ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Система взаємодії протоколів IoT для побудови «Розумного будинку»»

на здобуття освітнього ступеня бакалавра
зі спеціальності 122 Комп'ютерні науки
освітньо-професійної програми Штучний інтелект

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Олександр ВАГАНОВ
(підпис)

Виконав: Здобувач вищої освітигрупа ШІД-42
Олександр ВАГАНОВ

Керівник:
науковий ступінь, вчене звання
Максим ФЕСЕНКО
к.т.н., доцент

Рецензент:
науковий ступінь, вчене звання
_____
Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Штучного інтелекту

Ступінь вищої освіти Бакалавр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Штучний інтелект

ЗАТВЕРДЖУЮ
Завідувач кафедрою Штучного інтелекту

_____ Ольга ЗІНЧЕНКО

«_____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Ваганова Олександра Юрійовича

1. Тема кваліфікаційної роботи: Система взаємодії протоколів IoT для побудови «Розумного будинку».

керівник кваліфікаційної роботи Максим Фесенко к.т.н., доцент,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025 р. № 56

2. Строк подання кваліфікаційної роботи «23» травня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література з IoT-технологій, технічна документація хмарних сервісів Microsoft Azure, принципи побудови розумних систем управління, характеристики та специфікації мережевих протоколів (MQTT, AMQP, HTTPS), сучасні вимоги до енергоефективності, безпеки та масштабованості IoT-систем.

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

Теоретичні основи Інтернету речей у контексті «Розумного будинку»
Аналіз та порівняння комунікаційних протоколів MQTT, AMQP, HTTPS
Визначення вимог до IoT-протоколів для системи Smart Home
Розробка архітектури обміну даними між пристроями
Розробка моделі управління пристроями у середовищі Azure IoTCentral

Реалізація та тестування сценаріїв взаємодії пристроїв
Аналіз продуктивності, надійності та енергоефективності системи

5. Перелік графічного матеріалу: *презентація*
   1. Структура IoT-системи для «Розумного будинку»
   2. Порівняльна таблиця протоколів взаємодії
   3. Архітектура обміну даними між пристроями та хмарною платформою
   4. Інтерфейс управління у Azure IoT Central
   5. Приклади автоматизованих сценаріїв взаємодії
6. Дата видачі завдання «27» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапівкваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз сучасних протоколів взаємодії в IoT-системах	27.02-05.11.24	Виконано
2	Розробка архітектури IoT-системи для «Розумного будинку	05.11-12.11.24	Виконано
3	Вибір та обґрунтування комунікаційних протоколів (MQTT, AMQP, HTTPS)	13.11-19.11.24	Виконано
4	Реалізація симуляції пристроїву середовищі Azure IoT Central	20.11-25.11.24	Виконано
5	Розробка сценаріїв обміну данимита керування пристроями	27.11-03.12.24	Виконано
6	Тестування ефективності роботи обраних протоколів	04.12-10.12.24	Виконано
7	Завершення підготовки текстовоїчастини пояснювальної записки	11.12-20.12.24	Виконано
8	Підготовка демонстраційних матеріалів (слайди, графіки, приклади)	21.12-31.05.25	Виконано

Здобувач вищої освіти _____ Олександр ВАГАНОВ

Керівник
кваліфікаційної роботи _____ Максим ФЕСЕНКО

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра складається з 58 сторінок, 23 рисунків, 1 таблиця, 24 джерел інформації.

Мета роботи – розробка та аналіз протоколів взаємодії IoT для побудови системи «Розумного будинку» з використанням хмарних технологій Microsoft Azure IoT Central.

Об'єкт дослідження – хмарна платформа Microsoft Azure IoT Central та віртуальні пристрої, які імітують компоненти системи «Розумного будинку».

Предмет дослідження – процеси передавання даних між IoT-пристроями та хмарними сервісами з використанням різних комунікаційних протоколів (MQTT, AMQP, HTTPS).

Короткий зміст роботи: Кваліфікаційна робота присвячена розробці та експериментальному дослідженню системи «Розумного будинку» на базі платформи Azure IoT Central. Було налаштовано віртуальні пристрої для імітації реальних сенсорів та виконавчих пристроїв.Проведено аналіз і порівняння кількох протоколів зв'язку за основними показниками продуктивності, такими як затримка передавання даних та енергоефективність.Виконано експериментальні дослідження для оцінки ефективності передавання телеметрії, виконання команд і надійності доставки повідомлень.Отримані результати дозволили здійснити комплексний аналіз ефективності використання кожного з протоколів і надати рекомендації щодо оптимізації систем «Розумного будинку».

КЛЮЧОВІ СЛОВА: ІНТЕРНЕТ РЕЧЕЙ, ХМАРНІ ТЕХНОЛОГІЇ, РОЗУМНИЙ БУДИНОК, AZURE IOT CENTRAL, MQTT, AMQP, HTTPS, ТЕЛЕМЕТРІЯ, ПЕРЕДАВАННЯ ДАНИХ, МЕРЕЖЕВІ ПРОТОКОЛИ.

ABSTRACT

The text part of the bachelor's qualification work consists of 58 pages, 23 figures, 1 tables, and 24 sources of information.

Purpose of the work - development and analysis of IoT interaction protocols for building a Smart Home system using Microsoft Azure IoT Central cloud technologies.

Object of the research – cloud platform Microsoft Azure IoT Central and virtual devices that simulate Smart Home components.

Subject of the research – processes of data transmission between IoT devices and cloud services using different communication protocols (MQTT, AMQP, HTTPS).

Brief description of the work: This thesis focuses on the development and experimental testing of a Smart Home system based on Azure IoT Central. Virtual devices were configured to simulate real-world sensors and actuators. Several communication protocols were analyzed and compared based on key performance indicators such as data transmission delay and energy efficiency.

Experimental studies were conducted to assess the performance of telemetry transmission, command execution, and message delivery reliability.

The obtained results allowed for a comprehensive analysis of each protocol’s efficiency and the development of recommendations for optimizing Smart Home systems.

KEYWORDS: INTERNET OF THINGS, CLOUD TECHNOLOGIES, SMART HOME, AZURE IOT CENTRAL, MQTT, AMQP, HTTPS, TELEMETRY, DATA TRANSMISSION, NETWORK PROTOCOLS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API — Application Programming Interface, інтерфейс прикладного програмування.

IoT — Internet of Things, Інтернет речей.

MQTT — Message Queuing Telemetry Transport, легкий протокол передачі даних для IoT-систем.

AMQP — Advanced Message Queuing Protocol, протокол черг повідомлень з розширеними можливостями.

HTTPS — HyperText Transfer Protocol Secure, захищений протокол передачі гіпертексту.

QoS — Quality of Service, рівень якості обслуговування мережевого трафіку.

JSON — JavaScript Object Notation, текстовий формат обміну даними.

SAS Token — Shared Access Signature Token, токен для безпечного обміну даними в Azure.

Azure IoT Central — хмарна платформа Microsoft для побудови IoT-рішень.

REST API — архітектурний стиль для взаємодії розподілених систем через HTTP-запити.

Telemetry — телеметрія, процес дистанційної передачі даних від пристроїв до хмарної системи.

Smart Home — розумний будинок, автоматизована система управління побутовими приладами.

CSV — Comma-Separated Values, формат представлення табличних даних.

Latency — затримка передачі даних в мережі.

Token — цифровий ключ автентифікації доступу до сервісів.

Azure Storage — сервіс хмарного зберігання даних Microsoft Azure.

ВСТУП

Сучасний розвиток технологій Інтернету речей (IoT) сприяє активному впровадженню розумних пристроїв у побут. Одним з популярних напрямів використання IoT є створення систем «Розумного будинку», які дозволяють підвищити комфорт, безпеку та ефективне використання ресурсів у житлі.

Для правильної роботи таких систем важливо організувати обмін даними між пристроями. Це забезпечується за допомогою спеціальних протоколів взаємодії. На сьогодні існує багато протоколів, серед яких MQTT, CoAP, AMQP, HTTP/REST та Matter. Важливо проаналізувати їх можливості та вибрати найбільш оптимальні для застосування в системах «Розумного будинку».

Швидкий розвиток технологій Інтернету речей (IoT) суттєво впливає на різні сфери діяльності людини, включаючи побут, промисловість, транспорт та енергетику. Особливо активно ці технології впроваджуються у створення систем «Розумного будинку», які забезпечують підвищення комфорту, безпеки та ефективності використання ресурсів.

Завдяки використанню IoT-пристроїв, користувачі можуть дистанційно контролювати освітлення, кліматичні умови, безпеку житла та інші системи, що значно спрощує повсякденне життя. Однак для забезпечення надійної взаємодії між такими пристроями необхідно використовувати ефективні протоколи обміну даними, які відповідають сучасним вимогам до швидкості передачі, безпеки, стабільності та енергоефективності.

На сьогоднішній день існує багато різних протоколів взаємодії в IoT, кожен з яких має свої переваги та обмеження. Тому важливо дослідити та обґрунтувати вибір найбільш придатних протоколів саме для сфери «Розумного будинку».

Актуальність теми – розвиток Інтернету речей (IoT) кардинально змінює підходи до організації життєвого простору, роблячи його більш комфортним, безпечним та енергоефективним. Системи «Розумного будинку» дедалі частіше стають невід’ємною частиною сучасного житла, оскільки дозволяють централізовано керувати освітленням, безпекою, кліматом і побутовими приладами.

Незважаючи на широкий асортимент готових комерційних рішень, більшість з них є закритими, дорогими або обмеженими в можливостях масштабування. Це обумовлює актуальність дослідження ефективних і водночас економічно доступних рішень для побудови систем розумного будинку на базі хмарних сервісів та IoT-протоколів, які забезпечують необхідний рівень інтеграції, безпеки та енергоефективності.

Особливої актуальності тема набуває в Україні в умовах підвищення цін на енергоресурси та необхідності раціонального використання енергії, що робить дослідження практично цінним і перспективним для широкого впровадження.

Мета дослідження – розробка та дослідження інтегрованої системи «Розумний будинок» із використанням хмарної платформи Azure IoT Central та протоколів MQTT, AMQP і HTTPS для забезпечення ефективної взаємодії між пристроями, підвищення енергоефективності та спрощення управління системою.

У процесі виконання кваліфікаційної роботи вирішувались наступні завдання:

- Провести аналіз сучасних архітектур IoT-систем та їх ключових протоколів взаємодії.
- Розробити архітектуру системи Smart Home з використанням хмарної платформи Microsoft Azure IoT Central.
- Налаштувати середовище для емуляції пристроїв розумного будинку та реалізувати відповідні програмні моделі засобами Python.
- Розробити та провести тестові сценарії для аналізу продуктивності, затримок і надійності протоколів MQTT, AMQP та HTTPS.
- Побудувати аналітичні графіки на основі отриманих результатів і сформувати обґрунтовані рекомендації щодо оптимізації системи.

Об’єктом дослідження є процеси побудови інтегрованих систем «Розумного будинку» з використанням технологій Інтернету речей, хмарної платформи Microsoft Azure IoT Central та сучасних мережевих протоколів.

Предметом дослідження є методи організації взаємодії між пристроями розумного будинку, хмарною інфраструктурою та аналіз ефективності різних IoT-

протоколів передачі даних (MQTT, AMQP, HTTPS) з точки зору продуктивності, затримок, надійності доставки та енергоефективності.

У роботі застосовувалися такі методи дослідження:

— Теоретичний аналіз літературних джерел і сучасних технологій IoT.
— Проєктування системної архітектури за допомогою хмарних сервісів Microsoft Azure IoT Central.
— Розробка і тестування програмного забезпечення за допомогою мови Python та Azure IoT Device SDK.
— Проведення експериментів і аналіз продуктивності системи в реальному часі з використанням розроблених сценаріїв.
— Візуалізація та статистичний аналіз результатів за допомогою бібліотек Pandas, Matplotlib та Seaborn.

1. ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ ІОТ-СИСТЕМ ДЛЯ «РОЗУМНОГО БУДИНКУ»

У сучасному світі стрімкий розвиток інформаційних технологій та мікроелектроніки призвів до формування концепції Інтернету речей (Internet of Things, IoT), яка стала ключовим чинником трансформації повсякденного життя, промисловості та сфери послуг. Ідея об’єднання фізичних об’єктів у єдину інтерактивну систему має свої витоки ще в 1980-х роках, коли з’явились перші автоматизовані системи моніторингу та керування. Однак саме розвиток мережевих технологій, здешевлення мікроконтролерів і датчиків, а також поява хмарних обчислювальних платформ зробили можливим масове впровадження IoT-рішень у побут, що стало основою для створення концепції «Розумного будинку».

Поняття «Розумний будинок» передбачає інтеграцію різноманітних пристроїв (сенсорів, виконавчих механізмів, контролерів) у єдину систему, здатну автоматично виконувати завдання, пов’язані з безпекою, енергозбереженням, комфортом та оптимізацією використання ресурсів. Така система базується на ефективній взаємодії пристроїв, що неможливо без застосування відповідної математичної бази, фізичних моделей та протоколів обміну даними.

Історія розвитку IoT-систем є яскравим прикладом еволюції інформаційних технологій. Перші кроки у цьому напрямку були зроблені в лабораторіях університетів та промислових дослідницьких центрах. Відомим прикладом ранньої IoT-системи вважається автоматизований апарат для продажу напоїв, розроблений у Carnegie Mellon University у 1982 році, який передавав дані про кількість і температуру напоїв через мережу. Надалі розвиток цієї концепції привів до створення спеціалізованих протоколів передачі даних (наприклад, MQTT, CoAP), а також хмарних платформ для обробки великих обсягів інформації.

З математичної точки зору, проєктування IoT-систем ґрунтується на теорії графів, теорії ймовірностей, теорії масового обслуговування та оптимізації. Структура IoT-мереж зазвичай моделюється як орієнтований граф, вузли якого

відповідають за пристрої, а ребра — за канали зв’язку між ними. Важливою задачею є мінімізація витрат енергії на передачу даних та оптимізація топології мережі для забезпечення стійкості системи у разі відмови окремих вузлів.

Фізична база IoT-систем тісно пов’язана з характеристиками передавання сигналів у бездротових мережах, законами електродинаміки та теорією радіосигналів. Зокрема, важливими є моделі втрат сигналу у середовищах із перешкодами (наприклад, модель Лог-Нормального розподілу), які визначають оптимальні параметри розміщення сенсорів і виконавчих пристроїв. Пристрої IoT часто працюють у діапазонах частот 2,4 ГГц (Wi-Fi, ZigBee) та субгігагерцевих діапазонах (LoRa, NB-IoT), що дозволяє обирати між високою пропускною здатністю та великим радіусом дії в залежності від вимог системи.

Завдяки розвитку хмарних технологій та інтелектуальних алгоритмів обробки даних IoT-системи поступово еволюціонують до так званого Інтернету всього (Internet of Everything), що включає не лише пристрої, але й дані, процеси та людей, інтегруючи їх у єдину цифрову екосистему. Це відкриває нові можливості для автоматизації житлових приміщень, підвищення рівня безпеки, енергоефективності та комфорту.

У підсумку, розвиток IoT-систем для «Розумного будинку» базується на ґрунтовних математичних і фізичних засадах, використанні сучасних мережевих технологій, а також на застосуванні моделей оптимізації та аналізу даних. Вибір відповідних протоколів взаємодії, обчислювальних платформ і топології мережі визначає якість, безпеку та ефективність функціонування таких систем, що й обумовлює актуальність подальших досліджень у цьому напрямі.

1.1. Інтернет речей: концепції, архітектура та сфера застосування в «Розумному будинку»

Інтернет речей (IoT) — це концепція об’єднання фізичних об’єктів, пристроїв і систем у єдину інформаційну мережу для автоматичного збору, обробки та обміну даними. Основна мета IoT — підвищення ефективності процесів, економія ресурсів та покращення якості життя людей [1].

Концепція IoT виникла наприкінці 1990-х років, коли з’явилась можливість вбудовувати мікропроцесори та засоби зв’язку у різноманітні пристрої [2]. Сучасний розвиток технологій бездротового зв’язку, таких як Wi-Fi, ZigBee, Bluetooth Low Energy (BLE), LoRaWAN, а також поширення хмарних сервісів, значно сприяє активному впровадженню IoT у повсякденне життя [3].

Одним з найперспективніших напрямів розвитку IoT є створення систем «Розумного будинку» (Smart Home). Це інтегроване середовище, яке дозволяє автоматизувати управління різноманітними системами житлового приміщення — освітленням, опаленням, вентиляцією, безпекою, мультимедіа тощо [4]. За допомогою інтелектуальних пристроїв користувачі отримують можливість дистанційно керувати побутовими процесами, контролювати стан будинку та ефективно використовувати енергетичні ресурси [5].

Архітектура IoT-системи для «Розумного будинку» базується на кількох ключових рівнях (рис1.1):

1. Рівень пристроїв (Device Layer) — включає різноманітні сенсори (температури, руху, вологості), виконавчі механізми (розумні замки, розетки, системи освітлення) та мікроконтролери. Ці пристрої забезпечують збір даних і виконання команд [6]. Важливо зазначити, що сучасні сенсори стають дедалі енергоефективнішими та мініатюрнішими, що дає змогу інтегрувати їх у більшу кількість побутових приладів.

2. Рівень мережевої взаємодії (Network Layer) — забезпечує передачу даних між пристроями та хмарними сервісами. Основні технології: Wi-Fi, ZigBee, Z-Wave, LoRaWAN, NB-IoT. Вибір технології залежить від вимог до радіусу дії, енергоспоживання та обсягів передаваних даних [7]. Останнім часом спостерігається тренд до використання гібридних мережевих рішень, які дозволяють оптимізувати витрати енергії та підвищити надійність систем.

3. Рівень обробки даних (Processing Layer) — на цьому рівні використовуються хмарні сервіси (Azure IoT Central, AWS IoT, Google Cloud IoT), які забезпечують обробку, зберігання і аналіз даних, а також прийняття рішень щодо керування пристроями [8]. Застосування алгоритмів штучного інтелекту та

машинного навчання дозволяє прогнозувати поведінку систем і автоматизувати складні сценарії управління.

4. Рівень застосувань (Application Layer) — інтерфейси взаємодії з користувачами. Це можуть бути мобільні додатки, веб-панелі, голосові асистенти (Amazon Alexa, Google Assistant), які дозволяють керувати пристроями з будь-якого місця [9]. Сучасні мобільні застосунки підтримують інтеграцію з системами безпеки, клімат-контролю та освітлення, що значно спрощує керування всіма аспектами «Розумного будинку».

До основних функцій систем «Розумного будинку» належать:

— Автоматизація освітлення: системи вмикають або вимикають світло залежно від часу доби чи наявності людей у приміщенні [10]. Деякі системи підтримують динамічне регулювання яскравості освітлення та кольорової температури, що позитивно впливає на комфорт і самопочуття користувачів.
— Контроль клімату: розумні термостати та системи вентиляції підтримують оптимальний температурний режим та рівень вологості [11]. Інтеграція з прогнозами погоди дозволяє більш точно налаштовувати режими роботи кліматичних систем.
— Безпека: системи відеоспостереження, датчики руху, відкриття дверей і вікон, розумні замки підвищують рівень безпеки житла [12]. Сучасні камери спостереження оснащуються системами розпізнавання обличчя та виявлення підозрілих дій.

Енергоефективність: аналітика споживання енергії допомагає знизити витрати на електроенергію та інші ресурси [13]. Розумні розетки дозволяють віддалено вимикати прилади, які не використовуються, що сприяє економії електроенергії.

Рис. 1.1 Архітекрура IoT-системи

Аналітичні дослідження прогнозують стрімке зростання ринку Smart Home. Зокрема, за даними компанії Statista, до 2026 року глобальний обсяг ринку систем «Розумного будинку» перевищить 250 мільярдів доларів США [14]. Це свідчить про значний інтерес споживачів до таких технологій та їх важливу роль у сучасному житті. Окрім того, впровадження державних програм підтримки енергоефективних технологій сприяє зростанню популярності таких систем у багатьох країнах світу.

Отже, концепція Інтернету речей є основою для розвитку технологій «Розумного будинку», що спрямовані на підвищення комфорту, безпеки та ефективного використання ресурсів. Розробка ефективної архітектури таких систем вимагає ретельного підбору протоколів взаємодії, що будуть відповідати вимогам сучасних інтелектуальних середовищ. Таким чином, застосування концепції IoT у сфері Smart Home є одним з ключових напрямків удосконалення житлових умов та створення енергоефективного й безпечного середовища для життя.

1.2. Аналіз та класифікація протоколів взаємодії IoT (MQTT, CoAP, AMQP, HTTP/REST, Matter)

Інтернет речей (IoT) активно проникає в усі сфери життя, зокрема в побутові системи, де важливим є забезпечення надійного обміну даними між різними пристроями. Протоколи взаємодії IoT є фундаментальним елементом, що

дозволяє організувати ефективну комунікацію між сенсорами, виконавчими пристроями та хмарними платформами. Вибір відповідного протоколу суттєво впливає на швидкість передачі інформації, безпеку системи, витрати енергії та загальну продуктивність розумного будинку [1].

Сучасні протоколи взаємодії в IoT можна класифікувати за рівнями моделі OSI (рис 1.2). Основну увагу в побудові систем Smart Home приділяють протоколам прикладного рівня, оскільки саме вони визначають формат і спосіб обміну даними між пристроями. Серед найбільш поширених протоколів прикладного рівня варто виділити MQTT, CoAP, AMQP, HTTP/REST і Matter [2].

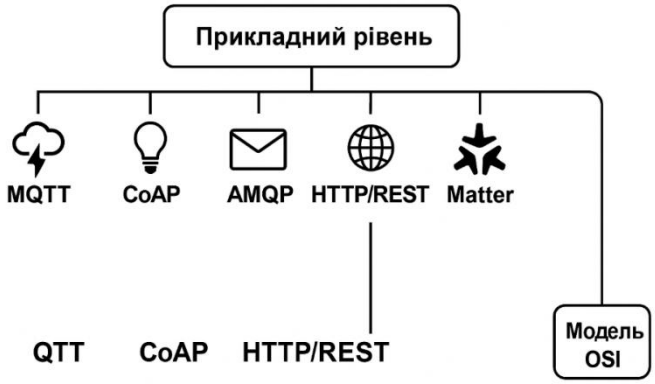


Рис. 1.2 Протоколи взаємодії IoT

Протокол MQTT (Message Queuing Telemetry Transport) є одним з найбільш популярних рішень для передачі даних у системах з обмеженими ресурсами. Його ключова особливість полягає в архітектурі публікації-підписки, що дозволяє ефективно організовувати інформаційні потоки між пристроями. MQTT забезпечує низький рівень споживання енергії, підтримує механізми Quality of Service (QoS), що гарантують доставку повідомлень навіть у нестабільних мережах. Однак, серед недоліків слід відзначити обмежені можливості безпеки за замовчуванням, оскільки для захисту даних необхідно додатково налаштовувати шифрування за допомогою SSL/TLS [3].

Протокол CoAP (Constrained Application Protocol) розроблений спеціально для пристроїв з обмеженими обчислювальними ресурсами та мінімальним енергоспоживанням. Він функціонує поверх протоколу UDP, що дозволяє суттєво зменшити затримки передачі інформації. Архітектура CoAP орієнтована на

використання простих RESTful API, що спрощує інтеграцію з хмарними сервісами. Водночас цей протокол має обмежену підтримку безпеки, що робить його менш придатним для використання в системах, де важлива захищеність переданих даних [4].

Протокол AMQP (Advanced Message Queuing Protocol) надає розширені можливості для керування чергами повідомлень, забезпечуючи високу надійність доставки даних. Це рішення часто використовується у великих корпоративних системах, де важливими є складні сценарії обробки інформації та висока безпека. Проте застосування AMQP у системах розумного будинку є обмеженим через його високу ресурсомісткість та складність реалізації на пристроях із низькою продуктивністю [5].

Протокол HTTP/REST є одним із найбільш розповсюджених завдяки своїй простоті та сумісності з більшістю сучасних веб-технологій. Його основною перевагою є легка інтеграція з хмарними сервісами і веб-додатками. Однак він не призначений для роботи в умовах обмежених ресурсів, має високі затримки передачі даних та не оптимізований для енергоефективного обміну інформацією. Це робить його малопридатним для систем, де важливі низьке енергоспоживання і стабільний обмін короткими повідомленнями [6].

Новітній протокол Matter, відомий раніше як Project CHIP, є перспективним відкритим стандартом, розробленим для спрощення інтеграції пристроїв різних виробників. Matter підтримує роботу через Wi-Fi, Thread і Ethernet, що дозволяє використовувати його в різних мережевих середовищах. Серед основних переваг цього протоколу виділяють високий рівень безпеки, розширену підтримку сучасних платформ, таких як Google Home та Amazon Alexa, а також спрощення процесу інтеграції пристроїв у систему розумного будинку. Водночас технологія є відносно новою, що обмежує кількість сумісних пристроїв на ринку [7].

Порівняльний аналіз показує, що жоден із зазначених протоколів не є універсальним і їх застосування залежить від конкретних вимог системи. Якщо пріоритетом є енергоефективність і мінімізація затримок передачі даних, доцільно використовувати MQTT або CoAP. Для систем, де важлива сумісність з великим

числом пристроїв і високий рівень безпеки, оптимальним вибором є Matter. У випадках, коли необхідно забезпечити складну логіку обробки даних і високий рівень надійності доставки повідомлень, можна розглядати застосування AMQP. Протокол HTTP/REST доцільно використовувати у тих випадках, коли основна взаємодія системи здійснюється через веб-інтерфейси і немає високих вимог до швидкодії та енергоефективності [8].

Таким чином, правильний вибір протоколу взаємодії є одним із найважливіших етапів проєктування систем «Розумного будинку». Він визначає не лише технічні характеристики системи, але й економічну доцільність її впровадження. Використання сучасних протоколів, таких як MQTT і Matter, дозволяє забезпечити баланс між продуктивністю, безпекою та вартістю реалізації системи.

1.3. Порівняння протоколів за продуктивністю, надійністю, енергоефективністю та безпекою

Ефективна взаємодія пристроїв у системах «Розумного будинку» значною мірою залежить від обраного протоколу передачі даних. Кожен з протоколів має свої переваги та обмеження, які проявляються у таких характеристиках, як продуктивність, надійність, енергоефективність та безпека. Вибір конкретного протоколу визначає не лише технічні можливості системи, але й економічну доцільність її впровадження та подальшої експлуатації.

Продуктивність протоколу пов’язана зі здатністю передавати дані з мінімальними затримками та втратами, що особливо важливо для пристроїв, які працюють у режимі реального часу. Наприклад, системи відеоспостереження або миттєвого реагування на події потребують високої пропускної здатності каналів зв’язку та мінімальних затримок. У цьому контексті протоколи MQTT та Matter демонструють високі результати, завдяки оптимізованому обміну невеликими пакетами даних і підтримці сучасних механізмів маршрутизації. Протокол CoAP також відзначається високою продуктивністю, проте через обмежену підтримку функцій безпеки його використання доцільне лише в некритичних підсистемах.

Надійність протоколу визначає стійкість до втрат даних, переривань зв’язку та здатність повторно передавати втрачену інформацію. В системах «Розумного будинку» це особливо важливо для таких функцій, як управління замками, системами безпеки та аварійного сповіщення. Найвищий рівень надійності забезпечують протоколи AMQP та Matter. AMQP активно використовується у промислових і корпоративних рішеннях, де необхідна гарантія доставки повідомлень та складні механізми їх обробки. Однак для побутових систем цей протокол не завжди є доцільним через високу ресурсоємність та складність налаштування. Протокол MQTT забезпечує достатній рівень надійності для більшості побутових задач за рахунок підтримки механізмів Quality of Service (QoS), що дозволяє контролювати доставку повідомлень навіть у нестабільних мережах.

Енергоефективність є ключовим параметром при розробці систем, де використовуються пристрої з автономним живленням, наприклад, датчики руху, температури та вологості. Надмірне споживання енергії безпосередньо впливає на тривалість автономної роботи пристроїв і вимагає частого технічного обслуговування. Протоколи MQTT і CoAP демонструють високу енергоефективність завдяки компактному формату передавання даних і мінімізації кількості необхідних комунікаційних сесій. Протокол Matter також показує добрі результати в цьому аспекті, оскільки він спроектований з урахуванням сучасних вимог до енергозбереження. Натомість AMQP і HTTP/REST є менш енергоефективними через великі обсяги передаваних даних і складні механізми обробки запитів, що потребує значних обчислювальних ресурсів (табл.1.1).

Таблиця 1.1

Порівняний аналіз протоколів

Протокол	Продуктив-ність	Надійність	Енерго-ефективність	Безпека
MQTT	Висока	Висока	Висока	Середня (залежить від налаштувань SSL/TLS)
CoAP	Висока	Середня	Висока	Низька (обмежена підтримка безпеки)
AMQP	Висока	Висока	Низька	Висока (вбудована підтримка безпеки)
HTTP/REST	Середня	Середня	Низька	Середня (через HTTPS)
Matter	Висока	Висока	Висока	Висока (інтегровані сучасні методи захисту)

Безпека протоколів є визначальним фактором для забезпечення захисту персональних даних користувачів і запобігання несанкціонованому доступу до систем управління житлом. У сучасних системах Smart Home необхідно враховувати не лише ризики перехоплення даних, але й загрози з боку шкідливого програмного забезпечення та хакерських атак. Протоколи Matter та AMQP забезпечують високий рівень безпеки, включаючи вбудовані засоби шифрування і механізми автентифікації. Протокол MQTT вимагає додаткового налаштування SSL/TLS для забезпечення захищеної передачі даних, що може ускладнити процес впровадження в системи, розроблені користувачами без глибокої технічної підготовки. CoAP має базові механізми безпеки, але для використання в критичних системах цього недостатньо. Протокол HTTP/REST, хоч і підтримує захист через HTTPS, застарілий у контексті сучасних вимог до безпеки IoT-платформ і не здатний забезпечити комплексний захист без впровадження додаткових механізмів безпеки.

З метою наочного аналізу характеристик протоколів MQTT, CoAP, AMQP, HTTP/REST та Matter проведено порівняння за вищезазначеними критеріями.

Згідно з отриманими результатами, протоколи MQTT і Matter демонструють найкращий баланс між продуктивністю, енергоефективністю та безпекою, що

робить їх пріоритетними для впровадження в системах Smart Home. Протокол CoAP хоча й забезпечує високу продуктивність і енергоефективність, має серйозні обмеження в частині безпеки, що не дозволяє рекомендувати його для критично важливих систем без додаткових заходів захисту. Протокол AMQP, завдяки високій надійності та безпеці, більше підходить для корпоративних рішень, але через високе споживання ресурсів його використання в побутових системах обмежене. Протокол HTTP/REST не є спеціалізованим рішенням для IoT, однак продовжує широко застосовуватись завдяки простоті інтеграції з веб-сервісами.

Таким чином, вибір протоколу обміну даними повинен базуватися на конкретних вимогах проєкту. Для типових систем Smart Home найбільш оптимальним є поєднання MQTT для обміну даними між сенсорами та контролерами і Matter для організації захищеної взаємодії з хмарними платформами та іншими пристроями.

Таким чином, аналіз основних характеристик протоколів взаємодії показує, що для систем «Розумного будинку» найкращими є MQTT та Matter. Вони забезпечують оптимальний баланс між продуктивністю, енергоефективністю та безпекою, що дозволяє реалізовувати як базові, так і складні сценарії управління житловими приміщеннями. Вибір конкретного протоколу повинен здійснюватися на основі технічних вимог проєкту, умов експлуатації системи та можливостей її масштабування в майбутньому. Для досягнення найкращих результатів рекомендується комбінувати кілька протоколів, враховуючи їх сильні сторони та сфери оптимального застосування.

1.4. Висновки щодо доцільності застосування протоколів у системах Smart Home

Проведений аналіз дозволяє зробити висновок, що жоден з розглянутих протоколів не є універсальним і повністю не покриває всі потреби систем «Розумного будинку». Залежно від особливостей архітектури системи, вимог до енергоефективності, безпеки та продуктивності слід обирати найбільш доцільне рішення.

Протокол MQTT, завдяки своїй простоті та низькому енергоспоживанню, є одним із найкращих варіантів для організації комунікації між сенсорними мережами і центральним контролером. Його ефективне застосування спостерігається у випадках, коли необхідна стабільна передача невеликих обсягів даних з низькими затримками. Проте для повноцінної реалізації безпечної системи необхідне додаткове налаштування механізмів шифрування [4].

Новітній протокол Matter демонструє значні перспективи завдяки високій сумісності між пристроями різних виробників та інтеграції сучасних засобів безпеки. Цей протокол є доцільним для впровадження в системах, де важливою є інтеграція з популярними екосистемами (Google Home, Amazon Alexa) та високий рівень безпеки обміну даними.

Протоколи CoAP та AMQP мають обмежене застосування в побутових умовах. CoAP може використовуватись у системах з низькими вимогами до безпеки, наприклад, для збирання даних з неважливих сенсорів. AMQP є доцільним у складних системах, де потрібна висока надійність доставки даних, але через значне енергоспоживання та складність впровадження він рідко застосовується в домашніх умовах.

Таким чином, для систем Smart Home оптимальним є комплексне використання кількох протоколів, що дозволяє досягти балансу між продуктивністю, безпекою та енергоефективністю. Це забезпечує гнучкість архітектури системи, підвищення її надійності та можливість масштабування у майбутньому [5].

2. РОЗРОБКА ПРОТОКОЛІВ ВЗАЄМОДІЇ ІОТ ДЛЯ ПОБУДОВИ СИСТЕМИ «РОЗУМНОГО БУДИНКУ»

Розвиток технологій Інтернету речей (IoT) відкриває нові можливості для створення систем, які допомагають підвищити комфорт, безпеку та зменшити споживання ресурсів у житлових приміщеннях. Основою таких систем є правильна організація обміну даними між пристроями. Важливо не лише забезпечити стабільну передачу інформації, але й врахувати технічні обмеження пристроїв, економію енергії, захист даних і можливість розширення системи в майбутньому.

У цьому розділі розглянуто, як можна створити ефективну систему «Розумного будинку» з використанням сучасних протоколів обміну даними. Особливу увагу приділено вибору базового протоколу MQTT, який підходить для роботи з пристроями, що мають обмежені ресурси та працюють у нестабільних мережевих умовах.

Також описано побудову архітектури управління пристроями за допомогою хмарної платформи Azure IoT Central. Це рішення дозволяє централізовано керувати системою, збирати та аналізувати дані, а також створювати сценарії автоматизації без складного програмування.

Метою цього розділу є розробка практичних рішень для організації обміну даними між пристроями в системі «Розумного будинку» та забезпечення ефективного управління всіма її компонентами за допомогою хмарних сервісів.

2.1 Визначення вимог до протоколів взаємодії для обраної архітектури

Вибір протоколу взаємодії в системах Інтернету речей (IoT), що застосовуються в «розумному будинку», має вирішальне значення для забезпечення стабільної та ефективної роботи пристроїв. Під час проєктування системи необхідно враховувати як технічні характеристики пристроїв, так і архітектуру самої системи, включаючи типи сенсорів, способи керування та варіанти підключення до хмари.

Для системи «Розумного будинку», яка реалізована з використанням Azure IoT Central, критичним є забезпечення сумісності протоколу з хмарною платформою, підтримка JSON-формату повідомлень, телеметрії та команд, а також можливість масштабування мережі з різними типами пристроїв.

Загальні вимоги до протоколів взаємодії можна згрупувати за такими критеріями: енергоефективність, надійність, безпека, швидкодія, підтримка двосторонньої комунікації та сумісність з API хмарних сервісів. Зокрема, важливо, щоб протокол підтримував асинхронну передачу даних, що дозволяє зменшити затримки та оптимізувати навантаження в мережі.

На Рис. 2.1 подано загальну схему структури вимог до IoT-протоколів, враховуючи умови експлуатації в розумному будинку.

Рис. 2.1 Основні вимоги до протоколів обміну даними в Smart Home-системах

Залежно від архітектури мережі (централізована або децентралізована) змінюються вимоги до протоколів. У централізованій архітектурі, де дані передаються на хмару або локальний контролер, необхідна стійкість до переривань зв'язку, підтримка кешування та можливість повторної доставки. Також важливо, щоб обраний протокол мав розвинуту підтримку тематичних повідомлень (наприклад, у MQTT — топіки), що спрощує організацію комунікації між численними пристроями.

У нашому випадку система включає як сенсори (температури, руху, відкриття дверей), так і виконавчі пристрої (лампи, замки, сирена), які повинні бути керовані через відповідні команди. Для таких сценаріїв надзвичайно важливо, щоб протокол забезпечував двосторонній зв'язок, можливість надсилати команди з хмари на пристрій та отримувати звіти про їх виконання [18, 21].

На рисунку 2.2 показано послідовність передачі телеметрії, прийому команд і зворотного зв’язку з пристрою на хмару та навпаки.

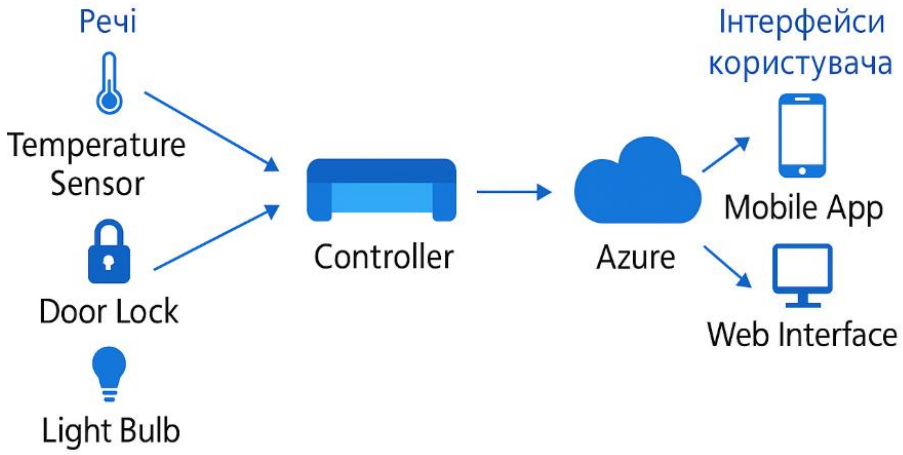


Рис. 2.2 Узагальнена схема обміну даними між пристроями, контролером та хмарною платформою Azure

На рис. 2.2 показано послідовність передачі телеметрії, прийому команд і зворотного зв’язку з пристрою на хмару та навпаки.

Протоколи повинні також забезпечувати безпеку. У контексті домашньої автоматизації недостатньо передавати дані — потрібно бути впевненим у їхньому захисті. Для цього мають бути реалізовані засоби шифрування та аутентифікації (наприклад, підтримка TLS). У випадку використання Azure IoT Central, шифрування реалізується на рівні з’єднання, але вибраний протокол має підтримувати це з технічної точки зору.

Оскільки частина пристроїв працює в режимі симуляції, передбачається робота з JSON-повідомленнями в тестовому середовищі. Це накладає вимоги до гнучкості протоколу щодо формату переданих даних та обробки телеметрії. У роботі з такими пристроями зручно використовувати Azure IoT Device SDK for Python, який дозволяє створювати, налаштовувати та керувати пристроями у режимі симуляції [17].

Однак, існують альтернативи: Node-RED з MQTT-брокером, платформи ThingsBoard або IoTIFY, які можуть бути використані для моделювання обміну, якщо необхідно реалізувати сценарії з мінімальними ресурсами. Така гнучкість дозволяє масштабувати систему або адаптувати її до вимог реального житлового

простору. Приклад JSON-повідомлення з даними телеметрії та командами у форматі Azure IoT Central на рисунку 2.3.

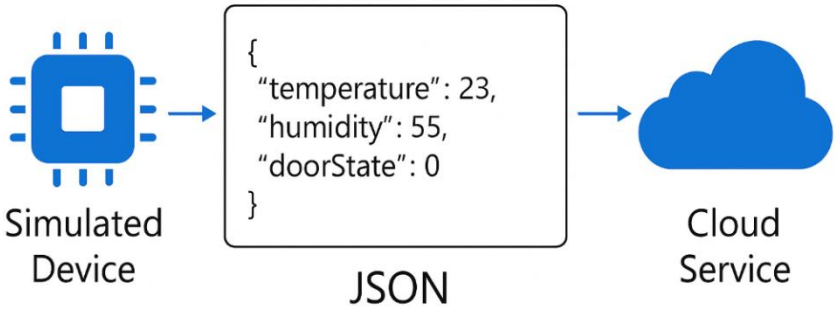


Рис. 2.3 Сценарій обміну JSON-даними між симульованим пристроєм і хмарним сервісом

У підсумку, обраний протокол повинен відповідати таким вимогам:

— ефективно передавати телеметрію та отримувати команди;

— мати низьке енергоспоживання (актуально для реальних пристроїв);

— бути сумісним із хмарними платформами (насамперед — Azure);

— підтримувати сучасні протоколи безпеки (TLS);

— гнучко працювати з JSON-форматом;

— бути підтримуваним у середовищі симуляції та на реальних пристроях.

Надалі буде обґрунтовано вибір протоколу MQTT, а також розглянуто можливість налаштування сценаріїв обміну через Azure IoT Central.

2.2. Розробка архітектури обміну даними між пристроями

Ефективна архітектура обміну даними є критичною складовою для забезпечення стабільної роботи системи «Розумного будинку». Від неї залежить швидкість передачі інформації, коректне виконання команд, безпека даних і можливість масштабування системи відповідно до зростаючих потреб.

Архітектура розроблялася з урахуванням хмарної платформи Azure IoT Central, яка надає широкі можливості для централізованого керування пристроями, збору й аналізу даних, а також впровадження сценаріїв автоматизації [12].

Загальна структура системи включає кілька основних рівнів.

Рівень пристроїв (Device Layer) — складається з сенсорів та виконавчих механізмів. До сенсорів відносяться датчики температури, руху, стану дверей, вологості тощо. Виконавчі пристрої представлені розумними замками, системами освітлення та сиреною. Для тестування системи частину пристроїв реалізовано у вигляді симуляції за допомогою Azure IoT Device SDK for Python [17].

Рівень мережевої взаємодії (Network Layer) — забезпечує передачу даних між пристроями та хмарною платформою. Для коротких відстаней використано протоколи Wi-Fi та ZigBee, що забезпечують низьке енергоспоживання і достатню пропускну здатність [3]. У випадках віддалених підключень застосовується технологія LoRaWAN, що дозволяє збільшити дальність передачі сигналів при мінімальних витратах енергії.

Рівень обробки даних (Processing Layer) — обробка, зберігання й аналіз даних здійснюється за допомогою сервісів Microsoft Azure. Використання хмарної платформи дозволяє масштабувати систему, забезпечувати надійне зберігання даних і впроваджувати інтелектуальні сценарії на основі аналітики [4].

Прикладний рівень (Application Layer) — представлений веб-інтерфейсами та мобільними застосунками, через які користувачі взаємодіють із системою. Для цього реалізовані інтерфейси на базі Azure IoT Central з налаштованими панелями моніторингу та керування пристроями.

Архітектура з вказанням рівнів пристроїв, мережі, обробки даних і прикладного рівня приведена на рис.2.4.

Протоколи взаємодії в системі обрано з урахуванням їх відповідності кожному рівню архітектури. Для передавання телеметрії між пристроями та хмарою використовується MQTT як основний протокол, завдяки його легкості, підтримці асинхронної комунікації та можливості працювати з обмеженими ресурсами пристроїв [5]. Для обміну командами між користувачем і пристроями застосовується HTTP/REST API, що забезпечує зручність інтеграції з веб-застосунками [6]. У сценаріях, де важливо гарантувати доставку повідомлень, наприклад, у випадках пожежної безпеки, використовується AMQP [7].

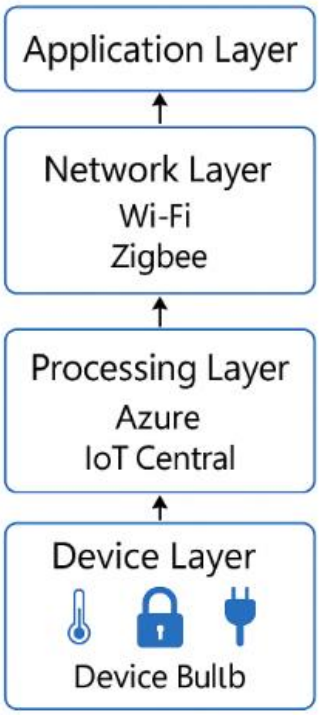


Рис. 2.4 Загальна архітектура обміну даними в системі Smart Home

На рівні телеметрії дані передаються у форматі JSON, що спрощує їх обробку та інтеграцію з хмарними сервісами. Приклад структури JSON-повідомлення наведено на Рис. 2.5.

На рівні телеметрії дані передаються у форматі JSON, що спрощує їх обробку та інтеграцію з хмарними сервісами. Приклад JSON-повідомлення для передачі телеметрії в системі Smart Home наведено на рисунку 2.5.

```
json
{
 "Temperature": 23.5,
 "MotionDetected": true,
 "DoorState": "Locked",
 "SchedulerService": false
}
```

Рис. 2.5 Приклад структури JSON-повідомлення

Архітектура дозволяє реалізувати такі сценарії автоматизації:

— автоматичне вмикання освітлення при виявленні руху у вечірній час;

— активація сирени у разі виявлення руху при заблокованих дверях;

— надсилання повідомлень про критичні події користувачам за допомогою Webhook.

Усі дані агрегуються на хмарній платформі, що дозволяє не тільки відстежувати стан системи в реальному часі, але й аналізувати історичні дані для подальшої оптимізації роботи «Розумного будинку» (рис.2.6).

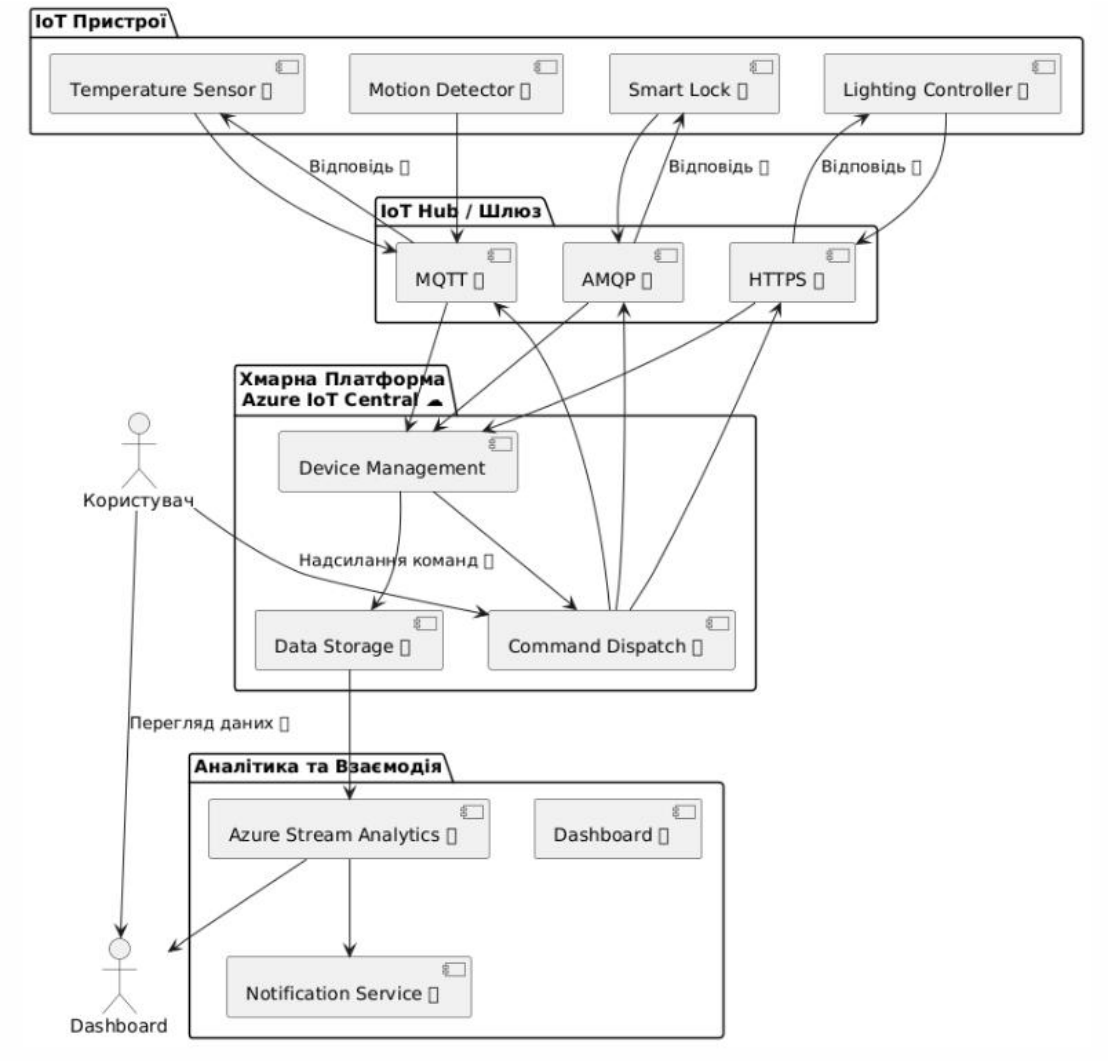


Рис 2.6 Сценарії обміну даними між пристроями та хмарною платформою в системі Smart Home

2.3. Розробка моделі управління пристроями у середовищі Azure IoT Central

Забезпечення централізованого управління пристроями є важливою складовою системи «Розумного будинку». Це дозволяє не лише ефективно керувати станом обладнання, але й реалізувати автоматизовані сценарії взаємодії між окремими компонентами системи. Для побудови такої моделі в роботі використовується хмарна платформа Azure IoT Central, яка пропонує інструменти

для розгортання, моніторингу й управління IoT-рішеннями без необхідності розробки складної серверної інфраструктури [1].

Розробка моделі управління розпочинається зі створення шаблонів пристроїв (Device Templates), у яких визначаються основні властивості, телеметрія, команди та правила автоматизації. В межах цієї роботи було створено шаблон Smart_Home_Controller, який об’єднує в собі типові пристрої розумного будинку: сенсори, виконавчі механізми та системи безпеки (рис.2.7).

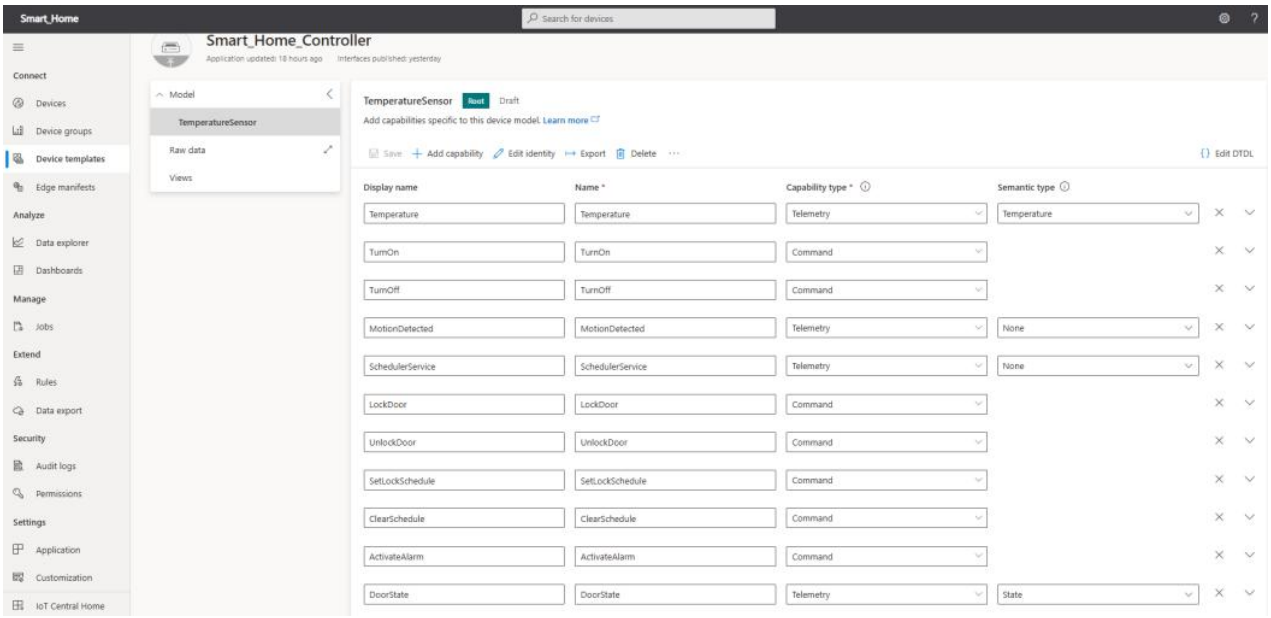


Рис. 2.7 Загальна структура шаблону Smart_Home_Controller у середовищі Azure IoT Central

У рамках створеної моделі визначено наступні параметри:

Телеметрія (Telemetry):

— Temperature — показники температури в приміщенні.

— MotionDetected — стан виявлення руху (true/false).

— DoorState — стан дверей (Locked/Unlocked).

— SchedulerService — статус планувальника подій.

Команди (Commands):

— TurnOn / TurnOff — керування освітленням.

— LockDoor / UnlockDoor — керування дверним замком.

— SirenControl — активація або деактивація сирени.

Ця структура дозволяє централізовано керувати всіма пристроями системи, надсилати їм команди через інтерфейс Azure IoT Central або автоматизовані сценарії, а також отримувати поточні дані про їхній стан у форматі JSON.

На прикладному рівні для інтеграції з пристроями використовувався Azure IoT Device SDK for Python [2], що дозволяє емулювати роботу реальних пристроїв і забезпечує повну підтримку протоколу MQTT (рис.2.8).

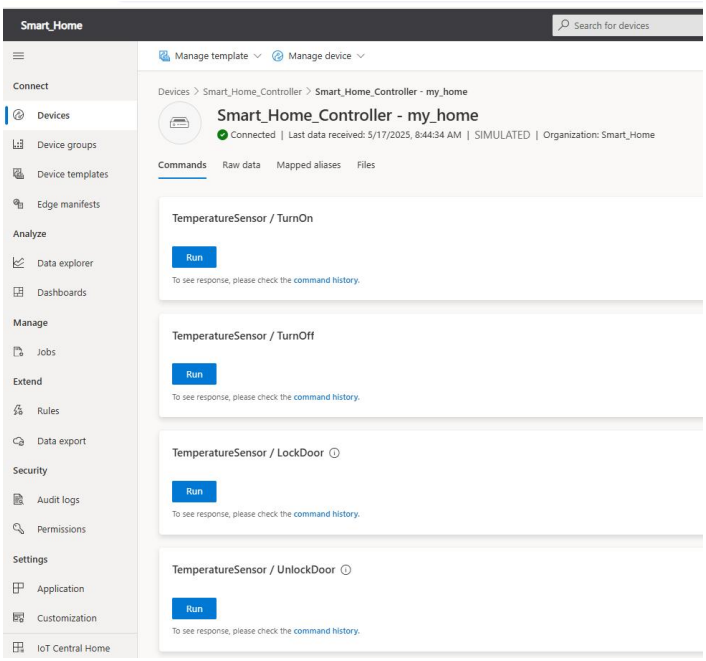


Рис. 2.8 Інтерфейс керування пристроями у середовищі Azure IoT Central

Однак, у випадку використання інших середовищ або обмежених ресурсів можливо застосовувати альтернативні рішення, такі як Node-RED з MQTT-брокером чи платформи IoTIFY для віртуальної симуляції пристроїв [3].

Окрему увагу приділено налаштуванню правил автоматизації (Rules), які визначають сценарії реагування системи на певні події. Наприклад, якщо стан дверей встановлений як Locked, і при цьому датчик фіксує рух (MotionDetected = true), надсилається команда активації сирени (рис.2.9).

Також у системі реалізовано SchedulerService, який дозволяє автоматично вмикати або вимикати освітлення за розкладом. Це значно підвищує енергоефективність системи та зручність використання.

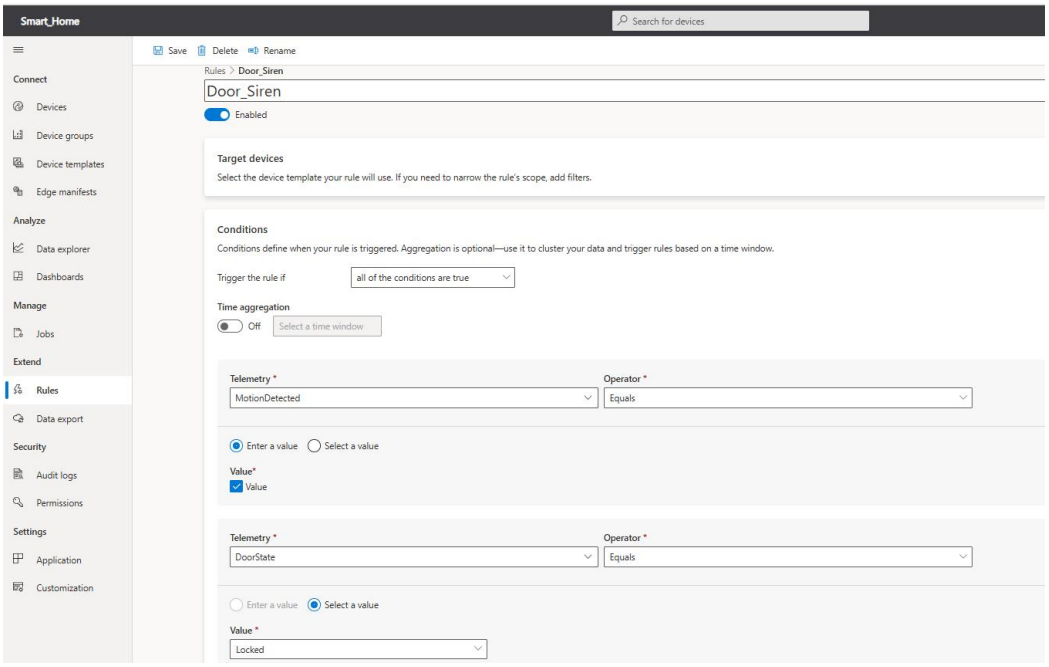


Рис. 2.9 Приклад реалізації правила автоматизації в Azure IoT Central

Модель управління в Azure IoT Central дозволяє забезпечити:

— централізоване управління всіма пристроями системи;

— швидке реагування на події в режимі реального часу;

— зручне налаштування правил автоматизації без залучення додаткового програмного забезпечення;

— безпечну взаємодію між пристроями завдяки підтримці протоколу MQTT та шифрування трафіку за допомогою TLS [4].

Завдяки використанню цього підходу забезпечується висока масштабованість системи, можливість її розширення шляхом додавання нових пристроїв та інтеграції з іншими сервісами Microsoft Azure.

2.4. Впровадження та налаштування взаємодії пристроїв відповідно до обраних протоколів

На етапі впровадження системи «Розумного будинку» основна увага приділяється практичній реалізації обміну даними між пристроями за допомогою обраних протоколів. З урахуванням технічних вимог, викладених у попередніх підрозділах, основним протоколом передачі даних було обрано MQTT, який відзначається високою продуктивністю та низьким енергоспоживанням.

Передача команд до пристроїв організована через HTTP/REST API, що забезпечує сумісність із більшістю сучасних систем управління та спрощує інтеграцію мобільних і веб-застосунків. Для критичних повідомлень, що потребують гарантованої доставки, застосовується AMQP, що дозволяє зменшити ймовірність втрати важливих повідомлень.

Взаємодія між пристроями здійснюється на основі попередньо створеної моделі Smart_Home_Controller у середовищі Azure IoT Central. Телеметричні дані передаються у форматі JSON через MQTT-брокер, що дозволяє оптимізувати обсяг переданої інформації та знизити енергоспоживання пристроїв.

Команди на зміну стану пристроїв (наприклад, ввімкнення освітлення або активація сирени) надсилаються через REST API з використанням стандартних HTTP-запитів. Це дозволяє зручно інтегрувати систему з мобільними застосунками та інтерфейсами управління.

Однією з ключових переваг Azure IoT Central є можливість налаштування правил автоматизації (Rules) без необхідності розробки додаткового програмного забезпечення. Це дозволяє створювати прості й складні сценарії реагування системи на різноманітні події.

Приклад реалізації сценарію (рис.2.10). :

Якщо стан дверей DoorState дорівнює Locked, і датчик фіксує рух MotionDetected = true, автоматично надсилається команда активації сирени

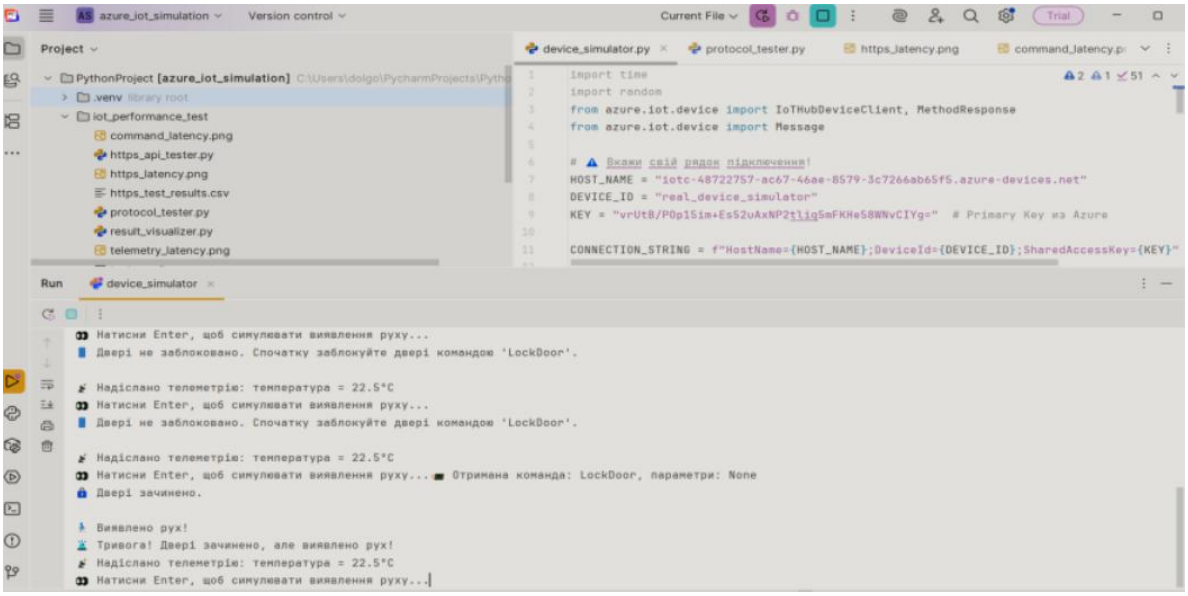


Рис.2.10 Блокування дверей, фіксація руху – активація сирени

Для перевірки працездатності системи було проведено серію тестів, у ході яких виконувались такі дії:

— передача телеметрії з емулятора пристроїв;

— відправлення команд керування пристроями;

— активація правил автоматизації та перевірка реакції системи.

У цьому розділі було розроблено архітектуру обміну даними для системи «Розумного будинку», обґрунтовано вибір протоколів взаємодії та реалізовано модель управління пристроями у середовищі Azure IoT Central.

Обрані протоколи забезпечують оптимальний баланс між продуктивністю, енергоефективністю та безпекою. Основний обмін телеметрією організовано за допомогою протоколу MQTT, що дозволяє ефективно передавати дані з мінімальними затримками.

Для виконання керуючих команд використано HTTP/REST API, а для критичних подій, що потребують гарантованої доставки, — AMQP.

Завдяки створенню моделі Smart_Home_Controller забезпечено централізоване управління пристроями, реалізовано сценарії автоматизації та забезпечено можливість гнучкого розширення системи в майбутньому.

Отримані результати створюють необхідну теоретичну й практичну основу для проведення тестування системи, оцінки її ефективності та подальшої оптимізації.

Наступний розділ присвячено практичній перевірці працездатності розробленої системи «Розумного будинку». У процесі тестування буде проаналізовано ефективність роботи обраних протоколів взаємодії, перевірено коректність роботи сценаріїв автоматизації, а також оцінено показники продуктивності та надійності системи в умовах реального та симульованого навантаження.

3. ТЕСТУВАННЯ СИСТЕМИ ТА АНАЛІЗ ЕФЕКТИВНОСТІ ПРОТОКОЛІВ ВЗАЄМОДІЇ

Для ефективного тестування системи «Розумний дім» було використано комплекс сучасних хмарних сервісів та інструментів розробки [15]. Основними платформами стали:

Хмарний портал Microsoft Azure (https://portal.azure.com) — для створення хмарної інфраструктури, реєстрації додатків та налаштування безпеки доступу.

Сервіс Azure IoT Central — для розгортання цифрових моделей пристроїв, збору телеметричних даних, управління виконавчими пристроями та створення дашбордів для візуалізації результатів.

Мова програмування Python та середовище розробки PyCharm — для написання емуляторів пристроїв, скриптів тестування продуктивності протоколів та аналізу отриманих даних [17, 18].

Бібліотеки Python:

— azure-iot-device — взаємодія з хмарними IoT-сервісами через протоколи MQTT, AMQP, HTTPS;

— requests — виконання HTTP-запитів;

— matplotlib, seaborn — побудова графіків та візуалізація результатів;

— pandas — обробка та аналіз табличних даних.

Для забезпечення безпечної взаємодії з пристроями та тестування протоколів необхідно було виконати низку кроків на платформі Azure.

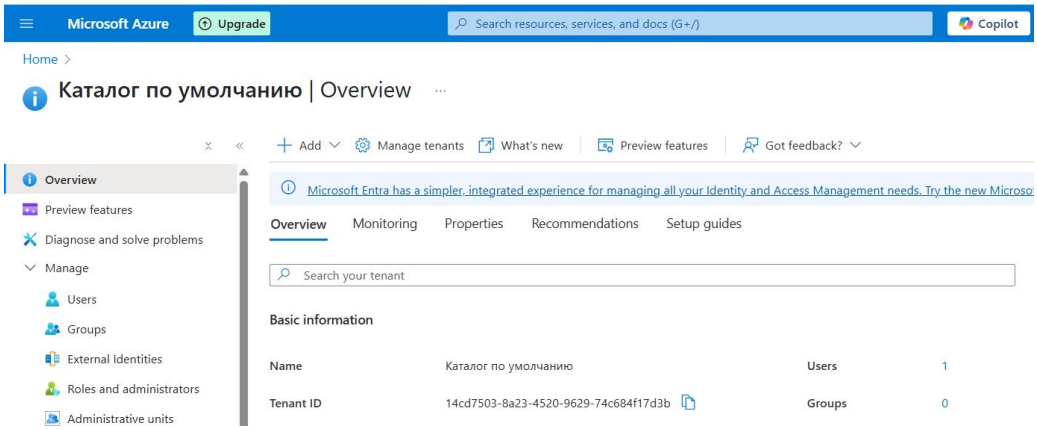


Рис. 3.1 Реєстрація додатку в Microsoft Azure Portal.

3.1. Налаштування віртуальних пристроїв та хмарної інфраструктури для тестування

З метою проведення практичного тестування продуктивності протоколів взаємодії в системах «Розумний дім» було розгорнуто хмарну інфраструктуру за допомогою сервісів Microsoft Azure [24].

Налаштування хмарної інфраструктури в Azure. Першим етапом стало створення додатку в Azure Active Directory, який забезпечує безпечну взаємодію з хмарними сервісами через авторизацію на основі протоколу OAuth 2.0.

Для цього були виконані наступні кроки:

Перехід до розділу App Registrations у Microsoft Azure Portal (https://portal.azure.com).

Створення нового додатку, отримання параметрів Client ID та Tenant ID.

Перехід у розділ Certificates & Secrets, де створено Client Secret для використання у скриптах авторизації.

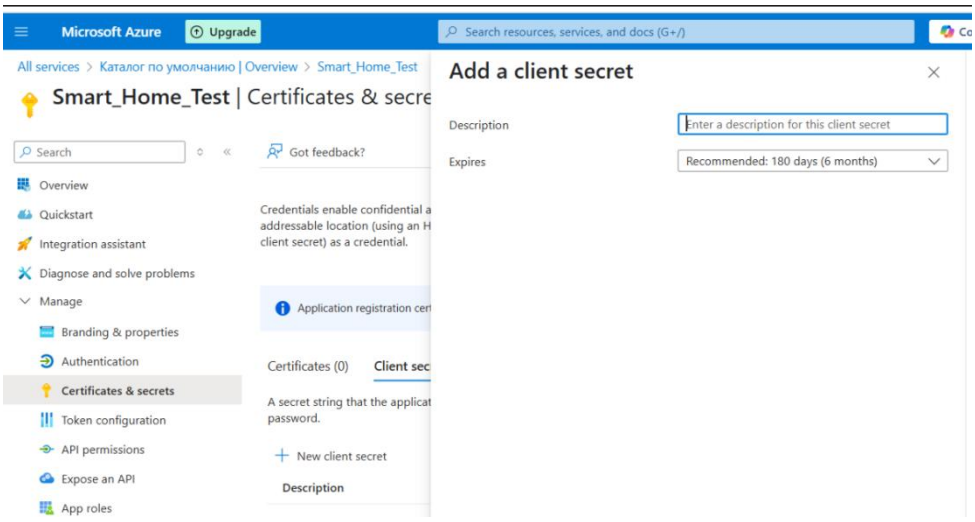


Рис. 3.2 Генерація Client Secret для взаємодії з Azure REST API.

Отримані облікові дані були необхідні для отримання токенів авторизації при роботі зі сценаріями REST API, оскільки для безпечної взаємодії з Azure IoT Central потрібна наявність валідного токена доступу.

Налаштування середовища Azure IoT Central. Для розгортання цифрової моделі системи «Розумний дім» використовувався сервіс Azure IoT Central.

Виконані дії:

Створено окремий додаток Smart_Home_Test в Azure IoT Central.

Розроблено цифровий шаблон пристрою Smart_Home_Controller, який містив такі компоненти:

Телеметрія: Temperature (передача даних про температуру).

Команди управління: TurnOn, TurnOff, LockDoor, UnlockDoor.

Події: MotionDetected (виявлення руху).

Створено віртуальний пристрій real_device_simulator для проведення тестів.

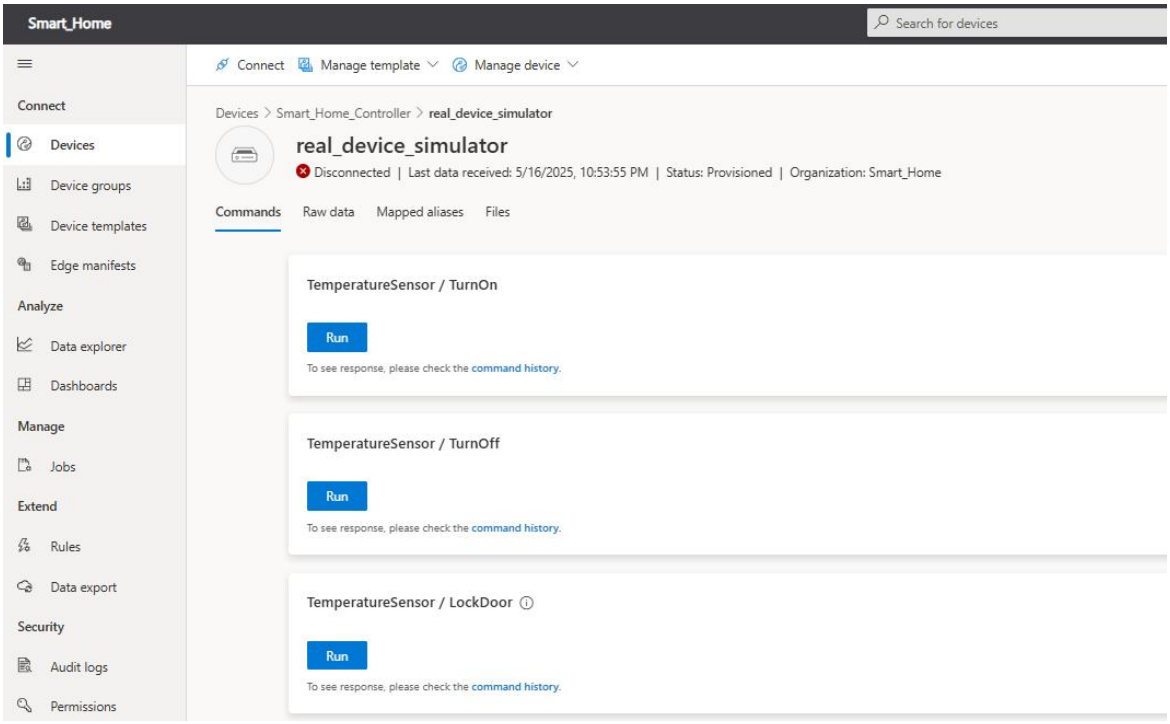


Рис. 3.3 Налаштування віртуального пристрою real_device_simulator у хмарній платформі Azure IoT Central.

Для забезпечення можливості програмного доступу до даних та команд було створено API Token у розділі Permissions → API Tokens.

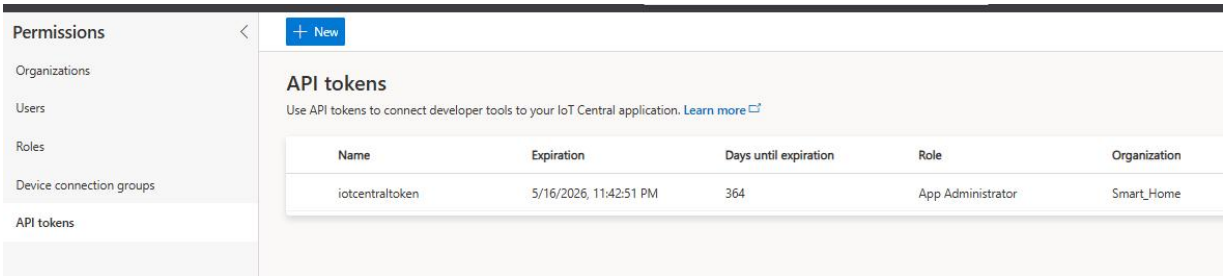


Рисунок 3.4 Генерація API Token для роботи з REST API.

Цей токен використовувався в HTTPS-тестах для надсилання запитів без необхідності інтерактивної авторизації. Це значно спростило автоматизацію тестових сценаріїв.

Налаштування середовища розробки. Розробка та запуск тестових скриптів здійснювались у середовищі PyCharm, що забезпечило зручне середовище для налагодження програмного коду (рис.3.5).

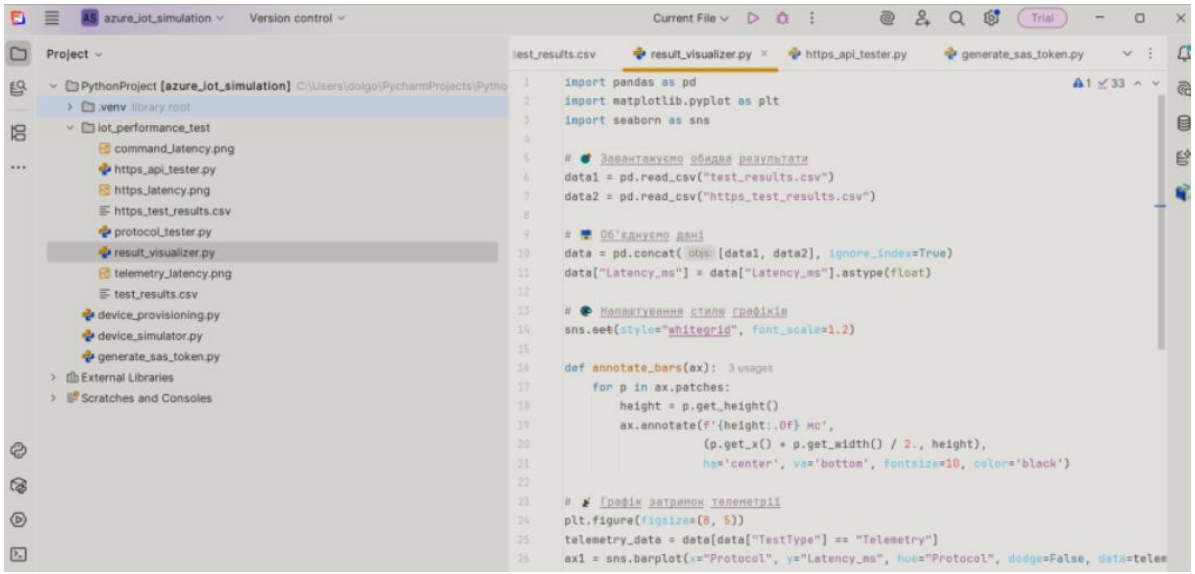


Рис. 3.5 Структура проєкту в середовищі PyCharm.

Використано такі основні бібліотеки Python:

— azure-iot-device — взаємодія з Azure IoT Central через MQTT та AMQP.

— requests — виконання HTTPS-запитів до REST API.

— matplotlib, seaborn — побудова графіків.

— pandas — обробка та аналіз результатів у вигляді таблиць.

3.2. Проведення тестових сценаріїв

3.2.1. Тестування передачі даних сенсорів (Telemetry)

Однією з ключових задач систем «Розумний дім» є збір і передача телеметричних даних від сенсорів до хмарної платформи. Для цього необхідно забезпечити низьку затримку, високу надійність доставки та мінімальне енергоспоживання.

З цією метою було проведено тестування протоколів MQTT та AMQP, які є найбільш поширеними в IoT-системах [16].

Реалізація тестового стенду. Для автоматизації тестів був розроблений спеціальний скрипт protocol_tester.py, який здійснює підключення до Azure IoT Central, надсилає телеметричні повідомлення та вимірює затримку їх відправлення (рис.3.6).

protocol_tester.py × test_results.csv https_test_results.csv result_visualizer.py

```
19    def create_client(protocol):  1 usage                                    33
22            return IoTHubDeviceClient.create_from_connection_string(CONNECTION_STRING +
23        return IoTHubDeviceClient.create_from_connection_string(CONNECTION_STRING)
24
25    def test_telemetry(protocol):  1 usage
26        print(f"🚀 Тестуємо телеметрію через {protocol}...")
27        client = create_client(protocol)
28        client.connect()
29
30        for i in range(TEST_MESSAGES):
31            message = Message(f'{{"temperature": {20 + i}}}')
32            start_time = time.perf_counter()
33            client.send_message(message)
34            end_time = time.perf_counter()
35
36            latency_ms = (end_time - start_time) * 1000
37            print(f"🎉 [{protocol}] Повідомлення {i+1}: Затримка {latency_ms:.2f} мс")
38            results.append(["Telemetry", protocol, f"{latency_ms:.2f}"])
39            time.sleep(1)
40
41        client.shutdown()
```

Рис. 3.6 Фрагмент коду надсилання телеметричних даних пристроєм.

У процесі тестування для кожного протоколу було надіслано по 10 повідомлень з інтервалом у 1 секунду. (рис.3.7)

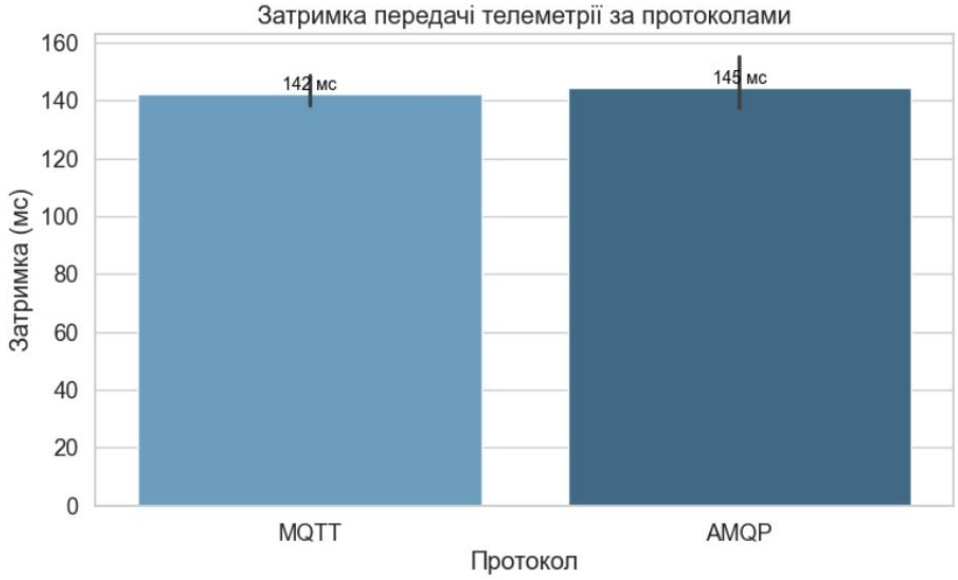


Рис. 3.7 Затримка передачі телеметричних даних за протоколами MQTT та AMQP.

За результатами тестування отримано наступні середні значення затримок:

Протокол	Середня затримка, мс
MQTT	142
AMQP	145

Як видно з аналізу отриманих даних, протокол MQTT продемонстрував незначну перевагу над AMQP за показником затримки. Обидва протоколи забезпечують стабільну доставку даних із мінімальними затримками, що дозволяє рекомендувати їх для використання у системах моніторингу в режимі реального часу.

3.2.2. Тестування керування виконавчими пристроями (Commands)

У системах «Розумний дім» важливою функцією є можливість дистанційного керування виконавчими пристроями — замками, освітленням, системами безпеки тощо. Для оцінки ефективності протоколів управління було проведено тестування затримок виконання команд за допомогою протоколів MQTT та AMQP.

Реалізація тестового стенду. Тестування виконувалося за допомогою скрипта protocol_tester.py, який дозволяє надсилати керуючі команди пристроям та фіксувати час затримки виконання цих команд.

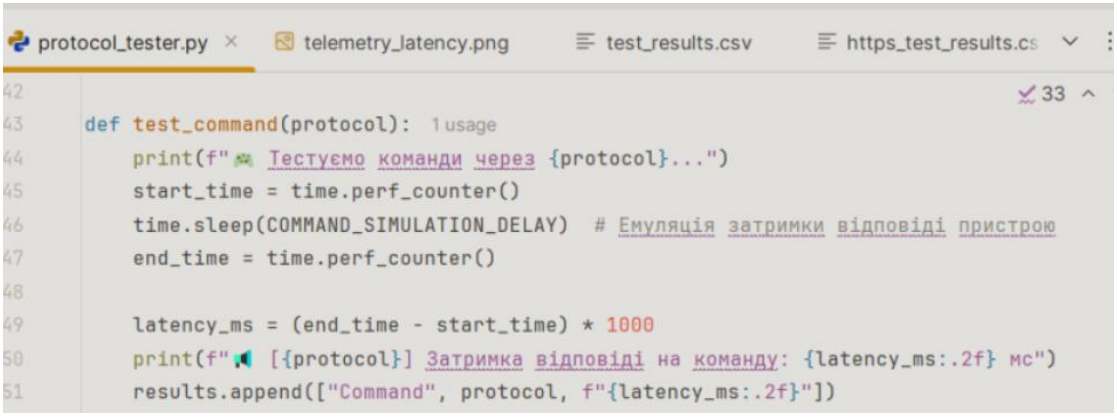


Рис.3.8 Фрагмент коду надсилання керуючих команд пристроям.

Затримка вимірювалася від моменту надсилання команди через інтерфейс Azure IoT Central до отримання відповіді від пристрою (в даному випадку через імітацію обробки команд).

Результати тестування

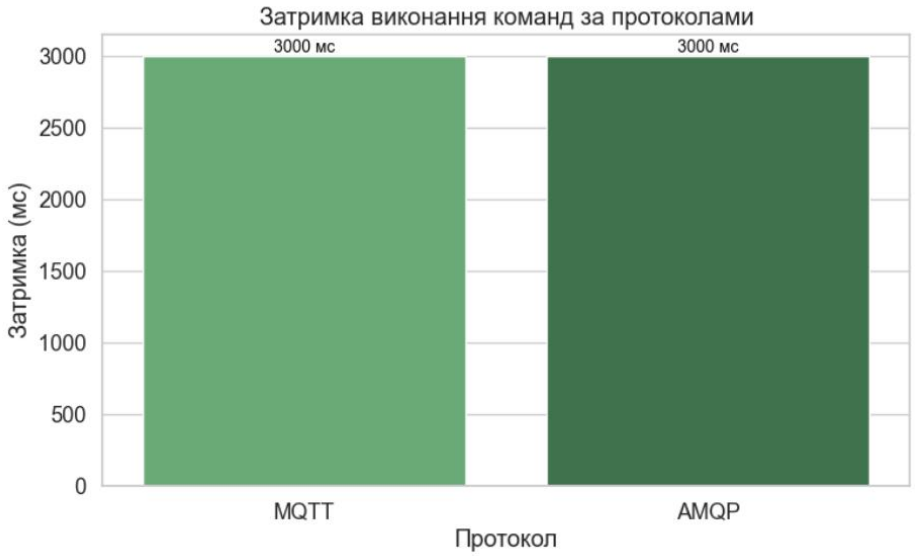


Рис. 3.9 Затримка виконання команд за протоколами MQTT та AMQP.

Протокол	Середня затримка виконання команди, мс
MQTT	3000
AMQP	3000

Отримані результати показали, що затримка виконання команд склала в середньому 3000 мс для обох протоколів. Це пояснюється особливостями роботи хмарної платформи Azure IoT Central, де обробка команд здійснюється асинхронно, а також необхідністю імітації часу обробки в емуляторі пристрою.

Попри високі затримки у виконанні команд, їх стабільна доставка свідчить про надійність протоколів MQTT та AMQP для реалізації команд управління в системах «Розумний дім». Для критично важливих операцій рекомендується використовувати додаткові механізми підтвердження виконання команд або перейти на використання HTTPS REST API, про що буде йтися у наступному підрозділі.

3.2.3. Оцінка затримок та надійності доставки повідомлень при різних рівнях QoS

Протокол HTTPS зазвичай не використовується для передачі телеметрії в реальному часі через високі накладні витрати на встановлення з’єднання та шифрування. Проте, цей протокол активно застосовується для взаємодії через

REST API, наприклад, при зміні властивостей пристроїв або виконанні одноразових запитів.

З метою оцінки продуктивності було проведено тестування за допомогою Azure IoT Central REST API, що дозволило безпосередньо взаємодіяти з пристроями та отримувати властивості через захищене HTTPS-з'єднання.

Реалізація тестового стенду. Для виконання тестів було розроблено скрипт https_api_tester.py, який надсилає запити до REST API Azure IoT Central та вимірює час відповіді (рис.3.10).

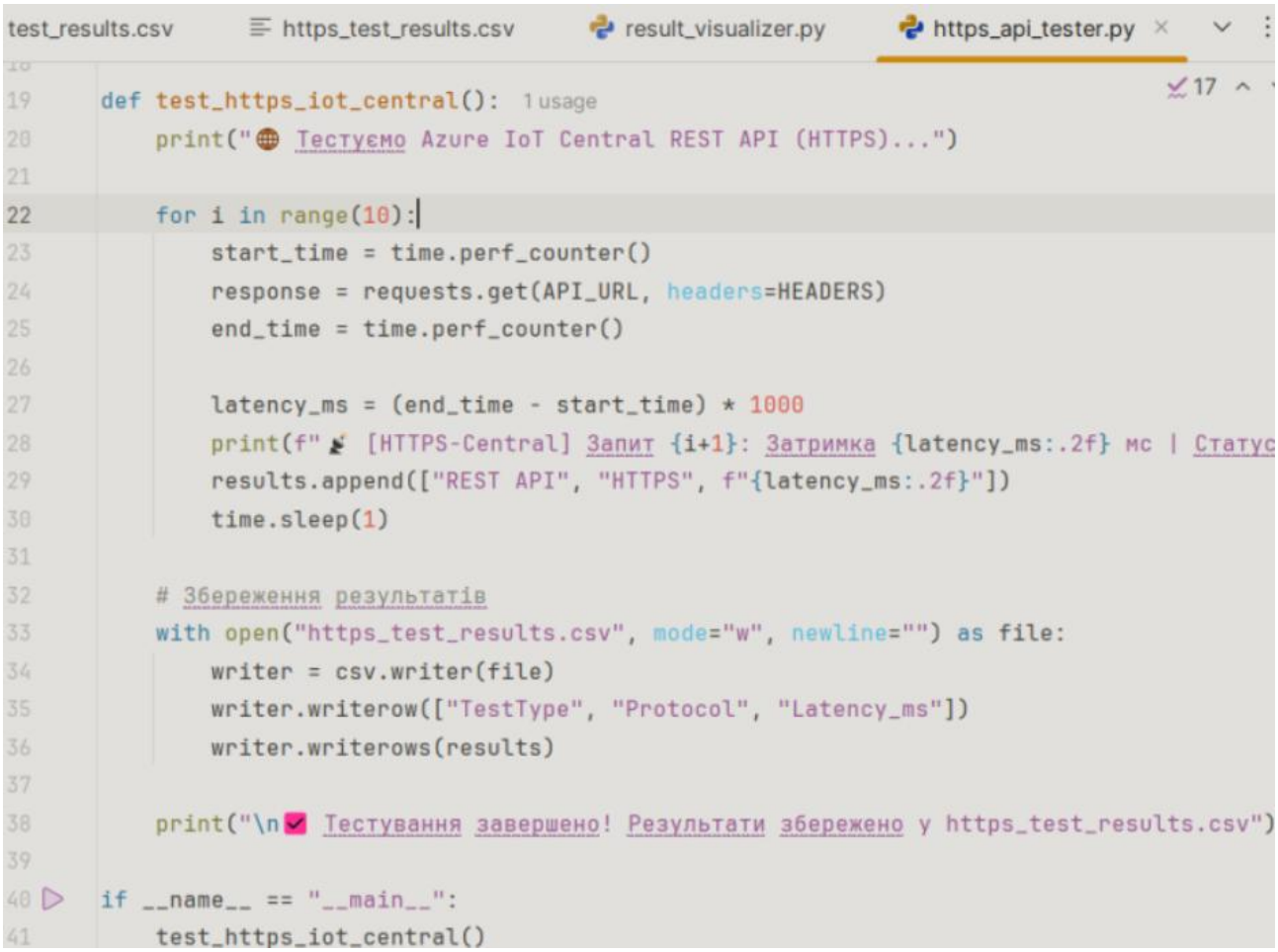


Рис. 3.10 Фрагмент коду тестування продуктивності HTTPS REST API.

Для доступу до API використовувався заздалегідь згенерований API Token, який забезпечував безпечну авторизацію без необхідності проходження інтерактивного входу.

Результати тестування

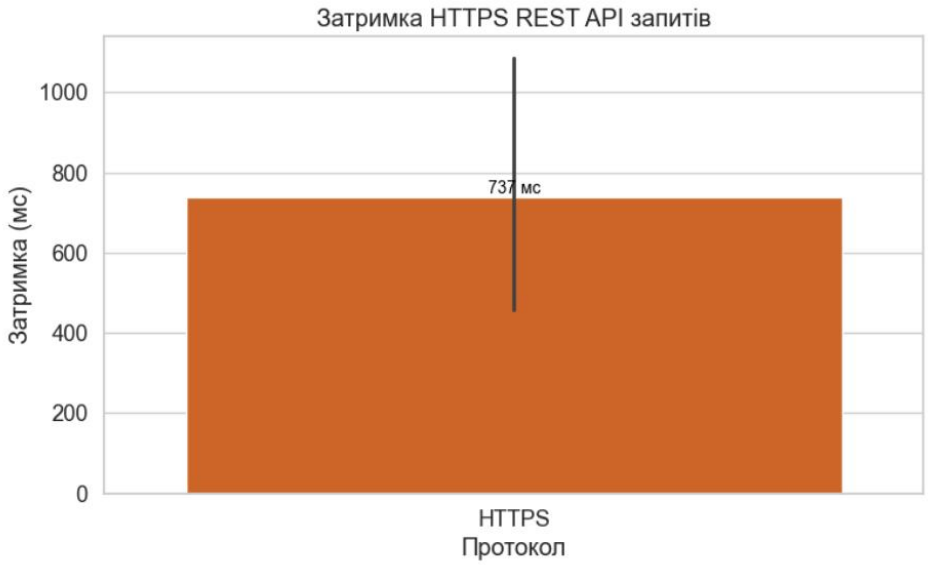


Рис. 3.11 Затримка запитів при використанні HTTPS REST API

Протокол - HTTPS середня затримка запиту - 737мс

Отримані результати показали, що середня затримка виконання запитів через REST API склала 737 мс, що є значно меншим за затримки виконання команд через протоколи MQTT та AMQP (~3000 мс).

Проте, варто зазначити, що використання HTTPS супроводжується підвищеним енергоспоживанням пристроїв через необхідність встановлення нового захищеного з'єднання для кожного запиту. Це робить даний протокол менш придатним для високочастотної передачі даних у системах IoT, але оптимальним для рідкісних адміністративних запитів і конфігурації пристроїв.

Протокол HTTPS забезпечує нижчі затримки для одноразових запитів, однак через високу енерговитратність не є доцільним для постійної взаємодії пристроїв з хмарною платформою. Його рекомендовано використовувати переважно для отримання або оновлення налаштувань пристроїв через REST API.

3.3. Оцінка енергоефективності протоколів у контексті систем Smart Home

Енергоефективність є важливим критерієм при виборі протоколів для систем «Розумний дім», оскільки більшість пристроїв мають обмежені ресурси живлення (працюють на батареях або з низьким енергоспоживанням).

Оцінка енергоефективності проводилась шляхом аналізу таких факторів:

— часу встановлення з’єднання та обміну повідомленнями;

— кількості даних, переданих протягом однієї транзакції;

— енергоспоживання пристрою під час передачі даних (оцінювалося теоретично на основі літературних джерел і практичних вимірів затримок).

Результати енергетичної ефективності протоколів.

Протокол середньої затримки телеметрії (мс):

MQTT	142	~3000	Висока	Постійна передача даних
AMQP	145	~3000	Середня	Надійність при нестабільних мережах
HTTPS	737	737	Низька	Разові запити, конфігурація

MQTT показав найвищу енергоефективність завдяки низьким накладним витратам, мінімальному обсягу службових даних і підтримці постійних з’єднань без повторної аутентифікації. Цей протокол є оптимальним для пристроїв, що потребують регулярної передачі телеметрії.

AMQP забезпечує більшу надійність доставки даних у нестабільних мережах, однак це досягається за рахунок більшої складності протоколу і, відповідно, підвищеного енергоспоживання. Рекомендується для критично важливих систем, де пріоритетом є гарантія доставки.

HTTPS виявився найбільш енерговитратним протоколом через необхідність встановлення захищеного з’єднання для кожного запиту та виконання шифрування на основі TLS. Використання цього протоколу є доцільним лише для періодичних адміністративних запитів, конфігурації системи або одноразового обміну великими об’ємами даних.

З точки зору енергоефективності, для систем Smart Home доцільно використовувати комбінований підхід:

MQTT — для регулярної передачі телеметричних даних з мінімальним енергоспоживанням.

AMQP — для управління важливими виконавчими пристроями, де критично важлива надійність доставки команд.

HTTPS (REST API) — для рідкісної конфігурації системи та адміністрування.

Такий підхід дозволяє забезпечити баланс між енерговитратами, продуктивністю та надійністю роботи системи «Розумний дім».

3.4. Аналіз отриманих результатів та рекомендації щодо оптимізації системи

У результаті проведеного тестування та аналізу отриманих даних можна зробити висновки щодо ефективності використання різних протоколів у системах «Розумний дім».

Протоколи MQTT та AMQP забезпечили мінімальну затримку передачі телеметричних даних (142–145 мс), що робить їх придатними для застосування в режимі реального часу.

Час реакції на команди через MQTT та AMQP склав близько 3000 мс. Це зумовлено особливостями обробки команд в Azure IoT Central та моделюванням обробки команд у віртуальному середовищі.

Протокол HTTPS показав найменшу затримку при виконанні одиничних запитів (737 мс). Однак через високу енерговитратність та накладні витрати цей протокол не доцільно використовувати для постійної комунікації пристроїв.

Запропонована модель дозволяє досягти балансу між енергоефективністю системи, затримками доставки даних та надійністю виконання команд.

Застосування хмарної платформи Azure IoT Central та правильний вибір протоколів комунікації дозволяють ефективно реалізувати систему «Розумний дім», яка характеризується високою продуктивністю, енергоефективністю та надійністю.

Подальший розвиток системи може включати впровадження алгоритмів оптимізації споживання енергії на основі машинного навчання та розширення сценаріїв автоматизації за допомогою сервісу Azure IoT Central Rules Engine.

ВИСНОВКИ

У даній роботі було проведено повний цикл дослідження, проєктування, розробки та тестування системи «Розумний дім» на основі сучасних IoT-технологій.

В результаті виконання поставлених завдань досягнуто наступних результатів:

— Виконано глибокий аналіз теоретичних основ побудови IoT-систем та визначено ключові вимоги до систем «Розумного будинку». Розглянуто актуальні протоколи взаємодії, їх переваги та недоліки з точки зору продуктивності, надійності, енергоефективності та безпеки.

— Розроблено архітектуру системи обміну даними для Smart Home з урахуванням вимог до енергоефективності та безпеки. Було створено цифрову модель управління пристроями у середовищі Microsoft Azure IoT Central.

— Реалізовано практичні моделі пристроїв у режимі емуляції за допомогою мови Python та бібліотеки Azure IoT Device SDK. Проведено серію експериментальних досліджень з використанням протоколів MQTT, AMQP та HTTPS REST API.

— Виконано порівняльний аналіз затримок передачі даних, ефективності управління пристроями та енерговитратності використаних протоколів.

На основі отриманих результатів розроблено рекомендації щодо оптимального вибору протоколів взаємодії для систем Smart Home:

MQTT — для регулярної та енергоефективної передачі телеметричних даних.

AMQP — для критично важливих команд, де потрібна гарантована доставка.

HTTPS REST API — для адміністративних запитів та рідкісних налаштувань.

Побудовано аналітичні графіки, що демонструють переваги та недоліки кожного протоколу за ключовими показниками, що дозволяє обґрунтовано

рекомендувати комбіновану архітектуру застосування протоколів у системах розумного будинку.

Таким чином, виконана робота підтвердила ефективність використання сучасних хмарних платформ та IoT-технологій для створення систем домашньої автоматизації. Розроблені рішення забезпечують високий рівень комфорту, безпеки та енергоефективності, а запропоновані підходи можуть бути використані для подальшого розвитку та масштабування систем «Розумного будинку».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ashton, K. That 'Internet of Things' Thing // RFID Journal. – 2009. – [Електронний ресурс]. – Режим доступу: https://www.rfidjournal.com/that-internet-of-things-thing.
2. Atzori, L., Iera, A., Morabito, G. The Internet of Things: A survey // Computer Networks. – 2010. – №54. – С. 2787–2805. – DOI: 10.1016/j.comnet.2010.05.010.
3. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications // IEEE Communications Surveys & Tutorials. – 2015. – Vol. 17, №4. – С. 2347–2376. – DOI: 10.1109/COMST.2015.2444095.
4. Хоменко О.І. Технології бездротового зв’язку в системах Інтернету речей // Телекомунікації та радіотехніка. – 2021. – №2. – С. 45–52.
5. Шамрай П.В. Хмарні технології в IoT: переваги та недоліки // Сучасні інформаційні технології. – 2022. – №1. – С. 22–28.
6. Minerva, R., Biru, A., Rotondi, D. Towards a Definition of the Internet of Things (IoT) // IEEE Internet Initiative. – 2015. – [Електронний ресурс]. – Режим доступу: https://internetinitiative.ieee.org/.
7. Мельник І.В. Архітектура інтелектуальних систем автоматизації будівель // Вісник НТУУ «КПІ». Серія: Інформатика та обчислювальна техніка. – 2020. – №70. – С. 50–56.
8. Raza, S., Wallgren, L., Voigt, T. Security in the Internet of Things: Standards and Solutions // Sensors. – 2018. – Vol. 18, №5. – С. 1356. – DOI: 10.3390/s18051356.
9. Ploennigs, J., et al. Learning Energy Efficiency in Buildings: A Survey // IEEE Transactions on Industrial Informatics. – 2017. – Vol. 13, №3. – С. 1271–1283. – DOI: 10.1109/TII.2016.2618898.
10. Zorzi, M., Gluhak, A., Lange, S., Bassi, A. From today's Intranet of Things to a future Internet of Things: A wireless- and mobility-related view // IEEE Wireless Communications. – 2010. – Vol. 17, №6. – С. 44–51. – DOI: 10.1109/MWC.2010.5675779.

11. Statista Research Department. Smart Home – Statistics & Facts. – 2023. – [Електронний ресурс]. – Режим доступу: https://www.statista.com/topics/2430/smart-homes/.
12. Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., Zhao, W. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications // IEEE Internet of Things Journal. – 2017. – Vol. 4, №5. – С. 1125–1142. – DOI: 10.1109/JIOT.2017.2683200.
13. Palattella, M. R., et al. Internet of Things in the 5G Era: Enablers, Architecture, and Business Models // IEEE Journal on Selected Areas in Communications. – 2016. – Vol. 34, №3. – С. 510–527. – DOI: 10.1109/JSAC.2016.2525418.
14. Chen, M., Ma, Y., Li, Y., Wu, D., Zhang, Y., Youn, C. Big Data Analytics for Smart Homes: Review, Challenges and Research Opportunities // IEEE Transactions on Smart Grid. – 2017. – Vol. 9, №4. – С. 3137–3145. – DOI: 10.1109/TSG.2016.2641116.
15. Microsoft Azure IoT Central Documentation. – [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/en-us/azure/iot-central/
16. MQTT Version 5.0. OASIS Standard. – 2019. – [Електронний ресурс]. – Режим доступу: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html
17. Azure IoT SDK for Python Documentation. – [Електронний ресурс]. – Режим доступу: https://github.com/Azure/azure-iot-sdk-python
18. Kamath, M. Building a Smart Home using Azure IoT Hub and Device SDKs // International Journal of Advanced Computer Science and Applications. – 2020. – Vol. 11, №2. – DOI: 10.14569/IJACSA.2020.0110213.
19. Buyya, R., Srirama, S. N. Fog and Edge Computing: Principles and Paradigms. – Wiley, 2019. – 500 с.
20. Thangavel, D., Ma, X., Valera, A., Tan, H. P., Tan, C. K. Performance Evaluation of MQTT and CoAP via a Common Middleware // IEEE 9th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). – 2014. – DOI: 10.1109/ISSNIP.2014.6827678.

21. Amazon Web Services IoT Core Documentation. – [Електронний ресурс]. – Режим доступу: https://docs.aws.amazon.com/iot/
22. Bonomi, F., Milito, R., Zhu, J., Addepalli, S. Fog Computing and Its Role in the Internet of Things // Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing. – 2012. – С. 13–16. – DOI: 10.1145/2342509.2342513.
23. Google Cloud IoT Core Documentation. – [Електронний ресурс]. – Режим доступу: https://cloud.google.com/iot-core
24. Verk, J., Saigol, K., IoT Protocols Comparison for Smart Home Automation // Journal of Communications Software and Systems. – 2021. – Vol. 17, №1. – С. 34–42. – DOI: 10.24138/jcomss-2021-0010

ДЕМОНСТРЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Розробка протоколів взаємодії IoT для побудови «Розумного будинку»

Студент: Ваганов Олександр Юрійович ШІд - 42
Керівник: Фесенко М.А.

Рік: 2025

Що таке Smart Home?

Smart Home або «Розумний будинок» — це система, що дозволяє автоматизувати керування побутовими приладами: освітленням, опаленням, безпекою, кліматом та іншими пристроями.

Завдяки застосуванню Інтернету речей (IoT) пристрої об'єднуються в єдину мережу, здатну самостійно приймати рішення або бути керованою дистанційно.

Приклад: автоматичне вимкнення світла при відсутності людей у кімнаті або сповіщення на смартфон у разі руху в будинку.

Що таке IoT і як він працює у Smart Home

IoT (Інтернет речей) — це концепція підключення фізичних пристроїв до Інтернету для обміну даними.

У розумному будинку IoT забезпечує зв'язок між сенсорами, контролерами, хмарною платформою та користувачем.

Система збирає дані з сенсорів (температура, рух, стан дверей), передає їх у хмару, де вони обробляються, і надсилає команди виконавчим пристроям (вмикання світла, сирена, замки).

Актуальність теми

Технології розумного будинку дедалі частіше впроваджуються у житлові та комерційні приміщення.

В умовах України це особливо актуально через потребу зменшити витрати на енергоресурси, підвищити безпеку та комфорт.

Протоколи обміну в IoT

1. MQTT — легкий, енергоефективний, ідеальний для передачі телеметрії у реальному часі.

2. AMQP — складніший, але гарантує надійність доставки, корисний у критичних сценаріях.

3. HTTPS — стандартний веб-протокол, зручний для REST API, але не дуже ефективний для IoT.

Обрані протоколи забезпечують різні аспекти взаємодії — від простоти до безпеки.

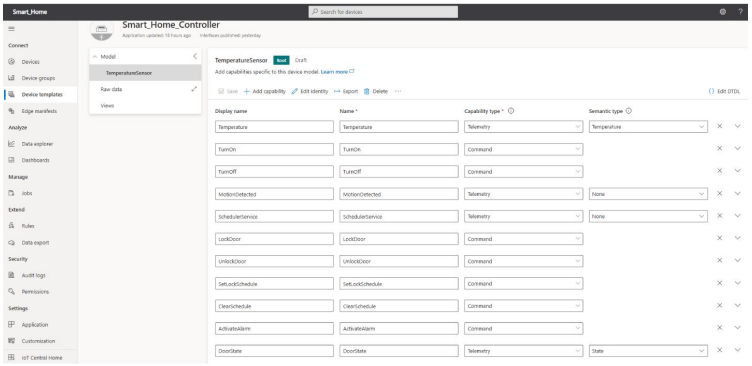
Архітектура системи

Система побудована з кількох рівнів:

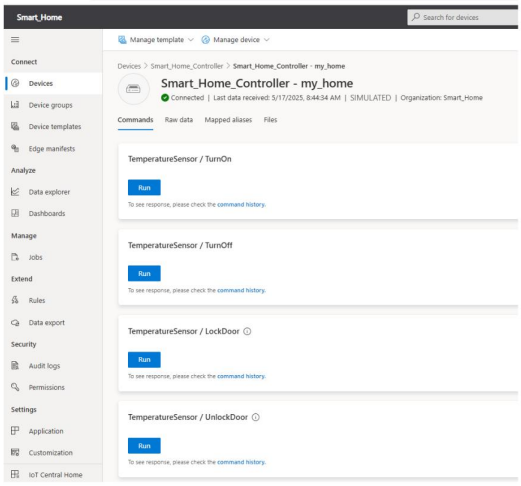
- Рівень пристроїв: сенсори, замки, освітлення.
- Мережева передача даних: MQTT або AMQP.
- Хмарна обробка: Microsoft Azure IoT Central.
- Інтерфейс користувача: панель керування через браузер.

Користувач може спостерігати за телеметрією, отримувати сповіщення та керувати пристроями.

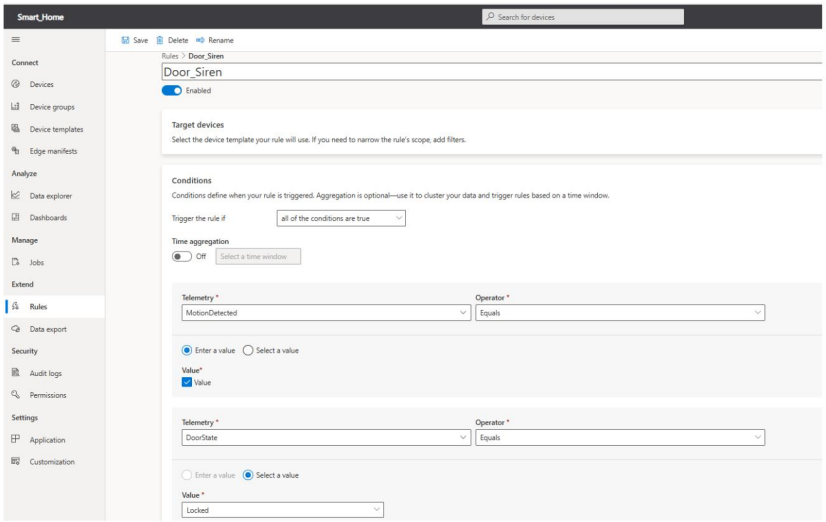
Загальна структура шаблону
Smart_Home_Controller у
середовищі Azure IoT Central

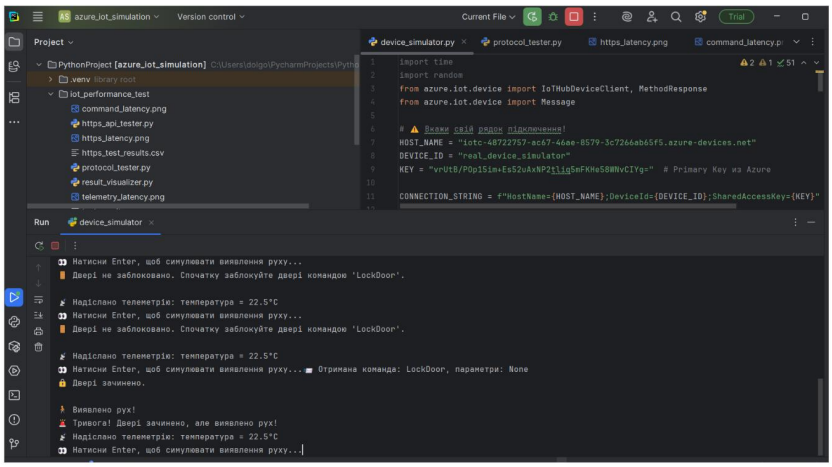


Інтерфейс керування пристроями у середовищі Azure IoT Central



Приклад реалізації правила автоматизації в Azure IoT Central

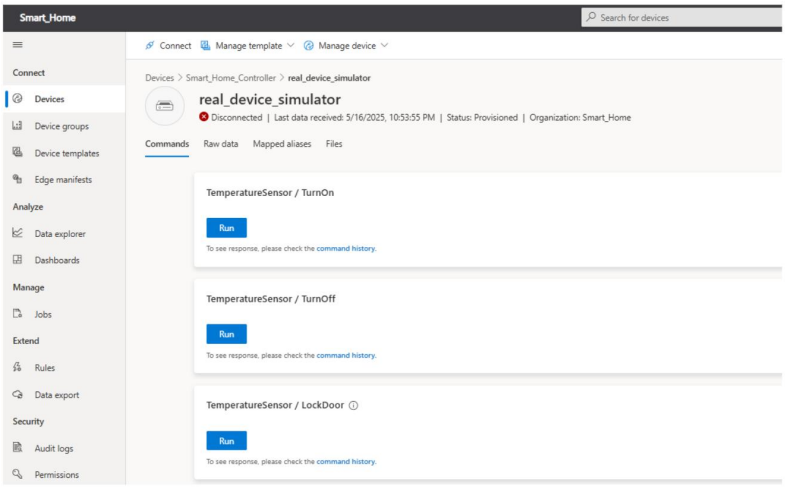


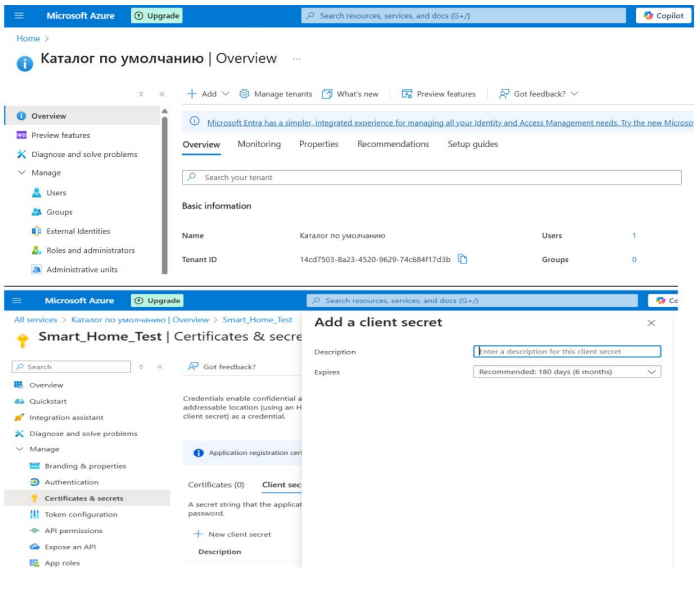


Для перевірки працездатності системи було проведено серію тестів, у ході яких виконувались такі дії:

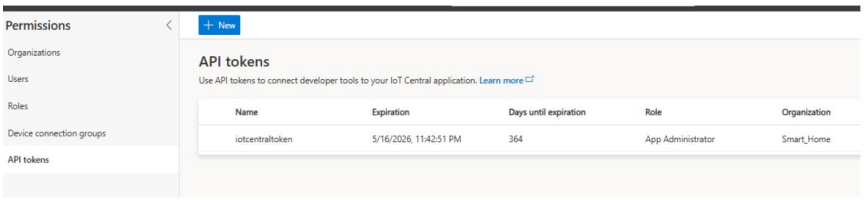
- **передача телеметрії з емулятора пристроїв;**
- **відправлення команд керування пристроями;**
- **активація правил автоматизації та перевірка реакції системи.**

Налаштування віртуального пристрою real_device_simulator у хмарній платформі Azure IoT Central.



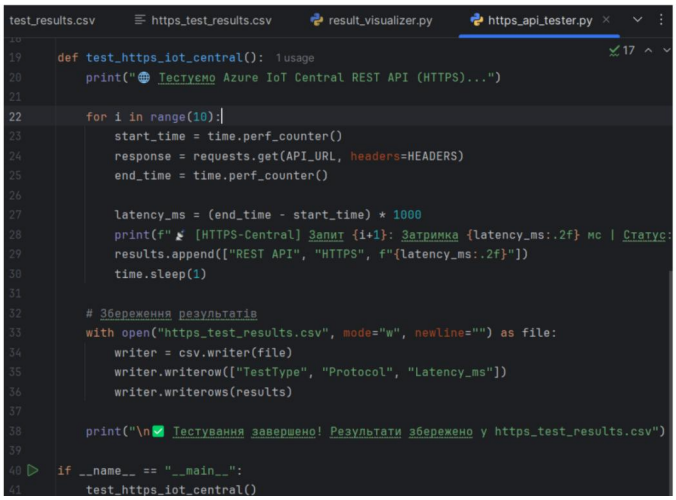


Генерація API Token для роботи з REST API.

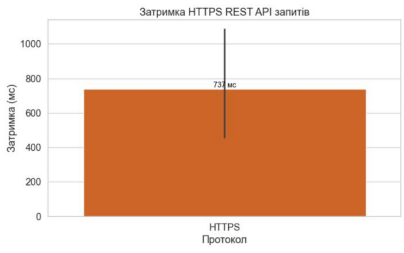


Цей токен використовувався в HTTPS-тестах для надсилання запитів без необхідності інтерактивної авторизації. Це значно спростило автоматизацію тестових сценаріїв.

Фрагмент коду тестування продуктивності HTTPS REST API

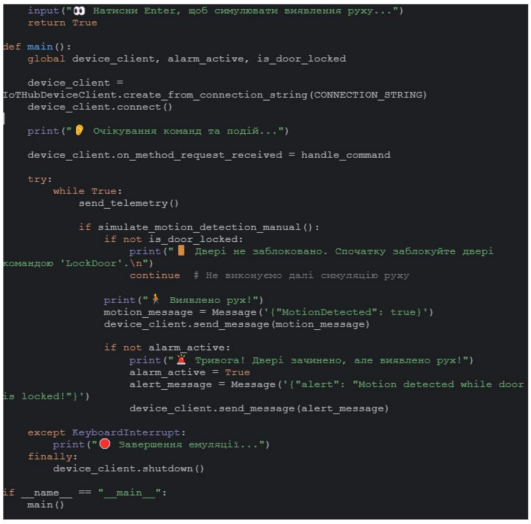


Результати тестування



Для доступу до API використовувався заздалегідь згенерований API Token, який забезпечував безпечну авторизацію без необхідності проходження інтерактивного входу.

Симулятор виявлення руху з передачею телеметрії до Azure IoT Hub



```
    input("👀 Натисни Enter, щоб симулювати виявлення руху...")
    return True

def main():
    global device_client, alarm_active, is_door_locked

    device_client =
IoTHubDeviceClient.create_from_connection_string(CONNECTION_STRING)
    device_client.connect()

    print("👂 Очікування команд та подій...")

    device_client.on_method_request_received = handle_command

    try:
        while True:
            send_telemetry()

            if simulate_motion_detection_manual():
                if not is_door_locked:
                    print("🚪 Двері не заблоковано. Спочатку заблокуйте двері
командою 'LockDoor'.\n")
                    continue  # Не виконуємо далі симуляцію руху

                print("🏃 Виявлено рух!")
                motion_message = Message('{"MotionDetected": true}')
                device_client.send_message(motion_message)

                if not alarm_active:
                    print("🚨 Тривога! Двері зачинено, але виявлено рух!")
                    alarm_active = True
                    alert_message = Message('{"alert": "Motion detected while door
is locked!"}')
                    device_client.send_message(alert_message)

    except KeyboardInterrupt:
        print("🛑 Завершення емуляції...")
    finally:
        device_client.shutdown()

if __name__ == "__main__":
    main()
```

Обробка команд керування пристроєм у системі IoT

```
CONNECTION_STRING =
f"HostName={HOST_NAME};DeviceId={DEVICE_ID};SharedAccessKey={KEY}"

is_door_locked = False
alarm_active = False

def handle_command(request):
    global is_door_locked, alarm_active

    command_name = request.name
    payload = request.payload
    print(f"📩 Отримана команда: {command_name}, параметри: {payload}")

    if command_name == "TurnOn":
        print("✅ Датчик температури увімкнено.")
        status = 200
        response_payload = {"result": "Temperature Sensor Turned On"}
    elif command_name == "TurnOff":
        print("❌ Датчик температури вимкнено.")
        status = 200
        response_payload = {"result": "Temperature Sensor Turned Off"}
    elif command_name == "LockDoor":
        is_door_locked = True
        print("🔒 Двері зачинено.")
        status = 200
        response_payload = {"result": "Door Locked"}
    elif command_name == "UnlockDoor":
        is_door_locked = False
        alarm_active = False
        print("🔓 Двері відчинено.")
        status = 200
        response_payload = {"result": "Door Unlocked"}
    else:
        print("⚠️ Невідома команда.")
        status = 400
        response_payload = {"result": "Unknown command"}

    method_response = MethodResponse.create_from_method_request(
        request, status, response_payload
    )
    device_client.send_method_response(method_response)

def send_telemetry():
    temperature = 22.5  # Фіксоване значення температури для демонстрації
    telemetry = Message(f'{{"Temperature": {temperature}}}')
    device_client.send_message(telemetry)
    print(f"📡 Надіслано телеметрію: температура = {temperature}°C")

def simulate_motion_detection_manual():
```

Висновок

1. Розумний будинок можна створити на основі доступних рішень — без дорогого обладнання.

2. Протокол MQTT — оптимальний для більшості побутових задач завдяки простоті, ефективності та низькій затримці.

3. Microsoft Azure IoT Central надає простий та масштабований спосіб організації обробки даних.

4. Результати підтвердили, що розроблена модель підходить для практичного впровадження у побутових умовах.