

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему

«Розробка комплексної системи розумного будинку на базі Arduino»
здобувача Пісного Артема Валентиновича

(ПІБ)

(підпис)

Науковий керівник:

д.е.н., професор

_____ Лондар Сергій Леонідович

**Робота допущена до захисту перед
екзаменаційною комісією з атестації
здобувачів вищої освіти**

завідувач кафедри:

к.е.н., доцент

_____ Тішков Б.О.

Київ 2025

Міністерство освіти і науки України
Київський національний економічний університет імені Вадима Гетьмана
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проєктної групи(гарант)
освітньо-професійної програми

_____ Помазун О.М.

“ ____ ” _____ 2025 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри

_____ Тішков Б.О.

“ ____ ” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти **Пісного Артема Валентиновича**

очної (денної) форми навчання

на підготовку кваліфікаційної бакалаврської роботи

на тему: «Розробка комплексної системи розумного будинку на базі Arduino»

Тему затверджено наказом ректора Університету від « 7 » березня 2025 р. № 466-
ст.

Кваліфікаційна бакалаврська робота виконується на матеріалах

навчальних посібників та матеріалів з мережі Інтернет.

План кваліфікаційної бакалаврської роботи

Розділ I. Характеристика та аналіз предметної галузі

Розділ II. Розробка вимог і моделювання інформаційної системи

Розділ III. Проєктування та реалізація компонентів системи

Об'єктом дослідження є процеси створення вбудованих та веб-орієнтованих інформаційних систем для автоматизації керування підсистемами житлових приміщень.

Предметом дослідження є методи, інструменти та підходи до проектування та реалізації інтелектуальних систем моніторингу та керування, які забезпечують інтеграцію апаратних компонентів з програмною логікою (Node-RED, Arduino) для реалізації концепції «розумного будинку».

Мета кваліфікаційної бакалаврської роботи — розробка інтелектуальної автоматизованої системи «Розумний будинок», яка забезпечує моніторинг середовища, автоматичне та ручне керування пристроями, а також віддалену взаємодію з користувачем через веб-інтерфейс і мобільні сервіси.

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі I навести характеристику предметної галузі й об'єкта дослідження, проаналізувати сучасний стан та тенденції розвитку систем розумного будинку, охарактеризувати основні напрямки автоматизації житлових приміщень. Визначити ключові технології, необхідні для реалізації інтелектуальної системи керування: сенсорні мережі, мікроконтролери, бездротові протоколи зв'язку, системи моніторингу та віддаленого керування. Проаналізувати літературні джерела, стандарти галузі, існуючі апаратні рішення та практичний досвід реалізації подібних проєктів.

У розділі II визначити основні вимоги до системи розумного будинку та обґрунтувати методи її проектування. При моделюванні інформаційної системи застосувати сучасні засоби автоматизованого проектування. Сформулювати постановку задачі, описати алгоритм роботи та побудувати інформаційну модель, що охоплює всі необхідні функції з урахуванням системного підходу.

У розділі III виконати проектування та реалізацію системи розумного будинку: інформаційного, технічного та програмного забезпечення. Оцінити

доцільність впровадження розробленої системи у практичній діяльності користувачів та її адаптацію до реальних умов експлуатації.

Завдання підготував

науковий керівник

Лондар Сергій Леонідович

« 10 » березня 2025 р.

Завдання одержав

здобувач

Пісний Артем Валентинович

« 10 » березня 2025 р..

ВІДГУК

на кваліфікаційну бакалаврську роботу здобувача навчально-наукового інституту «Інститут інформаційних технологій в економіці» освітньо-професійної програми «Комп'ютерні науки» **Пісного Артема Валентиновича** на тему **«Розробка комплексної системи розумного будинку на базі Arduino»**

Актуальність дипломної роботи зумовлена зростаючим попитом на доступні пересічному споживачеві системи «розумний будинок» в Україні а також відповідним активним розвитком технологій автоматизації житла. Власники житла прагнуть до впровадження дешевих і надійних інтелектуальних систем керування в умовах підвищення вимог до енергоефективності, безпеки та комфорту. Такі технології забезпечують можливість централізовано контролювати освітлення, будинковий клімат, безпеку та інші параметри, що підвищує якість життя та знижує комунальні витрати.

Автором проведено ґрунтовне теоретичне дослідження та вивчено існуючі підходи до створення автоматизованих систем «розумного будинку». Наявні рішення автоматизації, зокрема від Google Nest, Home Assistant, проаналізовано на наявність сильних та слабких сторін, і на цій основі сформульовано вимоги до розробки власної системи. Дипломантом систематизовано функціональні та нефункціональні вимоги, виконано проєктування з використанням UML-діаграм, розроблено електронні схеми та алгоритми роботи основних модулів та здійснено успішне тестування створеної системи.

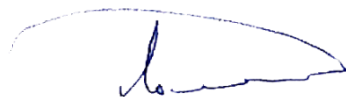
Робота виконана самостійно. У тексті чітко розмежовано власний виклад автора та запозичення з належними посиланнями на використані джерела. Автором комплексно проаналізовано існуючі рішення у сфері автоматизації житлових приміщень, спроєктовано власну систему «розумного будинку» із проєктуванням апаратної та програмної складової системи та створено її робочий прототип.

Найбільш важливими теоретичними висновками та практичними рекомендаціями є наступні. Автором здійснено систематизацію та узагальнення відомостей щодо проєктування і розробку прототипу системи «розумного будинку» на основі зарубіжних та вітчизняних джерел. Детально описано функціонал автоматизованого керування освітленням, кліматом, безпекою, процес моніторингу параметрів середовища за допомогою сенсорів. Вдало реалізовано взаємодію між апаратною частиною на базі Arduino та програмною платформою Node-RED, що дає змогу віддалено контролювати основні життєзабезпечуючі системи будинку.

Як і при виконанні будь якого проекту є й певні недоліки. Зокрема автором не надано розрахунків енергоспоживання системи, що має значення для підтвердження конкурентоспроможності на ринку проєктів автоматизації житла. Варто було б також додатково розглянути питання масштабованості системи. Проте ці зауваження не зменшують теоретичної та практичної цінності роботи.

Вважаю, що робота може бути допущена до захисту із попередньою оцінкою 90 балів.

**Науковий керівник,
д.е.н., професор**



Сергій ЛОНДАР

“11” червня 2025р.

АНОТАЦІЯ

кваліфікаційної бакалаврської роботи
здобувача першого (бакалаврського) рівня вищої освіти, 4 курсу,
виконаної на тему: «Розробка комплексної системи розумного будинку на базі
Arduino»

Київ: кафедра інформаційних систем в економіці, 2025 р.

У роботі детально розглянуто процес розробки інформаційної системи для реалізації концепції «розумного будинку». Проведено комплексний аналіз предметної галузі, виявлено основні функціональні та технічні вимоги до системи. Виконано моделювання, а також проєктування апаратних і програмних компонентів, що забезпечують інтеграцію різних сенсорів і виконавчих механізмів. Було створено прототип будинку та розроблено систему розумного дому, яка забезпечує автоматизований моніторинг та керування різноманітними параметрами житлового простору на базі платформи Arduino та програмного середовища Node-RED. Система дозволяє дистанційно контролювати освітлення, клімат, безпеку і енергоспоживання, підвищуючи комфорт і ефективність користування будинком.

РЕФЕРАТ

Кваліфікаційна бакалаврська робота містить 95 сторінок, 7 таблиць, 33 рисунків, список використаних джерел з 29 найменувань, двох додатків.

«Розробка комплексної системи розумного будинку на базі Arduino»

Ключові слова: розумний будинок, автоматизація, моніторинг, Arduino, сенсори, керування, інформаційна система.

У бакалаврській кваліфікаційній роботі здійснено процес проектування та реалізацію інформаційної системи в рамках концепції «розумного будинку». Робота присвячена розробці інтегрованого рішення для автоматизованого моніторингу та керування домашніми системами з використанням сучасних апаратних та програмних засобів.

Об'єкт дослідження – інформаційна система керування елементами розумного будинку.

Предмет дослідження – процеси розробки, інтеграції та взаємодії апаратних (сенсори, виконавчі пристрої) та програмних (Arduino, Node-RED) компонентів у складі інтелектуальної системи.

Метою дослідження є створення конкурентоспроможної інформаційної системи, здатної автоматично здійснювати моніторинг середовища, виявляти події (рух, витoki води, перевищення температури або рівня газу), управляти освітленням, вентиляцією, сигналізацією та інформувати користувача у зручній формі через мобільний застосунок або месенджер.

Відповідно до поставленої мети визначено такі завдання дослідження:

- провести аналіз предметної галузі та існуючих технологій автоматизації побутових процесів;
- сформулювати вимоги до інформаційної системи;
- здійснити моделювання архітектури системи та її функціональних модулів;

- розробити технічне, програмне та інформаційне забезпечення системи;
- реалізувати прототип системи з використанням Arduino-контролера, сенсорів, виконавчих механізмів та інтерфейсів взаємодії;
- оцінити ефективність і перспективи впровадження системи в реальних умовах.

Результатом дослідження є створена та протестована інтелектуальна система керування, що реалізує концепцію «розумного будинку» з можливістю масштабування та подальшого розвитку. У роботі описано архітектуру системи, алгоритми взаємодії компонентів, принципи збирання, обробки та передачі даних, а також наведено приклади сценаріїв її роботи в автоматичному та напівавтоматичному режимах.

Теоретична значущість полягає у систематизації знань з розробки вбудованих систем і інтеграції апаратного та програмного забезпечення.

Методична значущість – у побудові універсального підходу до створення масштабованих рішень для автоматизації побуту.

Практична значущість – у можливості застосування розробленої системи для покращення комфорту, безпеки та енергоефективності житлових приміщень.

Рік виконання кваліфікаційної бакалаврської роботи – 2025.

Рік захисту роботи – 2025.

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВЕЛИЧИН І ТЕРМІНІВ	12
ВСТУП	13
РОЗДІЛ 1 ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	16
1.1 Характеристика предметної галузі та об'єкта дослідження.....	16
1.2 Аналіз літературних джерел та практичного досвіду використання інформаційних систем і технологій у предметній галузі.....	18
РОЗДІЛ 2 РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	21
2.1 Аналіз і специфікація вимог до інформаційної системи	21
2.1.1 Бізнес-вимоги	21
2.1.2 Функціональні вимоги.....	23
2.1.3 Нефункціональні вимоги до системи	28
2.2. Постановка та алгоритм розв'язання задачі.....	30
2.2.1 Постановка задачі	30
2.2.2 Вихідна інформація	33
2.2.3 Вхідна інформація	35
2.2.4 Алгоритм розв'язання задачі	36
2.3 Моделювання інформаційної системи.....	40
РОЗДІЛ 3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ	46
3.1 Інформаційне забезпечення	46
3.1.1 Загальна характеристика інформаційного забезпечення (ІЗ);.....	47
3.1.2 Організація збору і передання первинної інформації;.....	48
3.1.3 Побудова системи класифікації та кодування	50
3.1.4 Проектування форм первинних документів, машинограм.....	52
3.1.5 Структура інформаційних масивів	55
3.2 Технічне забезпечення.....	56
3.2.1 Загальні положення та схема автоматизації	56
3.2.2 Структура комплексу технічних засобів.	60
3.2.3 Опис автоматизованого робочого місця.....	69
3.3 Програмне забезпечення	71
3.3.1 Структура програмного забезпечення	71

3.3.2 Системне Програмне Забезпечення Системи «Розумний Будинок».....	74
3.3.3 Прикладне програмне забезпечення	75
3.3.4 Програмна Документація Системи «Розумний Будинок»	77
3.4 Результати реалізації інформаційної підсистеми	80
ВИСНОВКИ.....	84
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	86
ДОДАТКИ.....	89

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВЕЛИЧИН І ТЕРМІНІВ

HTML - HyperText Markup Language (Мова гіпертекстової розмітки);

JSON - JavaScript Object Notation (Об'єктний запис JavaScript);

UML - Unified Modeling Language (Уніфікована мова моделювання);

CSS - Cascading Style Sheets (Каскадні таблиці стилів);

LED – Light Emitting Diode (Світлодіод);

PIR - Passive Infrared Sensor (Інфрачервоний датчик руху);

DHT - Digital Humidity and Temperature Sensor (Датчик температури і вологості);

RFID - Radio Frequency Identification (Ідентифікація за радіочастотами);

LDR - Light Dependent Resistor (Світлочутливий резистор (фоторезистор));

MQ-2 - Газові датчики (дим);

MQ-7 - Газові датчики (CO);

I2C - Inter-Integrated Circuit (Шина для комунікації між мікроконтролерами та сенсорами);

ВСТУП

У сучасних умовах зростання вартості енергоресурсів, посилення вимог до безпеки та підвищення інтересу до комфорту в побуті важливим завданням є впровадження розумних технологій в житлові приміщення. Існуючі на сьогодні системи розумного будинку здебільшого орієнтовані на закриті комерційні екосистеми або потребують значних фінансових витрат, що ускладнює їхнє використання у побуті пересічного користувача.

Наразі наявні чисельні комерційні рішення, такі як Google Nest або Amazon Alexa тощо, але більшість з них не підтримує локалізацію, потребує хмарних сервісів та зазвичай має високу вартість, що не відповідає можливостям пересічного українського споживача.

У зв'язку з цим розробка комплексної, відкритої та доступної системи розумного будинку на базі платформи Arduino [1] є надзвичайно актуальною. Така система мала би поєднати функції енергозбереження, безпеки життя та контролю життєвого мікроклімату з використанням недорогих апаратних рішень і гнучкого програмного забезпечення.

Дизайн Arduino-платформи загалом має забезпечити гнучкість, простоту у налаштуваннях, відкритість. Особливого значення набувають функції автоматичного керування освітленням та опаленням, моніторинг температури та вологості, виявлення руху, ідентифікація випадків протікання води, диму, а також інтеграція з існуючими системами безпеки.

Розробка Arduino-платформи буде оптимальним рішенням для реалізації серед подібних систем в умовах обмеженого бюджету. Вона сприятиме підвищенню ефективності використання ресурсів, зниженню та енергетичних витрат і покращення умов проживання.

Отже, створення відкритої, локалізованої системи на базі Arduino [1], з можливістю гнучкого налаштування та розширення, є значущим завданням.

Метою даної роботи є розробка комплексної системи розумного будинку з функціями енергозбереження, безпекою та мікрокліматичного контролю на базі Arduino, яка забезпечить автоматизоване керування освітленням, кліматом та

безпековими системами для підвищення комфорту і ефективності використання ресурсів у житлових приміщеннях.

Основними завданнями для досягнення поставленої мети є:

1. Провести аналіз існуючих систем розумного будинку та доступних апаратних і програмних засобів реалізації подібних рішень.
2. Сформулювати вимоги до системи, описати логіку її роботи, побудувати функціональні діаграми та архітектурні моделі.
3. Реалізувати систему на базі мікроконтролера Arduino, використовуючи сукупність датчиків температури і вологості, руху і освітлення, газу, протікання води, RFID-автентифікацію, а також засобів виводу інформації Oled-дисплей, світлодіоди, зумери.
4. Перевірити працездатність системи в реальних умовах, оцінити її енергоефективність, надійність і зручність використання. Оцінити економічну доцільність для пересічних споживачів.

Запропонована система має високу практичну цінність, адже передбачає недорого та ефективно реалізувати функції розумного будинку в українських реаліях, з урахуванням обмежених ресурсів, високих вимог до зручності та безпеки.

Об'єктом дослідження є процеси автоматизації житлових приміщень за допомогою мікроконтролерних систем. [2] Предметом дослідження є методи, інструменти та підходи до проектування і реалізації вбудованих систем автоматизації, що забезпечують комплексне керування житловим приміщенням з метою підвищення енергоефективності, комфорту та безпеки. Зокрема, дослідження охоплює побудову апаратно-програмної архітектури розумного будинку.

У ході дослідження застосовано комплекс методів, що забезпечують всебічний аналіз та реалізацію поставлених задач. Для проведення теоретичного аналізу були використані джерела, що висвітлюють принципи побудови вбудованих систем автоматизації, новітні підходи у сфері IoT [3, с. 20], а також практичні кейси впровадження розумних будинків у реальних умовах. Для моделювання системи використано підходи структурного та об'єкто-орієнтованого програмування з побудовою діаграм вимог, структур схем та логічної архітектури.

Практична частина ґрунтується на розробці програмного забезпечення для контролера Arduino з використання мов програмування C/C++, бібліотек U8g2, MRFC522, DHT, а також підключення датчиків та виконавчих елементів за допомогою цифрових і аналогових інтерфейсів.

Теоретична значущість полягає у систематизації сучасних підходів до проєктування енергоефективних та безпечних вбудованих систем для автоматизації житлових приміщень, з урахуванням обмежень щодо вартості, доступності компонентів і специфіки умов експлуатації в Україні. Вдосконалено принципи побудови інтелектуальних систем керування на базі відкритих апаратних платформ, зокрема Arduino, та можливості таких платформ для масштабування і адаптації.

Практична значущість полягає у створенні працездатного прототипу розумного будинку, в якому втілюється автоматичне керування світлом, температурою, вентиляцією та безпекою, з інтерактивною взаємодією з користувачем. Система може бути реалізована в реальних умовах приватного житла або невеликих комерційних об'єктів, що сприятиме зниженню споживання енергії, підвищенню рівня безпеки та покращенню якості життя мешканців.

Інформаційна база дослідження включає технічну документацію апаратних модулів (Arduino Mega, датчики DHT11, DHT22, MQ-2, MQ-7, PIR, LDR тощо), наукові праці українських та зарубіжних дослідників у сфері IoT і систем автоматизації, офіційні стандарти побудови вбудованих систем, систем пожежної безпеки, відкриті публікації розробників Arduino-спільноти, а також приклади реальних проєктів.

РОЗДІЛ 1 ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Характеристика предметної галузі та об'єкта дослідження

Предметна галузь дослідження охоплює сферу інтелектуальних систем керування житловими або комерційними об'єктами, зокрема систем «розумного будинку», що базуються на технологіях Інтернету речей (IoT), мікроконтролерах та сенсорних мережах. Ці системи дозволяють автоматизувати процеси моніторингу, керування електрообладнанням, освітленням, кліматом, безпекою та енергоспоживанням.

Системи розумного будинку знаходять широке застосування в умовах сучасного технологічного середовища, особливо з огляду на підвищення вимог до енергоефективності, безпеки житла та комфорту мешканців. Вони охоплюють рішення для приватних будинків, квартир, офісів, готельного бізнесу, малих комерційних об'єктів.

Об'єкт дослідження – процес побудови автоматизованої інформаційної системи управління елементами розумного будинку такими освітлення, температура, рух, сигналізація, виявлення газу/води, що ґрунтується на взаємодії сенсорів, виконавчих пристроїв та користувача через зручний інтерфейс. Проблемна ситуація пов'язана з необхідністю інтеграції численних сенсорів і пристроїв у єдину систему з високим рівнем надійності, масштабованості та зручності керування.

Предметна галузь охоплює:

- житлові будинки та квартири;
- офіси та приміщення комерційного призначення;
- користувачі (власники житла, члени сім'ї, адміністратори системи).

Серед типових рішень, що приймаються в рамках предметної галузі:

- активація/деактивація освітлення на основі руху або часу доби;

- регулювання температури та вологості відповідно до заданих параметрів;
- увімкнення сигналізації за відсутності авторизації користувача;
- оповіщення про небезпечні ситуації (витік газу, води);
- Дистанційне керування всіма процесами через інтерфейс користувача.

Фактори впливу на рішення:

- навколишні умови (світло, температура, вологість, рух);
- людський фактор (поведінка мешканців);
- доступність інтернету або локальної мережі;

Інформаційне забезпечення рішень складається з:

- внутрішньої інформації: дані з сенсорів (температури, руху, освітлення тощо), статуси виконавчих пристроїв (світлодіоди, LED-стрічки, зумери);
- зовнішньої інформації: параметри встановлені користувачем, дані з мобільного застосунку або хмарного серверу.

Методи оброблення інформації включають: об'єктно-орієнтоване проектування, алгоритмізацію подій, спрощені рішення до адаптації режимів роботи, аналіз сценаріїв взаємодії користувача з системою.

Прийняття рішень у таких системах може здійснюватись:

- у визначених умовах (наприклад, якщо температура перевищує 30°C – увімкнути вентилятор);
- в умовах ризику (наприклад, можлива помилкова активація сигналізації);
- в умовах невизначеності (недоступність інтернету, обрив живлення).

Отже, «розумний будинок» – це складна система, в якій поєднуються загальні принципи, методи та технології її реалізації для конкретного типу об'єктів,

інструментарій для автоматизації, алгоритмічного прийняття рішень, забезпечення зручності для користувача.

Специфіка предметної галузі зумовлюють необхідність та практичної реалізації гнучких та надійних рішень, здатних адаптуватися до різних умов середовища та потреб користувача.

1.2 Аналіз літературних джерел та практичного досвіду використання інформаційних систем і технологій у предметній галузі

Упродовж останніх десятиліть концепція розумного будинку (Smart Home) набула широкого поширення завдяки розвитку технологій Інтернету речей, мікроконтролерів, бездротових сенсорних мереж, хмарних сервісів та мобільних застосунків. «Розумний будинок» — це автоматизована система, яка забезпечує моніторинг і керування різними процесами в житлових приміщеннях (освітлення, безпека, клімат, енергоспоживання тощо) з метою підвищення комфорту, безпеки та енергоефективності.

Згідно з дослідженнями [4] [5] [6], ключовими технологіями розумного будинку є:

- мікроконтролери (наприклад, Arduino [1], ESP32, Raspberry Pi);
- бездротові протоколи зв'язку (Wi-Fi, Zigbee, , Bluetooth);
- сенсорні пристрої (датчики температури, вологості, руху, газу, освітлення, води);
- актори (LED-стрічки, вентилятори, зумери тощо);
- хмарні платформи (Blynk, Firebase, Home Assistant [7]) для збереження, візуалізації та віддаленого керування даними.

Сучасний розумний дім пропонує комплексний підхід до автоматизації, що значно підвищує комфорт та безпеку мешканців, а також сприяє енергоефективності. Уявіть, що ваш дім сам адаптується до ваших потреб: клімат-контроль автоматично підтримує ідеальну температуру та вологість,

використовуючи DHT-сенсори для керування вентиляторами, кондиціонерами та обігрівачами. При цьому освітлення реагує на ваше перебування, вмикаючи LED-стрічки або лампи за допомогою датчиків руху та освітленості, або ж за вашим дотиком до сенсорних кнопок.

Важливою складовою такого будинку є його безпека. Система сигналізації, що ґрунтується на RFID, PIR-датчиках та детекторах диму, газу й води, забезпечує надійний захист від несанкціонованого доступу та надзвичайних ситуацій. Всі дані про стан системи легко доступні для візуалізації на OLED-дисплеях, веб-інтерфейс, дозволяючи вам завжди бути в курсі подій. Більше того, інтелектуальні сценарії дозволяють системі автоматично реагувати на зміну умов, наприклад, виявлення руху вночі може автоматично увімкнути світло, роблячи ваше життя ще зручнішим та безпечнішим чи режим “Відпуску” при вмиканні якого світло у вечірній час буде вмикатись випадково в різних кімнатах роблячи вигляд присутності.

Популярні відкриті рішення, такі як OpenHAB [8], Home Assistant [7], Domoticz [9], часто використовуються для централізованого керування smart home-системами, проте їх встановлення і налаштування здебільшого потребують додаткових знань щодо мереж, Linux-систем, а також відповідної апаратної підтримки.

Таблиця 1.1 – Порівняння підходів до розумного будинку

Критерій	Arduino-підхід	Home Assistant	Комерційні системи (Ajax [27], Fibaro[29])
Гнучкість	Висока	Висока	Середня
Ціна реалізації	Низька	Середня	Висока
Потрібні знання	Середній рівень	Високий рівень	Мінімальні(готові рішення)
Локалізація	Повна контрольована	Часткова	Обмежена або платна
Масштабованість	Висока за умови правильного налаштування	Висока	Залежить від моделі
Інтеграція з мобільними застосунками	Так(наприклад через MQTT)	Так	Так (через фірмовий додаток)

Джерело: сформоване автором на основі виконаних досліджень

На основі аналізу зроблено висновок, що власноруч розроблена система розумного будинку на базі Arduino або ESP може бути доцільним варіантом для

навчальних і недорогих проєктів. Вона забезпечує повну гнучкість у виборі функціоналу, реалізації логіки, локалізації інтерфейсів і адаптації до потреб користувача.

У межах цієї кваліфікаційної роботи реалізується власна версія smart home-системи з підтримкою:

- багатозонного освітлення з ручним та автоматичним керуванням;
- моніторингу температури, вологості, руху, води, газу;
- візуального інтерфейсу на OLED-дисплеї;
- підтримки RFID-модуля для ідентифікації користувача;
- модульної архітектури з можливістю розширення.

Таким чином, аналіз літературних і практичних джерел підтверджує актуальність тематики і обраного підходу до реалізації автоматизованої системи розумного будинку на основі вільно доступних технологій.

РОЗДІЛ 2 РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Аналіз і специфікація вимог до інформаційної системи

Вимоги — це умови або можливості, що їх повинна мати інформаційна система або її компоненти для виконання контракту або стандартів, специфікацій чи інших формальних документів, а також їх документоване подання. Визначення і керування вимогами є важливою частиною системного інжинірингу, що значною мірою визначає успішність проекту розробки. Під час розроблення вимог до системи «Розумний будинок» було визначено бізнес-вимоги, функціональні вимоги та нефункціональні вимоги, які повинна враховувати система. Основна мета системи полягає в автоматизації контролю та управління побутовими процесами для підвищення комфорту, енергоефективності та безпеки житла.

2.1.1 Бізнес-вимоги

Бізнес-вимоги визначають основну функціональність та вимоги, що стосуються діяльності додатку з автоматизації. Для цього була розроблена діаграма бізнес-вимог (рис. 2.1), яка відображає основні аспекти функціонування та вимоги до системи

Бізнес-вимоги:

BR-01: Підвищення рівня безпеки будинку

Система повинна забезпечувати цілодобовий контроль за входами, вікнами, рухом усередині та витоками газу чи води, щоб запобігти несанкціонованому доступу та аваріям.

BR-02: Енергоефективність і зниження витрат на комунальні послуги

Автоматичне регулювання освітлення, температури та вентиляції повинно зменшувати споживання електроенергії та ресурсів без шкоди для комфорту мешканців.

BR-03: Зручне та централізоване керування будинком

Користувачі повинні мати можливість керувати всіма підсистемами будинку (освітлення, безпека, клімат, доступ) через єдиний інтерфейс — мобільний додаток або веб-панель.

BR-04: Адаптація до індивідуальних потреб користувача

Система повинна підтримувати налаштування персональних сценаріїв, які відповідають стилю життя, звичкам та пріоритетам мешканців.

BR-05: Високий рівень надійності та безперервної роботи

Усі компоненти системи повинні працювати стабільно, з можливістю самодіагностики, сповіщення про збої та аварійне перемикання в ручний режим.

BR-06: Віддалений моніторинг та управління в режимі реального часу

Користувач повинен мати змогу отримувати дані про стан системи та керувати пристроями.

BR-07: Надсилання сповіщень через SMS

Система повинна підтримувати надсилання миттєвих сповіщень про події в будинку (спрацювання датчиків руху, диму, витоку води, відкриття дверей тощо) через SMS на смартфони користувачів, забезпечуючи швидкий та зручний зв'язок з системою.

BR-08: Підвищення комфорту мешканців

Автоматизація побутових процесів (освітлення, мікроклімат, полив, вентиляція тощо) має створювати максимально комфортне середовище для життя.

BR-09: Захист персональних даних та конфіденційності

Система має гарантувати безпечне зберігання та передачу даних користувача, забезпечуючи захист від несанкціонованого доступу.

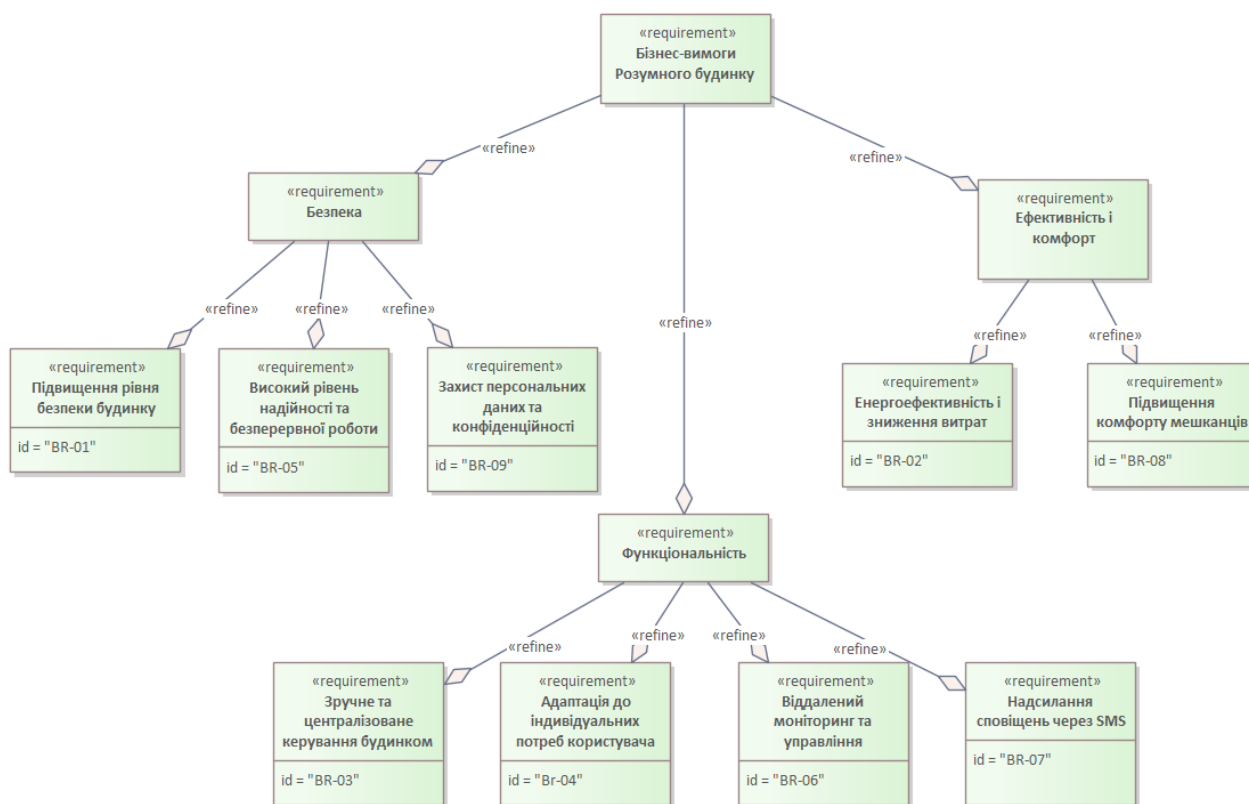


Рис 2.1 – Бізнес-вимоги

Джерело: сформоване автором на основі виконаних досліджень

2.1.2 Функціональні вимоги

Функціональні вимоги — це опис того, як саме повинна поводитися система розумного будинку. Вони визначають, які функції має виконувати система для задоволення потреб мешканців, таких як автоматичне керування освітленням, кліматом, безпекою тощо. Ці вимоги формують основу взаємодії користувача з системою — саме через них користувач бачить і відчуває функціональність «розумного будинку».

Деталізований опис функціональних вимог надається у документі Специфікації вимог до програмного забезпечення (SRS) [10]. Цей документ є ключовим у всіх фазах життєвого циклу розробки системи — від проєктування і реалізації до тестування, впровадження та технічного обслуговування. Він допомагає розробникам, тестувальникам, менеджерам проєктів і користувачам узгодити очікування та забезпечити якість кінцевого рішення.

Функціональні вимоги до системи

FR-01: Автоматичне вмикання/вимикання світла залежно від присутності у кімнаті.

FR-02: Керування яскравістю освітлення вручну або автоматично залежно від часу доби та освітленості.

FR-03: Підтримка різних сценаріїв освітлення (день/ніч, відпочинок, робота).

FR-04: Можливість дистанційного керування освітленням через мобільний застосунок.

FR-05: Автоматичне регулювання температури в приміщеннях на основі показників сенсорів.

FR-06: Автоматичне керування вологістю повітря (вентиляція, зволоження).

FR-07: Визначення рівня температури/вологи в режимі реального часу.

FR-08: Керування вентиляторами або кондиціонером залежно від кліматичних умов.

FR-09: Виявлення руху в приміщеннях за допомогою PIR-датчиків.

FR-10: Виявлення витoku води або загрози затоплення.

FR-11: Контроль за якістю повітря та наявністю небезпечних газів (наприклад, CO, дим).

FR-12: Активація сигналізації у разі вторгнення або неавторизованого доступу.

FR-13: Відправка сповіщень користувачу про небезпечні події.

FR-14: Розпізнавання користувача за допомогою RFID-технологій.

FR-15: Запис подій входу/виходу зареєстрованих користувачів.

FR-16: Режим блокування/розблокування системи вручну або за розкладом.

FR-17: Відображення інформації про стан сенсорів і пристроїв на OLED-дисплеї або в додатку.

FR-18: Ведення журналу подій системи для подальшого аналізу.

FR-19: Керування всіма зонами будинку через централізовану панель або мобільний застосунок.

FR-20: Можливість перемикавання між автоматичним та ручним режимом керування.

FR-21: Підтримка підключення нових сенсорів і пристроїв без зміни загальної архітектури.

FR-22: Сумісність із зовнішніми сервісами або платформами (наприклад, MQTT, Node-RED).

FR-23: Можливість створення користувацьких сценаріїв дій (наприклад, якщо температура $> 28^{\circ}\text{C}$ вмикається функція клімат-системи охолодження).

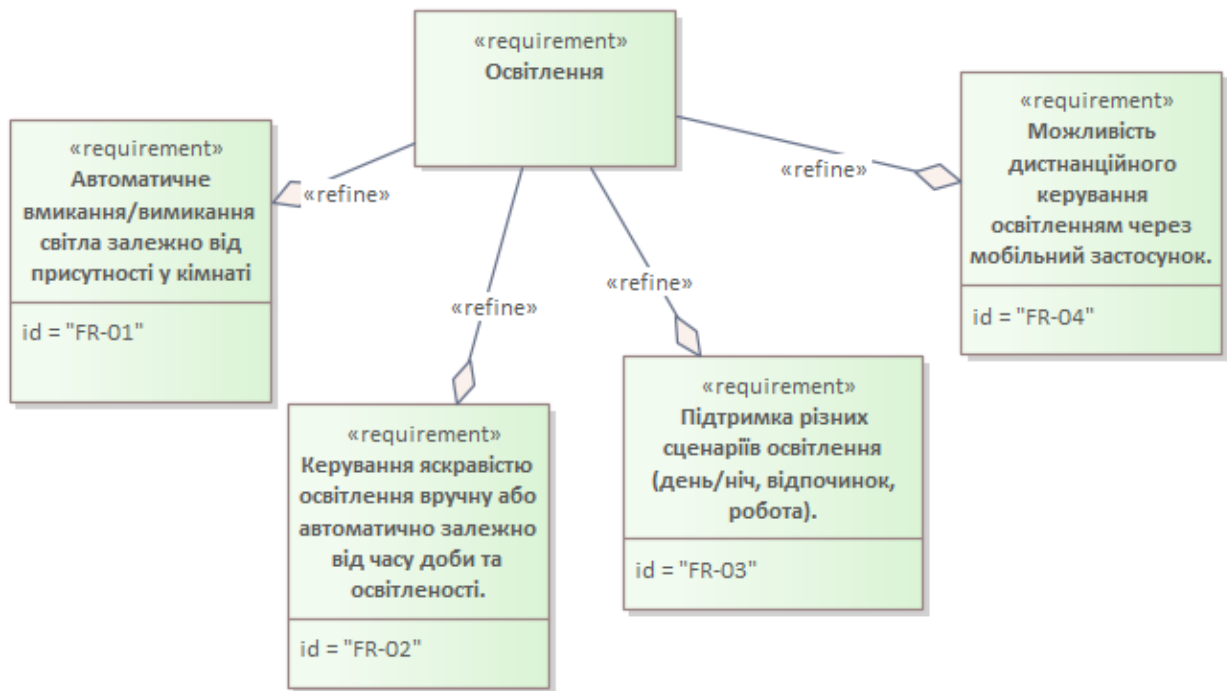


Рис. 2.2 – Контейнер функціональних вимог до Освітлення

Джерело: сформоване автором на основі виконаних досліджень

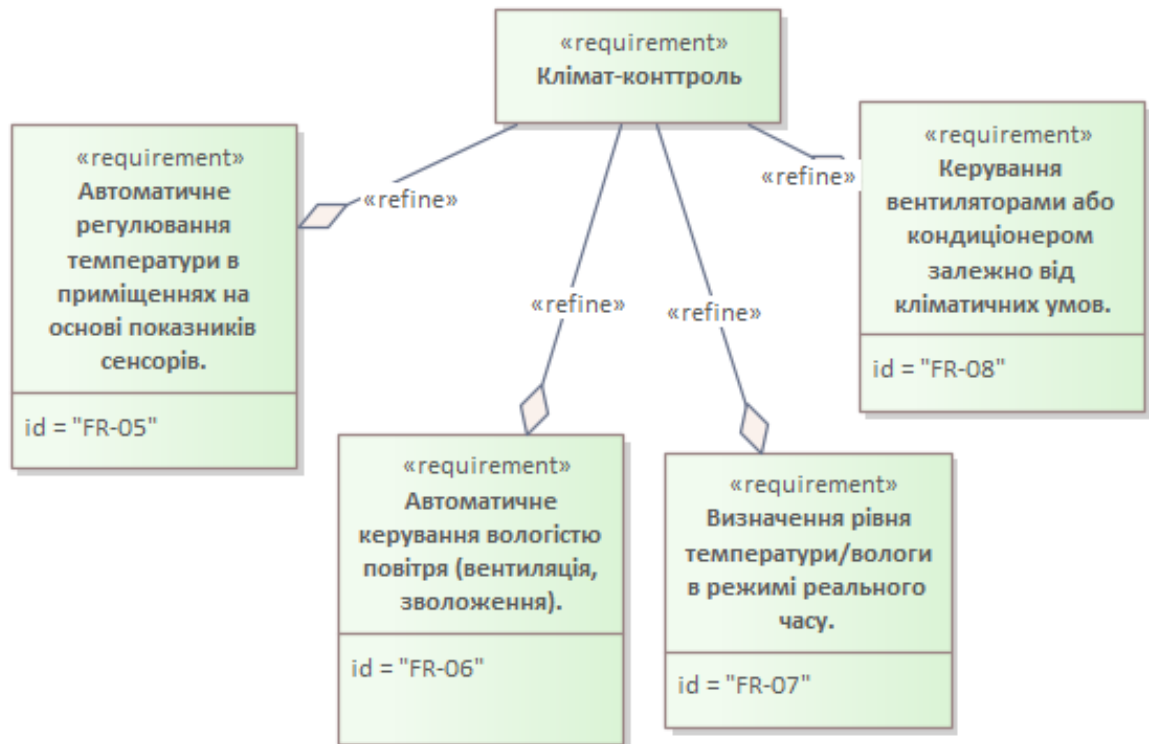


Рис. 2.3 – Контейнер функціональних вимог до Клімат контролю

Джерело: сформоване автором на основі виконаних досліджень

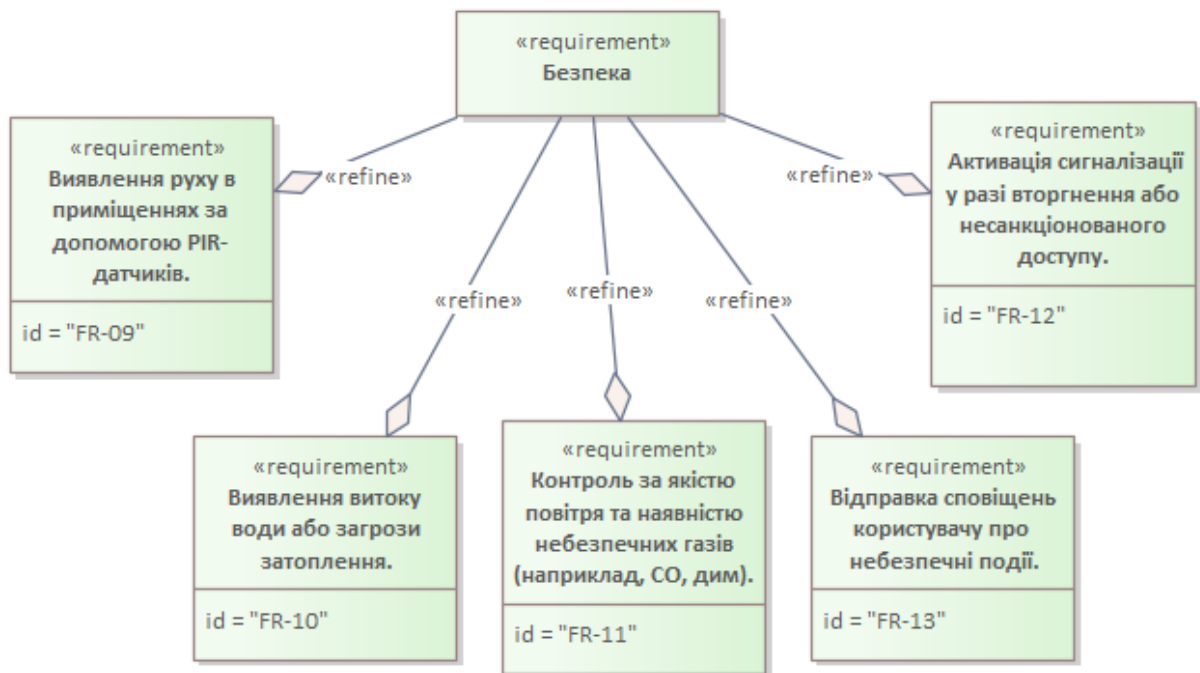


Рис. 2.4 – Контейнер функціональних вимог до Безпеки

Джерело: сформоване автором на основі виконаних досліджень

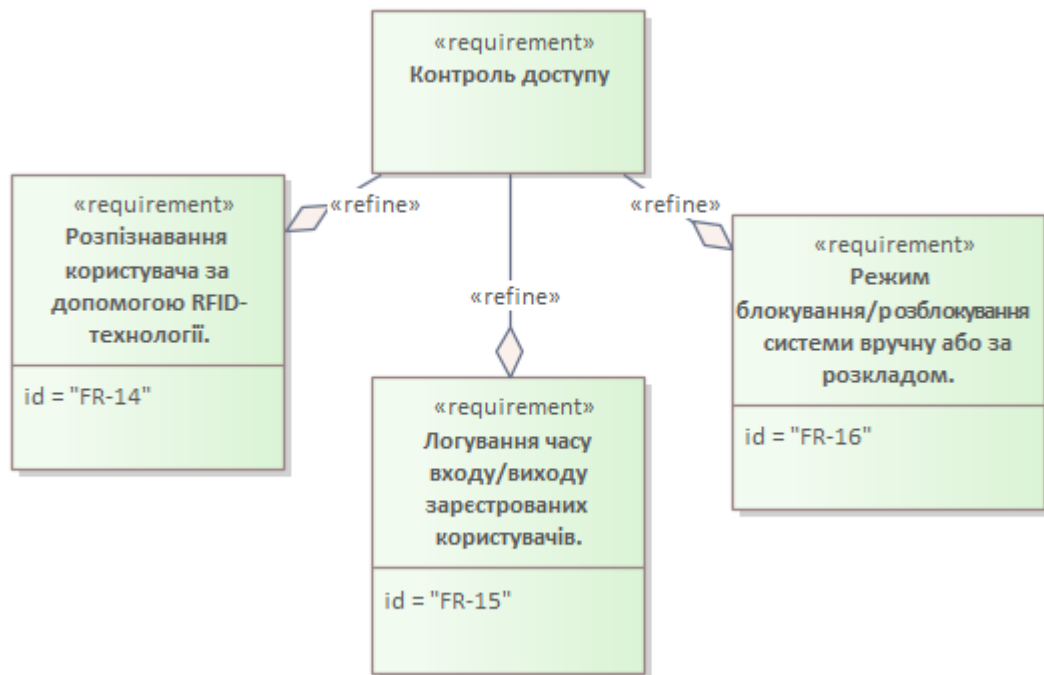


Рис. 2.5 – Контейнер функціональних вимог до Контролю доступу

Джерело: сформоване автором на основі виконаних досліджень

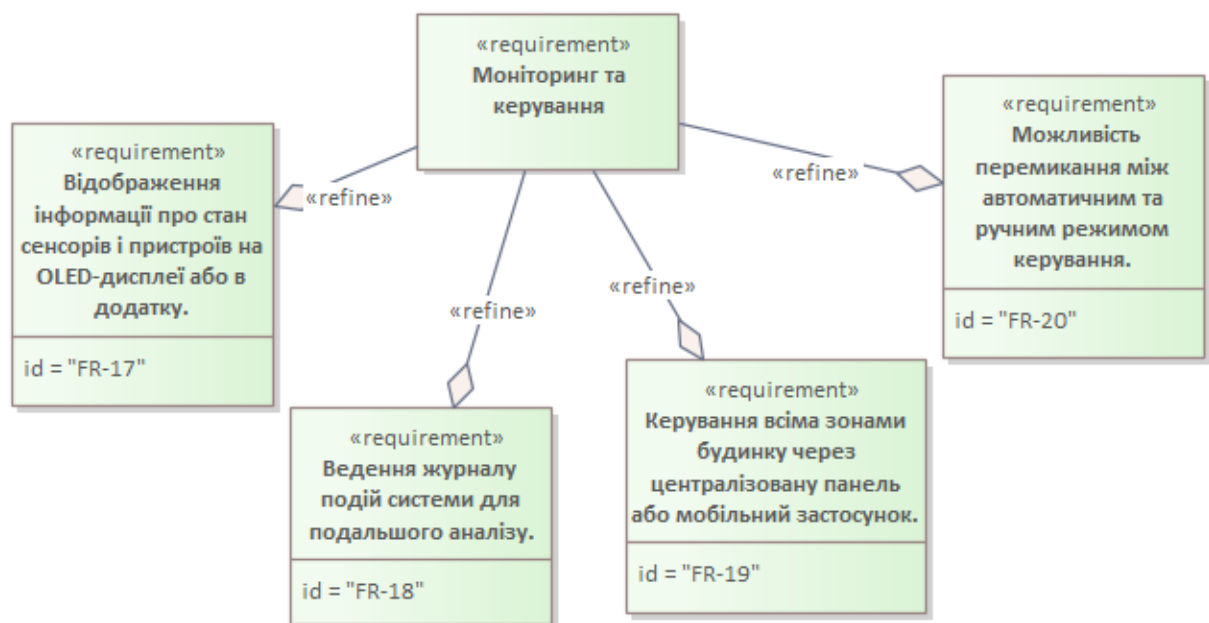


Рис. 2.6 – Контейнер функціональних вимог до Моніторингу та керування

Джерело: сформоване автором на основі виконаних досліджень

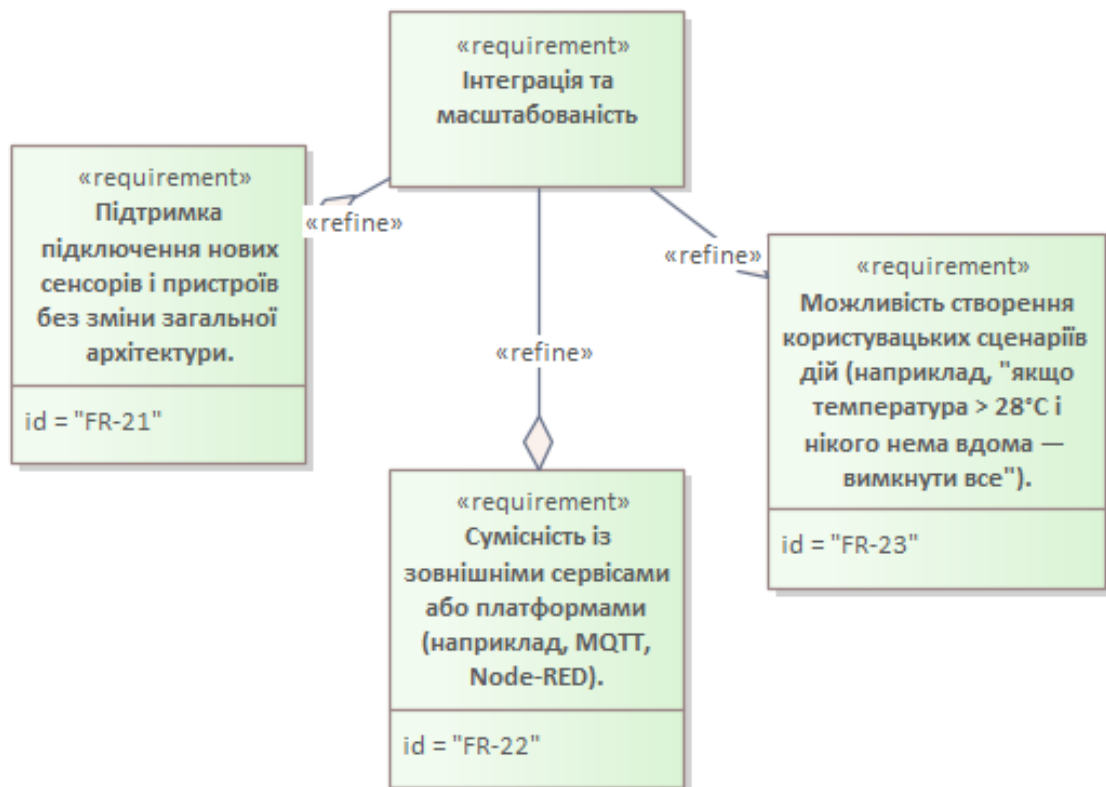


Рис. 2.7 – Контейнер функціональних вимог до Інтеграції та масштабованості

Джерело: сформоване автором на основі виконаних досліджень

Ці підсистеми визначають функціональну структуру інтелектуальної системи керування житлом, головна мета якої — автоматизоване забезпечення комфорту, безпеки та ефективного використання ресурсів у побуті. Кожна з підсистем — керування освітленням, клімат-контроль, моніторинг безпеки, контроль доступу, сповіщення та інтерфейс відображення, дистанційне керування, а також інтеграція з додатковими пристроями — виконує свою специфічну роль у загальній системі автоматизації. У сукупності ці підсистеми формують логічний зв'язок багаторівневої структури, яка реалізує основну бізнес-логіку та відповідає практичним потребам користувача в інтелектуальному керуванні домашнім середовищем.

2.1.3 Нефункціональні вимоги до системи

Нефункціональні вимоги (Non-Functional Requirements, NFR) визначають якісні характеристики системи розумного будинку, що впливають на її надійність, швидкодію, зручність використання, масштабованість та відповідність стандартам безпеки. Вони не описують, що система повинна робити, а як вона має виконувати свої функції. Згідно з міжнародним стандартом ISO/IEC 25000 [11], ці вимоги належать до категорії якості програмного забезпечення та інформаційних систем.

Нефункціональні вимоги до системи

NFR-01: Час реакції на події з сенсорів не перевищує 1 секунди у 95% випадків.

NFR-02: Керування пристроями (світло, вентиляція тощо) в режимі реального часу.

NFR-04: Система стабільно працює без перезавантаження щонайменше 30 днів.

NFR-05: У разі збою одного модуля — інші частини продовжують функціонувати.

NFR-06: Інтерфейс повинен бути зручним та логічним.

NFR-07: Можливість додавати нові пристрої без перепроєктування системи.

NFR-08: Підтримка Wi-Fi, Bluetooth, MQTT, Android, iOS.

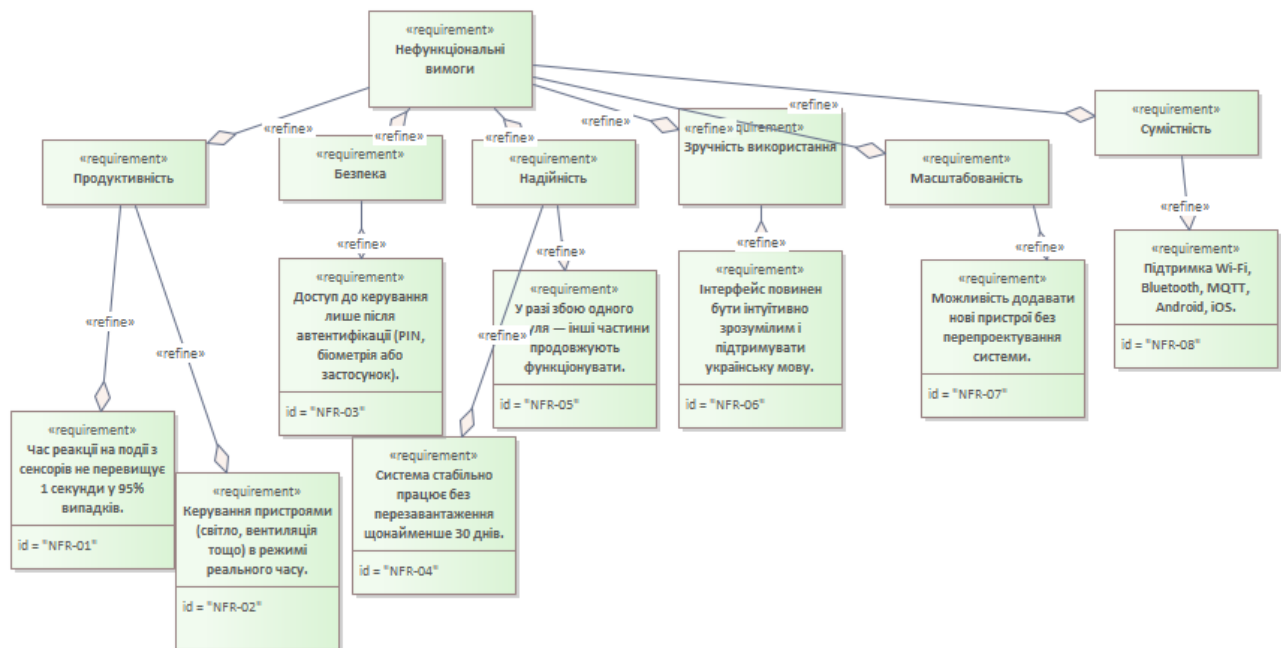


Рис. 2.8 – Нефункціональні вимоги

Джерело: сформоване автором на основі виконаних досліджень

Виконання нефункціональних вимог є критичним для забезпечення надійності, зручності та безпеки системи розумного будинку. На відміну від функціональних вимог, ці характеристики визначають загальну якість роботи, стабільність у різних умовах, адаптивність до нових технологій і відповідність очікуванням користувачів. Задоволення нефункціональних вимог гарантує, що система не лише «працює», а й забезпечує зручність у використанні на довгостроковій основі.

2.2. Постановка та алгоритм розв’язання задачі

2.2.1 Постановка задачі

У сучасних умовах активного розвитку цифрових технологій зростає потреба у впровадженні автоматизованих систем управління житлом, які забезпечують комфорт, енергоефективність та безпеку. Система «Розумний дім» є комплексом технічних та програмних засобів, що забезпечують централізоване або автоматизоване керування інженерними системами житлового приміщення,

зокрема освітленням, мікрокліматом, системами безпеки та моніторингу. Завданням даної автоматизованої системи є організація ефективного управління багатьма функціональними зонами житлового приміщення, де на основі показників з датчиків приймаються рішення про ввімкнення чи вимкнення світла, вентиляції, сирени тощо. При цьому застосовується комплекс мікроконтролерів, сенсорів (температури, вологості, руху, освітленості, диму, газу, води) та виконавчих пристроїв (вентилятори, сирени, світлодіодні стрічки), які взаємодіють відповідно до заданої логіки. Обґрунтуванням використання ЕОМ (у нашому випадку — мікроконтролерів Arduino) є потреба в безперервному моніторингу, швидкому реагуванні на події, паралельному обробленні даних з десятків сенсорів та забезпеченні взаємозв'язків між підсистемами.

Об'єктами управління в системі є джерела освітлення в окремих зонах, охоронна система (на основі датчиків руху та RFID-модуля), мікроклімат (датчики температури та вологості з керуванням вентиляторами), а також аварійні підсистеми реагування на витік води чи газу. Користувач може здійснювати як ручне керування (через сенсорні кнопки), так і автоматичне — коли система самостійно змінює стан виконавчих пристроїв залежно від ситуації.

Вхідна інформація надходить від численних сенсорів, що вимірюють параметри навколишнього середовища або фіксують зміни в оточенні. Зокрема, датчики температури та вологості, які подають інформацію про стан мікроклімату в кожній зоні. Датчики руху контролюють наявність присутності, фоторезисторні датчики фіксують рівень освітлення, а сенсори витоку води та газу — потенційні аварійні ситуації. Також вхідними даними є сигнали з RFID-міток, що ідентифікують користувачів для керування доступом.

Вихідна інформація формується у вигляді керуючих сигналів, що активують або деактивують відповідні пристрої (освітлення, вентилятори, сирени), а також у вигляді повідомлень для користувача — на OLED-дисплей або в інтерфейс мобільного додатку. Така інформація використовується для миттєвого реагування на події, побудови історії подій, прийняття рішень про зміну режимів роботи системи. Крім того, можуть генеруватися сигнали тривоги у разі виявлення загрози

(витік, несанкціонований доступ), що забезпечує своєчасне інформування користувача та запобігання можливим наслідкам.

Розв’язання задачі відбувається у безперервному режимі, з високою частотою опитування датчиків і майже миттєвою реакцією на події. Подача вихідної інформації також відбувається в реальному часі, що дозволяє швидко змінювати стани системи у відповідь на зміну вхідних параметрів. У разі виникнення аварійної ситуації, наприклад, втрати живлення або відмови одного з елементів, система може переходити в безпечний режим або очікувати ручного втручання користувача. Це дозволяє в перспективі розширити систему до рівня самонавчального або адаптивного середовища. Інтеграція різних модулів здійснюється централізовано, а розподіл функцій між автоматикою і людиною гнучкий: у штатному режимі управління здійснюється автоматично, однак користувач завжди має можливість вручну втрутитись у процеси через інтерфейс.

Таким чином, постановка задачі охоплює побудову комплексної автоматизованої системи, здатної керувати критично важливими компонентами житла в режимі реального часу, мінімізуючи вплив людського фактору та підвищуючи загальний рівень якості життя мешканців.

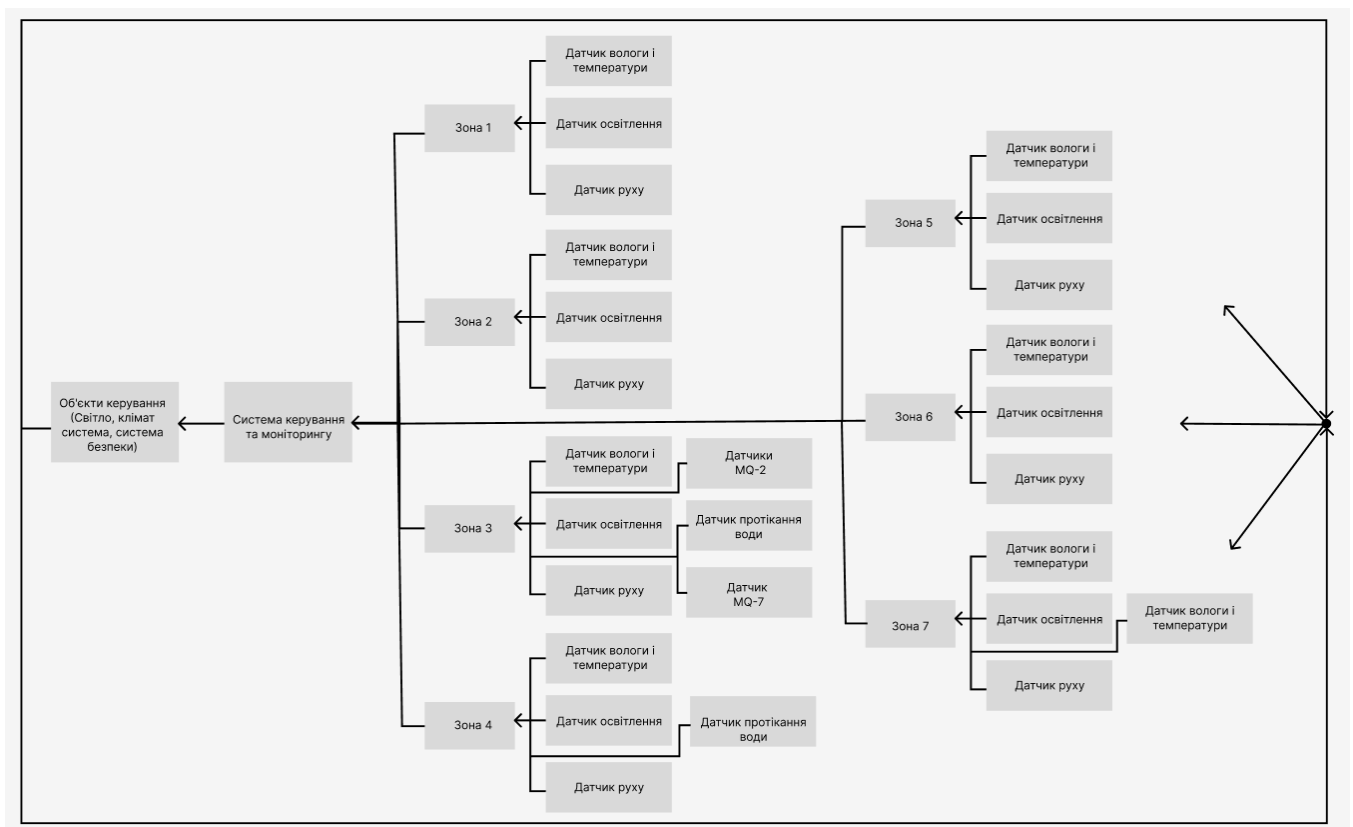


Рис. 2.9 – Інформаційна модель задачі

Джерело: сформоване автором на основі виконаних досліджень

2.2.2 Вихідна інформація

У контексті автоматизованої системи розумного будинку вихідна інформація є ключовим результатом обробки вхідних даних, отриманих від датчиків, інтерфейсів користувача, зовнішніх подій або сценаріїв керування. Вона формується у вигляді повідомлень, сигналів або керуючих команд і є основою для прийняття рішень як автоматичною системою, так і самими користувачами.

Вихідна інформація у системі розумного будинку виконує багатофункціональну роль, забезпечуючи зворотний зв'язок між технологією та користувачем для ефективного функціонування та зручності взаємодії. Вона призначена для оперативного інформування користувачів про поточний стан приміщень та обладнання, що дозволяє миттєво реагувати на будь-які зміни. На основі цих даних система здатна автоматично виконувати дії, такі як увімкнення вентиляції, освітлення або сирени, оптимізуючи комфорт та безпеку без прямого втручання людини.

Крім того, вихідна інформація є основою для реєстрації подій і ведення журналів, що є критично важливим для подальшої аналітики, аудиту або прийняття обґрунтованих управлінських рішень. Для кращого розуміння та звітності, ці дані можуть бути представлені у вигляді офіційних документів і графічних звітів. Завдяки інтеграції з мобільними додатками та іншими IoT-системами через стандартизовані формати повідомлень (як-от JSON, MQTT), забезпечується безперервний обмін інформацією та розширюються можливості управління. Нарешті, ця інформація є ключовою для забезпечення безпеки, сповіщаючи про аварійні ситуації чи несанкціонований доступ, а також дозволяє контролювати доступ, енергоспоживання та кліматичні параметри у режимі реального часу. Таким чином, вихідна інформація є не просто даними, а життєво важливим

інструментом, що забезпечує розуміння, контроль та ефективне управління вашим розумним будинком.

Таблиця 2.1 – Вихідні дані

№	Назва вихідного повідомлення	Ідентифікатор	Періодичність видання	Термін	Користувачі інформації
1	Звіт про клімат у приміщенні	CLIMATE	Щосекунди	Миттєво	Користувач
2	Повідомлення про вторгнення	ALERT_INTRUSION	У режимі реального часу	Негайно	Власник
3	Статус освітлення по зонах	LIGHT_STATUS	Щосекунди	Миттєво	Користувач
4	Графік освітлення	Інфографіка	У режимі реального часу	Затримка до 1 хвилини	Власник
5	Параметри керування кліматом (режим, рівень, цілі)	CLIMATE_CONTROL_STATE	За потреби	Миттєво	Власник
6	Повідомлення про зміну RFID доступу	ACCESS_LOG	У режимі реального часу	Негайно	Користувач
7	Графік температури	Інфографіка	У режимі реального часу	Затримка до 1 хвилини	Власник
8	Повідомлення про аварію (вода, газ, дим)	ALERT_EMERGENCY	У режимі реального часу	Негайно	Користувач

Джерело: сформоване автором на основі виконаних досліджень

2.2.3 Вхідна інформація

Вхідна інформація в системі «Розумний дім» є основою для її функціонування, забезпечуючи повноцінне розуміння навколишнього середовища та дій користувачів. Вона призначена для зчитування поточних фізичних параметрів середовища, таких як температура, вологість, рух, наявність газу, води та рівень освітленості. Це дозволяє системі постійно моніторити умови в приміщенні. Крім того, вхідні дані включають реагування на дій користувачів, що здійснюється за допомогою RFID-міток або сенсорних кнопок, дозволяючи системі виконувати задані команди. Система також здатна отримувати зовнішні події або тригери з інших модулів, що розширює її інтеграційні можливості та дозволяє створювати складніші сценарії взаємодії. Не менш важливою є підтримка постійної довідкової інформації, включно з ID зон, типами сценаріїв та параметрами налаштувань, що забезпечує стабільність та послідовність роботи. Вся ця інформація надходить у різноманітних форматах: від цифрових сигналів та аналогових значень до структурованих JSON-даних та збережених конфігурацій.

Вона обробляється або у режимі реального часу, забезпечуючи миттєву реакцію, або згідно з сценаріями, дозволяючи легко налаштовувати систему.

№	Назва вхідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надходження	Джерело
1	Дані з температурно-вологісних	DHT_DATA	JSON	Щосекунди	DHT11/DHT22 датчики
2	Стан PIR-датчиків руху	MOTION_EVT	Логічний сигнал	Подієво	PIR сенсори
3	Дані з фоторезисторів	LDR_VAL	Ціле число (0–1023)	Щосекунди	Аналогові LDR сенсори
4	Стан сенсорів диму/газу	GAS_ALERT	Логічне значення	Подієво	MQ-2, MQ-7
5	Сигнали від сенсорів води	WATER_EVT	Логічний сигнал	Подієво	Датчики протікання
6	Вхід з RFID зчитувача	RFID_IN	Строка	Подієво	RFID-зчитувач (MFRC522)
7	Сенсорні натискання на кнопки	TOUCH_EVT	Логічний сигнал	Подієво	Сенсорні кнопки
8	Команди користувача з панелі керування	USR_CMD	JSON команда	За запитом користувача	Node-RED Dashboard

Джерело: сформоване автором на основі виконаних досліджень

2.2.4 Алгоритм розв’язання задачі

У системі «Розумний будинок» обробляється великий обсяг даних, які надходять від сенсорів і пристроїв взаємодії з користувачем. Ця інформація є

основою для прийняття рішень щодо автоматичного керування окремими підсистемами, такими як освітлення, клімат-контроль, охорона, сигналізація тощо.

Основними джерелами даних є сенсори температури та вологості (типу DHT11 або DHT22), які встановлені у ключових зонах будинку. Ці сенсори забезпечують постійний моніторинг параметрів мікроклімату, що дозволяє системі реагувати на зміни та підтримувати комфортні умови шляхом керування вентиляторами або охолоджувальними пристроями. Інфрачервоні датчики руху (PIR) використовуються для виявлення присутності людей у приміщенні. Завдяки цим датчикам реалізується автоматичне вмикання освітлення, а також контроль за несанкціонованим проникненням при активованій охоронній системі. Для визначення рівня освітленості в приміщенні застосовуються фоторезистори (датчики LDR). Дані з цих сенсорів дозволяють системі оцінювати природне освітлення та приймати рішення щодо доцільності вмикання штучного світла, що сприяє енергоефективності.

Сенсорні кнопки керування встановлюються у зонах доступу користувачів і дозволяють вручну змінювати стани освітлення або інші параметри. Стани цих кнопок регулярно опитуються та зберігаються в окремому масиві. Для контролю доступу до функцій системи, зокрема охоронної сигналізації, використовується технологія RFID. Кожен користувач має індивідуальну мітку (картку), що дозволяє йому авторизуватись у системі. Після успішної ідентифікації можливе зняття будинку з охорони, зміна режиму або керування іншими функціями.

Таблиця 2.3 – Перелік масивів використовуваної інформації

Масив	Ідентифікатор	Макс. кількість записів
Дані температури/вологості	DHT_DATA	7
Сигнали з PIR сенсорів	PIR_DATA	7
Освітленість (LDR)	LUX_DATA	7
Стани кнопок керування	BTN_STATE	7
RFID-ідентифікатори	RFID_TAGS	10

Джерело: сформоване автором на основі виконаних досліджень

Результати розв'язання

На основі аналізу вхідної інформації система генерує набір рішень, які реалізуються у вигляді керуючих команд до відповідних виконавчих пристроїв. Це дозволяє забезпечити інтелектуальне та адаптивне управління середовищем у будинку без постійного втручання користувача.

Одним із основних напрямів є керування освітленням. У випадку фіксації руху в малоосвітленому приміщенні, система автоматично вмикає освітлення в цій зоні. Таке рішення не тільки підвищує зручність, але й сприяє економії енергії, оскільки освітлення вмикається лише за необхідності. Ще одним важливим результатом обробки даних є регулювання клімату. При виявленні перевищення допустимих значень температури або вологості система активує вентиляцію чи інші охолоджувальні засоби. Це дозволяє підтримувати комфортні умови для перебування мешканців.

У випадках виявлення руху в охоронюваній зоні, коли не було ідентифіковано RFID-мітку зареєстрованого користувача, система інтерпретує це як потенційну загрозу. У відповідь вмикається тривожна сигналізація, і відповідне повідомлення надсилається на дисплей або в мобільний застосунок через платформу Node-RED. Для зручності користувачів інформація про стан системи відображається на OLED-дисплеї або в мобільному застосунку. Це можуть бути повідомлення про активність у приміщеннях, зміни в температурному режимі, активацію охоронного режиму або виявлення надзвичайних ситуацій (наприклад, витік води або задимлення).

Таблиця 2.4 – Перелік масивів результатної інформації

Масив	Ідентифікатор	Макс. кількість записів
Команди освітлення	LED_CTRL	7
Стани охорони	SECURITY_LOG	75
Повідомлення на дисплей	OLED_MSG	10
Вентиляція/охолодження	FAN_STATE	2

Джерело: сформоване автором на основі виконаних досліджень

Математичний опис та алгоритм розв'язання задачі на ПК

У системі «розумного будинку» керування відбувається на основі аналізу даних, отриманих від різноманітних сенсорів і пристроїв керування. Зокрема, ключовим є контроль мікроклімату у приміщеннях, що здійснюється за показниками температури та вологості. Значення, отримані з сенсорів, порівнюються з встановленими порогами комфортності. Якщо температура чи вологість виходять за допустимі межі, система приймає рішення про активацію вентиляції або охолоджуючих пристроїв. Таким чином, створюються умови, які відповідають заданому рівню комфорту для користувачів.

Крім клімату, система постійно стежить за рівнем освітлення у різних зонах будинку. Для цього використовуються датчики освітленості (LDR) та датчики руху (PIR). Якщо виявлено присутність людини в недостатньо освітленій зоні, система автоматично вмикає відповідне освітлення. У випадках, коли освітленість достатня або відсутній рух, освітлення вимикається з метою енергозбереження.

Також важливою складовою є підсистема безпеки. Кожен користувач має RFID-мітку, за допомогою якої здійснюється авторизація та керування охоронною сигналізацією. У разі несанкціонованого доступу (наприклад, виявлення руху в охоронюваній зоні без попередньої авторизації) система активує тривожні сигнали, включає звукову сигналізацію та надсилає повідомлення на дисплей або мобільний застосунок.

Усі ці дії реалізуються за допомогою програмного алгоритму, що виконується в режимі реального часу. Алгоритм організовано у вигляді нескінченного циклу, в якому постійно виконується опитування сенсорів і обробка подій. На кожному кроці аналізуються всі доступні вхідні дані — значення з датчиків температури та вологи, інформація про освітлення, рух, натискання кнопок та ідентифікація RFID. Після аналізу даних система формує відповідні керуючі команди. Наприклад, може бути прийнято рішення про вмикання світлодіодного освітлення в певній зоні, запуск вентилятора, оновлення повідомлень на OLED-дисплеї або мобільному інтерфейсі. Усі дії послідовно пов'язані між собою, забезпечуючи комплексну автоматизацію процесів в межах розумного будинку.

Завдяки циклічній структурі та обробці подій за допомогою переривань, система здатна швидко реагувати на зміну умов, працюючи ефективно та без затримок. Такий підхід дозволяє підтримувати комфорт, безпеку та енергоефективність житла без постійного втручання користувача.

2.3 Моделювання інформаційної системи

З метою реалізації моделювання інформаційної підсистеми розумного будинку було обрано об'єктно-орієнтований підхід, який дозволяє ефективно відобразити взаємодію користувача із системою, а також логіку її внутрішніх процесів. Для побудови моделей поведінки використовувалося програмне середовище Enterprise Architect, що підтримує нотації UML.

Діаграма прецедентів [12] (Рис. 2.10) — одна з основних діаграм в UML, яка показує взаємозв'язки між акторами (користувачами) і функціональністю системи. На основі функціональних вимог було побудовано діаграму прецедентів для інформаційної підсистеми розумного будинку, яка ілюструє можливі дії користувача: керувати освітленням, пристроями, переглядати клімат і стан безпеки, отримувати сповіщення, а також проходити ідентифікацію через RFID для доступу до функцій. Гість має лише можливість ідентифікуватися через RFID для обмеженого доступу керування освітленням та перегляду даних клімат системи.

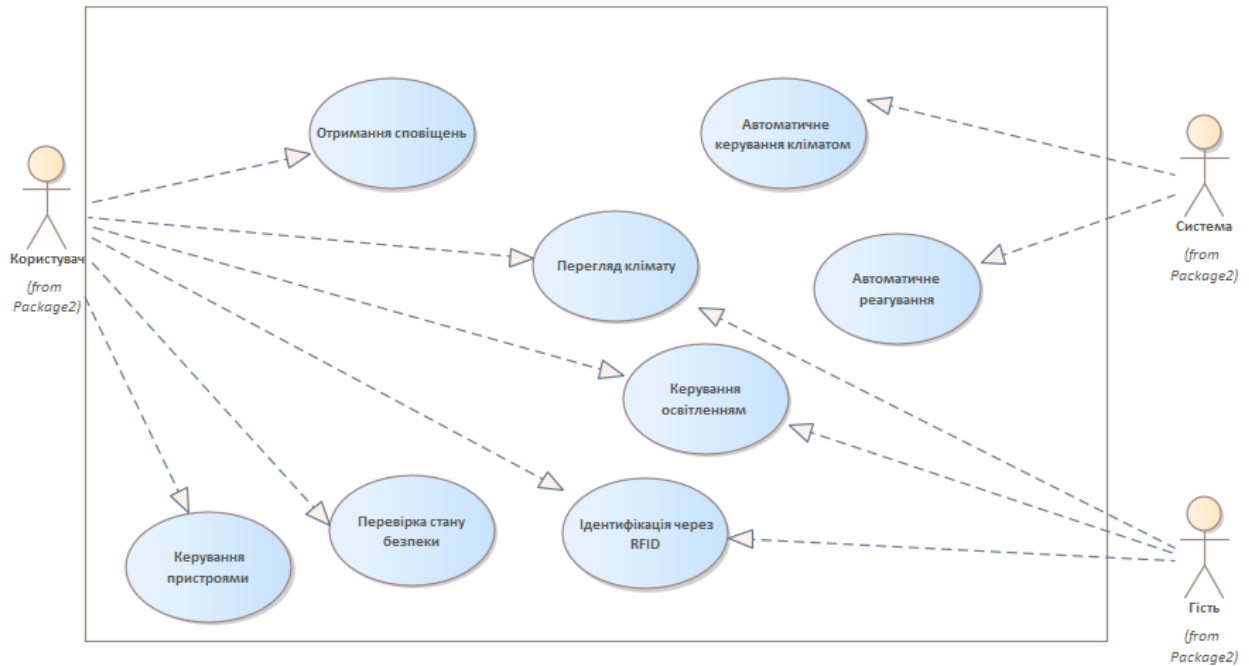


Рис. 2.10 – Діаграма прецедентів

Джерело: сформоване автором на основі виконаних досліджень

На основі прецедентів виконано деталізацію сценаріїв взаємодії у вигляді діаграм послідовності [13], які відображають часову послідовність обміну повідомленнями між компонентами системи. Це дозволяє детально описати внутрішню логіку функціональних процесів системи.

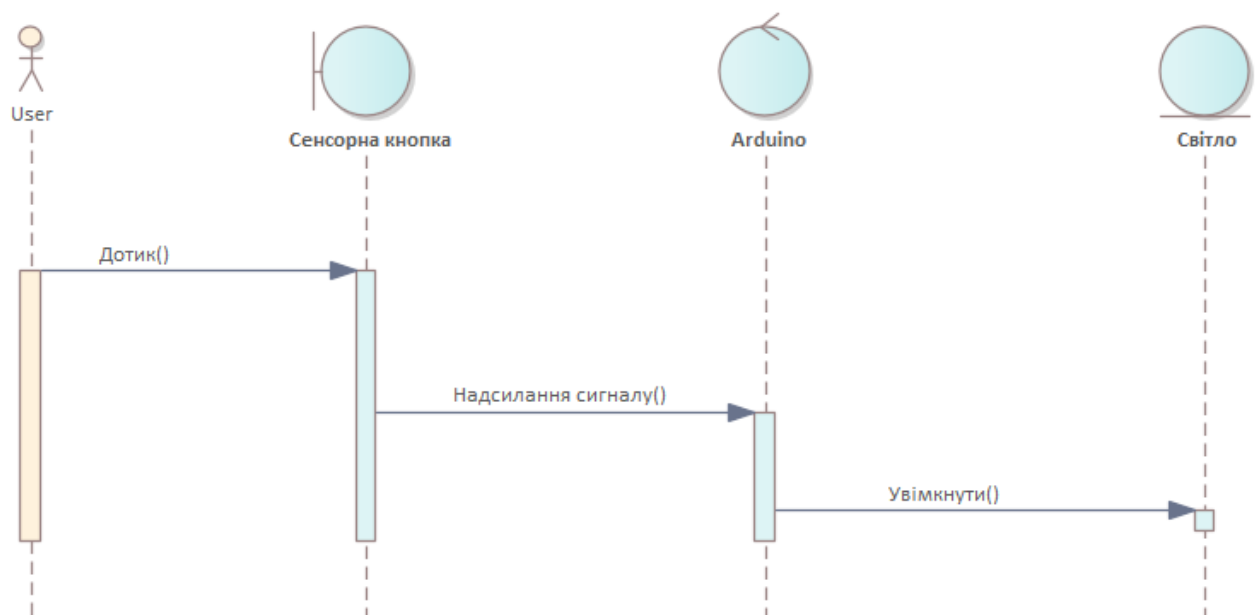


Рис. 2.11 - Увімкнення світла сенсорною кнопкою

Джерело: сформоване автором на основі виконаних досліджень

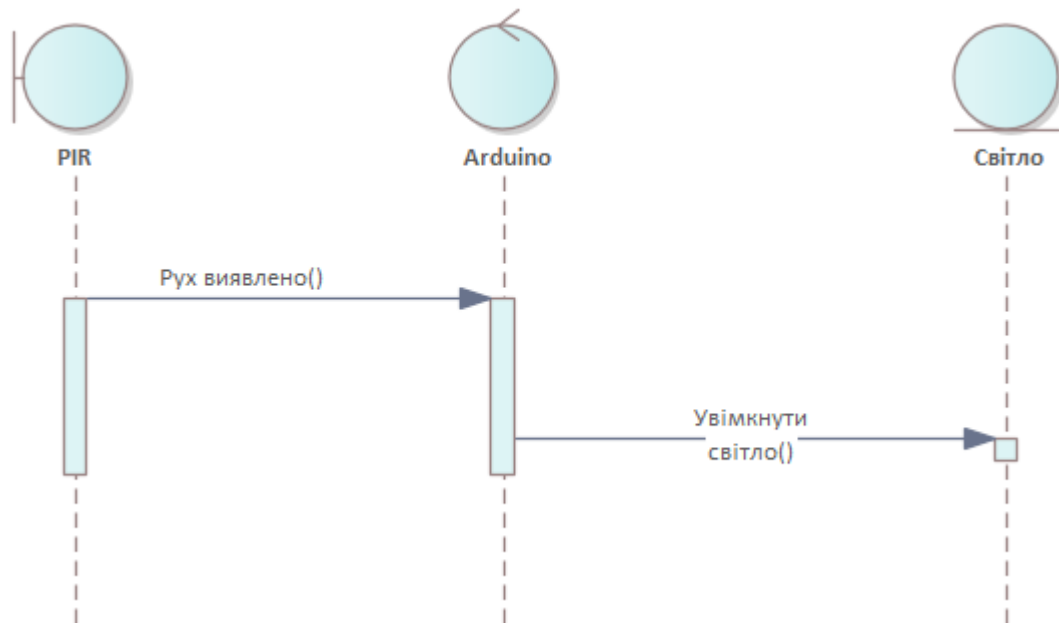


Рис. 2.11 - Автоматичне вмикання світла через датчик руху

Джерело: сформоване автором на основі виконаних досліджень

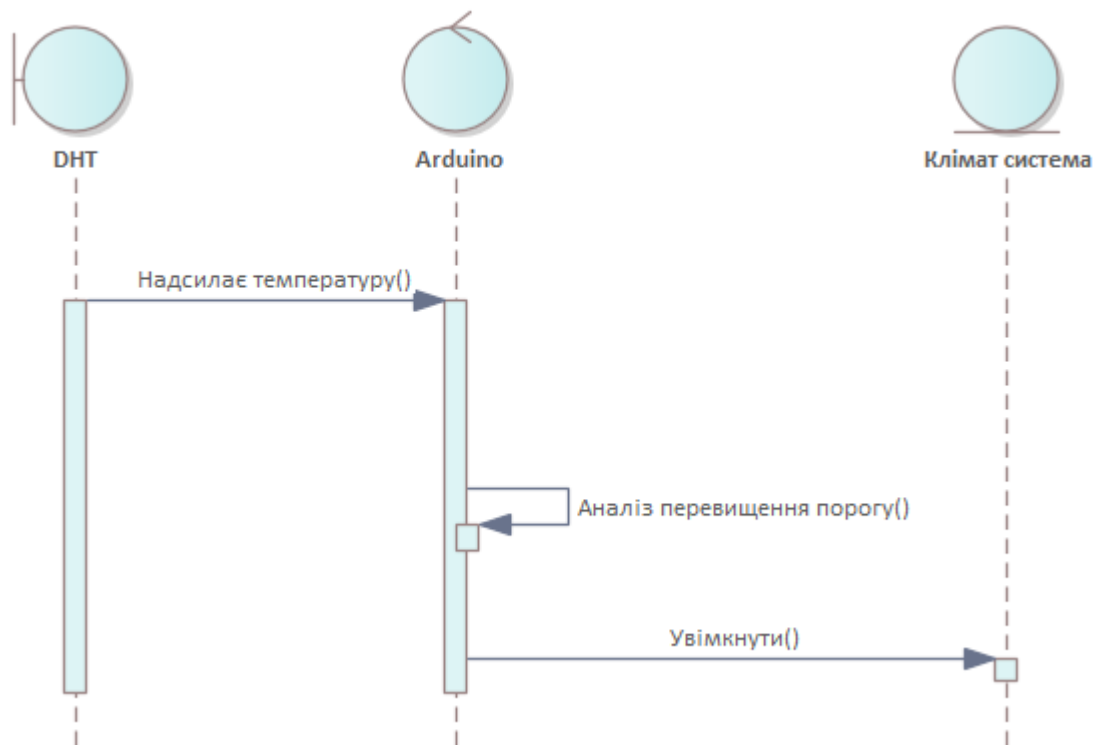


Рис. 2.11 - Виявлення високої температури та вмикання вентилятора

Джерело: сформоване автором на основі виконаних досліджень

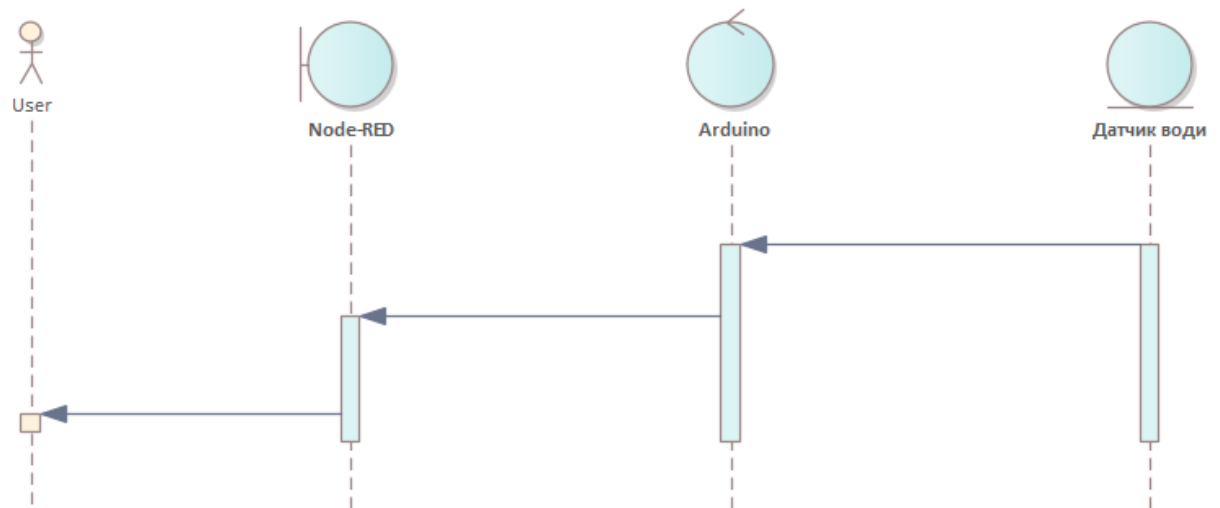


Рис. 2.11 - Детекція води та надсилання сповіщення через Node-RED

Джерело: сформоване автором на основі виконаних досліджень

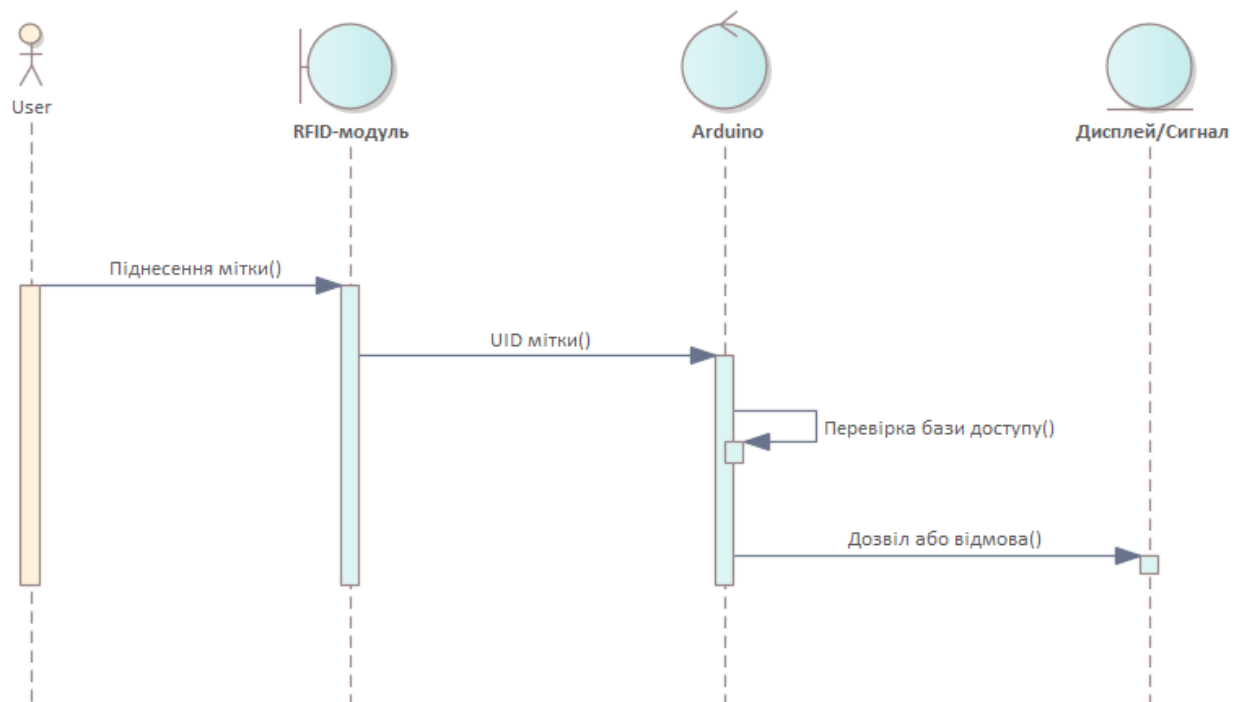


Рис. 2.11 - Розпізнавання RFID мітки для входу

Джерело: сформоване автором на основі виконаних досліджень

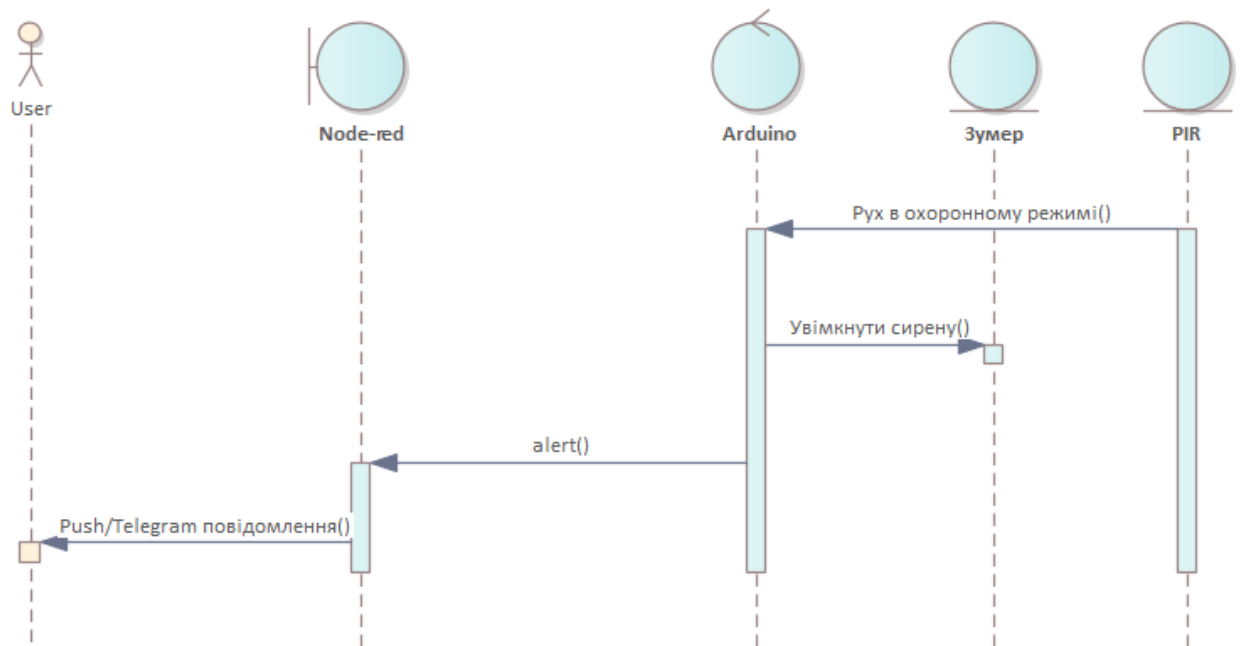


Рис. 2.11 - Тривога при виявленні вторгнення спрацьовує зумер та Node-RED alert

Джерело: сформоване автором на основі виконаних досліджень

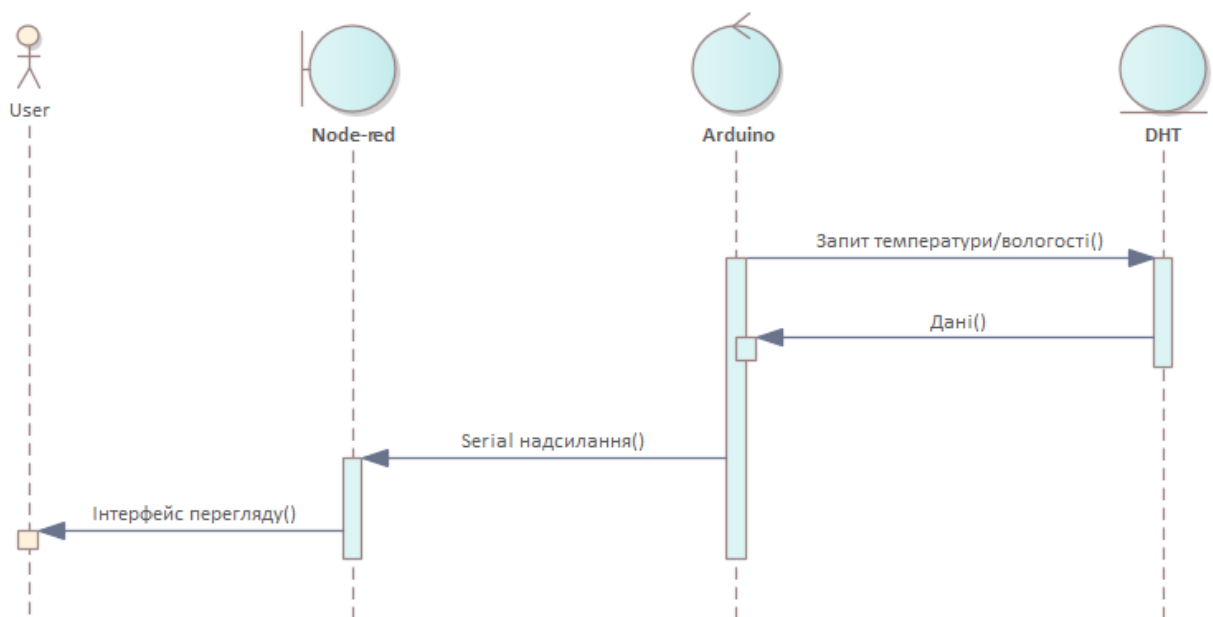


Рис. 2.11 - Перегляд температури/вологості через Node-RED

Джерело: сформоване автором на основі виконаних досліджень

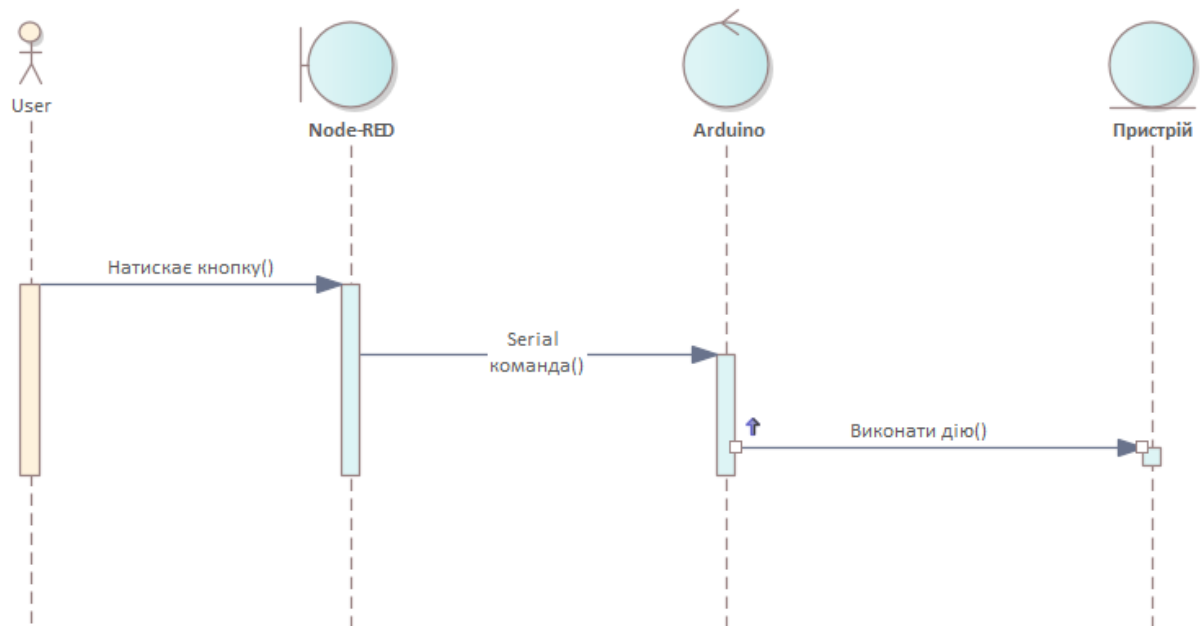


Рис. 2.11 - Ручне керування пристроєм через інтерфейс Node-RED

Джерело: сформоване автором на основі виконаних досліджень

Побудовані діаграми дозволяють чітко сформулювати уявлення про функціональні сценарії системи, їхню реалізацію, а також допомагають при розробці програмної логіки та інтерфейсів взаємодії користувача з системою розумного будинку.

РОЗДІЛ 3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

3.1 Інформаційне забезпечення

Інформаційне забезпечення — це фундамент будь-якої сучасної системи «Розумний будинок». Це комплекс усіх вхідних, проміжних та вихідних даних, які система постійно використовує для моніторингу та керування станом об'єкта в режимі реального часу. Його ефективність напряду залежить від того, наскільки добре продумана структура та наповнення цих даних, адже саме вони дозволяють системі блискавично реагувати на найменші зміни, запускати складні автоматичні сценарії та забезпечувати неперевершену взаємодію з користувачем, роблячи життя комфортнішим та безпечнішим.

До вхідних даних відносяться показники з багатьох датчиків: температури, вологості, руху, відкриття дверей/вікон, диму, газу, протікання води, освітленості. До них також належать прямі команди від користувача через мобільний додаток чи веб-застосунок, а також задані ним налаштування та сценарії. Не варто забувати й про зовнішні дані, такі як прогноз погоди, що допомагають системі оптимізувати свою роботу. Проміжні дані — це результат обробки вхідних: поточний стан пристроїв, виявлені події (наприклад, «рух виявлено у вітальні»), статистика та історія роботи, а також результати аналізу, що дозволяють системі «розуміти» ситуацію та приймати рішення. Вихідні дані — це команди для пристроїв (увімкнути світло, відрегулювати температуру) та сповіщення для користувача (push-повідомлення, звукові сигнали). Цей безперервний потік інформації є життєво важливим для здатності «Розумного будинку» адаптуватися, навчатися та діяти.

Саме бездоганна організація інформаційного забезпечення дозволяє вашому дому не просто існувати, а «думати» та «діяти». Це дозволяє йому миттєво реагувати на будь-яку подію – від спрацювання датчика диму до зміни освітленості – та виконувати запрограмовані сценарії, які роблять ваше повсякденне життя легшим та автоматизованим.

3.1.1 Загальна характеристика інформаційного забезпечення (ІЗ);

Інформаційне забезпечення «Розумного будинку» є досить унікальним, адже воно повністю зосереджене на рівні вбудованих мікроконтролерів. Це означає, що розробники свідомо відмовились від використання зовнішніх баз даних, а вся інформація обробляється та зберігається лише тимчасово в оперативній пам'яті мікроконтролера. Взаємодія з користувачем реалізується виключно через локальні засоби відображення, такі як OLED-дисплей, а також за допомогою звукової та світлової індикації та фізичних інтерфейсів, зокрема сенсорних кнопок. Така архітектура, хоч і накладає певні обмеження на можливості довготривалого зберігання та глибокого аналізу даних, забезпечує максимальну автономність та надзвичайну швидкість реакції системи на події.

До складу інформаційного забезпечення входять три ключові категорії даних. Перша – це первинна інформація, яка представляє собою необроблені показники, що надходять безпосередньо з різноманітних сенсорів, що контролюють довкілля та загальний стан будинку. Це включає дані з датчиків температури та вологості (DHT11/22), сигнали з PIR-датчиків руху, показники рівня освітленості від LDR-датчиків, інформацію про наявність витоків води чи газу, а також стани сенсорних кнопок та дані з RFID-карт, що використовуються для ідентифікації. Ці необроблені дані є елементами з яких система формує своє розуміння оточуючого середовища.

Друга категорія — оброблена інформація. Це дані, які вже пройшли етап первинної обробки безпосередньо мікроконтролером. Сюди відносяться стани логічних прапорів, що керують поведінкою системи (наприклад, `autoMode`, `alarmState`), рішення про вмикання або вимикання виконавчих пристроїв, таких як LED-стрічки для освітлення, кулери для охолодження, зумери для оповіщення, а також формування та виведення інформації на OLED-дисплей у зручному для користувача вигляді — тексту або графічних іконок. Цей рівень даних є критично важливим для внутрішньої системи та функціонування.

Нарешті, інформація для користувача охоплює всі дані, що безпосередньо відображають взаємодію користувача з системою. Це включає налаштування режимів керування, наприклад, перемикання між ручним та автоматичним режимами роботи, вибір бажаного режиму освітлення за допомогою сенсорних кнопок, а також дані авторизації за допомогою RFID-карток для керування системою сигналізації. Оскільки система функціонує в автономному режимі, і дані не зберігаються довготривало, вся ця інформація є актуальною виключно протягом поточного сеансу роботи, що спрощує реалізацію та підвищує швидкодію.

Цей архітектурний підхід, де інформаційне забезпечення повністю базується на поточних показниках сенсорів та станах елементів керування, оброблюваних в реальному часі безпосередньо мікроконтролером, має свої чіткі переваги та обмеження. З одного боку, він значно спрощує реалізацію системи та підвищує її швидкодію, оскільки усуває потребу у складному менеджменті баз даних та мінімізує будь-які мережеві затримки. З іншого боку, така модель обмежує можливості довготривалого зберігання історії подій, виконання глибокої аналітики за минулими періодами, а також використання складних алгоритмів машинного навчання, які зазвичай потребують великих обсягів історичних даних для навчання та оптимізації. Таким чином, ця система «Розумний будинок» є прикладом ефективного компромісу, де миттєва реакція та повна автономність стоять на першому місці.

3.1.2 Організація збору і передання первинної інформації;

У системі «Розумний будинок», що розглядається, збір первинної інформації є критично важливим етапом і здійснюється завдяки розгалуженій мережі датчиків, безпосередньо підключених до мікроконтролера, такого як Arduino Mega. Ці сенсори, діючи як чутливі органи системи, забезпечують безперервний моніторинг фізичних параметрів середовища та стану об'єкта в реальному часі. Важливо підкреслити, що вся подальша обробка та прийняття рішень відбувається виключно локально, що повністю виключає залучення будь-яких зовнішніх серверів чи баз даних.

Джерела первинної інформації надзвичайно різноманітні, кожен датчик виконує свою специфічну, але не менш важливу функцію. Датчики температури та вологості (DHT11/DHT22) [14] [15] використовуються для точного кліматичного контролю в окремих зонах приміщення, забезпечуючи комфортні умови. Датчики руху (PIR)[16] фіксують присутність людини, що може автоматично активувати освітлення або перевести систему в охоронний режим. Датчики освітленості (LDR)[17] визначають рівень природного світла, дозволяючи системі автоматично вмикати освітлення в сутінках або регулювати його інтенсивність. Сенсорні кнопки надають користувачеві можливість ручного керування, наприклад, LED-стрічками або перемиканням режимів роботи системи. RFID-модуль[18] [19] слугує для ідентифікації користувача та керування режимом охорони, додаючи рівень безпеки. Нарешті, газові (MQ-2[20], MQ-7[21]) та водяні датчики призначені для своєчасного виявлення потенційно аварійних ситуацій, таких як витік газу чи води.

Збір інформації відбувається за різними принципами: або періодично, або подієво. Наприклад, датчики DHT11/22 опитуються з чітко визначеною регулярністю, кожну секунду, що дозволяє постійно відстежувати зміни клімату. Натомість, PIR-датчики, які реагують на рух, ініціюють обробку даних лише при зміні свого стану. Це реалізовано за допомогою апаратних переривань або постійного опитування в головному циклі програми. Сенсорні кнопки зчитуються, з використанням безперервного опитування або спеціалізованих бібліотек, що дозволяє уникнути затримок. Аналогові датчики, такі як LDR, а також газові та водяні датчики, зчитують свої значення за допомогою функції `analogRead()`, перетворюючи фізичні величини на цифрові сигнали.

Передача первинної інформації у цій системі реалізується виключно всередині програми мікроконтролера. Зчитані дані з сенсорів передаються у вигляді змінних до відповідних функцій обробки, які відповідають за логіку керування світлом, охоронною системою або іншими підсистемами. Результати цієї обробки, такі як поточні значення температури, вологості або поточний статус системи, оперативно відображаються на OLED-дисплеї [22] у зручному для користувача вигляді, часто з використанням тексту або іконок.

На основі аналізу отриманих та оброблених даних, мікроконтролер активує відповідні вихідні піни, які, у свою чергу, керують виконавчими пристроями. Це можуть бути LED-стрічки для регулювання освітлення, кулери для контролю температури, зумери для звукових сповіщень тощо. Важливою та відмінною особливістю цієї архітектури є те, що, оскільки система є повністю локальною та автономною, жодні зовнішні протоколи передачі даних — як-от MQTT, Wi-Fi або Bluetooth — не використовуються. Вся інформація циркулює виключно в межах одного мікроконтролера, що значно спрощує архітектуру.

Така організація інформаційного забезпечення надає системі високу швидкодію та мінімальні затримки, а також повну відсутність залежності від зовнішніх сервісів.

3.1.3 Побудова системи класифікації та кодування

У розробленій системі «Розумний дім» побудова системи класифікації та кодування інформації є надзвичайно важливим, по суті, фундаментальним етапом. Цей процес дозволяє упорядкувати всі об'єкти керування, що, своєю чергою, значно спрощує подальшу обробку даних та їхнє ефективне виведення на інтерфейс користувача. Це особливо актуально для вбудованих систем, де ресурси мікроконтролера є обмеженими. У таких умовах, ефективність кожного біта кодування напряму впливає на загальну стабільність, швидкодію та надійність усієї системи, перетворюючи її з набору розрізнених елементів на злагоджений механізм.

Для забезпечення логічної структуризації та легкої навігації, всі об'єкти системи класифікуються за їхнім функціональним призначенням. Сюди входять датчики DHT11/22 (температура/вологість), PIR (рух), LDR (освітленість), MQ-2 та MQ-7 (газ), датчики води та RFID-модулі. Далі керуючі елементи, представлені фізичними інтерфейсами, такими як сенсорні кнопки, що дозволяють користувачеві взаємодіяти із системою.

Виділяються виконавчі пристрої які виконують команди, отримані від мікроконтролера. До них належать LED-стрічки для освітлення, зумери для

звукової сигналізації та вентилятори для кліматичного контролю. Крім того, в класифікації присутній інтерфейс користувача, що охоплює OLED-дисплей та світлодіоди для візуальної комунікації системи з користувачем, надаючи оперативну інформацію. І, нарешті, програмні стани/режими, які відображаються внутрішніми логічними змінними, такими як `AutoMode` (автоматичний режим роботи) або `AlarmState` (стан тривоги системи безпеки).

Для однозначної ідентифікації об'єктів у програмному коді системи розумного будинку та підвищення його читабельності застосовується системне кодування з чіткими префіксами. Наприклад, для сенсорів температури і вологості використовуються масиви `dhtPins[]`, `temperature[]` і `humidity[]`, де індексація (`[0]...[6]`) відповідає кожній зоні. Кнопки керування освітленням мають префікс `touchPins[]`, а відповідні світлодіодні стрічки — `ledPins[]`. Для датчиків руху застосовується масив `pirPins[]`, а для керування автоматичними режимами — змінні `autoMode[]`. Стани зон зберігаються у `ledState[]`, яскравість — у `brightness[]`, а часи останніх натискань — у `lastTouchTime[]`. Для вентиляторів використовуються змінні `fanKitchen` (на кухні) та `fanOutdoor` (зовнішній). Числові значення (наприклад, температурні пороги) мають префікс `val_` або зберігаються у відповідних змінних, таких як `climateStatus`, `fanStatus` або `coolingThreshold`. Така структура забезпечує логічну організацію коду, спрощує доступ до однотипних елементів через масиви та підвищує масштабованість і підтримуваність проєкту.

Умовні коди станів у системі реалізуються через прості бінарні значення — 0 для «вимкнено/неактивно» та 1 для «увімкнено/активно». Такий підхід є надзвичайно ефективним з погляду економії оперативної пам'яті. Для внутрішнього представлення станів та подій використовуються булеві змінні або типи `uint8_t`, що забезпечує мінімальне навантаження на ресурси контролера. Наприклад, змінна `alarmActive` (типу `bool`) визначає активність сигналізації, `autoMode[i]` — стан автоматичного режиму в зоні, `fanStatus` — роботу вентилятора, а `ledState[i]` — стан LED-стрічки в кожній зоні. Використання типів `bool` та масивів прискорює обробку логічних умов і дозволяє зберігати десятки станів у лічених

байтах пам'яті, що є критично важливим для мікроконтролерних систем, де ресурси обмежені.

3.1.4 Проектування форм первинних документів, машинограм

У системі «Розумний будинок», орієнтованій на автономну роботу в реальному часі без централізованої бази даних, традиційні поняття первинних документів у класичному розумінні замінюються динамічними цифровими даними. Ці дані формуються безпосередньо в процесі збору та обробки інформації, що надходить від підключених сенсорів та керуючих елементів. Відсутність паперових чи комплексних електронних документів значно спрощує архітектуру та підвищує загальну швидкодію системи. Однак, для ефективних налаштувань, швидкої діагностики та зрозумілого відображення поточного стану системи розробляються спеціалізовані машинограми. Ці стандартизовані форми виводу інформації виступають як ключовий інтерфейс користувача, забезпечуючи зв'язок між системою та її власником.

Форми первинних цифрових даних у цій системі становлять фундаментальну основу для всіх подальших операцій. До них належать:

- Дані сенсорів: це набір цифрових значень, які генеруються в момент опитування датчиків температури, вологості, руху, освітленості, а також свідчення про витoki газу чи води. Ці дані є вхідною інформацією, що відображає поточний фізичний стан довкілля та об'єкта.
- Стани керуючих пристроїв: представлені логічними значеннями (наприклад, true/false або 0/1), що відображають поточний статус ввімкнення/вимкнення освітлення, сигналізації або вентиляції. Це дозволяє системі розуміти статус своїх власних компонентів.
- Команди користувача: це сигнали, що надходять від фізичних сенсорних кнопок або RFID-карт, які ініціюють керування певними функціями системи, такими як активація режиму охорони або зміна налаштувань освітлення.

- Всі ці дані, у формі структурованих змінних та масивів, тимчасово зберігаються в оперативній пам'яті мікроконтролера, формуючи основу для негайної обробки та виводу.
- Машинограми відіграють ключову роль у візуалізації та звуковій індикації стану системи, перетворюючи внутрішні цифрові дані на зрозумілий для людини формат.

Основний візуальний елемент – це OLED-дисплей з роздільною здатністю 128×64 пікселів, який відображає комплексні візуальні машинограми. На цьому дисплеї користувач може бачити:

- Поточні показники температури та вологості для кожної контрольованої зони.
- Іконки, що візуалізують кліматичні режими (наприклад, сніжинка для охолодження, сонце для обігріву, крапля для вологості).
- Перемикання між екранами зон
- Активність вентиляції.

Додатковою формою є звукова машинограма, де звукові сигнали різної тривалості та частоти кодують специфічні події. Наприклад, короткий сигнал може свідчити про успішну авторизацію RFID-карткою, тоді як більш тривала, переривчаста сирена однозначно сигналізуватиме про тривогу. Світлова машинограма використовує світлодіодні індикатори різних кольорів, які сигнализують про режим роботи системи (наприклад, зелений для нормального режиму, червоний для тривоги).

Призначення та основні функції цих машинограм полягають у забезпеченні зрозумілого та оперативного представлення інформації про стан будинку для користувача в реальному часі. Вони не лише візуалізують дані, але й дають можливість швидко діагностувати роботу системи та негайно реагувати на будь-які аварійні події. Крім того, ці візуальні та звукові сигнали сприяють автоматизації керування, зменшуючи потребу у постійному ручному моніторингу з боку користувача, що значно підвищує зручність використання системи.

Особливість реалізації цих машинограм полягає в тому, що, оскільки система не передбачає централізованого зберігання даних, всі вони розроблені виключно програмними засобами мікроконтролера. Це включає застосування спеціалізованих графічних бібліотек (наприклад, U8g2) для ефективного відображення на OLED-дисплеї, а також пряме керування пінами мікроконтролера для активації світлових і звукових індикаторів. Такий підхід підкреслює автономний та локальний характер системи, забезпечуючи її високу швидкодію та незалежність від зовнішніх мереж.

Незважаючи на автономність локальної системи, для забезпечення розширеного моніторингу, детальної аналітики та більш гнучкого керування може бути реалізована інтеграція з Node-RED. У цьому контексті дашборд Node-RED виступає як потужна віртуальна машинограма, що значно доповнює локальний OLED-дисплей. Завдяки Node-RED, дані з мікроконтролера через послідовний порт можуть бути передані на сервер Node-RED.

На дашборді Node-RED користувач отримає:

- Розширену візуалізацію: Більші графіки зміни температури та вологості за тривалі періоди, інтерактивні схеми будинку з відображенням стану всіх датчиків та пристроїв.
- Коротку історію подій: Логи спрацювання датчиків, зміни режимів, аварійні сповіщення з датами та часом (Збереження 5 останніх подій).
- Зручні елементи керування: Веб-інтерфейс дозволить легко перемикати режими, активувати/деактивувати сигналізацію, керувати освітленням та вентиляцією з будь-якого пристрою, підключеного до мережі.

Таким чином, у той час як OLED-дисплей забезпечує миттєвий та локальний доступ до ключової інформації, дашборд Node-RED розширює можливості моніторингу та керування, надаючи глибший рівень аналітики та гнучкості

взаємодії. Це створює багатоваріантну систему інформування, адаптовану під різні сценарії використання.

3.1.5 Структура інформаційних масивів

У системі «Розумний будинок», де пріоритетом є автономна робота в реальному часі без централізованої бази даних, для ефективного зберігання та обробки інформації використовується чітка структура інформаційних масивів. Ці масиви компактно та впорядковано містять усі поточні значення сенсорів, стан виконавчих пристроїв та керуючі параметри. Такий підхід значно полегшує доступ, обробку та оперативне оновлення інформації, що є критично важливим для швидкодії системи.

Типи інформаційних масивів поділяються на дві основні категорії (масиви значень датчиків та масиви керуючих станів). Перші відповідають за зберігання сирих показників з різних сенсорів, що моніторять довкілля та стан будинку. До них входять:

- `int temperature[7]`; для значень температури по 7 зонах будинку.
- `int humidity[7]`; для показників вологості по тих же 7 зонах.
- `int lightLevel[7]`; для рівня освітленості з LDR-датчиків.
- `bool motionDetected[7]`; для станів датчиків руху (PIR)
- `bool waterLeakDetected[2]`; для датчиків витoku води (наприклад, на кухні та в ванній кімнаті).
- `bool gasLeakDetected[2]`; для газових датчиків (MQ-2, MQ-7 які використовуються на кухні).

Другі в свою чергу відображають поточний статус виконавчих пристроїв та елементів керування.

До яких належать:

- `bool ledState[7]`; для стану ввімкнення/вимкнення LED-стрічок у 7 зонах.
- `int ledBrightness[7]`; для рівня яскравості цих LED-стрічок.
- `bool buttonState[7]`; для стану сенсорних кнопок.

- `bool alarmActive`; для загального стану сигналізації системи.

Організація та індексація цих масивів реалізована у вигляді одновимірних структур, де кожен індекс відповідає конкретній зоні або типу пристрою. Наприклад, індекси від 0 до 6 можуть відповідати таким функціональним зонам будинку, як Коридор, Вітальня, Кухня, Ванна кімната, Дитяча, Спальня. Для датчиків витоку газу та води індексація може відповідати окремим локаціям, таким як кухня та ванна кімната, що забезпечує чітке просторове прив'язування даних. Такий підхід робить структуру даних інтуїтивно зрозумілою та легкою для навігації в коді. Призначення масивів є багатогранним. По-перше, вони слугують для зберігання поточних значень для оперативного доступу та швидкої обробки, забезпечуючи узгодженість даних між різними частинами програми. По-друге, масиви значно спрощують циклічну обробку даних, дозволяючи, наприклад, у циклі опитувати всі датчики температури або оновлювати стан освітлення по зонах через їхні індекси. Це зменшує дублювання коду та підвищує його ефективність. По-третє, така структура підвищує масштабованість системи, оскільки легко дозволяє додавати нові зони або типи пристроїв шляхом простого розширення відповідних масивів.

Ця ефективна структура інформаційних масивів є ключовим елементом, що забезпечує високу швидкодію, гнучкість та організованість роботи «Розумного будинку». Вона дозволяє системі функціонувати ефективно та надійно, без необхідності використання складних зовнішніх баз даних, що є вагомою перевагою для автономних мікроконтролерних рішень.

3.2 Технічне забезпечення

3.2.1 Загальні положення та схема автоматизації

Система «Розумного будинку» являє собою інтегровану апаратно-програмну платформу, що забезпечує комплексний моніторинг стану довкілля в приміщенні, оперативне виявлення загроз, ефективне керування освітленням та клімат-контролем, а також надійну систему безпеки. Вся функціональність побудована на

основі сенсорних даних та попередньо запрограмованих автоматичних сценаріїв. Ця система базується на мікроконтролері Arduino Mega 2560 [23], до якого підключено широкий спектр периферійних сенсорів для збору даних та виконавчих пристроїв для реалізації команд. Ключовою особливістю є розподілена логіка обробки сигналів та автоматизації, що реалізована як у прошивці мікроконтролера, так і в середовищі Node-RED. Останнє забезпечує гнучкий інтерфейс користувача, можливості віддаленого керування, візуалізацію поточних станів, довгострокове зберігання даних та, що найважливіше, дозволяє гнучко програмувати складні події та сценарії без безпосередньої зміни коду на Arduino.

Вся інформація, що надається та обробляється системою, чітко структурована та поділяється на дві основні категорії, представлені у вигляді інформаційних масивів для оптимізації доступу та обробки:

Масиви сенсорних значень відповідають за компактне та впорядковане зберігання даних, отриманих від різноманітних сенсорів у режимі реального часу. Ця частина системи, що надає актуальну інформацію про навколишнє середовище:

- `int temperature[7]`; Значення температури по зонах
- `int humidity[7]`; Значення вологості по зонах
- `int lightLevel[7]`; Рівень освітленості (LDR)
- `bool motionDetected[7]`; Стан датчиків руху (PIR)
- `bool waterLeakDetected[2]`; Стан датчиків витоку води
- `bool gasLeakDetected[2]`; Стан газових датчиків

Ці дані регулярно зчитуються в головному циклі `loop()` прошивки Arduino та використовуються не лише для прийняття миттєвих рішень всередині мікроконтролера, але й для ефективної передачі до середовища Node-RED для віддаленого моніторингу та аналізу.

Далі масиви керуючих станів та змінні відповідають за збереження поточного стану елементів, якими керує система, відображаючи «рішення» та команди системи:

`bool ledState[7]`; Стан LED-стрічок

int ledBrightness[7]; Яскравість LED-стрічок

bool buttonState[7]; Стан сенсорних кнопок

bool alarmActive; // Стан сигналізації

Загальна логіка автоматизації система використовує декілька продуманих сценаріїв автоматизації для забезпечення комфорту та безпеки:

- Ручне керування освітленням: Забезпечується через сенсорні кнопки в кожній зоні. Одноразове натискання вмикає/вимикає світло, довге натискання дозволяє регулювати яскравість, а подвійне — перемикає в автоматичний режим.
- Автоматичне вмикання світла: Реалізується при виявленні руху, але лише за умови, що рівень освітленості (виміряний LDR-датчиком) є нижчим за визначений поріг, що оптимізує енергоспоживання та обов'язково при увімкненому авто режимі освітлення.
- Клімат-контроль: Включає запуск вентиляції або кулерів на основі аналізу показників температури та вологості з відповідних зон. Також має як ручне управління пристроями так і атоматичне.
- Оповіщення про аварійні ситуації: У разі виявлення витoku газу або води, система негайно активує зумери (звукова сигналізація), щоб сповістити про небезпеку. В системі передбачено різне звучання сигналу зумеру для спрацювання датчиків води та газу.
- Система сигналізації з RFID-доступом: Активується за допомогою RFID-ключа. У захищеному режимі система постійно моніторить рух у контрольованих зонах, і у разі виявлення несанкціонованого руху негайно запускає тривогу.

Node-RED [24] відіграє центральну роль у забезпеченні розширеного функціоналу системи, діючи як хаб для моніторингу, керування та аналітики. Завдяки цій інтеграції реалізовані наступні можливості:

- Веб-інтерфейс: Створюється за допомогою функції dashboard Node-RED, що дозволяє користувачу зручно переглядати поточні значення всіх датчиків з будь-якого пристрою, підключеного до мережі.
- Дистанційне керування: Освітленням та вентиляцією можна керувати віддалено через можливості Node-RED.
- Мобільні сповіщення: Система може надсилати оперативні сповіщення на телефон користувача про важливі події, такі як виявлення руху, спрацювання тривоги або витік.
- Гнучке налаштування логіки: Найважливіше, Node-RED дозволяє налаштовувати пороги спрацювання, змінювати логіку автоматизації та створювати нові сценарії без необхідності перепрошивки або зміни коду на Arduino, що значно спрощує підтримку та розширення функціоналу.

Архітектурно, будинок поділений на 7 контрольованих зон, кожна з яких має власні сенсори (температури, вологості, руху, освітленості), а також виконавчі пристрої (LED-стрічка) та елементи керування (сенсорна кнопка). В певних зонах таких як кухня та ванна кімната встановлено додаткові датчики. На кухні є потреба встановлення датчику протікання води датчиків газу (MQ-2 та MQ-7), а також окремої вентиляції яка працює окремо від основної клімат системи. У той час у ванній кімнаті було вирішено застосувати лише датчик протікання води. Управління освітленням реалізується як локально (через фізичні кнопки для швидкого доступу), так і автоматично (на основі комбінації виявлення руху та рівня освітленості). Система клімат-контролю автоматично активує вентиляцію або кулери на основі поточних значень температури та вологості з масивів `temperature[]` та `humidity[]`. Система безпеки, що активується через RFID-ключ, ретельно моніторить рух і в разі виявлення несанкціонованого проникнення або витіку води/газу негайно запускає сигналізацію.

У цій комплексній схемі Node-RED виступає як центральний пункт логіки, моніторингу та керування. Він не лише забезпечує безшовну інтеграцію з мобільними додатками, відправку сповіщень та ведення історії даних, але й

дозволяє створювати складніші сценарії автоматизації, які виходять за межі можливостей самого мікроконтролера. Це дозволяє системі «Розумного будинку» бути не тільки автономною та швидкою, але й гнучкою, масштабованою та зручною у використанні.

3.2.2 Структура комплексу технічних засобів.

Комплекс технічних засобів для системи «Розумний будинок» розроблений як багатофункціональна інтегрована платформа, що забезпечує комплексне управління умовами середовища, безпекою, освітленням та автоматизацією житлових зон. Ця система ефективно поєднує можливості автономного мікроконтролера Arduino Mega 2560 з гнучкістю та потужністю візуального програмування, моніторингу та віддаленого керування через Node-RED.

Контролерна Частина

Arduino Mega 2560 (рис. 3.1) є центральним мозком системи, що забезпечує збір даних, виконання локальної логіки та керування виконавчими пристроями. Його вибір обумовлений наявністю великої кількості цифрових (понад 50) та аналогових (16) входів/виходів, що дозволяє безпосередньо підключити десятки сенсорів та виконавчих пристроїв без використання додаткових шин або розширювачів пінів. Це спрощує схему підключення та підвищує швидкодію системи за рахунок прямого доступу до периферії.

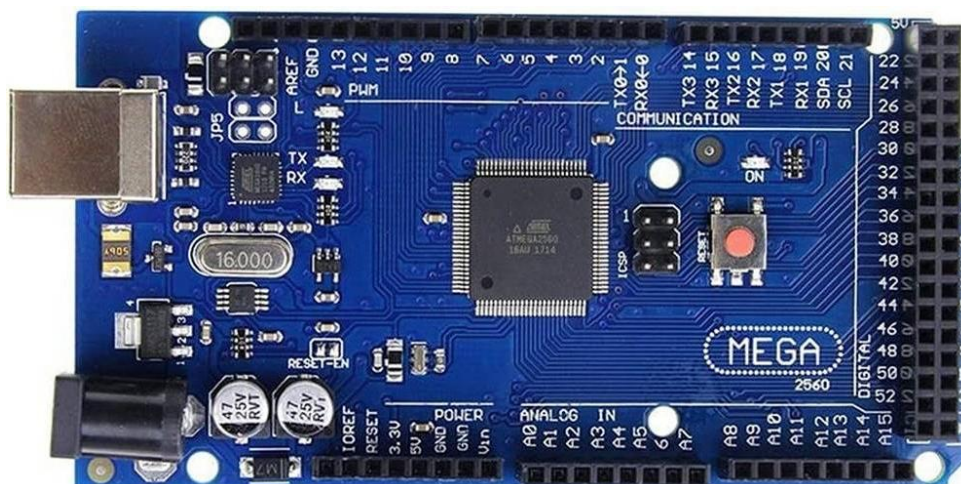


Рис. 3.1 - Arduino Mega 2560

Периферійні Сенсори

Система включає розгалужену мережу датчиків, що контролюють ключові параметри мікроклімату, безпеку та активність у приміщеннях. Всі сенсори підключаються безпосередньо до пінів Arduino Mega 2560, а їхні значення зберігаються в спеціально організованих масивах для ефективної обробки.

У «Розумного будинку» мікроклімат контролюється за допомогою цифрових датчиками DHT11/DHT22 (рис. 3.2). Ці датчики, підключені до Arduino Mega 2560. Вибір зроблено на користь DHT11 через їхню доступну ціну, що дозволяє економічно охопити більшість зон будинку, тоді як для зовнішнього датчика обрано DHT22, оскільки він має більші можливості роботи в ширшому діапазоні температур, від -40°C до $+80^{\circ}\text{C}$. А для внутрішніх зон такий діапазон не потрібен. Ці датчики надсилають дані температури та вологості, які зберігаються у масивах `temperature[7]` та `humidity[7]`. Інформація передається в Node-RED для детальної візуалізації та віддаленого моніторингу. Ключове застосування даних – автоматичне керування вентиляцією: при перевищенні заданих порогів температури чи вологості система автоматично активує кулери або зовнішні вентилятори, підтримуючи комфортні умови в приміщенні.



Рис. 3.2 – Датчики DHT 11 та DHT 22

Для адаптивного керування освітленням у «Розумного будинку» використовуються фоторезистори (LDR, рис. 3.3), які є недорогими та простими датчиками рівня природного світла. Кожен LDR, підключений до аналогового входу Arduino Mega 2560, змінює свій опір залежно від інтенсивності світла, дозволяючи мікроконтролеру зчитувати поточну освітленість у 7 зонах. Ці дані зберігаються в масиві `lightLevel[7]` і є ключовими для автоматизації: система може автоматично вмикати світло (LED-стрічки через MOSFET-драйвери) у сутінках або



Рис. 3.4 - Датчики руху HC-SR501

Для забезпечення критично важливої безпеки у «Розумного будинку» інтегровані газові датчики MQ-2 (рис. 3.5) та MQ-7 (рис. 3.6), по одному типу у стратегічно важливих зонах, таких як кухня. MQ-2 виявляє легкозаймисті гази та дим, тоді як MQ-7 спеціалізується на чадному газі (CO). Обидва датчики підключаються до аналогових входів Arduino Mega 2560, що дозволяє моніторити рівень концентрації газу, а не лише факт його присутності. Отримана інформація зберігається у масиві `gasDetected[2]`. У разі перевищення безпечного порогу концентрації система негайно активує зумери та світлодіодні індикатори, забезпечуючи швидке оповіщення про небезпеку та дозволяючи оперативно вжити заходів.

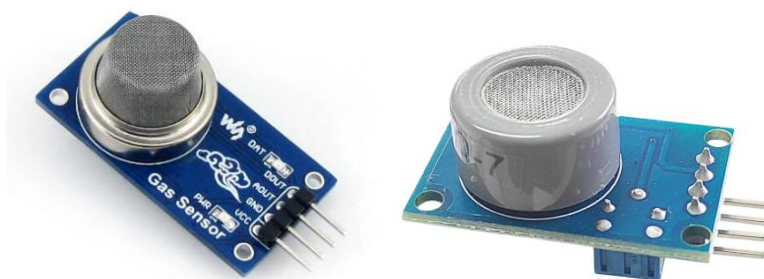


Рис. 3.5 – 3.6 – Датчики MQ-2 та MQ-7

Для запобігання збиткам від протікань у «Розумного будинку» інтегровані датчики витоку води (рис. 3.7), розміщені у важливих зонах, як-от кухня та ванна кімната. Ці сенсори працюють за принципом виявлення провідності води: коли вода потрапляє на їхні контакти, вони змінюють електричний сигнал. Датчики

підключаються до цифрових пінів Arduino Mega 2560, а отримана інформація негайно зберігається у булевому масиві `waterLeakDetected[2]`. Логіка реагування є пріоритетною: при виявленні витоку система миттєво активує зумери та може надсилати сповіщення на мобільний телефон через Node-RED, забезпечуючи швидке оповіщення та можливість ужити заходів.



Рис. 3.7 - Датчики витоку води

Категорія	Сенсори	Кількість	Піни
Температура та вологість	DHT11, DHT22	7	22-28
Освітлення	Фоторезистори (LDR)	7	A0-A6
Рух	HC-SR501	7	14-19, 2
Газ	MQ-2, MQ-7	2	A7, A8
Витік води	Вологозалежні датчики	2	40, 41

Джерело: сформоване автором на основі виконаних досліджень

Виконавча Частина

Для гнучкого та ефективного керування освітленням у семи зонах «Розумного будинку» використовуються LED-стрічки 2835 (рис. 3.8), які забезпечують як функціональне, так і декоративне освітлення. Оскільки LED-

стрічки зазвичай потребують більшого струму, ніж може безпосередньо видати мікроконтролер Arduino Mega 2560, для їхнього керування застосовуються MOSFET-драйвери. MOSFET функціонує як електронний перемикач, дозволяючи Arduino керувати потужним навантаженням (LED-стрічкою) за допомогою слабого керуючого сигналу зі своїх цифрових пінів. Важливою особливістю такого підключення є можливість реалізації ШІМ-регулювання яскравості. Підключення MOSFET-драйверів до ШІМ-пінів Arduino дозволяє контролювати яскравість світла, змінюючи тривалість імпульсів. Стан увімкнення/вимкнення та бажаний рівень яскравості для кожної з семи зон зберігаються у відповідних масивах LED_PINS [7] та brightness[7]. Це дозволяє системі не тільки вмикати/вимикати світло, але й плавно регулювати його інтенсивність відповідно до автоматичних сценаріїв (наприклад, за даними LDR-датчиків чи датчиків руху) або команд користувача з сенсорних кнопок чи Node-RED.

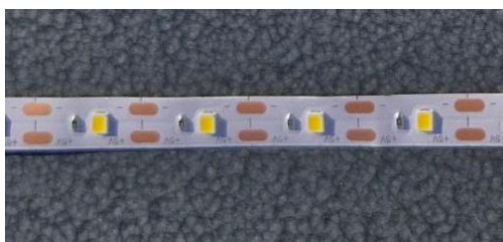


Рис. 3.8 – Led стрічка 2835

Для інтуїтивного керування освітленням у «Розумному будинку» використовуються сенсорні кнопки (рис. 3.9), інтегровані в кожній із семи зон. Ці кнопки, підключені до цифрових входів Arduino Mega 2560, реагують на дотик і зберігають свій стан у масиві TOUCH_PINS[7]. Логіка обробки натискань багатофункціональна: одиночне натискання вмикає/вимикає світло, утримання кнопки дозволяє регулювати яскравість через ШІМ-регулювання, а подвійне натискання перемикає освітлення в автоматичний режим. Таким чином, сенсорні кнопки забезпечують зручний ручний контроль, гармонійно доповнюючи автоматизовані сценарії.

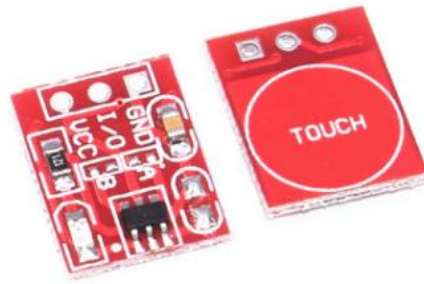


Рис. 3.9 – Сенсорні кнопки

Сенсори	Кількість	Піни
LED-стрічки	7	32-38
Сенсорні кнопки	7	3, 5, 6, 9, 10, 11, 12

Кліматичне Управління

Система «Розумний будинок» інтелектуально керує мікрокліматом, використовуючи дані з датчиків температури (`temperature[]`) та вологості (`humidity[]`). Для цього зовнішній кулер (рис. 3.10) автоматично активуються, коли показники температури або вологості перевищують встановлені пороги забезпечуючи ефективну циркуляцію повітря та охолодження приміщень. Також ми маємо окремий кулер (рис. 3.11) для кухні який має пороги нижчі ніж основний тож вмикається ще раніше та може допомагати з підтриманням температури та вологості наприклад під час приготування. Ключовим елементом візуалізації кліматичних даних є OLED-дисплей 0.96(рис. 3.12) (128x64 пікселів), підключений через I2C. На ньому користувач бачить поточні показники температури, вологості та іконки кліматичних режимів (сніжинка, сонце, крапля), що забезпечує інтуїтивне розуміння стану системи та її реакції на зміни середовища.

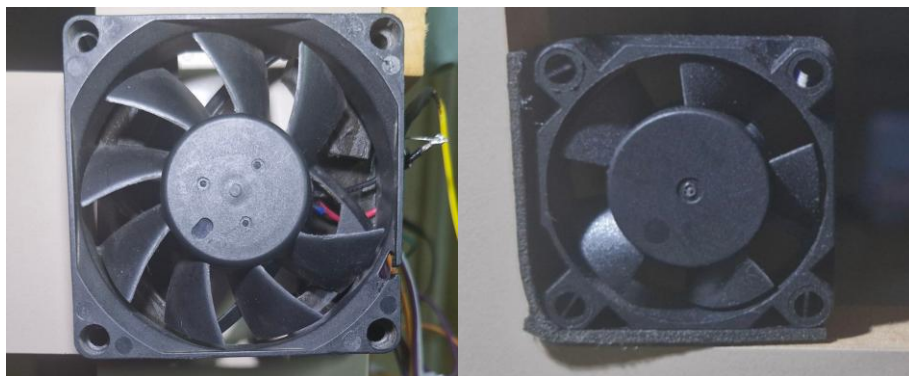


Рис. 3.10 – 3.11 - Зовнішній кулер та кухоний кулер



Рис. 3.12 -3.13 - OLED-дисплей

Сигналізація та Безпека

Система «Розумний будинок» забезпечує надійний захист завдяки комплексу сигналізації, що включає RFID-модуль MFRC522 (рис. 3.14) для авторизації доступу (активації/деактивації охорони через RFID-картки), підключений по SPI. Зумери, приєднані до цифрових пінів Arduino, видають звукові сигнали при спрацюванні тривоги (рух у захищеному режимі або витік газу/води), а також підтверджують успішну авторизацію. Різнокольорові LED-індикатори, також підключені до цифрових пінів, візуально інформують про стан системи (зелений — норма, червоний — тривога). Загальний статус сигналізації зберігається у булевій змінній `alarmActive`, що є центральною для логіки охорони. Цей комплексний підхід забезпечує багаторівневий захист, поєднуючи зручність доступу з ефективним оповіщенням про будь-які потенційні загрози.



Рис. 3.14 - MFRC522

Інтеграція та Керування через Node-RED

Node-RED виступає як верхній рівень керування та інтеграції, значно розширюючи можливості автономної системи «Розумний будинок». Це програмне середовище, що може працювати на окремому пристрої (наприклад, Raspberry Pi чи комп'ютері), забезпечує потужні інструменти для моніторингу та взаємодії. Ключовим елементом є веб-інтерфейс з дашбордом, який надає інтерактивну візуалізацію всіх даних із системи: поточні показники температури, вологості, стану датчиків руху, витоку води/газу, а також статус освітлення, сигналізації та вентиляторів. Користувач може легко дистанційно керувати цими елементами, що забезпечує гнучкість та зручність.

Для ефективного та швидкого обміну даними між Arduino та Node-RED використовується Serial port. Це забезпечує надійну комунікацію. Крім того, Node-RED дозволяє створювати автоматичні правила з розширеною логікою, що виходить за межі можливостей мікроконтролера. Наприклад, можна реалізувати складніші сценарії «розумний сон» де освітлення поступово згасає, а температура знижується. Або побудову графіків зміни температури та вологості за тривалий

період, Така інтеграція перетворює базову автономну систему на повноцінний, легко керований “Розумний будинок”.

Комплекс технічних засобів системи “Розумний будинок” являє собою потужне гібридне рішення, що поєднує автономність мікроконтролера Arduino Mega 2560 із широкими інтеграційними можливостями Node-RED. Ця архітектура, яка включає розгалужену мережу датчиків для моніторингу (температури, вологості, руху, газу, води) та виконавчих пристроїв для керування (LED-стрічки, вентилятори, зумери), забезпечує повну автоматизацію семи зон будинку за ключовими параметрами безпеки, мікроклімату та освітлення. В результаті користувач отримує не лише надійну та швидку систему для повсякденних потреб, але й масштабовану та адаптовану платформу, що може легко розширюватися та змінюватися відповідно до його зростаючих потреб та нових технологій, створюючи справді комфортний і безпечний житловий простір.

3.2.3 Опис автоматизованого робочого місця

Автоматизоване Робоче Місце (АРМ) у системі «Розумний будинок» є центральним інтерфейсом для моніторингу, налаштування та керування інтелектуальними підсистемами в реальному часі. Воно поєднує апаратну частину, програмну логіку на Arduino Mega 2560, інтуїтивний інтерфейс користувача на Node-RED, а також набір сенсорів і виконавчих пристроїв. Таке поєднання дозволяє користувачеві мати повний контроль над мікрокліматом, освітленням, безпекою та автоматизацією всіх зон будинку, забезпечуючи зручність та ефективність.

Апаратне забезпечення АРМ:

- Мікроконтролер: Arduino Mega 2560 – забезпечує зчитування даних з датчиків та керування виконавчими елементами (LED, зумери, вентилятори).

- Комунікація: Передача даних між Arduino та ПК/сервером здійснюється через USB.
- OLED-дисплей: 0.96” I2C-дисплей для локального виводу температури, вологості та стану вентиляції.
- Сенсори: включають температурні та вологісні датчики (DHT11, DHT22), PIR-датчики руху, датчики освітлення (LDR), сенсори витoku води, газові датчики (MQ-2, MQ-7).
- Кнопки: Сенсорні панелі для ручного ввімкнення освітлення.
- LED-стрічки: Використовуються як освітлення в семи зонах з можливістю зміни яскравості.
- Зумери та вентилятори: Використовуються для оповіщення та клімат-контролю.

На персональному комп’ютері або сервері розгорнуто Node-RED — платформу для візуального програмування логіки автоматизації.

Вона виконує функції:

- Збору та обробки даних з Arduino.
- Відображення стану всіх сенсорів і виконавчих пристроїв у графічному інтерфейсі.
- Керування обладнанням (ввімкнення освітлення, активація охорони, запуск вентиляторів тощо).
- Встановлення порогів спрацювання для автоматичних режимів.
- Надсилення повідомлень про тривожні події (виявлення газу, води, руху при активній сигналізації) в Node-RED або через SMS.

Функціональні можливості АРМ:

- Перегляд температури, вологості, освітлення, руху, витоків газу та води по зонах.

- Керування LED-освітленням вручну або автоматично за заданими умовами.
- Активація/деактивація сигналізації через RFID або інтерфейс Node-RED.
- Налаштування реакцій на тривожні події (звук, світло, повідомлення).
- Автоматичне охолодження приміщення при підвищеній температурі/вологості.
- Історія подій і станів системи для аналізу (Останні 5).

Таким чином, АРМ забезпечує централізовану, зручну та гнучку взаємодію користувача з усіма підсистемами розумного будинку, поєднуючи локальне управління та віддалений доступ через сучасний веб-інтерфейс.

3.3 Програмне забезпечення

3.3.1 Структура програмного забезпечення

Програмне забезпечення системи «Розумний будинок» складається з двох основних рівнів:

- Вбудоване ПЗ мікроконтролера Arduino
- Серверне ПЗ з інтерфейсом управління на базі Node-RED



Рис. 3.15 - Структура програмного забезпечення інформаційної системи «Розумний будинок»

Базове (Системне) Програмне Забезпечення

Базове або системне програмне забезпечення створює необхідне середовище для функціонування всієї системи «Розумний будинок» та забезпечує робоче середовище для прикладних програм. Сюди входить Arduino IDE— системне середовище, яке дозволяє компілювати та виконувати прошивку безпосередньо на мікроконтролері Arduino Mega 2560. Також до цього рівня належить Node-RED — середовище виконання серверної частини, яке може бути розгорнуте на персональному комп'ютері або міні-ПК типу Raspberry Pi. Операційна система, така як Windows або Linux, є обов'язковою для хостингу Node-RED, забезпечуючи стабільну роботу всіх програмних компонентів.

Прикладне Програмне Забезпечення

Прикладне програмне забезпечення призначене для реалізації конкретних завдань користувача, забезпечуючи функціональність «Розумного будинку». Воно поділяється на дві основні частини: вбудоване ПЗ на Arduino та серверне ПЗ на Node-RED.

Вбудоване ПЗ (Arduino)

Вбудоване програмне забезпечення, написане на мові C/C++ та скомпільоване через середовище Arduino IDE, виконує безпосередній збір даних із сенсорів, первинну обробку сигналів та керування виконавчими пристроями. Воно включає наступні функціональні модулі:

- Модуль зчитування датчиків відповідає за періодичне опитування та отримання значень з температурних/вологісних датчиків (DHT), фоторезисторів (LDR), PIR-сенсорів руху, а також датчиків витoku води та газу (MQ-2, MQ-7). Отримані результати систематично зберігаються у відповідних масивах: `temperature[]`, `humidity[]`, `lightLevel[]`, `motionDetected[]`, `gasLeakDetected[]`, та `waterLeakDetected[]`.
- Логіка керування освітленням контролює LED-стрічки на основі сигналів від сенсорних кнопок (для вмикання/вимкнення та зміни яскравості), а також з урахуванням даних про рух та рівень освітленості. Поточні стани освітлення зберігаються у змінних `ledState[]` та `brightness[]`.
- Клімат-контроль автоматично вмикає або вимикає вентилятори залежно від поточних показників температури та вологості, підтримуючи комфортні умови в приміщенні.
- Охоронна система реалізує моніторинг руху, газу та води, виконує авторизацію за допомогою RFID-модуля та забезпечує активацію/деактивацію охорони.
- Комунікаційний модуль відповідає за серійний зв'язок із Node-RED через USB (Serial), забезпечуючи передачу масивів даних та прийом керуючих команд.
- Інтерфейс локального виводу відображає на OLED-дисплеї ключову інформацію, таку як температура, вологість, а також статус охорони, вентиляції та освітлення.

Серверне ПЗ (Node-RED)

Серверне програмне забезпечення Node-RED використовує візуальний редактор потоків для реалізації складних сценаріїв та взаємодії з користувачем. Його функціонал включає:

- Збір і візуалізація даних: Node-RED приймає телеметрію від Arduino та відображає її у зручному веб-інтерфейсі у вигляді інтерактивних графіків, таблиць та індикаторів, надаючи користувачеві повний огляд стану будинку в реальному часі.
- Налаштування автоматизації: Цей модуль дозволяє гнучко керувати сценаріями автоматизації, встановлювати пороги температури/вологості, визначати умови за часом або подіями, а також налаштовувати реакції на тривоги.
- Управління пристроями: Node-RED забезпечує передачу команд на Arduino для дистанційного керування пристроями, такими як увімкнення/вимкнення світла, регулювання яскравості, активація охорони та керування вентиляторами.
- Оповіщення користувача: У випадку аварійних ситуацій (витік газу, витік води, виявлення руху при активній охороні), Node-RED автоматично надсилає сповіщення користувачеві та відображає їх у веб-інтерфейсі, забезпечуючи оперативне інформування.

3.3.2 Системне Програмне Забезпечення Системи «Розумний Будинок»

Для забезпечення повноцінного функціонування автоматизованої системи «Розумний будинок» було ретельно обґрунтовано вибір необхідного системного програмного забезпечення. Цей вибір забезпечує сумісну роботу апаратної частини, розробленого прикладного ПЗ, а також ефективну підтримку мережевої взаємодії, керування пристроями та зручний користувацький інтерфейс. Система базується на використанні декількох операційних систем: Linux (зокрема, Raspberry Pi OS / Ubuntu Server) як надійної та безкоштовної серверної платформи для розгортання Node-RED, SMS. MS Windows 10/11 використовується на комп'ютері розробника для налагодження Arduino-скетчів в Arduino IDE та адміністрування Node-RED. Крім того, Android необхідний на смартфонах

користувачів для взаємодії з SMS-повідомленнями та доступу до веб-інтерфейсу через мобільний браузер.

Для розробки, проектування та налагодження системи застосовувались спеціалізовані RAD-засоби: Arduino IDE – для створення та завантаження прошивки в мікроконтролер Arduino Mega 2560; Node-RED – як ключовий інструмент візуального програмування для побудови логіки автоматизації та веб-інтерфейсу (Dashboard). Для візуалізації електронних схем та системних діаграм використовувався Enterprise Architect. Система активно застосовує сучасні веб-технології: HTML, CSS для інтерфейсу Node-RED Dashboard та JavaScript для його логіки, а також Twilio для SMS-сповіщень.

3.3.3 Прикладне програмне забезпечення

У складі автоматизованої системи «Розумний будинок» прикладне програмне забезпечення відіграє ключову роль у забезпеченні взаємодії користувача з апаратною частиною системи. Його основне призначення — забезпечити доступ до керування, моніторингу та налаштувань системи у зручній формі.

Прикладне ПЗ системи належить до предметно-орієнтованого (керування розумним будинком), ПЗ для роботи глобальних мереж, а також частково до ПЗ загального призначення (Node-RED).

Система «Розумний будинок» надає користувачеві широкий спектр можливостей для взаємодії через інтуїтивно зрозумілий інтерфейс, реалізований переважно за допомогою Node-RED Dashboard.

- Моніторинг у реальному часі: Користувач може миттєво переглядати актуальні показники з усіх сенсорів будинку. Це включає цифрові значення, графіки або індикатори для температури, вологості, рівня освітленості, виявлення руху, наявності газу та витоку води. Ця функція забезпечує повну обізнаність про поточний стан середовища.

- Віддалене керування пристроями: Інтерфейс дозволяє здійснювати повний контроль над виконавчими елементами з будь-якого місця. Користувач може вручну вмикати/вимикати світло в окремих зонах, змінювати яскравість LED-стрічок за допомогою слайдерів, активувати або деактивувати охоронну сигналізацію, а також контролювати роботу системи вентиляції.

- Автоматизація сценаріїв: У Node-RED реалізовані гнучкі логічні правила (flows), що забезпечують розумну реакцію системи на події. Приклади включають автоматичне ввімкнення світла при виявленні руху в темному приміщенні або запуск вентилятора при підвищенні температури вище заданого порогу. Це робить будинок адаптивним до потреб мешканців.

- Оповіщення та сповіщення: У випадках критичних або тривожних ситуацій, таких як виявлення газу, витоку води або несанкціонованого доступу, програмне забезпечення автоматично надсилає сповіщення користувачу через SMS або веб-застосунок Node-RED, забезпечуючи швидку реакцію.

- Інтерфейс візуалізації: Node-RED Dashboard забезпечує просту та адаптивну візуалізацію, доступну як з ПК, так і зі смартфона. Він містить інтерактивні елементи керування (кнопки, слайдери яскравості), інформативні графіки температури та вологості, а також чіткі індикатори стану кожного пристрою, що робить взаємодію з системою інтуїтивно зрозумілою.

- Налаштування користувацьких параметрів: Система надає користувачеві можливість самостійно змінювати ключові параметри. Це включає налаштування порогів спрацювання автоматичних режимів, додавання нових сценаріїв поведінки системи, а також керування доступом до системи за допомогою RFID або PIN-кодів, що забезпечує високу гнучкість та персоналізацію.

Таким чином, прикладне програмне забезпечення в системі «Розумний будинок» виступає не лише інструментом керування, а й аналітичним та захисним

модулем, що дозволяє зробити повсякденне життя комфортнішим, безпечнішим і енергоефективнішим.

У Додатку А подано візуальні приклади інтерфейсу Node-RED Dashboard, що ілюструють основні функціональні можливості прикладного програмного забезпечення системи «Розумний будинок».

3.3.4 Програмна Документація Системи «Розумний Будинок»

Програмна документація до системи «Розумний будинок» охоплює ключові складові, необхідні для повного опису програмного забезпечення, та забезпечує всебічну підтримку як для користувачів, так і для технічного персоналу. Вона гарантує ефективну, стабільну та безпечну роботу інформаційної системи, спрямованої на автоматизацію процесів моніторингу, керування та взаємодії з житловим простором. Система дозволяє контролювати мікроклімат, освітлення, безпеку, а також організовувати зручну комунікацію з мешканцями.

Формуляр Програмного Забезпечення

Основні характеристики програми

- Назва: Програмне забезпечення для автоматизованої системи керування «Розумний будинок».
- Призначення: Надання засобів моніторингу, керування та автоматизації функцій будинку (освітлення, клімату, безпеки тощо).
- Середовище розробки: Node-RED, Arduino IDE, Twilio.
- Інтерфейс користувача: Node-RED Dashboard (веб-інтерфейс, адаптований для ПК і мобільних пристроїв).
- Мови програмування: C++ (Arduino), JSON (для структурування даних), HTML/CSS (інтерфейс).
- Тип ПЗ: Предметно-орієнтоване ПЗ, ПЗ для глобальних мереж, частково ПЗ загального призначення.

Комплектність програми

- Arduino скетчі для керування обладнанням;
- Node-RED flows для логіки автоматизації;
- Додаток Повідомлення для SMS-сповіщень;
- Web Dashboard (Node-RED Dashboard);
- Набір бібліотек (MFRC522, DHT, U8g2, OneWire, DallasTemperature);
- Документація з інсталяції та налаштування;

Відомості про призначення програми

Програмне забезпечення призначене для:

- Керування освітленням у 7 зонах (вручну й автоматично);
- Контролю температури, вологості, руху, витоків води й газу;
- Оповіщення користувача про критичні події через SMS;
- Ведення журналу станів сенсорів у реальному часі;
- Налаштування сценаріїв автоматизації (flows);
- Персоналізації інтерфейсу та функцій відповідно до потреб користувача.

Обмеження щодо мінімальної конфігурації технічних засобів

- Мікроконтролер: Arduino Mega 2560 (через велику кількість периферії).
- Серверна частина: Raspberry Pi, ПК з Linux/Windows, або хмарна платформа для запуску Node-RED.
- Інтерфейс доступу: браузер (Chrome, Firefox), мобільний телефон для отримання SMS.
- Пам'ять: мін. 32 КБ флеш-пам'яті для прошивки, мін. 512 МБ оперативної пам'яті для Node-RED.

Керівництво системного програміста

- Інсталяція Node-RED: встановлення через npm, імпорт flows, запуск як сервіс.
- Підключення MQTT-брокера (за потреби): налаштування з'єднання з пристроями.
- Налаштування SMS: реєстрація на сайт Twilio, створення віртуального номера.
- Прошивка Arduino: компіляція та завантаження скетчів через Arduino IDE.
- Інтеграція сенсорів: підключення за схемою, перевірка зчитування даних у серійному моніторі.
- Налаштування сценаріїв у Node-RED: редагування flows, застосування тригерів та умов.

Опис контрольного прикладу

Для перевірки функціоналу системи «Розумний будинок» було реалізовано сценарій автоматичного вмикання освітлення в кімнаті при виявленні руху в умовах низької освітленості. Очікувана поведінка полягала в тому, що PIR-сенсор фіксує рух, система перевіряє рівень освітлення за допомогою LDR, і якщо він нижчий за порогове значення, LED-стрічка автоматично вмикається. Результатом стало коректне спрацювання: світло увімкнулося в кімнаті, коли було виявлено рух у темряві, а факт події був записаний та відображений на Node-RED Dashboard, підтверджуючи ефективність та надійність реалізованої логіки.

Приклади програмного коду

У Додатку Б наведено приклади програмного коду, розробленого в межах створення системи «Розумний будинок».

3.4 Результати реалізації інформаційної підсистеми

У ході виконання випускної бакалаврської роботи було розроблено інтегровану систему «Розумний будинок», що поєднує в собі функції автоматичного керування освітленням, клімат-контролем, сигналізацією, а також взаємодією з користувачем через OLED-дисплей та сенсорні інтерфейси. Комплекс розробок базується на мікроконтролері Arduino Mega 2560 та використовує модульну архітектуру, яка дозволяє масштабування системи без необхідності повної перебудови логіки.

Основу комплексу розробок становить:

- Уніфікований алгоритм управління освітленням у 7 зонах, що включає ручне керування (сенсорні кнопки), автоматичне керування (датчики руху, освітлення), режими нічного освітлення та функції затримки виключення.
- Інтелектуальний блок клімат-контролю, який, за допомогою шести сенсорів DHT11 та одного DHT22 аналізує температуру й вологу, і автоматично активує вентилятори в кухонній зоні або зовнішню клімат систему. Результати виводяться на OLED-дисплей із відповідними піктограмами (сонце, сніжинка, крапля).
- Інтеграція безпеки за допомогою RFID-модуля та сенсорів газу, води, руху, а також динамічного звукового оповіщення про загрози (зумери).
- Подвійні та довгі натискання на сенсорні кнопки для розширення функціоналу без потреби в додаткових фізичних елементах керування.
- Модульна структура прошивки дозволяє легко додавати або змінювати функціональність (наприклад, додавання голосового керування або мобільного застосунку).

Пілотне тестування відбувалося на стенді на базі Arduino Mega 2560, OLED-дисплея 0.96, RFID-модуля RC522, 7 LED-стрічок, 7 сенсорних кнопок, 7 PIR-датчиків, 6 DHT11 та одного DHT22, сенсорів MQ-2, MQ-7, а також двох модулів виявлення води.



Рис. 3.16 - Готовий фізичний прототип системи «Розумний будинок»

Основні етапи апробації:

- Верифікація режимів освітлення (ручний, автоматичний) у кожній із 7 зон.
- Імітація руху, перепадів освітленості, виявлення води, газу.
- Реакція системи безпеки на неавторизоване проникнення (RFID).
- Перевірка автоматичного запуску вентиляторів при підвищенні температури/вологості.

Результати тестування

- Реакція системи на зміни показників спрацьовувала без затримок — середній час відповіді: 150–350 мс.
- Динамічна OLED-візуалізація дозволяє користувачу миттєво оцінити стан усіх сенсорів.
- Живлення та обробка даних усіх підключених сенсорів були стабільними протягом тривалого періоду (>13 годин роботи).
- Тестування показало високу стабільність алгоритмів управління, а також точність реагування на події.

Потенційна практична цінність

У порівнянні з існуючими комерційними рішеннями (Ajax, Fibaro, Xiaomi Smart Home), запропонована система є:

1. Відкритою для модифікацій — розроблена на відкритих платформах без ліцензійних обмежень.
2. Дешевшою у реалізації — орієнтовна собівартість системи в 5–8 разів нижча.
3. Гнучкою у налаштуванні — користувач може адаптувати систему під власні потреби.

Аналіз та висновки

Система повністю функціональна в умовах тестування, демонструє стабільну роботу. Обрана архітектура та алгоритмічні рішення показали адекватність, повноту реалізації й високу достовірність отриманих результатів. Серед можливих спрощень — відсутність захисту від глушіння RFID-сигналів.

Плани на майбутнє

Інтеграція з мобільним додатком через Wi-Fi (ESP8266/ESP32).

Запровадження алгоритмів машинного навчання для прогнозування температури та автоматичного налаштування режимів.

Використання реле замість LED для керування реальними приладами.

Додавання журналювання подій і зберігання їх у базі даних.

ВИСНОВКИ

У ході виконання дипломної роботи було здійснено комплексну розробку інформаційної системи «Розумний дім», призначеної для автоматизації керування побутовими процесами, значного підвищення рівня комфорту, безпеки та енергоефективності житлових приміщень. Ця система являє собою сучасне рішення, що об'єднує різноманітні технології для створення адаптивного та інтелектуального середовища проживання.

На початковому етапі аналізу предметної галузі було проведено глибоке літературне дослідження сучасних тенденцій використання інтелектуальних систем у побуті. Це включало аналіз науково-технічної літератури, яка охоплювала огляд існуючих комерційних та open-source рішень, а також практичні приклади реалізації аналогічних систем. Такий підхід дав змогу сформулювати чітке та вичерпне уявлення про функціональні потреби потенційних користувачів, врахувати їхні очікування щодо зручності та надійності, а також визначити пріоритетні напрями розробки задля забезпечення актуальності та конкурентоспроможності фінального продукту.

У другому розділі роботи особливу увагу було приділено формуванню вимог до інформаційної системи. На цьому етапі чітко визначено бізнес-вимоги, що відображають цілі та завдання, які система повинна вирішувати для користувачів та власників будинку. Детально сформульовано функціональні вимоги, які описують конкретні дії та можливості системи (наприклад, автоматичне керування освітленням, моніторинг температури, активація сигналізації). Окремо були ідентифіковані нефункціональні вимоги, що стосуються якості системи – її продуктивності, надійності, безпеки, зручності використання та масштабованості. Крім того, розроблено детальний алгоритм функціонування ключових компонентів, що є основою для подальшої реалізації. Проведено всебічне моделювання системи з використанням відповідних UML-діаграм.

Третій розділ присвячено проектуванню та безпосередній реалізації основних компонентів системи. Детально розглянуто інформаційне забезпечення,

включаючи структури даних для зберігання показників сенсорів та станів пристроїв. Описано технічне забезпечення, що охоплює вибір та інтеграцію мікроконтролера Arduino Mega 2560, сукупності різноманітних датчиків (температури, вологості, руху, освітленості, газу, води), виконавчих пристроїв (LED-стрічки, вентилятори, зумери) та модулів зв'язку. Значна увага зверталась на архітектуру програмного забезпечення, його модульну організацію та принципи взаємодії між компонентами. Реалізовано базовий функціонал (Minimum Viable Product – MVP), який включає інтеграцію з усіма запланованими датчиками, системи безпеки, автоматизоване керування освітленням, охолодженням, сигналізацією та іншими елементами системи. Архітектура рішення реалізована за трирівневим принципом (датчики/виконавчі пристрої – мікроконтролер – серверне ПЗ/інтерфейс), що забезпечує високу масштабованість, модульність, гнучкість та зручність у подальшому супроводі та розширенні.

Результати тестування MVP-системи підтвердили її повну працездатність, високу зручність використання для кінцевого користувача та ефективність при виконанні всіх ключових функцій. Проведені тестування показали надійність роботи всіх компонентів та стабільність системи в цілому. Отримана система є міцною та надійною базою для подальшого розширення функціоналу, інтеграції з мобільними додатками, хмарними сервісами та іншими зовнішніми платформами, що відкриває широкі перспективи.

Отже, поставлену мету дипломної роботи успішно досягнуто: в процесі виконання здійснено було комплексну розробку інформаційної системи типу «Розумний дім», яка є конкурентоздатною, повністю відповідає сучасним вимогам до функціональності, надійності, безпеки та зручності користування, надаючи користувачам потужний інструмент для керування своїм житловим простором.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

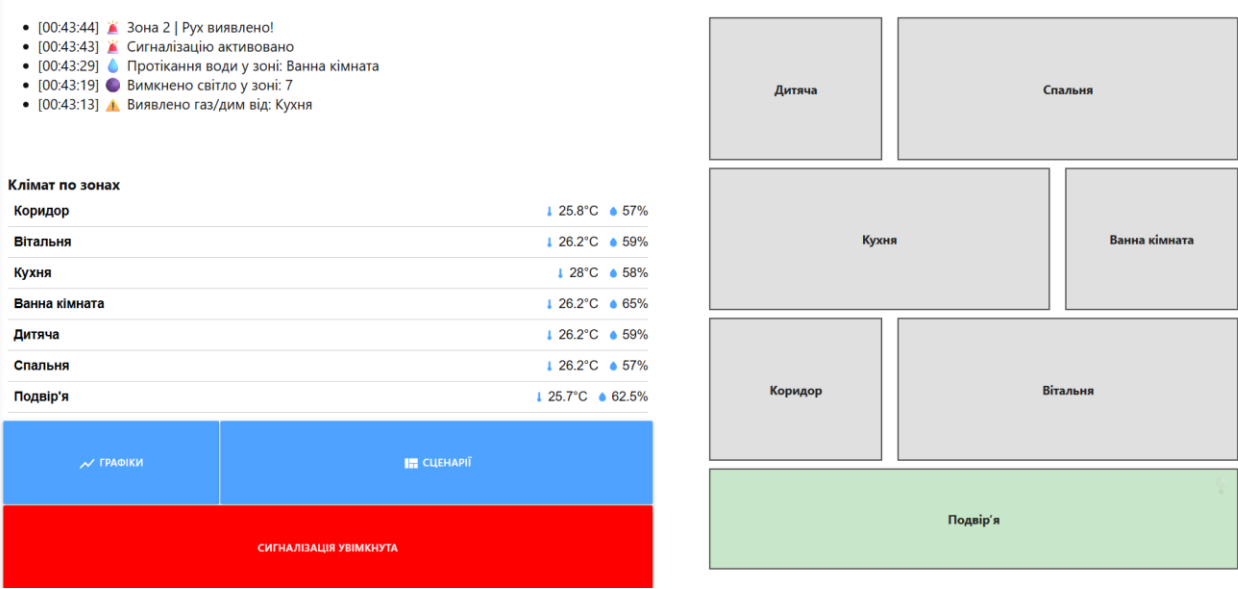
1. Arduino. Arduino Documentation. URL: <https://docs.arduino.cc> (дата звернення: 30.05.2025).
2. Розумне керування мікрокліматом. AW-Therm. URL: <https://aw-therm.com.ua/keruvannya-mikroklimatom-u-primishenni/> (дата звернення: 30.05.2025).
3. Author F., Author S. Title of the article. Journal Name. 2024. Vol. X, No. Y. P. 1–12. URL: <https://doi.org/10.1007/s43926-024-00084-3> (дата звернення: 30.05.2025).
4. Smart Home Monitoring System Using ESP32 Microcontrollers. URL: https://www.researchgate.net/publication/347062859_Smart_Home_Monitoring_System_Using_ESP32_Microcontrollers (дата звернення: 30.05.2025).
5. Internet of Things Based Indoor Smart Surveillance and Monitoring System using Arduino and Raspberry Pi. URL: https://www.researchgate.net/publication/353421410_Internet_of_Things_Based_Indoor_Smart_Surveillance_and_Monitoring_System_using_Arduino_and_Raspberry_Pi (дата звернення: 30.05.2025).
6. Smart Home Monitoring Using IoT Technologies. IntechOpen. URL: https://www.intechopen.com/chapters/74005?utm_source (дата звернення: 30.05.2025).
7. Home Assistant Demo. URL: <https://demo.home-assistant.io/#/lovelace/home> (дата звернення: 30.05.2025).
8. Items – openHAB Documentation. URL: <https://www.openhab.org/docs/configuration/items.html> (дата звернення: 30.05.2025).
9. Domoticz Wiki – Головна сторінка. URL: https://wiki.domoticz.com/Main_Page (дата звернення: 30.05.2025).
10. ISO/IEC/IEEE 29148:2011 – Systems and Software Engineering — Life Cycle Processes — Requirements Engineering. URL: <https://cdn.standards.iteh.ai/samples/45171/41f0a3198f10470fb5014f0e4468daa3/ISO-IEC-IEEE-29148-2011.pdf> (дата звернення: 30.05.2025).
11. ISO/IEC 25000:2014 – Systems and Software Engineering — Systems and Software Quality Requirements and Evaluation (SQuaRE). URL: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:25000:ed-2:v1:en> (дата звернення: 30.05.2025).
12. Use Case Diagram – Enterprise Architect User Guide. Sparx Systems. URL: https://sparxsystems.com/enterprise_architect_user_guide/17.1/modeling_languages/usecasediagram.html (дата звернення: 30.05.2025).

13. Sequence Diagram – Enterprise Architect User Guide. Sparx Systems. URL: https://sparxsystems.com/enterprise_architect_user_guide/17.1/modeling_languages/sequencediagram.html (дата звернення: 30.05.2025).
14. AOSONG. DHT11 Temperature & Humidity Sensor Datasheet. URL: https://www.electronicoscaldas.com/datasheet/DHT11_Aosong.pdf (дата звернення: 30.05.2025).
15. DHT22 – Temperature and Humidity Sensor Datasheet. AllDataSheet. URL: <https://www.alldatasheet.com/datasheet-pdf/pdf/1132459/ETC2/DHT22.html> (дата звернення: 30.05.2025).
16. HC-SR501 – PIR Motion Sensor Datasheet. AllDataSheet. URL: <https://www.alldatasheet.com/html-pdf/1131987/ETC2/HC-SR501/112/1/HC-SR501.html> (дата звернення: 30.05.2025).
17. Photoresistor (LDR) Light Sensor Module Datasheet. Digi-Key. URL: <https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/6649/ST0250%20datasheet.pdf> (дата звернення: 30.05.2025).
18. MFRC522 RFID Reader IC Datasheet. NXP Semiconductors. URL: <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf> (дата звернення: 30.05.2025).
19. MFRC522 RFID Reader Module Datasheet. Elecrow. URL: <https://www.elecrow.com/download/MFRC522%20Datasheet.pdf> (дата звернення: 30.05.2025).
20. MQ-2 Semiconductor Gas Sensor Datasheet. Pololu. URL: <https://www.pololu.com/file/0j309/mq2.pdf> (дата звернення: 30.05.2025).
21. MQ-7 Carbon Monoxide Gas Sensor Datasheet. SparkFun. URL: <https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf> (дата звернення: 30.05.2025).
22. OLED 0.96 inch 128x64 I2C Display Module Datasheet. Waveshare. URL: https://www.waveshare.com/wiki/0.96inch_OLED_Module (дата звернення: 30.05.2025).
23. Arduino Mega 2560 – Мікроконтролер. Mini-Tech. URL: <https://www.mini-tech.com.ua/ua/arduino-mega-2560> (дата звернення: 30.05.2025).
24. Node-RED User Guide. URL: <https://nodered.org/docs/user-guide/> (дата звернення: 30.05.2025).
25. Розумний будинок – показчик літератури. Запорізька обласна універсальна наукова бібліотека. URL: <https://zounb.zp.ua/wp-content/uploads/2021/07/Rozumnij-budinok-pokazhchik-6.04.21-s-oblozhkoj.pdf> (дата звернення: 30.05.2025).
26. Щодо застосування сигналізаторів довибухонебезпечних концентрацій паливних газів і мікроконцентрацій чадного газу в повітрі приміщень житлових будинків та громадських будинків і споруд. Державний комітет України з промислової безпеки, охорони праці та гірничого нагляду. URL: https://online.budstandart.com.ua/catalog/doc-page.html?id_doc=5010 (дата звернення: 30.05.2025).

27. Безпека. Системи: комплексні рішення для охорони та безпеки AJAX. URL: https://alarm.bezpeka.systems/ua/?gad_source=1&gad_campaignid=21880220581&gbraid=0AAAAAoh0QkR7HyM2-UeRpkBj42xSz8eyz&gclid=Cj0KCQjw0qTCBhCmARIsAAj8C4bjKuB5hW-i1tUgZ2HZ51Dqoa2-6DPjdd8mbi0vgs5UbWSt_5tRb6AaAn7rEALw_wcB (дата звернення: 30.05.2025).
28. Обережно чадний газ: інформація про легкість газу і ризики отруєння. Державна служба України з надзвичайних ситуацій. URL: <https://dsns.gov.ua/news/ostanni-novini/15766> (дата звернення: 30.05.2025).
29. Системи керування розумним будинком від компанії Fibaro. URL: <https://www.fibaro.com/uk/> (дата звернення: 30.05.2025).

ДОДАТКИ

Додаток А



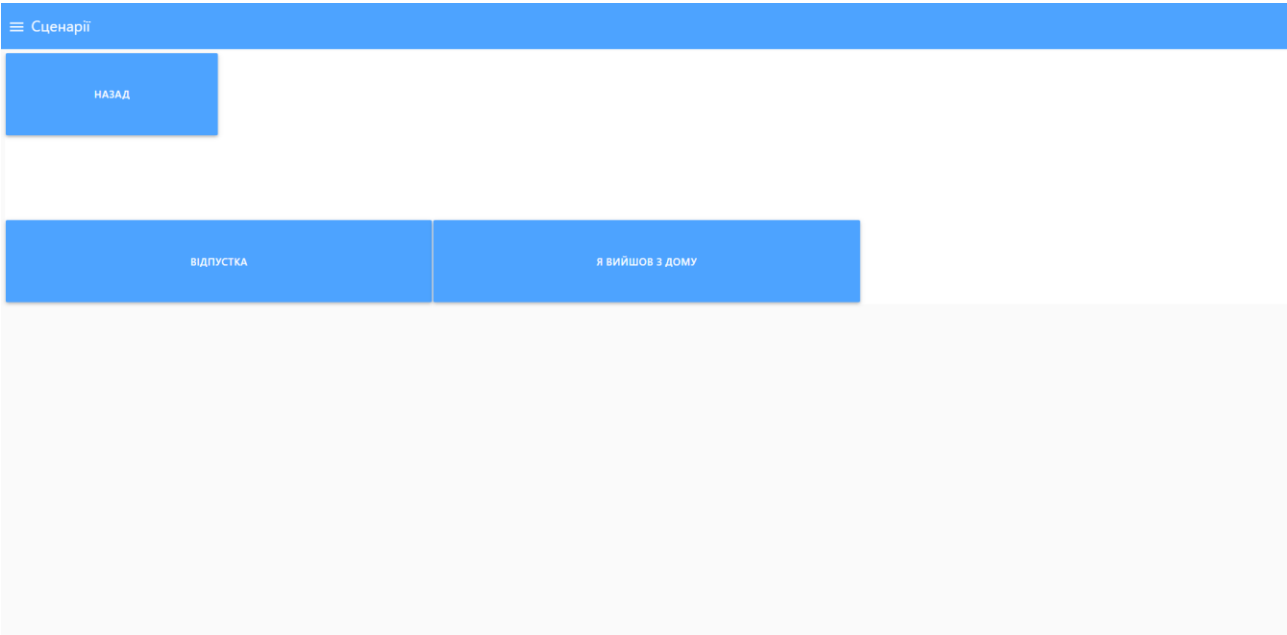


Рис. А3 Сторінка вибору сценаріїв

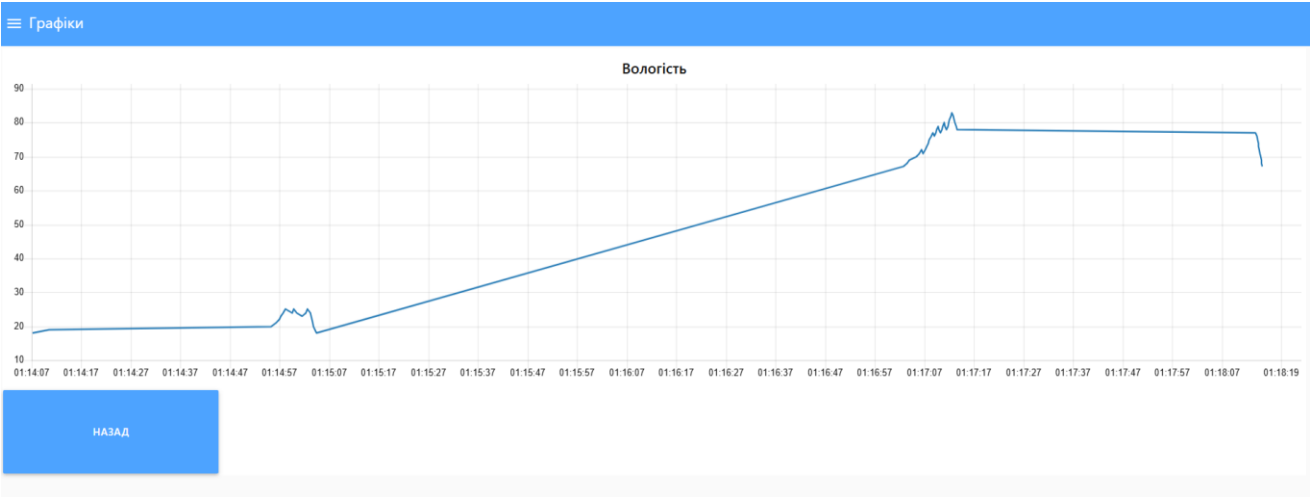


Рис. А4 Сторінка відстеження графіків

Фрагмент коду «Ініціалізація та розпіновка апаратних компонентів»

```
// === Бібліотеки ===
#include <SPI.h>
#include <MFRC522.h>
#include <DHT.h>
#include <Wire.h>
#include <U8g2lib.h>

// === RFID ===
#define RST_PIN 49
#define SS_PIN 53
MFRC522 mfrc522(SS_PIN, RST_PIN);
byte authorizedCard[4] = {0x37, 0xC4, 0xCA, 0x5F};
bool alarmActive = false;
int cardAttempts = 0;

// === Піни ===
#define NUM_ZONES 7
const int dhtPins[NUM_ZONES] = {22, 23, 24, 25, 26, 27, 28};
const int pirPins[NUM_ZONES] = {14, 15, 16, 17, 18, 19, 2};
const int ldrPins[NUM_ZONES] = {A0, A1, A2, A3, A4, A5, A6};

const int TOUCH_PINS[NUM_ZONES] = {32, 33, 34, 35, 36, 37, 38};
const int LED_PINS[NUM_ZONES] = {3, 5, 6, 9, 10, 11, 12};
const int PIR_PINS[NUM_ZONES] = {14, 15, 16, 17, 18, 19, 2};

// Додаткові сенсори
#define WATER_SENSOR_1 40
#define WATER_SENSOR_2 41
#define MQ2_PIN A7
#define MQ7_PIN A8
#define RED_LED_PIN 29
#define GREEN_LED_PIN 44
#define ALERT_LED_PIN 45
#define INTERNAL_BEEPER_PIN 46
#define EXTERNAL_BEEPER_PIN 47
#define COOLER_PIN 13
#define KITCHEN_COOLER_PIN 8
#define HOLD_TIME 1500
#define DOUBLE_TAP_TIME 400
#define FADE_DELAY 20
#define NUM_ZONES 7

// === OLED ===
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE);
```

Фрагмент коду «Оновлення OLED-дисплея та відображення стану кліматичної системи»

```

unsigned long lastChangeTime = 0; // Час останньої зміни
const unsigned long delayTime = 3000; // Затримка в мілісекундах
void updateDisplay() {
    u8g2.clearBuffer();
    u8g2.setFont(u8g2_font_ncenB08_tr);
    u8g2.drawUTF8(0, 10, "Climate: ");
    String statusText;
    if (pwmLevel == 0) statusText = "Off";
    else if (pwmLevel <= 100) statusText = "Quiet";
    else if (pwmLevel <= 170) statusText = "Standard";
    else statusText = "Turbo";
    u8g2.drawUTF8(50, 10, statusText.c_str());

    int iconX = 110, iconY = 10;
    bool isCoolingDueToTemp = false;
    for (int i = 0; i < 6; i++) if (temperature[i] >= 29.0) isCoolingDueToTemp = true;

    float maxHumidity = 0;
    for (int i = 0; i < 6; i++) if (humidity[i] > maxHumidity) maxHumidity =
humidity[i];

    if (pwmLevel > 0 && maxHumidity > 80.0) drawDrop(iconX - 5, iconY);
    if (pwmLevel > 0 && isCoolingDueToTemp && !isHeating) drawSnowflake(iconX + 10,
iconY - 3);
    if (isHeating) drawSun(iconX + 10, iconY);

    if (screenState == 0) {
        for (int i = 0; i < 3; i++) {
            u8g2.setCursor(0, 30 + (i * 10));
            u8g2.print("Zone "); u8g2.print(i + 1);
            u8g2.print(": "); u8g2.print(temperature[i]); u8g2.print("C ");
            u8g2.print(humidity[i]); u8g2.print("%");
        }
    } else {
        for (int i = 3; i < 6; i++) {
            u8g2.setCursor(0, 30 + ((i - 3) * 10));
            u8g2.print("Zone "); u8g2.print(i + 1);
            u8g2.print(": "); u8g2.print(temperature[i]); u8g2.print("C ");
            u8g2.print(humidity[i]); u8g2.print("%");
        }
    }
    if (millis() - lastChangeTime >= delayTime) {
        screenState = (screenState + 1) % 2;
        lastChangeTime = millis();
    }

    // === Вивід у COM-порт ===
    String mode = "Idle";

```

```
if (pwmLevel > 0 && maxHumidity > 80.0) mode = "Ventilation";
if (pwmLevel > 0 && isCoolingDueToTemp && !isHeating) mode = "Cooling";
if (isHeating) mode = "Heating";

Serial.print("Climate Mode: ");
Serial.println(mode);
Serial.print("Fan Status: ");
Serial.println(statusText);

u8g2.sendBuffer();
}
```

Фрагмент коду «Функції тривожного оповіщення (сигналізація): загальна, вода, газ»

```
void triggerAlarm() {
    unsigned long alarmStart = millis();
    unsigned long alarmDuration = 5000;
    while (millis() - alarmStart < alarmDuration) {
        digitalWrite(REDA_LED_PIN, HIGH);
        digitalWrite(ALERT_LED_PIN, HIGH);
        digitalWrite(INTERNAL_BEEPER_PIN, LOW);
        digitalWrite(EXTERNAL_BEEPER_PIN, LOW);
        delay(250);

        digitalWrite(REDA_LED_PIN, LOW);
        digitalWrite(ALERT_LED_PIN, LOW);
        digitalWrite(INTERNAL_BEEPER_PIN, HIGH);
        digitalWrite(EXTERNAL_BEEPER_PIN, HIGH);
        delay(250);
    }
}

void triggerWaterAlarm(String location) {
    Serial.print("[WATER_ALARM] Увага! Протікання води у зоні: ");
    Serial.println(location);
    for (int i = 0; i < 3; i++) {
        digitalWrite(INTERNAL_BEEPER_PIN, LOW);
        digitalWrite(EXTERNAL_BEEPER_PIN, HIGH);
        delay(200);
        digitalWrite(INTERNAL_BEEPER_PIN, HIGH);
        delay(200);
    }
    digitalWrite(INTERNAL_BEEPER_PIN, LOW);
    delay(1000);
    digitalWrite(INTERNAL_BEEPER_PIN, HIGH);
}

void triggerGasAlarm(String source) {
    Serial.print("[GAS_ALARM] Увага! Виявлено газ/дим від: ");
    Serial.println(source);
    for (int i = 0; i < 5; i++) {
        digitalWrite(REDA_LED_PIN, HIGH);
        digitalWrite(INTERNAL_BEEPER_PIN, LOW);
        delay(300);
        digitalWrite(REDA_LED_PIN, LOW);
        digitalWrite(INTERNAL_BEEPER_PIN, HIGH);
        delay(300);
    }
}
```