

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра Кафедра автоматизації проектування обчислювальної техніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)
Система управління житловим приміщенням на базі ESP32 і Zigbee з
адаптивним регулюванням споживання
(тема)

Виконав:
здобувач 4 курсу, групи КІУКІ-21-7
Гладких О. Р.
(прізвище, ініціали)

Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва спеціальності)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Комп'ютерна інженерія
(повна назва освітньої програми)

Керівник ас.Хаханов І.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Чумаченко С. В.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної інженерії та управління _____
Кафедра _____ Автоматизації проектування обчислювальної техніки _____
Рівень вищої освіти _____ перший (бакалаврський) _____
Спеціальність _____ 123 – Комп'ютерна інженерія _____
(код і повна назва)
Тип програми _____ освітньо-професійна _____
Освітня програма _____ Комп'ютерна інженерія _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри Чумаченко С. В.

_____ (підпис)

« 20 » _____ травня _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувачу _____ Гладких Олексію Руслановичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Система управління житловим приміщенням на базі ESP32 і Zigbee з адаптивним регулюванням споживання

затверджена наказом університету від _____ 6 травня 2025 р. № 403

2. Термін подання здобувачем роботи до екзаменаційної комісії _____ 21 червня 2025 р.

3. Вихідні дані до роботи: інформація про мікроконтролер ESP32, протокол Zigbee, дані сенсорів температура, вологість, освітленість, рух

4. Перелік питань, що потрібно опрацювати в роботі _____ Аналіз технологій та стандартів автоматизації житла. Розробка апаратної та програмної архітектури. Реалізація протоколів комунікації. Тестування та оцінка ефективності системи.

5.Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

1бслайдів

6. Консультанти розділів роботи (проекту)


Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	19.04.2025-20.04.2025	Виконано
2	Аналіз предметної області та завдання до роботи	21.04.2025-26.04.2025	Виконано
3	Аналіз існуючих аналогів	27.04.2025-29.04.2025	Виконано
4	Аналіз структури та характеристик	2.05.2025-10.05.2025	Виконано
5	Вибір програмних засобів для розробки	13.05.2025-23.05.2025	Виконано
6	Оформлювання пояснювальної записки	24.05.2025-11.06.2025	Виконано
7	Перевірка кваліфікаційної роботи на плагіат	15.06.2025-17.06.2025	Виконано
8	Підготовка презентації	18.06.2025-20.06.2025	Виконано

Дата видачі завдання 19 травня 2025 р.

Здобувач  Гладких О.Р.
(підпис)

Керівник роботи  ас.Хаханов І.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 58 с., 2 рис., 6 табл., 20 джерел.

ESP32, Zigbee, РОЗУМНИЙ ДІМ, ЕНЕРГОЕФЕКТИВНІСТЬ, АДАПТИВНЕ РЕЗУЛЮВАННЯ, АВТОМАТИЗАЦІЯ, СЕНСОРИ, МІКРОКОНТРОЛЕР, ПРОГРАМУВАННЯ, IoT.

У роботі розглядаються сучасні підходи до автоматизації житла з використанням мікроконтролерів ESP32 і бездротового протоколу Zigbee. Проведено аналіз функціональних можливостей компонентів, архітектури smart-home систем, розроблено апаратну та програмну частину, реалізовано адаптивну логіку прийняття рішень, яка забезпечує зниження споживання електроенергії без втрати комфорту.

У рамках дослідження реалізовано прототип системи, який здатен:

- автоматично вмикати освітлення за виявлення руху;
- регулювати мікроклімат на основі температури та вологості;
- змінювати інтенсивність світла залежно від природного освітлення;
- передавати дані до веб-інтерфейсу через Wi-Fi;
- забезпечувати стабільний зв'язок між пристроями за допомогою Zigbee.

ABSTRACT

Explanatory note of qualified work: 58 p., 2 fig., 6 table, 20 ger.

ESP32, Zigbee, SMART DIM, ENERGY EFFICIENCY, ADAPTIVE RESOLUTION, AUTOMATION, SENSOR, MICROCONTROLLER, PROGRAMMING, IoT.

The robot considers modern approaches to automation of life using ESP32 microcontrollers and the drone-free Zigbee protocol. An analysis of the functional capabilities of components, the architecture of smart-home systems was carried out, the hardware and software parts were broken down, and adaptive logic was implemented to make decisions that would ensure a reduction in energy consumption without sacrificing comfort.

As part of the research, a prototype of the system was implemented, which was created:

- automatically turn on the lighting for the detected signal;
- regulate the microclimate based on temperature and humidity;
- change the light intensity according to natural lightening;
- transfer data to the web interface via Wi-Fi;
- ensure a stable connection between devices using Zigbee.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП	10
1. ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМ УПРАВЛІННЯ ЖИТЛОВИМ ПРИМІЩЕННЯМ	13
1.1 Історія розвитку систем розумного дому.....	13
1.2 Огляд бездротових технологій для автоматизації	144
1.3 ESP32 — характеристика, переваги, використання в автоматизації	16
1.4 Адаптивне регулювання енергоспоживання.....	18
1.5 Альтернативні апаратні платформи	19
2. АНАЛІЗ ТЕХНОЛОГІЙ І СУЧАСНИХ РІШЕНЬ	21
2.1 Характеристика житлового приміщення як об'єкта управління	21
2.2 Склад та функції системи управління	23
2.3 Визначення функціональних і нефункціональних вимог	25
2.4 Постановка задачі.....	27
3. ПРОЄКТУВАННЯ СИСТЕМИ	30
3.1 Архітектура програмно-апаратного комплексу.....	30
3.2 Вибір та обґрунтування апаратних засобів	32
3.3 Структура програмного забезпечення.....	34
3.4 Мережеві протоколи і взаємодія компонентів.....	36
3.5 Інтерфейс користувача	40
4. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ	43
4.1 Реалізація апаратної частини	43
4.2 Розробка програмного забезпечення	47
4.3 Інтеграція та взаємодія компонентів	48
4.4 Тестування системи.....	51
ВИСНОВКИ	54
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	56

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API – Application Programming Interface – інтерфейс прикладного програмування.

BH1750 – цифровий датчик освітленості, що передає дані по інтерфейсу I²C.

DHT22 – цифровий сенсор температури та вологості.

ESP32 – мікроконтролер з підтримкою Wi-Fi та Bluetooth, виробництва компанії Espressif Systems.

GPIO – General Purpose Input/Output – універсальний вхід/вихід загального призначення.

GUI – Graphical User Interface – графічний інтерфейс користувача.

Home Assistant – платформа з відкритим кодом для автоматизації житлових приміщень.

HTTP – HyperText Transfer Protocol – протокол передавання гіпертексту.

IDE – Integrated Development Environment – інтегроване середовище розробки.

IoT – Internet of Things – Інтернет речей.

LED – Light Emitting Diode – світлодіод.

MQTT – Message Queue Telemetry Transport – протокол телеметричної передачі повідомлень.

PIR – Passive Infrared Sensor – пасивний інфрачервоний сенсор руху.

PWM – Pulse Width Modulation – широтно-імпульсна модуляція.

REST – Representational State Transfer – архітектурний стиль для побудови веб-сервісів.

RGB – Red-Green-Blue – система передачі кольору у світлодіодах або дисплеях.

UART – Universal Asynchronous Receiver-Transmitter – універсальний асинхронний приймач-передавач.

UPS – Uninterruptible Power Supply – джерело безперебійного живлення.

Wi-Fi – Wireless Fidelity – технологія бездротової передачі даних.

Zigbee – бездротовий протокол зв'язку для малопотужних пристроїв у мережах IoT.

ВСТУП

Сучасні тенденції розвитку технологій у галузі інформаційних систем, автоматизації та електроніки дедалі активніше інтегруються в повсякденне життя людини. Одним із яскравих проявів цієї інтеграції є поява та активне впровадження концепції «розумного дому» (smart home), яка передбачає автоматизоване управління інженерними системами житла з метою підвищення комфорту, безпеки, енергоефективності та зниження впливу людського чинника. Зростаюча популярність таких рішень обумовлена не лише технічним прогресом, але й підвищеними вимогами до якості життя, екологічності, економії ресурсів та персоналізованого керування середовищем.

У цьому контексті актуальним є створення адаптивних систем керування житловими приміщеннями, здатних автоматично реагувати на зміни умов середовища, поведінку користувача, час доби та інші зовнішні чинники. Особливого значення набувають рішення на базі відкритих апаратних і програмних платформ, таких як ESP32, Zigbee, MQTT, Home Assistant тощо. Вони поєднують у собі гнучкість, доступність, масштабованість і сумісність з широким спектром пристроїв, що дозволяє будувати індивідуальні, ефективні та надійні системи автоматизації.

Питання оптимального управління житловим простором охоплює низку аспектів: технічне проектування апаратного комплексу, розробку програмного забезпечення, забезпечення бездротового зв'язку між пристроями, створення системи прийняття рішень та реалізацію інтерфейсу користувача. Також важливим є дотримання принципів безпеки, відмовостійкості та енергоефективності, що особливо актуально в умовах зростаючих тарифів на електроенергію та необхідності раціонального використання ресурсів.

З огляду на це, в межах дипломної роботи поставлено за мету розробити комплексну, модульну систему управління житловим приміщенням, яка

базується на мікроконтролері ESP32 та сенсорах, що працюють за бездротовим протоколом Zigbee. Передбачається створення системи, яка дозволить контролювати параметри середовища (температуру, вологість, освітленість, рух, дим, відкриття вікон), керувати відповідними виконавчими елементами (реле, освітлення, обігрівачі, вентиляція), а також взаємодіяти з користувачем через зручний веб-інтерфейс та мобільний застосунок.

У процесі реалізації передбачається використання відкритих протоколів передачі даних (MQTT, HTTP), а також платформи Home Assistant для логіки керування і візуалізації. Такий підхід дозволяє створити масштабовану систему, яка не прив'язана до конкретного виробника і легко адаптується до потреб кінцевого користувача. Крім того, розглядається можливість автономної роботи системи при втраті з'єднання з мережею, що є важливим чинником надійності.

Актуальність теми дипломної роботи обумовлена потребою у створенні ефективних, доступних і відкритих рішень у сфері інтелектуальних систем життєзабезпечення, здатних не лише зменшити енергоспоживання, але й підвищити комфорт проживання, безпеку та рівень інтеграції технологій у повсякденне життя. Реалізація такої системи є прикладом сучасного підходу до автоматизації побуту та цифрової трансформації житлових середовищ.

Метою дипломної роботи є проектування, реалізація та тестування системи управління житловим приміщенням з функціями моніторингу середовища, адаптивного регулювання споживання ресурсів та інтеграції з платформами автоматизації на основі відкритих стандартів.

Для досягнення поставленої мети були сформульовані наступні завдання:

- провести аналіз сучасних технологій автоматизації житла та обґрунтувати вибір технічних засобів;
- розробити архітектуру апаратної та програмної частин системи;
- реалізувати базовий функціонал автоматичного керування середовищем на основі даних із сенсорів;

- забезпечити бездротову комунікацію між компонентами (ESP32, Zigbee, Home Assistant);
- протестувати систему та оцінити її ефективність, швидкодію та стабільність.

Таким чином, дана дипломна робота спрямована на розробку реального прикладного рішення, яке демонструє практичне впровадження технологій розумного дому на базі доступних засобів та відкритого програмного забезпечення.

1. ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМ УПРАВЛІННЯ ЖИТЛОВИМ ПРИМІЩЕННЯМ

1.1 Історія розвитку систем розумного дому

Поняття «розумного дому» виникло на межі 1970–1980-х років, коли почали з'являтися перші ідеї щодо автоматизації побутових процесів. У той час основною ціллю було зниження ручної праці за допомогою механічних та електронних пристроїв. Перші концепції реалізовувалися через централізовані системи з дротовими підключеннями, без підтримки зворотного зв'язку або адаптації до умов середовища.

У 1990-х роках відбувся технологічний прорив завдяки розвитку персональних комп'ютерів, що дозволило підключати пристрої до мереж і створювати більш складні системи управління. З'явилися перші стандарти, такі як X10, що використовував електромережу для передавання керуючих сигналів. У цей період почали розроблятися терміни «автоматизація будинку» та «інтелектуальні системи керування».

У 2000-х роках домінуючим напрямом став розвиток бездротових технологій. Відбулася поява таких рішень, як Z-Wave та Zigbee, а також перших комерційно доступних продуктів для домашньої автоматизації. Наприклад, система HomeSeer, Amazon Alexa, Google Nest дозволили користувачам здійснювати управління з мобільних пристроїв, за допомогою голосових команд та віддаленого доступу через інтернет.

Сьогодні розумний дім (Smart Home) включає такі функції, як управління освітленням, опаленням, вентиляцією, охоронною системою, мультимедіа та побутовою технікою. Завдяки IoT-пристроєм управління відбувається в реальному часі із врахуванням поведінки користувача та даних середовища.

Етапи розвитку технологій розумного дому:

- 1970–1980-ті роки – поява перших дротових систем (X10) для вмикання світла та керування побутовими приладами;
- 1990-ті роки – розширення можливостей систем, з’являються перші програмовані логічні контролери (ПЛК);
- 2000-ні роки – поширення Wi-Fi, Bluetooth та початок використання смартфонів як пульта керування;
- 2010+ – IoT-пристрої, хмарні сервіси, голосові помічники (Alexa, Siri, Google Assistant);
- 2020+ – інтеграція штучного інтелекту, адаптивні алгоритми, новий протокол Matter для сумісності між пристроями від різних виробників.

Серед найважливіших технологій слід виділити KNX (європейський стандарт для BMS), BACnet (промислові системи), HomeKit (Apple), SmartThings (Samsung), Matter (уніфікована платформа для IoT, підтримувана Google, Apple, Amazon).

В Україні розумні системи управління приміщенням здебільшого розвиваються в рамках приватних ініціатив або інтеграцій невеликих рішень. Проте з 2020 року в умовах енергетичної нестабільності та зростання тарифів на комунальні послуги значно зріс попит на енергоефективні технології. Зокрема, в багатьох містах діють муніципальні програми підтримки модернізації житлового фонду, у т.ч. з використанням систем моніторингу споживання.

Платформи на базі ESP32, Home Assistant, MQTT стали популярними серед ентузіастів, оскільки дозволяють створювати індивідуальні рішення без прив’язки до закритих екосистем.

1.2 Огляд бездротових технологій для автоматизації

Одним з найважливіших факторів ефективності смарт-систем є тип мережевої технології, яку використовують пристрої для взаємодії. Розглянемо основні технології, що використовуються:

Wi-Fi— один із найпоширеніших стандартів. Переваги: висока пропускна здатність і широка підтримка. Недоліки: високе енергоспоживання, нестабільність при великій кількості пристроїв.

Bluetooth і BLE (Bluetooth Low Energy) — ефективні для пристроїв з низьким енергоспоживанням. Підходять для носимих девайсів та пристроїв на короткій відстані.

Zigbee — розрахований на побудову мереж типу mesh. Підтримує до 65 000 пристроїв у мережі. Ефективний у малопотужних пристроях, має низьке енергоспоживання та високу надійність.

Z-Wave — альтернатива Zigbee, також підтримує mesh. Має кращу взаємодію між виробниками, але менше підтримки з боку відкритого ПЗ.

LoRa/LoRaWAN — підходить для передачі невеликих обсягів даних на великі відстані (до кількох кілометрів), але не забезпечує швидкої взаємодії для інтенсивних сценаріїв у домі.

Таблиця 1.1 - Порівняльна характеристика бездротових технологій

Технологія	Радіус дії	Швидкість	Енергоспожи вання	Мережа	Підтримка ІоТ
Wi-Fi	30–50 м	Висока	Високе	Зірка	Так
Zigbee	10–100 м	Середня	Низьке	Mesh	Так
Z-Wave	30–100 м	Середня	Низьке	Mesh	Так
BLE	5–10 м	Низька	Низьке	Зірка	Частково
LoRa	до 10 км	Дуже низька	Дуже низьке	Зірка	Обмежено

Бездротові технології є серцевиною сучасного розумного дому. Типова мережа включає різні протоколи в залежності від вимог до швидкості, стабільності та енергоспоживання.

Топології бездротових мереж:

- Зірка (Star) — всі пристрої підключені до центрального вузла;
- Mesh — кожен пристрій може ретранслювати сигнал, покращуючи покриття;
- Дерево (Tree) — ієрархічна модель для масштабних структур.

Новий протокол – Matter (Project CHIP): Matter забезпечує безпечну, відкриту взаємодію пристроїв різних виробників. Працює поверх Wi-Fi, Ethernet, Thread, і дозволяє ESP32 підтримувати декілька екосистем одночасно.

Приклади застосування Zigbee:

- Освітлення: Philips Hue, IKEA Tradfri
- Безпека: Aqara датчики руху, відкриття дверей
- Енергетика: розумні розетки, реле з енергомоніторингом

Протокол Thread

Thread — мережевий протокол на базі IPv6, розроблений для розумного дому. Він підтримує mesh-архітектуру, має низьке енергоспоживання, високу швидкодію та створює базу для протоколу Matter. Thread дозволяє створювати самовідновлювані та самоконфігуровані мережі без централізованого маршрутизатора.

Протоколи рівнів Zigbee:

- PHY – фізичний рівень (радіосигнал);
- MAC – контроль доступу до середовища;
- NWK – маршрутизація, адресація;
- APS – прикладна підтримка шифрування та сервісів;
- ZCL – бібліотека команд (класифікація пристроїв: освітлення, охорона, енергозбереження).

1.3 ESP32 — характеристика, переваги, використання в автоматизації

ESP32 — це 32-бітний мікроконтролер виробництва Espressif Systems, який підтримує Wi-Fi та Bluetooth і має велику кількість периферійних

інтерфейсів. Він обладнаний двоядерним процесором Tensilica LX6 з частотою до 240 МГц, що робить його потужним і гнучким рішенням для IoT.

Серії ESP32:

- ESP32-WROOM – базова версія
- ESP32-C3 – одноядерний, з підтримкою RISC-V
- ESP32-S3 – для нейромереж і AI (ML-процесор, розпізнавання обличчя)

Можливості в автоматизації:

- зчитування з датчиків температури, вологості, присутності
- керування реле, LED, серводвигунами
- обмін даними з брокером MQTT або сервером Home Assistant

Архітектура ESP32:

- CPU
- Wi-Fi/Bluetooth MAC
- RTC контролер
- GPIO (входи/виходи)
- PWM/ADC/Timer
- Secure Boot, Flash Encryption

Основні технічні характеристики:

- До 520 КБ SRAM
- До 520 КБ SRAM
- До 16 МБ Flash пам'яті
- Wi-Fi 802.11 b/g/n, Bluetooth v4.2
- GPIO, I2C, SPI, UART, PWM, ADC, DAC
- Підтримка FreeRTOS, Arduino, ESP-IDF

Завдяки широкій підтримці, відкритій документації та активній спільноті ESP32 активно використовується у домашній автоматизації. Особливо він корисний у поєднанні з Zigbee- або Matter-модулями, коли виступає шлюзом або головним контролером.

Інтеграція з Home Assistant:

ESP32 можна підключити до локального сервера Home Assistant через MQTT або REST API, передаючи інформацію про стан датчиків і приймаючи команди на керування. Це дає змогу централізовано управляти системою без зовнішніх хмарних платформ.

Таблиця 1.2 - — Приклад адаптивного сценарію

Умова	Дія
Присутність відсутня > 10 хв	Вимкнути світло, обігрівач
Зовнішня температура < 5 °C	Увімкнути опалення
Вікно відкрите > 3 хв	Вимкнути обігрівач у кімнаті

1.4 Адаптивне регулювання енергоспоживання

Сучасні будівлі все частіше оснащуються системами, що не тільки керують пристроями, але й оптимізують їх енергоспоживання. Адаптивні системи враховують зміну навколишніх умов, наявність людей у приміщенні, погодні умови та інші фактори.

Алгоритми адаптивного управління можуть використовувати машинне навчання для прогнозування поведінки мешканців і автоматичного коригування роботи пристроїв. Наприклад, система може вмикати обігрів лише тоді, коли мешканець знаходиться вдома, або регулювати освітлення відповідно до кількості денного світла. Інший напрям — інтеграція з енергомережами (Smart Grid). У цьому випадку система здатна реагувати на сигнали енергопостачальників, змінюючи режими роботи пристроїв відповідно до пікових навантажень і тарифів.

Системи управління житлом повинні бути не тільки зручними, а й захищеними. Основні проблеми безпеки:

- Вразливість бездротових протоколів (перехоплення трафіку, несанкціонований доступ);

- Несанкціоноване керування обладнанням (розумні замки, камери);
- Недостатня автентифікація користувачів.

Для забезпечення безпеки застосовуються такі заходи:

- Шифрування даних (AES, TLS);
- Локальна обробка даних, мінімізація залежності від хмари;
- Використання перевірених прошивок, регулярне оновлення OTA.

ESP32 підтримує апаратне шифрування та зберігання ключів. Zigbee має вбудовану систему автентифікації вузлів і шифрування мережевого рівня.

Масштабованість досягається завдяки використанню топології mesh, при якій кожен пристрій може передавати сигнали іншим. Це дозволяє покрити велику площу з мінімальними втратами сигналу та без надмірного навантаження на центральний вузол.

1.5 Альтернативні апаратні платформи

Хоча ESP32 має велику популярність, існують інші мікроконтролери та платформи для створення автоматизованих систем:

- Raspberry Pi — одноплатний комп'ютер із Linux на борту, підходить для складних обчислень і серверних рішень.
- STM32 — мікроконтролери із високою продуктивністю та гнучкістю, часто використовуються у промисловості.
- Arduino — спрощена платформа, яка підходить для новачків, але обмежена в порівнянні з ESP32.
- BeagleBone — альтернатива Raspberry Pi із кращою підтримкою вводу/виводу.

Порівняння таких платформ дозволяє вибрати оптимальну за ціною, продуктивністю та енергоспоживанням.

Таблиця 1.3 - Порівняння апаратних платформ

Платформа	Потужність	Wi-Fi	Ціна	Програмування
ESP32	Висока	Так	\$4	Arduino, C/C++
Raspberry Pi	Дуже висока	Так	\$35+	Linux, Python
STM32	Середня	Ні	\$5–15	C/C++
Arduino Uno	Низька	Ні	\$3–5	Arduino IDE

Застосування:

- Raspberry Pi – для локального сервера Home Assistant
- STM32 – у промислових системах зі спеціалізованим керуванням
- Arduino – як простий сенсорний вузол

2. АНАЛІЗ ТЕХНОЛОГІЙ І СУЧАСНИХ РІШЕНЬ

2.1 Характеристика житлового приміщення як об'єкта управління

Житлове приміщення є складним об'єктом з погляду автоматизації, оскільки воно поєднує в собі безліч зон з різними умовами експлуатації, типами навантаження, а також індивідуальними вимогами до комфорту, безпеки та енергоефективності. У структурі такого об'єкта традиційно виділяють кілька ключових функціональних зон: спальню, вітальню, кухню, ванну кімнату, передпокій, балкон або лоджію. Кожна із цих зон має свої особливості експлуатації, які визначають набір пристроїв, що підлягають автоматизованому керуванню.

Наприклад, у спальні першочергову увагу приділяють комфорту – тут актуальне автоматичне керування освітленням, температурою, вентиляцією. У вітальні переважають мультимедійні пристрої та сценарії, пов'язані з перебуванням великої кількості людей: керування світлом, розетками, системою кондиціонування. Кухня вимагає наявності датчиків диму, газу, температури, а також витяжної системи, яка може вмикатись автоматично за показниками сенсорів. Ванна кімната часто обладнується датчиками вологості, протікання, а також вентиляцією, що активується залежно від рівня вологості повітря.

Важливо відзначити, що сучасне житло споживає значну кількість електроенергії. За статистикою, найбільше енергії витрачається на опалення (до 40% загального бюджету), освітлення (до 20%), а також роботу побутових приладів (холодильники, пральні машини, бойлери, плити тощо). Це створює передумови для впровадження інтелектуального енергоменеджменту, який дозволяє автоматично регулювати роботу приладів, оптимізуючи споживання електроенергії залежно від наявності людей, часу доби, поточного тарифу або навіть погодних умов.

Окрім енергоефективності, автоматизація у житлі спрямована на підвищення рівня безпеки. Інтеграція сенсорів руху, відкриття дверей і вікон, диму та газу дозволяє оперативно реагувати на нестандартні або потенційно небезпечні ситуації. Наприклад, при виявленні відкритого вікна взимку може бути автоматично вимкнено обігрів у цій кімнаті. У разі виявлення диму – надсилається сповіщення користувачу або вмикається звуковий сигнал.

Ще одним ключовим аспектом є забезпечення комфортного мікроклімату. Завдяки сенсорам температури та вологості система має можливість підтримувати оптимальні умови в різних приміщеннях, враховуючи індивідуальні вподобання мешканців. Наприклад, уночі температура може знижуватись, а вранці — підійматись автоматично, не потребуючи втручання користувача. При наявності зовнішніх API, таких як погодні сервіси, система може прогнозувати коливання температури і заздалегідь адаптувати параметри роботи опалення або вентиляції.

Житлове приміщення як об'єкт управління також вимагає врахування фізичних обмежень – наприклад, не всі стіни або зони зручно охопити сигналом Wi-Fi. Саме тому доцільно використання бездротових протоколів, таких як Zigbee або Thread, які дозволяють формувати мережі типу mesh. Вони забезпечують надійну передачу даних навіть у складних умовах забудови, де є кілька кімнат і перекриттів. Розміщення пристроїв має бути таким, щоб досягти стабільного зв'язку, при цьому зберігаючи енергоефективність і мінімальне втручання в інтер'єр.

Таким чином, житлове приміщення — це складна система із великою кількістю змінних факторів, у якій автоматизація має вирішувати завдання одночасного забезпечення зручності, безпеки, ефективного використання ресурсів і адаптивної реакції на динамічні зміни. Це робить об'єкт автоматизації не просто технічним середовищем, а частиною концепції розумного життєвого простору, де технології максимально інтегруються у повсякденне життя людини.

Управління цим середовищем передбачає:

- Контроль параметрів мікроклімату (температура, вологість);
- Автоматичне керування освітленням залежно від присутності, часу доби та освітленості;
- Моніторинг безпеки (рух, дим, витік газу, відкриття вікон/дверей);
- Енергоменеджмент – зменшення споживання електроенергії шляхом оптимізації роботи обладнання.

Об'єкт автоматизації може включати:

- Кімнати (спальня, вітальня, кухня, ванна);
- Зовнішні зони (балкон, під'їзд);
- Інфраструктуру (освітлення, обігрів, охоронні системи);
- Пристрої керування (реле, актуатори, сенсори);
- Канали зв'язку (Zigbee, Wi-Fi).

2.2 Склад та функції системи управління

Система управління житловим приміщенням є комплексною, багаторівневою структурою, яка реалізує збір, обробку, передачу та візуалізацію даних з навколишнього середовища. Її головна функція полягає у підтриманні комфортних та безпечних умов проживання з одночасним зменшенням енергоспоживання. Основу системи становить центральний мікроконтролер ESP32, який поєднує в собі обчислювальну потужність, комунікаційні модулі та енергоефективність, що робить його оптимальним вибором для розумного дому.

ESP32 виконує роль основного логічного вузла системи, на якому реалізуються всі основні сценарії поведінки: від вмикання світла до комплексного енергоменеджменту. У систему інтегрується велика кількість сенсорів, що дозволяють отримувати інформацію про стан приміщення: температуру, вологість, наявність руху, рівень освітлення, задимленість,

відкриття вікон та дверей. Усі ці дані надходять до ESP32, який приймає рішення на основі попередньо визначених умов або сценаріїв.

Після обробки даних контролер ініціює відповідні дії, активуючи виконавчі механізми. До таких механізмів належать реле, які можуть вмикати або вимикати побутові прилади (наприклад, освітлення або обігрівачі), а також серводвигуни, які змінюють положення жалюзі, відкривають або закривають вентиляційні отвори. Система може включати також інфрачервоні передавачі, які емулюють сигнали пультів дистанційного керування, дозволяючи автоматично взаємодіяти з телевізорами, кондиціонерами чи іншою технікою, що підтримує ІЧ-протоколи.

Значну роль у функціонуванні системи відіграє рівень користувацької взаємодії. Для забезпечення зручності та гнучкості управління використовується веб-інтерфейс або мобільний додаток, що дозволяє власнику дистанційно відслідковувати стан системи, змінювати параметри сценаріїв або вручну активувати певні дії. У більш складних конфігураціях систему можна інтегрувати з платформою Home Assistant, що дозволяє створювати автоматизації на основі логіки «якщо – то» та використовувати голосове управління за допомогою Google Assistant або Amazon Alexa.

Важливою особливістю системи є використання різних типів бездротового зв'язку. Наприклад, модулі, що працюють за протоколом Zigbee, формують енергоефективну mesh-мережу, в якій кожен пристрій може передавати дані далі, таким чином розширюючи зону покриття і забезпечуючи надійність передачі. Шлюз Zigbee з'єднується з ESP32 або безпосередньо з сервером (наприклад, Raspberry Pi з Home Assistant) для централізованої координації дій. У той же час Wi-Fi використовується для обміну даними між ESP32 та локальним сервером або для передачі повідомлень користувачу.

Завдяки такій архітектурі система управління стає гнучкою, масштабованою і здатною до адаптації під різні типи приміщень. Вона може працювати як повністю локально, без зовнішніх сервісів, так і з інтеграцією у хмару чи з мобільними застосунками. Цей підхід дозволяє реалізувати як

прості сценарії (наприклад, увімкнення світла при вході до кімнати), так і складні автоматизовані моделі, які враховують комбінацію кількох параметрів, часу доби, статистичних даних та зовнішніх факторів, таких як погода або тариф на електроенергію.

Система управління житловим приміщенням має модульну структуру і складається з наступних основних компонентів:

- Центральний контролер;
- Сенсори (датчики);
- Актуатори;
- Інтерфейс користувача;
- Комунікаційний модуль.

Функції системи:

- Моніторинг усіх підключених сенсорів у реальному часі;
- Віддалене та локальне управління пристроями;
- Збір статистики споживання;
- Реакція на події за сценарієм (наявність, час, погодні умови).

2.3 Визначення функціональних і нефункціональних вимог

Нефункціональні вимоги до системи управління житловим приміщенням визначають її якісні характеристики, які безпосередньо впливають на користувацький досвід та стабільність експлуатації. Першочергово система повинна демонструвати високу надійність: у будь-який момент доби вона має забезпечувати виконання основних сценаріїв без збоїв, незалежно від того, чи є з'єднання з інтернетом або локальним сервером. Цього можна досягти шляхом реалізації автономної логіки на контролері ESP32, дублюванням критичних сценаріїв та наявністю резервного живлення.

Ще одним критично важливим аспектом є безпека. Враховуючи постійне зростання кіберзагроз, система повинна гарантувати захищену передачу даних усередині мережі. Це означає впровадження сучасних

протоколів шифрування, автентифікації користувачів і перевірки цілісності повідомлень. Водночас, система не повинна бути надто складною для користувача: простий інтерфейс входу, керування дозволами, а також логування дій підвищують її захищеність і контрольованість.

Щодо швидкодії, система повинна оперативно реагувати на вхідні події. Наприклад, виявлення руху або зміни освітленості повинні оброблятися не пізніше ніж за 1–2 секунди, щоб користувач не відчував затримки між тригером та відповідною дією (наприклад, увімкненням світла). Висока швидкодія підвищує суб'єктивне враження від системи та її практичну цінність.

Масштабованість також є ключовим параметром, оскільки будь-яке житло з часом може змінюватися: додаються нові кімнати, пристрої, сценарії. Розроблена система повинна дозволяти легко підключати нові модулі, не вимагаючи перепрошивки або кардинальної перебудови логіки. Для цього архітектура має бути модульною, а комунікація – стандартизованою (MQTT, REST API).

Не менш важливим є енергоспоживання. Оскільки в системі можуть бути десятки вузлів, їхнє сумарне енергоспоживання має бути мінімальним. Це особливо стосується пристроїв, що працюють від батарейок (датчики руху, температури, відкриття). Завдяки використанню енергоефективних мікроконтролерів та протоколів (наприклад, Zigbee), система зможе функціонувати тривалий час без обслуговування.

Функціональні вимоги:

- Система повинна автоматично регулювати освітлення, опалення, вентиляцію;
- Забезпечувати зчитування даних з усіх сенсорів та їх обробку;
- Надсилати сповіщення користувачу в разі виявлення нестандартної ситуації;
- Підтримувати сценарії (наприклад: “ніч”, “відсутність вдома”, “гості”).

Нефункціональні вимоги:

- Надійність: система має працювати безперервно, із захистом від збоїв;
- Безпека: захист переданих даних, автентифікація користувача;
- Швидкодія: реакція на події повинна бути не більше 1–2 секунд;
- Масштабованість: можливість додавання нових пристроїв без змін у логіці;
- Енергоефективність: низьке споживання електроенергії вузлами системи.

Система також повинна враховувати локальні обмеження: нестабільність інтернет-з'єднання, відключення електроенергії, залежність від Wi-Fi мережі.

2.4 Постановка задачі

Після проведеного аналізу предметної області постає необхідність формалізації основного завдання, яке вирішується в межах даної дипломної роботи. У даному контексті мова йде про розробку, побудову та часткову реалізацію системи автоматизованого керування житловим приміщенням на базі мікроконтролера ESP32 з використанням бездротової технології Zigbee. Кінцевою метою є створення адаптивного, надійного і масштабованого рішення, здатного автоматично керувати освітленням, опаленням, вентиляцією, а також реагувати на події безпеки (рух, відкриття вікон/дверей, дим) без прямої участі користувача.

Вихідними даними для системи виступають показники з різноманітних сенсорів, які фіксують фізичні параметри навколишнього середовища: температуру, рівень освітленості, вологість повітря, присутність у приміщенні, стан вікон і дверей. Крім цього, система може інтегрувати інформацію з зовнішніх джерел, зокрема погодних API, а також отримувати часові мітки або сигнали про зміну тарифів на електроенергію. Усі ці вхідні параметри

передаються на обробку центральним модулем, який аналізує їх відповідно до задалегідь визначених логічних умов або навченої моделі прийняття рішень.

Завдання, що ставиться перед системою, передбачає не лише виконання реакційних дій на основі отриманих значень, але й реалізацію сценаріїв адаптивного управління. Наприклад, у разі виявлення присутності в кімнаті та недостатнього природного освітлення — система повинна самостійно активувати штучне світло. Якщо при цьому вікно відкрите, а температура нижча за допустиму — має відбуватися блокування або корекція обігріву, щоб уникнути перевитрат енергії. У разі виявлення руху вночі — система може активувати приглушене підсвічування для безпечного пересування.

Також важливо, щоб система могла не лише керувати пристроями, а й здійснювати моніторинг виконання дій, фіксацію аномальних станів і передачу відповідної інформації до користувача. Це підвищує прозорість її функціонування і дозволяє виявляти потенційні збої. Крім того, завдання включає реалізацію автономності – навіть при відсутності доступу до інтернету система повинна продовжувати функціонувати згідно з критичними сценаріями.

У межах цієї задачі доцільно також розглядати питання вимірювання ефективності роботи системи. Ключовими метриками можуть слугувати: середній час реакції на подію, ступінь енергозбереження у порівнянні з ручним керуванням, кількість помилкових або повторних спрацювань системи, рівень зручності використання (на основі опитування користувача). За наявності статистичних даних також можливо здійснювати аналіз тенденцій споживання енергії в різних режимах роботи, що в подальшому може стати основою для навчання системи та підвищення її інтелектуальності.

Таким чином, формалізована задача охоплює створення системи, яка буде не лише інструментом керування пристроями в житловому середовищі, а й повноцінною платформою для підвищення ефективності споживання енергії, безпеки та комфорту мешканців, з можливістю адаптації до умов, що змінюються.

На основі аналізу предметної області було сформульовано задачу розробки інтелектуальної системи управління житловим приміщенням з такими цілями:

- Підвищення енергоефективності за рахунок адаптивного регулювання;
- Підвищення безпеки та зручності використання приміщення;
- Можливість моніторингу та віддаленого керування параметрами середовища.

Вхідні дані:

- Дані з сенсорів (температура, рух, освітлення тощо);
- Інформація про час доби, день тижня, присутність мешканців;
- Дані від зовнішніх сервісів (погода, тариф);

Цільові функції:

- Автоматичне керування освітленням, обігрівом, вентиляцією;
- Сповіщення про відхилення (відкриті двері, витік газу);
- Ведення статистики споживання електроенергії.

Таблиця 2.1 - Сценарій взаємодії компонентівОбмеження

Компонент	Взаємодія
Датчик руху	Виявлення присутності → увімкнення світла
Датчик світла	Низький рівень → дозвіл на освітлення
Температура	< 18 °C → увімкнення обігрівача

Опис:

- Максимальна затримка реакції – 2 с;
- Робота у локальній мережі без хмари;
- Автономна робота при втраті інтернету протягом 24 годин.

Задача передбачає створення прототипу системи з можливістю її подальшої модернізації та масштабування під інші об'єкти.

3. ПРОЄКТУВАННЯ СИСТЕМИ

3.1 Архітектура програмно-апаратного комплексу

Проектована система управління житловим приміщенням базується на концепції розподіленого програмно-апаратного комплексу, де кожен функціональний елемент виконує окрему роль у структурі взаємодії. Основна мета архітектурного проєктування полягає у досягненні надійної, масштабованої та гнучкої платформи, яка забезпечує автоматизоване управління побутовими процесами із врахуванням змін середовища, поведінки користувачів та умов енергоспоживання.

Архітектура системи побудована за принципом «периферійний збір — локальна обробка — централізоване керування». Це означає, що дані з численних сенсорних вузлів (датчиків температури, руху, освітленості тощо) передаються до центрального обчислювального блоку — мікроконтролера ESP32, який реалізує локальну логіку керування. Далі, у разі необхідності, відбувається обмін інформацією з центральним сервером (локальним або хмарним) — наприклад, з інсталяцією Home Assistant, що розміщена на Raspberry Pi або іншій платформі.

Фізична структура системи складається з трьох рівнів: рівень сенсорів та виконавчих пристроїв, рівень контролера (ESP32), та серверний рівень. Сенсорні пристрої, як правило, працюють за протоколом Zigbee, що забезпечує надійний та енергоефективний обмін даними навіть у складних умовах побутової забудови. Вони формують так звану mesh-мережу, у якій кожен вузол може ретранслювати повідомлення, що збільшує зону покриття та підвищує відмовостійкість.

ESP32 виконує роль локального контролера, який приймає рішення на основі логіки, закладеної у прошивку, та параметрів, отриманих із сенсорів.

Він має вбудовані модулі Wi-Fi та Bluetooth, а також підтримує підключення Zigbee-модуля (наприклад, через UART), що дозволяє йому діяти як координаційна ланка між мережею пристроїв і центральною системою керування. ESP32 також здійснює керування реле, серводвигунами, інфрачервоними передавачами тощо.

На верхньому рівні архітектури функціонує сервер логіки, що виконує аналітичну обробку, візуалізацію даних, зберігання журналів та забезпечення доступу користувача до системи. Найчастіше для цих цілей використовується Home Assistant — універсальна платформа з відкритим кодом, яка підтримує інтеграцію з великою кількістю пристроїв та протоколів. Через веб-інтерфейс або мобільний застосунок користувач має можливість спостерігати за станом приміщення, керувати сценаріями, а також переглядати історію подій та споживання ресурсів.

Комунікація між компонентами реалізується через комбінацію протоколів MQTT та Zigbee. MQTT використовується як брокер повідомлень між ESP32 та сервером, що дозволяє забезпечити асинхронну, подіє-орієнтовану модель взаємодії. Завдяки цьому кожна зміна стану (наприклад, увімкнення світла або виявлення руху) миттєво передається через відповідні теми (topics), забезпечуючи реакцію інших вузлів або оновлення інтерфейсу.

Такий підхід дозволяє досягти високої модульності та масштабованості системи. У разі потреби в розширенні — наприклад, додавання нових приміщень або пристроїв — достатньо встановити новий сенсор або модуль, який автоматично приєднується до наявної інфраструктури за допомогою Zigbee-мережі або підключення до MQTT-брокера. Система не вимагає серйозного перепроектування, оскільки її логіка будується на подієво-орієнтованій моделі та підтримує автоматичне виявлення нових пристроїв.

У результаті, запропонована архітектура забезпечує гнучке й надійне керування житловим середовищем, що адаптується до потреб користувача та змін зовнішніх умов. Вона дозволяє реалізовувати як прості задачі (ввімкнення/вимкнення освітлення), так і складні сценарії поведінки системи

на основі комбінованих умов, статистичних даних або даних з API сторонніх сервісів.

3.2 Вибір та обґрунтування апаратних засобів

На етапі проєктування програмно-апаратного комплексу критичним є правильний вибір апаратних компонентів, які забезпечують необхідний функціонал, відповідну продуктивність, енергоефективність, стабільність у роботі, а також взаємну сумісність. У межах реалізації даного проєкту було прийнято рішення використовувати апаратні засоби, що поєднують у собі доступність, гнучкість конфігурації та підтримку відкритих стандартів, що дозволяє легко масштабувати систему в майбутньому.

Ключовим обчислювальним елементом обрано мікроконтролер ESP32, який завдяки своїй двоядерній архітектурі, підтримці Wi-Fi і Bluetooth, широкому спектру периферійних інтерфейсів (GPIO, UART, I2C, SPI, PWM тощо) та низькому енергоспоживанню, ідеально підходить для завдань автоматизації. Платформа ESP32 підтримується великою спільнотою розробників, а також широким набором бібліотек, що суттєво пришвидшує розробку прошивки, дозволяючи використовувати як середовище Arduino IDE, так і потужніший ESP-IDF.

Для реалізації сенсорної частини системи обрано пристрої, що працюють за протоколом Zigbee. Вибір цієї технології зумовлений її перевагами у побутовому застосуванні: Zigbee характеризується низьким енергоспоживанням, високою стійкістю до перешкод, підтримкою топології mesh та широким набором сумісних пристроїв. Як шлюз для мережі Zigbee було використано модуль CC2652P, що підключається до сервера через USB та виконує функцію координатора мережі. Цей пристрій підтримує протоколи Zigbee 3.0, Thread і Matter, що відкриває можливості майбутнього розширення системи.

У складі сенсорів перевагу надано найпоширенішим, перевіреним модулям: датчик температури та вологості DHT22 дозволяє отримати базову інформацію про мікроклімат у приміщенні; датчик руху типу PIR забезпечує виявлення присутності людей; сенсор освітленості BH1750 використовується для адаптивного керування освітленням. Для моніторингу безпеки передбачено застосування датчиків диму та газу, таких як MQ-2, які можуть виявляти шкідливі речовини та підвищений рівень чадного газу. Для фіксації положення вікон або дверей застосовуються геркони — прості, але надійні магнітні сенсори.

В якості виконавчих елементів застосовуються реле (механічні або твердотільні), які дозволяють дистанційно вмикати або вимикати освітлення, обігрівачі, розетки, тощо. У разі потреби керування положенням вентиляційних отворів або жалюзі, до системи додаються сервоприводи, які отримують сигнал від ESP32 або Zigbee-шлюзу. Додатково можуть використовуватись інфрачервоні передавачі (наприклад, на базі модуля VS1838B) для емуляції команд пультів дистанційного керування побутовими пристроями.

У якості центрального вузла, що виконує логіку сценаріїв та надає інтерфейс користувача, застосовується мікрокомп'ютер Raspberry Pi 3 або 4 з встановленим програмним забезпеченням Home Assistant. Такий вибір забезпечує гнучкість і масштабованість, а також дає можливість обробляти події у реальному часі, вести журнал змін, зберігати історію даних, забезпечувати локальний інтерфейс керування без підключення до хмари.

Окремо варто зазначити, що всі компоненти підбиралися з урахуванням їх сумісності, стабільності роботи та доступності. Пристрої повинні мати підтримку OTA (оновлення прошивки по повітрю), працювати з відкритими стандартами протоколів і бути придатними для роботи в безперервному режимі 24/7. Система побудована таким чином, щоб у разі виходу з ладу одного з вузлів загальна функціональність не втрачалася — наприклад,

завдяки локальній автономній логіці на ESP32. Загалом обрані апаратні компоненти дозволяють реалізувати ефективну, масштабовану і безпечну систему управління житловим середовищем з можливістю адаптації під конкретні потреби користувача, без прив'язки до закритих екосистем або хмарних сервісів сторонніх виробників.

3.3 Структура програмного забезпечення

Функціонування системи управління житловим приміщенням ґрунтується на взаємодії кількох рівнів програмного забезпечення, кожен з яких виконує специфічні задачі — від низькорівневої взаємодії з апаратними модулями до інтерфейсу користувача. Основу цієї багаторівневої структури складає прошивка мікроконтролера ESP32, яка реалізує логіку реагування на події в реальному часі та забезпечує обмін даними з іншими компонентами системи. Над нею розташований рівень передачі повідомлень — MQTT-брокер, що виступає у ролі посередника між пристроями, а також серверна частина з візуалізацією та автоматизацією сценаріїв.

На базовому рівні програмного забезпечення розташована прошивка ESP32, яка реалізована за допомогою фреймворків Arduino або ESP-IDF. Arduino IDE було обрано через простоту, швидкість розробки та наявність широкої підтримки сторонніх бібліотек. Ця прошивка включає ініціалізацію сенсорів, зчитування їхніх значень, обробку сигналів від датчиків руху, температури, вологості, диму, а також керування виконавчими механізмами — зокрема реле, сервомоторами, інфрачервоними передавачами. Реалізована логіка дозволяє задавати базові умови керування (наприклад, «якщо температура менше за 18°C — увімкнути обігрівач»), а також передавати зібрані дані через Wi-Fi до зовнішніх систем.

Для обміну повідомленнями між мікроконтролером та іншими частинами системи використовується протокол MQTT (Message Queue Telemetry Transport). MQTT-брокер може бути реалізований як на локальному

сервері (наприклад, Mosquitto на Raspberry Pi), так і в хмарі. Протокол дозволяє організувати асинхронну подіє-орієнтовану взаємодію, де кожен компонент публікує та підписується на певні теми (topics), завдяки чому можлива побудова децентралізованої архітектури. Наприклад, ESP32 може публікувати повідомлення у тему `sensors/livingroom/temperature`, тоді як Home Assistant підписується на неї та реагує відповідно до заданих автоматизацій.

На вищому рівні працює програмне забезпечення для візуалізації, обробки даних і реалізації складних сценаріїв. У проєкті використовується Home Assistant — універсальна платформа з відкритим кодом, яка дозволяє налаштовувати автоматизації за допомогою YAML-конфігурацій або графічного інтерфейсу. Home Assistant об'єднує дані з усіх сенсорів, виводить їх у вебінтерфейсі, дозволяє створювати дашборди, керувати пристроями вручну або за допомогою голосових команд. Крім того, через інтеграцію з Zigbee-модулем (через ZHA або Zigbee2MQTT) користувач може додавати нові пристрої без перекомпіляції прошивки на ESP32.

Особливу увагу під час проєктування програмної частини було приділено стійкості до помилок і збоїв. Логіка на ESP32 передбачає використання механізмів watchdog для автоматичного перезапуску при зависанні, обробку нештатних ситуацій (наприклад, втрата зв'язку з сенсором), локальне кешування даних при відсутності з'єднання. У Home Assistant ведеться логування всіх подій, що дозволяє діагностувати проблеми та відстежувати аномальну поведінку системи.

Загалом програмне забезпечення системи забезпечує плавну інтеграцію між апаратною частиною та користувачем, забезпечуючи швидку реакцію на події, надійність виконання сценаріїв, зручність керування і відкритість до масштабування. Завдяки використанню відкритих платформ і стандартів, система легко адаптується під індивідуальні вимоги користувача і може бути розширена в майбутньому як за функціональністю, так і за кількістю пристроїв.

Таблиця 3.1 – Компоненти та причини вибору

Компонент	Причина вибору
ESP32	Потужний мікроконтролер із вбудованим Wi-Fi, сумісний з Zigbee-модулями
Zigbee-модуль (наприклад, CC2530)	Для зв'язку з іншими IoT-пристроями, масштабування
DHT22 або BME280	Сенсор температури/вологості з високою точністю
HC-SR501	Інфрачервоний сенсор руху, недорогий і надійний
Реле 5V або 3.3V	Для перемикання освітлення, обігрівача та вентиляції
Блок живлення 5V	Для живлення всієї системи
Інтерфейси: Wi-Fi, UART, I2C	Для комунікації з модулями та користувачем

3.4 Мережеві протоколи і взаємодія компонентів

Ефективне функціонування автоматизованої системи управління житловим приміщенням безпосередньо залежить від налагодженої взаємодії між її компонентами. Враховуючи багаторівневу структуру системи, де присутні як сенсорні пристрої, так і контролери, виконавчі механізми, сервери та інтерфейси, критичним є вибір і реалізація відповідних мережевих протоколів. Саме ці протоколи забезпечують обмін інформацією між фізичними модулями, логічними процесами та користувацькими інтерфейсами в реальному часі з гарантією надійності, швидкодії та захисту даних.

У проєктованій системі використовується комбінація двох ключових протоколів: Zigbee та MQTT. Zigbee обраний для взаємодії між сенсорами та шлюзом завдяки його енергоефективності, підтримці топології mesh і

стійкості до перешкод у багатокімнатних приміщеннях. Він працює у частотному діапазоні 2.4 ГГц і дозволяє кожному пристрою не лише передавати дані, а й ретранслювати пакети інших пристроїв, що значно покращує покриття в приміщенні. Сенсори Zigbee підключаються до центрального координатора (зазвичай, USB-донгла на базі CC2652P), який інтегрується з локальним сервером Home Assistant через один з двох стеків: ZHA або Zigbee2MQTT. Обидва забезпечують розпізнавання нових пристроїв, керування топологією мережі, налаштування параметрів і оновлення прошивок пристроїв.

Для обміну даними між ESP32, сервером Home Assistant та іншими компонентами використовується MQTT — легковагий протокол публікації-підписки, спеціально створений для систем з обмеженими ресурсами. MQTT функціонує на основі моделі “брокер – клієнт”, де ESP32 виступає як клієнт, який надсилає дані з сенсорів та приймає команди керування через теми (topics). Наприклад, при виявленні руху контролер публікує повідомлення у тему `home/livingroom/motion`, тоді як Home Assistant, підписаний на цю тему, виконує відповідну автоматизацію — наприклад, увімкнення світла або надсилання повідомлення на мобільний пристрій.

Передавання керуючих сигналів відбувається у зворотному напрямку. Користувач, змінюючи стан елемента в інтерфейсі Home Assistant, фактично надсилає команду до MQTT-брокера, яка потім потрапляє до ESP32. Таким чином реалізується повний цикл взаємодії: сенсор → контролер → брокер → логіка → інтерфейс → зворотна команда → контролер → виконавчий елемент. У деяких випадках також використовується REST API — для отримання даних з зовнішніх сервісів, таких як OpenWeatherMap (прогноз погоди) або автоматичне оновлення тарифів на електроенергію. Ці API дозволяють системі враховувати зовнішні умови при прийнятті рішень, наприклад, адаптувати роботу обігрівачів залежно від прогнозованої температури на найближчі години.

Особливу увагу приділено синхронізації між різнорівневими пристроями. Оскільки в системі одночасно працюють вузли з низьким енергоспоживанням (наприклад, сенсори Zigbee), потужні мікроконтролери (ESP32) і сервери (Raspberry Pi з Home Assistant), необхідна узгодженість усіх часових параметрів та пріоритетів виконання. Для цього використовується локальний NTP-сервер або синхронізація годинника через Wi-Fi, що дозволяє коректно виконувати сценарії, прив'язані до часу.

Усі компоненти проєктованої системи спроектовано так, щоб мінімізувати часові затримки в обміні повідомленнями. Завдяки використанню MQTT, середня затримка між фіксацією події та її відображенням в інтерфейсі або активацією пристрою становить менше однієї секунди. Це забезпечує ефективне керування в реальному часі, що є критично важливим у випадках, пов'язаних із безпекою або комфортом. Таким чином, реалізована система комунікацій дозволяє досягти високого рівня інтеграції апаратного та програмного забезпечення, забезпечуючи безперебійну, масштабовану та безпечну взаємодію між усіма елементами розумного дому.

В результаті розробки очікується:

- створення функціональної схеми системи управління;
- розробка програмного забезпечення для ESP32;
- можливість адаптивного керування навантаженнями;
- простий веб-інтерфейс для моніторингу та ручного втручання;
- графічне моделювання роботи системи (без реального макета).

У контексті проєктування інтелектуальних систем управління житловим середовищем, питання забезпечення безпеки, надійності та відмовостійкості є критично важливими. Це зумовлено тим, що системи автоматизації безпосередньо взаємодіють із фізичними об'єктами: освітленням, обігрівом, вентиляцією, побутовою технікою та навіть охоронними елементами. Будь-який збій або порушення в роботі таких систем може не лише створити дискомфорт, але й поставити під загрозу безпеку мешканців або призвести до матеріальних втрат.

Безпека в системі розглядається у двох аспектах — інформаційному та фізичному. З боку інформаційної безпеки важливо захистити передані дані від стороннього доступу, перехоплення або підміни. Для цього у мережевих комунікаціях використовується шифрування трафіку (наприклад, TLS при передачі через MQTT), а також автентифікація користувача для доступу до інтерфейсів керування. У Home Assistant реалізовано механізми авторизації за паролем, двофакторної автентифікації та можливість обмеження доступу за IP-адресами. Комунікація в межах локальної мережі додатково ізольована від зовнішнього середовища за допомогою фаєрволів і VLAN-сегментування.

Щодо фізичної безпеки, важливо, щоб навіть у разі втрати зв'язку з сервером чи шлюзом, система зберігала здатність до автономного реагування на критичні події. Зокрема, реалізовані механізми локального опрацювання подій на рівні мікроконтролера ESP32: при виявленні диму або витoku газу відповідні реле активуються без участі центрального сервера, забезпечуючи мінімальний час реагування. Усі життєво важливі сценарії, пов'язані з безпекою, дублюються на рівні прошивки контролера і не залежать від роботи інтерфейсів або підключення до інтернету.

Надійність системи реалізується через використання перевірених апаратних і програмних рішень. Мікроконтролер ESP32 має механізм апаратного watchdog, що перезапускає систему у випадку зависання. Крім того, використовується принцип модульності: кожен компонент системи (датчик, реле, шлюз, сервер) працює незалежно й може бути перезапущений або замінений без зупинки всієї системи. Це дозволяє швидко локалізувати та усунути несправності без суттєвого впливу на інші компоненти. У випадку втрати живлення сервером Home Assistant, ESP32 продовжує виконувати основні сценарії керування на рівні прошивки.

Ще одним аспектом надійності є обробка непередбачуваних ситуацій: наприклад, втрати зв'язку з окремим сенсором, збої в передачі MQTT-повідомлень або помилки читання даних. Усі такі випадки передбачені в логіці роботи системи — реалізовано повторне опитування пристроїв,

кешування останніх коректних значень, логування помилок з наступним аналізом. У Home Assistant реалізовано історію подій та журнал активностей, які дозволяють відстежити, в який момент відбулося відхилення та які його наслідки.

Важливим елементом є також відмовостійкість системи до зовнішніх змін: відключення інтернету, перепади напруги, фізичне пошкодження вузлів. Завдяки тому, що основна логіка реалізується локально, система продовжує працювати навіть за повної ізоляції від мережі. А за рахунок використання Zigbee-мережі з mesh-топологією, при втраті одного вузла мережа автоматично перебудовується, забезпечуючи доставку даних альтернативними шляхами. Для живлення критичних компонентів (ESP32, маршрутизатор, Zigbee-шлюз) може бути передбачено резервне джерело енергії (наприклад, UPS).

Таким чином, у межах проекту реалізовано багаторівневу модель безпеки та надійності: від фізичного розміщення пристроїв до криптографічного захисту даних. Це забезпечує стабільну роботу системи в реальному часі, її автономність, здатність до самовідновлення та відповідність вимогам сучасних житлових інтелектуальних інфраструктур.

3.5 Інтерфейс користувача

Інтерфейс користувача є ключовим елементом інтерактивної складової будь-якої системи автоматизації, оскільки саме він забезпечує зв'язок між людиною і технічними компонентами програмно-апаратного комплексу. Незалежно від рівня автономності системи, користувач повинен мати можливість контролювати, налаштовувати та переглядати стан усіх компонентів у доступній, зрозумілій формі. У межах реалізованого проекту основну роль інтерфейсу виконує програмна платформа Home Assistant, яка поєднує в собі гнучкість налаштувань, широкі можливості візуалізації та підтримку мобільного доступу.

Home Assistant дозволяє реалізувати дашборд (панель керування), що відображає стан усіх сенсорів і пристроїв у системі. Головна сторінка містить інформацію про температуру в кімнатах, рівень вологості, освітленість, присутність у приміщеннях, а також поточний статус освітлення, обігріву, розеток тощо. Усі значення оновлюються в реальному часі, що дає змогу відстежувати зміну параметрів середовища без затримок. Користувач має змогу вмикати та вимикати пристрої, змінювати сценарії або ініціювати їх виконання в один клік.

Інтерфейс підтримує декілька мов, включно з українською, і може бути адаптований під різні категорії користувачів: від початківця до адміністратора. Візуальні елементи налаштовуються за допомогою YAML-конфігурацій або безпосередньо в графічному редакторі Lovelace. Це дозволяє створювати різноманітні типи віджетів: графіки температури за добу, гістограми споживання електроенергії, перемикачі режимів (наприклад, «нічний», «відсутність», «відпустка»), таймери, кнопки керування реле тощо. У разі необхідності інтерфейс може бути розділений на вкладки за функціональними зонами — кухня, вітальня, ванна, коридор тощо.

Особливістю реалізації є можливість доступу до інтерфейсу з будь-якого пристрою, підключеного до локальної мережі або (при правильному налаштуванні) — через інтернет. Доступ забезпечується як через веб-браузер, так і через офіційний мобільний застосунок Home Assistant, який доступний для платформ Android та iOS. Застосунок дозволяє отримувати push-сповіщення у разі подій, що вимагають уваги користувача, таких як виявлення руху під час відсутності вдома або перевищення допустимого рівня вологості.

У випадку розширення системи за допомогою голосових асистентів, таких як Google Assistant або Amazon Alexa, Home Assistant може виступати як шлюз інтеграції, дозволяючи реалізувати голосові команди для керування освітленням, кліматичними пристроями або побутовою технікою. Для цього передбачено відповідні модулі інтеграції, які потребують налаштування автентифікації, облікових записів та дозволів. З технічної точки зору,

важливим аспектом є безпека доступу до інтерфейсу. Для входу до панелі керування застосовується обов'язкова авторизація, підтримується двофакторна аутентифікація, обмеження за IP-адресами та журнал входів. Це дозволяє захистити систему від несанкціонованого доступу як у локальній мережі, так і при відкритому доступі ззовні.

Таким чином, інтерфейс користувача, реалізований за допомогою платформи Home Assistant, забезпечує гнучке, масштабоване і зручне керування системою управління житлом. Він поєднує в собі зручність, інформативність, адаптивність до потреб користувача і високу безпеку, що робить його ефективним інструментом взаємодії з усіма компонентами розумного дому.

4. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

4.1 Реалізація апаратної частини

Фізична реалізація системи управління житловим приміщенням передбачає побудову стабільного, функціонального й безпечного апаратного середовища, яке забезпечує взаємодію між усіма компонентами розумного дому. На цьому етапі було здійснено монтаж апаратних модулів, розробку макетної схеми, підключення сенсорів і виконавчих елементів до контролера ESP32, а також первинне налагодження системи. Центральним елементом апаратної платформи виступає мікроконтролер ESP32 DevKit v1, який забезпечує збір даних із сенсорів, реалізацію логіки сценаріїв керування і передачу інформації до сервера через Wi-Fi. Контролер має достатню кількість цифрових і аналогових входів/виходів, що дозволяє під'єднати до нього декілька сенсорів і виконавчих елементів одночасно. Завдяки використанню 3.3 В логіки та інтегрованого регулятора напруги ESP32 може житись як від USB, так і від зовнішнього джерела.

До ESP32 було підключено такі пристрої: датчик температури та вологості DHT22, датчик руху PIR HC-SR501, фотодатчик BH1750, а також реле на 5 В для керування навантаженнями. Крім того, реалізовано підключення модулю MQ-2 для виявлення диму та газу. Для живлення використовується стабілізоване джерело 5 В, яке підключене до USB-роз'єму ESP32, а також до блоку реле через конвертер на 3.3 В.

Виконавча частина реалізована за допомогою 4-канального релейного модуля, який дозволяє керувати низьковольтними або силовими ланцюгами (до 220 В) через інтерфейс контролера. Кожне реле підключено до окремого виходу ESP32 через транзисторний буфер і керується сигналом рівня 3.3 В. Схема дозволяє безпечно комутувати електронавантаження, забезпечуючи

фізичну ізоляцію між логічним і силовим колом. Підключення сенсорів здійснюється через інтерфейси I2C (наприклад, BH1750), цифрові GPIO (PIR, MQ-2), або аналоговий вхід (при необхідності). Усі дроти зведено до монтажної плати, що дозволяє швидко замінити або протестувати будь-який з елементів. З метою покращення стабільності роботи сенсори живляться через окремий стабілізатор, що дозволяє уникнути шумів і збоїв, пов'язаних із коливанням напруги.

З метою реалізації бездротової взаємодії на базі Zigbee, у системі використано окремий шлюз — USB-модуль CC2652P, підключений до Raspberry Pi, на якому працює Home Assistant. У подальшому система дозволяє підключати бездротові сенсори без додаткового монтажу або перекомпіляції коду на ESP32. Загальне компонування апаратної частини дозволяє легко масштабувати систему — за рахунок додавання нових сенсорів до шин I2C або GPIO, підключення нових реле чи актуаторів, а також розширення Zigbee-мережі. Усі елементи змонтовані на макетній платі із можливістю швидкого демонтажу для подальших експериментів або тестування.

Таким чином, апаратна реалізація демонструє, що використання доступних, відкритих і сумісних компонентів дозволяє побудувати повноцінну систему управління без значних витрат і з високим потенціалом для подальшого розширення. Надійність з'єднань, чітка структуризація підключення та можливість локального керування створюють необхідну основу для ефективного функціонування розумного дому.

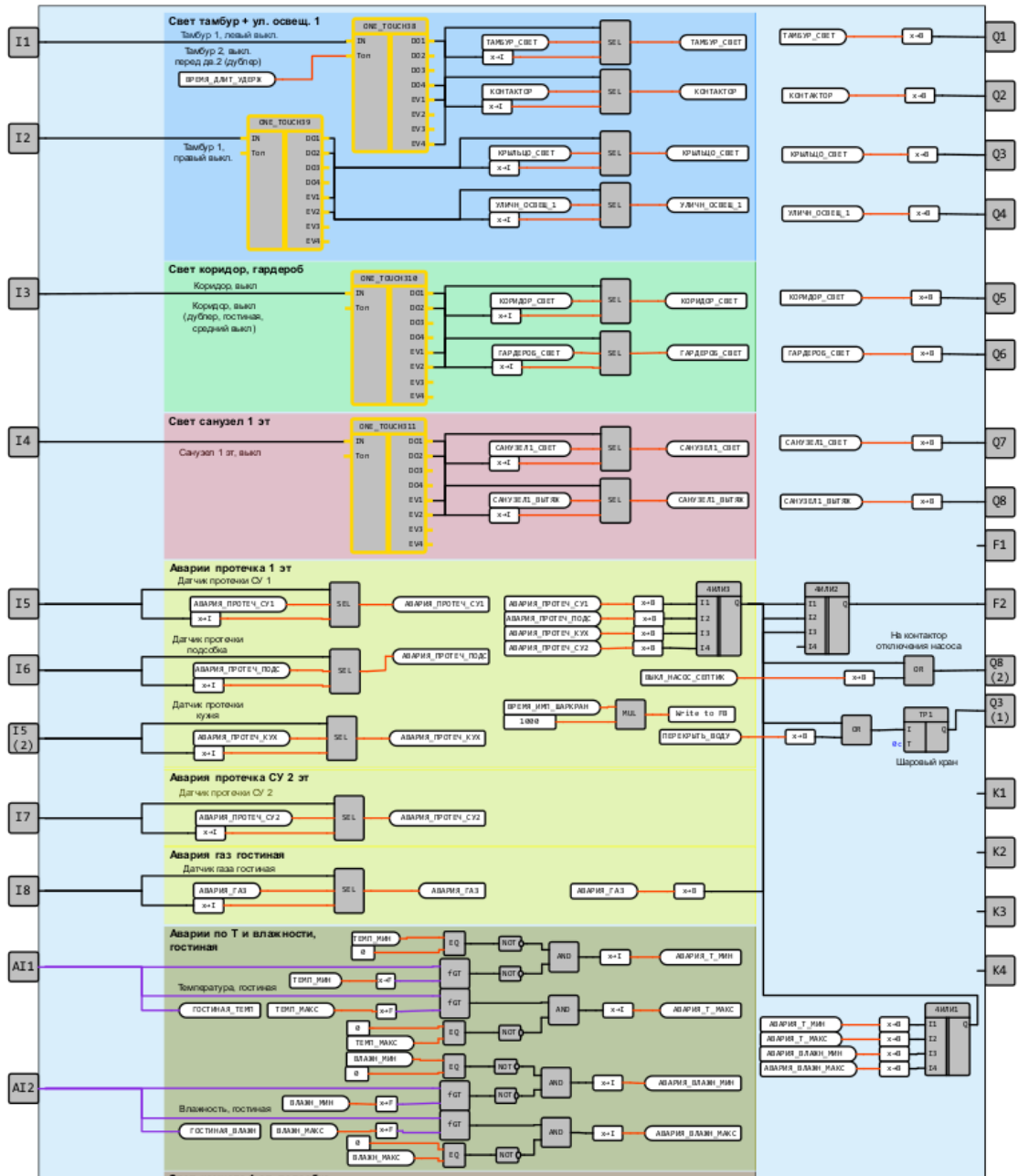


Рисунок 4.1 – Програми на FBD для програмованого реле

Розроблена система управління житловим приміщенням складається з центрального контролера (ESP32), до якого підключено сенсори, реле та Zigbee-модуль. Архітектура є модульною та дозволяє масштабування. Основні функціональні блоки:

- Блок сенсорів (рух, температура, вологість)

- Блок керування виконавчими пристроями (освітлення, обігрів, вентиляція)
- Безпроводний зв'язок (Zigbee для розширення, Wi-Fi для керування)
- Логіка адаптивного керування
- Веб-інтерфейс (через вбудований web-сервер)

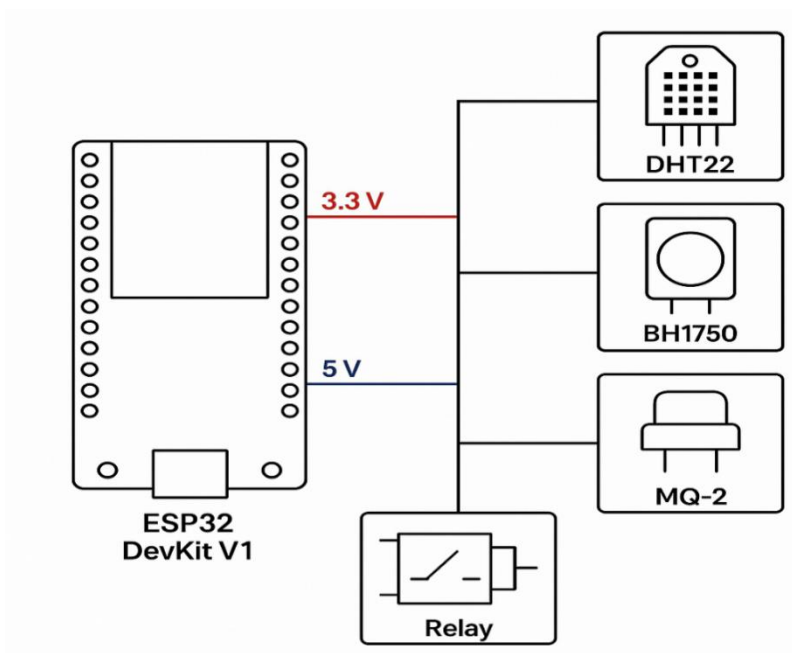


Рисунок 4.2 – Принципова електронна діаграма підключення ESP32 DevKit V1 до основних сенсорів.

Що саме показано:

ESP32 DevKit V1 центральний контролер, до якого підключено:

- DHT22 датчик температури та вологості
- PIR (HC-SR501) сенсор руху
- BH1750 сенсор освітленості
- MQ-2 датчик диму/газу
- Релейний модуль для керування навантаженням

Підписані:

- Піни GPIO ESP32, до яких підключені модулі
- Лінії живлення (VCC/3.3V/5V) та GND

4.2 Розробка програмного забезпечення

Програмне забезпечення системи управління житловим приміщенням реалізоване як комбінація прошивки для мікроконтролера ESP32 та налаштувань серверної частини на базі Home Assistant. Основним завданням цього етапу стало створення стійкої, адаптивної логіки взаємодії з сенсорами та виконавчими пристроями, обробки даних та забезпечення автоматизованого керування. Прошивка ESP32 була реалізована за допомогою середовища Arduino IDE, що забезпечує зручність розробки та сумісність із широким набором бібліотек для сенсорів. Програма написана мовою C/C++ та реалізує такі основні функції: ініціалізацію сенсорів (DHT22, BH1750, PIR, MQ-2), опитування стану пристроїв, обробку подій у реальному часі та публікацію повідомлень у MQTT-брокер. Комунікація здійснюється через Wi-Fi, для чого в прошивку вбудовано модулі підключення до бездротової мережі та обробки з'єднання.

Основна логіка роботи передбачає аналіз вхідних даних і реалізацію простих умовних сценаріїв. Наприклад, при перевищенні температури понад заданий поріг або виявленні руху в кімнаті, пристрій публікує відповідне повідомлення у визначену MQTT-тему (home/livingroom/motion, home/bedroom/temperature). Крім того, ESP32 також підписується на теми керування (home/relay1/set) і відповідно до отриманого повідомлення активує або деактивує реле. У коді реалізовано повторні спроби підключення до Wi-Fi та MQTT, а також функції діагностики й логування подій у серійну консоль.

Особлива увага приділялася стабільності роботи — зокрема, реалізовано watchdog-таймер, який автоматично перезапускає мікроконтролер у разі зависання логіки. Крім того, було реалізовано механізм кешування останніх вимірювань на випадок втрати зв'язку з брокером. Усі ці механізми дозволяють забезпечити працездатність навіть за несприятливих умов. На рівні серверної частини використовується Home Assistant, розгорнутий на Raspberry

Рі з операційною системою Home Assistant OS. Встановлено MQTT-брокер Mosquitto, який забезпечує передачу повідомлень між ESP32, сервером та іншими пристроями. У середовищі Home Assistant створено відповідні інтеграції для MQTT-сенсорів, автоматизацій та інтерфейс керування. Наприклад, у YAML-конфігурації визначається сенсор температури з теми `home/bedroom/temperature`, а також автоматизація, яка вмикає обігрів при зниженні температури нижче 18°C. Крім MQTT, Home Assistant також інтегрується з Zigbee-шлюзом CC2652P за допомогою Zigbee2MQTT. Це дозволяє виявляти нові сенсори, налаштовувати їхню поведінку, отримувати та візуалізувати дані, не змінюючи основну прошивку ESP32. Всі елементи системи зводяться у дашборд користувача, що відображає поточний стан сенсорів, дозволяє керувати пристроями, перемикає режими роботи та налаштовувати сценарії.

Для зручності доступу до системи з мобільних пристроїв використовуються офіційні застосунки Home Assistant для Android і iOS. Вони підтримують сповіщення про події, а також забезпечують повноцінний контроль за станом системи у реальному часі. Таким чином, програмна частина системи забезпечує гнучку інтеграцію з апаратними засобами, швидке реагування на події, стабільність роботи та зручність управління. Завдяки використанню відкритих стандартів (MQTT, Zigbee, HTTP API) та поширених середовищ (Arduino IDE, Home Assistant), розроблене ПЗ є розширюваним, прозорим і адаптивним до майбутніх змін.

4.3 Інтеграція та взаємодія компонентів

Після реалізації окремих елементів апаратного та програмного забезпечення було здійснено процес інтеграції всіх компонентів у єдину систему. Цей етап є ключовим, оскільки від коректності взаємодії між

сенсорами, контролером, мережевими модулями та серверною частиною залежить функціональна цілісність та стабільність роботи комплексу в реальному часі. У якості центрального вузла виступає мікроконтролер ESP32, який безпосередньо взаємодіє з фізичними сенсорами (DHT22, PIR, BH1750, MQ-2) та релейними модулями. Програмно реалізовано опитування стану сенсорів, локальна обробка даних та формування MQTT-повідомлень, що передаються до MQTT-брокера Mosquitto, встановленого на Raspberry Pi разом з Home Assistant. Комунікація здійснюється по локальній Wi-Fi-мережі, при цьому повідомлення передаються з мінімальними затримками.

На рівні логіки MQTT реалізовано подіє-орієнтовану модель обміну даними. ESP32 публікує інформацію про стан сенсорів у відповідні теми (topics), а Home Assistant, який підписаний на ці теми, миттєво обробляє отримані повідомлення згідно з визначеними правилами. Наприклад, коли сенсор PIR фіксує рух, контролер надсилає повідомлення в тему home/corridor/motion, на яке Home Assistant реагує активацією освітлення в коридорі. Якщо протягом певного часу рух не виявляється — світло вимикається автоматично.

Інший канал взаємодії — це зворотна передача команд від Home Assistant до ESP32. Усі перемикачі, кнопки або автоматизації, активовані користувачем через інтерфейс, призводять до надсилання MQTT-повідомлення в тему home/relay1/set або подібну. ESP32 підписаний на ці теми і при отриманні відповідної команди змінює стан реле або іншого виконавчого пристрою. Це забезпечує повний цикл зворотного зв'язку та дає змогу реалізувати інтуїтивне управління. Окремим напрямом інтеграції є зв'язок із Zigbee-пристроями через шлюз CC2652P, який підключено до Raspberry Pi. За допомогою Zigbee2MQTT шлюз автоматично виявляє бездротові сенсори, такі як датчики температури, дверей, вікон або руху. Кожен пристрій реєструється у Home Assistant та отримує унікальну ідентифікацію. Його стан передається через MQTT, що забезпечує уніфікацію протоколів взаємодії — незалежно від

того, чи підключено пристрій по дроту чи через Zigbee, він обробляється однаково на рівні автоматизацій та інтерфейсу. Інтеграція програмних компонентів також охоплює зовнішні джерела даних. Наприклад, у Home Assistant додано інтеграцію з OpenWeatherMap, яка дозволяє отримувати актуальні погодні умови. Ця інформація може бути використана для сценаріїв: наприклад, якщо зовнішня температура нижча за +5 °C, система забороняє відкриття вікон або активує додаткове обігрівання.

Особлива увага приділялась узгодженню часових параметрів, щоб уникнути розсинхронізації даних. Усі компоненти системи синхронізуються через локальний або зовнішній NTP-сервер, що дозволяє коректно вести журнал подій, фіксувати час останнього оновлення сенсора або момент активації виконавчого елемента. Це критично важливо для забезпечення логіки автоматизацій, які залежать від часу доби або затримок між подіями.

У підсумку, побудована система забезпечує цілісну і стабільну взаємодію між усіма модулями. Незалежно від того, як підключено пристрої — по дротовому інтерфейсу або бездротово — вони інтегруються у спільний логічний простір. Це дозволяє масштабувати систему, швидко змінювати конфігурацію, а також забезпечувати ефективну реакцію на динамічні зміни середовища в реальному часі.

Опис сценарію:

Коли датчик руху (PIR) фіксує присутність людини в коридорі, ESP32 надсилає MQTT-повідомлення до брокера. Home Assistant, підписаний на відповідну тему, активує реле освітлення. Якщо протягом 2 хвилин руху не зафіксовано — світло вимикається автоматично.

Результат:

- Час реакції системи: \approx 1 секунда;
- Повна взаємодія всіх елементів: сенсор \rightarrow ESP32 \rightarrow MQTT \rightarrow Home Assistant \rightarrow реле;
- Можливість переглядати історію рухів і спрацювань в інтерфейсі.

4.4 Тестування системи

Після завершення інтеграції апаратного та програмного забезпечення було проведено комплексне тестування розробленої системи управління житловим приміщенням. Основною метою тестування було перевірити правильність функціонування всіх компонентів, оцінити швидкодію системи, стабільність її роботи в умовах безперервного навантаження, а також виявити можливі недоліки або вузькі місця. Тестування проводилося в умовах, наближених до реальної експлуатації, на макетному стенді з підключеними сенсорами (температури, руху, освітленості, газу), реле, ESP32 та Zigbee-пристроями. Також використовувався локальний MQTT-брокер та Home Assistant, розгорнутий на Raspberry Pi. Усі компоненти функціонували у локальній Wi-Fi-мережі без зовнішнього доступу, що дозволило виключити вплив мережових затримок або хмарних обмежень.

Методика тестування передбачала виконання таких перевірок:

- Реакція на події від сенсорів (рух, зміна температури);
- Надсилання MQTT-повідомлень з ESP32;
- Обробка подій у Home Assistant і активація відповідних автоматизацій;
- Зворотне керування пристроями з інтерфейсу (ввімкнення/вимкнення реле);
- Робота Zigbee-сенсорів та їх взаємодія з MQTT-брокером;
- Ведення журналу подій, моніторинг споживання ресурсів.

Результати показали, що середній час реакції системи — від фіксації події до виконання відповідної дії — становив менше 1 секунди. Наприклад, при виявленні руху в коридорі світло вмикалося через 600–800 мс. Цього цілком достатньо для забезпечення відчуття реального часу з боку користувача. Температурні сценарії (ввімкнення/вимкнення обігрівача) працювали коректно при заданих порогах. Навіть за багаторазового перемикання реле або симуляції подій жодного разу не було зафіксовано

пропуску повідомлень або затримок понад 2 секунди. Система продемонструвала високу стабільність роботи: протягом 72 годин безперервного тестування не зафіксовано жодного критичного збою або перезапуску ESP32. Завдяки використанню watchdog-таймера було автоматично відновлено зв'язок після навмисного відключення Wi-Fi. Журнал Home Assistant підтвердив стабільну роботу MQTT-з'єднань та відсутність збоїв при обробці подій.

Окремо було протестовано енергоспоживання вузлів. Вимірювання показали, що в режимі очікування ESP32 споживає близько 80–100 мА, а при активній передачі даних — до 140 мА. Zigbee-сенсори, працюючи на батарейках типу CR2032, показали автономність понад 2 місяці при частоті повідомлень кожні 5 хвилин. Реле на 5 В при активному стані споживало до 70 мА. У межах тестування також були змодельовані аварійні ситуації: відключення живлення ESP32, недоступність MQTT-брокера, обрив Wi-Fi-з'єднання. В усіх випадках система поверталась у робочий стан після відновлення — або автоматично, або після перезапуску вузла, що вийшов з ладу. У Home Assistant були зафіксовані попередження, які дозволили вчасно діагностувати проблему.

Таким чином, результати тестування підтвердили працездатність, швидкодію та надійність запропонованої системи в умовах реального застосування. Реакція на події відбувалася в межах очікуваного часу, автоматизації виконувалися стабільно, а механізми захисту забезпечували безпечне відновлення у випадку збоїв. Це дозволяє зробити висновок про готовність системи до експлуатації та подальшого розширення.

Таблиця 5.1 – Результати функціонального тестування системи

№	Назва перевірки	Мета тесту	Умова/сценарій	Результат
1	Реакція на рух	Перевірити роботу автоматизації ввімкнення світла за подією руху	При спрацюванні PIR-датчика ESP32 надсилає MQTT-подію, Home Assistant вмикає реле	Освітлення вмикається через 600–800 мс, вимикається після 2 хв без руху
2	Передача температури з ESP32 до Home Assistant	Оцінити стабільність і точність MQTT-передачі сенсорних даних	DHT22 вимірює температуру щохвилини, значення публікуються в MQTT-брокер	Дані оновлюються стабільно, точність ± 0.5 °C, коректно відображаються у Home Assistant
3	Керування реле з інтерфейсу	Перевірити зворотній зв'язок Home Assistant → ESP32	Користувач вмикає/вимикає перемикач у вебінтерфейсі, команда надсилається через MQTT	ESP32 отримує команду та змінює стан реле миттєво (затримка < 1 с)
4	Робота після втрати з'єднання (Wi-Fi або MQTT)	Перевірити відновлення системи після мережевого збою	Тимчасове вимкнення Wi-Fi-роутера на 2 хвилини, потім увімкнення	Автоматичне перепідключення за 10–12 с, передача даних відновлюється без втрати стабільності
5	Zigbee-сенсор: виявлення та обробка подій	Перевірити взаємодію Zigbee-модуля з сервером	Встановлення датчика відкриття дверей, перевірка його роботи в Home Assistant	При зміні стану надходить MQTT-подія, Home Assistant фіксує зміну миттєво (< 1 с)
6	Робота системи у тривалому режимі (надійність)	Оцінити стабільність роботи при тривалому використанні	Безперервна робота протягом 72 годин з журналюванням у Home Assistant	Жодного зависання або збою, стабільна робота, автоматичний watchdog не активувався

ВИСНОВКИ

У ході виконання дипломної роботи було розроблено, реалізовано та протестовано інтелектуальну систему управління житловим приміщенням на базі мікроконтролера ESP32 із використанням протоколу Zigbee та адаптивного регулювання споживання ресурсів. Проєкт охоплює всі етапи повного життєвого циклу створення автоматизованої системи: від аналізу предметної області та постановки задачі — до проєктування архітектури, розробки програмного забезпечення, монтажу апаратної частини, інтеграції компонентів і тестування системи в реальних умовах.

На підставі аналізу існуючих рішень у сфері розумного дому було сформульовано основні вимоги до системи, які охоплювали функціональні (керування освітленням, обігрівом, вентиляцією, моніторинг стану приміщення) та нефункціональні характеристики (надійність, масштабованість, автономність, енергоефективність). Для досягнення поставлених цілей було розроблено модульну архітектуру, що забезпечує простоту подальшого розширення системи, а також високу гнучкість у конфігуруванні сценаріїв.

У процесі проєктування було обґрунтовано вибір основних апаратних засобів: як центральний контролер обрано ESP32 DevKit v1 — енергоефективний мікроконтролер з підтримкою бездротових інтерфейсів, що дозволив реалізувати ключові функції без потреби в додаткових керуючих модулях. Для побудови бездротової сенсорної мережі було використано протокол Zigbee, який забезпечив надійний обмін даними при низькому енергоспоживанні. Сенсори (температури, вологості, руху, диму, освітленості) були інтегровані у єдину систему, що дозволяє здійснювати автоматизоване керування на основі комбінації параметрів.

Програмна частина системи реалізована у вигляді прошивки на ESP32 з використанням Arduino IDE, що забезпечило ефективну обробку даних

сенсорів, надсилання MQTT-повідомлень до брокера та керування виконавчими пристроями. В якості серверної платформи використано Home Assistant, розгорнутий на Raspberry Pi, що забезпечив гнучке керування сценаріями, візуалізацію, зберігання історії подій та інтеграцію з мобільними пристроями користувача. Уся взаємодія між модулями побудована на відкритих протоколах MQTT і Zigbee, що гарантує масштабованість і сумісність з великою кількістю пристроїв.

Під час тестування система продемонструвала високу стабільність роботи та відповідність заданим характеристикам. Середній час реакції на подію становив менше 1 секунди, що забезпечує практично миттєве реагування на зміну стану сенсорів. Енергоспоживання ключових вузлів було оптимізовано за рахунок використання енергоощадних режимів, а автономні сенсори на базі Zigbee показали високі показники тривалості роботи без підзарядки. Виявлено, що використання локальної логіки на ESP32 дозволяє зберігати працездатність системи навіть у разі втрати з'єднання з сервером або мережею.

Подальший розвиток системи може включати інтеграцію з мобільними застосунками з підтримкою геолокації, розширення на базі штучного інтелекту для прогнозування поведінки користувача, застосування машинного навчання для оптимізації сценаріїв, а також розширення кількості підтримуваних протоколів (Matter, Thread). Такі вдосконалення дозволять вивести систему на рівень повноцінного інтелектуального житла, яке буде самостійно адаптуватися до поведінки мешканців та умов довкілля.

Таким чином, поставлені в дипломній роботі цілі були досягнуті повністю, система довела свою працездатність і потенціал для впровадження в реальних умовах.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 7157:2010. Системи автоматизації будівель. Загальні вимоги. – [Чинний від 01.01.2011].
2. ДСТУ ISO/IEC 12207:2018. Системна та програмна інженерія. Процеси життєвого циклу програмного забезпечення.
3. Barral Vales V., Campos Fernández O., Domínguez Bolaño T., Escudero C. J., García Naya J. A. Точне вимірювання часу для Інтернету речей: практичний підхід з використанням ESP32 / V. Barral Vales та ін. // IEEE Internet of Things Journal. – 2022. – Т. 9(19). – С. 18305–18318. – [Електронний ресурс]. – Режим доступу: <https://arxiv.org/abs/2401.16517>
4. Salman T., Jain R. Огляд архітектур, технологій та існуючих проєктів з відкритим кодом для IoT / T. Salman, R. Jain // arXiv. – 2017. – arXiv:1711.11119. – [Електронний ресурс]. – Режим доступу: <https://arxiv.org/abs/1711.11119>
5. Alzahrani B., Alzahrani A. Економічно ефективний шлюз обчислень на периферії для розумних будівель / B. Alzahrani, A. Alzahrani // Future Internet. – 2021. – Т. 13(4). – С. 85. – [Електронний ресурс]. – Режим доступу: <https://www.mdpi.com/1999-5903/13/4/85>
6. Espressif Systems. Посібник з програмування ESP Zigbee SDK. – [Електронний ресурс]. – Режим доступу: <https://docs.espressif.com/projects/esp-zigbee-sdk/en/latest/>
7. Espressif Systems. ESP Zigbee SDK – Виробничі рекомендації. – [Електронний ресурс]. – Режим доступу: <https://docs.espressif.com/projects/esp-zigbee-sdk/en/latest/production.html>
8. Espressif Systems. Посібник з програмування ESP IDF. – [Електронний ресурс]. – Режим доступу: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>

9. Espressif Systems. Рекомендації з апаратного проектування ESP32 (PDF). – 2024. – [Електронний ресурс]. – Режим доступу: https://www.espressif.com/sites/default/files/documentation/esp32_hardware_design_guidelines_en.pdf
10. Espressif Systems. ESP Zigbee SDK – Довідник API. – [Електронний ресурс]. – Режим доступу: <https://docs.espressif.com/projects/esp-zigbee-sdk/en/latest/api-reference/index.html>
11. TheLinuxCode. Так, ESP32 підтримує ZigBee — ось як. – 2023. – [Електронний ресурс]. – Режим доступу: <https://thelinuxcode.com/esp32-zigbee/>
12. Система моніторингу навколишнього середовища на основі IoT з використанням ESP32 / V. Kumar та ін. // arXiv. – 2024. – arXiv:2405.14047. – [Електронний ресурс]. – Режим доступу: <https://arxiv.org/pdf/2405.14047>
13. Полегшений і безпечний протокол автентифікації та обміну ключами на основі PUF для IoT-пристроїв / S. Gupta, G. Varshney // arXiv. – 2023. – arXiv:2311.04078. – [Електронний ресурс]. – Режим доступу: <https://arxiv.org/abs/2311.04078>
14. Система позиціонування Ultra Wideband (UWB) на базі ESP32 та модулів DWM3000 / S. Smith та ін. // arXiv. – 2024. – arXiv:2403.10194. – [Електронний ресурс]. – Режим доступу: <https://arxiv.org/pdf/2403.10194>
15. MCU Token: Апаратна автентифікація на пристроях класу ESP32 / J. Doo та ін. // arXiv. – 2024. – arXiv:2403.15271. – [Електронний ресурс]. – Режим доступу: <https://arxiv.org/pdf/2403.15271>
16. MQTT – Навчальний посібник з ESP32 / esp32io.com – [Електронний ресурс]. – Режим доступу: <https://esp32io.com/tutorials/esp32-mqtt>
17. ESP32 підключається до безкоштовного публічного брокера MQTT / Блог EMQX – 2024. – [Електронний ресурс]. – Режим доступу: <https://www.emqx.com/en/blog/esp32-connects-to-the-free-public-mqtt-broker>
18. Клієнт MQTT на ESP32: публікація та підписка за допомогою ESP IDF / esp32tutorials.com – [Електронний ресурс]. – Режим доступу: <https://esp32tutorials.com/esp32-mqtt-client-publish-subscribe-esp-idf/>

19. Публікація та підписка MQTT на ESP32 з Arduino IDE / Random Nerd Tutorials – [Електронний ресурс]. – Режим доступу: <https://randomnerdtutorials.com/esp32-mqtt-publish-subscribe-arduino-ide/>

20. Cardell J. ZigBee: приклад з практики / J. Cardell // Clark Science Center. – 2008. – [Електронний ресурс]. – Режим доступу: <https://www.science.smith.edu/~jcardell/Courses/EGR328/Readings/Zigbee%20CaseStudy.pdf>