швидку інсталяцію Home Assistant на Raspberry Pi, а також автоматичне оновлення та управління системою.

4.1.4 Загальна архітектура програмного компонента

Загальна архітектура програмного компонента системи контролю енергоефективності житлового приміщення реалізована за централізованою моделлю з розподіленими елементами Інтернету речей (IoT), яка забезпечує гнучкість, масштабованість та високий рівень локального контролю. Центральним вузлом та інтегруючою платформою системи є Home Assistant, розгорнутий на одноплатковому комп'ютері Raspberry Pi 4B. Детальна архітектура представлена на Рисунку 4.1.

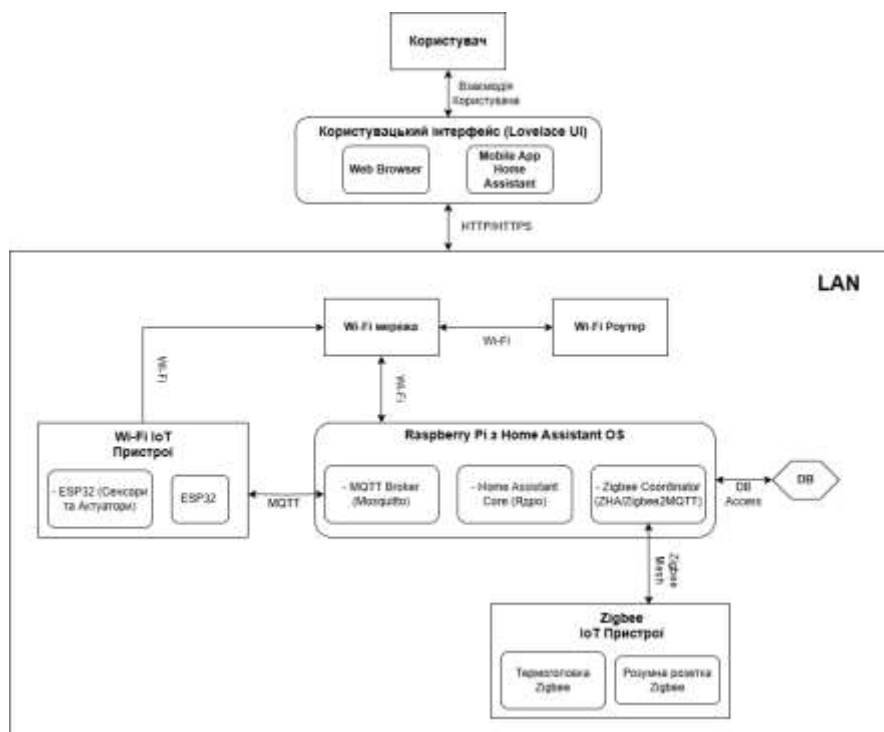


Рисунок 4.1. - Архітектура програмного компонента системи контролю енергоефективності житлового приміщення.

Розглянемо основні компоненти та їхні взаємозв'язки.

Центральний контролер – Raspberry Pi 4B з Home Assistant. Raspberry Pi 4B є апаратною основою системи, забезпечуючи достатню обчислювальну потужність для функціонування Home Assistant та підтримки всіх необхідних мережевих протоколів.

Home Assistant Core – це ядро програмного забезпечення, що відповідає за логіку автоматизації, обробку подій, управління станами пристроїв та взаємодію з користувачем.

MQTT Broker (Mosquitto) – інтегрований сервіс, що слугує центральним вузлом обміну повідомленнями за протоколом MQTT. Він забезпечує надійну та ефективну комунікацію між Home Assistant Core та Wi-Fi IoT пристроями.

Zigbee Coordinator (ZHA/Zigbee2MQTT) – програмний модуль, що взаємодіє з фізичним Zigbee-адаптером (підключеним до Raspberry Pi), формуючи Zigbee-

мережу. Він дозволяє Home Assistant керувати Zigbee IoT пристроями та отримувати від них дані.

Database (SQLite/MariaDB) – база даних, що використовується Home Assistant для зберігання історії станів пристроїв, логів, даних сенсорів та конфігураційної інформації. За замовчуванням це SQLite, але може бути налаштована на MariaDB для підвищення продуктивності та надійності. Взаємодія з базою даних відбувається через спеціалізовані DB Access інтерфейси.

Wi-Fi Router – основний компонент мережевої інфраструктури, що створює Wi-Fi Network та забезпечує бездротове підключення для всіх Wi-Fi пристроїв у локальній мережі. Raspberry Pi та всі Wi-Fi IoT пристрої підключаються до Wi-Fi Router, використовуючи Wi-Fi як фізичний рівень зв'язку. Роутер також виступає в ролі шлюзу для доступу до зовнішніх мереж.

Wi-Fi IoT Devices (ESP32) – це розроблені елементи IoT, які базуються на мікроконтролерах ESP32. Ці пристрої оснащені різноманітними сенсорами для збору даних (наприклад, температура, вологість, рух) та виконавчими пристроями (актуаторами) для керування (наприклад, освітлення, реле). Зв'язок з MQTT Broker на Raspberry Pi здійснюється через протокол MQTT, який працює поверх Wi-Fi з'єднання, забезпечуючи двонаправлений обмін даними.

Zigbee IoT Devices – до цієї категорії належать комерційні або власні пристрої, що використовують протокол Zigbee, такі як термоголовки для системи опалення, розумні розетки та інші датчики. Ці пристрої формують окрему бездротову мережу, керовану Zigbee Coordinator на Raspberry Pi, що дозволяє їм ефективно взаємодіяти з Home Assistant.

User (Власник Житла) – кінцевий користувач системи, який взаємодіє з нею для моніторингу, керування та налаштування автоматизації.

User Interface (Lovelace UI) – це основний інтерфейс Home Assistant, доступний через веб-браузер або спеціалізований мобільний додаток Home Assistant.

Користувач взаємодіє з цим інтерфейсом (Human Interaction), який, у свою чергу, обмінюється даними з Home Assistant Core за протоколами HTTP/HTTPS. Зовнішні сервіси (External Services / Cloud).

4.2 Вибір та обґрунтування технологій і протоколів зв'язку

Ефективність, надійність та масштабованість комп'ютерної системи контролю енергоефективності житлового приміщення значною мірою залежать від обґрунтованого вибору технологій та протоколів зв'язку. Для забезпечення безшовних комунікацій між різнорідними IoT-пристроями, централізованого управління та інтеграції з користувацьким інтерфейсом було обрано комбінацію провідних бездротових технологій та протоколів прикладного рівня, що оптимально відповідають вимогам та цілям даної кваліфікаційної роботи.

Вибір технологій бездротової передачі даних базувався на потребах у покритті, енергоефективності та пропускній здатності для різних типів пристроїв.

Технологія Wi-Fi, що відповідає стандарту IEEE 802.11, була обрана як основний засіб бездротового зв'язку для центрального контролера (Raspberry Pi) та тих IoT-пристроїв, що мають постійне джерело живлення і потребують відносно високої пропускну здатності. Основне обґрунтування такого вибору полягає у повсюдному поширенні Wi-Fi-інфраструктури у побутових умовах, що значно спрощує розгортання нових пристроїв та мінімізує додаткові витрати на мережеве обладнання.

Кожен пристрій, підключений до Wi-Fi мережі, отримує унікальну IP-адресу, що дозволяє Home Assistant безпосередньо взаємодіяти з ними у локальній мережі, використовуючи стандартні мережеві протоколи.

Слід зазначити, що Wi-Fi має вище енергоспоживання порівняно з протоколами низького енергоспоживання, що обмежує його застосування в пристроях з автономним живленням від батарей.

Для інтеграції пристроїв, критично залежних від низького енергоспоживання та необхідності формування масштабованої та стійкої до перешкод мережі, було обрано технологію Zigbee. Це стосується, зокрема, термоголовок системи опалення, розумних розеток.

Основною перевагою Zigbee є його низьке енергоспоживання, що дозволяє пристроям працювати від батарей протягом тривалих періодів (місяці або роки), забезпечуючи автономність багатьох датчиків та виконавчих пристроїв. Однією з найважливіших особливостей Zigbee є mesh-топологія мережі. Це означає, що мережа є самоорганізованою та самовідновлюваною, а пристрої, що постійно живляться від електромережі (наприклад, розумні розетки), можуть виступати маршрутизаторами. Надійність передачі даних у Zigbee посилюється завдяки використанню частотно-імпульсної модуляції та механізмів підтвердження доставки, що підвищує стійкість до радіоперешкод у діапазоні 2.4 ГГц.

Крім того, інтеграція Zigbee-координатора безпосередньо з Home Assistant забезпечує повний локальний контроль над мережею Zigbee, усуваючи залежність від зовнішніх хмарних сервісів.

Zigbee-мережа складається з трьох типів пристроїв, що виконують різні ролі для забезпечення її функціонування. В таблиці 4.1 вказані ролі різних пристроїв у Zigbee-мережі.

Таблиця 4.1 - Ролі різних пристроїв у Zigbee-мережі.

Роль пристрою	Опис	Приклади використання в системі
Координатор	Єдиний пристрій у мережі, що відповідає за її створення (формування), управління, маршрутизацію та зберігання інформації про мережу.	USB-адаптер Zigbee, підключений до Raspberry Pi, що інтегрується з Home Assistant.
Маршрутизатор	Передає дані між іншими пристроями, розширюючи зону покриття мережі.	Розумні розетки Zigbee, розумні лампи, деякі датчики з постійним

	Завжди підключений до постійного джерела живлення, оскільки не може "спати".	живленням, що розташовані для покращення покриття.
Кінцевий пристрій	Має низьке енергоспоживання, більшу частину часу знаходиться у сплячому режимі. Не може ретранслювати сигнали. Ідеальний для живлення від батарей.	Термоголовки, датчики руху/відкриття/вологості, кнопки з батарейним живленням, що потребують тривалої автономної роботи.

Для обміну даними між компонентами системи та взаємодії з користувачем було обрано протоколи, що забезпечують ефективність, надійність та безпеку.

MQTT є ключовим протоколом прикладного рівня, обраним для асинхронної взаємодії Home Assistant з Wi-Fi IoT-пристроями. Його легковажність та архітектура "публікація/підписка" роблять його оптимальним рішенням для середовищ IoT, де ресурси можуть бути обмеженими, а мережеві умови – нестабільними.

Це забезпечує високу масштабованість, оскільки відправники (видавці) та отримувачі (підписники) повідомлень не потребують прямої інформації один про одного, що спрощує додавання нових пристроїв та функцій без внесення змін до існуючої логіки.

MQTT підтримує три рівні якості обслуговування (Quality of Service, QoS), що дозволяє адаптувати надійність доставки повідомлень до вимог конкретного сценарію:

- QoS 0 (At most once): Повідомлення надсилається без підтвердження доставки, що підходить для періодичних показників датчиків, де втрата окремих пакетів не є критичною (наприклад, показники температури, що оновлюються з високою частотою).
- QoS 1 (At least once): Повідомлення доставляється як мінімум один раз. Брокер підтверджує отримання, проте можливі дублікати. Цей рівень використовується

для подій, що вимагають гарантованої доставки, але дублювання яких не спричинить негативних наслідків (наприклад, зміна стану пристрою).

- QoS 2 (Exactly once): Забезпечує найвищий рівень надійності, гарантуючи доставку повідомлення рівно один раз. Хоча цей рівень вимагає більшої кількості мережових обмінів, він є незамінним для критично важливих команд, де втрата або дублювання неприпустимі (наприклад, команди аварійного вимкнення обладнання).

Широка підтримка MQTT, включаючи інтеграцію брокера Mosquitto безпосередньо в Home Assistant та наявність клієнтських бібліотек для мікроконтролерів (зокрема ESP32), забезпечує безперешкодну та ефективну взаємодію між компонентами системи.

Протоколи HTTP (Hypertext Transfer Protocol) та його захищений варіант HTTPS (HTTP Secure) є фундаментальними для взаємодії користувача з системою Home Assistant.

Основним призначенням HTTP/HTTPS у даній системі є забезпечення доступу до користувацького інтерфейсу Lovelace UI. Це дозволяє користувачам отримувати доступ до системи, переглядати дані сенсорів, керувати пристроями та налаштовувати автоматизацію через будь-який стандартний веб-браузер або спеціалізований мобільний додаток Home Assistant. Широка доступність та універсальність HTTP/HTTPS на всіх платформах (десктоп, мобільні пристрої) забезпечує високу зручність та гнучкість у керуванні системою.

Обрані технології та протоколи зв'язку працюють у синергії, формуючи надійну, ефективну та адаптивну архітектуру, що задовольняє вимоги до сучасної системи контролю енергоефективності житлового приміщення.

Wi-Fi виступає як базовий фізичний рівень, що забезпечує мережеве підключення для центрального контролера (Raspberry Pi) та IoT-пристроїв на базі

ESP32 до локальної мережі. Це дозволяє здійснювати високошвидкісний обмін даними та, за потреби, доступ до глобальної мережі Інтернет.

MQTT надбудований над Wi-Fi, виконуючи роль легкої та масштабованої шини повідомлень. Він забезпечує асинхронну взаємодію між Wi-Fi-пристроями та центральним контролером, оптимально використовуючи мережеві ресурси та підтримуючи надійну передачу даних навіть в умовах обмеженої пропускної здатності.

Zigbee створює окрему, енергоефективну mesh-мережу для певних категорій пристроїв, що критично залежать від тривалого терміну служби батарей. Ця мережа інтегрується з Home Assistant через спеціалізований координатор, забезпечуючи надійний зв'язок та розширене покриття.

HTTP/HTTPS надає основний механізм для взаємодії користувача з системою. Забезпечуючи доступ до веб-інтерфейсу, ці протоколи дозволяють зручно та безпечно керувати системою з будь-якого пристрою, що підтримує веб-браузер.

Така багатопроTOCOLна архітектура дозволяє оптимально використовувати сильні сторони кожної технології, мінімізуючи їхні потенційні недоліки. Це забезпечує високу продуктивність для Wi-Fi-пристроїв, низьке енергоспоживання для Zigbee-пристроїв та зручний, захищений інтерфейс для кінцевого користувача, що в сукупності відповідає всім функціональним та нефункціональним вимогам проекту.

4.3 Розробка архітектури баз даних

Для потреб даної системи було обрано реляційну базу даних, що дозволяє структуровано зберігати інформацію про об'єкти системи та їхню поведінку у часі. Home Assistant за замовчуванням використовує SQLite як вбудовану систему керування базами даних, і саме цей варіант був обраний для даного проекту. Такий

вибір обґрунтовується декількома ключовими факторами, що роблять SQLite оптимальним рішенням для невеликих та середніх інсталяцій, особливо на одноплаткових комп'ютерах з обмеженими ресурсами.

Насамперед, SQLite вирізняється простотою розгортання та відсутністю потреби у складному адмініструванні. Оскільки база даних зберігається як єдиний файл на файловій системі Raspberry Pi, це значно спрощує початкове налаштування та подальше резервне копіювання даних. Крім того, мінімальні системні вимоги SQLite до обчислювальних ресурсів та оперативної пам'яті ідеально відповідають можливостям Raspberry Pi.

Використання вбудованої бази даних дозволяє спростити загальну архітектуру системи та зменшити накладні витрати на її обслуговування.

База даних Home Assistant призначена для збереження історичних даних, які є фундаментом для візуалізації трендів енергоспоживання, аналізу показників датчиків, виявлення аномалій та підтримки функціоналу автоматизації, що базується на минулих станах об'єктів. Основними категоріями інформації, що постійно записується до сховища даних, є історія станів пристроїв, різноманітні події системи та службові логи.

Кожен запис про стан сутності (entity) включає її унікальний ідентифікатор (entity_id), поточний стан (state), асоційовані атрибути (attributes) у форматі JSON, а також точні мітки часу останньої зміни стану (last_changed) та останнього оновлення запису (last_updated). Події, що відбуваються в системі, такі як спрацювання автоматизації або виявлення руху, також фіксуються із зазначенням типу події (event_type) та додаткових даних (event_data). Загальна структура бази даних Home Assistant оптимізована для ефективного запису та вилучення великих обсягів послідовних даних, що відображають динаміку змін в IoT-середовищі.

Структура та призначення ключових полів бази даних Home Assistant наведено у Таблиці 4.2

Таблиця 4.2 – Структура та призначення ключових полів бази даних Home Assistant.

Поле	Тип даних	Опис	Приклад значення
entity_id	VARCHAR	Унікальний ідентифікатор сутності (пристрою або датчика).	sensor.room_temperature
State	VARCHAR	Поточний стан сутності (числове, текстове, булеве значення).	22.5, on, open
attributes	JSON	Додаткові атрибути, пов'язані зі станом сутності (наприклад, одиниці виміру, назва виробника).	{"unit_of_measurement": "°C", "device_class": "temperature"}
last_changed	TIMESTAMP	Час останньої зміни стану сутності.	2025-05-27 10:30:00.123456
last_updated	TIMESTAMP	Час останнього оновлення запису (навіть якщо стан не змінився, але змінилися атрибути).	2025-05-27 10:30:00.123456
event_type	VARCHAR	Тип події (для таблиці подій).	automation_triggered, state_changed
event_data	JSON	Дані, пов'язані з подією	{"automation_id": "auto_light_on"}

Взаємодія Home Assistant з системою керування базами даних реалізується через внутрішній модуль Recorder. Цей модуль виконує функцію агрегації даних, збираючи інформацію про стани всіх сутностей та події, що відбуваються у системі. Він також відповідає за фільтрацію даних, що дозволяє зменшити обсяг зберігання, та їх подальший запис до бази даних. Крім того, модуль Recorder надає необхідні механізми для ефективного вилучення історичних даних.

Ці дані є основою для побудови динамічних графіків та діаграм у користувацькому інтерфейсі Lovelace UI, формування звітів щодо енергоспоживання та підтримки функціоналу автоматизації, що вимагає звернення до попередніх станів системи.

Доступ до бази даних здійснюється безпосередньо Home Assistant Core через спеціалізовані інтерфейси, що забезпечують високу ефективність операцій читання та запису. Це гарантує мінімальні затримки при обробці даних, що є критично важливим для забезпечення моніторингу та управління системою в реальному часі.

4.4 Реалізація логіки автоматизації на основі даних сенсорів

У межах даної кваліфікаційної роботи реалізація логіки автоматизації здійснювалась у середовищі Home Assistant, що було розгорнуте на віртуальній машині. Віртуалізація виконувалась з використанням готового образу Home Assistant OS, який дозволив змодельовати роботу системи так, ніби вона інстальована на реальному пристрої Raspberry Pi.

Такий підхід забезпечив повну функціональність платформи, включаючи вбудований супервізор, брокер MQTT, графічний інтерфейс Lovelace, інтеграцію з Zigbee-шлюзами та підтримку сторонніх компонентів.

Емуляція дозволила протестувати типові сценарії роботи розумного будинку в умовах, максимально наближених до реальних. Було реалізовано логіку взаємодії між сенсорами, виконавчими пристроями та платформою управління енергоефективністю. Всі автоматизації описані за допомогою мови YAML, яка є основним форматом для створення сценаріїв у Home Assistant. Це забезпечило повну прозорість, гнучкість і можливість глибокої модифікації логіки роботи системи.

Подальші підрозділи описують структуру автоматизацій, принципи роботи з даними сенсорів, а також конкретні приклади сценаріїв, спрямованих на підвищення енергоефективності житлового приміщення.

4.4.1 Інтерфейс системи Home Assistant

Після запуску віртуальної машини з образом Home Assistant OS система автоматично ініціює створення локального екземпляра розумного дому. При першому вході у веб-інтерфейс користувачу пропонується пройти початкову реєстрацію, в межах якої задається ім'я користувача, пароль та місцезнаходження (рис 4.2-4.4).



Рисунок 4.2 – Головний екран веб-інтерфейсу під час першого запуску.

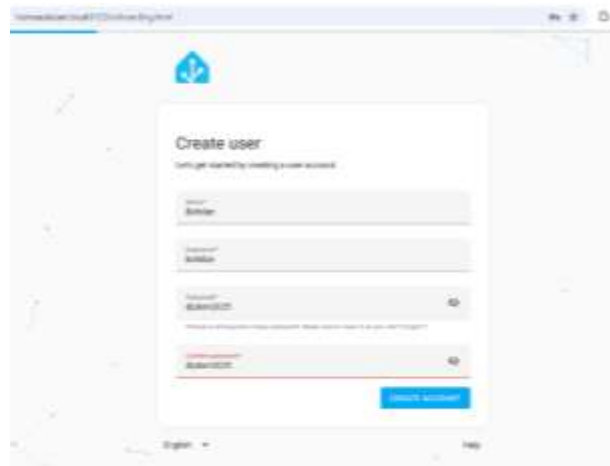


Рисунок 4.3 – Створення облікового запису.

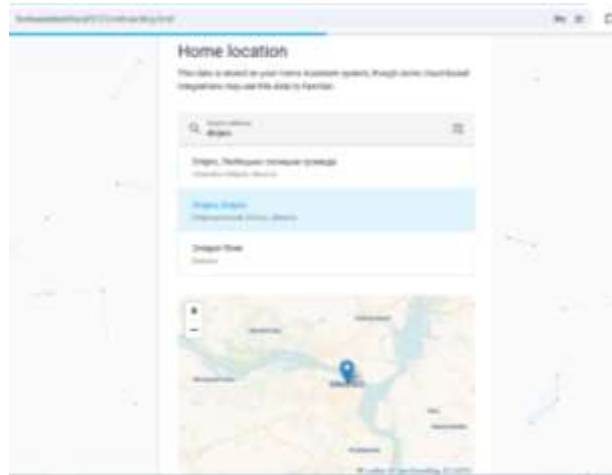


Рисунок 4.4 – Створення облікового запису.

Після проходження початкової реєстрації відкривається основне робоче середовище системи – веб-інтерфейс Home Assistant (рис. 4.5).

Меню включає такі базові пункти, як Overview (огляд системи), Map (геолокація пристроїв), Energy (енергетичний моніторинг), Logbook (журнал подій), History (історія зміни параметрів), Media, To-do lists, а також технічні розділи: Developer tools, Settings, Notifications. Саме з цих пунктів здійснюється не лише моніторинг, а й управління автоматизаціями, налаштування інтеграцій, перегляд споживання енергії тощо.



Рисунок 4.5 – Веб-інтерфейс Home Assistant.

Інтерфейс підтримує динамічну побудову карт і панелей, які можуть адаптуватись до поточного стану системи. Наприклад, можна створити окремі

вкладки для кожної кімнати або сценарію. Це забезпечує швидкий доступ до керування ключовими функціями і робить платформу зручною як для технічного користувача, так і для звичайного мешканця.

4.4.2 Структура та принципи створення автоматизацій в Home Assistant

Після успішного розгортання та початкового налаштування Home Assistant, ключовим етапом для реалізації логіки енергоефективного управління житловим приміщенням є створення автоматизацій. Саме автоматизації перетворюють набір підключених сенсорів та виконавчих пристроїв на інтелектуальну систему, здатну самостійно реагувати на зміни умов та виконувати заздалегідь визначені дії, мінімізуючи участь користувача та оптимізуючи споживання ресурсів.

Доступ до функціоналу створення та управління автоматизаціями здійснюється через головне меню налаштувань системи. При першому вході до цього розділу, або якщо жодної автоматизації ще не було створено, користувач побачить інформаційне повідомлення про призначення автоматизацій та кнопку для ініціювання створення нового сценарію, як це зображено на рисунку 4.6.

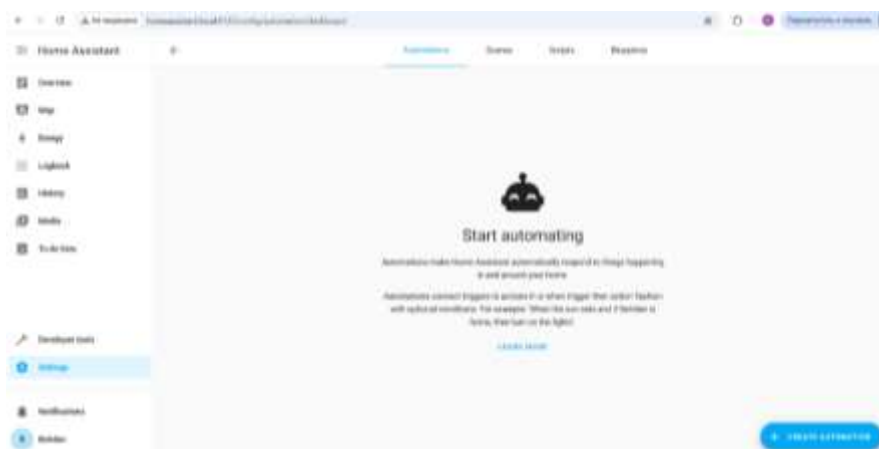


Рисунок 4.6 – Початковий вигляд розділу "Automations & Scenes" в Home Assistant.

Home Assistant надає механізм для побудови автоматизацій, який базується на трьох основних компонентах: тригерах (Triggers), умовах (Conditions) та діях (Actions).

Тригери (Triggers): Це події або зміни станів, які ініціюють запуск автоматизації. Тригером може слугувати сигнал від сенсора (наприклад, датчик руху зафіксував рух, сенсор температури показав значення вище заданого порогу), настання певного часу (наприклад, 7:00 ранку щодня), зміна стану іншого пристрою (наприклад, вимкнення світла у коридорі), або навіть системна подія Home Assistant (наприклад, запуск системи). Одна автоматизація може мати декілька тригерів; спрацювання будь-якого з них запустить перевірку умов.

Умови (Conditions): Це додаткові перевірки, які виконуються після спрацювання тригера. Якщо умови істинні (або не задані), то автоматизація продовжить виконувати дії. Якщо ж хоча б одна з умов хибна, виконання автоматизації на цьому етапі припиняється.

Дії (Actions): Це ті операції, які система повинна виконати, якщо спрацював тригер та всі умови (якщо вони є) виявилися істинними. Дією може бути управління пристроєм (увімкнути світло, встановити температуру на кондиціонері, закрити штори), надсилання сповіщення користувачу, виклик іншої автоматизації або сцени, встановлення таймера тощо.

Для створення автоматизації натиснемо на кнопку "CREATE AUTOMATION" (Створити автоматизацію), розташовану у правому нижньому кутку екрана (рис. 4.6). Після цього система пропонує користувачеві обрати спосіб створення нової автоматизації. Як видно з рисунка 4.7, Home Assistant може запропонувати декілька варіантів.

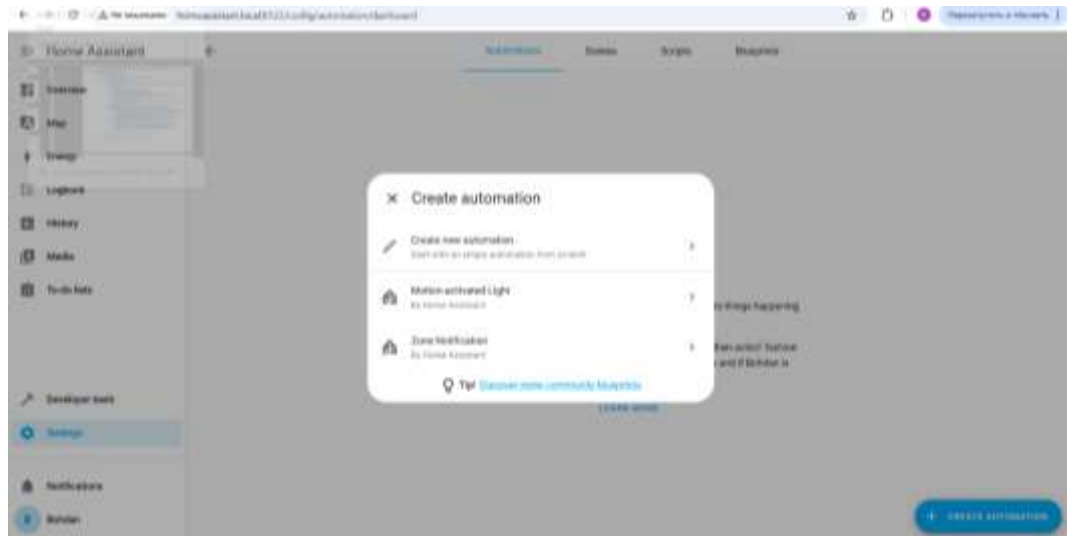


Рисунок 4.7 – Діалогове вікно вибору способу створення автоматизації в Home Assistant.

Для цілей кваліфікаційної роботи, а також для детального вивчення механізму роботи автоматизацій, було обрано перший варіант – "Create new automation" (Створити нову автоматизацію), що дозволяє розпочати з чистого аркуша. Це дасть нам змогу покроково налаштувати кожен елемент сценарію.

У наступних підрозділах розглянемо практичні приклади створення автоматизацій, орієнтованих на підвищення енергоефективності.

4.4.3 Сценарії автоматизації

Одним із базових сценаріїв автоматизації в системах розумного дому є керування освітленням на основі присутності людей в приміщенні. Такий підхід дозволяє мінімізувати випадки залишеного ввімкненого світла в порожніх кімнатах, що безпосередньо впливає на зниження загального споживання електроенергії.

У межах даного сценарію було реалізовано автоматичне вмикання та вимикання освітлення у коридорі на основі даних від датчика присутності та датчику освітленості. Датчик присутності, а також виконавчий пристрій (реле або димер) було підключено до мікроконтролера ESP32, на який попередньо завантажено прошивку ESPHome. Саме в конфігураційному файлі ESPHome

задаються назви сутностей, які згодом ідентифікуються в системі Home Assistant. Зокрема, сенсор присутності було зареєстровано під назвою `binary_sensor.hall_motion`, датчик освітленості під назвою `sensor.hall_luminance`, а керуючий пристрій системи освітлення - як `light.hallway`.

Логіка роботи сценарію полягає у наступному: якщо датчик присутності фіксує присутність у приміщенні та датчик освітленості повідомляє про низький рівень природного світла, що запобігає зайвому вмиканню світла вдень, система вмикає освітлення. Якщо протягом 2 хвилин рух не фіксується – освітлення автоматично вимикається. На рисунку 4.8 наведено YAML-конфігурацію реалізації автоматизації.

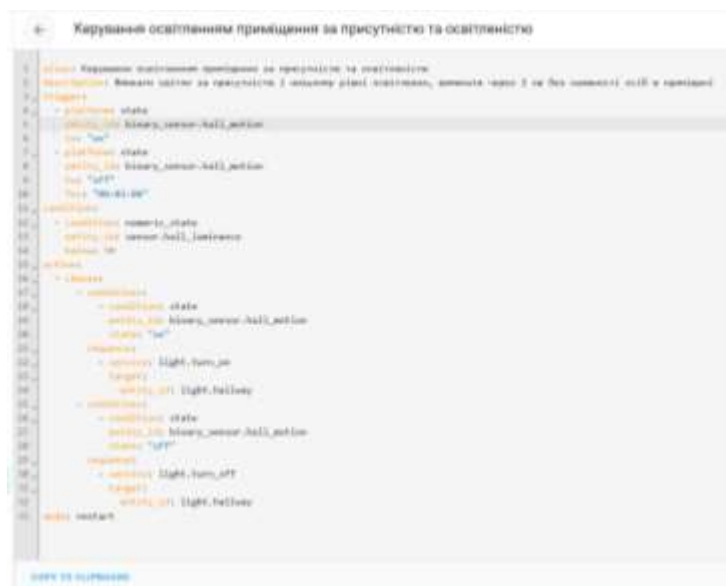


Рисунок 4.8 - YAML-конфігурація реалізації автоматизації системи освітлення.

Для автоматичного регулювання опалення в кімнаті використовується датчик температури, підключений до мікроконтролера ESP32 з прошивкою ESPHome, який передає виміри в Home Assistant через MQTT. Температурний сенсор у системі позначений як сутність `sensor.room_temperature`.

Керування термоголовками здійснюється за допомогою протоколу Zigbee. Кожна термоголовка, яка управляє потоком теплоносія в радіаторі, інтегрована в

Home Assistant через Zigbee-координатор (наприклад, CC2531 або інший шлюз). У Home Assistant термоголовка представлена як сутність `climate`, наприклад, `climate.radiator_valve_1`. Імена сутностей задаються під час інтеграції Zigbee і можуть бути змінені користувачем для зручності (через інтерфейс або конфігурацію).

Основна логіка автоматизації полягає у порівнянні поточної температури кімнати з пороговими значеннями, які задає користувач у Home Assistant (через `input_number.temp_min` та `input_number.temp_max`). Якщо температура падає нижче мінімального порогу, термоголовка відкривається (подає тепло), а якщо вище максимального – закривається.

Додатково система контролює насос опалення, який підключений до окремого реле, керованого через інший ESP32 (позначимо його як `esp_pump`). Цей ESP32 несе відповідальність виключно за вмикання і вимикання насоса. Логіка полягає в тому, що якщо всі термоголовки закриті (тобто не потребують тепла), насос вимикається, аби не працювати даремно. Якщо хоча б одна термоголовка відкрита, насос увімкнено.

Зв'язок між Home Assistant та ESP32 з насосом здійснюється через MQTT або ESPHome, де реле, підключене до насоса, доступне як сутність `switch.pump_relay`. На рисунку 4.9 наведено YAML-конфігурацію реалізації автоматизації системи опалення.

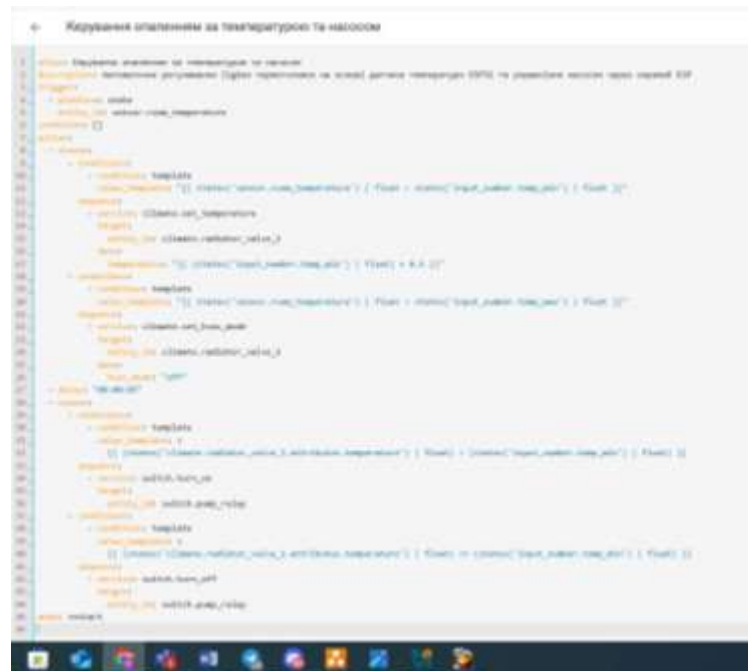


Рисунок 4.9 - YAML-конфігурація реалізації автоматизації системи опалення.

Третій сценарій автоматизації реалізує керування розумними розетками, підключеними через Zigbee-шлюз. Основна мета – зменшити паразитне споживання енергії побутовими пристроями, які залишаються увімкненими в режимі очікування. Керування реалізовано через Home Assistant шляхом створення умов, при яких живлення на розетки автоматично відключається.

Для тестування сценарію було змодельовано розетки, підключені до Zigbee2MQTT. Кожна з них має унікальний `entity_id`, наприклад: `switch.tv_socket`, `switch.kettle_socket`, `switch.bedroom_socket`.

Автоматизація виконується за наступною логікою: після 23:00 система перевіряє, чи присутній хтось у кімнаті за даними сенсора присутності. Якщо сенсор фіксує відсутність руху (`binary_sensor.presence_bedroom == off`), Home Assistant надсилає команду на вимкнення розеток.

YAML-скрипту для цього сценарію представлено на рисунку 4.10.

```

1 | object: Вимкнення розеток у нічний час
2 | trigger:
3 |   - platform: time
4 |     at: "21:00:00"
5 | condition:
6 |   - condition: state
7 |     entity_id: binary_sensor.presence_bedroom
8 |     state: "off"
9 | action:
10 |  - service: switch.turn_off
11 |    target:
12 |      entity_id:
13 |        - switch.tv_socket
14 |        - switch.kettle_socket
15 |        - switch.bedroom_socket
16 | mode: single
17 |

```

Рисунок 4.10 - YAML-конфігурація реалізації автоматизації системи управління споживанням побутових приладів.

Цей підхід дозволяє уникати непотрібного енергоспоживання від пристроїв, які часто залишаються в неактивному стані, але все одно споживають електроенергію. З точки зору архітектури системи, даний сценарій є повністю масштабованим: додавання нових розеток до списку не потребує зміни логіки, достатньо вказати їх entity_id у відповідному блоці target.

Реалізація цього сценарію завершує цикл базових автоматизацій у межах однієї кімнати. Оскільки всі приміщення в обраному житловому будинку мають однакову кількість сенсорів і типові електричні точки, розроблені сценарії можуть бути масштабовані на всі зони будівлі шляхом повторного використання логіки з мінімальними змінами ID.

4.5 Побудова користувацького інтерфейсу та об'єднання логічної структури конфігурацій

Реалізація інтерфейсу здійснена за допомогою графічного рушія Lovelace UI, що є вбудованим у платформу Home Assistant. Завдяки розширеному візуальному редактору Lovelace, з'являється можливість формувати уніфіковані панелі керування без необхідності втручання у конфігураційні файли формату YAML.

Функціональні сутності Home Assistant формуються за допомогою окремих конфігураційних файлів, кожен з яких виконує конкретне призначення. Для забезпечення коректної взаємодії між усіма складовими, використовується центральний конфігураційний файл `configuration.yaml`, що об'єднує інші підфайли через директиви типу «`!include`».

У ході проекту були створені умовні шаблони наступних файлів:

- `sensors.yaml` – модель представлення температурних показників з MQTT-пристроїв;
- `input_number.yaml` – визначення користувацьких змінних для встановлення порогів спрацювання;
- `automations.yaml` – опис умов та дій у відповідь на зміну станів системи;
- `scripts.yaml` – опис послідовних сценаріїв для запуску на вимогу або в межах автоматизацій.

Усі ці файли були логічно інтегровані в межах системи, створюючи повноцінну архітектуру обробки даних, логічного реагування та візуального відображення в інтерфейсі.

На основі попередньо описаних сутностей було реалізовано графічний інтерфейс керування, що включає наступні елементи, доступні для користувача:

1. Відображення актуальної температури у вибраній кімнаті;
2. Інтерактивний повзунок для зміни цільової температури;
3. Інтерактивну кнопку для ручного вмикання або вимикання опалення;
4. Панель статистики енергоспоживання;

Інтерфейс має вкладкову структуру, в якій кожна вкладка відповідає окремій підсистемі, наприклад: “Опалення”, “Освітлення”, “Безпека”. У межах вкладок реалізовані карти типу `gauge`, `entities`, `button`, що забезпечують як контроль, так і моніторинг параметрів. Інтерфейс представлено на рисунку 4.11.



Рисунок 4.11 - Графічний інтерфейс керування.

З міркувань зручності для кінцевого користувача, інтерфейс було налаштовано як окремий дашборд Home Assistant, який завантажується автоматично при відкритті системи. У випадку використання мобільного застосунку або браузера, система автоматично перенаправляє користувача на відповідну сторінку керування, що виключає необхідність додаткових навігаційних дій.

Враховуючи технічні обмеження, пов'язані з виконанням кваліфікаційної роботи в умовах обмеженого доступу до апаратних засобів, розгортання та тестування конфігурацій було здійснено у середовищі віртуальної машини, що повністю імітує функціонал фізичної інсталяції Home Assistant OS.

Це надало змогу реалізувати повноцінну конфігурацію інтерфейсу, перевірити коректність логічних зв'язків, здійснити перевірку структури YAML-файлів, а також апробувати візуальну частину керування системою. Незважаючи на віртуальний характер реалізації, усі етапи були повністю ідентичними фізичному розгортанню, що гарантує релевантність отриманих результатів.

Усі конфігураційні сценарії, реалізовані у форматі YAML, функціонують не окремо, а у взаємозв'язку через загальну інфраструктуру Home Assistant. Якщо в системі існує кілька автоматизацій з різними умовами або діями для цього тригера – всі вони будуть виконані відповідно до заданої логіки. Таким чином, зміна стану

одного пристрою може ініціювати каскад подій – наприклад, одночасне ввімкнення освітлення та надсилання повідомлення на мобільний пристрій.

Завдяки єдиному конфігураційному середовищу, всі YAML-файли взаємодіють як частини одного сценарного механізму, який керується Home Assistant. Всі автоматизації зберігаються в єдиному файлі `automations.yaml`, або розподіляються по структурі `packages`, що дозволяє групувати автоматизації за функціональними зонами (наприклад, окремо для освітлення, клімату, енергоспоживання тощо). Візуалізація цієї взаємодії реалізована у вигляді інтерактивних карт інтерфейсу Lovelace. Кожен компонент, доданий у систему, має свій блок керування, через який користувач може бачити поточний стан, історію змін, а також примусово активувати або деактивувати пристрій чи сценарій.

Таким чином, система працює як єдиний цілісний механізм, де кожна автоматизація, хоч і визначена окремо, є частиною глобальної логіки реагування та управління, що реалізована через Home Assistant. Це забезпечує не лише гнучке масштабування, а й стабільну роботу незалежно від кількості пристроїв та складності сценаріїв.

4.6 Висновки до розділу

У межах четвертого розділу було реалізовано повноцінний програмний компонент системи керування енергоефективністю житлового приміщення, що базується на платформі Home Assistant. У процесі роботи, здійснено розгортання програмного середовища на віртуальній машині, що дозволило відтворити повноцінну інфраструктуру розумного будинку з імітацією дійсної логіки взаємодії між пристроями.

Нами описані та протестовані типові сценарії автоматизації - керування освітленням за присутністю, температурне регулювання через термоголівки, а також контроль енергоспоживання побутових приладів. Усі автоматизації були

реалізовані у форматі YAML та збережені у централізованій конфігурації, що забезпечує єдність логіки та узгодженість дій системи. Комунікація між сенсорами, виконавчими пристроями та сервером Home Assistant здійснювалася за допомогою протоколів MQTT, Zigbee та локальної мережі.

Особливу увагу було приділено візуальній частині системи – інтерфейсу користувача Lovelace, який забезпечує зручне управління усіма аспектами системи навіть для користувачів без технічного бекграунду. Завдяки можливості налаштування індивідуальних панелей керування, система набуває не лише функціональності, але й практичної доступності для кінцевого користувача.

Таким чином, реалізована програмна частина продемонструвала, що з використанням відкритих інструментів Home Assistant можливо створити масштабовану, адаптивну та функціональну систему керування житлом, здатну зменшити енергоспоживання без втрати комфорту, і що найважливіше – придатну для подальшого впровадження в реальних умовах.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було досягнуто основної мети дослідження – інтеграції елементів Інтернету речей у систему контролю енергоефективності житлового комплексу.

У рамках реалізації завдання було здійснено повноцінне проектування кіберфізичної системи, що охоплює як апаратну, так і програмну складову, та забезпечує автоматизоване управління ключовими параметрами середовища в житлових приміщеннях.

Було розроблено комп'ютерну систему, яка включає апаратні модулі збору та обробки даних, локальну мережу зв'язку та сервер Home Assistant як керуючий центр.

Було досягнуто ефективної взаємодії між компонентами системи в результаті системної інтеграції сенсорів температури, освітлення, присутності та розумних виконавчих пристроїв, що дозволяє реалізовувати сценарії автоматизації, адаптовані до поведінки мешканців та умов зовнішнього середовища.

Було реалізовано мережеву інфраструктуру, яка забезпечує безпечний доступ до системи та обмін даними між усіма її компонентами. Налаштовано логічну та фізичну топологію мережі, виконано конфігурування маршрутизації, NAT, OSPF та сервісів DHCP, що дозволило створити надійне середовище для функціонування IoT-рішень.

Особливу увагу було приділено питанням енергоефективності, що проявилось у впровадженні керування освітленням на основі присутності, температурної регуляції з заданих значень користувачем та автоматизованого моніторингу енергоспоживання побутових приладів. Завдяки використанню відкритих протоколів і платформ, система зберігає масштабованість, гнучкість та адаптивність.

Результати роботи засвідчили ефективність обраного архітектурного підходу та підтвердили можливість використання розробленої системи в реальних умовах житлового комплексу.

Таким чином, у даній кваліфікаційній роботі було успішно досягнуто поставлену мету та завдання, продемонстровано можливості сучасних технологій IoT у сфері енергоменеджменту та доведено доцільність інтеграції таких рішень для підвищення ефективності управління ресурсами у житлових приміщеннях.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Морозов М. В. Енергоефективність житлових будинків. *Актуальні питання енергозбереження як вимога безпеки життєдіяльності*. 2018. С. 95–100.
2. Д. В. Степанов Д. М. Резидент В. В. Мартиненко. Аналіз заходів для підвищення енергоефективності житлового будинку. *Вісник вінницького політехнічного інституту*. 2024. Вип. 6. С. 62–67.
3. Цвіркун, Л., Соболевський. Аналіз особливостей використання туманних комп'ютерних середовищ для побудови IoT інфраструктури. *Information Technology: Computer Science, Software Engineering and Cyber Security*. 2025. С. 238–243.
4. Leonardo Angelini, Elena Mugellini, Omar Abou Khaled and Nadine Couture. Internet of Tangible Things (IoTТ): Challenges and Opportunities for Tangible Interaction with IoT. *Informatic*. 2018.
5. Waqar Tariq, Abid Mustafa, Zahid Rasool, S.M.Haseeb, S.Mubarak Ali, Ashad Mustafa, Shujat Khan, Saad Irfan Warsi. Building Management System for IQRA University. *Informatic*. 2012.
6. Лановський М. О., Гура В. І. Впровадження цифрових технологій інтернету речей для поліпшення енергоощадних технологій у будівлях. *Наукові записки Міжнародного гуманітарного університету*. 2022. С. 151–154.
7. Уривський Л.О. et al. Дослідження і розробка рішень Інтернету речей широкого застосування // Sciences of Europe. Global Science Center LP, 2019. Вип. 1, № 36. С. 39–54.
8. Metallidou С.К., Psannis К.Е., Egyptiadou Е.А. Energy Efficiency in Smart Buildings: IoT Approaches // IEEE Access. Institute of Electrical and Electronics Engineers Inc., 2020. Вип. 8. С. 63679–63699.
9. Жураковський Б.Ю., Зенів І.О. Технології інтернету речей. Навчальний посібник. Київ: КПІ ім. Ігоря Сікорського, 2021. 271 с.

10. Цвіркун Л. І., Євстігнєєва А. А., Панферова Я. В. Розробка програмного забезпечення комп'ютерних систем. Програмування: навч. посіб. 3-тє вид., випр. Дніпропетровськ: НГУ, 2016. 223 с.
11. Панферова Я.В., Кмітіна І. В., Цвіркун Л. І. Комп'ютерні мережі. Методичні вказівки до виконання лабораторних робіт студентами напряму підготовки 6.050102 Комп'ютерна інженерія. Дніпропетровськ: НГУ, 2012. 31 с.
12. Abdullah. A Thesis on Energy Management and Cost Analysis. *Daffodil International University Journal of Science and Technology*. 2020. 46 p.
13. Основи Інтернету Речей.
URL: <http://edu.asu.in.ua/mod/book/tool/print/index.php?id=112> (дата звернення: 05.06.2025).
14. Building Management System to support building renovation.
URL: <https://journals.ucc.ie/index.php/boolean/article/view/boolean-2010-37/html-en> (дата звернення: 05.06.2025).
15. Сертифікація енергетичної ефективності будівель. *Державне агентство з енергоефективності та енергозбереження України*.
URL: <https://sae.gov.ua/diialnist/sertyfikatsiia-enerhetychnoi-efektyvnosti-budivel> (дата звернення: 05.06.2025).
16. Raspberry Pi OS. *Raspberry Pi*.
URL: <https://www.raspberrypi.com/documentation/computers/os.html#use-gpio-from-python> (дата звернення: 05.06.2025).
17. Automating Home Assistant. *Home Assistant*.
URL: <https://www.home-assistant.io/docs/automation/> (дата звернення: 05.06.2025).
18. Що таке Zigbee і чому це важливо для вашого розумного будинку? *Xterm*.
URL: <https://surli.cc/hjciob> (дата звернення: 05.06.2025).

19. What's the Difference Between YAML and JSON? AWS.

URL: https://aws.amazon.com/compare/the-difference-between-yaml-and-json/?nc1=h_ls (дата звернення: 05.06.2025).

20. The internet of things with esp32. *Esp32*.

URL: <http://esp32.net/> (дата звернення: 05.06.2025).

Додаток А

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
“ДНІПРОВСЬКА ПОЛІТЕХНІКА”

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
YAML-КОНФІГУРАЦІЇ РЕАЛІЗАЦІЇ АВТОМАТИЗАЦІЇ

Текст програми

804.02070743.25010-01 12 01

Листів 7

2025

АНОТАЦІЯ

Дана програма містить у собі фрагменти конфігураційних сценаріїв автоматизації, розроблених для платформи Home Assistant.

Сценарії реалізують логіку управління кліматом, освітленням та побутовими приладами на основі даних сенсорів та користувацьких параметрів.

Конфігураційні файли написані мовою розмітки YAML і призначені для інтеграції у систему автоматизації житлового приміщення з метою підвищення енергоефективності.

Автоматизації відлагоджені у середовищі Home Assistant OS із використанням віртуального стенду.

ЗМІСТ

1. YAML-скрипт керування освітленням.....	4
2. YAML-скрипт керування опаленням.....	5
3. YAML-скрипт керування побутовими приладами.....	7

1 YAML-СКРИПТ КЕРУВАННЯ ОСВІТЛЕННЯМ

alias: Керування освітленням приміщення за присутністю та освітленістю

trigger:

- platform: state
entity_id: binary_sensor.hall_motion
to: "on"
 - platform: state
entity_id: binary_sensor.hall_motion
to: "off"
- for:
minutes: 2

condition:

- condition: numeric_state
entity_id: sensor.hall_luminance
below: 120

action:

- choose:
 - conditions:
 - condition: state
entity_id: binary_sensor.hall_motion
state: "on"
 - sequence:
 - service: light.turn_on
target:
entity_id: light.hallway
 - conditions:
 - condition: state
entity_id: binary_sensor.hall_motion
state: "off"
 - sequence:
 - service: light.turn_off
target:
entity_id: light.hallway

mode: restart

2 YAML-СКРИПТ КЕРУВАННЯ ОПАЛЕННЯМ

alias: Керування опаленням за температурою та насосом

description: >-

Автоматичне регулювання відкриття термоголовок та керування насосом через ESP на основі температурного сенсора.

trigger:

- platform: state
- entity_id: sensor.room_temperature

condition: []

action:

- choose:
 - conditions:
 - condition: template
 - value_template: >


```
{{ states('sensor.room_temperature') | float < states('input_number.temp_min') | float }}
```

sequence:

- service: climate.set_temperature

target:

entity_id: climate.radiator_valve_1

data:

temperature: 25

- conditions:

- condition: template

value_template: >

```
{{ states('sensor.room_temperature') | float > states('input_number.temp_max') |
```

```
float }}
```

sequence:

- service: climate.set_temperature

target:

entity_id: climate.radiator_valve_1

data:

temperature: 5

- delay: 00:00:02

```

- choose:
  - conditions:
    - condition: template
      value_template: >
        {{ states('climate.radiator_valve_1').attributes.temperature | float >
states('input_number.temp_min') | float }}
      sequence:
        - service: switch.turn_on
          target:
            entity_id: switch.pump_relay
    - conditions:
      - condition: template
        value_template: >
          {{ states('climate.radiator_valve_1').attributes.temperature | float <=
states('input_number.temp_min') | float }}
        sequence:
          - service: switch.turn_off
            target:
              entity_id: switch.pump_relay
mode: restart

```

3 YAML-СКРИПТ КЕРУВАННЯ ПОБУТОВИМИ ПРИЛАДАМИ

alias: Вимкнення розеток у нічний час

description: >-

Вимкнення розеток після 23:00 при відсутності руху в кімнаті для уникнення паразитного споживання.

trigger:

- platform: time
- at: "23:00:00"

condition:

- condition: state
- entity_id: binary_sensor.presence_bedroom
- state: "off"

action:

- service: switch.turn_off

target:

entity_id:

- switch.tv_socket
- switch.kettle_socket
- switch.bedroom_socket

mode: single

Додаток Б

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
“ДНІПРОВСЬКА ПОЛІТЕХНІКА”

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
НАЛАШТУВАННЯ МЕРЕЖІ КОМП'ЮТЕРНОЇ СИСТЕМИ

Текст програми

804.02070743.25010-01 12 01

Листів 5

2025

АНОТАЦІЯ

Дані конфігурації містять частину програмного коду, що використовувався для налаштування мережевих компонентів комп'ютерної системи управління енергоефективністю житлового комплексу з елементами Інтернету речей. Конфігураційні файли включають налаштування головного маршрутизатора системи та маршрутизатора одного з житлових будинків.

Налаштування забезпечують взаємодію між локальними мережами IoT-пристроїв, серверів і користувацьких пристроїв у межах кожного будинку, а також централізований доступ до основного серверного центру через головний маршрутизатор. В конфігурації реалізовані механізми динамічної маршрутизації OSPF, базові правила NAT, інтерфейсна адресація, активація IP-маршрутизації та організація шлюзів за замовчуванням для кожної підмережі.

Ці налаштування були реалізовані в середовищі Cisco Packet Tracer і є частиною моделювання архітектури мережі житлового комплексу з метою демонстрації логіки роботи системи в рамках дипломного проєкту.

ЗМІСТ

1. Налаштування головного маршрутизатора комп'ютерної системи.....4
2. Налаштування маршрутизатора R1 першого будинку житлового комплексу5

1 НАЛАШТУВАННЯ ГОЛОВНОГО МАРШРУТИЗАТОРА КОМП'ЮТЕРНОЇ СИСТЕМИ

```

version 12.2
hostname Router
ip dhcp excluded-address 192.168.101.1 192.168.101.9

ip dhcp pool 192.168.101.1
network 192.168.101.0 255.255.255.0
default-router 192.168.101.1

interface GigabitEthernet0/0
ip address 10.0.1.1 255.255.255.252
ip nat inside
duplex auto
speed auto

interface GigabitEthernet1/0
ip address 10.0.2.1 255.255.255.252
ip nat inside
duplex auto
speed auto

interface GigabitEthernet2/0
ip address 10.0.3.1 255.255.255.252
ip nat inside
duplex auto
speed auto

interface GigabitEthernet3/0
ip address 10.0.4.1 255.255.255.252
ip nat inside
duplex auto
speed auto

interface GigabitEthernet4/0
ip address 192.168.100.1 255.255.255.0
ip nat inside
duplex auto
speed auto

interface GigabitEthernet5/0
ip address 192.168.101.1 255.255.255.0
ip nat inside
duplex auto
speed auto

interface GigabitEthernet6/0
ip address 209.165.200.4 255.255.255.0
ip nat outside
duplex auto
speed auto

router ospf 1
router-id 1.1.1.1
log-adjacency-changes
network 192.168.100.0 0.0.0.255 area 0
network 192.168.101.0 0.0.0.255 area 0
network 10.0.1.0 0.0.0.3 area 0
network 10.0.2.0 0.0.0.3 area 0
network 10.0.3.0 0.0.0.3 area 0
network 10.0.4.0 0.0.0.3 area 0

ip nat inside source list 1 interface
GigabitEthernet6/0 overload
ip classless
ip flow-export version 9

access-list 1 permit 10.0.1.0 0.0.0.3
access-list 1 permit 10.0.2.0 0.0.0.3
access-list 1 permit 10.0.3.0 0.0.0.3
access-list 1 permit 10.0.4.0 0.0.0.3
access-list 1 permit 192.168.100.0 0.0.0.255
access-list 1 permit 192.168.101.0 0.0.0.255

line con 0
line aux 0
line vty 0 4
login

```

2 НАЛАШТУВАННЯ МАРШРУТИЗАТОРА R1 ПЕРШОГО БУДИНКУ ЖИТЛОВОГО КОМПЛЕКСУ

```
version 12.2
hostname Router

ip dhcp excluded-address 192.168.3.1
192.168.3.9

ip dhcp pool 192.168.3.1
network 192.168.3.0 255.255.255.0
default-router 192.168.3.1

interface GigabitEthernet0/0
ip address 192.168.1.1 255.255.255.0
duplex auto
speed auto

interface GigabitEthernet1/0
ip address 192.168.2.1 255.255.255.0
duplex auto
speed auto

interface GigabitEthernet2/0
ip address 192.168.3.1 255.255.255.0
duplex auto
speed auto

interface FastEthernet3/0
ip address 10.0.1.2 255.255.255.252
duplex auto
speed auto

router ospf 1
router-id 2.2.2.1
log-adjacency-changes
network 192.168.1.0 0.0.0.255 area 0
network 192.168.2.0 0.0.0.255 area 0
network 192.168.3.0 0.0.0.255 area 0
network 10.0.1.0 0.0.0.3 area 0

ip classless
ip flow-export version 9

line con 0
line aux 0
line vty 0 4
login
```