Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет прикладних інформаційних технологій та електроінженерії
(повна назва факультету)
Кафедра автоматизації технологічних процесів і виробництв
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: Розробка та дослідження IoT-системи для безпечного моніторингу та керування домашніми пристроями з підтримкою OpenHAB

Виконав: студент VI курсу, групи КАм-61

спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
(шифр і назва спеціальності)

_____ Павелко Р.Р.
(підпис) (прізвище та ініціали)

_____ Биць Д.М.
(підпис) (прізвище та ініціали)

Керівник _____ Трембач Р.Б.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ Козбур І.Р.
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ Савків В.Б.
(підпис) (прізвище та ініціали)

Рецензент _____ Дідич І.С.
(підпис) (прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет прикладних інформаційних технологій та електроінженерії
(повна назва факультету)

Кафедра автоматизації технологічних процесів і виробництв
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Савків В.Б.
(підпис) (прізвище та ініціали)
«_____» листопада 2025 р.__

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
(шифр і назва спеціальності)

Студенту Павелку Роману Романовичу, Биць Дмитру Мирославовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та дослідження IoT-системи для безпечного моніторингу та керування домашніми пристроями з підтримкою OpenHAB

Керівник роботи Трембач Ростислав Богданович, к.т.н. , доцент кафедри АВ.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 20 » листопада 2025 року № 4/7-1007

2. Термін подання студентом завершеної роботи 23 грудня 2025р.

3. Вихідні дані до роботи Наукові публікації про IoT-системи для безпечного моніторингу та керування домашніми пристроями з підтримкою OpenHAB

4. Зміст роботи

Вступ, 1 Аналітична частина, 1.1 Актуальність домашньої IoT-автоматизації, 1.2 Моделі архітектури домашніх IoT-систем і місце openHAB, 1.3 Протоколи взаємодії та криптографічні механізми захисту, 1.4 Аналіз openHAB як платформи інтеграції та автоматизації, 1.5 Вимоги безпеки та нормативно-методичні орієнтири, 1.6 Висновок до першого розділу, 2 Технологічна частина, 2.1 Вимоги до системи та критерії вибору технологій, 2.2 Вибір програмних засобів та середовищ розробки, 2.3 Вибір протоколів обміну та механізмів захисту каналів, 2.4 Висновок до другого розділу, 3 Конструкторська частина, 3.1 Архітектурна компоновка та функціональні вузли системи, 3.2 Конструкція периферійного сенсорного вузла, 3.3 Алгоритмічна та програмна реалізація системи, 3.4 Модель взаємодії вузлів: MQTT-топіки, QoS і буферизація, 3.5 Конструктивна інтеграція з openHAB, 3.6 Конструктивні заходи кіберзахисту, 3.7 Висновок до третього розділу 4 Науково дослідна частина, 4.1 Мета, дослідницькі питання та гіпотези, 4.2 Експериментальний стенд та відтворюваність, 4.3 Метрики, методи вимірювання та критерії якості, 4.4 Базова продуктивність MQTT, 4.5 Вплив захисту транспортного рівня, 4.6 Енергетична ціна безпеки на периферійних вузлах, 4.7 Контроль доступу та токенізація, 4.8 Стійкість до атак/збоїв та перевірка захисних контурів, 4.9 Обробка результатів, статистична значущість та валідність, 4.10 Висновок до четвертого розділу, 5 Спеціальна частина, 5.1 Нормативно-методична рамка безпеки, 5.2 Модель загроз і політика доступу, 5.3 Захищений віддалений доступ, 5.4 Захищена телеметрія та керування через MQTT, 5.5 Приватність даних і правові вимоги, 5.6 Експлуатаційна безпека в житловому середовищі, 5.7 Набір практичних заходів для системи, 5.8 Висновок до п'ятого розділу, 6 Охорона праці та безпека в надзвичайних ситуаціях Висновки, Перелік джерел, Додатки.

Перелік графічного матеріалу 1 Титульна сторінка. 2 Тема, Мета, Об'єкт, Предмет дослідження. 3 Завдання дослідження. 4 Актуальність дослідження. 5 Контекст і межі IoT-системи 6 Компоненти розумного дому, 7 Модель загроз і довірчі межі, 8 Цільова архітектура з openHAB, 9 Мережева топологія LAN/IoT VLAN/VPN, 10 Use Case діаграма, 11 Структура сенсорного та актуаторного вузла 12 Демонстраційний стенд/макет, 13 Експериментальний стенд, 14 Точки вимірювання (t0–t3) і метрики, 15 Порівняння режимів захисту каналу, 16 Висновки. 17 Завершальний

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	доц. каф. КТ Тотосько О.В.		
Безпека в надзвичайних ситуаціях	асист. каф. ОХ Теслюк В.М.		

7. Дата видачі завдання 20 листопада 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	20.11.2024	*Виконано*
2.	Підбір джерел про системи моніторингу	30.11.2025-01.12.2025	*Виконано*
3.	Вибір компонентної бази	02.12.2025-03.12.2025	*Виконано*
4.	Виконання дослідження автоматизованої системи	04.12.2025-05.12.2025	*Виконано*
5.	Переклад та опрацювання джерел	06.12.2025-07.12.2025	*Виконано*
6.	Розроблення архітектури системи	08.12.2025-09.12.2025	*Виконано*
7.	Розроблення панелі керування системи	10.12.2025-11.12.2025	*Виконано*
8.	Оформлення розділу «Аналітична частина»	12.12.2025-13.12.2025	*Виконано*
9.	Оформлення розділу «Конструкторська частина»	13.12.2025-14.12.2025	*Виконано*
10.	Оформлення розділу «Науково дослідна частина»	14.12.2025-15.12.2025	*Виконано*
11.	Оформлення розділу «Технологічна частина»	15.12.2025-16.12.2025	*Виконано*
12.	Оформлення розділу «Спеціальна частина»	16.12.2025-17.12.2025	*Виконано*
13.	Виконання розділу «Охорона праці та безпека в НС»	17.12.2025-18.12.2025	*Виконано*
14	Підготовка графічного матеріалу	18.12.2025-19.12.2025	*Виконано*
15.	Оформлення кваліфікаційної роботи	19.12.2025-20.12.2025	*Виконано*
16.	Нормоконтроль	20.12.2025-21.12.2025	*Виконано*
17.	Перевірка на плагіат	21.12.2025	*Виконано*
18.	Попередній захист кваліфікаційної роботи	22.12.2025	*Виконано*
19	Захист кваліфікаційної роботи	23.12.2025	

Студент _____ Павелко Р.Р.

Студент _____ Биць Д.М.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Трембач Р.Б.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка та дослідження IoT-системи для безпечного моніторингу та керування домашніми пристроями з підтримкою OpenHAB // Кваліфікаційна робота освітнього рівня «Магістр» // Павелко Роман Романович, Биць Дмитро Мирославович // Тернопільський національний технічний університет імені Івана Пулюя, факультет прикладних інформаційних технологій та електроінженерії, кафедра автоматизації технологічних процесів і виробництв, група КАм-61 // Тернопіль, 2025 // С. 73, рис. – 10, табл. – 1, кресл. – 17, додат. – 9, бібліогр. – 54.

Ключові слова: інтернет речей; розумний дім, openHAB, MQTT, кібербезпека, захист даних.

Кваліфікаційна робота присвячена розробці та дослідженню IoT-системи безпечного моніторингу й керування домашніми пристроями на базі платформи openHAB.

В першому розділі розглянуто сучасні підходи до побудови систем «розумного дому», основні архітектури, протоколи взаємодії та загрози інформаційній безпеці.

В другому розділі подано технологічні рішення реалізації системи, обґрунтовано вибір платформи openHAB і протоколів обміну даними.

В третьому розділі описано конструкторські рішення апаратної та програмної реалізації сенсорних і виконавчих вузлів.

В четвертому розділі виконано експериментальні дослідження характеристик системи та впливу механізмів захисту на її роботу.

В п’ятому розділі розглянуто питання безпечної експлуатації, керування доступом і відповідності сучасним стандартам кібербезпеки.

В шостому розділі кваліфікаційної роботи розглянуто забезпечення безпечної роботи з обладнанням.

Об’єкт дослідження: IoT-системи домашньої автоматизації, призначені для моніторингу та керування побутовими пристроями в умовах локального та віддаленого доступу.

Предмет дослідження: методи, архітектурні рішення та засоби забезпечення безпечного обміну даними і керування пристроями в IoT-системі розумного дому на базі платформи openHAB.

ANNOTATION

Development and research of an IoT system for secure monitoring and control of home devices with OpenHAB support // The educational level "Master" qualification work // Pavelko Roman Romanovych, Byts Dmytro Myroslavovych// Ternopil Ivan Pulyuy National Technical University, Faculty of Applied Information Technologies and Electrical Engineering, Department of Automation of Technological Processes and Production, KAm-61 group // Ternopil, 2025 // P. 73, fig. - 10, tables - 1, chair. - 17, annexes - 9, ref. - 54.

Key words: Internet of Things; smart home, openHAB, MQTT, cybersecurity, data protection.

Qualification work is devoted to the development and research of an IoT system for secure monitoring and control of home devices based on the openHAB platform.

The first chapter discusses modern approaches to building smart home systems, basic architectures, interaction protocols, and threats to information security.

The second chapter presents technological solutions for implementing the system and justifies the choice of the openHAB platform and data exchange protocols.

The third chapter describes the design solutions for the hardware and software implementation of sensor and actuator nodes.

The fourth chapter presents experimental studies of the system's characteristics and the impact of protection mechanisms on its operation.

The fifth chapter examines issues of safe operation, access control, and compliance with modern cybersecurity standards.

The sixth chapter of the thesis examines ensuring safe operation of the equipment.

Object of research: IoT home automation systems designed to monitor and control household devices in local and remote access conditions.

Subject of research: methods, architectural solutions and means of ensuring secure data exchange and device control in a smart home IoT system based on the openHAB platform.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ACL (Access Control List) — список керування доступом, що визначає дозволені дії для користувачів або пристроїв.

API (Application Programming Interface) — програмний інтерфейс для взаємодії між компонентами системи.

CoAP (Constrained Application Protocol) — легкий прикладний протокол для IoT-пристроїв з обмеженими ресурсами.

DTLS (Datagram Transport Layer Security) — протокол захисту даних для з’єднань на основі UDP.

E2E (End-to-End) — наскрізний, від джерела до кінцевого споживача.

HTTP/HTTPS (HyperText Transfer Protocol / Secure) — протокол передавання гіпертексту / його захищена версія.

IoT (Internet of Things) — Інтернет речей.

JWT (JSON Web Token) — формат токена для безпечної передачі тверджень між сторонами.

MQTT (Message Queuing Telemetry Transport) — легкий брокер-орієнтований протокол обміну повідомленнями.

openHAB (Open Home Automation Bus) — платформа з відкритим кодом для домашньої автоматизації.

QoS (Quality of Service) — рівень якості обслуговування під час передавання повідомлень.

VPN (Virtual Private Network) — віртуальна приватна мережа.

VLAN (Virtual Local Area Network) — віртуальна локальна мережа.

Wi-Fi — технологія бездротового локального зв’язку.

мс (ms) — мілісекунда, одиниця вимірювання часу затримки.

В (V) — вольт, одиниця електричної напруги.

Вт (W) — ват, одиниця потужності.

Гц (Hz) — герц, одиниця частоти..

ЗМІСТ

ВСТУП

Актуальність теми. Стрімке поширення концепції «розумного дому» як прикладного сегмента Інтернету речей (IoT) супроводжується зростанням кількості підключених сенсорів, виконавчих пристроїв і програмних сервісів, що взаємодіють у межах домашньої мережі та через віддалений доступ. Паралельно з розширенням функціональності підвищуються вимоги до надійності, керованості, енергоефективності й, насамперед, до інформаційної безпеки, оскільки компрометація домашньої IoT-інфраструктури здатна спричинити не лише витік приватних даних, а й неконтрольований вплив на фізичні об’єкти (електроживлення, опалення, вентиляцію, замки, сигналізацію). У реальних впровадженнях критичним стає узгодження двох взаємно суперечливих чинників: з одного боку — необхідності простої інтеграції гетерогенних пристроїв різних виробників, з іншого — потреби послідовно впроваджувати механізми автентифікації, авторизації та захисту каналів зв’язку без істотної деградації користувацького досвіду і збільшення вартості системи.

Платформи домашньої автоматизації з відкритим кодом, зокрема OpenHAB, є поширеним підходом до побудови інтеграційного рівня, який забезпечує підключення різних протоколів і типів пристроїв, централізоване керування та сценарну логіку. Водночас практичний досвід використання таких платформ показує, що «безпека за замовчуванням» часто залежить від коректності конфігурації, політик доступу, сегментації мережі та дисципліни оновлень, а також від загальної захищеності домашнього Wi-Fi середовища. У ряді прототипних рішень додаткові шари захисту реалізують шляхом поєднання брокер-орієнтованої телеметрії (наприклад, MQTT) з криптографічними механізмами та токен-орієнтованою ідентифікацією користувачів. Показовим є підхід, у якому OpenHAB 2 розгортається на одноплатному комп’ютері (Raspberry Pi) як центральному сервері, а вузли збору даних і керування реалізуються на мікроконтролерних платформах із бездротовими модулями; при цьому віддалений доступ обмежується механізмами JWT, а конфіденційність даних підсилюється застосуванням

симетричного шифрування (AES). Такий напрямок є актуальним, однак потребує поглибленого наукового обґрунтування вибору архітектури, протоколів і криптопримітивів, а також експериментального підтвердження компромісу «безпека–затримка–надійність–вартість» у типових сценаріях.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Магістр» є розробка та дослідження IoT-системи для безпечного моніторингу та керування домашніми пристроями з підтримкою OpenHAB, орієнтованої на комбінований режим доступу та підсиленої механізмами контролю доступу й захисту даних на прикладному рівні. Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

– дослідити методи управління мікромережами з використанням IoT.

– виконати аналітичний огляд сучасних підходів до побудови систем «розумного дому» та їхніх типових векторів атак;

– обґрунтувати вибір архітектури та протоколів взаємодії між серверною частиною, вузлами збору даних і виконавчими механізмами;

– спроєктувати конструктивну та програмну структуру прототипу на базі OpenHAB із використанням брокера повідомлень і мікроконтролерних вузлів;

– реалізувати практичні елементи системи;

– розробити методику експериментальних досліджень і виконати вимірювання ключових показників (затримки, надійності доставки, відтворюваності сповіщень) та оцінити вплив механізмів безпеки на ці показники; сформувати рекомендації щодо безпечної експлуатації й типових конфігураційних практик для домашнього середовища.

Об’єкт дослідження IoT-системи домашньої автоматизації з централізованою інтеграційною платформою та мережею периферійних сенсорних/виконавчих вузлів.

Предмет дослідження методи, засоби та архітектурні рішення побудови безпечної телеметрії та керованого доступу до функцій «розумного дому» на основі OpenHAB, а також закономірності впливу обраних протоколів і механізмів захисту на часові та надійнісні характеристики обміну даними.

Наукова новизна одержаних результатів роботи полягає в обґрунтуванні та комплексній оцінці архітектурного рішення IoT-системи «розумного дому» на основі OpenHAB з підсиленням безпеки через чітко визначену модель доступу, захист передавання даних та формалізовану методику експериментальної перевірки показників якості сервісу з урахуванням накладних витрат безпекових механізмів. Складовою новизни є систематизація типових ризиків експлуатації платформи домашньої автоматизації в умовах реальної мережевої інфраструктури і формування практичних рекомендацій зменшення ризику компрометації за рахунок конфігураційних, організаційних заходів, узгоджених з обраною архітектурою.

Практичне значення одержаних результатів. полягає у створенні прототипу IoT-системи безпечного моніторингу та керування домашніми пристроями з підтримкою OpenHAB, який може бути відтворений на доступній елементній базі (одноплатний комп’ютер як сервер та мікроконтролерні вузли як периферія), а також у розробленні технологічних рекомендацій з розгортання, налаштування й експлуатації. Отримані результати можуть використовуватися під час проєктування домашніх або маломасштабних комерційних рішень автоматизації (квартири, приватні будинки, невеликі офіси), а також у навчальному процесі при викладанні дисциплін, пов’язаних із IoT, мережевими протоколами та кібербезпекою.

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на XIII Науково-технічній конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя (м. Тернопіль, 2025 р.).

Публікації. Основні результати кваліфікаційної роботи опубліковано у конференції (Див. додаток А).

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, шістьох розділів, висновків, списку літератури з 54 найменувань та 9 додатків. Загальний обсяг кваліфікаційної роботи складає 85 сторінки, з них 73 сторінки основного тексту, який містить 10 рисунків та 1 таблицю.

1 АНАЛІТИЧНА ЧАСТИНА

Домашня автоматизація на базі IoT трансформується з «набору розумних гаджетів» у кіберфізичну систему, яка безпосередньо впливає на безпеку, приватність і безперервність побутових процесів (доступ до помешкання, освітлення, опалення, вентиляція, сигналізація, енергоспоживання тощо). При цьому типова домашня інфраструктура поєднує різнорідні пристрої та протоколи, а контур керування часто виходить за межі локальної мережі через мобільні застосунки й хмарні сервіси.

У межах цієї роботи ядром системи обрано openHAB як платформу інтеграції та автоматизації, а ключовим пріоритетом визначено безпечний моніторинг і керування домашніми пристроями з урахуванням реальних загроз і обмежень ресурсів IoT-вузлів. Базовою відправною точкою слугує прототип захищеної бездротової системи домашньої автоматизації на OpenHAB 2, у якій для автентифікації та захисту даних застосовано JWT і AES, а також проведено тестування із використанням інструментів аудиту безпеки Wi-Fi [1].

1.1 Актуальність домашньої IoT-автоматизації

Практична цінність smart home визначається двома факторами: по-перше, розширенням переліку керованих підсистем (від «розумних» ламп до інтегрованих сценаріїв мікроклімату), по-друге, зростанням залежності користувача від віддаленого доступу. Водночас саме віддалений доступ перетворює домашню систему на частину Інтернет-периметра, що збільшує площу атаки: з’являються ризики перехоплення трафіку, підміни команд, компрометації облікових даних, атак на брокери/шлюзи та експлуатації вразливостей API.

Для openHAB характерною є наявність REST-інтерфейсу як універсального механізму інтеграції (керування Items, отримання станів, робота з ресурсами платформи). Однак документація прямо вказує, що доступ до REST API через Інтернет можливий, але становить «значний ризик безпеки», і

користувачам рекомендується забезпечувати захищені канали доступу [6]. Це формує базову аналітичну вимогу: архітектура системи має бути спроєктована так, щоб (а) мінімізувати потребу в прямій експозиції сервісів у мережу Інтернет, (б) застосовувати чітку модель автентифікації/авторизації та криптографічного захисту, (в) забезпечувати контроль цілісності й журналювання дій.\n\nОкремо варто врахувати висновок із базового прототипу: автори відзначають, що «одним із найбільших викликів» для OpenHAB є відсутність примусового механізму контролю доступу для користувачів, через що безпека істотно залежить від «міцності бездротової мережі», і тому запропоновано власну модель автентифікації/авторизації [1].

1.2 Моделі архітектури домашніх IoT-систем і місце openHAB

У загальному вигляді домашня IoT-система може бути описана як багаторівнева архітектура: рівень «польових» пристроїв (сенсори/актуатори), рівень локального шлюзу/контролера (edge), рівень інтеграції та автоматизації (платформа), рівень клієнтських застосунків і, за потреби, хмарний рівень (для віддаленого доступу/повідомлень/аналітики). Для безпеки принципово важливо, що перенос функцій «керування» на edge зменшує залежність від зовнішніх сервісів і дає можливість зберігати критичні сценарії (сигналізація, аварійне вимкнення, протікання води тощо) працездатними навіть при втраті Інтернет-каналу.

openHAB концептуально розділяє фізичний світ і прикладний рівень. На стороні фізичного світу виступають Things та їх канали (Channels), а прикладна логіка і UI працюють із Items як «віртуальним шаром», що має стан і використовується через події [2], [3]. Items можуть бути прив’язані до каналів binding’ів, отримуючи показники сенсорів або керуючи актуаторами, що робить їх універсальними точками інтеграції для правил автоматизації [4]. Така модель є зручною для побудови безпекових контурів: можна централізовано визначати, які Items доступні для читання/керування конкретними ролями або зовнішніми

сервісами, а також які події мають підлягати журналюванню чи виявленню аномалій.

Над Items у платформі працює механізм правил. У відкритій документації описано Rules DSL як інтегрований «легкий, але потужний» рушій правил, який використовується для реалізації сценаріїв автоматизації та реакції на події [5]. Для аналітичної частини це означає, що безпека повинна враховувати не лише транспорт і доступ до API, а й логіку правил: помилково визначене правило може виконувати небажані дії, а доступ до редагування правил має бути суворо обмежений.

На рисунку 1.1 наведено компоненти розумного дому.

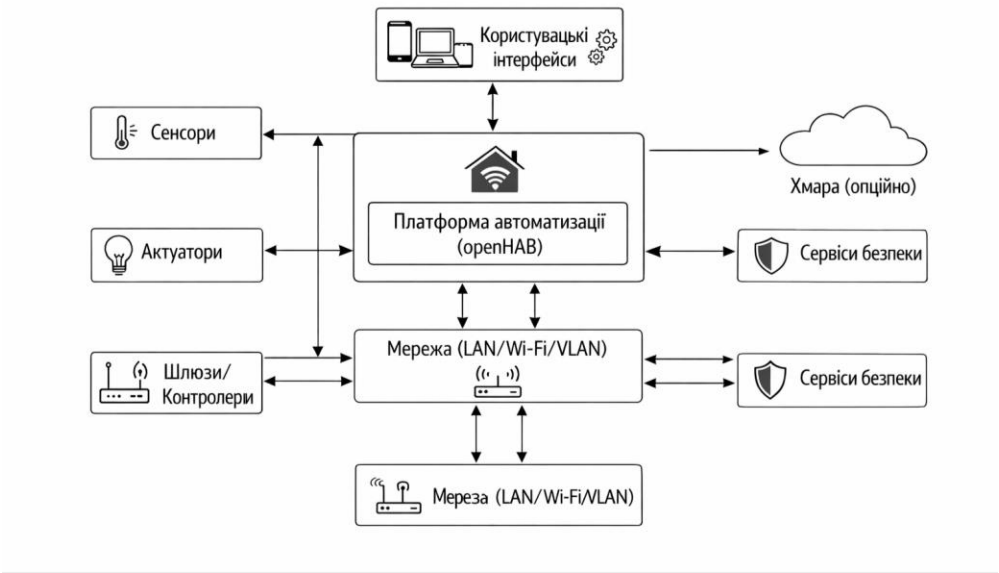

Рисунок 1.1 – Компоненти розумного дому

1.3 Протоколи взаємодії та криптографічні механізми захисту

У домашніх IoT-інфраструктурах найчастіше співіснують кілька моделей обміну даними: (1) «запит-відповідь» (HTTP/REST), (2) «публікація-підписка» (MQTT), (3) полегшені REST-підходи для обмежених пристроїв (CoAP). MQTT визначено як «легкий» протокол клієнт-серверної взаємодії з патерном publish/subscribe, що орієнтований на M2M/IoT та обмежені середовища [9], а його профіль закріплено також на рівні стандарту ISO/IEC 20922 [10].

CoAP описано як спеціалізований веб-протокол передачі для «обмежених вузлів» і «мереж із втратами», де типово є жорсткі обмеження на пам’ять/пропускну здатність [13]. З точки зору проектування домашньої системи це створює можливість будувати енергоефективні датчики (наприклад, на MCU з мінімальними ресурсами) з CoAP-інтерфейсом, одночасно зберігаючи інтеграцію з HTTP-екосистемою через шлюзи.\n\nКриптографічний захист транспортного рівня для HTTP/MQTT традиційно реалізується через TLS. TLS 1.3 прямо визначається як протокол, що запобігає прослуховуванню, підміні та підробці повідомлень у каналі «клієнт-сервер» [11]. Для протоколів на базі UDP (у т.ч. CoAP) застосовують DTLS; DTLS 1.3 визначає аналогічні цілі захисту — конфіденційність і цілісність у небезпечних мережах [12].

На рівні прикладної автентифікації поширеним механізмом є токени. JWT визначається як «компактний, URL-безпечний» формат представлення тверджень (claims), який може бути підписаний або захищений MAC і/або зашифрований, що дозволяє будувати stateless-автентифікацію [14]. У базовому прототипі підкреслено, що JWT підходить для REST-API саме через stateless-природу, зменшуючи потребу зберігати стан сесій на сервері, і тим самим спрощує реалізацію [1].

На рисунку 1.2 наведено модель загроз і довірчі межі.

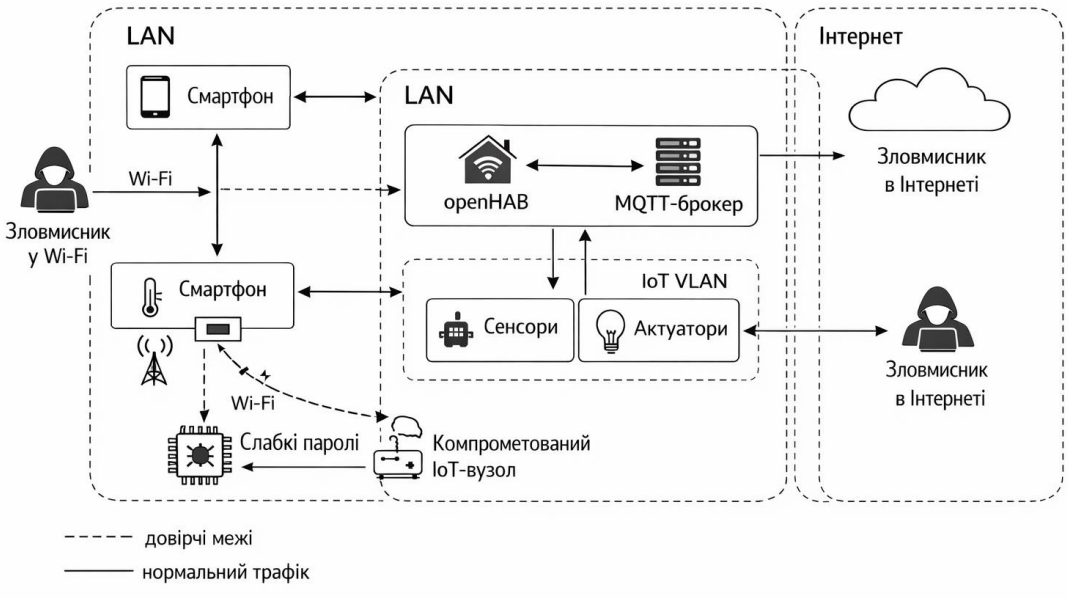

Рисунок 1.2 – Модель загроз і довірчі межі

Практично важливим є і те, що в проаналізованому прототипі дані виявлено такими, що передаються двома шляхами — локально через хост openHAB та через хмарний канал; при цьому зазначено, що частині пристроїв «зазвичай не потрібно» підключення до Інтернету для зміни станів, а Інтернет-доступ стає потрібним переважно для віддаленого керування [1].

Це прямо підтримує аналітичну позицію роботи: локальні сценарії доцільно залишати локальними, а віддалений доступ — реалізовувати через захищені механізми (VPN/тунелювання/керовані проксі-сервіси).

1.4 Аналіз openHAB як платформи інтеграції та автоматизації

openHAB надає структурну модель інтеграції на основі Things/Channels/Items, що дозволяє уніфікувати керування різнорідними пристроями. Things надають канали, які «представляють функції» конкретної фізичної сутності; канали зв’язуються з Items, і вже Items стають доступними для UI та правил [3], [4]. Items визначаються як базові типи даних зі станом, який можна читати/змінювати; при прив’язці до каналу binding’а Item отримує телеметрію з сенсора або надсилає команди актуатору [4].

Для інтеграції із зовнішніми компонентами ключову роль відіграє REST API. Документація прямо фіксує можливість Інтернет-доступу до REST API, але підкреслює «значний ризик безпеки» такого рішення [6]. В окремому розділі щодо безпечного доступу openHAB вказує, що найбезпечнішим варіантом віддаленого доступу «ймовірно» є побудова VPN до домашньої мережі, що дозволяє керувати інстансом так, ніби користувач знаходиться локально [7]. Також існують сервіси на кшталт myopenHAB, які позиціонуються як спосіб «безпечного доступу» без відкриття портів назовні [8], однак у межах цієї роботи такі варіанти доцільно розглядати як додаткові, з чіткою оцінкою довіри до зовнішнього провайдера й моделі загроз.

З позиції аналітичної частини важливо виокремити дві площини безпеки openHAB-системи:

Безпека контуру керування: хто і в який спосіб має право змінювати стан Items/Things, викликати правила, встановлювати конфігурації.

Безпека контуру даних: як захищаються телеметрія та журнали (історія станів, події тривоги), оскільки вони можуть містити чутливу інформацію про побутові звички.

У базовому прототипі ці аспекти підсилено шляхом додавання токенної автентифікації JWT для сценарію «через Інтернет» і шифрування локального бездротового каналу з використанням AES-256 [1].

Аналітично це підводить до висновку, що для магістерського рівня роботи доцільно формалізувати безпекові вимоги системно: не лише «додати шифрування», а побудувати модель доступу, управління ключами/токенами, процедури ротації та політики мінімальних привілеїв.

1.5 Вимоги безпеки та нормативно-методичні орієнтири

Для формування вимог до IoT-системи доцільно опиратися на стандартизовані підходи до оцінювання ризику та на базові каталоги кіберздатностей (capabilities). NIST SP 800-30 визначає ризик-орієнтовану рамку проведення оцінювання ризиків як частини процесу менеджменту ризиків, що забезпечує керівництво інформацією для вибору заходів реагування [16]. Для побудови загроз-орієнтованої частини корисними є підходи threat modeling; NIST SP 800-154 трактує threat modeling як форму оцінки ризику, що моделює «аспекти атаки та захисту» для логічної сутності (даних, застосунку, хоста, системи) [17].

На рисунку 1.3 наведено порівняння MQTT/CoAP/HTTPS

	MQTT	CoAP	HTTPS
Модель взаємодії	Pub/Sub	Запит/відповідь	Запит/відповідь
Накладні витрати	Низькі	Дуже низькі	Високі
Підтримка телеметрії	Так	Так	Обмежена
Підтримка керування	Так	Так	Так
Типовий захист	**TLS 1.3**	DTLS 1.3 / OSCORE	**TLS 1.3**
Плюси	Полегшена для IoT Зберігання стану	Дуже малий трафік Оптимізований для вузлів	Безпека де-факто Web-сумісність
Мінуси	Не оптимізований для вузлів	Складність захисту Затримки зв'язку	Високі накладні витрати Часті перепідключення

------ довірчі межі

——— нормальний трафік

Рисунок 1.3 – Порівняння MQTT/CoAP/HTTPS

На рівні вимог до пристроїв NISTIR 8259A пропонує «core baseline» кіберздатностей IoT-пристрою як стартову точку для організацій при визначенні необхідних функцій безпеки, що підтримують поширені контролі (захист пристроїв, даних і екосистеми) [15]. У межах цієї роботи такі підходи застосовні для:

– формалізації мінімальних вимог до MCU/датчиків (ідентифікація, безпечні оновлення, захист конфігурацій);

– формалізації вимог до шлюзу/серверу openHAB (журналювання, контроль доступу, сегментація, керування секретами);

– визначення вимог до клієнтських застосунків (захист токенів, політики сесій, безпечні канали).

Додатково на міжнародному рівні ISO/IEC 27400 задає загальні настанови щодо ризиків, принципів і контролів для безпеки та приватності IoT-рішень [18]. Для аналітичної частини це дає методологічну «рамку»: вимоги мають бути узгоджені не лише з технічним протоколом (TLS/DTLS), а з повним життєвим циклом — від введення в експлуатацію і керування обліковими даними до інцидент-реакції.

Узагальнюючи, проблема безпеки домашньої IoT-автоматизації для обраної теми формулюється так: потрібно спроєктувати систему на openHAB, яка збереже гнучкість інтеграції (Things/Items/Rules/REST), але мінімізує ризики віддаленого доступу і забезпечить формалізовану модель автентифікації/авторизації та захисту телеметрії. На практиці це означає відмову від «прямої експозиції» керуючих інтерфейсів, орієнтацію на захищені канали (VPN/тунелювання), застосування сучасних криптопримітивів (TLS 1.3 / DTLS 1.3, AES-256 як частина локального контуру) та токен-орієнтованої stateless-автентифікації (JWT) із коректною валідацією підписів і політиками строку дії [11]–[14].

1.6 Висновок до першого розділу

У розділі виконано аналітичне обґрунтування проблеми безпечної домашньої IoT-автоматизації та визначено роль openHAB як інтеграційного ядра системи. Показано, що функціональні переваги платформи (уніфікована модель Things/Channels/Items і рушій правил) поєднуються з необхідністю системного підходу до кіберзахисту, оскільки віддалений доступ до REST API є суттєвим джерелом ризику і потребує захищених каналів та політик доступу. З урахуванням базового прототипу на OpenHAB 2 встановлено, що практично обґрунтованою є стратегія розділення локального та хмарного шляхів даних і застосування різних механізмів захисту для кожного, зокрема JWT для сценарію віддаленого доступу та AES-256 для локального бездротового сегмента.

Результатом аналітичної частини є сформована основа вимог для наступних розділів: (1) вибір моделей взаємодії (publish/subscribe для телеметрії, REST для інтеграції), (2) застосування сучасних криптографічних протоколів (TLS 1.3 / DTLS 1.3), (3) формалізація автентифікації/авторизації на рівні API (JWT) і (4) перехід від «точкових» заходів безпеки до ризик-орієнтованого проєктування за методичними орієнтирами NIST та ISO/IEC.

2 ТЕХНОЛОГІЧНА ЧАСТИНА

Технологічна частина спрямована на обґрунтування вибору програмно-апаратних компонентів IoT-системи «розумного дому» з підтримкою OpenHAB та на визначення способів взаємодії між рівнями «сенсори/актуатори – мережа – інтеграційна платформа – клієнт». У контексті безпеки ключовим є поєднання сумісності (гетерогенність пристроїв і протоколів) із керованими механізмами захисту: від локального шифрування трафіку до контрольованого віддаленого доступу.

Як технологічну основу прийнято референсну архітектуру прототипу, де Raspberry Pi виступає центральним сервером з інстансом OpenHAB і брокером MQTT, а периферійні вузли реалізовано на мікроконтролерній платформі (Arduino) з підключенням ESP8266 та сенсорів/релейних модулів [1].

Така схема є типовою для edge-орієнтованого «розумного дому», де критичні сценарії можуть виконуватися локально, а віддалений доступ будується як окремий, більш суворо контрольований контур.

2.1 Вимоги до системи та критерії вибору технологій

У роботі вихідні вимоги формуються на перетині функціональних задач «розумного дому» та не-функціональних обмежень, притаманних IoT. Функціонально система має підтримувати моніторинг параметрів середовища (температура, вологість, стан навантажень, події тривоги) і керування виконавчими пристроями (світло, реле навантажень, сценарії енергозбереження) через єдину платформу автоматизації. Концепція такого керування прямо корелює з описом домашньої автоматизації як системи керування освітленням, атмосферними умовами, системами спостереження та побутовими приладами з можливістю віддаленого доступу [1].

Не-функціональні вимоги визначають технологічний стек. Для домашнього середовища пріоритетними є: (1) низька латентність реакції на команди (помітна «затримка» знижує довіру користувача до автоматизації), (2)

надійність доставки повідомлень сенсорів і команд керування, (3) масштабованість при додаванні нових пристроїв, (4) доступність реалізації на масовій елементній базі, (5) кіберзахист без критичних накладних витрат на ресурси MCU. Саме останній пункт визначає необхідність розділення контурів доступу: локальний обмін у межах домашньої мережі може мати менший ризик за умови належного захисту Wi-Fi, тоді як Інтернет-доступ створює додаткові загрози перехоплення й підміни трафіку [1].

Рисунок 2.1 відображає цільову архітектуру з openHAB.

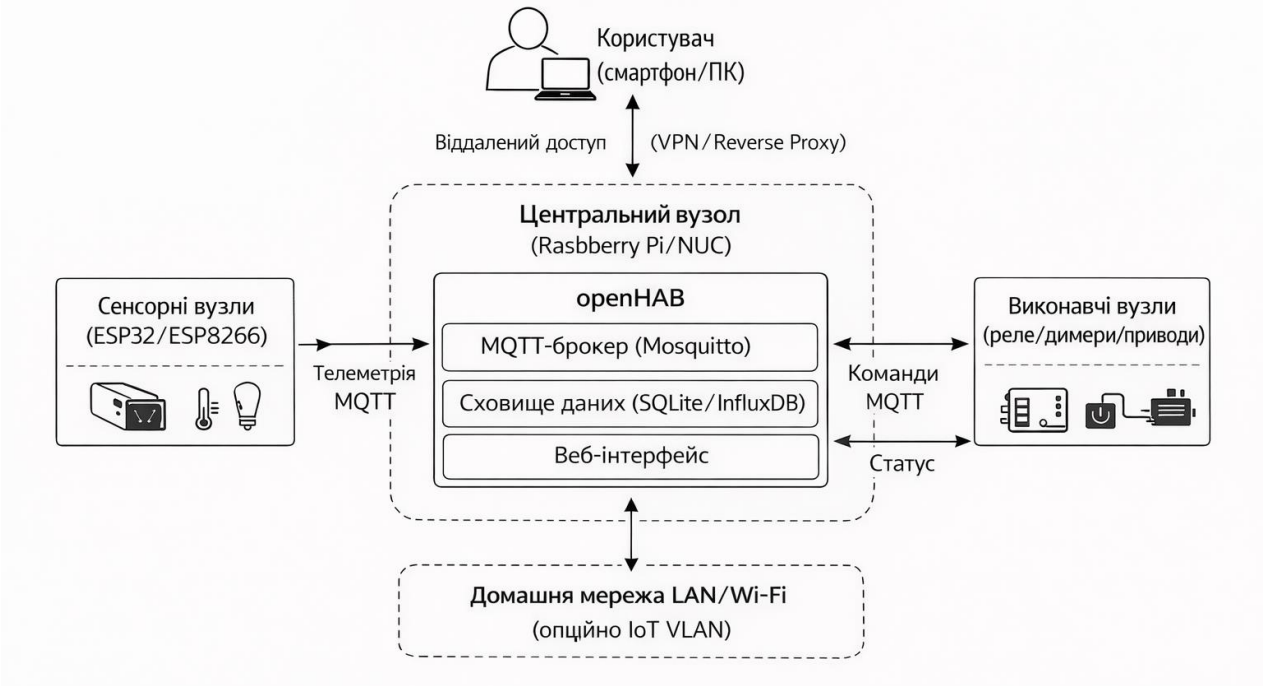

Рисунок 2.1 – Цільова архітектура з openHAB

Для платформи OpenHAB технологічним «обмеженням безпеки» є те, що доступ до REST API через Інтернет можливий, але розглядається як значний ризик; отже, віддалені сценарії мають проєктуватися із застосуванням захищених з’єднань або сервісів контрольованого доступу [6].

Це напряму впливає на вибір технологій периметра (VPN/Reverse Proxy/керовані тунелі), а також на рішення щодо автентифікації й авторизації для віддалених клієнтів.

2.2 Вибір програмних засобів та середовищ розробки

Базова апаратна конфігурація в роботі розглядається як «центральний сервер + периферійні вузли». У прототипі-референсі Raspberry Pi із встановленим OpenHAB функціонує як центральний сервер системи та одночасно хостить MQTT-сервер, через який пристрої обмінюються повідомленнями за моделлю publish/subscribe [1].

Периферійний контур побудовано на Arduino Mega 2560, яка взаємодіє з маршрутизатором через модуль ESP8266; вона «отримує та транслює стан» приладів через MQTT і реалізує фізичне керування через актуатори (релейні схеми, двигуни) та зчитування даних із сенсорів [1].

Рисунок 2.2 відображає потоки даних (телеметрія/керування).

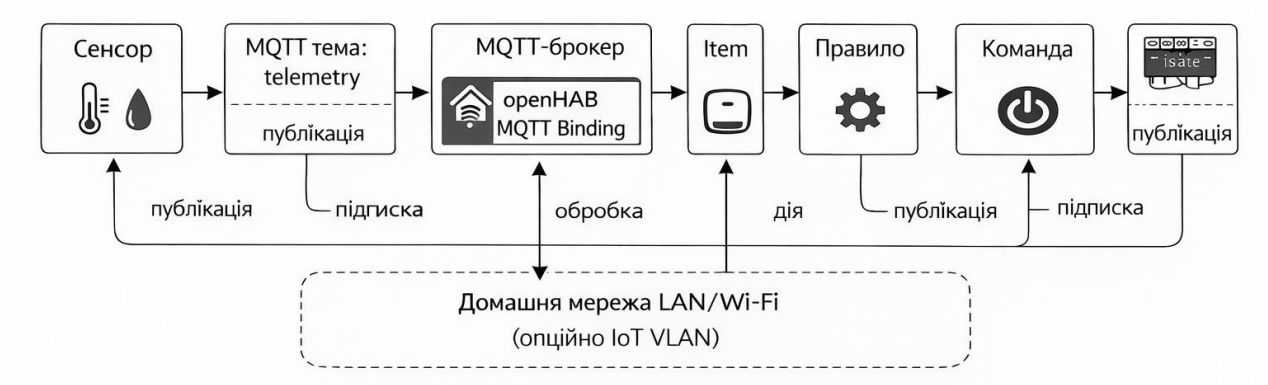

Рисунок 2.2 – Потоки даних (телеметрія/керування)

Вибір Raspberry Pi як серверної платформи технологічно обґрунтовується потребою в постійному виконанні служб OpenHAB, брокера повідомлень та, за потреби, локальної БД/журналювання. Універсальність Linux-оточення на Raspberry Pi також спрощує застосування мережевих засобів захисту (проксі, сертифікати, сегментація), а також централізоване оновлення та контроль конфігурації. Для практичного розгортання OpenHAB у Raspberry Pi-середовищі доцільним є використання openHABian як інсталяційного профілю/образу, орієнтованого на швидке й відтворюване налаштування [19].

Мережева топологія у типовій реалізації зосереджується навколо домашнього маршрутизатора як центральної точки з’єднання: він реєструє IP-

адреси підключених пристроїв і забезпечує їх взаємодію в межах локального сегмента [1].

З погляду стійкості та безпеки доцільно технологічно передбачити: окремий сегмент (VLAN/гостьова мережа) для IoT-вузлів, мінімізацію прямого доступу IoT-пристроїв до Інтернету, а також винесення «керуючих» інтерфейсів у захищений контур доступу. Документація openHAB прямо вказує на корисність розміщення інстансу за reverse proxy (з автентифікацією та сертифікатами), причому openHABian надає можливість активувати попередньо налаштований NGINX reverse proxy з підтримкою валідних сертифікатів Let’s Encrypt [7].

Для віддаленого доступу можливі дві технологічні стратегії. Перша — організація VPN до домашньої мережі й робота з OpenHAB як із локальним ресурсом (це мінімізує експозицію сервісів назовні та зменшує поверхню атаки) [7]. Друга — застосування керованого сервісу віддаленого доступу, який декларує можливість доступу без відкриття портів у Інтернет (наприклад, myopenHAB) [8]. У межах цієї роботи пріоритет надається підходам, де віддалений доступ є контрольованим, із явною автентифікацією та журналюванням.

2.3 Вибір протоколів обміну та механізмів захисту каналів

Для телеметрії та команд керування між вузлами й сервером в обраній архітектурі центральним є MQTT як легкий publish/subscribe протокол, що придатний для M2M та IoT середовищ [9]. Це відповідає реалізації в референсному прототипі, де Raspberry Pi хостить MQTT-сервер, а вузли публікують/підписуються на стани приладів, після чого виконуються автоматизовані або ініційовані користувачем дії [1].

Для інтеграції платформи з клієнтами й зовнішніми компонентами технологічно природним є використання REST API OpenHAB, однак можливість Інтернет-доступу до REST API позначається як значний ризик безпеки, що вимагає додаткових захисних заходів [6]. Звідси випливають дві

вимоги до каналу: по-перше, конфіденційність і цілісність мають забезпечуватися сучасними криптографічними протоколами (TLS 1.3 для TCP-орієнтованих сервісів; DTLS 1.3 — для UDP-орієнтованих сценаріїв) [11], [12]. По-друге, автентифікація та авторизація для віддалених клієнтів має бути формалізована на рівні прикладного протоколу.

У прототипі-референсі це вирішено розділенням двох сценаріїв передавання. Для випадку «через Інтернет» описано ризик перехоплення та підміни пакетів і запропоновано процедуру JWT-автентифікації у вигляді послідовності кроків: генерація токена брокером, додавання до запитів клієнтом, перевірка підпису сервером та контроль терміну дії [1].

У випадку «локальної взаємодії» зазначено, що ризик нижчий за умови шифрування бездротової мережі, і підкреслено застосування AES-256 як практично стійкого механізму шифрування для локального контуру [1].

Важливо, що JWT як технологія визначається стандартом як компактний, URL-безпечний спосіб перенесення тверджень (claims) із можливістю цифрового підпису або MAC/шифрування, що робить її зручною для stateless-сценаріїв [14]. У референсному дослідженні додатково наголошується на stateless-природі JWT як аргументі на користь його застосування для REST API, оскільки це дозволяє уникати збереження стану сесії на сервері [1].

Отже, з технологічної точки зору в межах цієї роботи приймається принцип: локальний обмін — максимально «edge-локалізований» і захищений мережевим шифруванням та сегментацією, а віддалений доступ — лише через контрольовані канали та токен-орієнтований контроль доступу.

2.4. Вибір датчиків, виконавчих модулів і програмних засобів реалізації\nДля «розумного дому» технологічно доцільно стартувати з сенсорів, що дають найбільшу практичну користь при мінімальній складності інтеграції: температура/вологість, контроль стану живлення/навантажень, базові датчики безпеки. У референсному прототипі наведено приклад інтеграції DHT11 як сенсора температури та вологості з підключенням до Arduino, конвертацією значень у середовищі Arduino IDE та публікацією даних через ESP8266 на сервер Raspberry Pi для подальшого зберігання й віддаленого моніторингу [1].

Також описано наявність вузлів вимірювання струму й напруги у схемі та використання реле для керування навантаженнями, що підтримує сценарії енергоменеджменту [1].

На рівні виконавчих механізмів у домашніх системах центральним є релейний комутаційний модуль як «точка підключення» побутових приладів до керуючої логіки [1].

Технологічно це задає два режими експлуатації, релевантні для безпеки й надійності: автоматизований (керування через OpenHAB) та ручний (локальне керування через апаратний контур), що прямо відображено в референсному описі, де OpenHAB відповідає за «remote mode», а окремо розроблена схема комутації підтримує «manual mode» [1].

Така двоконтурність у технологічній частині розглядається як необхідний елемент відмовостійкості: навіть при деградації мережі/сервера базові функції можуть бути відновлені вручну.\n\nЩодо програмних засобів, OpenHAB функціонує як інтеграційний рівень і середовище сценарної логіки, а периферійна частина реалізується у вигляді прошивок MCU. У референсній реалізації задіяно Arduino IDE для взаємодії з сенсорами й підготовки даних, а роль мережевого інтерфейсу MCU виконує ESP8266, який здійснює публікацію даних/отримання команд через MQTT [1].

Для клієнтського рівня передбачено доступ або через веб-інтерфейс, або через спеціалізований Android-застосунок із контролем доступу (email/password), що підтверджує необхідність уніфікованої моделі ідентифікації користувачів [1].

Окремим технологічним аспектом є інструментарій перевірки коректності обраних рішень. У дослідженні базового прототипу наведено підхід, коли для аналізу руху даних та вибору протоколів застосовано моніторинг трафіку Wireshark, що дозволило виявити два шляхи передавання — «через хмару» та «через локальний OpenHAB host» — і відповідно спроєктувати різні механізми захисту для кожного [1].

Це задає методологічну основу технологічної частини поточної роботи: технології обираються не декларативно, а на підставі вимірювань і аналізу

трафіку/поведінки системи в реальній мережі, що надалі буде розвинено у науково-дослідній частині.

Рисунок 2.3 відображає мережеву топологія LAN/IoT VLAN/VPN.

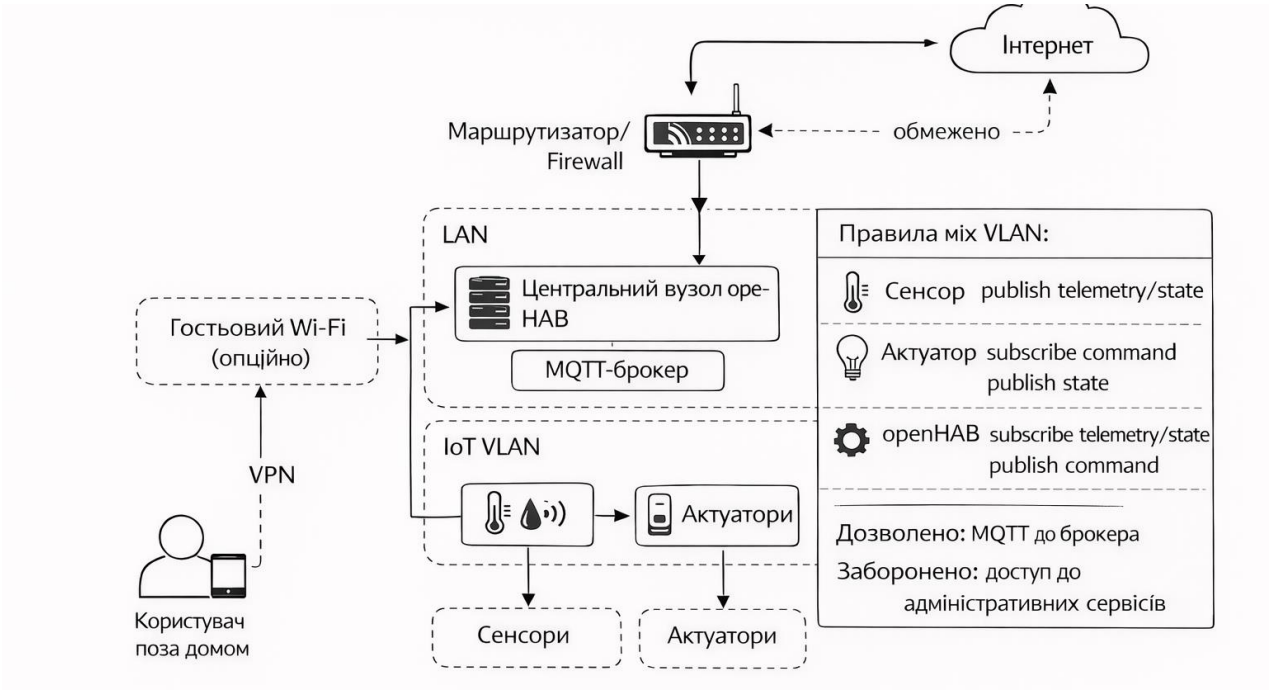

Рисунок 2.3 – Мережева топологія LAN/IoT VLAN/VPN

2.4 Висновок до другого розділу

У технологічній частині обґрунтовано вибір базового програмно-апаратного стеку IoT-системи «розумного дому» з підтримкою OpenHAB. Прийнято архітектуру «центральний сервер (Raspberry Pi + OpenHAB + MQTT) – периферійні вузли (MCU + ESP8266/сенсори/реле) – клієнти (web/mobile)», яка підтверджена референсним прототипом і забезпечує практичну відтворюваність у домашньому середовищі. Транспортною основою для телеметрії та команд керування визначено publish/subscribe модель MQTT як технологічно придатну для IoT, тоді як інтеграційний рівень платформи спирається на OpenHAB-інтерфейси з урахуванням ризиків прямого Інтернет-доступу до REST API.\n\nЗ позиції безпеки зафіксовано технологічний принцип розділення контурів: локальний обмін у межах домашньої мережі має бути захищений належною криптографією/конфігурацією бездротового середовища,

тоді як віддалений доступ потребує контрольованого каналу та прикладної автентифікації/авторизації на базі токенів (JWT). Обрані рішення узгоджуються як із описом референсної реалізації (JWT-процедура для Інтернет-сценарію та AES-256 для локального контуру), так і з сучасними криптографічними протоколами захисту каналів (TLS/DTLS 1.3), що формує технологічну основу для подальшого проєктування конструктивної частини та експериментальної валідації.

3 КОНСТРУКТОРСЬКА ЧАСТИНА

Конструкторська частина визначає практичну форму реалізації запропонованої IoT-системи: склад програмно-апаратних вузлів, їх взаємозв’язки, інтерфейси підключення сенсорів і актуаторів, логіку розміщення в домашній інфраструктурі, а також конструктивні рішення, що забезпечують надійність та безпеку експлуатації. На відміну від технологічної частини, де обґрунтовується вибір стеку, у цьому розділі фіксується саме «як» система буде зібрана, підключена й інтегрована — від топології вузлів до конкретних схем підключення й моделі даних у OpenHAB.

У ролі методологічної основи для опису архітектури доцільно спиратися на еталонні підходи IoT, де виділяються концептуальна модель, референсна модель та архітектурні представлення (views), що дозволяє системно описати компонентний склад і взаємодії між рівнями [26].

3.1 Архітектурна компоновка та функціональні вузли системи

З урахуванням типових IoT-архітектур, описаних у фундаментальних оглядах і візійних роботах, доцільним є поділ системи на рівні: польові пристрої (сенсори/актуатори), мережевий/транспортний рівень, інтеграційний рівень керування (платформа автоматизації), клієнтський рівень (панель керування/мобільний доступ) [23], [24]. Така стратифікація є практичною, оскільки дозволяє ізолювати зміни на рівні фізичних пристроїв від логіки автоматизації, а також відокремити внутрішні процеси будинку від зовнішніх каналів віддаленого доступу.

У запропонованій компоновці виділяються три ключові конструктивні вузли:

Центральний вузол (Home Hub / Edge Gateway).

Це постійно увімкнений обчислювальний вузол (типово одноплатний комп’ютер), на якому розгортається OpenHAB як ядро інтеграції, сервер MQTT

(брокер) і допоміжні служби (журналювання, резервне копіювання, часові сервіси). Він концентрує правила автоматизації, надає уніфіковану модель Items/Things та забезпечує «єдине місце правди» для станів системи, що відповідає концепції уніфікації внутрішніх представлень елементів дому через Items і прив'язку до зовнішнього світу через binding'и [4].

Периферійні вузли (Sensor/Actuator Nodes).

Це мікроконтролерні модулі, які безпосередньо підключаються до сенсорів і актуаторів. Вони виконують зчитування, первинну фільтрацію/валідацію показників, формування повідомлень телеметрії та приймання команд керування.

Саме цей рівень є найбільш чутливим до обмежень ресурсів і фізичних факторів (живлення, завади, умови монтажу), що підкреслюється в оглядах IoT як необхідність враховувати обмеженість вузлів та неоднорідність середовищ [20], [23].

Комунікаційний вузол домашньої мережі (Router/AP + сегментація).

Це інфраструктурний компонент, що фізично з'єднує систему. З конструктивної точки зору він задає розташування вузлів, доступність каналів і можливість сегментації IoT-пристроїв.

Оскільки значна частина ризиків у smart home виникає на межі локальна мережа ↔ Інтернет, конструктивні рішення мають одразу передбачати мінімізацію експозиції керуючих інтерфейсів і можливість відокремлення IoT-сегмента від «користувацького» сегмента (ПК/смартфони).

Описана компоновка узгоджується з практикою побудови прототипів, де центральний вузол на базі Raspberry Pi поєднує OpenHAB та MQTT-брокер, а периферія реалізується на MCU із Wi-Fi-модулем для обміну повідомленнями [1]. Водночас у межах даної роботи цей підхід розвивається саме як конструктивна схема з чіткою декомпозицією функцій і обмеженням доступів між контурами.

3.2 Конструкція периферійного сенсорного вузла

Сенсорний вузол розглядається як модульна конструкція, що складається з: мікроконтролера, інтерфейсних шин, сенсорних модулів, підсистеми живлення, елементів захисту та комунікаційного інтерфейсу. Вибір мікроконтролера визначається двома критеріями: достатність ресурсів для мережевої взаємодії та криптографії, і наявність апаратних інтерфейсів для швидкого підключення датчиків. Огляди IoT підкреслюють, що в гетерогенних середовищах різні вузли мають різні вимоги до енергоспоживання, пам’яті, швидкодії та бездротових інтерфейсів, тому «єдиного універсального» вузла зазвичай не існує [20], [24].

На Рисунку 3.1 наведено структуру сенсорного вузла.

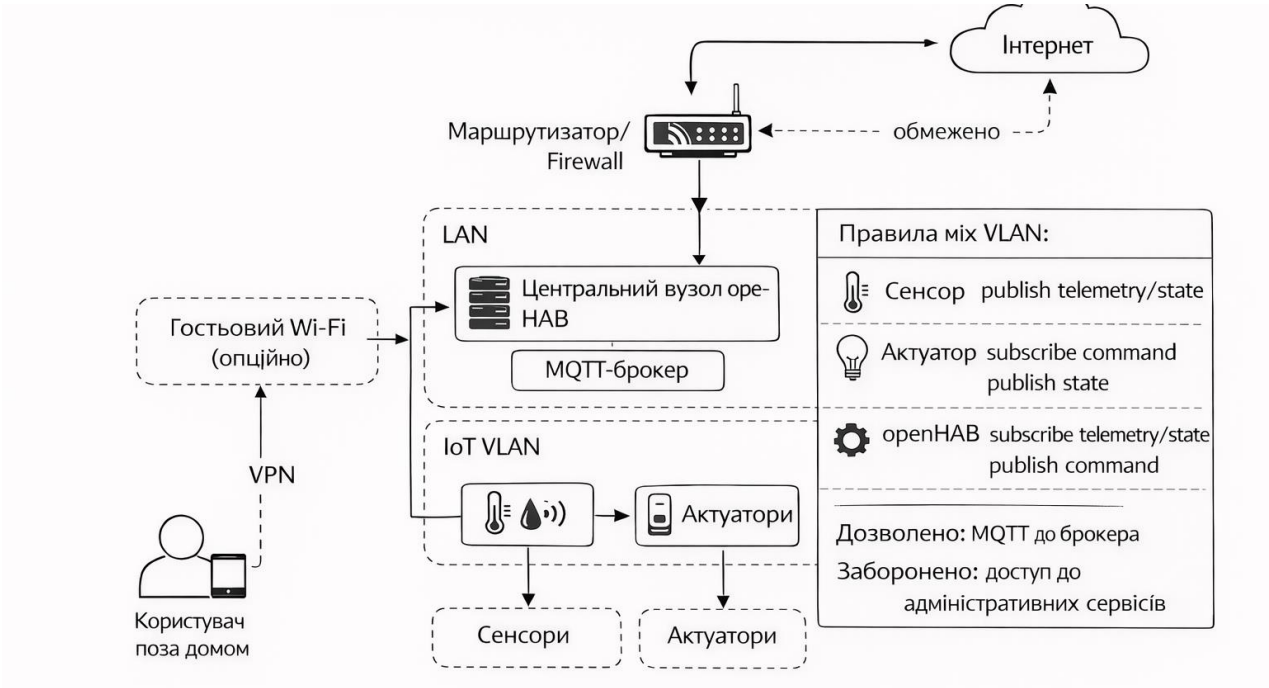

Рисунок 3.1 – Структура сенсорного вузла

У практичній конструкції доцільно передбачити два типи вузлів:

Тип А (телеметричний): сенсори середовища (температура/вологість/тиск), мінімум виконавчих елементів, орієнтація на стабільну телеметрію та низьке енергоспоживання. Як базовий сенсор середовища доцільно застосувати BME280 як комбінований цифровий сенсор

вологості, тиску та температури з компактним корпусом і низьким енергоспоживанням, що спрощує монтаж і дає вищу якість даних для правил OpenHAB порівняно з найпростішими датчиками [29].

Тип B (керуючий): вузол з релейними/транзисторними виходами для керування навантаженнями, часто з локальними захисними умовами (наприклад, «не вмикати нагрівач, якщо температура вище порога», навіть якщо команда надійшла помилково). Цей тип вузлів потребує більш жорстких вимог до живлення, ізоляції й розведення (мала напруга ↔ мережеві 230 В), тому конструктивно має бути відокремлений від чисто сенсорних модулів або винесений у окремий корпус/щит.

На Рисунку 3.2 наведено структуру актуаторного вузла.

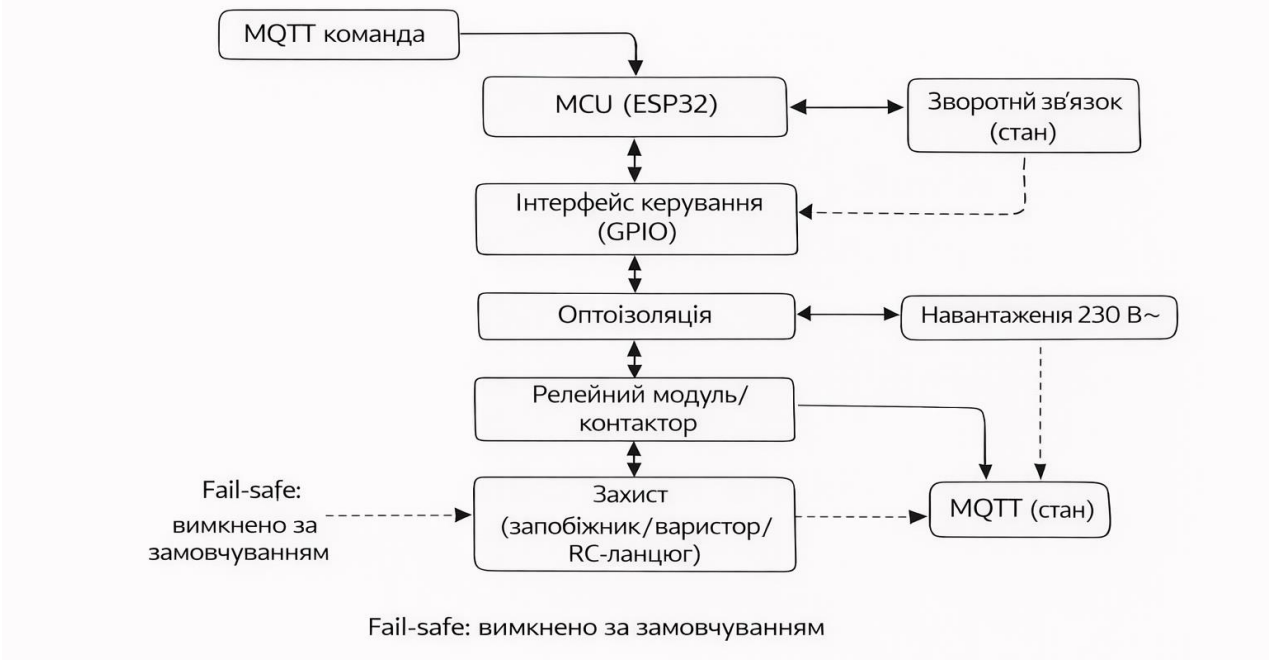

Рисунок 3.2 – Структура актуаторного вузла

Для підключення сенсорів у межах вузла рекомендується застосовувати стандартні низькошвидкісні шини: I²C (BME280 та інші екологічні датчики), 1-Wire (термодатчики на кшталт DS18B20, якщо потрібні довгі лінії), UART/SPI (для зовнішніх модулів чи дисплеїв). Таке рішення зменшує кількість провідників і підвищує повторюваність монтажу.

З конструктивної точки зору важливо передбачити:

- розміщення сенсора так, щоб він вимірював повітря, а не тепло від плати/стабілізатора (фізичне винесення, вентиляційні отвори);
- захист входів від статичних розрядів та наведень (особливо при довгих проводах до датчиків);
- апаратний watchdog/перезапуск для зменшення «зависань» у випадку деградації Wi-Fi або нештатних станів стеку.

3.3 Алгоритмічна та програмна реалізація системи

Виконавчий вузол є критичним з погляду кіберфізичних ризиків: команда з мережі трансформується у фізичний вплив (увімкнення/вимкнення навантаження). Тому конструкторські рішення повинні одночасно забезпечувати електробезпеку та мінімізувати наслідки компрометації.

Конструктивно вузол містить:

- низьковольтну частину (MCU, інтерфейси, логіка, живлення 3.3/5 В);
- силову частину (реле/симісторні ключі/контактори залежно від навантаження);
- межу ізоляції між частинами (оптоізоляція керування, фізичний зазор, трасування плати/використання готових модулів з сертифікацією).

Для керування побутовими навантаженнями в лабораторних і прототипних системах часто використовують релейні модулі, які забезпечують просту дискретну комутацію [1]. Водночас для реального домашнього застосування доцільно конструкторськи передбачити:

- розділення каналів на групи (освітлення/розетки/нагрівальні прилади) з різними правилами безпеки;
- запобіжний елемент у силовому ланцюгу, а також придушення перенапруг (для індуктивних навантажень);
- виконання монтажу в негорючому корпусі з фіксацією провідників і маркуванням.

Особливо важливо, щоб вузол керування реалізовував fail-safe поведінку: при втраті зв’язку або перезавантаженні MCU силові виходи мають переходити в визначений безпечний стан. Це зменшує ризик небезпечних сценаріїв при відмові мережі або при атаках типу відмови в обслуговуванні.

3.4 Модель взаємодії вузлів: MQTT-топіки, QoS і буферизація

З точки зору конструкції системи, протокол — це не лише «вибір технології», а й схема іменування, частота публікацій, наявність retained-повідомлень, QoS-рівні та правила повторної доставки. У publish/subscribe архітектурі MQTT ці рішення безпосередньо визначають надійність моніторингу й керування [9].

Для smart home конструктивно доцільно уніфікувати топіки за принципом:

home/<zone>/<node>/<sensor_or_actuator>/<property>

де <zone> — кімната/поверх, <node> — ідентифікатор вузла, <property> — тип показника або стан каналу. Такий підхід спрощує масштабування й відладку, а також відображення каналів у OpenHAB через MQTT binding.

Вибір QoS для повідомлень має враховувати компроміс «надійність ↔ накладні витрати». Емпіричні порівняння CoAP та MQTT демонструють, що на продуктивність суттєво впливають режими доставки і наявність підтверджень, а також характер трафіку (періодична телеметрія чи подійні повідомлення) [27]. У конструктивному плані це означає: для періодичних екологічних показників достатньо QoS 0 або QoS 1 (залежно від вимог до повноти), тоді як для команд керування та аварійних подій доцільно застосовувати щонайменше QoS 1, а також дублювати критичні стани через retained-повідомлення.

На практиці також варто вбудувати в периферійні вузли невеликий буфер (черга) телеметрії на випадок короткочасних розривів Wi-Fi; після відновлення зв’язку вузол може «догнати» передачу, позначаючи час вимірювання в payload. Така конструктивна деталь підвищує достовірність історії спостережень і підтримує наукову частину (можливість коректного аналізу даних).

3.5 Конструктивна інтеграція з OpenHAB

У OpenHAB конструктивна модель інтеграції реалізується через конфігурацію Things/Channels і зв'язування каналів з Items, що стають базовими сутностями для UI та правил автоматизації [4]. Для MQTT-інтеграції OpenHAB надає окремий binding, який дозволяє налаштовувати підключення до брокера через Things, а також створювати «Generic MQTT Things» з довільною кількістю каналів для підписки/публікації у топіки [30].

Конструкторські це означає, що кожен фізичний вузол системи має мати відповідний «віртуальний двійник» у OpenHAB: MQTT Thing (для вузла) і набір Channels (для сенсорів/виходів). Після цього визначаються Items, які відображаються в інтерфейсі та використовуються у правилах. Власне правила (Rules) є механізмом створення «домашньої автоматизації» на основі тригерів, умов і дій, а текстова DSL-реалізація описується як інтегрований рушій правил, придатний для реальних сценаріїв [5].

На Рисунку 3.3 наведено демонстраційний стенд/макет

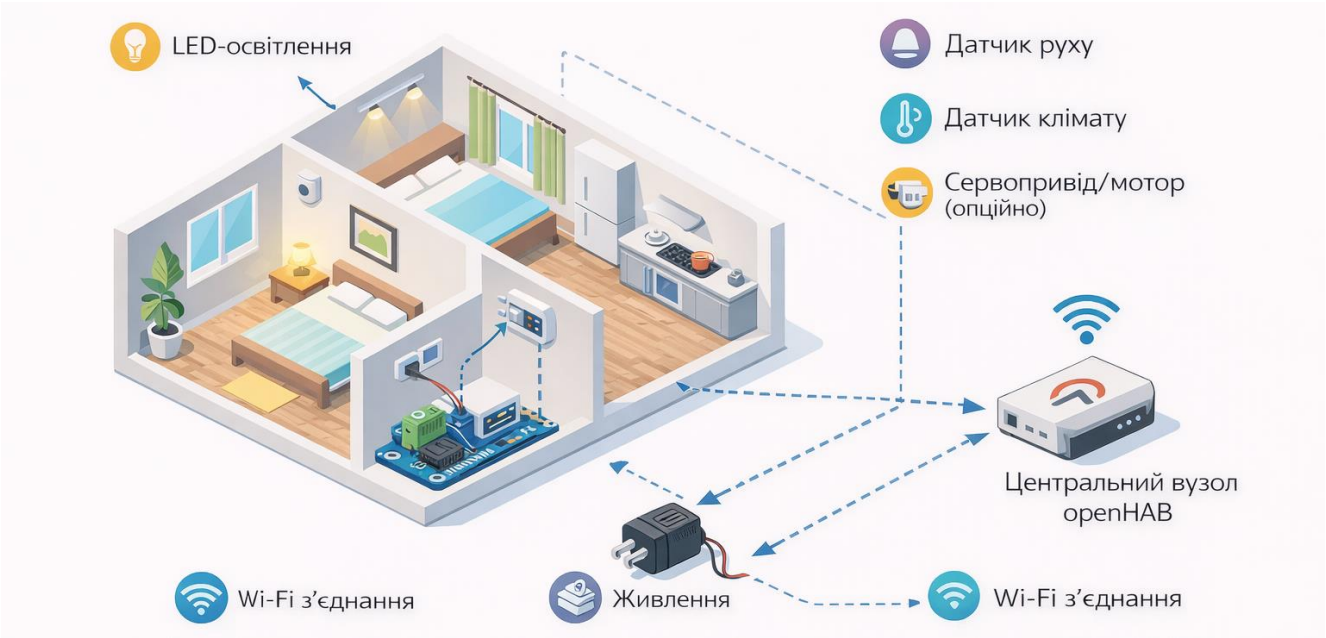

Рисунок 3.3 – Демонстраційний макет

Для забезпечення керованості й безпеки важливо, щоб конструкція правил включала:

- нормалізацію даних (перевірка допустимих діапазонів, відсікання аномалій);
- розмежування команд «користувач» і «автоматизація» (наприклад, через окремі Items або метадані);
- журналювання критичних дій (вмикання силових каналів, зміна режимів охорони).

Такі рішення є відповіддю на типові загрози IoT, де атаки можуть використовувати слабкі місця контролю доступу та неоднорідність реалізацій [21], [22].

3.6 Конструктивні заходи кіберзахисту

Оскільки системи «розумного дому» поєднують приватні дані та фізичне керування, безпека має бути не «додатком», а конструкторською властивістю системи. Оглядові роботи з безпеки IoT підкреслюють, що ключовими проблемами залишаються автентифікація/авторизація, захист даних у транзиті та на пристрої, управління ключами, а також безпечні оновлення [21], [22]. У прикладному вимірі ці вимоги доцільно узгоджувати з базовими «мінімальними положеннями» для споживчих IoT-пристроїв, зокрема забороною універсальних дефолтних паролів, вимогами до безпечних оновлень та захисту даних [25].

У межах запропонованої конструкції доцільно закласти такі рішення:

Захищений транспорт. Для всіх зовнішніх (віддалених) каналів керування — використання захищених з’єднань із сучасними криптографічними протоколами (TLS 1.3 для TCP-сервісів) [11]. При цьому MQTT-обмін доцільно підсилювати TLS-режимом, оскільки сам по собі MQTT не забезпечує криптографічного захисту каналу, а безпека залежить від додаткових механізмів (TLS, керування сертифікатами, політики ACL на брокері).

Керування ідентичностями пристроїв. Кожен периферійний вузол повинен мати унікальну ідентичність на рівні брокера (окремі облікові дані/сертифікат або принаймні окремий логін/пароль) і мінімально необхідні права доступу до топіків. Це підтримує принцип найменших привілеїв і зменшує наслідки компрометації одного вузла.

Безпечні оновлення та конфігурації. Конструктивно система має передбачати механізм оновлення прошивки вузлів (локально або OTA), а також контроль версій конфігурації OpenHAB (резервні копії, відновлення), що узгоджується з базовими вимогами до IoT-пристроїв щодо можливості безпечного оновлення [25].

Сегментація мережі та “зони довіри”. IoT-вузли розміщуються в окремому сегменті (або хоча б окремій Wi-Fi мережі), а OpenHAB-сервер виступає «мостом» між сегментами тільки в необхідних точках. Це зменшує ризик латерального переміщення з компрометованого IoT-вузла на пристрої користувача.

Вказані заходи не суперечать базовій ідеї прототипів з OpenHAB, але переводять рішення в інженерну площину “secure-by-construction”, де безпека підтримується на рівні топології, прав доступу, конфігурацій та процедур експлуатації.

3.7 Висновок до третього розділу

У розділі сформовано конструкторську модель IoT-системи для безпечного моніторингу та керування домашніми пристроями з підтримкою OpenHAB. Запропоновано компоновку з центральним вузлом (OpenHAB + MQTT + служби експлуатації) та периферійними сенсорними/виконавчими вузлами, що підключаються через стандартизовані інтерфейси сенсорів і взаємодіють через MQTT-топіки з визначеними правилами QoS і буферизації. Показано, що конструктивні рішення для виконавчих вузлів повинні поєднувати електробезпеку (ізоляція, розділення силової і логічної частини, fail-safe) та кіберзахист (сегментація, мінімальні привілеї, захищені канали й

оновлення), що відповідає типовим викликам безпеки IoT, описаним у наукових оглядах.

Отримані конструктивні специфікації створюють основу для наступного розділу — науково-дослідної частини, де ці рішення мають бути експериментально верифіковані через вимірювання затримок, надійності доставки та оцінку накладних витрат безпекових механізмів.

4 НАУКОВО ДОСЛІДНА ЧАСТИНА

Науково-дослідна частина спрямована на експериментальну валідацію спроєктованої IoT-системи «розумного дому» з підтримкою OpenHAB із фокусом на безпечну телеметрію та керування. У межах даної роботи дослідження не зводиться до демонстрації працездатності прототипу; натомість розглядаються кілька варіантів експериментів, що перевіряють компроміси між продуктивністю, надійністю доставки, енергетичними витратами та кіберзахистом. Логіка побудови експериментів спирається на відому проблему IoT-систем: посилення безпеки (TLS/DTLS, токени, контроль доступу) часто збільшує накладні витрати, а отже потребує вимірювань і статистично обґрунтованого порівняння [21], [22].

4.1 Мета, дослідницькі питання та гіпотези

Метою експериментального дослідження є кількісно оцінити, як обрані технологічні та конструктивні рішення впливають на ключові експлуатаційні показники системи: затримку керування, надійність доставки повідомлень, споживання ресурсів центральним вузлом і периферійними вузлами, а також стійкість до типових атак у домашньому IoT-середовищі.

Для систематизації дослідження формулюються дослідницькі питання (RQ), орієнтовані на перевірку практичної придатності безпечних режимів:

RQ1 (продуктивність): як змінюються затримка та пропускна здатність при різних режимах MQTT (QoS, розмір payload, частота публікацій) у типових умовах домашньої мережі? Підґрунтям є емпіричні оцінки MQTT на Raspberry Pi як брокері/клієнті, де основними метриками використовуються transmission time та throughput [32].

RQ2 (накладні витрати безпеки каналу): який приріст затримки, навантаження CPU та енергетичних витрат спричиняє застосування MQTT поверх TLS (зокрема TLS 1.3), і як це співвідноситься з альтернативами (CoAP/DTLS, HTTPS) у сценаріях телеметрії та керування? Це питання

мотивоване результатами робіт про помітний вплив TLS на енергоспоживання та продуктивність MQTT, особливо в умовах погіршення мережі [31], а також сучасними вимірюваннями TLS-захищеного MQTT на ARM-пристроях із порівнянням шифросуїт [34].

RQ3 (контроль доступу прикладного рівня): як токен-орієнтована автентифікація (JWT) впливає на час обробки запитів та загальну латентність керування у порівнянні з простішими механізмами, і які політики життєвого циклу токенів (термін дії, оновлення, відкликання) є доцільними для smart home? Питання обґрунтовується тим, що JWT є стандартизованим механізмом перенесення тверджень [14], але практичні реалізації мають вимірювані накладні витрати, які в окремих дослідженнях оцінюються на рівні однозначних відсотків [39].

RQ4 (стійкість): наскільки стійкими є брокер і система в цілому до збоїв/атак (мережеві втрати, спроби несанкціонованої публікації, DDoS-патерни), і як впливають захисні заходи (сегментація/VLAN, ACL топіків, rate limiting, ізоляція доступу) на збереження працездатності? Це питання підтримується сучасними роботами, які прямо досліджують стійкість Mosquitto на Raspberry Pi до «fuzzy DDoS» і демонструють доцільність стрес-тестування брокера як окремого елементу IoT-ланцюга [41].

У межах роботи перевіряються такі гіпотези: (H1) перехід від незахищеного MQTT до MQTT/TLS 1.3 збільшує затримку та ресурсоємність, але залишається прийнятним для домашніх сценаріїв за умови правильного вибору параметрів (QoS, розмір повідомлень, шифросуїта); (H2) CoAP/DTLS може давати порівнювані або кращі характеристики на окремих типах трафіку, але має інші накладні витрати на handshake і складніші аспекти управління безпекою; (H3) JWT додає помірний overhead, який стає суттєвим лише при надто частих операціях керування або неправильній політиці оновлення/відкликання токенів; (H4) керовані захисні заходи на рівні брокера та мережі підвищують стійкість до атак без критичного зниження QoS.

4.2 Експериментальний стенд та відтворюваність

Для забезпечення відтворюваності експериментів стенд проєктується як повторювана конфігурація «центральний вузол + брокер + набір периферійних вузлів + мережевий емулятор умов».

Центральний вузол виконує ролі OpenHAB-ядра та телеметричного/керуючого шлюзу. Брокер MQTT реалізується на Mosquitto, оскільки він є широко використаним у дослідженнях продуктивності MQTT на Raspberry Pi [32]. Периферійний рівень складається з 2–6 вузлів (сенсорних та виконавчих), що генерують періодичну телеметрію (наприклад, 1–5 Гц для екологічних датчиків) та реагують на команди керування.

Критичною передумовою коректного вимірювання латентності є синхронізація часових шкал компонентів. Для цього на центральному вузлі та вузлах, що беруть участь у вимірюванні односторонньої затримки, налаштовується NTPv4 згідно специфікації RFC 5905, яка описує протокол та алгоритми синхронізації [42].

У разі потреби більш високої точності вимірювань допускається розширення методики через PTP-підхід (IEEE 1588) або локальний NTP-сервер, однак у базовому домашньому сценарії NTP зазвичай є практично достатнім для коректних порівнянь між режимами за однакових умов.

Для дослідження впливу деградації мережі застосовується підхід керованого введення втрат/затримок. Аналогічні підходи використовуються у роботах з порівняння CoAP та MQTT із безпекою, де експерименти проводяться в «реалістичному мережевому сценарії» та змінюваних умовах (латентність, пропускна здатність, втрати) [33].

4.3 Метрики, методи вимірювання та критерії якості

Для повноти оцінювання застосовується набір метрик, що описують як транспортні характеристики, так і витрати ресурсів:

Затримка (latency) та її розподіл. Вимірюється енд-ту-енд затримка керування: від моменту формування команди у клієнтському/правиловому контурі OpenHAB до моменту фактичної зміни стану актуатора на периферійному вузлі (підтвердженої відповідним повідомленням/зворотним каналом). Також вимірюється «телеметрична затримка» від моменту зчитування сенсора до появи оновленого стану Item у OpenHAB. Оскільки мережеві затримки в IoT мають асиметрії, а методики одностороннього вимірювання залежать від синхронізації годинників, у протоколі експерименту фіксується, чи використовується одностороння оцінка (OWD) з NTP-синхронізацією або двостороння (RTT) як допоміжна метрика [42].

Таблиця 4.1 План експериментів і метрики оцінювання

Експеримент	**Варіанти/режими**	**Основні метрики**
E1: Латентність керування	MQTT vs MQTT/TLS (QoS 0/1)	P50/P95/P99 затримки, E2E
E2: Надійність доставки	Частота 1–50 msg/s, QoS 0/1	Delivery ratio, повторні доставки
E3: Ресурсні витрати	N вузлів: 5/10/20	CPU/RAM брокера та openHAB

Надійність доставки (delivery ratio) та дублювання. Оцінюється частка доставлених повідомлень/команд у заданому інтервалі часу, кількість дублікативних повідомлень (особливо важливо при QoS 1/2) та порядок доставки. Для smart home практично важливим є також «час відновлення» після короткого розриву зв’язку (time-to-recover), оскільки Wi-Fi деградації є типовими.

Ресурсні витрати на центральному вузлі. Вимірюються CPU, пам’ять, I/O (за потреби) для OpenHAB, брокера та допоміжних служб. Порівняння проводиться для різних режимів безпеки (без TLS, TLS 1.3, різні шифросуїти), оскільки сучасні емпіричні дослідження демонструють суттєву залежність затримки/енергоспоживання від обраної шифросуїти (наприклад, порівняння AES_256_GCM і ChaCha20_Poly1305 на ARMv8-платформах) [34].

Енергетичні витрати периферійних вузлів. Для вузлів на ESP-класі або подібних MCU оцінюється енерговитрата на сеанс зв’язку та на криптографічні операції. Це важливо, оскільки вимірювальні роботи з TLS на ESP32 показують, що вибір криптографічних компонентів протоколу впливає на споживання енергії, а отже і на автономність вузла [37].

Кіберзахист у вимірюваних термінах. Оскільки «безпека» не є лише числом, вводяться операціоналізовані показники: наявність/відсутність можливості пасивного перехоплення вмісту повідомлень (конфіденційність), можливість/неможливість підміни команд без компрометації ключів (цілісність), успішність/неуспішність несанкціонованої публікації/підписки (контроль доступу), збереження працездатності під навантаженням/атаками (availability). Для протоколів застосовуються сучасні рекомендації з використання TLS 1.3 [11] та DTLS 1.3 [12] як базових транспортних механізмів захисту.

На Рисунку 4.1 наведено порівняння режимів захисту каналу

Рисунок 4.1 – Порівняння режимів захисту каналу

4.4 Базова продуктивність MQTT

У першому варіанті дослідження формується базова лінія (baseline) продуктивності системи без увімкнених «важких» безпекових надбудов, щоб мати коректну точку відліку для подальших порівнянь. Експеримент планується як факторний: змінюються QoS (0/1/2), розмір payload (наприклад, 32–256 байт для телеметрії), частота публікацій (1–10 повідомлень/сек на вузол), кількість вузлів (2–10). Основними вихідними величинами є медіана/перцентилі затримки та throughput. Доцільність саме такого дизайну підтверджується тим, що дослідження продуктивності MQTT на різних моделях Raspberry Pi використовують transmission time і throughput як базові метрики та показують, що конфігурація брокера та апаратна платформа істотно впливають на мережеву поведінку [32]. Для OpenHAB-контексту важливим є також відокремлення «затримки протоколу» від «затримки платформи»: OpenHAB додає власний цикл обробки подій (оновлення Items, виконання правил, UI), тому експеримент включає вимірювання на трьох точках: (1) час надходження повідомлення на брокер, (2) час оновлення Item у OpenHAB, (3) час підтвердженого впливу на актуатор. Така декомпозиція дозволяє надалі локалізувати, що саме є «вузьким місцем»: транспорт, брокер, або логіка автоматизації. Очікуваним науковим результатом цього варіанта є побудова емпіричних залежностей «QoS/частота/розмір → затримка/втрата/навантаження», які надалі використовуються як контрольні криві при увімкненні TLS/DTLS та токенів. Це також дозволяє параметризувати правила OpenHAB так, щоб вони не створювали надмірного навантаження (наприклад, відмовитися від зайвої «надчастої» публікації, якщо система не потребує такої дискретності).

4.5 Вплив захисту транспортного рівня

Другий варіант є ключовим для даної роботи, оскільки безпека телеметрії та керування в IoT часто досягається ціною підвищених витрат. Дослідження

проводиться у трьох режимах передавання (за максимально однакових сценаріїв телеметрії/керування):

B1: MQTT без TLS (baseline). B2: MQTT поверх TLS 1.3. B3: CoAP поверх DTLS (для порівняння телеметрії/керування у lightweight UDP-режимі) або HTTPS для REST-викликів керування.

Підстави для такого порівняння є двоякими. По-перше, існують роботи, що спеціально аналізують «performance impact» захисту MQTT через TLS та виявляють, що негативний вплив посилюється при погіршенні мережі (втрати пакетів), що є типовим для Wi-Fi [31]. По-друге, порівняльні дослідження CoAP і MQTT із безпекою на реалістичних сценаріях вимірюють латентність, споживання CPU та пропускну здатність і дозволяють коректно інтерпретувати, де саме виникають накладні витрати при DTLS/захищених режимах [33]. У межах MQTT/TLS піддослідження додатково вводиться фактор «шифросуїта», оскільки сучасні експериментальні роботи на ARM-платформах показують, що вибір між ChaCha20_Poly1305 та AES_256_GCM може змінювати end-to-end latency й енергетичну ефективність залежно від інтерфейсу (Wi-Fi/Ethernet) і розміру повідомлень [34]. Таким чином, окремо тестуються принаймні дві конфігурації TLS 1.3 з різними AEAD-алгоритмами, щоб отримати не загальне твердження «TLS уповільнює», а практичний висновок «який режим доцільніший в умовах нашої системи». Для CoAP/DTLS доцільним є фокус на handshake-витратах та режимах повідомлень confirmable/non-confirmable, оскільки саме вони істотно впливають на латентність і навантаження; прикладом такого підходу є дослідження криптографічної продуктивності CoAP+DTLS на Raspberry Pi, де вимірюються витрати на handshake, шифрування/дешифрування та поведінка CoAP POST у різних режимах [36]. У протоколі експерименту це реалізується через розділення «первинного встановлення захищеного каналу» (cold start) та «стабільного обміну» (steady state). Для smart home це принципово, оскільки багато сценаріїв працюють тривалий час із рідкими reconnect, і в такому режимі handshake-накладні витрати можуть бути менш критичними, ніж накладні витрати на кожне повідомлення. Очікуваний науковий вихід цього варіанта: (1) порівняльні

графіки розподілів затримки (P50/P95/P99) для MQTT/TLS, CoAP/DTLS та (опційно) HTTPS; (2) оцінка приросту CPU/пам’яті на центральному вузлі; (3) оцінка «стабільності» під деградованою мережею (втрати, jitter), яка особливо важлива, оскільки в емпіричних дослідженнях захищений MQTT демонструє різні профілі втрат і повторних передавань залежно від QoS та умов [31].

4.6 Енергетична ціна безпеки на периферійних вузлах

Третій варіант спрямований на дослідження того, що часто ігнорується на рівні «платформних» експериментів: енерговитрати та ресурсоємність криптографії на периферії. Для систем «розумного дому» це має два практичні наслідки: по-перше, автономні вузли (на батареї) можуть різко втрачати ресурс; по-друге, навіть при живленні від мережі підвищена ресурсоємність може збільшувати ймовірність збоїв (watchdog, зависання стека) при перевантаженні.

Методика полягає у вимірюванні струму/потужності MCU-вузла в режимах: (C1) MQTT без TLS, (C2) MQTT/TLS, (C3) CoAP/DTLS (за можливості), при різних частотах передачі та при фіксованому розмірі payload. Результати інтерпретуються через «енергію на доставлене повідомлення» та «енергію на одиницю часу» у steady state. Наукове обґрунтування підходу спирається на роботи, що прямо аналізують споживання енергії TLS-шифрів на ESP32, демонструючи залежність витрат від алгоритмів і компонентів протоколу [37]. Додатково, загальні підходи до бенчмаркінгу TLS у вбудованих/IoT середовищах підтримуються дослідженнями, які пропонують benchmark-підходи для аналізу TLS на IoT-платформах [38]. Очікуваний результат цього варіанта: рекомендації щодо доцільної частоти телеметрії та «пакування» повідомлень, а також практичний вибір режиму захисту для вузлів різного класу. Наприклад, у певних конфігураціях може бути доцільним застосування TLS лише для вузлів, що керують критичними актуаторами, тоді як сенсорні вузли низького ризику працюють у локальному сегменті з меншими накладними витратами, але з посиленими мережевими заходами (сегментація, ізоляція доступу).

4.7 Контроль доступу та токенізація

Четвертий варіант фокусується на прикладному контурі безпеки: автентифікації та авторизації користувача/клієнта для операцій керування (через API або проксі-рівень). У контексті IoT «розумного дому» токенізація є привабливою, оскільки дозволяє реалізувати stateless-контроль доступу та зменшує потребу зберігати сесійний стан на сервері [14]. Разом з тим, JWT має практичні ризики та вимагає дисципліни щодо терміну дії, відкликання та безпечного зберігання на клієнті; окремі праці аналізують сценарії атак та контрзаходи для JWT-потоків [40].

Експериментальна частина включає порівняння трьох підходів керування доступом до віддалених операцій:

- (D1) доступ лише через VPN (тобто API не експонується назовні, а клієнт працює як локальний);
- (D2) доступ через reverse proxy з TLS та Basic/Auth або ключем API (як контрольний простіший режим);
- (D3) доступ через reverse proxy з TLS та JWT (access token з коротким TTL + refresh token або регулярне перевидавання).

Оцінюються метрики: час встановлення «готовності до керування» (login/отримання токена), середня латентність керуючого запиту з урахуванням перевірки токена, overhead на сервері (CPU), а також поведінка при відкликанні/компрометації токена. Практичні дані про накладні витрати JWT можуть бути співставлені з наявними оцінками, де performance overhead використання JWT вимірюється як ненульовий, але помірний у типовому web-потоці [39].

Науковий результат цього варіанта полягає у формалізації «політики токенів» для smart home: рекомендовані TTL, частота оновлення, принципи відкликання. Питання відкликання є важливим, оскільки JWT за своєю природою часто не відкликається «миттєво» без додаткових механізмів, що й відзначається у дослідженнях щодо revocation-механізмів і їхнього впливу на UX/продуктивність [40].

4.8 Стійкість до атак/збоїв та перевірка захисних контурів

П’ятий варіант орієнтований на експериментальну перевірку того, що запропоновані захисні заходи справді змінюють «площину ризику». У smart home типовими загрозами є не лише перехоплення трафіку, а й атаки на брокер/інфраструктуру, несанкціонована публікація команд у топіки, спроби перевантаження сервісів та компрометація слабких облікових даних. Для MQTT-екосистеми існують роботи, що систематизують атаки/вразливості та обговорюють напрями посилення (ACL, TLS, ізоляція, моніторинг) [43].

У протоколі експерименту передбачаються контрольовані сценарії:

- E1: несанкціонована публікація/підписка. Перевіряється, чи здатен сторонній клієнт під’єднатися до брокера та записувати у керуючі топіки або читати чутливу телеметрію. Порівнюється стан «без ACL» та «з ACL/пер-клієнт правами».
- E2: MITM/підміна. Для незахищеного MQTT перевіряється можливість пасивного читання/підміни (в контрольованій лабораторній мережі), тоді як для MQTT/TLS фіксується відсутність читабельного payload та неможливість підміни без компрометації ключів (за коректної валідації сертифікатів).
- E3: деградація мережі як «м’яка атака на доступність». Втрати/затримки інжектуються керовано, і перевіряється, як змінюється затримка керування та частка доставлених повідомлень при QoS 0/1/2 у TLS і без TLS. Доцільність цього сценарію підтверджується тим, що вплив TLS на MQTT є суттєвішим при погіршенні мережевих умов [31].
- E4: стрес/DoS на брокер. Оцінюється стійкість Mosquitto до навантаження, зокрема за патернами, близькими до DDoS. Актуальність такого тесту підкріплюється тим, що існують дослідження, які емпірично аналізують стійкість Mosquitto на Raspberry Pi проти «fuzzy DDoS» та пропонують методи тестування як частину оцінки безпеки MQTT-інфраструктури [41].

Окремий практично корисний аспект цього варіанта — перевірка впливу сегментації (VLAN/ізольований IoT-сегмент) на наслідки компрометації одного

вузла. У працях, що аналізують MQTT security та мережеві конфігурації, підкреслюється, що поєднання апаратних обмежень із VLAN-техніками може розглядатися як засіб зменшення ризиків і стабілізації поведінки під навантаженням [35].

Науковий результат цього варіанта формується як матриця «загроза → контроль → вимірюваний ефект», де поряд із якісним висновком (атака/витік можливі чи ні) фіксуються кількісні наслідки для доступності (скільки падає delivery ratio, як зростає P95 latency, який запас міцності CPU на центральному вузлі).

4.9 Обробка результатів, статистична значущість та валідність

Щоб отримані висновки мали наукову коректність, експерименти проводяться серіями, з повторенням (наприклад, 20–30 запусків на кожну конфігурацію або тривалі прогони по 10–30 хв), а дані аналізуються на рівні розподілів, а не лише середніх значень. Для затримок характерні «важкі хвости», тому практично значущими є перцентилі (P95/P99) та інтерквартильний розмах.

Порівняння режимів може виконуватися як:

- параметрично (t-test/ANOVA) за умови близькості до нормальності або після перетворень,
- непараметрично (Mann–Whitney U / Kruskal–Wallis) при вираженій ненормальності,

із розрахунком довірчих інтервалів для різниці медіан/перцентилів. Для «часток доставлених повідомлень» (delivery ratio) доцільно застосовувати інтервали для пропорцій та порівняння пропорцій між режимами.

Окремо фіксуються загрози валідності: (1) нестабільність Wi-Fi середовища (зовнішні завади), (2) відмінності в реалізаціях клієнтів/бібліотек MQTT/CoAP, (3) вплив фонових процесів на центральному вузлі. Для зменшення цих загроз застосовується контрольований профіль навантаження, фіксація версій, вимірювання системних ресурсів паралельно з мережевими

метриками та порівняння лише на однакових апаратних умовах. Сам підхід «порівняльного аналізу latency/CPU/bandwidth у контрольованих мережевих умовах» узгоджується з методологією досліджень CoAP vs MQTT із безпекою [33].

4.10 Висновок до четвертого розділу

У науково-дослідній частині запропоновано комплексну програму експериментальної валідації IoT-системи «розумного дому» з підтримкою OpenHAB, що охоплює не один, а кілька взаємодоповнюючих варіантів досліджень. По-перше, визначено базову лінію продуктивності MQTT-контуру з урахуванням QoS, частоти телеметрії та масштабування кількості вузлів, що дозволяє коректно порівнювати подальші захищені режими з контрольним станом. По-друге, спроєктовано порівняльне дослідження накладних витрат захисту транспортного рівня (MQTT/TLS 1.3, CoAP/DTLS, опційно HTTPS), де враховано як steady state, так і handshake-впливи, а також вибір шифросуїт, що згідно з сучасними вимірюваннями на ARM-платформах впливає на latency/energy профіль. По-третє, запропоновано окремий енергетичний експеримент на периферійних вузлах, оскільки криптографія може істотно впливати на автономність і стабільність MCU. По-четверте, сформовано методику оцінки токен-орієнтованого контролю доступу (JWT) з урахуванням політик життєвого циклу та відкликання. По-п’яте, заплановано сценарії стійкості до атак/збоїв, включно зі стрес-тестуванням брокера, що відповідає сучасним підходам до емпіричного аналізу безпеки MQTT-екосистеми.

Таким чином, розділ створює науково обґрунтовану основу для подальшого отримання експериментальних даних та їхньої інтерпретації у межах магістерської роботи, забезпечуючи зв’язок між архітектурними рішеннями, механізмами кіберзахисту та кількісними показниками якості обслуговування.

5 СПЕЦІАЛЬНА ЧАСТИНА

Спеціальна частина зосереджується на вимогах і заходах, які забезпечують безпечну експлуатацію розробленої IoT-системи домашнього моніторингу та керування на базі openHAB в умовах реального середовища. На відміну від попередніх розділів, де основний акцент робився на архітектурі, виборі технологій, конструктивних рішеннях і дослідницьких методиках, у цьому розділі системно розглянуто організаційні, нормативні, правові та експлуатаційні аспекти «безпечності» як інтегральної властивості системи: кіберзахист, приватність даних, керування доступом, правила віддаленого доступу, безпечне конфігурування брокера телеметрії та мінімальні вимоги до захисту пристроїв, а також заходи, що знижують ризики при монтажі й експлуатації у житловому середовищі. Для формалізації вимог використано практично орієнтовані базові положення ETSI EN 303 645 та рекомендації NIST SP 800-213, які дозволяють пов’язати обрані технічні рішення з управлінням ризиками на рівні системи [47; 48].

5.1 Нормативно-методична рамка безпеки

Побудова захищеної IoT-системи для житлового середовища потребує відмови від «ад-hoc» підходів, коли безпека додається постфактум, і переходу до моделей, де безпека є вимогою до архітектури та життєвого циклу. У межах цієї роботи доцільно використовувати дві взаємодоповнювальні рамки:

1. ETSI EN 303 645 (споживчий IoT) задає базові положення кібербезпеки для пристроїв та їхніх супутніх сервісів, орієнтуючись на типові «елементарні» слабкості (зокрема легко вгадувані паролі), керування оновленнями, захист даних, розкриття вразливостей і мінімізацію поверхні атаки [47].

2. NIST SP 800-213 розглядає IoT-пристрій як елемент інформаційної системи та пропонує спосіб формування вимог до кіберзахисту «з перспективи

пристрою», але в контексті організаційного управління ризиками, сумісності з контролями та експлуатаційних сценаріїв [48].

Перенесення цих підходів на «розумний дім» із openHAB практично означає: (а) формалізувати активи й загрози, (б) визначити довірчі межі (контролер–LAN–IoT-сегмент–хмара), (в) встановити політики доступу та віддаленої взаємодії, (г) забезпечити контрольовані канали телеметрії та керування, (ґ) узгодити обробку даних із принципами приватності.

5.2 Модель загроз і політика доступу

У домашньому середовищі типова «поверхня атаки» формується не лише з публічних сервісів, а й з «внутрішніх» компонентів: Wi-Fi мережі, мобільних застосунків, локальних API, брокера повідомлень, інтеграцій із хмарними асистентами та оновлень прошивок. Для розробленої системи доцільно визначити політику доступу через тріаду: користувачі, сервісні облікові записи, пристрої.

openHAB як центр керування. У типовій інсталяції openHAB має вбудований вебсервіс і механізми аутентифікації/контролю доступу, а також варіанти безпечного віддаленого доступу (VPN, хмарний сервіс, робота за reverse-proxy) [44].

Критичний нюанс API. За замовчуванням openHAB може надавати права «користувача» для запитів у локальній мережі без явної аутентифікації (через параметр «Implicit User Role»). Документація openHAB прямо вказує, що цю поведінку можна змінити в налаштуваннях безпеки API, і при вимкненні «Implicit User Role» доступ до API вимагатиме автентифікації обліковим записом користувача або адміністратора [45].

З огляду на це, політика доступу в системі повинна включати:

- відмову від неавтентифікованого доступу до API, якщо система інтегрується з будь-якими зовнішніми компонентами або використовується сегментація мережі (щоб унеможливити «перетікання» довіри між сегментами);

- мінімізацію прав сервісних облікових записів: інтеграції (боти, сценарії, шлюзи) не повинні використовувати адміністративні облікові дані;
- розділення контурів «керування» і «моніторингу»: наприклад, читання телеметрії може бути дозволене ширше, а команди керування — лише після додаткової перевірки (роль/2FA на рівні зовнішнього доступу).

У прикладному вимірі це означає, що вразливості «локального» характеру (небезпечно налаштований Wi-Fi, компрометація смартфона, гостьовий доступ) можуть швидко ескалювати до можливості керування реле, замками чи системами мікроклімату. Саме тому в межах ETSI EN 303 645 особлива увага приділяється уникненню тривіальних конфігурацій і керованості доступів, а NIST SP 800-213 — зв'язку таких конфігурацій із ризиками системи [47; 48].

5.3 Захищений віддалений доступ

Віддалений доступ є одним із найризиковіших сценаріїв для «розумного дому», оскільки навіть коректно реалізована телеметрія втрачає цінність у разі компрометації каналу керування або облікового запису.

У документації openHAB наголошено, що доступ можна організовувати через VPN, хмарний сервіс, або reverse proxy, причому reverse proxy також може бути використаний як шар автентифікації та керування сертифікатами [44].

З погляду «мінімального експонування» для розробленої системи приймається така пріоритизація:

1. VPN як основний механізм (найменша площина атаки за умови закритих портів openHAB у публічний Інтернет і доступу лише через тунель).
2. Reverse proxy лише за потреби, із обов'язковою додатковою автентифікацією на периметрі, строгими TLS-налаштуваннями та обмеженням доступних шляхів (наприклад, блокування або додаткова авторизація для адміністративних endpoint'ів). openHAB також прямо зазначає, що reverse proxy може надати простіший шлях до коректних сертифікатів і захисту доступу [44].

3. Хмарні сценарії (за необхідності інтеграцій із зовнішніми асистентами) — із суворим контролем токенів, сегментацією доступу та журналюванням дій.

Ключовим принципом тут є не «максимальна зручність», а контрольованість: для будь-якого віддаленого доступу мають існувати механізми відкликання (revocation), ротації секретів і локального відновлення контролю над системою.

5.4 Захищена телеметрія та керування через MQTT

У межах розробленої системи MQTT є практичним вибором для телеметрії та команд керування (особливо для енергоощадних вузлів). Проте саме MQTT-контур часто стає «слабкою ланкою», якщо брокер розгорнутий із типовими налаштуваннями та без обмежень тем (topic-ACL).

Базові вимоги до Mosquitto. Для брокера Mosquitto критично забезпечити одночасно: (а) шифрування, (б) аутентифікацію клієнтів, (в) авторизацію доступу до тем. Механізм TLS із сертифікатною аутентифікацією підтримується через параметр require_certificate: якщо він увімкнений, клієнт має надати валідний сертифікат для успішного підключення; якщо вимкнений — TLS може захищати лише канал, а автентифікація обмежується логіном/паролем [46].

Авторизація за темами. Mosquitto підтримує підключення ACL-файлу (acl_file), який керує доступом клієнтів до публікації/підписки на конкретні теми [46].

Для «розумного дому» це означає, що сенсор не повинен мати можливості публікувати у теми керування реле, а актуатор — читати теми з «чутливою» телеметрією (наприклад, присутність/розклад). Така політика знижує ризик ескалації при компрометації одного вузла: інцидент локалізується в межах його тем.

Рекомендований профіль конфігурації MQTT-контуру (на рівні принципів):

- використання TLS-листенера для зовнішніх/міжсегментних підключень;
- заборона анонімних клієнтів, використання окремих облікових даних або сертифікатів на клас пристроїв;
- ACL за принципом «deny-by-default» із явним дозволом лише потрібних тем;
- контроль імен клієнтів (clientId) та відмова від дублювання ідентичностей;
- окремі теми для: телеметрії, команд, статусів, службових повідомлень, журналів.

Технічно це узгоджується і з ETSI EN 303 645 (керованість доступів і мінімізація тривіальних слабкостей), і з NIST SP 800-213 (визначення вимог до кіберзахисту IoT-вузлів як елементів системи) [47; 48].

5.5 Приватність даних і правові вимоги

Домашня IoT-система оперує даними, які легко перетворюються на профіль поведінки мешканців: часові ряди температури/вологості, споживання енергії, події руху, відкриття дверей, сценарії присутності. Навіть якщо система не збирає «паспортних» даних, комбінація подій ідентифікує людину опосередковано.

Принципи обробки (GDPR). Стаття 5 GDPR визначає ключові принципи: законність/прозорість, цільове обмеження, мінімізацію даних, точність, обмеження зберігання, цілісність і конфіденційність [49].

Українське регулювання. Закон України «Про захист персональних даних» встановлює, що він регулює правовідносини, пов’язані із захистом і обробкою персональних даних та спрямований на захист прав, зокрема права на приватність у зв’язку з обробкою персональних даних [50].

З практичної точки зору, для запропонованої IoT-системи це трансформується у набір інженерних рішень приватності:

- мінімізація збору: збирати лише те, що необхідно для сценаріїв керування (наприклад, замість сирого потоку подій руху — агреговані стани «рух виявлено/не виявлено» у вікні часу);
- локальність обробки: перевага локального виконання правил openHAB і локального зберігання журналів над відправкою у зовнішні сервіси;
- контроль строків зберігання: телеметрія високої деталізації зберігається коротко (для діагностики), а статистика — довше, але без надлишкової гранулярності;
- обмеження доступу та аудит: доступ до історії та журналів має бути розділений за ролями, а критичні дії (зміни правил, доступ до «сигналів присутності», інтеграції з хмарою) — фіксуватися.

У роботі ці принципи розглядаються не як «юридичний додаток», а як вимоги до архітектури даних: якщо дані не збираються або агрегуються на вході, ризики приватності зменшуються «структурно», і система стає стійкішою до витоків.

5.6 Експлуатаційна безпека в житловому середовищі

Навіть «ідеально» захищена з кіберпогляду IoT-система може створювати фізичні ризики, якщо її монтаж порушує правила електробезпеки або якщо виконавчі модулі (реле, контактори) підключені без захисних елементів. У рамках цієї роботи спеціальна частина фіксує інженерні правила, які мають виконуватися при інтеграції системи з побутовими електромережами.

Електрозахист і УЗО (RCD). У практиці житлових установок застосування пристроїв захисного вимкнення (RCD) розглядається як ефективний засіб захисту людей і майна від електричних небезпек, а IEC 60364-орієнтовані підходи широко вимагають додаткового захисту через RCD для певних груп кіл у житлі [практичні огляди, узгоджені з IEC-підходами].

Для IoT-виконавчих модулів це означає:

- використання сертифікованих реле/контакторів із достатнім запасом по струму;
- розділення слабкострумової частини (3.3/5/12/24 В) і силової частини (230 В), дотримання відстаней та ізоляції;
- захист від перегріву (корпуси, монтаж на DIN-рейку, вентиляція), коректне під’єднання PE/заземлення там, де це потрібно;
- відмова від «саморобних» підключень до мережі без належної кваліфікації.

Експлуатаційний контроль і профілактика. Оскільки IoT-контур змінює «поведінку» навантажень (частота комутацій, сценарії керування), важливо впровадити профілактичні процедури: періодичну перевірку контактів, оцінку нагріву виконавчих модулів у типових режимах, контроль стану джерел живлення та резервування критичних функцій (наприклад, аварійне вимкнення реле).

5.7 Набір практичних заходів для системи

Для того щоб спеціальна частина мала не лише описовий, а й прикладний характер, сформовано узгоджений набір заходів, який може бути використаний як «контрольний лист» під час впровадження.

Контур openHAB (периметр і доступ):

- застосовувати шифровані канали доступу та не залишати відкритими інтерфейси керування без потреби; openHAB описує підходи до захищеного доступу та варіанти безпечного віддаленого підключення [44].
- вимкнути неавтентифікований доступ до API через налаштування «Implicit User Role», якщо система виходить за межі простого локального сценарію або використовується сегментація/віддалений доступ [45].

Контур MQTT (Mosquitto) (телеметрія/керування):

- увімкнути TLS та, за можливості, застосувати сертифікатну автентифікацію, де require_certificate=true вимагає сертифікат клієнта [46].

- налаштувати авторизацію за темами через acl_file із політикою «дозволено лише необхідне» [46].

Рамка вимог (ETSI/NIST) (керованість і ризики):

- уникати «легко вгадуваних» конфігурацій доступу й забезпечувати керованість безпеки як частину життєвого циклу пристроїв і сервісів (оновлення, контроль доступів, захист даних) [47].
- формувати вимоги до IoT-вузлів як до елементів системи з огляду на ризики інтеграції та експлуатації [48].

Приватність і відповідність:

- впровадити мінімізацію збору та строків зберігання, а також захист цілісності й конфіденційності даних відповідно до принципів GDPR [49] та норм українського законодавства про персональні дані [50].

5.8 Висновок до п’ятого розділу

У спеціальній частині обґрунтовано, що безпека IoT-системи для «розумного дому» на базі openHAB має трактуватися як сукупність взаємопов’язаних вимог до архітектури, конфігурації, експлуатації та правомірної обробки даних. На основі ETSI EN 303 645 і NIST SP 800-213 формалізовано підхід, у якому захищеність визначається не окремою технологією (наприклад, TLS), а керованою системою контролів: політиками доступу до openHAB, безпечним віддаленим доступом, захистом контурів телеметрії/керування через MQTT із автентифікацією та ACL, а також принципами приватності та відповідності правовим нормам. Показано, що openHAB-специфічні параметри (зокрема режим доступу до REST API) суттєво впливають на профіль ризиків і повинні налаштовуватися свідомо відповідно до моделі загроз. Окремо підкреслено, що безпечність у житловому середовищі включає і фізичні аспекти (електробезпека, правила монтажу виконавчих модулів), оскільки IoT-керування змінює режими роботи навантажень і підвищує вимоги до дисципліни експлуатації.

6 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

6.1 Питання щодо охорони праці

Тема кваліфікаційної роботи освітнього рівня «Магістр» зосереджена на розробленні та дослідженні IoT-системи для безпечного моніторингу та керування домашніми пристроями з підтримкою OpenHAB.

Доцільно проаналізувати загальні вимоги безпеки, що висуваються до виробничого обладнання та технологічних процесів.

Ключовими складовими безпеки праці на виробництві є:

• безпечне виробниче обладнання;

• безпечні технологічні процеси;

• організація безпечного виконання робіт.

ГОСТ 12.2.003–91 є базовим нормативним документом щодо загальних вимог безпеки до виробничого обладнання (за винятком обладнання, яке є джерелом іонізуючих випромінювань).

Вимоги безпеки до конкретних груп, типів і моделей виробничого обладнання формуються відповідно до ГОСТ 12.2.003–91 з урахуванням його призначення, виконання та умов експлуатації.

Безпечність виробничого обладнання забезпечується:

• вибором принципу дії, джерел енергії та параметрів робочих процесів;

• зменшенням (мінімізацією) енергії, що споживається або накопичується;

• застосуванням вбудованих у конструкцію засобів захисту та інформування про потенційно небезпечні ситуації;

• використанням засобів автоматизації, дистанційного керування і контролю;

• дотриманням ергономічних вимог, а також обмеженням фізичних і нервово-психологічних навантажень працівників.

Виробниче обладнання, як при автономній роботі, так і у складі технологічних комплексів, має відповідати вимогам безпеки протягом усього строку експлуатації.

Обладнання, експлуатація якого супроводжується виділенням шкідливих речовин чи мікроорганізмів або пожежо- та вибухонебезпечних речовин, повинно містити вбудовані пристрої для локалізації таких виділень. Якщо вбудованих пристроїв немає, конструкцією мають бути передбачені місця для приєднання автономних засобів локалізації. За потреби ці пристрої виконують із дотриманням чинних вимог щодо стану повітря робочої зони та охорони довкілля.

Якщо обладнання є джерелом шуму, ультра- та інфразвуку, вібрації, виробничих випромінювань (електромагнітних, лазерних тощо), його слід проєктувати так, щоб вплив перелічених шкідливих факторів на працівників не перевищував граничних значень, установлених відповідними нормативами.

Важливою складовою безпеки обладнання є також конструкція робочого місця: його габарити, взаємне розташування органів керування, засобів відображення інформації, допоміжного устаткування тощо. Під час розроблення робочого місця необхідно виконувати вимоги ГОСТ 12.2.032–78, ГОСТ12.2.033–84, ГОСТ12.2.049–80, ГОСТ12.2.061–81 та інших чинних нормативів. При цьому розміри робочого місця та його елементів мають забезпечувати виконання операцій у зручних робочих позах і не обмежувати рухи працівника. Конструкція крісла і підставки для ніг повинна відповідати ергономічним вимогам.

Повне або часткове припинення енергопостачання з подальшим відновленням, а також пошкодження мережі керування енергопостачанням не повинні призводити до виникнення небезпечних ситуацій.

Засоби захисту, інтегровані в конструкцію виробничого обладнання, повинні: забезпечувати можливість контролю їх працездатності; виконувати призначення безперервно під час роботи обладнання; діяти до повної нормалізації небезпечного або шкідливого фактора, що спричинив спрацювання; зберігати працездатність у разі відмови інших засобів захисту.

Якщо перед початком роботи обладнання потрібно активувати засоби захисту, схема керування має передбачати відповідні блокування тощо.

ГОСТ 12.3.002–75 є чинним нормативним документом щодо загальних вимог безпеки виробничих процесів.

Безпека виробничих процесів насамперед визначається безпекою обладнання. Вона забезпечується врахуванням вимог безпеки під час формування технічного завдання на проєктування, розроблення ескізного і робочого проєкту, виготовлення та випробування дослідного зразка, а також під час передавання у серійне виробництво згідно з ГОСТ 15.001—73 «Розробка і постановка продукції на виробництво. Основні положення».

Основні вимоги безпеки до технологічних процесів передбачають: усунення прямого контакту працівників із вихідними матеріалами, заготовками, напівфабрикатами, готовою продукцією та відходами, які можуть бути джерелами небезпеки; заміну процесів і операцій, пов’язаних із виникненням небезпечних і шкідливих виробничих факторів, на такі, де ці фактори відсутні або мають меншу інтенсивність; комплексну механізацію та автоматизацію виробництва, застосування дистанційного керування процесами й операціями за наявності небезпечних та шкідливих факторів; герметизацію обладнання; використання засобів колективного захисту; раціональну організацію праці й відпочинку для профілактики монотонності та гіподинамії, а також обмеження важкості праці; своєчасне отримання інформації про появу небезпечних і шкідливих факторів на окремих операціях (системи такого інформування доцільно реалізовувати за принципом автоматичної дії з виведенням на попереджувальну сигналізацію); упровадження систем контролю та керування технологічним процесом, які забезпечують захист працівників і аварійне вимкнення обладнання; своєчасне видалення та знешкодження відходів, що є джерелами небезпечних і шкідливих факторів, а також забезпечення пожежної і вибухової безпеки.

Розміщення виробничого обладнання, вихідних матеріалів, заготовок, напівфабрикатів, готової продукції та відходів у виробничих приміщеннях і на робочих місцях не повинно створювати небезпеку для персоналу. Відстані між

одиницями обладнання, а також між обладнанням і стінами приміщень, будівель та споруд мають відповідати вимогам чинних норм технологічного проєктування, будівельним нормам і правилам.

До чинників, що формують умови праці, належать також раціональні технологічні методи й організація виробництва. Зокрема, істотне значення мають зміст праці, форми побудови трудових процесів, рівень спеціалізації працівників під час виконання виробничих операцій, вибір режимів праці та відпочинку, трудова дисципліна, психологічний клімат у колективі, організація санітарного і побутового забезпечення працівників відповідно до чинних вимог.

До осіб, допущених до участі у виробничому процесі, висуваються вимоги щодо відповідності їхніх фізичних, психофізіологічних, а в окремих випадках і антропометричних даних характеру роботи. Контроль стану здоров’я працівників має здійснюватися при допуску до роботи та періодично відповідно до чинних нормативів. Періодичність таких оглядів визначають залежно від небезпечних і шкідливих факторів виробничого процесу в порядку, установленому Міністерством охорони здоров’я.

В умовах автоматизованого виробництва необхідним є також суворе дотримання вимог безпеки під час ремонту та налагодження автоматичних машин і їхніх систем. Одним із перспективних напрямів комплексної автоматизації виробничих процесів є використання промислових роботів..

6.2 Питання щодо безпеки в надзвичайних ситуаціях

Тема кваліфікаційної роботи освітнього присвячена підвищенню рівня повноти та достовірності подання інформації щодо IoT-системи для безпечного моніторингу та керування домашніми пристроями з підтримкою OpenHAB, тому доцільно розглянути питання з вимог електробезпеки до приміщень з ЕОМ.

• Характерною рисою сучасного етапу науково-технічного прогресу є формування автоматизованих систем управління на основі широкого впровадження електронно-обчислювальної техніки, що зумовлює такі наслідки:

• зростання кількості об’єктів і параметрів, які необхідно контролювати та регулювати;

• розвиток дистанційного керування об’єктами, динаміку яких визначають за сигналами, що надходять через засоби відображення інформації;

• ускладнення технологічних процесів і підвищення швидкості їх перебігу;

• трансформацію умов праці;

• зміну структури та змісту трудової діяльності, появу професії оператора, який здійснює управління, налагодження та контроль функціонування різних автоматизованих систем;

• створення різноманітних систем типу «людина–машина», робота яких залежить від узгодженої взаємодії технічних засобів і діяльності людини в межах єдиного контуру регулювання.

Під системою «людина—машина» розуміють комплекс, що включає людину-оператора (або групу операторів) та сукупність технічних засобів, за допомогою яких виконується трудова діяльність.

У певних ситуаціях можливості людини суттєво перевищують можливості машини, в інших — навпаки, технічні засоби мають переваги над людиною. Зокрема, людина здатна діяти в непередбачуваних умовах, вирізняється гнучкістю та адаптивністю до змін зовнішнього середовища, може працювати за різними програмами. Для машини практично неможливо заздалегідь описати всі випадкові події; їй властиві обмежена гнучкість і висока вартість багатоваріантних режимів роботи. Людина може формувати цілісне уявлення про явища й події за неповної інформації, тоді як у машини такі можливості значно обмежені. Так само людина має ширший вибір способів дій, здатна оперативно використовувати резерви та виправляти помилки, тоді як у технічних засобів цей потенціал обмежений. Водночас для людини характерна обмежена «пропускна здатність» у сприйманні та обробці інформації, тоді як у машини вона, як правило, вища. Працездатність людини знижується через втому, розсіювання уваги, виникнення різних емоційних станів і переживань, тоді як машина здебільшого зберігає стабільну роботоздатність. Крім того,

людина порівняно повільно і менш точно виконує обчислювальні операції, тоді як машина забезпечує високу швидкість і точність розрахунків. У зв’язку з цим доцільною є автоматизація тих функцій, які можуть виконуватися без участі людини. За людиною переважно залишаються функції прийняття рішень, що й визначає підвищення ефективності праці мислячого оператора порівняно з роботою суто автоматичних пристроїв. Саме оператор посідає центральне місце у великих системах управління.

Взаємодія людини з технічними засобами потребує підвищених вимог, що передбачає, з одного боку, адаптацію техніки до можливостей людини (проєктування машин з урахуванням людських характеристик), а з іншого — адаптацію людини до машини (відбір і підготовку фахівців). Зазначені особливості операторської праці дають підстави розглядати її як специфічний вид професійної діяльності, основні етапи якої можна подати так.

— Сприймання інформації про об’єкти керування та навколишнє середовище, важливої для розв’язання завдань, поставлених перед системою «людина-машина». Для цього оператор має помітити сигнали, серед їх сукупності виокремити найсуттєвіші та здійснити розшифрування. У результаті формується початкове уявлення про стан керованого об’єкта. Якість сприймання визначається типом і кількістю індикаторів, організацією інформаційного поля та характеристиками поданої інформації.

— Оцінювання та переробка інформації. На цьому етапі зіставляються задані та фактичні режими роботи системи, здійснюються аналіз і узагальнення даних, виділяються критичні об’єкти й ситуації. Відповідно до критеріїв важливості та терміновості визначається черговість обробки повідомлень. На результат впливають спосіб кодування, обсяг інформації та динаміка змін у системі, а також відповідність інформаційного навантаження можливостям пам’яті й мислення оператора.

— Прийняття рішення щодо необхідних дій на основі проведеного аналізу й оцінки інформації, а також з урахуванням мети та умов роботи системи, можливих способів дії й наслідків правильних або помилкових рішень.

Ефективність рішення визначається типом задачі, складністю логічних умов, алгоритмом дій і кількістю можливих варіантів.

— Реалізація прийнятого рішення шляхом виконання конкретних дій або надання відповідних команд. На цьому етапі важливими операціями є перекодування рішення в машинний формат, пошук потрібного органу керування, виконання маніпуляцій з ним тощо. Якість виконання рухових дій залежить від кількості органів керування, їх типу та принципів розміщення.

На кожному з етапів необхідний контроль власних дій, який може здійснюватися інструментально або візуально, що підвищує надійність роботи оператора. Перші два етапи розглядають як отримання інформації, а наступні два — як її реалізацію. Отримання інформації відбувається через сприймання оператором інформаційної моделі об’єкта керування, тобто різних носіїв інформації. Після декодування сигналів формується логічне знання про керований процес — так звана концептуальна модель. Концептуальна модель дає змогу операторові інтегрувати окремі елементи керованого процесу в єдину картину і, спираючись на прийняте рішення, здійснювати ефективні керівні впливи, тобто практично реалізовувати отриману інформацію..

6.3 Висновки до шостого розділу

Розроблення магістерської кваліфікаційної роботи, присвяченої IoT-системі безпечного моніторингу та керування домашніми пристроями з підтримкою OpenHAB, потребує ґрунтовного опрацювання вимог охорони праці (ОП) і безпеки в надзвичайних ситуаціях (БНС). Належний рівень безпеки праці забезпечується трьома базовими складовими: застосуванням безпечного виробничого обладнання, використанням безпечних технологічних процесів і раціональною організацією виконання робіт. Вимоги до безпеки обладнання, зокрема визначені ГОСТ 12.2.003–91, передбачають упровадження конструктивних рішень, спрямованих на зниження ризиків (наприклад, шляхом мінімізації запасеної/споживаної енергії, використання дистанційного керування та обов’язкових вбудованих засобів захисту), дотримання

ергономічних норм (ГОСТ 12.2.032–78 та інші) і забезпечення стійкості обладнання до відмов, зокрема у випадках переривання та відновлення енергопостачання.

Безпека технологічних процесів, регламентована ГОСТ 12.3.002–75, вимагає виключення безпосереднього контакту персоналу з небезпечними чинниками, комплексної механізації й автоматизації, упровадження систем контролю для аварійного вимкнення та оперативного інформування про небезпечні стани, а також обов’язкового дотримання вимог до розміщення обладнання і раціональної організації режимів праці та відпочинку. Окремий акцент має бути зроблений на вимогах до фізичного й психофізіологічного стану працівників, які допускаються до виконання робіт, з урахуванням необхідності періодичного контролю їхнього здоров’я.

У межах IoT-рішень особливої ваги набуває взаємодія в системі «людина–машина», де оператор, попри високий рівень автоматизації, залишається ключовою ланкою, насамперед у частині прийняття рішень за непередбачуваних умов. Його діяльність охоплює послідовне сприймання даних з індикаторів, їх оцінювання та переробку з формуванням концептуальної моделі процесу, вибір рішення і реалізацію керівних впливів; при цьому надійність роботи підтримується постійним контролем власних дій. Отже, успішне впровадження проєкту передбачає узгоджену інтеграцію технічних заходів безпеки та ергономічних рішень із підвищенням надійності людського фактора, що дозволяє мінімізувати ризики небезпечних ситуацій, зумовлених як технічними збоями, так і помилками персоналу.

ВИСНОВКИ

У виконаній роботі кваліфікаційного рівня «магістр» розв’язано комплексне науково-прикладне завдання, що полягає у розробці та дослідженні IoT-системи для безпечного моніторингу та керування домашніми пристроями з використанням платформи openHAB як інтеграційного ядра. Обрана проблематика є актуальною через швидке зростання кількості побутових IoT-пристроїв та паралельне підвищення кіберризиків, коли компрометація цифрового контуру (мережі, брокера повідомлень, API або облікових даних) може призводити не лише до витоку конфіденційної інформації, а й до фізичного впливу на житлове середовище шляхом несанкціонованого керування освітленням, живленням, мікрокліматом чи іншими актуаторами. У роботі послідовно реалізовано підхід, за якого безпека трактується як системна властивість архітектури та експлуатації, а не як «додаткова опція» протоколу, що узгоджується з сучасними поглядами на проблеми захисту, приватності та довіри в IoT [21], [22].

У межах аналітичної частини систематизовано підходи до побудови smart home рішень та окреслено ключові виклики, характерні для гетерогенних середовищ, у яких одночасно співіснують різні класі пристроїв, протоколи та програмні платформи. Показано, що центральною проблемою практичних впроваджень є необхідність узгодити продуктивність (затримка, стійкість до втрат, масштабованість), зручність керування (єдиний інтерфейс, сумісність) та вимоги кібербезпеки (конфіденційність і цілісність трафіку, автентифікація, авторизація, контроль доступу) без надмірного ускладнення системи. На основі відомих архітектурних підходів та референсних моделей IoT уточнено ролі компонентів і їхні взаємодії, що створило основу для подальшого вибору технологій і визначення довірчих меж (trust boundaries) між сегментами домашньої мережі, IoT-контуром та зовнішніми сервісами [23], [24], [26]. Додатково обґрунтовано доцільність локально-орієнтованого керування та зменшення експозиції сервісів у публічний Інтернет, оскільки саме віддалений

доступ та інтерфейси керування є найбільш ризиковими з позиції атак та зловживань [44], [45].

У технологічній частині обґрунтовано вибір openHAB як центральної платформи автоматизації, що забезпечує інтеграцію пристроїв і протоколів, уніфікацію даних у вигляді елементів стану (Items) і реалізацію логіки керування через правила. Як канал телеметрії та команд для периферійних вузлів обрано підхід publish/subscribe на базі MQTT, що є практично доцільним для подійних потоків і великої кількості однотипних повідомлень у домашньому середовищі [9]. Водночас у роботі послідовно враховано, що MQTT у базовому вигляді не забезпечує криптографічного захисту, а тому інтегрується з TLS 1.3 для забезпечення конфіденційності та цілісності трафіку, з можливістю розширення до сертифікатної автентифікації клієнтів і строгого контролю доступу [11]. Для альтернативних сценаріїв легких вузлів і UDP-орієнтованих обмінів проаналізовано CoAP та відповідні механізми захисту, зокрема DTLS 1.3 і OSCORE як інструменти, що можуть бути актуальними в спеціалізованих умовах або при обмежених ресурсах пристроїв [10], [12], [17], [28]. Розглянуто можливість токен-орієнтованого контролю доступу (JWT) як сучасного стандартного інструменту перенесення авторизаційних тверджень у розподілених системах, що дає підґрунтя для посилення периметра віддаленого доступу та інтеграцій із зовнішніми сервісами [14]. У підсумку технологічна частина сформувала узгоджене бачення архітектури даних і керування: телеметрія та стани надходять у систему через MQTT, логіка обробки та автоматизації реалізується на рівні openHAB, а команди керування повертаються до актуаторів контрольованими каналами з фіксацією результатів у зворотному потоці станів.

У конструкторській частині деталізовано практичну реалізацію системи на рівні вузлів, інтерфейсів і правил побудови контурів керування. Визначено структуру сенсорних вузлів, що відповідають за зняття параметрів середовища, формування телеметрії та її передачу до брокера, а також структуру актуаторних вузлів, які здійснюють комутацію навантажень і потребують підвищеної уваги до електробезпеки, гальванічної розв’язки та fail-safe

поведінки у разі збоїв зв’язку чи перезапуску. Показано, що надійність і безпечність системи у житловому середовищі формується на стику програмних політик і апаратних рішень: навіть коректна криптографія не компенсує ризиків, якщо силовий контур зібраний без дотримання правил захисту та ізоляції, а сценарії керування не передбачають безпечних дефолтних станів. У межах інтеграції з openHAB визначено принципи уніфікації пристроїв через модель Things/Channels/Items і організацію правил автоматизації як відтворюваного механізму керування. Важливим інженерним результатом стало проєктування структури MQTT-тем і узгодження її з політикою доступів, що дозволяє реалізувати принцип найменших привілеїв: сенсор не повинен мати права публікувати команди, а актуатор — читати чутливу телеметрію, яка не потрібна для виконання його функцій. Такий підхід знижує ризики ескалації при компрометації окремого вузла і підтримує сегментацію системи на функціональні домени.

Науково-дослідна частина надала роботі експериментально-методичну основу та перетворила запропоновану систему на об’єкт дослідження, що оцінюється через формалізовані метрики та відтворювані сценарії. Сформовано методику вимірювань для кількісної оцінки затримки керування та доставки повідомлень, у тому числі з використанням перцентильних характеристик (P50/P95/P99) для адекватного опису поведінки системи в умовах «хвостових» затримок. Запропоновано підхід до розділення часових інтервалів за точками вимірювання (від формування повідомлення на стороні клієнта/вузла до фактичного виконання команди актуатором і повернення підтвердження), що дозволяє локалізувати вузькі місця між брокером, платформою автоматизації та периферією. Окремо передбачено дослідження надійності доставки (delivery ratio), стійкості до деградацій мережі (затримка, втрати, джиттер), а також ресурсних витрат центрального вузла та брокера (CPU/RAM) при масштабуванні кількості підключених пристроїв. Показано, що наукова новизна підходу полягає не в застосуванні окремого протоколу чи інструмента, а в формуванні комплексної програми досліджень, яка дозволяє оцінити компроміс «безпека—продуктивність—стійкість» для реалістичних сценаріїв

smart home, а також визначити практичні режими конфігурації системи залежно від цілей і обмежень.

Спеціальна частина забезпечила відповідність роботи сучасним вимогам кібербезпеки споживчого IoT та вимогам до приватності й правомірної обробки даних. Як рамкові орієнтири використано ETSI EN 303 645, який визначає базові вимоги до кібербезпеки споживчих IoT-пристроїв, та NIST SP 800-213, що задає системний підхід до формування вимог до IoT-пристроїв як компонентів ширшої інформаційної системи [47], [48]. У роботі показано, що такі документи є практично корисними не лише для декларативного опису, а й для інженерного «перекладу» в конкретні технічні політики: керування обліковими даними, контроль оновлень, мінімізація поверхні атаки, журналювання, керованість конфігурації, обмеження доступів і побудова контурів віддаленого доступу. Для openHAB-орієнтованої системи спеціально акцентовано важливість коректного налаштування доступу до REST API і відмови від неавтентифікованих режимів доступу в контексті сегментації мережі або віддаленого керування [45]. Для MQTT-контуру визначено необхідність TLS-захисту та авторизації за темами через ACL як базової умови «жорсткого» конфігурування (hardening) брокера Mosquitto [46]. Питання приватності розглянуто як інженерне завдання, оскільки телеметричні дані «розумного дому» здатні розкривати патерни поведінки мешканців; тому запропоновано принципи мінімізації даних, обмеження строків зберігання, локальності обробки та контролю доступу до журналів і історії, що узгоджується з принципами GDPR та нормами українського законодавства про персональні дані [49], [50].

ПЕРЕЛІК ДЖЕРЕЛ

1 Sowah, R. A., Boahene, D. E., Owoh, D. C., Addo, R., Mills, G. A., Owusu-Banahene, W., Buah, G., & Sarkodie-Mensah, B. (2020). Design of a Secure Wireless Home Automation System with an Open Home Automation Bus (OpenHAB 2) Framework. Journal of Sensors, 2020, Article ID 8868602.

2 Fagan, M., Megas, K. N., Scarfone, K., & Smith, M. (2020). Foundational cybersecurity activities for IoT device manufacturers (NISTIR 8259). National Institute of Standards and Technology.

3 Fagan, M., Marron, J., Brady, K. G., Jr., Cuthill, B. B., Megas, K. N., & Herold, R. (2021). IoT non-technical supporting capability core baseline (NISTIR 8259B). National Institute of Standards and Technology.

4 openHAB Foundation. (n.d.). Items (Configuration). openHAB Documentation.

5 openHAB Foundation. (n.d.). Textual Rules (Rules DSL). openHAB Documentation.

6 openHAB Foundation. (n.d.). openHAB REST API (Documentation).

7 openHAB Foundation. (n.d.). Securing Communication and Access

8 openHAB Foundation. (n.d.). myopenHAB (Service).

9 OASIS. (2014). MQTT Version 3.1.1. OASIS Standard.

10 Shelby, Z., Hartke, K., & Bormann, C. (2014). The Constrained Application Protocol (CoAP) (RFC 7252). RFC Editor.

11 Rescorla, E. (2018). The Transport Layer Security (TLS) Protocol Version 1.3 (RFC 8446). IETF.

12 Rescorla, E., Tschofenig, H., Fossati, T., & Tucker, M. (2022). The Datagram Transport Layer Security (DTLS) Protocol Version 1.3 (RFC 9147). IETF.

13 Hardt, D. (2012). The OAuth 2.0 Authorization Framework (RFC 6749). RFC Editor.

14 Jones, M., Bradley, J., & Sakimura, N. (2015). JSON Web Token (JWT) (RFC 7519). IETF.

15 Fagan, M., Megas, K. N., Scarfone, K., & Smith, M. (2020). IoT Device

Cybersecurity Capability Core Baseline (NISTIR 8259A). National Institute of Standards and Technology.

16 1OASIS. (2019). MQTT Version 5.0. OASIS Standard.

17 Selander, G., Mattsson, J., Palombini, F., & Seitz, L. (2019). Object Security for Constrained RESTful Environments (RFC 8613). RFC Editor.

18 International Organization for Standardization. (2022). ISO/IEC 27400:2022 Cybersecurity — IoT security and privacy — Guidelines.

19 openHAB Foundation. openHABian Installation on Raspberry Pi Systems

20 Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376.

21 Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in IoT: The road ahead. Computer Networks, 76, 146–164.

22 Roman, R., Zhou, J., & Lopez, J. (2013). On the features and challenges of security and privacy in distributed IoT. Computer Networks, 57, 2266–2279.

23 Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. Computer Networks, 54, 2787–2805.

24 Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29, 1645–1660.

25 ETSI. (2020). ETSI EN 303 645 V2.1.1: Cyber Security for Consumer Internet of Things. European Telecommunications Standards Institute.

26 ISO/IEC. (2018). ISO/IEC 30141:2018 Internet of Things (IoT) — Reference architecture. International Organization for Standardization / International Electrotechnical Commission.

27 Thangavel, D., Ma, X., Valera, A., Tan, H. X., & Tan, C. K. Y. (2014). Performance Evaluation of MQTT and CoAP via a Common Middleware. In Proceedings of the IEEE 9th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP) (pp. 1–6).

28 Bormann, C., & Lemay, S. (2018). CoAP (Constrained Application Protocol)

over TCP, TLS, and WebSockets (RFC 8323). RFC Editor.

29 Bosch Sensortec. (2022). BME280 Data Sheet (Rev. 1.23).

30 openHAB Foundation. (n.d.). MQTT Binding; MQTT Things and Channels Binding. openHAB Add-ons Documentation.

31 Prantl, T., Lichtblau, B., & Sommer, R. (2021). Performance Impact Analysis of Securing MQTT Using TLS. In Proceedings of the ACM/SPEC International Conference on Performance Engineering (ICPE).

32 Ford, T. N., & colleagues. (2022). Performance Evaluation of Different Raspberry Pi Models as MQTT Servers and Clients.

33 Seoane, V., Pozo, F., & colleagues. (2021). Performance evaluation of CoAP and MQTT with security. Computer Networks, 197, 108338.

34 Gavriilidis, N. O., & colleagues. (2025). Empirical Evaluation of TLS-Enhanced MQTT on IoT Platforms (TLS 1.3; cipher suite trade-offs). Applied Sciences, 15(15), 8398.

35 Gentile, A. F., & colleagues. (2024). A Network Performance Analysis of MQTT Security (including QoS tiers and network/VLAN considerations).

36 CoAP + DTLS: A Comprehensive Overview of Cryptographic Performance on an IoT Scenario. (2021).

37 Fischer, T., & colleagues. (2019). Analyzing power consumption of TLS ciphers on an ESP32.

38 Salles, R., & colleagues. (2023). TLS Protocol Analysis Using IoTST—An IoT Benchmark.

39 Soni, N. (2024). Impact of Performance on Security: JWT Token.

40 Jánoky, L. V., & colleagues. (2018). An analysis on revoking mechanisms for JSON Web Tokens. Journ.of Information Security and Applications, 41, 172–181.

41 Patel, J., & Gamess, E. (2024). A Study of the Resilience of the Mosquitto MQTT Broker Running on Raspberry Pi Against Fuzzy DDoS Attacks.

42 Mills, D. L., Martin, J., Burbank, J., & Kasch, W. (2010). Network Time Protocol Version 4: Protocol and Algorithms Specification (RFC 5905). IETF.

43 Kombate, Y., & colleagues. (2024). Securing MQTT: unveiling vulnerabilities and innovating resilience assessment approaches.

44 openHAB. (n.d.). Securing Communication and Доступ 2025, www.openhab.org

45 openHAB. (n.d.). openHAB REST API. Доступ 2025, www.openhab.org

46 Eclipse Mosquitto Project. mosquitto.conf — Доступ 2025, , mosquitto.org

47 ETSI. (2024). ETSI EN 303 645 V3.1.3 (2024-09): Cyber Security for Consumer Internet of Things: Baseline Requirements.

48 Fagan, M., Megas, K. N., Scarfone, K., & Smith, M. (2021). IoT Device Cybersecurity Guidance for the Federal Government (NIST SP 800-213). National Institute of Standards and Technology.

49 European Parliament and Council of the European Union. (2016). Regulation (EU) 2016/679 (General Data Protection Regulation).

50 Verkhovna Rada of Ukraine. (2010). Law of Ukraine: On Personal Data Protection (No. 2297-VI). NATLEX (ILO).

51 Stanko, A., Wieczorek, W., Mykytyshyn, A., Holotenko, O., & Lechachenko, T. (2024). Real-time air quality management: Integrating IoT and Fog computing for effective urban monitoring. CITI'2024: 2nd International Workshop on Computer Information Technologies in Industry 4.0, June 12–14, 2024, Ternopil.

52 Stanko, A., Palka, O., Matiichuk, L., Martsenko, N., & Matsiuk, O. (2021, September). Smart City: A Review of Model Architecture and Technology. In 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT) (Vol. 2, pp. 273-277). IEEE.

53 Основи наукових досліджень і теорія експерименту : Навчальний посібник для здобувачів освітнього ступеня «Магістр» спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» / укл. Ю. Б. Капаціла, П. О. Марущак, В. Б. Савків, О. П. Шовкун. Тернопіль : ФОП Паляниця В.А., 2023. 186 с. https://elartu.tntu.edu.ua/handle/lib/40843

54 Коноваленко І. В. Платформа .NET та мова програмування C# 8.0 : навч.

посіб. / І. В. Коноваленко, П. О. Марущак. – Тернопіль : В. А. Паляниця, 2020 – 320 с. http://library.megu.edu.ua:8180/jspui/handle/123456789/4115

55 Проектування мікропроцесорних систем керування: навчальний посібник / І.Р. Козбур, П.О. Марущак, В.Р. Медвідь, В.Б. Савків, В.П. Пісьціо. – Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2022. – 324 с. https://elartu.tntu.edu.ua/handle/lib/39189

56 Капаціла Ю.Б., Савків В.Б. Методичні вказівки до виконання кваліфікаційної роботи бакалавра спеціальності «Автоматизація, комп'ютерно-інтегровані технології та робототехніка». Тернопіль.: Видавництво ТНТУ. 2025. 60 с. https://elartu.tntu.edu.ua/handle/lib/48495

57 Капаціла Ю.Б., Шовкун О.П. Дослідження релейного електронного регулятора температури. /Методичні вказівки до лабораторної роботи з курсу «Приводи та автоматика мехатронних систем» для здобувачів освітнього ступеня «бакалавр» спеціальності «Автоматизація, комп'ютерно-інтегровані технології та робототехніка». Тернопіль: ТНТУ. 2025. 15 с. http://elartu.tntu.edu.ua/handle/lib/50560.

58 Козбур І.Р., «Дослідження часових характеристик неперервних лінійних систем», по курсу «Теорія автоматичного управління», для студентів 3 курсу спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / Авт.: Козбур І.Р., Козбур Г.В. Марущак П.О., Савків В.Б. – Тернопіль: ТНТУ, ФПТ, каф. АВ, – 2022. – 19 с. https://elartu.tntu.edu.ua/handle/lib/39206

59 Капаціла Ю.Б., Марущак П.О., Савків В.Б. Методичні вказівки з виконання курсової роботи з дисципліни «Основи наукових досліджень» для здобувачів освітнього ступеня «Магістр» спеціальності 174 «Автоматизація та комп'ютерно-інтегровані технології». Тернопіль: ТНТУ, 2023. 32 с

60 Автоматизація виробничих процесів. Навчальний посібник для технічних спеціальностей вищих навчальних закладів. / Я.І. Проць, В.Б. Савків, О.К. Шкодзінський, О.Л. Ляшук. Тернопіль: ТНТУ ім. І. Пулюя,

2011. 344 с. https://elartu.tntu.edu.ua/handle/123456789/1551

61 Методичні вказівки по роботі з програмним симулятором "AVR simulator IDE" з курсу "Мікропроцесорні та програмні засоби автоматизації" / укл. : В.Р. Медвідь , В.П. Пісьціо. - Тернопіль : ТНТУ імені Івана Пулюя, 2020. - 21 с.

62 Моделювання нелінійних систем керування у пакеті MATLAB SIMULINK, методичні вказівки до виконання лабораторної роботи по курсу «Комп’ютерні методи дослідження систем автоматичного управління», для студентів 4 курсу спеціальності 6.050201 «Системна інженерія» / укл. : І.Р. Козбур , Г.В. Козбур , Р.І. Михайлишин. – Тернопіль : ТНТУ імені Івана Пулюя, 2019. - 19 с. https://elartu.tntu.edu.ua/handle/lib/28057

63 Моделювання систем керування в пакеті MATLAB SIMULINK, методичні вказівки до виконання лабораторної роботи по курсу «Комп’ютерні методи дослідження систем автоматичного управління», для студентів 4 курсу спеціальності 6.050201 «Системна інженерія» / укл. : І.Р. Козбур , Г.В. Козбур , Р.І. Михайлишин. – Тернопіль : ТНТУ, 2019. - 23 с. https://elartu.tntu.edu.ua/handle/lib/28056

64 Проектування та аналіз електричних схем в програмному середовищі Multisim. Методичні вказівки до самостійної роботи студентів з курсу "Проектування мікропроцесорних систем керування технологічними процесами" / укл. : В.Р. Медвідь , В.П. Пісьціо . - Тернопіль : ТНТУ Імені Івана Пулюя, 2018. - 26 с. https://elartu.tntu.edu.ua/handle/lib/26404

65 Проектування та аналіз електричних схем в програмному середовищі Proteus VSM. Методичні вказівки до самостійної роботи студентів з курсу "Проектування мікропроцесорних систем керування технологічними процесами" / укл. : В.Р. Медвідь , В.П. Пісьціо. - Тернопіль : ТНТУ, 2018. - 26 с. https://elartu.tntu.edu.ua/handle/lib/26397

66 Методичні вказівки до курсового проектування з курсу "Проектування систем автоматизації" / Савків В.Б., Шкодзінський О.К., Пісьціо В.П. Тернопіль : ТНТУ, 2025. 128 с. https://elartu.tntu.edu.ua/handle/lib/48646

67 Трембач Р.Б., Медвідь В.Р. Методичні вказівки до виконання курсового проекту з дисципліни “Електроніка і мікросхемотехніка”. – Тернопіль: ТНТУ ім. І. Пулюя, 2024. – 53 с. https://elartu.tntu.edu.ua/handle/lib/44635

68 Трембач Р.Б., Медвідь В.Р. Методичні вказівки до виконання до виконання лабораторних робіт з дисципліни “Електроніка і мікросхемотехніка” Модуль 3. «Імпульсна техніка, вторинні джерела живлення, основи цифрової електроніки» – Тернопіль: ТНТУ, 2024. -26 с.

69 Трембач Р.Б., Шовкун О.П. Методичні вказівки до виконання до виконання лабораторних робіт з дисципліни “Інформаційно – вимірювальні системи” Модуль І. «Вимірювальна техніка» – Тернопіль: ТНТУ., 2024. – 67 с.

70 Методичні вказівки для написання розділу «Безпека життєдіяльності, основи охорони праці» в кваліфікаційних роботах здобувачів освітнього рівня ,,бакалавр”. Для студентів всіх форм навчання рівень вищої освіти перший (бакалаврський)/ укл.: О. Я. Гурик , І. Б. Окіпний. – Тернопіль: ТНТУ імені Івана Пулюя, 2021. - 20 с.

71 Навчально-методичний посібник до практичних заняття з дисципліни «Безпека життєдіяльності, основи охорони праці» для студентів освітнього ступеня ,,бакалавр" усіх спеціальностей та форм навчання / Укладачі : О. Я. Гурик, І. Б. Окіпний, В. С. Сенчишин, С. Ю. Мариненко, О. І. Король. Тернопіль : ТНТУ імені Івана Пулюя, 2025. 123 с. https://elartu.tntu.edu.ua/handle/lib/48496

72 Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Комп’ютерні мережі. Книга 1 [навчальний посібник]. Львів : «Магнолія 2006», 2013. 256 с.

73 Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Комп’ютерні мережі. Книга 2. [навчальний посібник]. Львів : "Магнолія 2006", 2014. 312 с.

74 Микитишин А. Г., Митник М. М., Стухляк П. Д. Телекомунікаційні системи та мережі. Тернопіль: Тернопільський національний технічний

університет імені Івана Пулюя, 2017. 384 с.

75 Буров Є., Митник М. Комп’ютерні мережі. (у 2-х томах). Львів, Магнолія, 2018.

76 Комплексна безпека інформаційних мережевих систем. Навчальний посібник для студентів спеціальності 174 «Автоматизація, комп’ютерно-інтегровані технології та робототехніка» / А. Г. Микитиший, М. М. Митник, О. С. Голотенко, В. В. Карташов. – Тернопіль: ФОП Паляниця В.А., 2023. – 324 с. https://elartu.tntu.edu.ua/handle/lib/42625

77 Пилипець М. І. Правила заповнення основних форм технологічних документів : навч.-метод. посіб. / Уклад. Пилипець М. І., Ткаченко І. Г., Левкович М. Г., Васильків В. В., Радик Д. Л. Тернопіль : ТДТУ, 2009. 108 с. https://elartu.tntu.edu.ua/handle/lib/42995

78 Стручок В.С. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання / В.С.Стручок. — Тернопіль: ФОП Паляниця В. А., 2022. — 156 с.

ДОДАТКИ

Додаток А

Тези конференцій

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

XIII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

17-18 грудня 2025 року

ТЕРНОПІЛЬ
2025

УДК 004.738.5:004.77:004.056
М. Трембач, к. т. н., доц.; Р. Павелко; Д. Биць
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ІОТ-СИСТЕМА БЕЗПЕЧНОГО МОНІТОРИНГУ ТА КЕРУВАННЯ ДОМАШНІМИ ПРИСТРОЯМИ З ПІДТРИМКОЮ OPENHAB

UDC 004.738.5:004.77:004.056
M. Trembach, PhD., Assoc. Prof.; R. Pavelko; D. Byts

IOT SYSTEM FOR SECURE MONITORING AND CONTROL OF HOME DEVICES WITH OPENHAB SUPPORT

Зростання кількості побутових IoT-пристроїв (датчики мікроклімату, розумні розетки, освітлення, охоронні сенсори, HVAC) підвищує вимоги до надійності та кіберстійкості домашніх автоматизованих систем. Типові ризики зумовлені гетерогенністю протоколів, різною якістю прошивок, наявністю зовнішніх інтеграцій і помилками конфігурації, що в сукупності створює передумови для несанкціонованого доступу, витоку телеметрії та небезпечних керувальних впливів. У цьому контексті актуальним є підхід, за якого логіка керування зосереджується в локальному контурі, а взаємодія з пристроями нормалізується через єдиний центр автоматизації з чітко визначеними інтерфейсами та політиками доступу.

Обґрунтуємо архітектуру IoT-системи «розумного дому» на базі openHAB як платформи інтеграції та автоматизації, а також визначення практичних заходів безпеки для моніторингу й керування домашніми пристроями. openHAB доцільно розглядати як інтеграційне ядро, що підтримує модель сутностей (Items/Things), правила автоматизації та програмні інтерфейси взаємодії, зокрема REST API для керування станами і доступу до даних компонентів системи [1]. Це спрощує побудову єдиного керувального контуру для різних технологій підключення та забезпечує масштабованість через модульність інтеграцій.

Запропонована архітектура передбачає три рівні: польовий (сенсори/актуатори), транспортно-шлюзовий (MQTT/локальні брокери, адаптери протоколів), прикладний (openHAB із правилами автоматизації, журналюванням подій і рольовим доступом до інтерфейсів). Безпечне керування реалізується через мінімізацію «поверхні атаки» (відмова від зайвих відкритих сервісів на пристроях), шифрування каналів (TLS для MQTT/HTTP-взаємодії), сегментацію домашньої мережі (окрема VLAN/SSID для IoT), а також керування обліковими даними та сесіями для користувацьких інтерфейсів і зовнішніх інтеграцій. Вимоги до базового рівня кіберзахисту доцільно узгоджувати з положеннями ETSI EN 303 645 (зокрема щодо унікальних/надійних паролів, безпечного оновлення, захисту персональних даних і розкриття вразливостей) [2] та з набором ключових кіберздатностей IoT-пристрою за NISTIR 8259A (керування ідентифікацією, конфігурацією, оновленнями, захистом даних, контролем доступу до інтерфейсів тощо). Виділимо критерії ефективності системи пропонується оцінювати за показниками: стійкість до типових сценаріїв (компрометація пристрою/облікового запису), коректність політик доступу, відмовостійкість локального контуру керування та стабільність обміну телеметрією в умовах перешкод і втрат пакетів.

Отримані положення формують практичну основу для створення «розумного дому», де openHAB виступає центром інтеграції та автоматизації, а безпека досягається поєднанням стандартизованих вимог до пристроїв і системних заходів захисту на рівні мережі, інтерфейсів і керувальної логіки [1, 2]. Це забезпечує контрольоване масштабування системи, зменшує залежність від хмарних сервісів і підвищує довіру до рішень домашньої автоматизації в умовах зростання кіберзагроз.

Література

1. Eclipse Foundation. openHAB Documentation. Джерело: https://www.openhab.org
2. ETSI. (2024). ETSI EN 303 645 V3.1.3 (2024-09): Cyber Security for Consumer Internet of Things.

.