

**Пермяков В. А.**, магістрант кафедри електронних обчислювальних машин  
Харківського національного університету радіоелектроніки  
ORCID: 0009-0008-8252-6036

**Піскар'єв О. М.**, кандидат технічних наук, доцент,  
доцент кафедри електронних обчислювальних машин  
Харківського національного університету радіоелектроніки  
ORCID: 0000-0002-6980-984X  
Scopus Author ID: 57200141599

## МЕТОД ЗАБЕЗПЕЧЕННЯ ВІДМОВОСТІЙКОГО ДИСТАНЦІЙНОГО КЕРУВАННЯ ІНТЕЛЕКТУАЛЬНИМИ СИСТЕМАМИ ЖИТЛОВОГО СЕРЕДОВИЩА НА ОСНОВІ МОБІЛЬНИХ ПЛАТФОРМ

Основною метою дослідження є обґрунтування та розробка методу, що забезпечує повну функціональну автономність локальних систем автоматизації житлового середовища. Робота спрямована на створення архітектури, яка зберігає можливість керування освітленням, опаленням та безпекою навіть за умов повної відсутності зовнішніх каналів зв'язку та припинення централізованого електропостачання, що є критичним недоліком сучасних хмарних IoT-рішень. Методологічну основу дослідження становить концептуальний перехід від традиційної хмарної моделі до парадигми туманних обчислень. Для практичної реалізації обрано локальний сервер на базі мікрокомп'ютера Raspberry 3B+, який функціонує як брокер повідомлень за протоколом MQTT. Технічне рішення включає застосування схеми розділеного живлення контурів від джерела безперебійного живлення для виключення енергетичних втрат на подвійне перетворення. Програмна частина реалізована мовою C++ для забезпечення низькорівневої взаємодії з GPIO-інтерфейсом, тоді як клієнтський мобільний додаток побудовано на платформі Flutter із використанням принципу «Local-First» та бази даних sqflite для надійного локального кешування інформації. Наукова новизна отриманих результатів полягає у розробці гібридного методу організації інформаційного обміну та енергозабезпечення в інтелектуальних системах. Запропонований підхід поєднує децентралізовану обробку даних на локальному Edge-шлюзі з безпосереднім живленням виконавчих механізмів постійним струмом. Це дозволяє усунути критичну залежність від зовнішніх хмарних провайдерів, ліквідуючи «єдину точку відмови», притаманну більшості існуючих комерційних систем автоматизації. Результати практичної апробації розробленого комплексу засвідчили його високу експлуатаційну надійність та стабільність. Встановлено, що система зберігає повну керуваність у критичних режимах одночасного знеструмлення об'єкта та втрати інтернет-з'єднання. Локалізація комунікаційного ядра дозволила скоротити час відгуку системи на команди до діапазону 5–10 мс, а також забезпечити високий рівень конфіденційності, оскільки персональні дані користувача фізично залишаються в межах локальної мережі житлового приміщення.

Ключові слова: IoT, Fog Computing, Raspberry Pi, Flutter, розумний будинок, відмовостійкість, MQTT.

**Permiakov V. A., Piskarev O. M. Method for ensuring failure-resistant remote control of intelligent systems in the living environment based on mobile platforms**

The main objective of the research is to justify and develop a method that ensures the complete functional autonomy of local home automation systems. The work is aimed at creating an architecture that retains the ability to control lighting, heating and security even in the absence of external communication channels and in the event of a centralised power supply failure, which is a critical shortcoming of modern cloud-based IoT solutions. The methodological basis of the research is the conceptual transition from the traditional cloud model to the fog computing paradigm. For practical implementation, a local server based on a Raspberry Pi 3B+ microcomputer was selected, which functions as a message broker using the MQTT protocol. The technical solution includes the use of a split-power supply circuit from an uninterruptible power supply (UPS) to eliminate energy losses due to double conversion. The software component is implemented in C++ to ensure low-level interaction with the GPIO interface, whilst the client mobile application is built on the Flutter platform using the 'Local-First' principle and the SQLite database for reliable local caching of information. The scientific novelty of the results lies in the development of a hybrid method for organising information exchange and power supply in intelligent systems. The proposed approach combines decentralised data processing on a local Edge gateway with direct DC power supply to actuators. This eliminates critical dependence on external cloud providers, removing the 'single point of failure' inherent in most existing commercial automation systems. The results of practical testing of the developed system demonstrated its high operational reliability and stability. It has been established that the system retains full controllability in critical scenarios involving simultaneous power failure at the facility and loss of internet



© В. А. Пермяков, О. М. Піскар'єв, 2026

Стаття поширюється на умовах ліцензії відкритого доступу CC BY 4.0

---

*connection. Localisation of the communication core has reduced the system's response time to commands to a range of 5–10 ms, whilst also ensuring a high level of confidentiality, as the user's personal data remains physically within the local network of the residential premises.*

*Key words: IoT, Fog Computing, Raspberry Pi, Flutter, smart home, fault tolerance, MQTT.*

**Постановка проблеми.** В умовах сучасних проблем, зокрема непередбачуваних відключень електроенергії та нестабільності інтернет-з'єднання, класичні хмарні архітектури IoT (Internet of Things) демонструють критичну вразливість. Системи, що покладаються на пропріетарні сервери, наприклад такі як система «Xiaomi MiJa», втрачають працездатність при зникненні доступу до інтернет-зв'язку, перетворюючи «розумний будинок» на некерований об'єкт.

Розробка програмного забезпечення для систем розумного будинку стикається з такими проблемами, як інтеграція, комунікація, безпека та стабільність. Серед ключових питань – сумісність пристроїв, енергоефективність, безпека даних і доступність для користувача. Рішення включають стандартизацію, шлюзи, API, енергоефективний код, протоколи з низьким енергоспоживанням, шифрування даних і моніторинг загроз. Розробка програмного забезпечення для систем розумного будинку має вирішальне значення, при цьому основна увага приділяється інноваціям, практичній цінності та доступності для користувача.

Актуальним завданням є розробка методу, який забезпечує повну функціональність локальної автоматики (керування опаленням, освітленням, безпекою, комфортом і т.д.) незалежно від наявності зовнішніх каналів зв'язку та централізованого електропостачання.

**Аналіз останніх досліджень та публікацій.** Питання розробки та оптимізації систем розумного будинку активно висвітлюються в сучасній науковій літературі. Зокрема, парадигму туманних обчислень (Fog Computing) ґрунтовно досліджено у роботі «A Review of Fog Computing Concept, Architecture, Application, Parameters and Challenges» [1, с. 565], де автори доводять, що перенесення обчислювальних потужностей ближче до кінцевих вузлів мережі дозволяє суттєво розвантажити хмарні сервери та зменшити комунікаційні затримки. Важливість використання легковагових протоколів для таких розподілених мереж підкреслюється в огляді «MQTT Protocol for the IoT – Review Paper» [2, с. 2], де MQTT розглядається як оптимальний стандарт для пристроїв з обмеженими апаратними ресурсами. Ці теоретичні висновки підтверджуються практичними дослідженнями «Design and Implementation of a Framework for Smart Home Automation Based on Cellular IoT, MQTT, and Serverless Functions» [3, с. 4], автори яких розробили ефективний фреймворк автоматизації на базі MQTT, що демонструє високу надійність передачі телеметрії. Крім того, аналіз продуктивності «Performance evaluation of CoAP and MQTT with security support for IoT environments» [4, с. 17] доводить, що даний протокол зберігає високу швидкість обміну даними навіть при додатковому навантаженні, пов'язаному з інтеграцією механізмів криптографічного захисту.

Значна увага в сучасних публікаціях приділяється питанням конфіденційності інформації. Так, у дослідженні «Internet of Things (IoT) of Smart Homes: Privacy and Security» [5, с. 2] детально проаналізовано ключові вразливості IoT-мереж і зазначено, що постійна передача персональних даних на зовнішні сервери створює критичні ризики кібератак. Для вирішення цих проблем науковці пропонують впроваджувати полегшені схеми локальної автентифікації «Light Weight Authentication Scheme for Smart Home IoT Devices» [6, с. 2] та створювати ізольовані системи моніторингу на зразок платформи «RADAR-IoT: An Open-Source, Interoperable, and Extensible IoT Gateway Framework...» [7 с. 5], які здатні повноцінно функціонувати без обов'язкового звернення до хмарних інтерфейсів.

Щодо апаратно-програмної реалізації, сучасні науковці відзначають високу ефективність використання локальних шлюзів (Edge Gateways) та периферійних обчислень для обробки складних сценаріїв безпосередньо на об'єкті, як показано у дослідженні «Edge AI for Real-Time Anomaly Detection in Smart Homes» [8, с. 2]. Для взаємодії з такими апаратними комплексами все частіше застосовуються кросплатформні фреймворки. Наприклад, у роботах «Development of Water Level Monitoring Applications in Smart Home Systems Using Flutter» [9, с. 225] та «Development of A Prototype of An IoT Based Smart Home with Security System Flutter Mobile» [10, с. 5] обґрунтовано переваги використання технології Flutter, яка завдяки реактивній архітектурі дозволяє створювати високопродуктивні мобільні інтерфейси для IoT-систем, ефективно замінюючи застарілі підходи до розробки.

**Мета статті** – обґрунтування та розробка методу, що забезпечує повну функціональну автономність локальних систем автоматизації житлового середовища. Робота спрямована на створення архітектури, яка зберігає можливість керування освітленням, опаленням та безпекою навіть за умов повної відсутності зовнішніх каналів зв'язку та припинення централізованого електропостачання, що є критичним недоліком сучасних хмарних IoT-рішень.

**Виклад основного матеріалу.** Методологічну основу дослідження становить концептуальний перехід від традиційної хмарної моделі до парадигми туманних обчислень. Для практичної реалізації обрано локальний сервер на базі мікрокомп'ютера Raspberry Pi 3B+, який функціонує як брокер повідомлень за протоколом MQTT. Технічне рішення включає застосування схеми розділеного живлення контурів від джерела безперебійного живлення для виключення енергетичних втрат на подвійне перетворення. Програмна

---

частина реалізована мовою C++ для забезпечення низькорівневої взаємодії з GPIO-інтерфейсом, тоді як клієнтський мобільний додаток побудовано на платформі Flutter із використанням принципу «Local-First» та бази даних sqflite для надійного локального кешування інформації.

Наукова новизна отриманих результатів полягає у розробці гібридного методу організації інформаційного обміну та енергозабезпечення в інтелектуальних системах. Запропонований підхід поєднує децентралізовану обробку даних на локальному Edge-шлюзі з безпосереднім живленням виконавчих механізмів постійним струмом. Це дозволяє усунути критичну залежність від зовнішніх хмарних провайдерів, ліквідує «єдину точку відмови», притаманну більшості існуючих комерційних систем автоматизації.

Для практичного досягнення заявленої відмовостійкості на рівні центрального хабу було відмовлено від використання стандартних дистрибутивів операційних систем. Замість цього, за допомогою інструментарію крос-компіляції Buildroot, було згенеровано мінімалістичний та оптимізований образ ОС на базі ядра Linux. Таке рішення дозволило перевести файлову систему в режим «тільки для читання» (Read-Only), що повністю виключає ризик пошкодження системних файлів під час раптових відключень електроенергії. Відсутність зайвих фонових процесів скоротила час «холодного старту» сервера до кількох секунд, забезпечуючи миттєве відновлення роботи комунікаційного ядра.

У якості кінцевих виконавчих пристроїв та модулів збору телеметрії застосовано мікроконтролери сімейства ESP8266. На відміну від типових комерційних рішень, ці модулі інтегровані в систему без використання пропрієтарних API. Кожен пристрій функціонує як незалежний клієнт, що здійснює обмін даними виключно через локальний MQTT-брокер. Вбудована логіка мікроконтролерів передбачає автономне збереження останнього заданого стану у разі тимчасової втрати зв'язку з хабом, що унеможливує хаотичну поведінку автоматика під час мережевих збоїв.

Загалом сучасні системи розумного будинку є прикладом активного впровадження технологій Internet of Things (IoT) у побутове середовище. Такі системи інтегрують апаратні та програмні компоненти з метою забезпечення автоматизації, підвищення рівня комфорту, безпеки та ефективного управління ресурсами. Їх популярність зростає завдяки доступності, розширеним функціональним можливостям і здатності до інтеграції з іншими технологічними платформами.

Основними причинами активного розвитку та впровадження систем розумного будинку можна назвати зростання попиту на автоматизацію через прагнення користувачів до підвищення комфорту та спрощення рутинних задач, надаючи технологіям можливість виконувати їх автономно. Ефективне управління енергоспоживанням також є перевагою розумного будинку, бо за допомогою використання різних датчиків та алгоритмів керування дозволяє контролювати витрати ресурсів. Ще однією суттєвою перевагою є можливість додаткового забезпечення безпеки приміщення. Система передбачає підключення відеокамер, датчиків руху та сигналізації, що забезпечує підвищення рівня безпеки об'єкта.

Гнучкість і персоналізація є визначними характеристиками розумного будинку. Система може адаптуватися під індивідуальні потреби кожного користувача, дозволяючи встановлювати певні сценарії роботи залежно від часу доби, погодних умов чи навіть присутності мешканців у будинку. Наприклад, освітлення може автоматично регулювати яскравість на основі кількості природного світла, а кліматична система – підтримувати комфортний мікроклімат у кожній окремій кімнаті керуючи системами кондиціонування.

Системи розумного будинку будуються на поєднанні трьох основних складових: апаратної частини, програмного забезпечення та хмарних сервісів. Апаратна частина включає мікроконтролери, датчики та виконавчі пристрої, які забезпечують збір і передачу даних. Програмне забезпечення дозволяє обробляти цю інформацію та надавати інструменти для керування пристроями, наприклад, через мобільні додатки чи веб-інтерфейси. Хмарні сервіси забезпечують дистанційний доступ до системи та зберігання великих обсягів даних, що дає можливість моніторингу та контролю системи з будь-якої точки світу.

В якості контролера, що є центральною частиною системи, можна використовувати різні варіанти. Як варіант можливо використання звичайного персонального комп'ютера чи ноутбука, що є доволі простим та потужним рішенням. Проте в такого варіанту є суттєвий недолік – енергоспоживання, через надмірну потужність, яку середньостатистична система розумного будинку не використовує, збільшуються витрати на енергію та кінцевий продукт стає менш автономним у випадку відсутності постійного енергоживлення.

Іншим варіантом, що залишає універсальність класичних комп'ютерів – використання одноплатних комп'ютерів, таких як «Raspberry Pi», «Orange Pi» та інших. Таке рішення дозволяє виконувати зручне налаштування системи за допомогою ОС «GNU Linux», та можливості майже нескінченного масштабування, шляхом створення додаткових мережних розширень (створення локального Proxi сервера, VPN тунелю і т.д.). В порівнянні з класичними ПК таке рішення споживає суттєво менше енергії, проте і потужність значно менша.

У випадку, коли просто необхідно реалізувати просту систему керування та моніторингу стану в приміщенні можна використовувати мікроконтролер (наприклад ESP32, STM32 і т.д.), що дозволить зекономити на купівлі та електроенергії, проте майже неможливе швидке налаштування та масштабування, через те, що зміни налаштувань в мікроконтролері виконуються шляхом його повного перепрограмування.

Далі необхідно обрати схему підключення та взаємодію пристроїв системи (датчики, відеокамери і т.д.). Для реалізації поставленої задачі можна використовувати різні підходи, від створення локальної

---

мережі до CAN-bus та його аналогів. При використанні мережних технологій можливо використання як дротових так і бездротових технологій.

З бездротових технологій для реалізації поставленої мети в перспективі розширення можливостей дуже зручним є варіант використання технології «ZigBee», що є основною метою даної технології. Перевагою даного рішення є мале енергоспоживання та мобільність пристроїв системи в межах приміщення, проте недоліком є необхідність реалізації окремого джерела автономного живлення для роботи кожного елементу системи в період відсутності електропостачання.

Цікавим рішенням є використання CAN-bus в системі розумного будинку. З'єднання пристроїв таким чином дає можливість реалізувати дуже надійну, відносно недорогу та енергоефективну систему, проте швидкість передачі даних обмежена до 1 Мбіт/с, що не дозволить якісно реалізувати потокову передачу відеосигналу з камери спостереження, проте достатня для моніторингу показників датчиків та їх керування. Для роботи CAN-bus необхідно використовувати спеціальні модулі, які мають бути синхронізовані між собою.

Безпека даних у розробленій системі базується на принципі локалізації інформаційних потоків. Оскільки критичні дані керування та телеметрії циркулюють виключно в межах ізольованої локальної мережі без обов'язкової передачі на зовнішні хмарні сервери, ризик витоку персональної інформації мінімізується [5, с. 2; 7, с. 5]. Для захисту бездротового сегмента мережі використовується протокол шифрування WPA3, а обмін даними між мобільним пристроєм та сервером захищено TLS-шифруванням [6]. Додатковий рівень безпеки забезпечується фізичною ізоляцією сервера керування від публічної мережі Інтернет у разі виявлення кіберзагроз.

Для дослідження та реалізації відмовостійкості системи було використано архітектуру туманних обчислень [1, с. 566]. Основою апаратної частини системи обрано одноплатний комп'ютер «Raspberry Pi» 3В+. На відміну від мікроконтролерів (наприклад ESP32), дане рішення дозволяє розгорнути повноцінний локальний сервер із власною базою даних та брокером повідомлень, що забезпечує можливість обробки складних сценаріїв автоматизації безпосередньо на об'єкті мережі. Такий вибір гарантує збереження керуваності системою навіть при повній відсутності інтернет-з'єднання. Обчислювальна потужність «Raspberry Pi» дозволяє одночасно підтримувати роботу серверної частини, інтерфейсу бази даних та служб моніторингу енергоспоживання [8, с. 4].

Програмна реалізація низькорівневого керування виконавчими механізмами виконана мовою програмування C++ в середовищі ОС «GNU Linux». Вибір цієї мови обумовлений необхідністю прямої взаємодії з GPIO-інтерфейсом одноплатного комп'ютера з мінімальними затримками [8, с. 4].

Для розробки макету застосунку був використаний графічний редактор «Figma», завдяки якій вдалося спроектувати зручний і зрозумілий інтерфейс, який враховує сучасні принципи дизайну.

Для розробки мобільного застосунку було обрано мову програмування Dart та її фреймворк Flutter. Ця мова була обрана через її високий рівень продуктивності, легкості у вивченні та тісну інтеграцію з фреймворком Flutter. Мова програмування Dart забезпечує швидку компіляцію коду у машинний, що дозволяє створювати додатки з високою швидкістю роботи. Крім того, Dart підтримує імперативний і реактивний підходи, що забезпечує ефективну реалізацію динамічних користувацьких інтерфейсів. Використання вбудованих функцій, таких як асинхронне програмування з `async` та `await`, спрощує роботу з запитами до серверів і забезпечує стабільну взаємодію з хмарними сервісами та мікроконтролерами [9].

Завдяки фреймворку Flutter можна створювати застосунки, які однаково добре працюють на Android та iOS, скорочуючи час розробки та підтримки. Цей фреймворк надає багато можливостей для реалізації інтерактивного інтерфейсу користувача, що є важливим для систем розумного будинку (рис. 1) [10, с. 40].

Програмна реалізація серверної частини виконана у вигляді системної служби (system daemon), що гарантує автоматичну ініціалізацію при завантаженні операційної системи та забезпечує безперервний моніторинг стану виконавчих релейних модулів. Застосування об'єктно-орієнтованого підходу дозволило розробити уніфікований програмний інтерфейс, який абстрагує логіку керування від фізичної реалізації комутації, спрощуючи масштабування системи.

Архітектуру клієнтського додатку на платформі Flutter було оптимізовано для роботи в режимі «Local-First». На відміну від типових рішень, що покладаються на хмарні SDK, у проекті імплементовано стек бібліотек для автономної взаємодії. Зокрема, пакет `mqtt_client` забезпечує високошвидкісний двосторонній обмін даними за протоколом MQTT [3, 4], `connectivity_plus` відповідає за адаптивне перемикання мережних інтерфейсів, а `sqlite` забезпечує локальне кешування історії подій безпосередньо на пристрої користувача. Критичні сповіщення (наприклад, про перехід на резервне живлення) реалізовано через локальний механізм `flutter_local_notifications`.

Фундаментальною відмінністю розробленої системи є відмова від архітектури BaaS (Backend-as-a-Service) на користь туманних обчислень (Fog Computing) [1]. Роль комунікаційного ядра виконує локальний MQTT-брокер «Mosquito», розгорнутий на «Raspberry Pi». Така децентралізація забезпечує критичні переваги: повну незалежність від інтернет-з'єднання, гарантовану приватність даних (зберігання в межах локального контуру) та суттєве зменшення латентності системи з 200–500 мс, характерних для хмарних рішень, до 5–10 мс у локальній мережі.

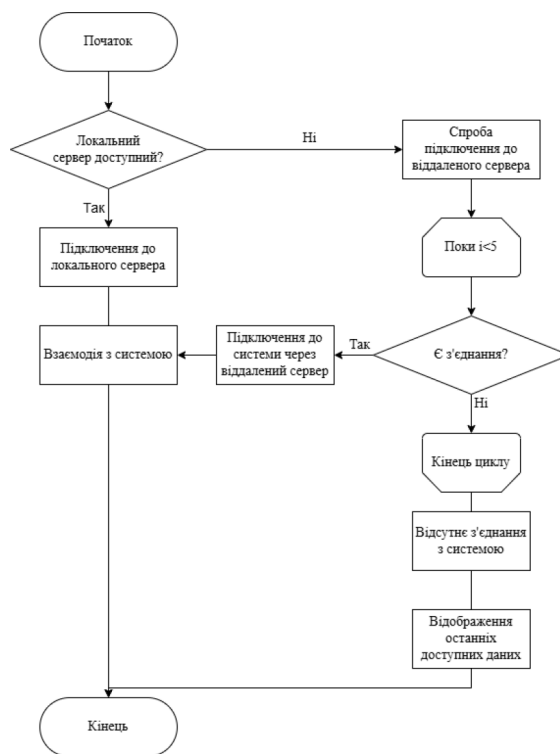


Рис. 1. Блок-схема алгоритму роботи додатку

Для емпіричного підтвердження ефективності розробленого методу було проведено порівняльне тестування мережевих затримок (Round-Trip Time, RTT) при керуванні виконавчими пристроями в межах локальної бездротової мережі. Експериментальний стенд складався з локального Edge-шлюзу (Raspberry Pi 3B+ під управлінням ОС Buildroot), маршрутизатора та клієнтського мобільного терміналу. Тестування проводилося у двох режимах: за наявності стабільного доступу до глобальної мережі (штатний режим) та при фізичному відключенні WAN-каналу провайдера (аварійний режим).

Для порівняння розглядалася типова хмарна архітектура (SaaS), де команди від смартфона обов'язково маршрутизуються через зовнішній API-сервер (наприклад, базові Wi-Fi пристрої, що не підтримують протоколи локального керування LAN), та розроблений гібридний метод локального MQTT-брокера. Результати вимірювань наведено у Таблиці 1.

Таблиця 1

**Результати порівняльного тестування затримок при локальному керуванні системою**

Архітектура системи	Затримки в штатному режимі	Затримки при збої інтернету	Працездатність системи
Хмарна	30 мс	Недоступно	Недоступна (втрата зв'язку з API)
Запропонована	5 мс	5 мс	Недоступність віддаленого керування

Аналіз отриманих даних, наведених у Таблиці 1, доводить перевагу запропонованої архітектури в умовах нестабільного зв'язку. На відміну від суто хмарних рішень, які втрачають керування навіть за умови знаходження користувача в одній локальній мережі з пристроями, локалізація комунікаційного ядра дозволила забезпечити стовідсоткове збереження функціональності автоматики. Крім того, виключення ланки пересилання пакетів через інтернет-маршрутизатори дозволило суттєво мінімізувати час реакції системи на команди користувача.

Запропонована демонстраційна система є основою для подальшого розвитку та вдосконалення. У перспективі можливе додавання штучного інтелекту для аналізу поведінки користувача і автоматичної оптимізації сценаріїв керування будинком. Використання Big Data для обробки великого обсягу даних дозволить створювати прогнози щодо енергоспоживання та подальшого вдосконалення ефективності системи. Запропонована архітектурна схема взаємодії системи наведена на рис. 2.

Крім того, впровадження голосових асистентів на базі існуючих рішень, таких як «Google Assistant» або «Amazon Alexa», забезпечить ще більшу інтеграцію системи у повсякденне життя користувачів. Перспективним напрямом розвитку є інтеграція з пристроями екосистеми розумного будинку за допомогою

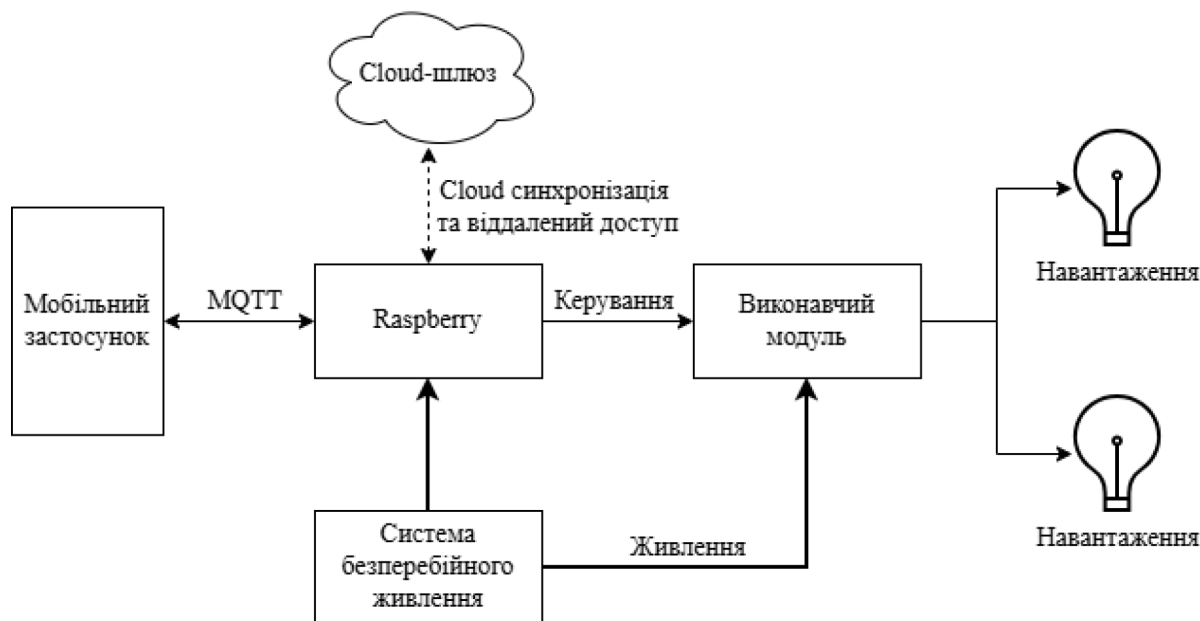


Рис. 2. Схема взаємодії системи

протоколів MQTT, Zigbee та Z-Wave, що забезпечить розширення функціональних можливостей і підвищення масштабованості системи.

Таким чином, розроблена система демонструє можливості сучасних технологій для побудови ефективного, безпечного та комфортного розумного будинку. Результати дослідження можуть бути використані для подальшого вдосконалення системи, її масштабування та інтеграції в реальні житлові чи комерційні проекти, що відкриває нові горизонти для технологічного розвитку у сфері автоматизації будинків.

Для забезпечення стабільності роботи системи, у розробці використовуються засоби аналізу продуктивності додатків, доступні у фреймворка Flutter. Це дозволяє виявляти вузькі місця в мобільному додатку й оптимізувати його для зменшення затримок у передачі даних між пристроями.

У разі масштабування системи, бездротові сенсори на базі ESP32 інтегруються в загальну мережу через MQTT. Таке рішення забезпечує швидку та енергоефективну передачу даних між компонентами системи, особливо у сценаріях, коли потрібно відправляти часті оновлення даних від датчиків [2].

**Висновки та перспективи.** Розробка системи на базі одноплатного комп'ютера «Raspberry Pi» та протоколу MQTT з дистанційним керуванням через мобільний застосунок дозволила створити повністю автономне рішення з можливістю масштабування. Інтеграція локального сервера з системою безперебійного живлення забезпечує працездатність розумного будинку навіть за умов повної відсутності інтернет-з'єднання та електропостачання, що є критичною перевагою над хмарними аналогами.

Подальший розвиток системи може включати інтеграцію технологій штучного інтелекту для оптимізації сценаріїв керування та використання Big Data для прогнозування енергоспоживання. Розроблена система демонструє потенціал для застосування в реальних проектах, що сприятиме подальшому вдосконаленню технологій автоматизації та підвищенню ефективності розумних будинків.

#### Список використаних джерел:

1. Ghani N., Bakar Sajak A., Qureshi R., Azril Zuhairi M., Ahmad Baidowi Z. A Review of Fog Computing Concept, Architecture, Application, Parameters and Challenges. JOIV : International Journal on Informatics Visualization. 2024. Vol. 8, No. 2. P. 564–575. DOI: <https://doi.org/10.62527/joiv.8.2.2187>

2. Usmani M. F. MQTT Protocol for the IoT – Review Paper. High Integrity Systems. 2023. P. 1–10. URL: <https://www.researchgate.net/publication/373640610>

3. Esposito M., Belli A., Palma L., Pierleoni P. Design and Implementation of a Framework for Smart Home Automation Based on Cellular IoT, MQTT, and Serverless Functions. Sensors. 2023. Vol. 23, No. 9. P. 4459. DOI: <https://doi.org/10.3390/s23094459>

4. Seoane V., Garcia-Rubio C., Almenares F., Campo C. Performance evaluation of CoAP and MQTT with security support for IoT environments. Computer Networks. 2021. Vol. 197. P. 108338. DOI: <https://doi.org/10.1016/j.comnet.2021.108338>

5. Magara T., Zhou Y. Internet of Things (IoT) of Smart Homes: Privacy and Security. Journal of Electrical and Computer Engineering. 2024. Vol. 2024. DOI: <https://doi.org/10.1155/2024/7716956>

- 
6. Kumar V., Malik N., Singla J., Jhanjhi N. Z., Amsaad F., Razaque A. Light Weight Authentication Scheme for Smart Home IoT Devices. *Cryptography*. 2022. Vol. 6, No. 3. P. 37. DOI: <https://doi.org/10.3390/cryptography6030037>
  7. Ranjan Y., Chang J., Sankesara H., Conde P., Rashid Z., Dobson R. J. B., Folarin A. RADAR-IoT: An Open-Source, Interoperable, and Extensible IoT Gateway Framework for Health Research. *Sensors*. 2024. Vol. 24, No. 14. P. 4614. DOI: <https://doi.org/10.3390/s24144614>
  8. Reis M. J. C. S., Serôdio C. Edge AI for Real-Time Anomaly Detection in Smart Homes. *Future Internet*. 2025. Vol. 17, No. 4. P. 179. DOI: <https://doi.org/10.3390/fi17040179>
  9. Hendriawan M., Haryono H., Budiman T. Development of Water Level Monitoring Applications in Smart Home Systems Using Flutter. *Journal of Information System, Informatics and Computing*. 2023. Vol. 7, No. 2. P. 213–240. URL: <https://journal.stmikjayakarta.ac.id/index.php/jisicom/article/view/1197>
  10. Hamzah N. A. Development of A Prototype of An IoT Based Smart Home with Security System Flutter Mobile. *ResearchGate*. 2023. P. 1–8. URL: <https://www.researchgate.net/publication/347789214>

#### References:

1. Ghani, N., Bakar Sajak, A., Qureshi, R., Azril Zuhairi, M., & Ahmad Baidowi, Z. (2024). A Review of Fog Computing Concept, Architecture, Application, Parameters and Challenges. *JOIV : International Journal on Informatics Visualization*, 8(2), 564–575. <https://doi.org/10.62527/joiv.8.2.2187>
2. Usmani, M. F. (2023). MQTT Protocol for the IoT – Review Paper. *High Integrity Systems*, 1–10.
3. Esposito, M., Belli, A., Palma, L., & Pierleoni, P. (2023). Design and Implementation of a Framework for Smart Home Automation Based on Cellular IoT, MQTT, and Serverless Functions. *Sensors*, 23(9), 4459. <https://doi.org/10.3390/s23094459>
4. Seoane, V., Garcia-Rubio, C., Almenares, F., & Campo, C. (2021). Performance evaluation of CoAP and MQTT with security support for IoT environments. *Computer Networks*, 197, 108338. <https://doi.org/10.1016/j.comnet.2021.108338>
5. Magara, T., & Zhou, Y. (2024). Internet of Things (IoT) of Smart Homes: Privacy and Security. *Journal of Electrical and Computer Engineering*, 2024. <https://doi.org/10.1155/2024/7716956>
6. Kumar, V., Malik, N., Singla, J., Jhanjhi, N. Z., Amsaad, F., & Razaque, A. (2022). Light Weight Authentication Scheme for Smart Home IoT Devices. *Cryptography*, 6(3), 37. <https://doi.org/10.3390/cryptography6030037>
7. Ranjan, Y., Chang, J., Sankesara, H., Conde, P., Rashid, Z., Dobson, R. J. B., & Folarin, A. (2024). RADAR-IoT: An Open-Source, Interoperable, and Extensible IoT Gateway Framework for Health Research. *Sensors*, 24(14), 4614. <https://doi.org/10.3390/s24144614>
8. Reis, M. J. C. S., & Serôdio, C. (2025). Edge AI for Real-Time Anomaly Detection in Smart Homes. *Future Internet*, 17(4), 179. <https://doi.org/10.3390/fi17040179>
9. Hendriawan, M., Haryono, H., & Budiman, T. (2023). Development of Water Level Monitoring Applications in Smart Home Systems Using Flutter. *Journal of Information System, Informatics and Computing*, 7(2), 213–240.
10. Hamzah, N. A. (2023). Development of A Prototype of An IoT Based Smart Home with Security System Flutter Mobile. *ResearchGate*, 1–8.

Дата першого надходження статті до видання: 26.03.2026

Дата прийняття статті до друку після рецензування: 20.04.2026

Дата публікації (оприлюднення) статті: 30.05.2026