

О. А. ГОЛЕНКО

студент кафедри інформатики і комп'ютерних наук
Херсонський національний технічний університет
ORCID: 0009-0003-4818-6406

М. В. КАРАМУШКА

доцент кафедри інформатики і комп'ютерних наук
Херсонський національний технічний університет
ORCID: 0000-0001-5982-4598

ДОСЛІДЖЕННЯ МОДЕРНІЗАЦІЇ ТА ВПРОВАДЖЕННЯ ІНТЕРНЕТ ТЕХНОЛОГІЙ І СИСТЕМ МОНІТОРИНГУ ТА УПРАВЛІННЯ У ЖИТЛОВИХ ПРИМІЩЕННЯХ

Стаття присвячена модернізації та впровадженню систем Інтернету речей для моніторингу і керування у житлових приміщеннях, де одночасно працюють спадкові дротові та радіорішення на кшталт RS-485 з протоколом Modbus, інфрачервоні та радіоканали на частотах чотириста тридцять три і вісімсот шістьдесят вісім мегагерц, а також сучасні пристрої з передаванням даних через Wi-Fi, Bluetooth Low Energy, Zigbee і Thread. Мета роботи полягає у формуванні методичних засад інтеграції, що забезпечують сумісність, локальний контроль, приватність і роботу без мережі Інтернет, без прив'язки до конкретного виробника. Запропоновано еталонну архітектуру локально орієнтованого хаба на базі Raspberry Pi. Вона містить шар адаптерів до різних фізичних та прикладних протоколів, шар нормалізації що перетворює нативні повідомлення у канонічні події та команди з уніфікованими одиницями виміру і часовими мітками у форматі UTC ISO 8601, подієву шину MQTT з гарантіями доставки, реєстр пристроїв і їхніх можливостей, рушій правил, а також зовнішнє програмне API через REST та WebSocket. Сформовано канонічну модель даних і доменне API з поділом полів на основні та розширені. Наведено профілі для датчиків температури, руху, протікання, лічильників енергії та керованих пристроїв на кшталт вимикачів і розеток. Показано мапінг із Modbus, з Bluetooth Low Energy, з інфрачервоних і радіопристроїв, а також з IP-пристроїв. Описано онбординг і захист, зокрема використання токенів або сертифікатів, шифрування транспортного рівня, рольову авторизацію та ведення журналів. Подано механізми роботи офлайн, серед яких зберігання та пересилання після відновлення, дедуплікація за ідентифікатором пристрою і часовою міткою. Реалізовано прототип із брокером, бек-ендом на Python з фреймворком FastAPI, сховищем SQLite та файловим буфером, модулем правил або Node-RED і локальною веб-панеллю з можливістю інтеграції з Home Assistant. Запропоновано методику оцінювання, що враховує затримку виконання команд, надійність доставки, час відновлення після збоїв і енергетичний ефект у сценаріях клімату та освітлення. Практична цінність полягає у поетапній міграції від розрізнених рішень до уніфікованої платформи з підвищеною надійністю і зниженими витратами інтеграції. Окреслено подальший розвиток, зокрема підтримку шин BACnet і KNX через адаптери, автоматичне витягування можливостей пристроїв з описів Web of Things і Matter та інструменти валідації якості даних для аналітики і штучного інтелекту.

Ключові слова: Інтернет речей, житлові приміщення, Raspberry Pi, протоколи, MQTT, безпека, офлайн-стійкість, інтеграція систем, уніфікація даних.

О. А. GOLENKO

Student at the Department of Computer Science
Kherson National Technical University
ORCID: 0009-0003-4818-6406

M. V. KARAMUSHKA

Associate Professor at the Department of Computer Science
Kherson National Technical University
ORCID: 0000-0001-5982-4598

RESEARCH ON MODERNIZATION AND IMPLEMENTATION OF INTERNET TECHNOLOGIES AND MONITORING AND MANAGEMENT SYSTEMS IN RESIDENTIAL PREMISES

This article explores the modernization and deployment of Internet of Things (IoT) technologies for monitoring and control in residential environments where legacy infrastructures – such as RS-485 with Modbus, infrared channels, and sub-GHz radios at 433/868 MHz – coexist with modern devices relying on Wi-Fi, Bluetooth Low Energy, Zigbee, and Thread. The research aims to establish methodological foundations for integration that ensure interoperability, local control, privacy, and resilience to Internet disruptions while avoiding dependence on a single vendor. A reference architecture for a locally oriented hub based on Raspberry Pi is proposed. The architecture comprises an adapter layer for heterogeneous physical and application protocols, a normalization layer that transforms native messages into canonical events and commands with unified measurement units and UTC ISO 8601 timestamps, an MQTT event bus with delivery guarantees, a device and capability registry, a rules engine, and external interfaces via REST and WebSocket. A canonical data model and domain API are introduced, separating core and extended fields. Profiles are provided for temperature, motion, leak detection, and energy-metering sensors, as well as controllable devices such as switches and outlets. The study demonstrates protocol mapping from Modbus, BLE, infrared, and sub-GHz radios, along with IP-based devices. Security and onboarding procedures include provisioning tokens or certificates, encrypted transport, role-based access control, and event logging. Offline operation is supported through store-and-forward, deduplication, and recovery mechanisms. A functional prototype was implemented using a Mosquitto broker, a FastAPI back end with SQLite storage and file-based buffering, Node-RED for automation, and a local web dashboard with optional integration into Home Assistant. An evaluation methodology is proposed to measure command latency, reliability of delivery, recovery time after faults, and energy efficiency in climate-control and lighting scenarios. The practical contribution lies in enabling a stepwise migration from fragmented legacy solutions to a unified, vendor-independent platform that improves reliability and reduces integration costs. Future work will extend support for BACnet and KNX via adapters, leverage Web of Things and Matter descriptions for automatic capability extraction, and develop data-quality validation tools for analytics and AI applications.

Key words: Internet of Things, residential environments, Raspberry Pi, protocols, MQTT, security, offline resilience, system integration, data unification.

Постановка проблеми

Сучасні інформаційні технології стрімко змінюють підходи до облаштування житлових приміщень – від офісних будівель, шкіл, гуртожитків і багатоквартирних будинків до приватних апартаментів. Технології Інтернету речей (IoT) відкривають нові можливості для моніторингу та керування домашньою інфраструктурою [1]. Особливої уваги потребує розроблення інтегрованих IoT-систем, спроможних взаємодіяти з побутовими пристроями, датчиками, «розумними» замками, камерами, охоронними системами та лічильниками ресурсів для підвищення ефективності управління і зниження енерговитрат. Попри велику кількість індивідуальних продуктів на ринку «розумних» рішень, відсутність універсальної системи, здатної централізовано керувати всіма аспектами житла, зумовлює потребу в аналізі доступних підходів і подальших дослідженнях.

У житловому секторі триває інтенсивна модернізація: поряд із новими IoT-пристроями (сенсори мікроклімату, «розумні» вимикачі, лічильники з передаванням даних, голосові асистенти) продовжують функціонувати спадкові системи підтримки будинку – дротові датчики охорони, домофони, котли та контролери опалення, аналогові/імпульсні лічильники, ІЧ-керовані прилади, радіопристрої на 433-868 МГц, інтерфейси RS-485 або Modbus, а також KNX та BACnet. Така співіснування «старого» і «нового» супроводжується фрагментацією протоколів LAN, Wi-Fi, BLE, Zigbee, IR, власні радіоінтерфейси і невідповідністю прикладних моделей – від відкритих MQTT, HTTP та CoAP до закритих API та пропрієтарних форматів [2].

Наслідками є несумісність і «силоси» даних, різні одиниці виміру та часові мітки, дублювання шлюзів, зростання витрат інтеграції та експлуатації, ускладнене масштабування і залежність від постачальників/хмарних сервісів. Додатково житлові системи висувають підвищені вимоги до приватності, офлайн-стійкості (робота під час збоїв інтернету/живлення) і кібербезпеки локальної мережі. Відсутність єдиного підходу до уніфікації протоколів і семантики даних гальмує модернізацію: нові пристрої складно інтегрувати з наявною інфраструктурою, а «старі» – адаптувати під сучасні сценарії автоматизації та аналітики.

Отже, постає науково-прикладна задача: обґрунтувати методичні засади модернізації та впровадження IoT-систем у житлових приміщеннях, які забезпечують інтероперабельність «старих» і «нових» пристроїв. Необхідно визначити принципи уніфікації обміну даними (на рівні формату й семантики), правила мапінгу спадкових протоколів до єдиної моделі, а також вимоги до офлайн-стійкості, безпеки та керованого розширення. Це створить основу для поетапної інтеграції і подальшої аналітики/ІІІ без прив'язки до конкретного вендора чи транспорту.

Аналіз останніх досліджень і публікацій

В сьогодення підходи до сумісності в «розумному» житлі рухаються у двох взаємо доповнювальних напрямках перший з яких уніфікація семантики пристроїв та другий уніфікація прикладного рівня поверх IP. Стандарти W3C Web of Things (WoT) 1.1 та CSA Matter 1.4 є показовими прикладами, перший пропонує незалежну від протоколів,

структуровану для автоматизованої обробки модель опису «речей», що знімає частину фрагментації на рівні описів та зв'язувань через HTTP/MQTT/CoAP також другий уніфікує класи пристроїв, атрибути й команди поверх IP та поступово розширює домени такі як енергоменеджмент, мережна інфраструктура, спрощений онбординг NFC і multi-device QR [3]. На транспортному рівні MQTT 5.0 лишається де-факто стандартом публікації/підписки, але не стандартизує семантику повідомлення, тож потрібна нормалізація на рівні хаба [4].

Різні дослідження наголошують на важливості інтеграції IoT для підвищення енергоефективності будинків і безпеки мешканців. Огляди й прикладні роботи показують ефективність в поєднанні IoT + AI а саме алгоритми прогнозування та оптимізації дозволяють автоматизувати рутинні завдання, підлаштовувати освітлення та клімат під графік присутності, зменшувати споживання енергії без втрати комфорту [5]. Продукти масового ринку на кшталт Google Home та Amazon Alexa дають зручний віддалений контроль, але їхній сценарний рушій традиційно був менш гнучким для індивідуальних потреб, але останнім часом Google розширює редактор автоматизацій, однак частина функцій ще не підтримується [6]. Дослідники також підкреслюють значення відкритої архітектури, стандартів як чинника зменшення витрат інтеграції та розширення функціональності – саме цю роль відіграє WoT як уніфікуюча ланка описів і зв'язувань між різними екосистемами. Водночас хмаро-центричні підходи несуть ризики приватності та витоків даних, що підтверджується систематичними оглядами безпеки та конфіденційності й резонансними кейсами залежності від сторонніх сервісів. Тому локальний контроль і можливість офлайн-роботи визнаються критично важливими для житлових систем [7].

На практиці попит на локально-орієнтовані рішення відбивають зрілі open-source платформи на кшталт Home Assistant або openHAB із фокусом на локальний контроль та приватність і розгортання на Raspberry Pi. Паралельно існують комерційні локальні хаби в основі Hubitat із акцентом на офлайновість і швидкі автоматизації. Ці системи доповнюють Matter та WoT, виступаючи «мостами» для спадкових технологій та уніфікуючи дані через хаб-рівень.

Для наочного порівняння наведено таблицю технологій, розглянутих вище, для впровадження та модернізації IoT у розумному приміщенні.

Таблиця 1

Порівняльна таблиця технологій

Технології	Тип контролю	Off-line	Рівень кастомних автоматизацій	Приватність/ локальність	Розширюваність/ інтеграції	Коментар
Home Assistant	Локальний (з опціями хмари)	Так	Високий (GUI + YAML; скрипти)	“local control & privacy first”	3300+ інтеграцій	Підтримка локального голосу (Assist) та DIY-стеків
openHAB	Локальний (open-source)	Так	Високий (Rules DSL/JS; UI)	Локальний сервер/ захист	Широкі “add-ons”	Рекомендований для RPi; технологічно агностичний
Hubitat	Локальний (комерційний)	Так	Середньо-високий (Rule Machine)	Акцент на “Local, Private”	Zigbee/Z-Wave/ Matter тощо	Може працювати без інтернету; оновлення – за мережею
Google Home	Переважно хмара + локальні мости (Matter/Thread)	Частково	Середній	Данні в хмарі, локальний control покриває не все	Широкі брендові інтеграції	Нещодавне розширення автоматизацій; деякі дії ще не підтримано
Amazon Alexa	Переважно хмара + локальні мости (Matter/Zigbee)	Частково	Середній	Залежність від хмарних інтеграцій	Широкі брендові інтеграції	Приклади збоїв/деактивації хмарних зв'язок (IFTTT, вендорські скіли)

Формулювання мети дослідження

Метою роботи є розроблення методичних засад інтеграції IoT-систем у житлових приміщеннях, що забезпечують взаємодію спадкових та сучасних пристроїв і підсистем. Дослідження зосереджене на уніфікації обміну даними та проектуванні бек-енд хаба з адаптерами до різних протоколів і єдиним доменним API. Результатом є рекомендації та еталонна архітектура локально-орієнтованого шлюзу для поетапної інтеграції й підвищення енергоефективності та безпеки.

Передбачено проаналізувати сучасні стандарти та платформи «розумного» житла й окреслити їхні обмеження для житлових сценаріїв. Далі – класифікувати пристрої та підсистеми та визначити вимоги до їхньої взаємодії з хабом. Сформулювати канонічну модель даних і команд із поділом на обов'язкові та розширені поля, уніфікувавши одиниці виміру та часові мітки. Спроектувати бек-енд хаба з шаром адаптерів до різних протоколів та єдиним доменним API. Описати процедури онбордингу та безпеки, а також політики офлайн-стійкості. Розробити методику оцінювання – затримку виконання команд, надійність доставки подій, стійкість до відмов інтернету/живлення та вплив на енергоефективність. Надати практичні рекомендації з поетапної модернізації – план міграції від спадкових систем до уніфікованої архітектури й типові сценарії впровадження.

Викладення основного матеріалу дослідження

У межах дослідження використано багатопланову архітектуру локально орієнтованого хаба, який виконує роль центрального вузла інтеграції «старих» і сучасних підсистем житла показано на Рисунку 1.

Ядро рішення складається з:

- Шар адаптерів – шлюзи до різних фізичних/канальних та прикладних протоколів: RS-485 чи Modbus, ІЧ, Wi-Fi, BLE за потреби Zigbee або Matter).
- Шар нормалізації – перетворення нативних повідомлень на канонічні події/команди зі спільними одиницями виміру, часовими мітками (UTC) та індикаторами якості.
- Подієва шина – внутрішня подієва шина (MQTT) з QoS, ретенцією та дедуплікацією.
- Реєстр пристроїв та можливостей – реєстр пристроїв, їхніх можливостей, версій профілів і політик доступу.
- Правила/Автоматизація – двигун сценаріїв та правил (локальне виконання без хмари).
- Зовнішнє API – єдине прикладне API (REST/WS) для клієнтів, мобільних застосунків і «peer-hub» інтеграцій.

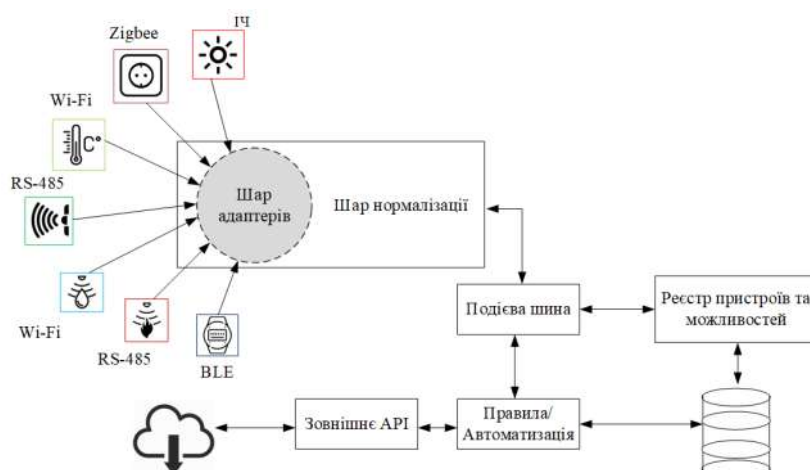


Рис. 1. Референсна архітектура локально орієнтованого хаба (шари та потоки даних)

Канонічна модель даних і доменне API передбачають профілі з поділом на обов'язкове «ядро» та розширення.

Мінімальний набір полів для телеметрії: *device_id*, *capability*, *ts* (UTC ISO 8601), *value*, *unit* (SI), *quality*, *schema_ver*. Для команд: *device_id*, *command*, *params*, *ts*, *schema_ver*. Така організація забезпечує єдину семантику й формат даних для аналітики та ШП, зворотну сумісність через версіонування схем, а також прозорий обмін між кількома хабами без додаткових перетворень, нижче в таблиці 2 представлений приклад канонічний профілів для типових пристроїв.

Таблиця 2

Приклад канонічних профілів для типових пристроїв (температура, вологість, рух, вимикач)

№	Клас пристрою	Обов'язкові поля	Додаткові поля
1	sensor.temperature	device_id, capability, ts (UTC ISO 8601), value, unit=°C, quality, schema_ver	battery_level, signal_strength, location
2	sensor.motion	device_id, capability, ts, value (bool), quality, schema_ver	battery_level, signal_strength, occupancy_timeout, sensitivity
3	switch.light	device_id, command(s)/state, ts, schema_ver	brightness (0–100), color_temp (K), transition_ms, scene_id
4	sensor.open	device_id, capability, ts, value (open/closed), quality, schema_ver	battery_level, signal_strength, tamper
5	meter.energy	device_id, capability, ts, value (Wh або kWh), unit, quality, schema_ver	interval_sec, voltage (V), current (A), power (W), phase
6	switch.outlet	device_id, state (on/off), ts, schema_ver	power (W), energy (Wh/kWh), child_lock, overload_protect

Шар адаптерів і мапінг протоколів реалізує уніфікацію вхідних даних і команд: значення з RS-485/Modbus передає до прикладу такі параметри.

Reg 0x0001 # температура
Reg 0x0002 # вологість

Зчитуються періодично й нормалізуються – перетворення у °C або %, додання ts (UTC) та прапорця quality.

У BLE локальний міст підписується на характеристики наприклад.

0x2A6E # Temperature

Потім конвертує бінарні дані в єдині одиниці, додає метадані та публікує у шину.

Для ІЧ або 868 МГц коди пакетів зіставляються з каталогом команд фбо станів.

```
mode=COOL, temp=16
mode=HOT, temp=22
```

Відправляються як стандартний command.

Вже пристрої Wi-Fi та IP публікують телеметрію безпосередньо у канонічному форматі або через легкий device agent на мікроконтролері

У разі використання готових інтегрованих модулів Zigbee або Matter програмні мости відображають кластерні та атрибутивні моделі на відповідні capabilities без змін ядра системи.

Онбординг і безпека реалізуються так, під час початкового під'єднання пристрій отримує provisioning-токен або сертифікат, згенерований хабом до прикладу для Wi-Fi задаються SSID та пароль і адреси API, для BLE виконується парування з локальним мостом. Далі увесь транспорт шифрується TLS/mTLS для IP, захищені сесії для BLE, доступ контролюється рольовою авторизацією, а події або команди за потреби підписуються для гарантії цілісності. Ведуться обов'язкові журнали доступів і змін конфігурації. На Рисунку 2 можемо спостерігати дану схему.



Рис. 2. Онбординг та безпека

Офлайн-стійкість і надійність доставки забезпечуються схемою store-and-forward з повторною відправкою після відновлення зв'язку, використанням рівнів MQTT QoS, збереженням для останнього стану та механізмами heartbeat для виявлення офлайн-пристроїв. Дедуплікація подій виконується за ключем (device_id, ts), а політики ретенції визначаються класом даних чи критичні або некритичні.

Реєстр пристроїв і хабів реалізуються через Device & Capability Registry, який зберігає ідентифікатори, типи, виробника, версії прошивки/профілю, перелік підтримуваних capabilities та дозволів доступу. Для міжвузлової взаємодії передбачено обмін: інші хаби під'єднуються через «Зовнішнє API» або обмінюються подіями безпосередньо по MQTT або WebSocket за узгодженими схемами.

Реалізація прототипу. Еталонний стенд реалізовано на Raspberry Pi із таким стеком:

- MQTT-брокер (Mosquitto) – «Подієва шина»;
- Python-сервіси для під'єднання датчиків – «Шар адаптерів»;
- Python сервер (Python/Flask або FastAPI) – «Шар нормалізації»;
- Бек-енд (Python/Flask або FastAPI) – «Зовнішнє API»;
- SQLite (метадані) + файловий буфер телеметрії – «Сховище»;
- Node-RED (опційно) – «Автоматизації»;
- Клієнтський інтерфейс – «Зовнішнє API».

Приклади застосування та сценарії:

1. Оптимізація клімату за присутністю. Дані датчиків руху, відкриття та графік користувача формують цільові температури; команди надсилаються до котла чи кондиціонера через «Подієву шину».

2. Виявлення протікань і аварій. Сенсор води по 433 МГц через хаб сповіщення та перекриття електроклапана.

3. Освітлення з плавним зонуванням. Кнопки чи датчики освітленості BLE корегують сцени освітлення. Збереження останнього стану для швидкого відновлення після зникнення живлення.

Ми перевіряємо чотири прості речі: наскільки швидко система реагує, чи не губить повідомлення, як швидко «оживає» після збоїв і чи дає економію енергії. Для швидкості ми надсилаємо команду (наприклад, увімкнути

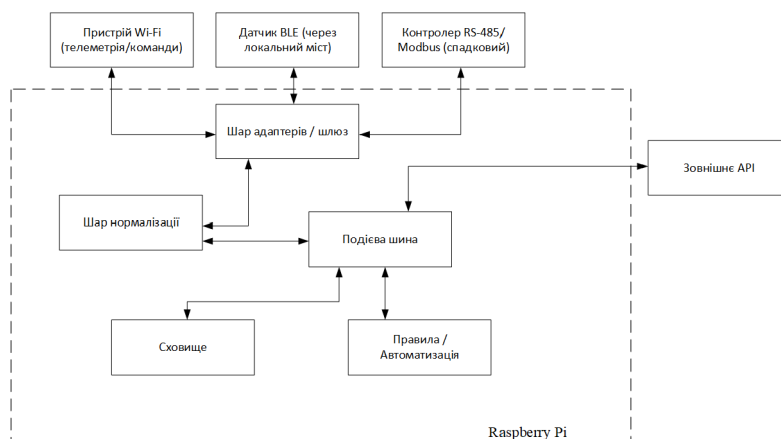


Рис. 3. Фізична топологія стенда та програмні компоненти на Raspberry Pi

світло) й фіксуємо час до фактичного увімкнення – чим менше, тим краще. Рахуємо середній результат і «повільні» випадки. Для надійності навмисно створюємо проблеми – відключаємо інтернет або Wi-Fi – і дивимось, скільки подій система втратила або продублювала в ідеалі – майже нуль. Для відновлення просто перезавантажуємо хаб і міряємо, через скільки секунд правила знову починають працювати без нашого втручання. Для економії енергії порівнюємо показники лічильників «до» і «після» увімкнення автоматизацій, якщо кіловат-годин стало менше – значить, є користь. Усі заміри робимо кілька разів на однакових сценаріях, фіксуємо час у журналах і подаємо результат на графіках та в таблиці з простим порівнянням «було/стало».

Обмеження та подальший розвиток. Поточна реалізація орієнтована на побутові сценарії та обмежений набір протоколів, повноцінну підтримку промислових шин планується додавати через окремі адаптери. Подальший розвиток передбачає автоматичне витягування можливостей з описів WoT та Matter, розширення канонічних профілів і впровадження інструментів перевірки якості даних для подальшої аналітики та застосувань III.

Висновки

У роботі обґрунтовано потребу в уніфікації протоколів і семантики даних для житлових IoT-систем, де одночасно використовуються «старі» дротові та радіорішення та нові IP-пристрої. Запропоновано еталонну архітектуру локально орієнтованого хаба на Raspberry Pi з шарами адаптерів і нормалізації, подієвою шиною MQTT, реєстром пристроїв і їхніх можливостей, рушієм правил та зовнішнім API – що забезпечує інтеграцію без прив'язки до конкретного вендора. Сформовано канонічну модель даних і доменне API з поділом полів на основні та розширені; продемонстровано мапінг Modbus, BLE, IЧ, 433–868 МГц, Wi-Fi та IP і містків Zigbee та Matter у єдині «можливості пристроїв», що спрощує аналітику та застосування III. Описано процедури онбордингу й захисту на основі provisioning-токенів та TLS шифруванні, а також механізми офлайн-стійкості, необхідні для надійної роботи без хмари. Реалізовано еталонний стенд і визначено методику оцінювання – затримка команд, надійність доставки, час відновлення після збоїв та енергетичний ефект у сценаріях клімату й освітлення. Запропоноване рішення придатне для поетапної модернізації житла: воно надає єдиний спосіб підключення різномірних пристроїв, підвищує приватність за рахунок локального виконання, знижує витрати інтеграції та спрощує масштабування. Обмеженням нинішньої версії є фокус на побутових сценаріях і вибраних протоколах; подальша робота передбачає підтримку повних профілів BACnet та KNX через адаптери, автоматичне витягування «можливостей пристроїв» з описів WoT, Matter та впровадження інструментів валідації якості даних для аналітики й III.

Список використаної літератури

1. Sensors (MDPI). Special Issue: Smart Homes. Description of the direction “IoT for smart homes”. <https://www.mdpi.com/journal/sensors>.
2. Orfanos V. A. et al. A Comprehensive Review of IoT Networking Technologies. *Micromachines*, 2023. <https://www.mdpi.com/2224-2708/12/2/30>
3. W3C. Web of Things (WoT) Thing Description 1.1 Recommendation. 2023. <https://www.w3.org/TR/wot-thing-description11/>
4. Connectivity Standards Alliance. Matter 1.4.1: Enhanced Setup Flow, Multi-Device QR, NFC Onboarding. 2025. <https://csa-iot.org/newsroom/a-smarter-start-matter-1-4-1-makes-setup-easier/>
5. OASIS. MQTT Version 5.0 – Specification. 2019+. <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>
6. Ali D.M.T.E. et al. AI-Driven Innovations in Building Energy Management Systems: A Review of Potential Applications and Energy Savings. *Energies*, 2024. <https://www.mdpi.com/1996-1073/17/17/4277>

7. Google Home: new automation editor (update overview). Android Central, 2025. <https://www.androidcentral.com/accessories/smart-home/google-home-new-automation-editor-lets-you-do-so-much-more>.

References

1. Sensors (MDPI). Special Issue: Smart Homes. Description of the direction “IoT for smart homes”. <https://www.mdpi.com/journal/sensors>
2. Orfanos V. A. et al. A Comprehensive Review of IoT Networking Technologies. Micromachines, 2023. <https://www.mdpi.com/2224-2708/12/2/30>
3. W3C. Web of Things (WoT) Thing Description 1.1 Recommendation. 2023. <https://www.w3.org/TR/wot-thing-description11/>
4. Connectivity Standards Alliance. Matter 1.4.1: Enhanced Setup Flow, Multi-Device QR, NFC Onboarding. 2025. <https://csa-iot.org/newsroom/a-smarter-start-matter-1-4-1-makes-setup-easier/>
5. OASIS. MQTT Version 5.0 – Specification. 2019+. <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>
6. Ali D.M.T.E. et al. AI-Driven Innovations in Building Energy Management Systems: A Review of Potential Applications and Energy Savings. Energies, 2024. <https://www.mdpi.com/1996-1073/17/17/4277>
7. Google Home: new automation editor (update overview). Android Central, 2025. <https://www.androidcentral.com/accessories/smart-home/google-home-new-automation-editor-lets-you-do-so-much-more>

Дата першого надходження рукопису до видання: 26.09.2025

Дата прийнятого до друку рукопису після рецензування: 23.10.2025

Дата публікації: 28.11.2025