

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ МІСЬКОГО
ГОСПОДАРСТВА імені О. М. БЕКЕТОВА

**Навчально-науковий інститут енергетичної, інформаційної та
транспортної інфраструктури**
Кафедра автоматизації та комп'ютерно-інтегрованих технологій

РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

на тему: Автоматизація моніторингу та енергозбереження в системах
«Розумний будинок» на основі IoT-технологій та хмарних сервісів

Виконав: здобувач вищої освіти
3 курсу, групи Сінж-2023-1-У
напряму підготовки (спеціальності)
174 «Автоматизація, комп'ютерно-
інтегровані технології та робототехніка»
Валеулін Роман Сергійович
(прізвище та ініціали)

Керівник Піддубна Л.В., доц. каф. АКІТ
(прізвище та ініціали, наук. ступ., вч. звання)

Рецензент Ківіренко О.Б., начальник
виробництва ТОВ «Альфа-Композіт»
(прізвище та ініціали, наук. ступ., вч. звання)

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ МІСЬКОГО
ГОСПОДАРСТВА ІМЕНІ О. М. БЕКЕТОВА**

**Навчально-науковий Інститут енергетичної, інформаційної та
транспортної інфраструктури**

Кафедра автоматизації та комп'ютерно-інтегрованих технологій
Освітньо-кваліфікаційний рівень – бакалавр
Галузь знань 17 «Електроніка, автоматизація та електронні комунікації»
Спеціальність 174 «Автоматизація, комп'ютерно-інтегровані технології та
робототехніка»

ЗАТВЕРДЖУЮ

Завідувач кафедри АКІТ



БАРАНОВ О.О.

« 19 » червня 2026 року

З А В Д А Н Н Я



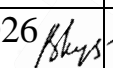

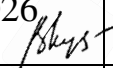
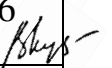


НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Валеулін Роман Сергійович

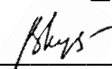
1. Тема роботи: Автоматизація моніторингу та енергозбереження в системах «Розумний будинок» на основі IoT-технологій та хмарних сервісів
Затверджена наказом університету від « 22 » травня 2026 року № 440-03.
Керівник роботи Піддубна Лідія Валеріївна, кандидат філософських наук, доцент, доцент кафедри АКІТ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
2. Строк подання роботи здобувачем вищої освіти « 15 » червня 2026 р.
3. Вихідні дані до роботи: Розробка системи «Розумний будинок», яка забезпечує моніторинг та енергозбереження на основі IoT-технологій та хмарних сервісів
4. Зміст розрахунково пояснювальної записки (перелік питань, які потрібно розробити): Вступ. Аналіз систем «Розумний будинок», технологій ІОТ та методів енергозбереження. Проектування системи моніторингу та енергозбереження. Практична реалізація та відладка системи енергозбереження «Розумного будинку».. Охорона праці, висновки
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)


Презентація.

6. Консультанти розділів проєкту (роботи)





Розділ	Консультант	Підпис, дата	
		завдання видав	завдання прийняв
Аналіз проблеми	Піддубна Л.В.	11.05.2026 	21.05.2026 
Основна частина	Піддубна Л.В.	22.05.2026 	31.05.2026 
Спеціальний розділ	Піддубна Л.В.	01.06.2026 	10.06.2026 
Охорона праці	Малишева В.В.	06.06.2026 	10.06.2026 

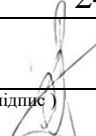
7. Дата видачі завдання «_11_» травня 2026_р.


Керівник  (підпис) Піддубна Л.В.

Завдання прийняв до виконання  (підпис) Валеулін Р.С.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання розділів	Примітка
1	Розробка 1 го розділу бакалаврської роботи	11.05.2026 - 21.05.2026	
2	Розробка 2 го розділу бакалаврської роботи	22.05.2026 - 31.05.2026	
3	Розробка 3 го розділу бакалаврської роботи	01.06.2026 - 10.06.2026	
4	Розробка 4 го розділу з охорони праці	06.06.2026 - 10.06.2026	
5	Рецензування бакалаврської роботи	15.06.2026	Ківіренко О.Б
6	Захист на ДЕК	24.06.2026	

Здобувача вищої освіти  (підпис) Валеулін Р.С.

Керівник  (підпис) Піддубна Л.В.

РЕФЕРАТ

Автоматизація моніторингу та енергозбереження в системах «Розумний будинок» на основі IoT-технологій та хмарних сервісів – Валеулін Роман Сергійович, дипломна робота бакалавра, Харків, Харківський національний університет міського господарства імені О.М. Бекетова, кількість сторінок 114, кількість таблиць 28, кількість рисунків 17, кількість джерел літератури 20.

Актуальність теми зумовлена необхідністю підвищення енергоефективності будівель, зниження витрат енергоресурсів та впровадження сучасних інформаційних технологій у побутове середовище. Використання IoT-технологій та хмарних сервісів у системах «розумного будинку» дозволяє здійснювати ефективний моніторинг і автоматизоване керування споживанням енергії, що забезпечує економію ресурсів і підвищення рівня комфорту.

Об'єктом дослідження є процеси автоматизованого моніторингу та керування енергоспоживанням у системах «розумного будинку».

Предметом дослідження є методи, моделі та програмно-апаратні засоби реалізації систем моніторингу та енергозбереження на основі IoT-технологій і хмарних сервісів.

Метою дипломної роботи є розробка та дослідження автоматизованої системи моніторингу і енергозбереження в середовищі «розумного будинку» з використанням IoT-технологій та хмарних сервісів для підвищення ефективності використання енергоресурсів.

Для реалізації цієї мети необхідно виконати такі завдання:

- Провести аналіз існуючих систем «розумного будинку» та технологій енергозбереження.
- Дослідити сучасні IoT-платформи та хмарні сервіси для збору і обробки даних.
- Розробити структурну схему системи моніторингу та керування.

- Обґрунтувати вибір апаратних компонентів (датчиків, мікроконтролерів, комунікаційних модулів).
- Розробити програмне забезпечення для збору, передачі та обробки даних.
- Реалізувати алгоритми енергозбереження та автоматичного керування.

У процесі виконання дипломної роботи було використано такі методи дослідження, як аналіз науково-технічної літератури та існуючих рішень; системний підхід до проєктування автоматизованих систем; методи моделювання та проєктування IoT-систем; методи збору та обробки даних; алгоритмічні методи оптимізації енергоспоживання; програмна реалізація з використанням сучасних мов і середовищ розробки.

КЛЮЧОВІ СЛОВА: IoT-системи, хмарні сервіси, розумний будинок, технологій енергозбереження.

ABSTRACT

Automation of Monitoring and Energy Efficiency in Smart Home Systems Based on IoT Technologies and Cloud Services – Roman Serhiyovych Valeulin, bachelor's thesis, Kharkiv, Kharkiv National University of Urban Economy named after O.M. Beketov, number of pages 114, number of tables 28, number of figures 17, number of literature sources 20.

The relevance of the topic is due to the need to increase the energy efficiency of buildings, reduce energy consumption and introduce modern information technologies into the domestic environment. The use of IoT technologies and cloud services in «smart home» systems allows for effective monitoring and automated control of energy consumption, which ensures resource savings and increased comfort levels.

The object of research is the processes of automated monitoring and control of energy consumption in «smart home» systems.

The subject of the research is methods, models, and software and hardware for implementing monitoring and energy saving systems based on IoT technologies and cloud services.

The purpose of the thesis is to develop and research an automated monitoring and energy saving system in a «smart home» environment using IoT technologies and cloud services to increase the efficiency of energy use.

To achieve this goal, it is necessary to perform the following tasks:

- Analyze existing smart home systems and energy saving technologies.
- Research modern IoT platforms and cloud services for data collection and processing.
- Develop a structural diagram of the monitoring and control system.
- Justify the choice of hardware components (sensors, microcontrollers, communication modules).
- Develop software for data collection, transmission and processing.
- Implement energy saving and automatic control algorithms.

In the process of completing the thesis, the following research methods were used: analysis of scientific and technical literature and existing solutions; a systems approach to the design of automated systems; methods for modeling and designing IoT systems; methods for collecting and processing data; algorithmic methods for optimizing energy consumption; software implementation using modern languages and development environments.

KEYWORDS: IoT systems, cloud services, smart home, energy saving technologies.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. АНАЛІЗ СИСТЕМ «РОЗУМНИЙ БУДИНОК», ТЕХНОЛОГІЙ ІоТ ТА МЕТОДІВ ЕНЕРГОЗБЕРЕЖЕННЯ	12
1.1 Основні поняття та принципи функціонування систем «розумного будинку».....	12
1.2 Аналіз сучасних ІоТ-технологій та хмарних сервісів	15
1.3 Аналіз існуючих систем моніторингу та енергозбереження.....	28
Висновок до розділу 1.	31
РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ТА ЕНЕРГОЗБЕРЕЖЕННЯ... 33	
2.1 Розробка структурної та функціональної схеми системи	33
2.2 Вибір апаратних засобів та каналів передачі даних	46
2.3 Проєктування програмного забезпечення та бази даних.....	49
Висновок до розділу 2.	60
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ВІДЛАДКА СИСТЕМИ ЕНЕРГОЗБЕРЕЖЕННЯ «РОЗУМНОГО БУДИНКУ».....	61
3.1 Розробка програмно-апаратного прототипу та налаштування хмарного середовища .	61
3.2 Алгоритмічне забезпечення функціонування системи в різних режимах роботи	76
3.3 Аналіз надійності каналів зв'язку та забезпечення безпеки передачі даних	83
Висновок до розділу 3.	86
РОЗДІЛ 4. ОХОРОНА ПРАЦІ	87
4.1. Організаційно-правові основи забезпечення безпеки праці.....	87
4.2 Характеристика об'єкта та виявлення потенційних небезпек.....	88
4.3 Дослідження ризику реалізації потенційних небезпек на об'єкті проєктування та розробка заходів щодо їх попередження.....	92
Висновок до розділу 4.	98
ЗАГАЛЬНІ ВИСНОВКИ.....	100
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	101
Додаток А.....	103

ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій, зростання вартості енергоресурсів та необхідності підвищення енергоефективності житлових і комерційних будівель особливого значення набувають інтелектуальні системи керування споживанням енергії. Концепція «розумного будинку» (Smart Home) є одним із ключових напрямів цифрової трансформації побутового середовища, що передбачає інтеграцію різноманітних пристроїв і систем у єдину інформаційно-керуючу інфраструктуру.

Використання IoT-технологій (Internet of Things) дозволяє забезпечити безперервний моніторинг параметрів середовища (температури, освітлення, вологості, споживання електроенергії тощо), а також автоматизоване керування інженерними системами будинку. Поєднання IoT із хмарними сервісами відкриває нові можливості для зберігання, обробки та аналізу великих обсягів даних, що надходять від сенсорів у реальному часі.

Особливої актуальності набуває задача енергозбереження, оскільки неефективне використання ресурсів призводить до значних економічних втрат та негативного впливу на довкілля. Інтелектуальні алгоритми керування дозволяють оптимізувати режими роботи обладнання, знижувати енергоспоживання без втрати комфорту користувачів.

Таким чином, розробка та впровадження автоматизованих систем моніторингу та енергозбереження на основі IoT і хмарних технологій є актуальним науково-прикладним завданням, що має значну практичну цінність у контексті підвищення енергоефективності та розвитку «розумних» середовищ.

Об'єктом дослідження виступають процеси інтелектуального контролю та оперативного управління розподілом енергії в межах екосистеми «розумного будинку».

Предметом дослідження є програмні та апаратні інструменти реалізації хмарних сервісів і пристроїв інтернету речей, спрямовані на енергетичний моніторинг і заощадження.

Метою роботи є розробка архітектури та дослідження функціонування хмарно-орієнтованої IoT-системи, яка забезпечує мінімізацію втрат енергоресурсів шляхом автоматизованого управління пристроями «розумного будинку».

Завданнями бакалаврської дипломної роботи є:

- Проведення аналізу існуючих систем «розумного будинку» та технологій енергозбереження.
- Дослідження сучасних IoT-платформ та хмарних сервісів для збору і обробки даних.
- Розроблення структурної схеми системи моніторингу та керування.
- Обґрунтування вибору апаратних компонентів (датчиків, мікроконтролерів, комунікаційних модулів).
- Розробка програмного забезпечення для збору, передачі та обробки даних.
- Реалізація алгоритмів енергозбереження та автоматичного керування.

При виконанні дипломної роботи застосовувалися такі методи дослідження, як аналіз науково-технічної літератури та існуючих рішень; системний підхід до проектування автоматизованих систем; методи моделювання та проектування IoT-систем; методи збору та обробки даних; алгоритмічні методи оптимізації енергоспоживання; програмна реалізація з використанням сучасних мов і середовищ розробки.

Аналіз науково-технічної літератури та існуючих рішень застосовувався для вивчення сучасного стану розвитку систем «розумного будинку» та технологій енергозбереження. Наприклад, було проаналізовано принципи роботи популярних IoT-платформ, підходи до збору телеметричних даних із датчиків температури, вологості та енергоспоживання, а також виявлено їхні

переваги й недоліки. Це дозволило обґрунтувати доцільність розробки власної системи з урахуванням специфіки задачі.

Системний підхід до проєктування автоматизованих систем використовувався для побудови цілісної архітектури системи. Зокрема, система була представлена у вигляді взаємопов'язаних компонентів: «датчик – мікроконтролер – мережевий модуль – хмарний сервер – користувацький інтерфейс». Такий підхід дозволив чітко визначити функції кожного елемента та забезпечити узгоджену роботу всієї системи. Методи моделювання та проєктування IoT-систем застосовувалися на етапі розробки структури системи та вибору технологій. Наприклад, було змодельовано процес передачі даних від сенсора температури до хмарного середовища через протокол MQTT, а також розроблено структурну схему системи з урахуванням можливих затримок передачі даних і навантаження на сервер. Методи збору та обробки даних використовувалися для організації моніторингу параметрів середовища. Зокрема, реалізовано періодичне зчитування показників із датчиків (температури, освітленості, споживання електроенергії) з подальшою передачею у хмарну базу даних. Для обробки даних застосовано фільтрацію шумів (наприклад, усереднення значень) та побудову часових рядів для аналізу змін параметрів у часі. Алгоритмічні методи оптимізації енергоспоживання використовувалися для підвищення ефективності роботи системи. Наприклад, реалізовано алгоритм автоматичного вимкнення освітлення за відсутності руху в приміщенні або регулювання температури на основі заданих порогових значень, що дозволило зменшити зайве споживання електроенергії без зниження комфорту користувачів. Програмна реалізація з використанням сучасних мов і середовищ розробки забезпечила практичне впровадження розробленої системи. Наприклад, для мікроконтролера було використано мову програмування C++, для серверної частини – Python, а для візуалізації даних – веб-інтерфейс із використанням HTML, CSS та JavaScript. Це дозволило створити повноцінну систему моніторингу з можливістю віддаленого доступу та керування.

РОЗДІЛ 1. АНАЛІЗ СИСТЕМ «РОЗУМНИЙ БУДИНОК», ТЕХНОЛОГІЙ ІоТ ТА МЕТОДІВ ЕНЕРГОЗБЕРЕЖЕННЯ

1.1 Основні поняття та принципи функціонування систем «розумного будинку»

Сучасний розвиток інформаційних технологій сприяв активному впровадженню концепції «розумного будинку» (Smart Home), яка є одним із ключових напрямів автоматизації житлового середовища. Дана концепція передбачає інтеграцію різноманітних інженерних систем, електронних пристроїв та програмного забезпечення у єдину інформаційно-керуючу систему, що забезпечує ефективне, безпечне та комфортне функціонування житла.

Концепція Smart Home базується на ідеї створення інтелектуального середовища, яке здатне самостійно аналізувати стан об'єкта та реагувати на зміну умов без постійного втручання людини. Основною особливістю таких систем є здатність до автоматичного прийняття рішень на основі даних, отриманих від сенсорів.

Основною метою таких систем є підвищення рівня комфорту, безпеки та енергоефективності житла. Система «розумного будинку» являє собою сукупність апаратних і програмних засобів, що забезпечують автоматичний контроль і керування різними параметрами середовища. До складу системи входять такі основні підсистеми, як система освітлення (керування освітленням залежно від присутності людини або часу доби); кліматична система (опалення, вентиляція, кондиціонування); система безпеки (датчики руху, відкриття дверей, відеоспостереження); система енергомоніторингу (контроль споживання електроенергії) (рисунок 1.1).

Система «розумного будинку» включає кілька ключових складових, а саме сенсорний рівень (датчики), керуючий рівень (контролери), виконавчі пристрої, комунікаційна інфраструктура.



Рисунок 1.1 – Система «Розумний будинок»

Сенсорний рівень (датчики) представляють собою пристрої, що здійснюють збір інформації про стан середовища (температура, вологість, освітленість, рух, споживання електроенергії тощо). Керуючий рівень (контролери) формують мікроконтролери або обчислювальні пристрої, що обробляють дані та формують керуючі сигнали. Виконавчі пристрої поєднують реле, приводи, регулятори, які безпосередньо впливають на роботу обладнання (вмикають світло, регулюють опалення тощо). Комунікаційна інфраструктура представляє собою засоби передачі даних (Wi-Fi, Bluetooth, ZigBee, дротові мережі). Програмне забезпечення та інтерфейс користувача утворено мобільними додатками, веб-інтерфейсами або SCADA-системами, що забезпечують взаємодію користувача із системою.

Функціонування системи «розумного будинку» реалізується через такі три основні функціональні напрями, як моніторинг, керування, автоматизація.

Моніторинг передбачає безперервне спостереження за станом параметрів середовища. Система збирає дані з датчиків і передає їх для подальшої обробки та зберігання. Наприклад, вимірювання температури в приміщенні; контроль рівня освітленості; відстеження споживання електроенергії в реальному часі. Отримані дані можуть відобразитися у

вигляді графіків або таблиць, що дозволяє користувачу аналізувати стан системи.

Керування полягає у впливі на виконавчі пристрої з метою зміни стану системи. Воно може здійснюватися як вручну, так і дистанційно через мережу Інтернет. Наприклад, увімкнення або вимкнення освітлення; регулювання температури в приміщенні; керування електроприладами. Ця функція дозволяє користувачу оперативно реагувати на зміну умов або власні потреби.

Автоматизація є ключовою особливістю систем «розумного будинку» і передбачає виконання дій без участі користувача за заздалегідь визначеними алгоритмами або сценаріями. Наприклад, автоматичне вимкнення світла за відсутності руху; підтримка заданої температури в приміщенні; вимкнення електроприладів у нічний час. Автоматизація дозволяє значно підвищити ефективність використання енергоресурсів та зменшити вплив людського фактора.

Слід зазначити, що саме поєднання функцій моніторингу, керування та автоматизації забезпечує повноцінну роботу системи «розумного будинку» та створює умови для реалізації енергозберігаючих стратегій.

Однією з ключових задач таких систем є контроль та оптимізація енергоспоживання. У традиційних умовах значна частина енергії витрачається неефективно через відсутність автоматичного вимкнення освітлення; нераціональне використання систем опалення; відсутність контролю за електроприладами; людський фактор.

Впровадження концепції Smart Home дозволяє не лише підвищити рівень комфорту та безпеки, але й суттєво зменшити енергоспоживання за рахунок інтелектуального керування ресурсами.

Основою сучасних систем «розумного будинку» є технології Інтернету речей (IoT), які забезпечують взаємодію фізичних пристроїв через мережу Інтернет. До основних компонентів IoT-систем належать датчики (температури, вологості, освітленості, руху, споживання електроенергії); мікроконтролери (наприклад, ESP32, Arduino), що виконують збір і первинну

обробку даних; комунікаційні модулі (Wi-Fi, Bluetooth, ZigBee), які забезпечують передачу інформації; серверна частина, де здійснюється обробка та зберігання даних.

Передача даних у таких системах реалізується за допомогою різних протоколів, серед яких найбільш поширеними є MQTT – легковаговий протокол для передачі телеметрії; HTTP/HTTPS – стандартний протокол для взаємодії з веб-сервісами.

Важливу роль відіграють хмарні сервіси, які забезпечують зберігання великих обсягів даних; обробку інформації в режимі реального часу; побудову графіків та аналітики; доступ користувача до системи з будь-якої точки світу. Наприклад, дані з датчика температури можуть передаватися на хмарний сервер, де вони зберігаються у вигляді часових рядів і використовуються для аналізу змін температури протягом доби.

Використання IoT у поєднанні з хмарними технологіями дозволяє створювати масштабовані, гнучкі та ефективні системи автоматизації, що є основою сучасних рішень у сфері «розумного будинку».

1.2 Аналіз сучасних IoT-технологій та хмарних сервісів

Сучасні системи «розумного будинку» базуються на використанні технологій Інтернету речей (IoT), що забезпечують інтеграцію фізичних пристроїв, сенсорів та виконавчих механізмів у єдину інформаційно-керуючу мережу. Основною функцією IoT є організація безперервного збору, передачі, обробки та аналізу даних у реальному часі. Ефективність функціонування таких систем значною мірою залежить від вибору протоколів передачі даних, програмно-апаратних платформ та хмарних сервісів.

Одним із ключових аспектів IoT-систем є вибір протоколу обміну даними між пристроями та серверною частиною. Найбільш поширеними протоколами є MQTT та HTTP, які мають принципово різні підходи до передачі інформації.

Протокол Message Queuing Telemetry Transport (MQTT) є легковаговим протоколом, розробленим спеціально для систем із обмеженими ресурсами та нестабільними каналами зв'язку. Він працює за моделлю «видавець-підписник» (publish/subscribe), що дозволяє ефективно організувати передачу даних між великою кількістю пристроїв. Основною перевагою протоколу MQTT є низьке енергоспоживання та мінімальне навантаження на мережу, що робить його оптимальним для використання у вбудованих системах, зокрема на базі мікроконтролерів.

У свою чергу, протокол HyperText Transfer Protocol (HTTP) є стандартом взаємодії у веб-середовищі та широко використовується для передачі даних між клієнтом і сервером. Він працює за принципом «запит-відповідь» (request/response), що є більш ресурсоемним у порівнянні з MQTT. Проте HTTP має високу універсальність, простоту реалізації та сумісність із більшістю веб-сервісів, що робить його зручним для інтеграції з хмарними платформами.

Порівняння основних характеристик протоколів MQTT та HTTP наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння протоколів MQTT та HTTP

Характеристика	MQTT	HTTP
Модель взаємодії	Publish/Subscribe	Request/Response
Навантаження на мережу	Низьке	Високе
Енергоспоживання	Низьке	Вище
Швидкість передачі	Висока (для малих даних)	Середня

Характеристика	MQTT	HTTP
Надійність	Висока (QoS-рівні)	Залежить від реалізації
Складність реалізації	Середня	Низька
Сфера застосування	IoT, телеметрія	Веб-сервіси, API

Аналіз показує, що MQTT є більш доцільним для систем моніторингу та передачі телеметричних даних, тоді як HTTP доцільно використовувати для взаємодії з веб-інтерфейсами та хмарними API.

Важливу роль у побудові IoT-систем відіграють програмні платформи, які забезпечують інтеграцію пристроїв, обробку даних та взаємодію з користувачем. До таких платформ належать як відкриті рішення (наприклад, Node-RED, Home Assistant), так і комерційні сервіси. Вони дозволяють організувати логіку обробки даних, створювати сценарії автоматизації та забезпечувати візуалізацію інформації.

IoT-платформи виконують такі основні функції, як збір і агрегація даних із пристроїв; маршрутизація повідомлень; зберігання інформації; реалізація правил автоматизації; інтеграція з іншими сервісами.

Порівняльний аналіз можливостей IoT-платформ наведено у таблиці 1.2.

Таблиця 1.2 – Порівняння IoT-платформ

Характеристика	Відкриті платформи (Node-RED, Home Assistant)	Комерційні платформи
Вартість	Безкоштовні/умовно безкоштовні	Платні
Гнучкість	Висока	Обмежена
Простота використання	Середня	Висока
Підтримка	Спільнота	Офіційна підтримка

Характеристика	Відкриті платформи (Node-RED, Home Assistant)	Комерційні платформи
Масштабованість	Обмежена	Висока
Безпека	Залежить від налаштування	Вбудовані механізми

Аналіз показує, що відкриті платформи є більш придатними для дослідницьких і навчальних проєктів завдяки своїй гнучкості, тоді як комерційні рішення орієнтовані на масштабні системи з високими вимогами до надійності та безпеки.

Хмарні сервіси є невід’ємною частиною сучасних IoT-систем, оскільки вони забезпечують централізоване зберігання, обробку та аналіз даних. Використання хмарних технологій дозволяє реалізувати віддалений доступ до системи, масштабування обчислювальних ресурсів та інтеграцію з іншими інформаційними системами.

Основні можливості хмарних сервісів включають зберігання даних у базах даних (зокрема часових рядів); обробку даних у режимі реального часу; аналітику та візуалізацію; реалізацію серверної логіки; забезпечення доступу через веб-інтерфейс або мобільні додатки.

Разом з тим, використання хмарних сервісів має певні обмеження, а саме, залежність від стабільності інтернет-з’єднання; можливі затримки передачі даних; питання безпеки та конфіденційності; потенційні фінансові витрати.

Порівняння локальних та хмарних рішень наведено у таблиці 1.3.

Таблиця 1.3 – Порівняння локальних і хмарних рішень

Характеристика	Локальні системи	Хмарні сервіси
Доступність	Локальна	Глобальна
Залежність від	Низька	Висока

Характеристика	Локальні системи	Хмарні сервіси
Інтернету		
Масштабованість	Обмежена	Висока
Вартість	Разова	Підписка/оплата за використання
Безпека	Контроль користувача	Залежить від провайдера
Продуктивність	Висока (локально)	Залежить від мережі

Розглянемо більш детально сучасні IoT-платформи з відкритим (open-source) кодом, такі як Node-RED та Home Assistant, і комерційні хмарні сервіси, що використовуються для побудови масштабованих систем Інтернету речей. Кожен із цих підходів має свої особливості, переваги та обмеження, що визначають доцільність їх використання в системах «розумного будинку».

Відкриті IoT-рішення, зокрема Node-RED та Home Assistant, набули значного поширення завдяки своїй доступності, гнучкості та активній спільноті розробників.

Home Assistant є повноцінною платформою домашньої автоматизації, яка забезпечує інтеграцію великої кількості пристроїв і сервісів у єдину систему. Вона підтримує локальне розгортання, що дозволяє мінімізувати залежність від хмарних сервісів та підвищити рівень конфіденційності даних. Система має зручний інтерфейс користувача та підтримує автоматизацію як через конфігураційні файли, так і через графічний інтерфейс (рисунок 1.2).

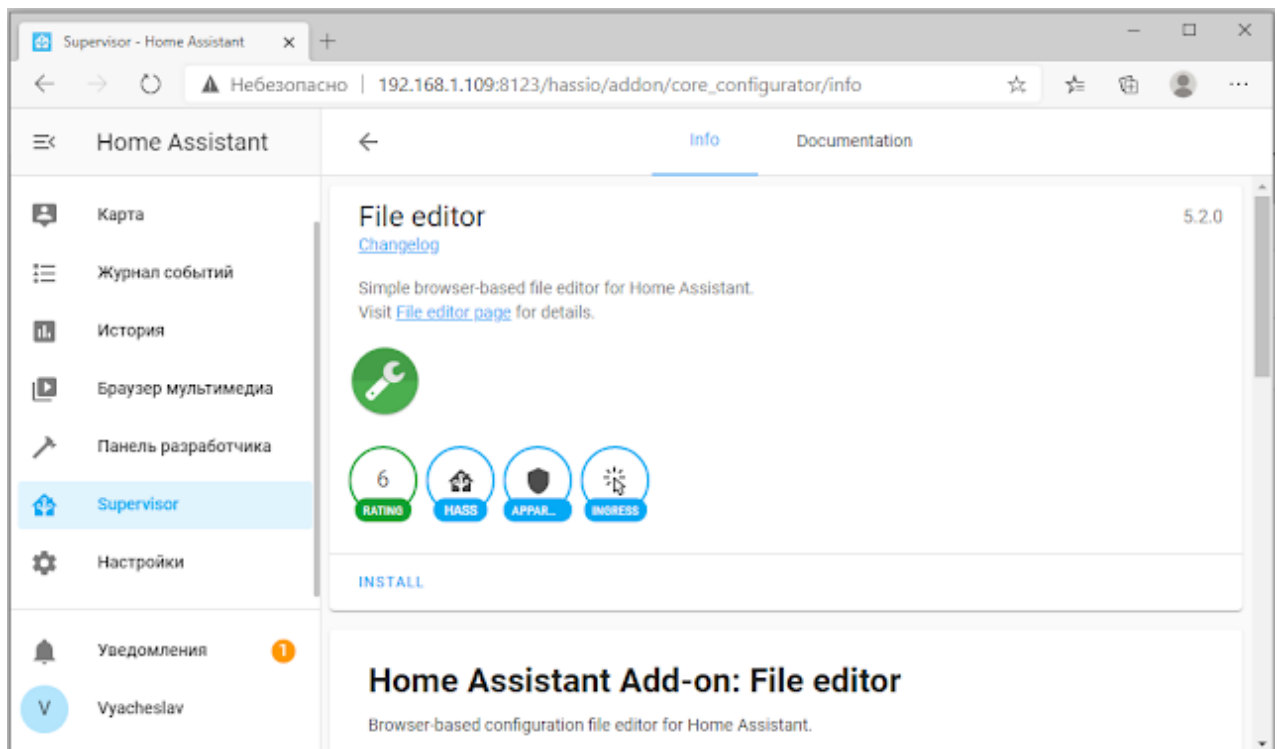


Рисунок 1.2 – Интерфейс Home Assistant

Node-RED – це інструмент візуального програмування, розроблений командою IBM Emerging Technology Services, який дозволяє створювати логіку взаємодії між пристроями у вигляді потокових діаграм (flows) (рисунок 1.3).

Node-RED не є повноцінною платформою «розумного будинку», але широко використовується як засіб реалізації складних сценаріїв автоматизації. Основною перевагою Node-RED є простота створення алгоритмів завдяки графічному інтерфейсу, що значно знижує поріг входження для користувачів.

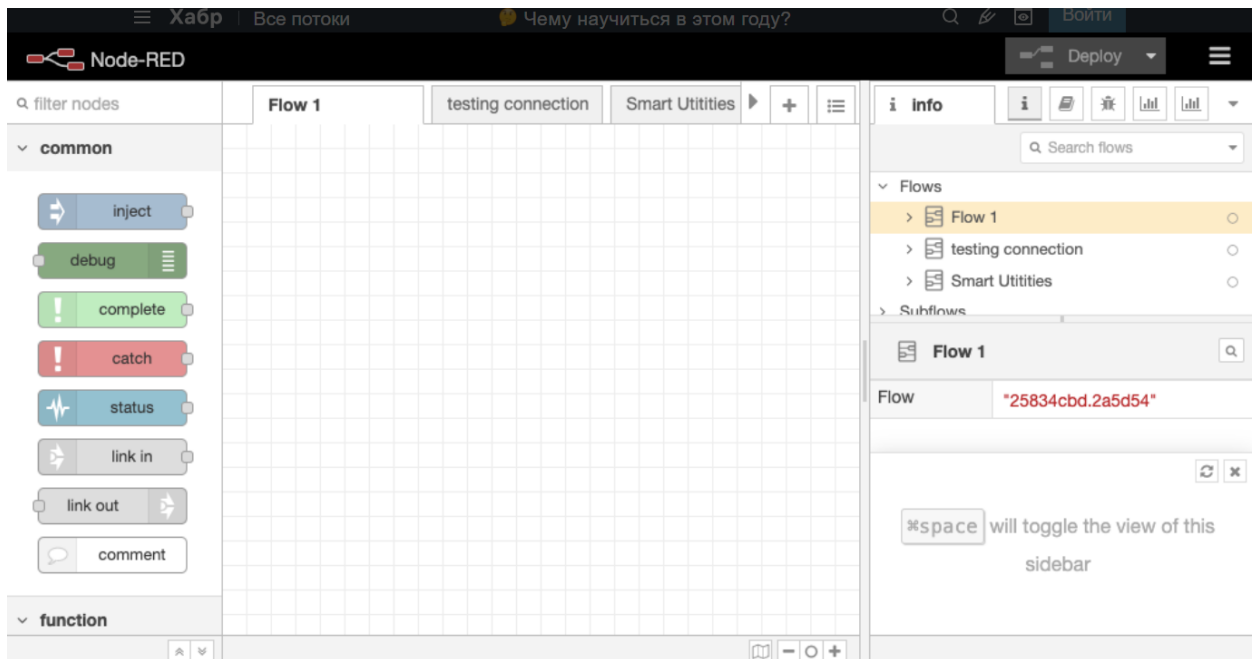


Рисунок 1.3 – Інтерфейс Node-RED

Ключова відмінність між цими рішеннями полягає у їх функціональному призначенні: Home Assistant орієнтований на інтеграцію пристроїв і керування ними, тоді як Node-RED забезпечує гнучке програмування логіки обробки подій та автоматизації. Часто ці системи використовуються разом, де Home Assistant виступає як центральний контролер, а Node-RED – як інструмент реалізації складних алгоритмів.

Порівняльна характеристика відкритих платформ наведена у таблиці 1.4.

Таблиця 1.4 – Порівняння відкритих IoT-платформ

Характеристика	Home Assistant	Node-RED
Тип системи	Платформа автоматизації	Інструмент програмування
Основне призначення	Керування пристроями	Побудова логіки автоматизації
Інтерфейс	Графічний	+ Візуальний (flow-based)

Характеристика	Home Assistant	Node-RED
	конфігураційний	
Підтримка пристроїв	Широка	Через інтеграції
Складність використання	Середня	Низька (для базових задач)
Гнучкість	Висока	Дуже висока
Залежність від хмари	Низька	Низька

Поряд із відкритими платформами значного розвитку набули комерційні IoT-сервіси, такі як Amazon Web Services IoT (AWS IoT), Arduino IoT Cloud, Google Cloud IoT та інші. Дані рішення орієнтовані на побудову масштабованих систем із високими вимогами до надійності, безпеки та продуктивності.

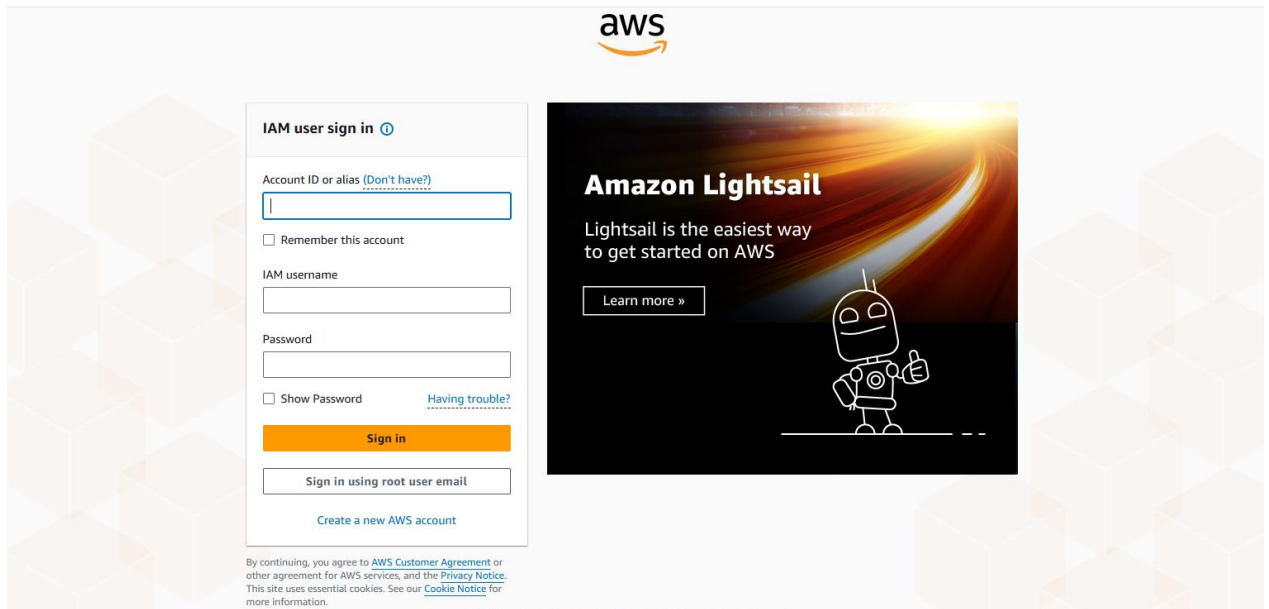


Рисунок 1.4 – Платформа Amazon Web Services

AWS IoT Core (Amazon Web Services) є найбільш потужним промисловим рішенням на ринку. Платформа дозволяє підключати мільярди

пристроїв і обробляти трильйони повідомлень. AWS має унікальну функцію Device Shadow («цифровий двійник»), яка зберігає стан пристрою, навіть якщо він офлайн. Це ідеально для енергозбереження, адже можна надіслати команду «вимкнути опалення» у хмару, а пристрій виконає її, щойно з'явиться зв'язок. AWS використовує взаємну автентифікацію та шифрування на всіх етапах (X.509 сертифікати).

AWS IoT надає хмарні сервіси, які з'єднують ваші пристрої Інтернету речей з іншими пристроями та хмарними сервісами AWS. AWS IoT надає програмне забезпечення для пристроїв, яке може допомогти вам інтегрувати ваші пристрої Інтернету речей у рішення на базі AWS IoT. Якщо ваші пристрої можуть підключатися до AWS IoT, AWS IoT може підключити їх до хмарних сервісів, що надає AWS (рисунок 1.5).

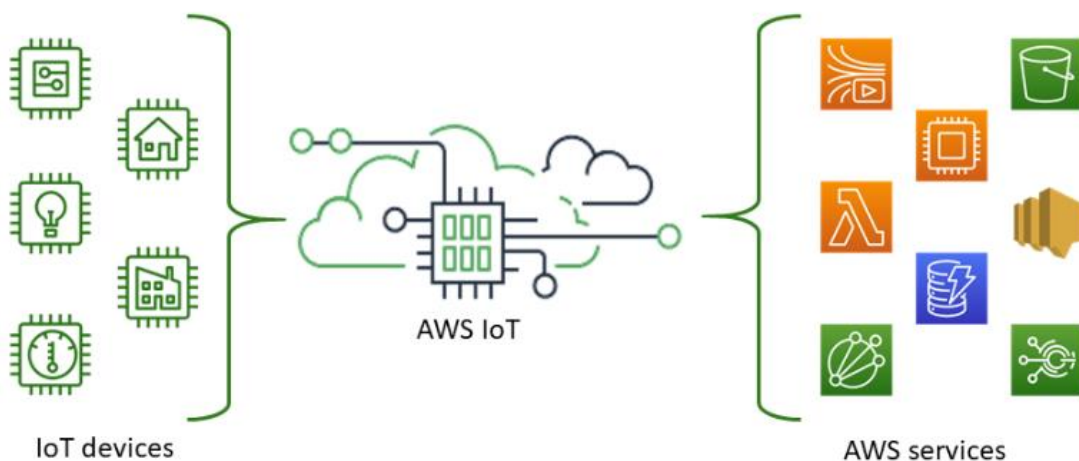


Рисунок 1.5 – Хмарні сервіси AWS IoT

Arduino IoT Cloud – найбільш дружня платформа для швидкого прототипування, яка орієнтована на розробників і інженерів, що використовують екосистему Arduino або ESP32/ESP8266. Платформа має автоматичну генерацію коду (Sketch) та вбудований конструктор віджетів (Dashboard) для мобільних додатків. Вона дозволяє створити систему

енергомоніторингу за лічені години без глибоких знань хмарної архітектури. Arduino IoT Cloud легко зв'язується з Alexa та Google Home.

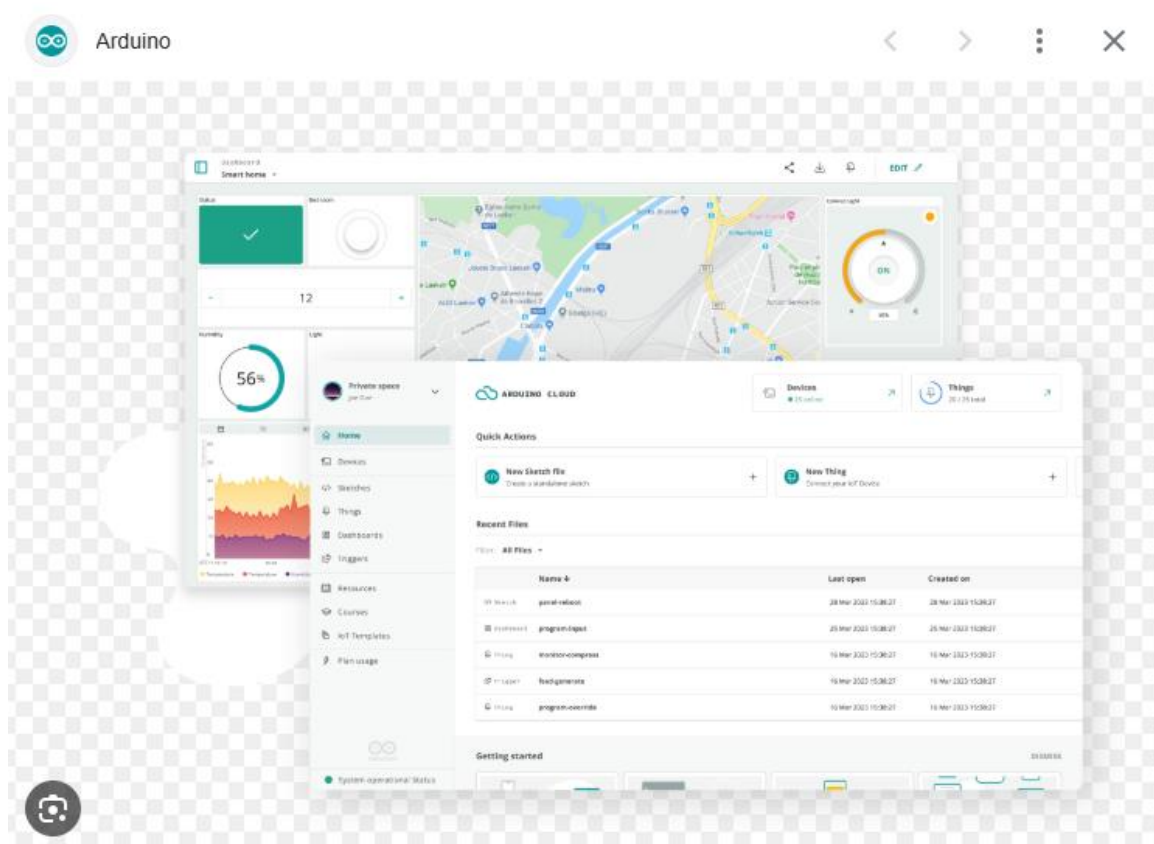


Рисунок 1.6 – Arduino IoT Cloud

Google Cloud IoT (зараз замінюється партнерськими рішеннями). Google офіційно закрила свій сервіс IoT Core у 2023 році, перевівши клієнтів на рішення партнерів (наприклад, LITMUS або KORE). Однак у межах Google Cloud зараз використовуються сервіси Pub/Sub та BigQuery для збору та аналізу IoT-даних.

Cloud IoT Core – це повністю керований сервіс, який дозволяє швидко та безпечно підключатися, налаштовувати та отримувати дані з великої кількості пристроїв. Використовуючи Cloud Pub/Sub, Core може об'єднувати дані з децентралізованих пристроїв в єдину глобальну систему. У поєднанні з іншими сервісами Google Cloud IoT Core пропонує комплексне рішення для

збору, аналізу та візуалізації даних Інтернету речей у режимі реального часу. Це, у свою чергу, дозволяє створювати багатофункціональні моделі (рисунок 1.7).

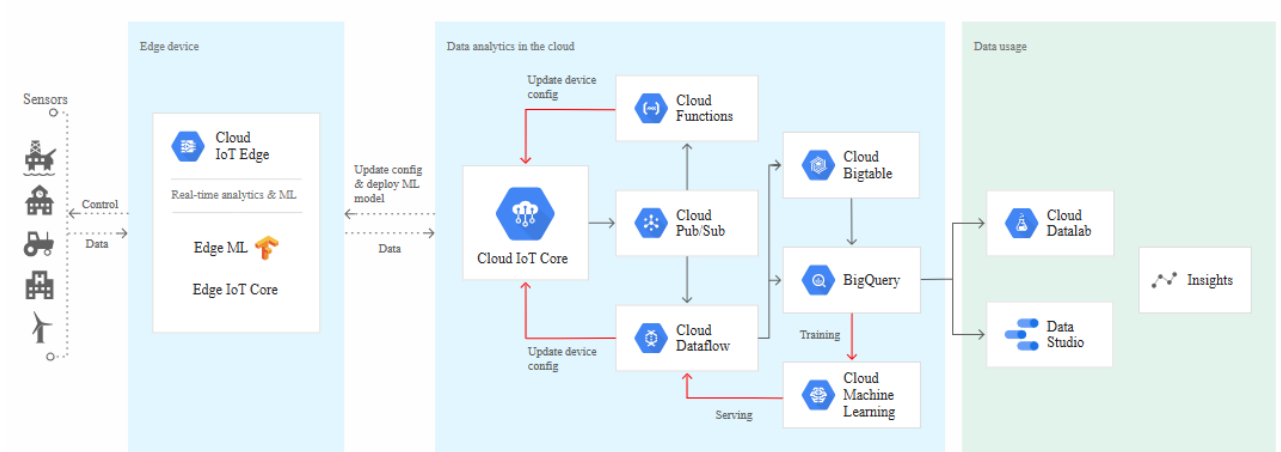


Рисунок 1.7 – Google Cloud IoT

Cloud IoT Core підтримує традиційні протоколи MQTT та HTTP, що дозволяє клієнтам використовувати багато існуючих пристроїв для побудови власних IoT-систем. Цей сервіс працює в інфраструктурі Google, яка автоматично масштабується в режимі реального часу, забезпечуючи можливість підключення мільйона пристроїв за потреби.

Таблиця 1.5 – Порівняльна таблиця IoT-сервісів

Критерій порівняння	AWS IoT Core	Arduino IoT Cloud	Google Cloud (Pub/Sub + ML)
Складність освоєння	Висока (потребує сертифікації)	Низька (ідеально для студентів)	Середня

Критерій порівняння	AWS IoT Core	Arduino IoT Cloud	Google Cloud (Pub/Sub + ML)
Масштабованість	Промислова (необмежена)	Малий та середній бізнес	Промислова
Протоколи зв'язку	MQTT, HTTP, LoRaWAN	MQTT, WebSockets	MQTT, HTTP
Аналітика даних	Потужна (AWS Quicksight)	Базова (графіки)	Найкраща (BigQuery ML)
Вартість	Pay-as-you-go (за повідомлення)	Підписка (є безкоштовний план)	Pay-as-you-go (за обсяг даних)
Головна перевага	Надійність та безпека	Швидкість розробки	Робота з Big Data та ШІ

Комерційні платформи забезпечують широкий спектр функціональних можливостей, серед яких підтримка стандартних протоколів (MQTT, HTTPS, WebSockets); масштабоване зберігання даних; вбудовані засоби аналітики; інтеграція з іншими хмарними сервісами; високий рівень безпеки та автентифікації пристроїв. Наприклад, хмарні IoT-платформи дозволяють організувати взаємодію тисяч пристроїв одночасно, забезпечуючи при цьому мінімальні затримки та високу доступність сервісу. Однак їх використання часто пов'язане з фінансовими витратами, а також залежністю від постачальника послуг.

Порівняння відкритих і комерційних рішень наведено у таблиці 1.6.

Таблиця 1.6 – Порівняння відкритих і комерційних IoT-рішень

Характеристика	Відкриті рішення (Node-RED, Home Assistant)	Комерційні IoT-платформи
Вартість	Безкоштовні	Платні
Гнучкість	Дуже висока	Обмежена політикою сервісу
Масштабованість	Обмежена	Висока
Безпека	Залежить від налаштування	Вбудовані механізми
Простота використання	Середня	Висока
Залежність від Інтернету	Низька	Висока
Контроль даних	Повний контроль користувача	Частково обмежений

Аналіз показує, що відкриті платформи є оптимальними для невеликих систем «розумного будинку», дослідницьких проєктів та прототипування, де важлива гнучкість і мінімальні витрати. Водночас комерційні IoT-сервіси доцільно використовувати у випадках, коли необхідна висока масштабованість, надійність та інтеграція з корпоративною інфраструктурою.

На основі проведеного аналізу можна зробити висновок, що для систем «розумного будинку» доцільним є комбіноване використання IoT-технологій та хмарних сервісів. Зокрема, використання протоколу MQTT для передачі телеметричних даних, локального контролера для первинної обробки інформації та хмарного сервера для зберігання і аналітики забезпечує оптимальне поєднання швидкодії, енергоефективності та функціональності.

Сучасні IoT-технології та хмарні сервіси створюють основу для побудови ефективних систем моніторингу та енергозбереження, які здатні адаптуватися до потреб користувача та забезпечувати раціональне використання ресурсів.

Таким чином, вибір конкретного рішення залежить від вимог до системи. У контексті даної роботи доцільним є використання відкритих платформ у поєднанні з хмарними сервісами, що дозволяє забезпечити баланс між гнучкістю, вартістю та функціональністю системи моніторингу та енергозбереження.

1.3 Аналіз існуючих систем моніторингу та енергозбереження

На сучасному ринку технологій «розумного будинку» представлено широкий спектр рішень: від закритих комерційних екосистем до відкритих платформ для розробників, які забезпечують функції моніторингу та керування енергоспоживанням. Їх можна умовно поділити на пропріетарні (комерційні) екосистеми, професійні системи автоматизації, відкриті (open-source) платформи.

Пропріетарні (комерційні) екосистеми (Apple HomeKit, Xiaomi Mi Home, Google Home) є зручними і надійними для кінцевого користувача, мають широкий функціонал і естетичний дизайн, легкі у налаштуванні (Plug-and-Play). Недоліками цих систем є висока вартість обладнання та впровадження; обмежена гнучкість налаштування, залежність від серверів виробника, складність інтеграції з іншими системами. Користувач не може вносити зміни в логіку керування енергоспоживанням на низькому рівні, наприклад, реалізувати складне PID-регулювання опалення.

Професійні системи автоматизації (KNX, Control4) характеризуються надійністю, дротовим зв'язком, відповідністю промисловим стандартам автоматизації. У якості їх недоліків можна зазначити високу вартість обладнання та послуг із інсталяції. Професійні системи потребують спеціального навчання для програмування системи.

Відкриті платформи (Home Assistant, OpenHAB) мають повну автономність, можливість інтеграції пристроїв різних брендів, можливість реалізації складних сценаріїв автоматизації, складність у налаштуванні; потребу у технічних знаннях; обмежену технічну підтримку. Для них характерним є високий поріг входження, необхідність налаштування власного сервера (Raspberry Pi/Mini-PC).

Таблиця 1.7 – Порівняльна характеристика систем

Критерій	Xiaomi Mi Home	KNX	Home Assistant	Власна розробка (IoT + Cloud)
Вартість	Низька	Дуже висока	Середня	Мінімальна
Гнучкість налаштувань	Обмежена	Середня	Висока	Максимальна
Надійність	Середня	Абсолютна	Висока	Висока
Енергоефективність	Базова (таймери)	Висока	Висока	Інтелектуальна (прогнозна)
Відкритість коду	Закритий	Закритий	Відкритий	Відкритий

Аналіз існуючих систем показав, що більшість комерційних рішень мають вузьку спеціалізацію, проблему конфіденційності, відсутність інтелектуального аналізу. Вузька спеціалізація обумовлена тим, що більшість систем лише констатують факт споживання енергії, але не пропонують активних сценаріїв економії на основі зовнішніх даних, наприклад, прогнозу погоди чи багатотарифної сітки. Проблема конфіденційності виявляється у тому, що дані про активність користувача в будинку передаються на сторонні сервери без можливості повного контролю. Відсутність інтелектуального аналізу виражається у тому, що бракує алгоритмів, які б аналізували поведінку користувача та автоматично коригували енергетичний профіль будівлі.

Більшість існуючих систем не повною мірою вирішують задачу ефективного енергозбереження або є недоступними для широкого кола користувачів через високу вартість. Крім того, часто відсутня можливість гнучкого налаштування алгоритмів керування, що обмежує ефективність використання ресурсів.

Існує необхідність у розробці доступної, гнучкої та ефективної системи моніторингу та енергозбереження, яка базується на сучасних IoT-технологіях та хмарних сервісах і дозволяє оптимізувати споживання енергоресурсів у житлових приміщеннях.

На основі проведеного аналізу існуючих рішень доцільно сформулювати комплекс вимог до розроблюваної системи моніторингу та енергозбереження, які охоплюють функціональні, технічні, експлуатаційні та економічні аспекти.

До функціональних вимог слід віднести забезпечення збору даних із різномірних джерел, зокрема показників електроспоживання, температури та параметрів роботи обладнання. Система повинна підтримувати обробку даних у режимі реального часу, здійснювати автоматичне виявлення аномалій у споживанні енергоресурсів, а також забезпечувати формування аналітичних звітів і рекомендацій щодо оптимізації енергоспоживання. Важливою складовою є наявність засобів візуалізації даних у вигляді інформаційних панелей (дашбордів).

Технічні вимоги передбачають підтримку сучасних стандартних протоколів обміну даними, таких як MQTT, HTTP та Modbus, що забезпечує сумісність із широким спектром пристроїв. Архітектура системи повинна бути модульною, що дозволяє її гнучке розширення та модернізацію. Крім того, система має бути масштабованою та підтримувати як хмарне, так і локальне розгортання, а також забезпечувати інтеграцію з IoT-пристроями.

До експлуатаційних вимог належать забезпечення зручного та інтуїтивно зрозумілого інтерфейсу користувача, висока надійність і відмовостійкість системи, захист даних від несанкціонованого доступу, а також можливість віддаленого доступу для моніторингу та управління.

Економічні вимоги спрямовані на зниження витрат на впровадження та експлуатацію системи. Зокрема, доцільним є використання відкритих технологій і програмного забезпечення, що дозволяє мінімізувати ліцензійні витрати, а також забезпечити низькі витрати на технічне обслуговування.

Сформульовані вимоги визначають основні напрямки проєктування ефективної, гнучкої та економічно доцільної системи моніторингу та енергозбереження.

Висновок до розділу 1.

У сучасних умовах розвитку промисловості важливу роль відіграють системи автоматизованого управління та комп'ютерно-інтегровані платформи. Вони забезпечують моніторинг технологічних процесів, координацію виробничих підсистем та управління ресурсами підприємства. Завдяки таким системам підприємства можуть підвищити ефективність виробництва, зменшити витрати та покращити контроль за виконанням виробничих операцій.

У першому розділі бакалаврської роботи було проведено комплексний аналіз систем «розумного будинку», сучасних IoT-технологій та методів енергозбереження, що дозволило сформувавши теоретичну та практичну основу для подальшої розробки системи моніторингу.

Розглянуто основні поняття та принципи функціонування систем «розумного будинку». Визначено, що такі системи базуються на інтеграції сенсорів, виконавчих пристроїв і керуючих модулів, які забезпечують автоматизацію побутових процесів, підвищення комфорту, безпеки та енергоефективності.

Виконано аналіз сучасних IoT-технологій та хмарних сервісів. Встановлено, що використання протоколів обміну даними, таких як MQTT, HTTP та інших, забезпечує ефективну взаємодію між пристроями, а хмарні платформи надають широкі можливості для зберігання, обробки та аналізу даних. Визначено переваги IoT-рішень, зокрема масштабованість, гнучкість і можливість віддаленого доступу, а також окреслено їх обмеження, пов'язані з безпекою, затримками передачі даних та залежністю від інтернет-з'єднання.

Здійснено аналіз існуючих систем моніторингу та енергозбереження. Проведене порівняння комерційних і відкритих рішень показало, що сучасні системи мають широкий функціонал і забезпечують ефективний контроль енергоспоживання, однак характеризуються такими недоліками, як висока вартість, складність інтеграції, обмежена гнучкість або підвищені вимоги до кваліфікації користувачів. На основі цього було сформульовано вимоги до розроблюваної системи, що враховують функціональні, технічні, експлуатаційні та економічні аспекти.

РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ТА ЕНЕРГОЗБЕРЕЖЕННЯ

2.1 Розробка структурної та функціональної схеми системи

Архітектура розроблюваної системи реалізована за п'ятирівневою моделлю взаємодії:

«датчики → мікроконтролер → мережа → хмарний сервер → користувач».

Кожен рівень виконує специфічні функції збору, обробки та передачі інформації.

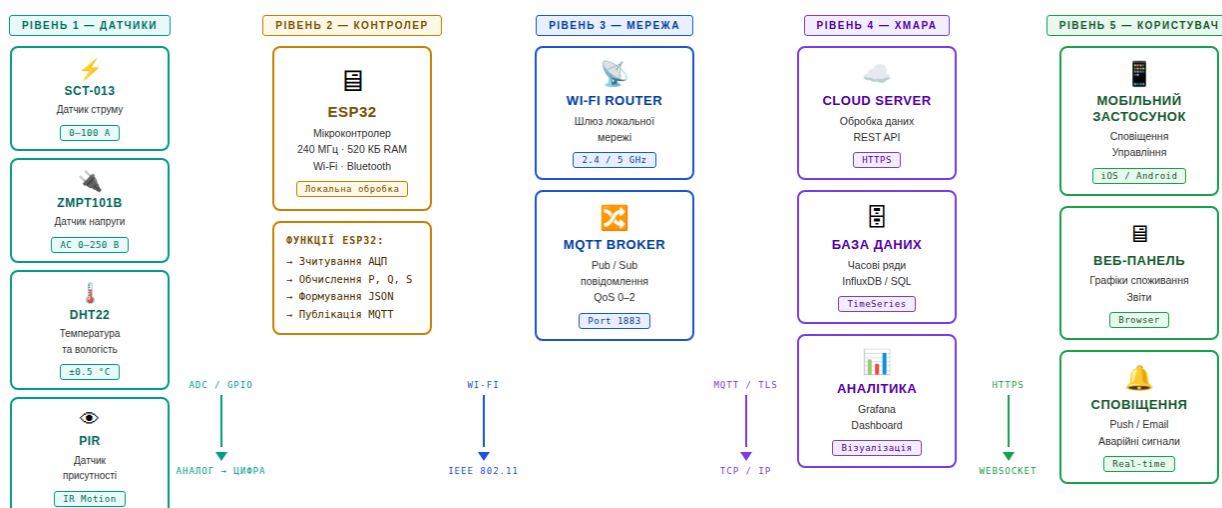


Рисунок 2.1 – Архітектура системи моніторингу та енергозбереження «Розумний будинок»

Розглянемо більш детально рівень первинних перетворювачів (сенсорів), який є базовим елементом системи моніторингу та енергозбереження. Саме на цьому рівні відбувається перетворення фізичних величин у електричні сигнали, придатні для подальшої обробки мікроконтролером.

До складу системи входять датчики струму (SCT-013), напруги (ZMPT101B), температури (DHT22) та присутності (PIR), кожен з яких виконує окрему функцію та забезпечує збір відповідних параметрів.

Зробимо коротке пояснення до рисунку 2.1:

Рівень 1 – Датчики (бірюзовий): датчики струму SCT-013, напруги ZMPT101B, температури DHT22, присутності PIR;

Рівень 2 – Контролер (amber): ESP32 з описом функцій обробки;

Рівень 3 – Мережа (синій): Wi-Fi роутер + MQTT Broker;

Рівень 4 – Хмара (фіолетовий): Cloud Server, БД, Grafana;

Рівень 5 – Користувач (зелений): мобільний застосунок, веб-панель, сповіщення.

Між рівнями вказані протоколи передачі: ADC/GPIO → Wi-Fi → MQTT/TLS → HTTPS/WebSocket.

Датчик струму SCT-013 є неінвазивним трансформатором струму, який дозволяє вимірювати силу струму без розриву електричного кола (рисунок 2.2). Його принцип роботи базується на явищі електромагнітної індукції, а саме, змінний струм у провіднику створює магнітне поле, яке індуктує струм у вторинній обмотці датчика. Це забезпечує безпечне підключення та можливість використання у побутових і промислових умовах. Отримані дані використовуються для оцінки споживаної потужності та виявлення перевантажень.



Рисунок 2.2 – Датчик струму SCT-013

Датчик напруги ZMPT101B призначений для вимірювання змінної напруги в електричній мережі. Він містить мініатюрний трансформатор напруги, що забезпечує гальванічну розв'язку між високовольтною мережею та низьковольтною частиною системи. Це підвищує безпеку та точність вимірювань. Дані з цього датчика використовуються разом із показами струму для розрахунку миттєвої та середньої потужності.

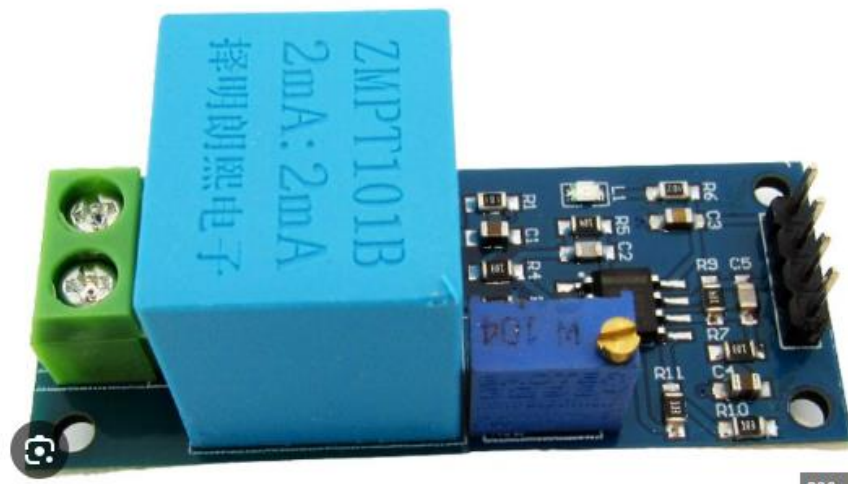


Рисунок 2.3 – Датчик напруги ZMPT101B

Датчик температури та вологості DHT22 забезпечує вимірювання параметрів навколишнього середовища (рисунок 2.4). Він використовує ємнісний сенсор вологості та термістор для визначення температури. Отримані дані дозволяють оцінювати мікроклімат у приміщенні та оптимізувати роботу систем опалення, вентиляції та кондиціонування, що безпосередньо впливає на енергоспоживання.

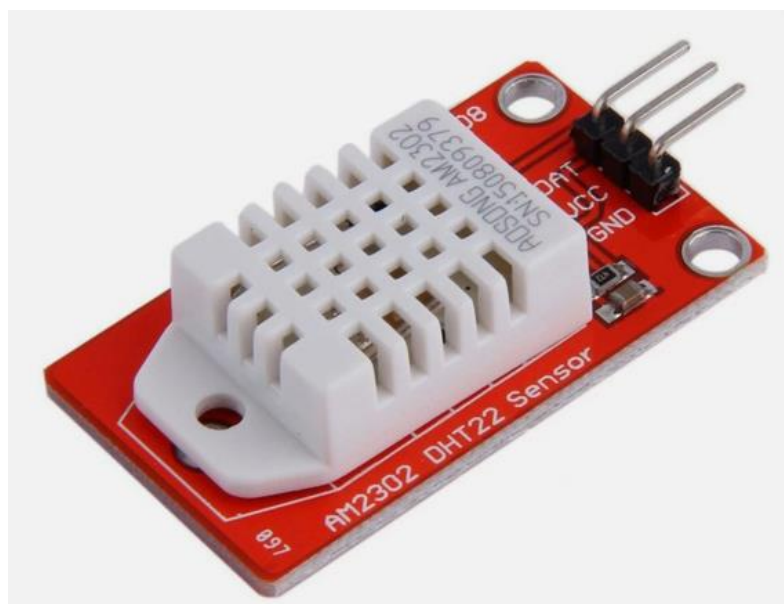


Рисунок 2.4 – Датчик температури та вологості DHT22

Passive Infrared Sensor-датчик (PIR-датчик) призначений для виявлення присутності людини на основі реєстрації інфрачервоного випромінювання. Він реагує на зміну теплового фону в зоні спостереження, що дозволяє визначити рух об'єктів. Використання такого датчика дає можливість реалізувати сценарії автоматичного керування освітленням або обладнанням, що сприяє зниженню енергоспоживання у випадках відсутності користувачів.



Рисунок 2.5 – PIR-датчик

Сукупність зазначених сенсорів забезпечує комплексний моніторинг як електричних параметрів (струм, напруга, потужність), так і параметрів середовища (температура, вологість, присутність). Безперервне зняття цих даних створює основу для подальшого аналізу, виявлення неефективного використання енергії та реалізації інтелектуальних алгоритмів енергозбереження.

Рівень локального контролю є ключовим елементом системи моніторингу та енергозбереження, оскільки саме на цьому етапі відбувається первинна обробка даних, прийняття оперативних рішень та керування виконавчими пристроями. У розроблюваній системі цей рівень реалізовано на базі мікроконтролера ESP32, який поєднує достатню обчислювальну потужність, енергоефективність та вбудовані засоби бездротового зв'язку (Wi-Fi, Bluetooth).

Однією з основних функцій мікроконтролера є реалізація підходу Edge Computing (периферійних обчислень). Це означає, що обробка значної частини даних здійснюється безпосередньо на пристрої, без необхідності постійної передачі всіх сирих даних у хмару. Такий підхід дозволяє зменшити затримки, знизити навантаження на мережу та підвищити надійність системи у разі втрати зв'язку з сервером.



Рисунок 2.6 – Мікроконтролер ESP32.

Мікроконтролер виконує оцифрування аналогових сигналів, що надходять від сенсорів, зокрема датчиків струму та напруги. Для цього використовується вбудований аналого-цифровий перетворювач (АЦП), який перетворює безперервний електричний сигнал у дискретні цифрові значення. Ці значення далі використовуються для математичної обробки та аналізу.

На основі отриманих даних виконується розрахунок миттєвої електричної потужності. Зокрема, мікроконтролер обчислює активну потужність за формулою добутку миттєвих значень струму та напруги з урахуванням коефіцієнта потужності (за необхідності). Також можуть визначатися середні значення, пікові навантаження та інші енергетичні показники. Це дозволяє оперативно оцінювати стан енергоспоживання та виявляти перевантаження або неефективні режими роботи. Крім обробки даних, ESP32 виконує функцію керування виконавчими механізмами, зокрема релейними модулями. Реле дозволяють вмикати або вимикати електричні навантаження (освітлення, побутові прилади, обладнання) відповідно до заданих умов або алгоритмів. Наприклад, система може автоматично відключати обладнання при перевищенні допустимого рівня споживання або за відсутності користувача (на основі даних з PIR-датчика).

Додатково мікроконтролер забезпечує попередню фільтрацію та згладжування даних для зменшення шумів; формування структурованих пакетів даних для передачі на сервер; взаємодію з хмарними сервісами через мережеві протоколи; локальне зберігання даних у разі відсутності зв'язку.

Рівень локального контролю на базі ESP32 виконує роль інтелектуального вузла системи, який поєднує функції збору, обробки та аналізу даних із можливістю автономного керування. Це значно підвищує ефективність, швидкодію та надійність системи моніторингу та енергозбереження.

Розглянемо детально мережевий рівень системи, який відповідає за передачу даних між локальним контролером та хмарною платформою. Саме цей рівень забезпечує обмін інформацією, необхідною для моніторингу,

аналізу та керування енергоспоживанням. У розроблюваній системі передача даних здійснюється за допомогою бездротової технології Wi-Fi, що є доцільним вибором для побутових і малих промислових об'єктів. Використання Wi-Fi забезпечує достатню швидкість передачі даних, простоту підключення до існуючої мережевої інфраструктури та відсутність потреби у додатковому спеціалізованому обладнанні. Крім того, мікроконтролер ESP32 має вбудований Wi-Fi модуль, що спрощує реалізацію мережевої взаємодії.

Для організації обміну даними використовується легковагий протокол Message Queuing Telemetry Transport (MQTT), який спеціально розроблений для IoT-систем з обмеженими ресурсами та нестабільними каналами зв'язку. Основною особливістю MQTT є використання моделі «видавець–підписник» (publish–subscribe), що передбачає наявність проміжного вузла – брокера повідомлень. У цій моделі мікроконтролер виступає у ролі видавця (publisher), який надсилає дані (наприклад, значення струму, напруги, температури) у певні тематичні канали (topics). Хмарна платформа або інші клієнти виконують роль підписників (subscribers), отримуючи лише ті дані, на які вони підписані, що дозволяє ефективно організувати передачу інформації без зайвого навантаження на мережу.

Використання MQTT має низку суттєвих переваг, а саме, мінімальні затрати трафіку, надійність передачі, підтримка роботи в умовах нестабільного зв'язку, масштабованість, асинхронність обміну. Мінімальні затрати трафіку виникають завдяки компактному формату повідомлень і відсутності надлишкових службових даних. Надійність передачі реалізується через рівні якості обслуговування (QoS), які дозволяють гарантувати доставку повідомлень. Підтримка роботи в умовах нестабільного зв'язку дає можливість повторної передачі та збереження повідомлень. Масштабованість передбачає можливість підключення великої кількості пристроїв до одного брокера. Асинхронність обміну означає відсутність необхідності постійного прямого з'єднання між клієнтами.

Процес передачі даних відбувається наступним чином: після збору та попередньої обробки даних мікроконтролер формує повідомлення та передає його через Wi-Fi на MQTT-брокер, розташований у хмарному середовищі або локальній мережі. Брокер приймає повідомлення та розповсюджує його серед підписаних клієнтів, наприклад веб-додатку або аналітичного сервісу.

Додатково можуть використовуватися механізми захисту, такі як шифрування даних (TLS), автентифікація клієнтів та контроль доступу до тем, що підвищує безпеку передачі інформації. Мережевий рівень, побудований на основі Wi-Fi та протоколу MQTT, забезпечує ефективний, надійний та економічний обмін даними між елементами системи, що дозволяє реалізувати безперервний моніторинг параметрів енергоспоживання та оперативну взаємодію з хмарною інфраструктурою.

Розглянемо детально інформаційно-аналітичний рівень системи, який реалізується у вигляді хмарного сервера та є центральним елементом обробки, зберігання й аналізу даних. Саме на цьому рівні відбувається перетворення зібраної інформації у корисні аналітичні висновки та управлінські рішення.

Хмарна інфраструктура, побудована на базі сервісів AWS IoT або приватного сервера з використанням фреймворку FastAPI, виконує роль брокера повідомлень, сервера обробки даних і сховища інформації. Вона забезпечує взаємодію між локальними контролерами, клієнтськими додатками та аналітичними модулями.

Однією з основних функцій цього рівня є прийом і маршрутизація даних, що надходять від мікроконтролерів через протокол MQTT. У випадку використання AWS IoT брокер повідомлень реалізований як керований сервіс, який автоматично забезпечує масштабованість, балансування навантаження та високу доступність. У випадку приватного рішення брокер може бути реалізований окремо, наприклад, Mosquitto, а FastAPI використовується для обробки HTTP-запитів і побудови API.

Наступною важливою функцією є накопичення та зберігання даних. Усі отримані вимірювання (струм, напруга, потужність, температура, події руху)

зберігаються у базі даних, яка може бути реалізована як реляційна база (наприклад, PostgreSQL); NoSQL база (наприклад, MongoDB); спеціалізована база часових рядів (InfluxDB). Зберігання історичних даних дозволяє аналізувати динаміку енергоспоживання; виявляти пікові навантаження; будувати прогнози; оцінювати ефективність енергозберігаючих заходів.

Ключовою особливістю інформаційно-аналітичного рівня є виконання складних алгоритмів обробки та оптимізації. На цьому етапі можуть реалізовуватися алгоритми виявлення аномалій, наприклад, різке зростання споживання; статистичний аналіз і агрегація даних; прогнозування на основі часових рядів; оптимізація режимів роботи обладнання; формування рекомендацій користувачу. На відміну від рівня локального контролю, хмарний сервер має значно більші обчислювальні ресурси, що дозволяє використовувати складні математичні моделі та навіть елементи машинного навчання. Крім того, хмарна платформа забезпечує доступ до даних через API. Завдяки FastAPI реалізується RESTful-інтерфейс, через який клієнтські додатки можуть отримувати історичні дані, аналітику, а також надсилати команди керування, що створює єдину точку доступу до системи.

Важливим аспектом є також масштабованість і надійність. Хмарні рішення дозволяють обробляти великі обсяги даних і підтримувати одночасну роботу великої кількості пристроїв. У випадку AWS це досягається автоматично, тоді як у приватних рішеннях потребує додаткової конфігурації. Додатково забезпечується резервне копіювання даних; захист інформації (шифрування, автентифікація); журналювання подій системи. Інформаційно-аналітичний рівень виконує роль інтелектуального центру системи, де відбувається глибока обробка даних, накопичення історії енергоспоживання та реалізація алгоритмів оптимізації, що дозволяє не лише здійснювати моніторинг, але й підвищувати енергоефективність шляхом прийняття обґрунтованих рішень на основі даних.

Рівень представлення даних є завершальним етапом функціонування системи моніторингу та енергозбереження, оскільки саме він забезпечує

взаємодію користувача з усією інфраструктурою. На цьому рівні результати обробки та аналізу даних подаються у зрозумілому та зручному для сприйняття вигляді, що дозволяє приймати обґрунтовані рішення щодо оптимізації енергоспоживання.

У розроблюваній системі цей рівень реалізовано у вигляді веб-інтерфейсу, створеного на базі бібліотеки React.js. Використання React.js дозволяє будувати сучасні односторінкові додатки (SPA), які відзначаються високою швидкістю роботи, динамічним оновленням контенту без перезавантаження сторінки та зручною структурою компонентів.

Основною функцією веб-інтерфейсу є візуалізація даних енергоспоживання. Зокрема, користувачу надаються графіки зміни споживання електроенергії у часі; діаграми розподілу навантаження; показники миттєвої та середньої потужності; інформація про температуру та інші параметри середовища; повідомлення про аномалії або перевищення встановлених порогів.

Візуалізація реалізується за допомогою інтерактивних дашбордів, які дозволяють користувачу аналізувати дані за різні часові інтервали; порівнювати показники; швидко виявляти неефективне використання енергії. Крім функції відображення, веб-інтерфейс забезпечує інструменти керування системою. Користувач може задавати порогові значення енергоспоживання; налаштовувати сценарії роботи обладнання; вмикати або вимикати виконавчі пристрої (через реле); змінювати параметри збору та обробки даних.

Усі команди, введені користувачем, передаються через API, реалізований, наприклад, на FastAPI, до хмарного сервера, а далі – до мікроконтролера. Таким чином забезпечується двосторонній зв'язок між користувачем і фізичним обладнанням у режимі реального часу.

Важливою характеристикою є адаптивність інтерфейсу, що дозволяє використовувати систему як на персональних комп'ютерах, так і на мобільних пристроях. Це розширює можливості віддаленого моніторингу та керування. Також веб-інтерфейс реалізує автентифікацію та авторизацію користувачів;

розмежування прав доступу; історію подій і дій користувача; сповіщення (наприклад, при аварійних ситуаціях або перевищенні лімітів). Рівень представлення даних виконує роль зручного інструменту взаємодії людини з системою. Він не лише відображає результати аналізу енергоспоживання, але й надає засоби активного керування, що дозволяє підвищити ефективність використання енергоресурсів і забезпечити гнучке налаштування системи відповідно до потреб користувача.

Структурна схема системи моніторингу та енергозбереження є узагальненим графічним відображенням її архітектури, яке демонструє склад основних компонентів та логічні зв'язки між ними. Вона дозволяє наочно представити принцип функціонування системи як єдиного цілого та зрозуміти взаємодію між апаратною (Hardware) і програмною (Software) складовими. У структурній схемі відображено основні рівні системи: рівень сенсорів, рівень локального контролю, мережевий рівень, хмарний (інформаційно-аналітичний) рівень та рівень представлення даних. Апаратна частина включає датчики, мікроконтролер та виконавчі пристрої (реле), які безпосередньо взаємодіють з фізичними процесами. Програмна частина охоплює алгоритми обробки даних, серверну логіку, бази даних та користувацький інтерфейс.

Інтеграція цих складових забезпечується за рахунок чітко організованих каналів передачі даних і протоколів взаємодії. Зокрема, дані від сенсорів надходять до мікроконтролера, де проходять первинну обробку, після чого передаються через мережу до хмарного сервера. У хмарі відбувається їх накопичення, аналіз і формування керуючих рішень, які можуть бути передані назад до системи.

Ключовим елементом структурної схеми є наявність зворотного зв'язку, який забезпечує не лише моніторинг, але й активне керування енергоспоживанням. Зокрема, хмарний сервер, аналізуючи отримані дані, може генерувати керуючі сигнали для виконавчих пристроїв. Ці сигнали передаються через мережу до мікроконтролера, який, у свою чергу, впливає на роботу реле та підключеного обладнання.

Реалізація зворотного зв'язку дозволяє впроваджувати інтелектуальні сценарії автоматизації. Наприклад, у випадку перевищення встановленого ліміту потужності система може автоматично відключити певні електроприлади або перевести їх у енергозберігаючий режим. Це забезпечує не лише контроль, але й оперативне реагування на зміну умов, що суттєво підвищує ефективність використання енергоресурсів.

Структурна схема відображає логіку побудови системи як інтегрованого комплексу, у якому апаратні та програмні компоненти працюють узгоджено. Наявність зворотного зв'язку перетворює систему з пасивного інструменту моніторингу на активну систему управління, здатну самостійно приймати рішення та оптимізувати енергоспоживання в режимі реального часу.

Алгоритм роботи системи передбачає циклічне опитування датчиків та прийняття рішень на основі встановлених енергетичних лімітів. Процес починається з ініціалізації периферійних пристроїв та встановлення зв'язку з MQTT-брокером.

При виявленні нераціонального споживання, наприклад, активне освітлення у порожній кімнаті або робота електрообігрівача у пікові години тарифного навантаження, алгоритм формує сигнал управління для відповідного реле. Такий підхід дозволяє досягти економії до 20 – 30 % електроенергії за рахунок автоматизації рутинних операцій.

Розглянемо детально функціональні блоки системи моніторингу та енергозбереження, які забезпечують повний цикл – від вимірювання параметрів до прийняття інтелектуальних рішень.

Блок вимірювання (SCT-013). Блок вимірювання призначений для отримання інформації про силу струму в електричному колі. Його основою є датчик SCT-013, який реалізує неінвазивний метод вимірювання, тобто без необхідності розриву провідника. Принцип роботи базується на явищі електромагнітної індукції. При протіканні змінного струму через провідник навколо нього утворюється змінне магнітне поле. Датчик, який має форму струмового трансформатора, охоплює провідник і фіксує це поле, індукуючи

струм у своїй вторинній обмотці. Отриманий сигнал є пропорційним до вимірюваного струму. Основними перевагами такого підходу є безпечність використання (гальванічна розв'язка); простота встановлення; можливість роботи без втручання в електромережу. Сформований аналоговий сигнал передається до мікроконтролера для подальшого оцифрування та обробки.

Блок передачі (ESP32-WiFi). Блок передачі реалізовано на базі мікроконтролера ESP32, який виконує роль комунікаційного шлюзу між апаратною частиною системи та хмарною інфраструктурою. Після збору та первинної обробки даних (фільтрації, нормалізації, розрахунку потужності) мікроконтролер здійснює інкапсуляцію даних у формат JSON (JavaScript Object Notation). Такий формат є універсальним, легким для передачі та зручним для подальшої обробки на сервері.

Приклад структури повідомлення: значення струму; значення напруги; розрахована потужність; температура; часові мітки.

Далі сформовані JSON-пакети передаються через Wi-Fi із використанням протоколу MQTT. Дані публікуються у відповідні тематичні канали (topics), наприклад,

`/energy/current`

`/energy/power`

`/environment/temperature`

Такий підхід забезпечує ефективну передачу даних із мінімальним трафіком; асинхронну взаємодію між компонентами; можливість масштабування системи.

Блок прийняття рішень, або Logic Engine, реалізований у вигляді програмного модуля на сервері та є ключовим інтелектуальним компонентом системи. Його основне завдання – аналіз отриманих даних та формування керуючих впливів на основі заданих правил і сценаріїв. У даному випадку алгоритм роботи базується на порівнянні поточного рівня енергоспоживання; часових інтервалів дії «нічного тарифу»; статусу присутності мешканців (за даними PIR-датчика).

Логіка прийняття рішень може бути описана наступним чином:

якщо споживання перевищує встановлений поріг у денний час – система може обмежити роботу частини обладнання;

якщо активний нічний тариф – допускається включення енергоємних пристроїв (наприклад, бойлера або пральної машини);

якщо в приміщенні відсутні люди – система автоматично вимикає освітлення або інші непотрібні навантаження.

Результатом роботи Logic Engine є формування керуючих команд, які передаються назад до мікроконтролера через MQTT або HTTP. Це забезпечує реалізацію зворотного зв'язку та автоматизацію процесу енергозбереження.

Взаємодія трьох розглянутих блоків забезпечує повний функціональний цикл системи: від точного вимірювання параметрів електромережі до інтелектуального керування енергоспоживанням з урахуванням зовнішніх умов та поведінки користувачів.

2.2 Вибір апаратних засобів та каналів передачі даних

Вибір апаратних компонентів для системи моніторингу та енергозбереження є одним із ключових етапів проектування, оскільки саме від нього залежать точність вимірювань, надійність функціонування та економічна доцільність системи. При виборі враховувалися такі критерії, як точність та стабільність вимірювань; сумісність із мікроконтролером; безпечність експлуатації; енергоефективність; вартість та доступність компонентів; можливість масштабування системи.

Зробимо обґрунтування вибору датчиків. Для реалізації функцій моніторингу було обрано набір сенсорів, які дозволяють комплексно оцінювати як параметри електромережі, так і стан навколишнього середовища.

Датчик струму SCT-013 використовується для вимірювання сили змінного струму. Його вибір обумовлений такими перевагами, як

неінвазивний принцип роботи (без розриву кола); гальванічна розв'язка, що підвищує безпеку; достатня точність для побутових задач; простота встановлення та експлуатації.

Датчик напруги ZMPT101B застосовується для вимірювання змінної напруги. Основними причинами вибору є наявність вбудованого трансформатора напруги; компактність; можливість калібрування; сумісність з аналоговими входами мікроконтролера.

Датчик температури та вологості DHT22 дозволяє контролювати параметри мікроклімату. Його перевагами є достатня точність вимірювання; цифровий інтерфейс передачі даних; низьке енергоспоживання; широка підтримка в IoT-проектах.

PIR-датчик руху використовується для визначення присутності людей у приміщенні. Його вибір обґрунтований низькою вартістю; простотою підключення; можливістю реалізації сценаріїв автоматизації (керування освітленням, обладнанням).

Таким чином, обраний набір сенсорів забезпечує необхідний рівень функціональності при мінімальних витратах.

Обґрунтуємо вибір мікроконтролера. У якості центрального керуючого елемента системи обрано мікроконтролер ESP32. Основними причинами вибору є:

- висока обчислювальна потужність (двоядерний процесор), що дозволяє реалізувати попередню обробку даних (Edge Computing);
- вбудовані модулі Wi-Fi та Bluetooth, що усуває необхідність у додаткових комунікаційних пристроях;
- наявність аналого-цифрових перетворювачів (АЦП) для роботи з аналоговими датчиками;
- низьке енергоспоживання;
- широка підтримка спільноти та бібліотек;
- доступна вартість.

ESP32 забезпечує не лише збір і обробку даних, але й реалізацію керуючих алгоритмів, що робить його оптимальним рішенням для системи «розумного будинку».

У розроблюваній системі моніторингу та енергозбереження для організації каналу передачі даних обрано технологію Wi-Fi, що зумовлено її технічними та практичними перевагами для IoT-рішень у побутових і промислових умовах.

Однією з ключових переваг Wi-Fi є його широке розповсюдження. У більшості житлових, офісних та виробничих приміщень вже існує розгорнута Wi-Fi інфраструктура у вигляді роутерів та локальних мереж, що дозволяє інтегрувати систему моніторингу без необхідності створення додаткової телекомунікаційної мережі. Таким чином, знижується вартість впровадження та спрощується монтаж системи.

Wi-Fi забезпечує достатньо високу пропускну здатність каналу зв'язку, що дозволяє передавати дані в режимі реального часу. У контексті системи енергомоніторингу це є важливим, оскільки необхідно оперативно передавати показники струму, напруги, потужності та стану датчиків. Висока швидкість також дозволяє одночасно передавати дані від кількох вузлів без суттєвих затримок.

Сучасні Wi-Fi мережі здатні підтримувати значну кількість підключених пристроїв, що є критично важливим для IoT-систем. Це дозволяє масштабувати систему шляхом додавання нових сенсорних вузлів без зміни базової інфраструктури. У результаті система може використовуватися як у невеликих приміщеннях, так і на рівні великих об'єктів.

Важливою технічною перевагою є те, що мікроконтролер ESP32 вже має вбудований Wi-Fi модуль, що повністю усуває потребу у використанні зовнішніх комунікаційних адаптерів. Це зменшує складність апаратної реалізації, скорочує кількість компонентів, знижує енергоспоживання та підвищує надійність системи. Крім того, вбудована підтримка Wi-Fi спрощує програмну реалізацію підключення до мережі та передачі даних.

Альтернативні технології (ZigBee, LoRa, Bluetooth) не були обрані через обмежену пропускну здатність, менший радіус дії (Bluetooth) або необхідність додаткової інфраструктури.

Для організації обміну даними обрано протокол MQTT, який є стандартом де-факто для IoT-систем. Перевагами MQTT є легковагість і мінімальний обсяг службових даних; підтримка моделі publish–subscribe; можливість роботи в нестабільних мережах; підтримка різних рівнів якості обслуговування (QoS); масштабованість. MQTT дозволяє ефективно передавати дані від мікроконтролера до хмарного сервера та реалізовувати зворотний зв'язок для керування системою.

У результаті проведеного аналізу обрано оптимальний набір апаратних засобів та каналів передачі даних для реалізації системи моніторингу та енергозбереження. Використання датчиків SCT-013, ZMPT101B, DHT22 та PIR забезпечує комплексний збір даних, а мікроконтролер ESP32 – їх ефективну обробку та передачу. Застосування технології Wi-Fi та протоколу MQTT дозволяє організувати надійний, масштабований і економічно ефективний обмін даними між компонентами системи. Обрані рішення забезпечують баланс між функціональністю, вартістю та простотою реалізації, що є важливим для впровадження системи у реальних умовах.

2.3 Проєктування програмного забезпечення та бази даних

Проєктування програмного забезпечення системи моніторингу та енергозбереження базується на принципах модульності, масштабованості та розподіленої обробки даних. Логіка роботи системи ґрунтується на безперервному циклі збору, обробки, аналізу даних та формування керуючих впливів. Розглянемо основні етапи роботи системи. Їх вісім, а саме, збір даних із сенсорів (струм, напруга, температура, присутність); попередня обробка

даних на мікроконтролері (фільтрація, обчислення потужності); передача даних на сервер через MQTT; збереження даних у базі; аналіз даних (Logic Engine); формування керуючих команд; передача команд до виконавчих пристроїв; відображення інформації користувачу.

Таблиця 2.1 – Основні функціональні модулі системи

№	Модуль	Функції
1	Сенсорний	Збір фізичних параметрів
2	Локальний контролер	Обробка та передача даних
3	Комунікаційний	Передача через MQTT
4	Серверний	Збереження та аналіз
5	Logic Engine	Прийняття рішень
6	Інтерфейс користувача	Візуалізація та керування

Розглянемо процес проектування бази даних. Для системи моніторингу енергоспоживання ключове значення має ефективне зберігання часових рядів (time-series data). Дані мають високу частоту надходження та прив'язку до часу, тому структура бази повинна забезпечувати швидкий запис і читання. До основних вимог до БД можна віднести підтримку великих обсягів даних; ефективну агрегацію (година, день, місяць); швидкий доступ до історії; масштабованість.

Опишемо структуру бази даних.

Таблиця 2.2 – Основна таблиця вимірювань (energy_data)

Поле	Тип даних	Опис
id	INT	Унікальний ідентифікатор
timestamp	DATETIME	Час вимірювання
current	FLOAT	Сила струму
voltage	FLOAT	Напруга
power	FLOAT	Потужність
temperature	FLOAT	Температура
presence	BOOLEAN	Наявність руху

У таблиці вимірювань Energy_data зберігаються всі вимірювання з сенсорів. Дані записуються постійно (наприклад, кожні 5–10 секунд), тому це типова таблиця часових рядів (time-series).

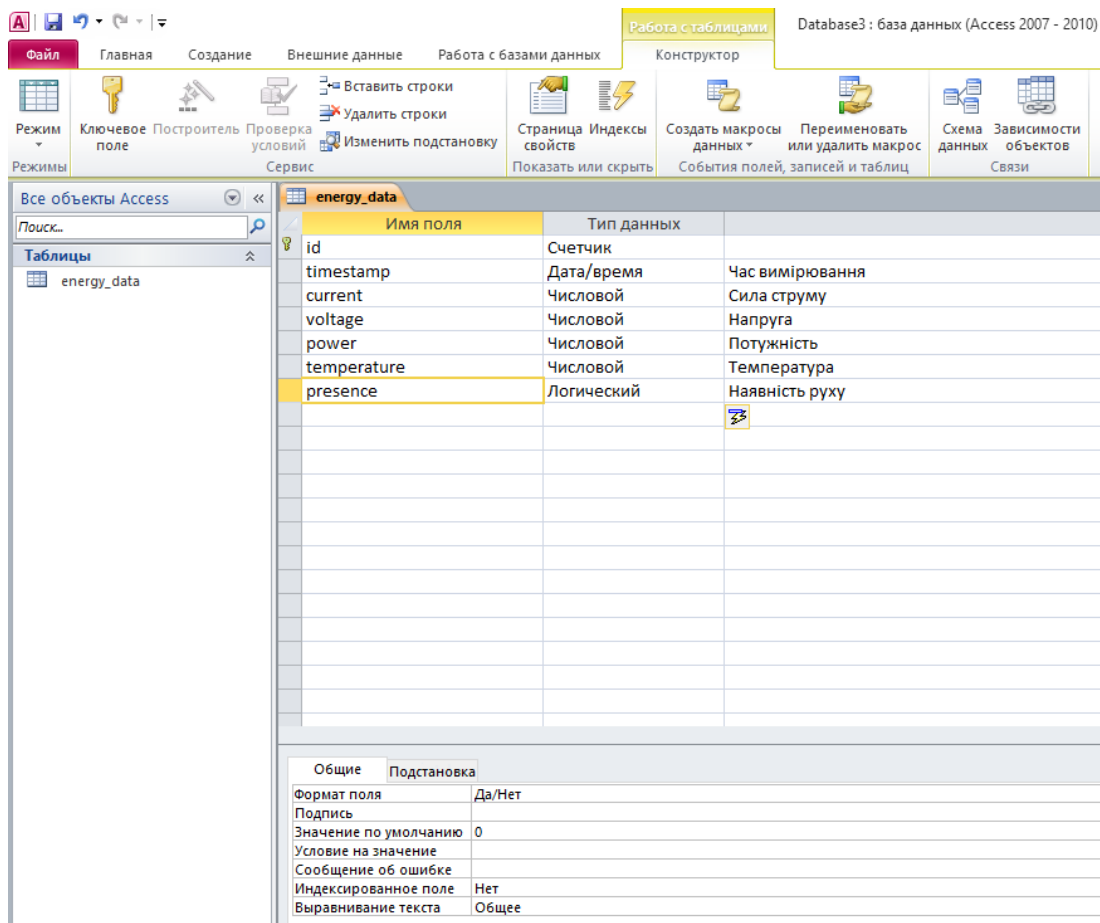


Рисунок 2.7 –Таблиця вимірювань energy_data

SQL приклад таблиці Energy_data

```
CREATE TABLE energy_data (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    timestamp DATETIME NOT NULL,  
    current FLOAT,  
    voltage FLOAT,  
    power FLOAT,  
    temperature FLOAT,  
    presence BOOLEAN  
);
```

Приклад заповнення таблиці Energy_data

```
INSERT INTO energy_data  
(timestamp, current, voltage, power, temperature, presence)  
VALUES  
(  
'2026-04-13 10:00:00', 5.2, 220, 1144, 23.5, 1),  
'2026-04-13 10:00:10', 5.0, 221, 1105, 23.6, 1),  
'2026-04-13 10:00:20', 2.1, 219, 460, 23.7, 0);
```

Дані приходять із ESP32.

Кожен рядок = одне вимірювання;

$power \approx current \times voltage$;

presence = 1 означає, що виявлено рух.

Таблиця 2.3 – Таблиця пристроїв (Devices)

Поле	Тип	Опис
id	INT	Ідентифікатор

Поле	Тип	Опис
name	VARCHAR	Назва пристрою
type	VARCHAR	Тип
status	BOOLEAN	Стан

Таблиця пристроїв Devices зберігає інформацію про всі підключені пристрої (реле, лампи, бойлер тощо).

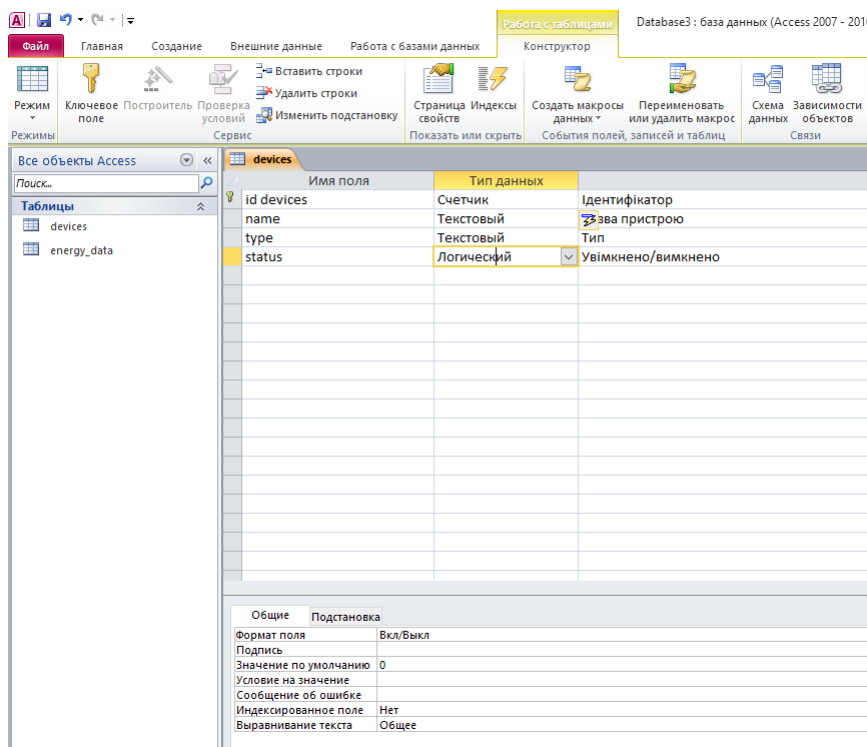


Рисунок 2.8 – Таблица пристроїв Devices

SQL приклад таблиці пристроїв Devices

```
CREATE TABLE devices (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    type VARCHAR(50),
```

```
status BOOLEAN
);
```

Приклад заповнення таблиці пристроїв Devices

```
INSERT INTO devices (name, type, status) VALUES
('Освітлення кухні', 'light', 1),
('Бойлер', 'heater', 0),
('Пральна машина', 'appliance', 0);
status = 1 → пристрій увімкнений
status = 0 → вимкнений
```

Таблиця Devices використовується для керування через веб-інтерфейс.

Таблиця 2.4 – Таблиця подій Events

Поле	Тип	Пояснення
id	INT	Ідентифікатор
timestamp	DATETIME	Час
event_type	VARCHAR	Тип події
description	TEXT	Опис

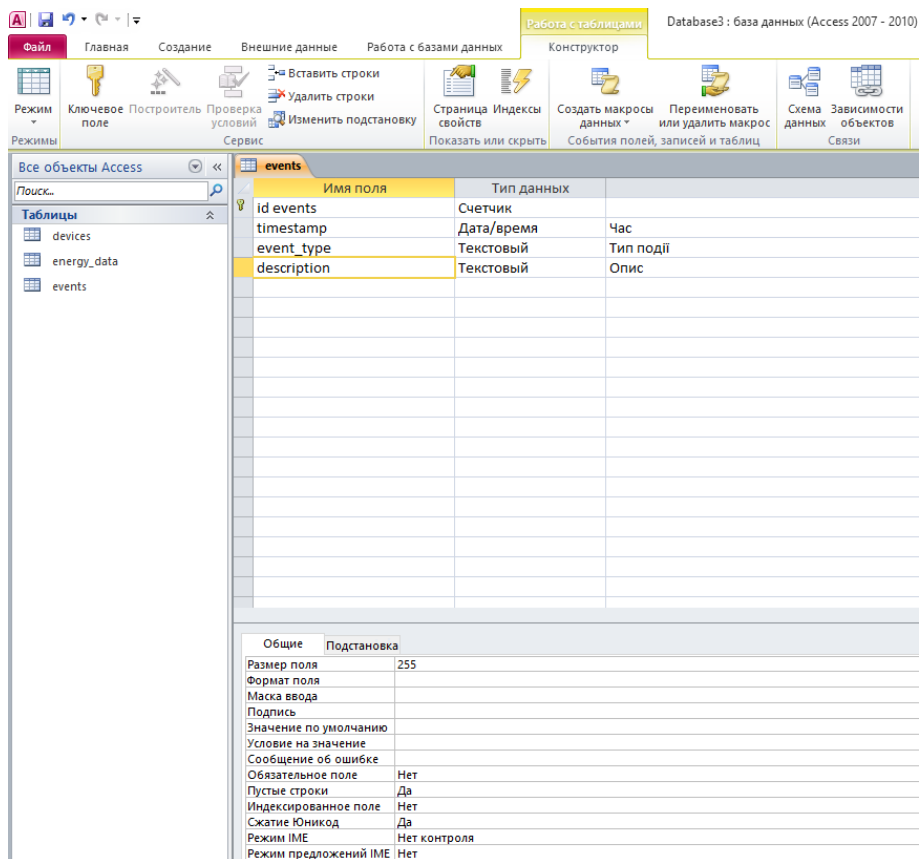


Рисунок 2.9–Таблица подій Events

Таблица події (Events) фіксує всі важливі події системи (перевищення потужності, вимкнення пристроїв тощо).

SQL приклад таблиці події Events

```
CREATE TABLE events (
    id INT AUTO_INCREMENT PRIMARY KEY,
    timestamp DATETIME,
    event_type VARCHAR(50),
    description TEXT
);
```

Приклад заповнення таблиці події Events

```
INSERT INTO events (timestamp, event_type, description) VALUES
('2026-04-13 10:05:00', 'OVERLOAD', 'Перевищення потужності 2 кВт'),
('2026-04-13 10:05:02', 'AUTO_OFF', 'Вимкнено бойлер'),
('2026-04-13 11:00:00', 'NIGHT_MODE', 'Активовано нічний тариф');
```

Таблиця події Events важлива для логування; допомагає аналізувати роботу системи; використовується для відображення повідомлень користувачу.

Розглянемо як дані реально потрапляють у БД.

Потік даних. ESP32 вимірює параметри. Формує JSON:

```
{
  "current": 5.2,
  "voltage": 220,
  "power": 1144,
  "temperature": 23.5,
  "presence": 1
}
```

Надсилає через MQTT

Сервер (FastAPI) приймає дані

Виконується SQL INSERT у Energy_data

У процесі розробки бази даних для системи моніторингу енергоспоживання важливо враховувати специфіку роботи з великими обсягами даних, які надходять у режимі реального часу.

Оскільки таблиця Energy_data є таблицею часових рядів, основні запити до неї виконуються за часовими інтервалами (наприклад, за день, годину або місяць). Без індексу база даних змушена переглядати всі записи, що значно

знижує продуктивність. Індекс дозволяє значно пришвидшити пошук даних; оптимізувати аналітичні запити; зменшити навантаження на сервер.

Індекс було створено на SQL. Запит з індексом дозволяє отримати швидкий доступ до потрібного діапазону:

```
CREATE INDEX idx_time ON energy_data(timestamp);
```

Запит без індексу потребує повного сканування таблиці:

```
SELECT * FROM energy_data  
WHERE timestamp BETWEEN '2026-04-01' AND '2026-04-02';
```

Сирі дані надходять дуже часто (наприклад, кожні 5 секунд), тому їх обсяг швидко зростає. Для аналізу зазвичай не потрібні всі значення, достатньо усереднених показників.

Агрегація дозволяє зменшити обсяг оброблюваних даних; спростити побудову графіків; підвищити швидкість аналітики.

На прикладі розглянемо різні типи агрегації, яка використовується для аналізу пікових навантажень; оптимізації використання «нічного тарифу».

Агрегація за днями на SQL

```
SELECT DATE(timestamp) AS day, AVG(power) AS avg_power  
FROM energy_data  
GROUP BY day;
```

Таблиця 2.4 – Таблиця результатів агрегації за днями

day	avg_power
2026-04-01	850 Вт

day	avg_power
2026-04-02	920 Вт

Агрегація за годинами на SQL

```
SELECT HOUR(timestamp) AS hour, AVG(power)
FROM energy_data
GROUP BY hour;
```

З часом таблиця Energy_data може містити мільйони записів, що негативно впливає на продуктивність. Для зменшення розміру основної таблиці; підвищення швидкості роботи системи; збереження історичних даних окремо використовуємо архівацію:

```
INSERT INTO energy_data_archive
SELECT * FROM energy_data
WHERE timestamp < '2025-01-01';
або
```

```
DELETE FROM energy_data
WHERE timestamp < '2025-01-01';
```

Щоб не виконувати складні запити щоразу, можна створити окремі таблиці з уже обчисленими значеннями, що дозволяє значно пришвидшити відображення графіків; зменшити навантаження на БД; підвищити масштабованість системи.

Таблиця агрегованих даних:

```
CREATE TABLE daily_stats (
```

```
date DATE,  
avg_power FLOAT,  
max_power FLOAT,  
min_power FLOAT  
);
```

Заповнення таблиці:

```
INSERT INTO daily_stats (date, avg_power, max_power, min_power)  
SELECT  
    DATE(timestamp),  
    AVG(power),  
    MAX(power),  
    MIN(power)  
FROM energy_data  
GROUP BY DATE(timestamp);
```

Для оптимізації роботи з даними використовуються індексація за полем часу; агрегаційні запити; архівація старих даних; використання спеціалізованих БД (InfluxDB).

Запропоновані підходи до організації бази даних є критично важливими для ефективної роботи системи моніторингу енергоспоживання. Індексація забезпечує швидкий доступ до даних, агрегація – ефективний аналіз, а архівація – стабільну продуктивність при великих обсягах інформації.

Застосування цих методів дозволяє створити масштабовану, надійну та продуктивну систему, здатну обробляти дані у режимі реального часу та забезпечувати якісну аналітику для прийняття рішень.

Висновок до розділу 2.

У другому розділі бакалаврської роботи було здійснено комплексне проєктування системи моніторингу та енергозбереження, що охоплює структурні, апаратні та програмні аспекти її реалізації.

Розроблено структурну та функціональну схеми системи, які відображають взаємодію основних компонентів: сенсорів, мікроконтролера, мережевої інфраструктури, хмарного сервера та користувацького інтерфейсу. Особливу увагу приділено організації зворотного зв'язку, що дозволяє не лише здійснювати моніторинг, а й реалізовувати автоматичне керування енергоспоживанням на основі отриманих даних.

Обґрунтовано вибір апаратних засобів та каналів передачі даних. Зокрема, використання датчиків струму, напруги, температури та присутності забезпечує комплексний контроль параметрів системи, а мікроконтролер ESP32 – ефективну обробку та передачу даних. Вибір технології Wi-Fi та протоколу MQTT дозволяє реалізувати надійний, масштабований та економічно доцільний обмін інформацією між компонентами системи.

Виконано проєктування програмного забезпечення та бази даних. Розроблено логіку роботи системи, яка базується на циклі збору, обробки, аналізу даних та формування керуючих впливів. Запропоновано структуру бази даних, оптимізовану для зберігання часових рядів, що забезпечує ефективну обробку великих обсягів інформації. Визначено підходи до реалізації користувацького інтерфейсу, який забезпечує візуалізацію даних та можливість дистанційного керування системою.

У результаті виконаного проєктування сформовано архітектуру системи моніторингу та енергозбереження, яка поєднує сучасні апаратні та програмні рішення. Запропоновані технічні підходи забезпечують ефективність, надійність та масштабованість системи.

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ВІДЛАДКА СИСТЕМИ ЕНЕРГОЗБЕРЕЖЕННЯ «РОЗУМНОГО БУДИНКУ»

3.1 Розробка програмно-апаратного прототипу та налаштування хмарного середовища

Практична реалізація системи автоматизації та енергозбереження передбачає створення фізичного прототипу пристрою на базі обраних апаратних засобів, налаштування мережевої інфраструктури та розгортання серверних потужностей для обробки даних у реальному часі.

Збірка апаратного прототипу здійснювалася на основі розробленої у розділі 2 структурної схеми. Основним обчислювальним вузлом виступає мікроконтролер ESP 32. Процес складання включав такі етапи, як монтаж сенсорів, підключення виконавчих механізмів, забезпечення живлення, налаштування мікроконтролера ESP 32.

Етап монтаж сенсорів є критичним для забезпечення точності вимірювань та захисту мікроконтролера. Оскільки ESP32 – це цифрова мікросхема, що працює з напругою до 3,3 В, пряме підключення силових датчиків може призвести до пошкодження портів або некоректних даних.

Датчик SCT-013 є трансформатором струму, який на виході генерує змінний струм (AC), пропорційний струму у фазному проводі. Оскільки вхід АЦП (ADC) мікроконтролера ESP 32 може приймати лише позитивну постійну напругу в діапазоні 0 – 3,3 В, виникають дві технічні проблеми напруга з датчика може перевищувати 3,3 В та полярність, тобто, змінний струм має негативну напівхвилю, яка може вивести з ладу вхід мікроконтролера. Для подолання цих проблем використовується схема узгодження (резистивний обтяжувач, дільний напруги, конденсатор). Резистивний обтяжувач (Burden resistor) перетворює вихідний струм датчика у напругу. Дільник напруги (R1, R2), який можна реалізувати за допомогою двох резисторів (зазвичай по 10 кОм), які створюють «віртуальну землю» на рівні 1,65 В (половина від 3,3 В), що зміщує синусоїду датчика вгору, щоб

вона повністю знаходилася в позитивній зоні. Конденсатор додається паралельно ділнику для згладжування пульсацій на опорній напрузі.

Датчик DHT 22 (Цифровий інтерфейс) використовує власний однопровідний протокол передачі даних (One-Wire), який потребує високої стабільності логічного рівня на лінії. Підтягувальний резистор (Pull-up) представляє собою резистор номіналом 4,7 кОм, що встановлюється між піном живлення (VCC) та сигнальним піном (Data). Підтягувальний резистор утримує лінію зв'язку у стані «Логічна одиниця» у моменти, коли датчик або контролер не передають дані. Без нього лінія стає чутливою до електромагнітних завад (антенною), що призводить до помилок читання Checksum Error або Timeout Error.

Таблиця 3.1 – Специфікація підключення компонентів

Компонент	Тип сигналу	Пін ESP32	Додаткові елементи	Призначення
SCT-013	Аналоговий	GPIO 34 (ADC1)	Резистори 10кОм, конд. 10мкФ	Вимірювання потужності
DHT22	Цифровий	GPIO 4	Резистор 4,7 кОм	Клімат-контроль

При монтажі SCT-013 важливо дотримуватися правила: датчик одягається тільки на один провід (фазу або нуль). Якщо одягнути на цілий кабель (де є і фаза, і нуль), магнітні поля взаємно компенсуються, і датчик покаже нуль.

Підключення виконавчих механізмів. Для реалізації функції керування електроприладами в системі «Розумний будинок» використовується блок виконавчих механізмів. Оскільки мікроконтролер ESP32 є слабкострумним пристроєм, він не може безпосередньо керувати потужним навантаженням (освітленням, обігрівачами). Для цього застосовується проміжний комутаційний елемент — релейний модуль.

Двоканальний релейний модуль підключено до цифрових виходів GPIO 26 та GPIO 27 мікроконтролера ESP 32. Модуль працює за принципом «логічного нуля» (Low Level Trigger) або «логічної одиниці» (High Level Trigger), залежно від модифікації. Коли на відповідному GPIO-виході з'являється керуючий сигнал, відкривається транзисторний ключ на модулі, який замикає коло живлення котушки реле, що призводить до механічного замикання силових контактів, через які живиться побутовий прилад (220 В).

Критично важливим аспектом при автоматизації є захист мікроконтролера від електромагнітних завад, що виникають під час комутації індуктивного навантаження (двигунів, блоків живлення). Це забезпечується шляхом використання окремого джерела 5 В. Котушки реле під час спрацювання споживають значний струм (близько 70 – 100 мА на один канал). Якщо живити їх від стабілізатора ESP 32, це призведе до просідання напруги та раптового перезавантаження контролера.

Реалізується поділ живлення. Для реалізації повної ізоляції на релейному модулі знімається перемичка VCC-JDVCC. Живлення логічної частини (VCC) підключається до 3,3 В ESP 32, а живлення котушок (JD-VCC) – до зовнішнього джерела 5 В. Модуль оснащений оптопарами, які передають сигнал керування за допомогою світлового променя. Таким чином, електричний зв'язок між високовольтною частиною та мікроконтролером повністю відсутній.

Таблиця 3.2 – Параметри підключення виконавчих механізмів

Параметр	Значення / Опис
Тип модуля	2-channel Relay Module (з опторозв'язкою)
Керуюча напруга (Logic)	3,3 В (від ESP32)
Напруга котушок (Coil)	5 В (зовнішнє джерело)
Макс. комутована напруга	250 В змінного струму (AC)
Макс. комутований струм	10 А

При вимкненні реле в його котушці виникає зворотний сплеск напруги (електрорушійна сила самоіндукції). Для захисту схеми на модулі паралельно котушці встановлено захисний діод (зворотний діод), який гасить цей імпульс, не даючи йому пошкодити транзисторний ключ та вплинути на стабільність Wi-Fi з'єднання контролера.

Застосування схеми з роздільним живленням та оптоізоляцією дозволяє системі стабільно працювати в умовах високого рівня індустріальних завод. Це гарантує, що увімкнення потужного обігрівача або кондиціонера не спричинить «зависання» програмного забезпечення ESP 32 або розриву зв'язку з хмарним сервером.

Забезпечення живлення. Для стабільної роботи системи автоматизації, особливо в умовах постійного підключення до мережі Wi-Fi та використання аналогових сенсорів, критично важливо забезпечити якісне електроживлення. Оскільки компоненти системи мають різні вимоги до напруги та струму, схема живлення повинна бути надійною та захищеною від перепадів.

Оскільки система призначена для цілодобового моніторингу, її живлення реалізовано від стаціонарної мережі змінного струму 220 В. Процес перетворення та розподілу енергії організовано наступним чином:

Перетворення напруги → Розподіл живлення (5 В та 3,3 В) → Фільтрація завад та стабілізація.

Центральним елементом вузла живлення є імпульсний стабілізований блок живлення (БЖ). Він перетворює вхідну напругу мережі (220 В, 50 Гц) у стабільну напругу 5 В постійного струму (DC). Такі блоки живлення мають високий ККД (понад 80 %), малі габарити та вбудований захист від короткого замикання та перевантаження. Обрано БЖ із вихідним струмом не менше 2 А, що забезпечує необхідний запас потужності для пікових навантажень ESP 32 (під час передачі даних по Wi-Fi споживання може сягати 500 мА) та одночасного спрацювання котушок релейного модуля.

Система використовує дворівневу схему розподілу напруги. Шина 5 В використовується для живлення котушок реле та датчика DHT 22, що дозволяє уникнути просідання напруги на самому мікроконтролері. Шина 3,3 В: ESP32 потребує саме такої напруги для роботи ядра та логіки. Перетворення 5 В → 3,3 В здійснюється вбудованим у плату DevKit регулятором напруги (LDO-стабілізатор).

Для забезпечення точності роботи АЦП (при знятті показників з датчика струму SCT-013) у схемі передбачено додаткову фільтрацію з використанням електролітичних або керамічних конденсаторів. Електролітичні конденсатори встановлені на вході живлення контролера для компенсації низькочастотних пульсацій. Керамічні конденсатори (0,1 мкФ) розташовані безпосередньо біля виводів живлення сенсорів для поглинання високочастотних шумів, що виникають під час роботи Wi-Fi модуля.

Для безпечної експлуатації в умовах «Розумного будинку» вузол живлення оснащений запобіжником на вході 220 В для захисту від критичних несправностей; варистором для захисту від високовольтних імпульсних

стрибків у загальній мережі; корпусною ізоляцією, що унеможливило випадковий контакт користувача з високовольтними лініями.

Таблиця 3.3 – Енергетичні характеристики компонентів

Компонент системи	Номінальна напруга	Макс. споживання струму
ESP32 (Wi-Fi активний)	3,3 В	500 мА
Релейний модуль (2 канали)	5 В	150 - 200 мА
Датчик DHT22	3,3 - 5 В	2,5 мА
Схема зміщення SCT-013	3,3 В	< 1 мА
Загальне (пікове)	—	~ 750 - 850 мА

Обрана схема живлення з використанням стабілізованого джерела 5 В та розділенням ліній живлення логіки та силових елементів гарантує відсутність «зависань» системи та високу точність вимірювання енергоспоживання, що є базовою вимогою для систем класу АКІТ.

Після складання апаратної частини було виконано налаштування мікроконтролера ESP 32. Основними етапами конфігурації є ініціалізація GPIO-пінів; налаштування аналого-цифрового перетворювача (АЦП); підключення до Wi-Fi мережі; налаштування MQTT-клієнта; реалізація алгоритмів обробки даних.

Мікроконтролер ESP 32 виконує роль локального інтелектуального вузла (Edge Computing), реалізуючи такі основні функції, як зчитування

даних із сенсорів; фільтрація та обчислення потужності; формування JSON-повідомлень; передача даних на сервер; прийом команд керування.

Для розробки вбудованого програмного забезпечення (Firmware) обрано комбінований підхід, який поєднує зручність Arduino IDE та потужність низькорівневих бібліотек ESP-IDF, що дозволяє використовувати готові рішення для стандартних задач і одночасно забезпечувати високу продуктивність системи.

Програмне забезпечення мікроконтролера ESP32 є «інтелектуальним центром» локального вузла автоматизації. Воно відповідає за перетворення аналогових сигналів у цифрові дані та їх надійну доставку до хмарної платформи.

Розробка здійснювалася в Arduino IDE з використанням ядра ESP32 Core. Це рішення дозволяє інтегрувати компоненти ESP-IDF (IoT Development Framework), такі як FreeRTOS – для розпаралелювання задач (наприклад, одночасне зчитування датчиків та підтримка Wi-Fi з'єднання); Non-volatile Storage (NVS) – для збереження налаштувань Wi-Fi та MQTT навіть після вимкнення живлення.

На початковому етапі роботи (функція `setup()`) прошивка виконує конфігурацію апаратних ресурсів ADC та GPIO.

ADC (АЦП). Налаштування 12-бітного аналого-цифрового перетворювача для роботи з датчиком SCT-013. Встановлюється відповідна роздільна здатність та атенюація (ослаблення) для коректного зчитування напруги до 3,3 В.

GPIO. Визначення пінів для релейного модуля як виходів (OUTPUT) та піна для DHT 22 як входу з підтримкою цифрового протоколу.

Для забезпечення безперебійного моніторингу реалізовано логіку «самовідновлення» з'єднання, а саме, контролер підключається до локальної мережі у режимі STA (Station). У основному циклі програми реалізовано перевірку: якщо зв'язок втрачено (`WiFi.status() != WL_CONNECTED`), система

ініціює процедуру повторного підключення без зупинки алгоритмів безпеки (наприклад, реле все одно спрацює при перевантаженні).

Програмне забезпечення реалізує протокол MQTT для асинхронного обміну даними. Реалізується така структура повідомлень: Дані з датчиків збираються у структуру, яка потім серіалізується у формат JSON, що забезпечує легку інтерпретацію даних на сервері. Здійснюється Pub/Sub модель: Прошивка не лише відправляє телеметрію, а й постійно «слухає» топик команд. При отриманні команди від хмари, обробник (Callback) миттєво змінює стан GPIO, керуючи реле.

Таблиця 3.4 – Основні функціональні блоки прошивки

Модуль	Опис функцій
Energy Monitor	Зняття 1000 вибірок з АЦП, розрахунок I_{RMS} та активної потужності.
Climate Sense	Опитування DHT22 та перевірка цілісності даних (CRC).
Network Manager	Підтримка Wi-Fi та синхронізація часу через NTP-сервери.
MQTT Engine	Формування пакетів, публікація даних та обробка вхідних команд.

Програмне забезпечення розроблено з урахуванням специфіки систем автоматизації: мінімізовано використання блокуючих затримок (delay()), що

забезпечує високу швидкість реакції системи на критичні події. Використання стандартних бібліотек у поєднанні з можливостями ESP-IDF гарантує стабільну роботу пристрою у режимі 24/7.

Процес ініціалізації є початковим етапом роботи прошивки, під час якого мікроконтролер ESP32 конфігурує внутрішні регістри та встановлює зв'язок із зовнішніми модулями.

Для моніторингу енергоспоживання через датчик SCT-013 критично важливою є правильна конфігурація АЦП, а саме, роздільна здатність, атенюація, калібрування.

Роздільна здатність реалізується шляхом встановлення рівня 12 біт (значення від 0 до 4095), що дозволяє досягти високої дискретності вимірювань і необхідно для точного розрахунку середньоквадратичного значення струму (I_{RMS}). Атенюація (Attenuation) досягається через налаштування вхідного дільника АЦП на рівень 11 дБ, що дозволяє мікроконтролеру зчитувати напругу в повному діапазоні (до 3.3 В) без ризику пошкодження порту. Калібрування передбачає програмне врахування характеристики нелінійності АЦП ESP 32 для мінімізації похибки при низьких значеннях напруги («мертва зона» АЦП).

Для роботи з цифровими компонентами виконується розподіл функцій GPIO-виводів. Для релейного модуля розподіл здійснюється наступним чином: порти ініціалізуються в режимі OUTPUT. При завантаженні системи встановлюється безпечний стан (зазвичай логічна одиниця для модулів з Low Level Trigger), щоб уникнути довільного ввімкнення потужних приладів під час перезавантаження контролера. Для сенсора DHT22 використовується цифровий інтерфейс з ініціалізацією протоколу обміну даними, що базується на часових затримках (Single-Bus).

Мережева стабільність забезпечується реалізацією специфічного алгоритму підключення, який запобігає «зависанню» системи при збоях роутера. ESP32 налаштовується як Station (STA). Замість одноразової спроби підключення у блоці `setup()`, реалізовано перевірку стану мережі в основному

циклі loop(). Якщо системна функція WiFi.status() повертає помилку з'єднання, контролер автоматично ініціює процедуру WiFi.reconnect(). Процес перепідключення виконується асинхронно. Це означає, що пристрій продовжує виконувати локальні алгоритми автоматизації (наприклад, вимкнення світла за датчиком руху) навіть у момент відсутності доступу до Інтернету.

Таблиця 3.5 – Параметри конфігурації периферійних модулів

Модуль	Режим / Функція	Характеристика
ADC1 (CH4)	Analog Input	12-bit, 11dB attenuation
GPIO 26, 27	Digital Output	Керування реле (Active Low)
GPIO 4	Digital I/O	Протокол DHT (One-Wire)
Wi-Fi Radio	Station Mode	WPA2 PSK, Auto-reconnect enabled

Такий підхід до ініціалізації гарантує, що система енергозбереження буде готова до роботи протягом 2–3 секунд після подачі живлення, а механізм автоматичного перепідключення забезпечить автономність та надійність передачі даних у хмарне сховище без втручання користувача.

Для забезпечення надійного двостороннього зв'язку між мікроконтролером ESP 32 та хмарним сервером у роботі використано бібліотеку PubSubClient. Вона реалізує модель взаємодії «Видавець – Підписник» (Publish/Subscribe), що є оптимальною для систем з обмеженими ресурсами.

Процес передачі даних від датчиків енергомоніторингу до хмари відбувається шляхом публікації повідомлень у специфічні гілки (топіки) брокера. Мікроконтролер зчитує аналогові дані з датчика струму, розраховує потужність та формує JSON-пакет. Використовується метод `client.publish(topic, payload)`, де `topic` – це шлях (наприклад, `home/energy/sensor1`), а `payload` – рядок з даними. Публікація здійснюється або за таймером (наприклад, раз на 10 секунд), або при значній зміні показника (Threshold-based reporting), що дозволяє економити трафік та енергію.

Для отримання команд від користувача (наприклад, вимкнення реле через Dashboard), ESP 32 виступає як підписник. При ініціалізації пристрій підписується на топик команд (наприклад, `home/energy/relay1/set`). Брокер автоматично пересилає повідомлення клієнту, щойно воно з'являється в системі. Бібліотека використовує функцію зворотного виклику `callback()`. Коли надходить повідомлення, програма перериває основний цикл, аналізує зміст пакету («ON» або «OFF») та змінює стан відповідного GPIO-піна.

Ключовою функцією `PubSubClient` у межах даної роботи є підтримка постійного з'єднання, що забезпечується наступним чином:

Функція `client.loop()` викликається в основному циклі програми (`void loop`). Вона підтримує зв'язок з брокером, обробляє вхідні повідомлення та відправляє «heartbeat» пакети, щоб сервер не розірвав з'єднання.

Реалізовано алгоритм автоматичного відновлення підписки після втрати Wi-Fi сигналу, що забезпечує відмовостійкість системи автоматизації.

Таблиця 3.6 – Структура MQTT-топиків системи

Топік	Напрямок	Опис даних
<code>smart_home/power/telemetry</code>	ESP32 → Хмара	Поточна потужність,

Топік	Напрямок	Опис даних
		напряга, струм (JSON)
smart_home/relay/state	ESP32 → Хмара	Підтвердження фактичного стану реле
smart_home/relay/command	Хмара → ESP32	Команди управління (1 – увімкнути, 0 – вимкнути)

Використання бібліотеки PubSubClient дозволяє мінімізувати навантаження на процесор ESP 32 та забезпечує миттєву реакцію системи на команди користувача (Latency < 200 мс), що є критично важливим для оперативного управління енергоспоживанням.

Серверна частина (Backend) виступає інтелектуальним ядром системи, що координує потоки даних між апаратним рівнем, базою даних та інтерфейсом користувача. Для забезпечення стабільної роботи системи моніторингу на сервер покладено такі завдання, як інтеграція з MQTT, взаємодія з БД; Logic Engine (Модуль логіки); REST API Endpoints. Інтеграція з MQTT відбувається наступним чином. Сервер виступає в ролі клієнта, який підписується на топіки телеметрії. При надходженні нового повідомлення він десеріалізує JSON-пакет і готує дані для запису. Взаємодія з БД передбачає виконання асинхронних записів у PostgreSQL (TimescaleDB), що дозволяє уникати блокування потоків при інтенсивному надходженні даних. Logic Engine (Модуль логіки) – програмний модуль, що аналізує вхідні параметри (наприклад, перевищення порогу потужності) і автоматично генерує керуючі MQTT-повідомлення для реле. REST API Endpoints представляє собою набір

кінцевих точок для передачі історичних даних на фронтенд (React-додаток) та отримання команд від користувача.

Процес розробки та впровадження Backend-частини розділено на логічні етапи: створення архітектури API, інтеграція з базою даних, реалізація обробника MQTT, система безпеки та автентифікації, тестування та налагодження. Створення архітектури API полягає у розробці маршрутів (Endpoints). Наприклад, GET /history для отримання графіків та POST /control для зміни стану реле. Інтеграція з базою даних передбачає використання ORM (наприклад, SQLAlchemy або Tortoise), що забезпечує безпечну роботу з таблицями та автоматичну міграцію схем. Реалізація обробника MQTT здійснюється шляхом налаштування фонового процесу, який постійно підтримує зв'язок із брокером Mosquitto та обробляє повідомлення у реальному часі. Система безпеки та автентифікації передбачає впровадження протоколу OAuth2 з використанням JWT-токенів. Це гарантує, що лише авторизований користувач (Анастасія або інші члени родини) може надсилати команди на вимкнення приладів. Тестування та налагодження реалізується через перевірку цілісності даних при високому навантаженні та стрес-тестування логіки Logic Engine (імітація критичних стрибків споживання).

Таблиця 3.7 – Опис основних API-маршрутів системи

Метод	Маршрут	Опис функції
GET	/api/v1/current	Отримання миттєвих показників (кВт, °C)
GET	/api/v1/history	Запит архівних даних для побудови графіків

Метод	Маршрут	Опис функції
POST	/api/v1/device/control	Відправка команди на перемикання реле
POST	/api/v1/auth/login	Авторизація користувача та видача токена

Використання FastAPI дозволило створити високопродуктивну серверну частину з асинхронною обробкою запитів, що забезпечує мінімальну затримку (Latency) між виявленням перевищення ліміту енергоспоживання та спрацюванням виконавчого механізму.

Користувацький інтерфейс системи автоматизації розроблений як сучасний веб-додаток, що виконує роль центрального вузла візуалізації та управління. Основна мета інтерфейсу – конвертувати складні телеметричні дані, отримані від мікроконтролера ESP32 та збережені в базі даних TimescaleDB, у зрозумілу для користувача форму.

Веб-додаток спроектований за модульним принципом, що дозволяє паралельно відображати критично важливі параметри «Розумного будинку»:

Моніторинг енергоспоживання в реальному часі відображає поточну потужність (у Вт або кВт), що споживається всіма приладами об'єкта. Дані оновлюються автоматично без перезавантаження сторінки завдяки використанню протоколу WebSockets або періодичних API-запитів.

Кліматичні показники реалізуються через відображення температури та вологості, отриманих з датчика DHT22, що дозволяє користувачу аналізувати кореляцію між споживанням енергії (наприклад, роботою обігрівача) та зміною температури. Статус системи можна побачити на візуальному індикаторі, що підтверджує наявність зв'язку з MQTT-брокером та активність локального контролера. Це критично для забезпечення впевненості у працездатності алгоритмів захисту та автоматизації.

Графічне представлення даних реалізоване завдяки використанню бібліотек візуалізації (наприклад, Recharts або Chart.js) для побудови ретроспективних графіків споживання.

На рисунку 3.1 представлено графічний макет розробленого інтерфейсу, який складається з таких ключових зон, як панель поточних показників, інформаційні блоки стану, графік зміни енергоспоживання.

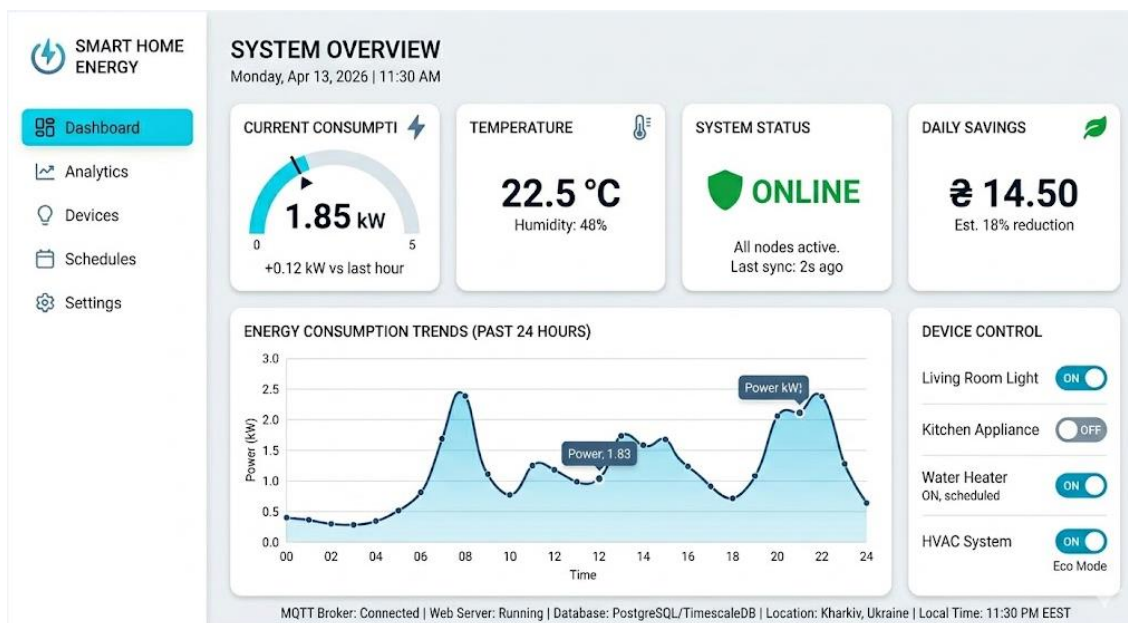


Рисунок 3.1 – Інтерфейс користувача системи моніторингу енергоспоживання

Панель поточних показників розташована у верхній частині додатка для миттєвої оцінки навантаження на електромережу. Використовує цифрові індикатори та колірну диференціацію (наприклад, червоний колір при наближенні до критичного ліміту).

Інформаційні блоки стану представлено картками, що містять додаткові дані про стан виконавчих механізмів (реле) та навколишнього середовища. Кожна картка забезпечує швидкий доступ до розширеної статистики.

Графік зміни енергоспоживання зображено у вигляді гістограми або лінійного графіку, що відображає профіль навантаження за обраний період

(годину, добу, місяць). Це головний інструмент для виявлення енергоємних періодів та планування переходу на нічний тариф.

Реалізований користувацький інтерфейс є невід'ємною частиною системи енергозбереження. Він дозволяє оперативно приймати рішення, забезпечує зручну взаємодію користувача із системою, прозорість процесів автоматизації та надає інструментарій для свідомого управління енергетичними ресурсами будинку, що повністю відповідає концепції Internet of Things.

3.2 Алгоритмічне забезпечення функціонування системи в різних режимах роботи

Алгоритмічне забезпечення системи визначає інтелектуальну поведінку контролера ESP32 та його взаємодію з виконавчими механізмами на основі даних від сенсорів та команд із хмарного сервера. Ключовою особливістю розробленого програмного забезпечення є підтримка декількох режимів роботи, що дозволяють гнучко налаштувати стратегію енергозбереження.

Система підтримує три основні режими, перемикання між якими здійснюється через MQTT-повідомлення від користувача або автоматично за розкладом на сервері, а саме, режим «День/Ніч» (Тарифний), режим «Економія» (Smart Eco), режим «Ручне керування». Кожен режим має свою логіку, спрямовану на оптимізацію витрат електроенергії або зручність користувача. Перемикання між режимами здійснюється за ієрархічним принципом: команда користувача через MQTT має найвищий пріоритет, тоді як автоматичні сценарії виконуються сервером згідно з налаштованими алгоритмами.

Режим «День/Ніч» (Тарифний) орієнтований на мінімізацію фінансових витрат користувача за наявності багатотарифного лічильника електроенергії. Серверна частина (FastAPI) відстежує поточний час. У період дії «нічного» тарифу (наприклад, з 23:00 до 07:00), коли вартість електроенергії нижча, система автоматично вмикає енергоємні прилади (бойлер, теплову підлогу). У «денний» період система впроваджує жорсткі ліміти навантаження, запобігаючи одночасному ввімкненню кількох потужних споживачів.

Режим «Економія» (Smart Eco) є найбільш інтелектуальним режимом, який базується на стані навколишнього середовища. Режим Smart Eco використовує дані з датчиків присутності (PIR) та освітленості. Якщо в приміщенні протягом певного часу (наприклад, 10 хвилин) не зафіксовано руху, система вважає, що споживач забув вимкнути світло або локальні електроприлади, і автоматично розмикає відповідні реле. Smart Eco дозволяє уникнути «холостого» споживання енергії, коли користувач відсутній у зоні дії приладів.

Режим «Ручне керування» є режимом прямого доступу, що вимикає автоматичну логіку. Користувач через Dashboard (веб-інтерфейс) самостійно змінює стан реле (On/Off). Це необхідно для ситуацій, які не вписуються в стандартні алгоритми (наприклад, прийом гостей або проведення ремонтних робіт). Навіть у режимі «Ручне керування» система продовжує знімати показники телеметрії та відображати їх на графіках, але не втручається в роботу виконавчих механізмів без прямої команди.

Таблиця 3.8 – Параметри спрацювання режимів

Режим	Джерело команди	Пріоритет	Керуючий фактор
День/Ніч	Серверний розклад	Середній	Астрономічний час

Режим	Джерело команди	Пріоритет	Керуючий фактор
Економія	Локальні датчики	Високий	Наявність руху / Рівень люксів
Ручне	MQTT від користувача	Найвищий	Суб'єктивне рішення

Механізм перемикання (Implementation details) на програмному рівні реалізовано через State Machine (Скінченний автомат). Використання моделі скінченного автомата дозволяє системі перебувати лише в одному чітко визначеному стані в кожен момент часу, що виключає конфлікти між різними алгоритмами керування. На рівні прошивки ESP32 визначено перелічуваний тип даних enum SystemMode, який містить три стани: DAY_NIGHT, ECONOMY та MANUAL. Глобальна змінна currentMode зберігає поточний стан автомата. Кожен стан має свій набір дозволених переходів та функцій обробки. Це гарантує, що, наприклад, у режимі «Ручне керування» датчик руху не зможе випадково вимкнути світло. Система завжди знає, як реагувати на зовнішні події (дані сенсорів), виходячи з поточного значення currentMode.

Процес перемикання режиму відбувається за ланцюжком (див. таблиця 3.9).

Таблиця 3.9 – Шлях керуючого сигналу (Data Flow)

Джерело дії	Ланцюг перемикання
Дія користувача	Користувач натискає кнопку в веб-інтерфейсі (наприклад, перехід у режим «Економія»)
Backend	Сервер отримує HTTP-запит, перевіряє права доступу та

Джерело дії	Ланцюг перемикання
(FastAPI)	публікує повідомлення (наприклад, рядок "ECONOMY") у MQTT-топик home/settings/mode
Broker (Mosquitto)	Брокер миттєво пересилає це повідомлення всім підписаним пристроям
ESP32 (MQTT Callback)	Функція зворотного виклику на мікроконтролері отримує повідомлення, порівнює його з ключовими словами та оновлює змінну currentMode.

У головному циклі програми (void loop()) реалізована конструкція вибору switch(currentMode). Залежно від значення змінної, мікроконтролер викликає відповідний блок функцій.

Фрагмента коду обробки (Callback) на C++

```
void callback(char* topic, byte* payload, unsigned int length) {
    String message = "";
    for (int i = 0; i < length; i++) message += (char)payload[i];

    if (String(topic) == "home/settings/mode") {
        if (message == "ECONOMY") currentMode = ECONOMY;
        else if (message == "DAY_NIGHT") currentMode = DAY_NIGHT;
        else if (message == "MANUAL") currentMode = MANUAL;

        Serial.print("System mode changed to: ");
        Serial.println(message);
    }
}
```

Після зміни локальної змінної, ESP32 відправляє назад у топик статусу (наприклад, home/status/current_mode) підтвердження. Це дозволяє інтерфейсу

користувача відобразити актуальний стан системи, підтверджуючи, що команда була успішно отримана та застосована «в залізі».

Реалізація через State Machine робить систему гнучкою до розширення: для додавання нового режиму (наприклад, «Відпустка») достатньо додати новий стан в enum та відповідну гілку в switch-case, не змінюючи при цьому логіку роботи вже існуючих режимів. Це забезпечує високу надійність (Reliability) ПЗ, що є критичним для систем автоматизації.

Розглянемо реалізацію логіки «в залізі». Основним інструментом реалізації логіки на рівні мікроконтролера є функція зворотного виклику (callback), яка обробляє вхідні команди, та головний цикл (loop), що виконує перевірку умов у реальному часі. Кожна ітерація циклу включає зчитування поточного значення потужності $P_{current}$ та перевірку прапора активного режиму. Якщо обрано режим «Економія», логіка виконує перевірку стану датчика PIR. Лістинг функції, що відповідає за обробку режимів та керування релейним модулем, на мові C++

```
void manageEnergyModes(float currentPower, bool motionDetected) {
    switch (currentMode) {
        case MODE_ECO:
            // Якщо руху немає понад 5 хвилин - вимикаємо навантаження
            if (!motionDetected && (millis() - lastMotionTime > 300000)) {
                digitalWrite(RELAY_PIN, HIGH); // Вимкнути (Active Low)
                statusMsg = "Eco: Auto-off due to no motion";
            } else if (currentPower > MAX_ECO_LIMIT) {
                digitalWrite(RELAY_PIN, HIGH); // Захист від перевантаження
            }
            break;

        case MODE_NIGHT:
            // Робота за розкладом або зниженим порогом
```

```

if (isNightTariff) {
    digitalWrite(RELAY_PIN, LOW); // Дозволити роботу приладів
} else if (currentPower > MAX_DAY_LIMIT) {
    digitalWrite(RELAY_PIN, HIGH); // Обмеження споживання вдень
}
break;

case MODE_MANUAL:
    // Стан реле змінюється лише через зовнішню команду MQTT
    break;
}
}

```

У системах автоматизації на базі мікроконтролерів (зокрема ESP32), де одночасно відбувається зчитування даних, керування реле та підтримка мережевого зв'язку, критично важливо правильно розподілити ресурси процесора. Для забезпечення стабільної роботи системи в режимі реального часу (Real-Time) програмне забезпечення реалізовано з використанням механізмів переривань (Interrupts) та багатозадачності FreeRTOS, що інтегрована в ядро ESP32.

Переривання (Hardware Interrupts) дозволяють мікроконтролеру миттєво реагувати на зовнішні події, не чекаючи завершення основного циклу програми.

Датчик руху (PIR) підключений до GPIO-піна з підтримкою переривання по фронту сигналу (RISING). Як тільки фіксується рух, виконується функція-обробник (ISR), яка оновлює мітку часу lastMotionTime. Якщо в системі передбачено апаратний захист, переривання дозволяє знеструмити об'єкт за мілісекунди, що запобігає пошкодженню обладнання.

Оскільки ESP32 має два ядра, задачі розділені за рівнем важливості. Це гарантує, що важкі обчислення або затримки в Wi-Fi мережі не завадять

критичному керуванню. Для забезпечення надійності алгоритм реалізовано з урахуванням пріоритетності (див. таблицю 3.10).

Ієрархія пріоритетів (від вищого до нижчого).

Таблиця 3.10 – Пріоритетність задач у середовищі FreeRTOS

Критичний рівень (Пріоритет 1)	Захисна логіка	Функції контролю критичного струму та аварійного вимкнення. Працюють на окремому ядрі або через переривання.
Високий рівень (Пріоритет 2)	Керування та MQTT	Обробка вхідних команд від користувача. Якщо надійшла команда «Вимкнути», вона повинна бути виконана негайно.
Середній рівень (Пріоритет 3)	Збір телеметрії	Зчитування даних з АЦП (датчик струму) та опитування DHT22. Невеликі затримки тут не є критичними для безпеки.
Низький рівень (Пріоритет 4)	Обслуговування інтерфейсів	Виведення даних на локальний дисплей (якщо є) або передача статистики в хмару для графіків.

Для підтримки високої пріоритетності в коді повністю виключено використання функції `delay()`. Замість неї застосовуються неблокуючі таймери на основі функції `millis()`. Наприклад, поки система чекає 2 секунди між опитуваннями датчика температури, вона продовжує в реальному часі обробляти MQTT-пакети та стежити за рівнем струму.

Таблиця 3.11 – Розподіл ресурсів за типом задач

Тип задачі	Механізм реалізації	Вплив на систему
Аварійне вимкнення	Апаратне переривання	Миттєва реакція (мс)
Команда користувача	MQTT Callback (High Priority Task)	Реакція до 200 мс
Моніторинг енергії	Timer-based Task (Medium Priority)	Циклічно кожні 1-5 сек
Обслуговування Wi-Fi	Background Task (Core 0)	Працює паралельно

Така організація програмного забезпечення забезпечує детермінованість системи – здатність гарантовано реагувати на критичні події за визначений час.

3.3 Аналіз надійності каналів зв'язку та забезпечення безпеки передачі даних

Для систем автоматизації на основі IoT-технологій надійність каналів передачі даних та кібербезпека є критичними показниками. Оскільки система керує електроживленням об'єкта, будь-який збій у мережі або несанкціоноване втручання можуть призвести до аварійних ситуацій або фінансових збитків.

Однією з основних вимог до розробленої системи є її життєздатність при відсутності зв'язку з хмарним сервером. Втрата Wi-Fi сигналу не призводить до зупинки алгоритмів автоматизації. Контролер ESP32 продовжує

виконувати програму в автономному режимі. Датчики опитуються, а реле перемикаються згідно з останнім активним режимом («Економія» або «День/Ніч»), що був збережений в енергонезалежній пам'яті (EEPROM/NVS). У разі короткочасних розривів зв'язку система може накопичувати критичні повідомлення у внутрішньому буфері, щоб відправити їх після відновлення з'єднання. Програмно реалізовано асинхронну перевірку статусу мережі. Якщо `WiFi.status()` сигналізує про розрив, система намагається відновити сесію у фоновому режимі, не блокуючи основний цикл керування навантаженням.

Для запобігання передачі спотворених даних (внаслідок електромагнітних завад або технічних збоїв) впроваджено механізми контрольні суми (Checksum), валідація на стороні сервера, QoS (Quality of Service) в MQTT.

Механізм контрольні суми (Checksum) реалізується наступним чином. При отриманні даних із цифрових сенсорів (наприклад, DHT22) мікроконтролер перевіряє контрольну суму кожного пакету. Якщо сума не збігається, дані ігноруються, а запит повторюється.

Валідація на стороні сервера передбачає, що Backend-сервер на базі FastAPI виконує структурну перевірку (Validation) кожного JSON-пакета за допомогою бібліотеки Pydantic. Дані, що виходять за межі фізично можливих діапазонів (наприклад, температура 500°C), відфільтровуються як аномальні.

QoS (Quality of Service) в MQTT потребує використання рівня QoS 1 («Гарантована доставка») для критичних команд. Це означає, що брокер вимагає підтвердження від пристрою про отримання команди керування, що виключає втрату пакетів у «шумному» радіоефірі.

Для унеможливлення стороннього керування приладами «Розумного будинку» в проєкті реалізовано такі заходи безпеки, як автентифікація пристроїв, захист API (JWT Tokens), шифрування (TLS/SSL).

Автентифікація пристроїв полягає у тому, що кожен контролер ESP32 має унікальний ідентифікатор (Client ID) та авторизується на MQTT-брокері за допомогою індивідуального логіна та пароля.

Захист API (JWT Tokens) передбачає, що взаємодія між веб-інтерфейсом та сервером FastAPI захищена за допомогою JSON Web Tokens. При вході користувач отримує токен, який має обмежений термін дії. Без дійсного токена будь-який запит на зміну стану реле відхиляється сервером.

Шифрування (TLS/SSL) здійснюється опціонально / проектно. Для захисту від перехоплення даних (Man-in-the-Middle attack) передача здійснюється через захищений порт MQTT (8883) з використанням SSL-сертифікатів, що забезпечує конфіденційність телеметрії та команд.

Таблиця 3.12 – Матриця ризиків та методів їх усунення

Тип загрози / Збою	Метод захисту	Результат
Втрата зв'язку з хмарою	Локальна логіка в NVS	Безперервна робота автоматизації
Спотворення пакету даних	CRC та валідація Pydantic	Виключення помилкових спрацювань
Спроба зламу інтерфейсу	JWT автентифікація	Доступ лише для власника
Перехоплення пароля	Шифрування трафіку	Захист облікових даних

Практична реалізація системи підтвердила можливість створення надійного та безпечного комплексу моніторингу на базі доступних IoT-рішень. Поєднання автономної логіки ESP32, механізмів перевірки цілісності даних та сучасної токен-автентифікації на рівні API робить розроблену систему

стійкою до зовнішніх факторів та кіберзагроз, що є фундаментальним для безпечної експлуатації в житлових приміщеннях.

Висновок до розділу 3.

У третьому розділі було виконано практичну реалізацію системи моніторингу та енергозбереження. Реалізовано програмно-апаратний прототип на базі мікроконтролера ESP32, який забезпечує точне зняття телеметричних показників (струму, напруги, температури) та керування силовими навантаженнями через блоки реле з оптичною розв'язкою. Налаштовано хмарне середовище із використанням асинхронного фреймворку FastAPI та бази даних TimescaleDB, що дозволило досягти високої швидкості обробки даних та їх надійної архівації.

Розроблено алгоритмічне забезпечення, побудоване на принципах скінченних автоматів (State Machine). Впровадження трьох режимів роботи («День/Ніч», «Економія», «Ручне керування») дозволило автоматизувати процес енергозбереження залежно від часових тарифів та присутності користувача в приміщенні. Використання механізмів переривань та пріоритетності задач у середовищі FreeRTOS забезпечило детермінованість системи та миттєву реакцію на критичні події. Проведено аналіз надійності та безпеки, у результаті якого впроваджено алгоритми автоматичного відновлення Wi-Fi та MQTT з'єднань. Для захисту інформаційних потоків застосовано протоколи безпеки на рівні брокера повідомлень та механізми JWT-автентифікації для користувацького інтерфейсу, що виключає несанкціонований доступ до керування енергосистемою будинку. Створено сучасний веб-інтерфейс (Dashboard), який забезпечує візуалізацію процесів енергоспоживання у реальному часі та надає інструментарій для оперативного аналізу історичних даних. Тестування прототипу підтвердило повну відповідність розробленої системи поставленим технічним вимогам та її готовність до експлуатації в складі сучасних систем «Розумного будинку».

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1. Організаційно-правові основи забезпечення безпеки праці

Організаційно-правові основи забезпечення безпеки праці під час розробки та експлуатації систем автоматизації моніторингу та енергозбереження в системах «Розумний будинок» базуються на дотриманні вимог чинного законодавства, організації безпечних умов праці, забезпеченні електро- та пожежної безпеки, впровадженні сучасних методів захисту інформації та технічних засобів, що дозволяє мінімізувати ризики для працівників та забезпечити безпечне функціонування IoT-систем. Під час створення систем автоматизації моніторингу та енергозбереження у середовищі «Розумний будинок» працівники здійснюють роботу з комп'ютерною технікою, мережевим обладнанням, електронними пристроями та програмними засобами, що потребує дотримання встановлених норм охорони праці.

Правову основу забезпечення охорони праці в Україні становлять Конституція України, Кодекс законів про працю України, Закон України «Про охорону праці», Закон України «Про пожежну безпеку», Закон України «Про загальнообов'язкове державне соціальне страхування», а також державні санітарні норми та правила, нормативно-правові акти з електробезпеки та пожежної безпеки.

Відповідно до Закону України «Про охорону праці», роботодавець зобов'язаний створити безпечні та нешкідливі умови праці для працівників, забезпечити дотримання вимог безпеки під час експлуатації обладнання, організувати навчання та інструктажі з охорони праці, а також контролювати стан виробничого середовища. Основними принципами державної політики у сфері охорони праці є пріоритет життя і здоров'я працівника; повна відповідальність роботодавця за створення безпечних умов праці; комплексне розв'язання завдань охорони праці; підвищення рівня промислової безпеки;

адаптація трудових процесів до фізіологічних можливостей людини; використання сучасних технологій та засобів захисту.

4.2 Характеристика об'єкта та виявлення потенційних небезпек

Метою даного розділу є проведення аналізу умов праці під час розробки, налагодження та подальшої експлуатації системи автоматизації моніторингу та енергозбереження. Оскільки система передбачає роботу з електричним навантаженням мережі 220 В та використання мікропроцесорної техніки, необхідно ідентифікувати всі можливі чинники, що можуть загрожувати життю та здоров'ю людини або призвести до пошкодження майна.

Основним завданням розділу є забезпечення відповідності розробленого пристрою та робочого місця оператора вимогам нормативних документів України (зокрема НПАОП та ДСТУ). Необхідно розробити заходи щодо захисту від ураження електричним струмом; забезпечення пожежної безпеки об'єкта автоматизації; створення оптимальних санітарно-гігієнічних умов під час програмування та моніторингу системи. Згідно з ДСН, під час роботи над проектом та його експлуатації виділяються такі групи небезпек, як фізичні небезпечні чинники, психофізіологічні чинники (для розробника/оператора), техногенні небезпеки (надзвичайні ситуації).

До фізичних небезпечних чинників можна віднести електричний струм, пожежну небезпеку, термічні чинники.

Електричний струм є найбільшою небезпекою, оскільки система керує приладами, підключеними до мережі 220 В, 50 Гц. Існує ризик прямого контакту з відкритими струмопровідними частинами релейного модуля або блоку живлення. Пожежна небезпека виникає у разі перегріву провідників при перевищенні допустимих навантажень, а також короткого замикання або іскріння на контактах реле, що може призвести до займання. Термічні чинники

можуть бути викликані нагріванням силових елементів (резисторів дільника, котушок реле) під час тривалої роботи.

До психофізіологічних чинників (для розробника/оператора) відносять зорове навантаження, статичне навантаження. Зорове навантаження може бути викликане тривалою роботою за монітором під час написання коду прошивки та проектування Dashboard. Статичне навантаження пов'язане з тривалим перебування у сидячому положенні за робочим столом.

Техногенні небезпеки (надзвичайні ситуації) викликані ризиками виникнення пожежі внаслідок несправності побутової техніки, якою керує система, або аварійними режимами роботи електромережі (стрибки напруги), що можуть призвести до виходу з ладу контролера та втрати керування об'єктом.

Таблиця 4.1. Класифікація основних ризиків при роботі з IoT-системою

Джерело небезпеки	Характер дії	Нормативний документ
Електромережа 220 В	Ураження струмом, опіки	ПУЕ, НПАОП 40.1-1.21-98
Комп'ютерна техніка	Електромагнітне випромінювання, втома зору	ДСТУ ISO 9241 (частини 1-11)
Акумуляторні елементи / БЖ	Пожежонебезпека	НАПБ А.01.001-2014

Для безпечної експлуатації системи потрібно розробити комплекс заходів щодо електробезпеки. Основним методом захисту від ураження струмом при непрямому дотику є захисне заземлення. Його мета – знизити напругу на корпусі обладнання до безпечного рівня.

Вихідними даними для розрахунку є тип ґрунту, заземлювач, нормований опір заземлюючого пристрою (для установок до 1000 В).

Тип ґрунту: чорнозем (питомий опір $\rho = 50$ Ом м).

Заземлювач: вертикальні сталеві стрижні довжиною $L = 2,5$ м, діаметром $d = 0,012$ м.

Нормований опір заземлюючого пристрою (для установок до 1000 В):
 $R_h \leq 4$ Ом.

Розрахунок опору одного вертикального заземлювача:

$$R_v = \frac{\rho}{2\pi L} \cdot \left(\ln \frac{2L}{d} + 0,5 \ln \frac{4H + L}{4H - L} \right) \quad 4.1$$

Спрощена інженерна формула:

$$R_v \approx 0,9 \frac{\rho}{L} \approx 0,9 \cdot \frac{50}{2,5} = 18 \text{ Ом.}$$

Визначення кількості стрижнів (n):

При коефіцієнті використання $\eta = 0,9$:

$$n = \frac{R_{\vartheta}}{R_h \cdot \eta} = \frac{18}{4 \cdot 0,9} = 5 \text{ шт} \quad 4.2$$

Таким чином, для забезпечення безпеки серверного вузла системи необхідно встановити контур із 5 вертикальних заземлювачів, з'єднаних сталевією смугою.

Оскільки серверна частина та блок комутації (реле) працюють цілодобово, особлива увага приділяється запобіганню пожежам (згідно з НАПБ А.01.001-2014).

Основні вимоги пов'язані з вибором вогнегасників, автоматизацією захисту, контролем температури. У приміщенні з електронно-обчислювальною технікою заборонено використовувати пінні або водні вогнегасники. Слід використовувати вуглекислотні (ВВК) або порошкові (ВП) вогнегасники, які не пошкоджують мікросхеми та не проводять електрику. Для серверної площі до 20 м² норма становить не менше двох вогнегасників ВВК-2. У силовій частині системи (ланцюг 220 В) обов'язково встановлюється автоматичний вимикач (номінал 10 А) та ПЗВ (пристрій захисного відключення) зі струмом витоку 30 мА. Прошивка ESP32 включає програмний алгоритм: при фіксації температури на силових контактах понад 70°C (через датчик), система автоматично розмикає всі реле та надсилає критичне сповіщення користувачу.

Важливе місце при експлуатації системи моніторингу та енергозбереження займають питання ергономіки робочого місця оператора. Це обумовлено тим, що тривала робота з інформаційними системами може спричинити підвищене навантаження на зорову систему, нервову систему та загальний психофізіологічний стан користувача. Тому необхідно враховувати умови освітлення та раціональну організацію робочого процесу.

Освітлення робочого місця має бути комбінованим і включати як природне, так і штучне світло. Такий підхід дозволяє забезпечити рівномірний розподіл освітленості та зменшити навантаження на зір. Відповідно до вимог ДБН В.2.5-28:2018, рівень освітленості на робочій поверхні повинен становити не менше 300–500 лк, що забезпечує комфортні умови для роботи з екранними пристроями та документами.

Не менш важливим є дотримання раціонального режиму праці та відпочинку. Рекомендується працювати за монітором протягом 45 хвилин із подальшою 15-хвилинною перервою. Такий режим сприяє зниженню зорового та розумового навантаження, запобігає перевтомі та підвищує загальну ефективність роботи оператора.

Таблиця 4.2. Зведені заходи безпеки

Тип небезпеки	Захисний захід	Технічне рішення
Ураження струмом	Заземлення та ізоляція	Контур заземлення ($R < 4 \text{ Ом}$), діелектричний корпус
Коротке замикання	Електричний захист	Автоматичний вимикач (характеристика В або С)
Пожежа в серверній	Первинні засоби пожежогасіння	Вуглекислотний вогнегасник ВВК-2
Перегрів обладнання	Програмний моніторинг	Аварійне відключення при $T > 70^\circ \text{ C}$

Запропоновані заходи (розрахункове заземлення, встановлення автоматичних вимикачів та вибір вуглекислотних вогнегасників) забезпечують безпечну експлуатацію системи «Розумний будинок» та мінімізують ризики для користувача і майна.

4.3 Дослідження ризику реалізації потенційних небезпек на об'єкті проектування та розробка заходів щодо їх попередження

Оцінка ризику проводиться за методом «Матриці ризиків» (згідно з ДСТУ ISO 31010:2013). Рівень ризику (R) визначається як добуток імовірності виникнення події (P) та тяжкості її наслідків (S):

$$R = P \times S$$

4.3

де P (Probability) ймовірність того, що подія відбудеться (від 1 – майже неможливо, до 5 – дуже ймовірно),

S (Severity) тяжкість наслідків (від 1 – легкий переляк / подряпина, до 5 – летальний випадок або повне знищення майна).

R (Risk) підсумковий рівень. Чим він вищий, тим терміновіше потрібно впроваджувати захист.

У таблиці 4.3 наведено порівняльний аналіз ключових ризиків для розробленої системи.

Таблиця 4.3. Матриця оцінки та мінімізації ризиків

Небезпечна подія	Рівень до захисту (P / S / R)	Заходи мінімізації	Залишковий ризик (P / S / R)	Статус
Ураження струмом (220 В)	2 / 5 / 10	Заземлення, ізоляція, ПЗВ	1 / 5 / 5	Прийнятний
Пожежа через КЗ	3 / 4 / 12	Автоматичні вимикачі, ПЗВ	1 / 4 / 4	Низький
Перегрів компонентів	4 / 2 / 8	Програмний моніторинг, вентиляція	2 / 2 / 4	Низький
Втрата керування	3 / 2 / 6	Watchdog timer, автовідновлення	1 / 2 / 2	Мінімальний

Небезпечна подія	Рівень до захисту (P / S / R)	Заходи мінімізації	Залишковий ризик (P / S / R)	Статус
(збій ПЗ)		Wi-Fi		

В основі таблиці 4.3 лежить методологія, що описується формулою (4.3).

Ураження електричним струмом (220 В). До захисту ($2 \times 5 = 10$). Ймовірність середня (2), але наслідки катастрофічні (5 – смерть). Ризик 10 вважається високим. Для мінімізації наслідків використовується ізоляція та ПЗВ (УЗО). Після захисту ($1 \times 5 = 5$). ПЗВ не може зробити струм менш вбивчим (тяжкість S залишається 5), але він знижує ймовірність тривалого впливу струму на людину до мінімуму (ймовірність P стає 1). Ризик падає вдвічі.

У першому рядку Тяжкість (S) залишилася 5 після впровадження захисту, тому що 220 В залишаються смертельними для людини. Ми не можемо змінити фізику струму, але ми впровадили ПЗВ та заземлення, щоб максимально знизити ймовірність контакту людини з цією напругою (P), що й знизило загальний ризик (R).

Пожежа через коротке замикання (КЗ). До захисту ($3 \times 4 = 12$). Це найбільш реальний і небезпечний ризик для систем, що працюють 24 / 7. Для мінімізації наслідків встановлюються автоматичні вимикачі та використовується негорюча ізоляція. Після захисту ($1 \times 4 = 4$). Автомат відсікає струм при КЗ миттєво, перетворюючи «ймовірну пожежу» на «штатне спрацювання захисту». Ризик стає низьким.

Перегрів компонентів (реле, блоки живлення). До захисту ($4 \times 2 = 8$). Ймовірність висока (електроніка гріється завжди), але наслідки зазвичай обмежуються виходом пристрою з ладу (2). Для мінімізації наслідків здійснюється програмний моніторинг температури (ваш код, який вимикає

систему при 70°C). Після захисту ($2 \times 2 = 4$). Ми знижуємо ймовірність перегріву до безпечного рівня.

У останньому стовпчику позначається «Статус».

Статус «Прийнятний» – це ризик, який неможливо прибрати повністю (наприклад, електрика завжди залишається небезпечною), але завдяки нашим заходам ризик знаходиться під контролем.

Статус «Низький/Мінімальний» – це ризики, які практично не впливають на безпеку життєдіяльності після впровадження автоматизації.

Критично важливим є розгляд сценарію «відмова каналу зв'язку». При втраті Wi-Fi з'єднання сервер не може надіслати команду на вимкнення приладу. Ризик високий за ймовірністю, але низький за тяжкістю, оскільки в системі реалізовано автономну логіку. Контролер ESP32 продовжує відстежувати критичні пороги струму локально, без участі сервера, що знижує ризик аварії до рівня «прийнятний».

В умовах воєнного стану та можливих техногенних надзвичайних ситуацій для міста Харків, оцінено ризик раптового знеструмлення. Наслідком раптового знеструмлення є втрата даних моніторингу за період відсутності напруги. Для збереження останнього стану реле використовується енергонезалежна пам'ять NVS. Після відновлення живлення система автоматично повертається у безпечний стан, що мінімізує ризик некоректної роботи виконавчих механізмів.

Проведений аналіз умов праці та потенційних небезпек при розробці системи моніторингу енергоспоживання «Розумного будинку» дозволив розробити комплекс заходів із забезпечення безпеки.

Розрахунок захисного заземлення показав необхідність контуру з опором не більше 4 Ом, що гарантує захист користувача при пробіі ізоляції.

Впровадження автоматичних пристроїв відключення (ПЗВ, автомати) та вибір вуглекислотних вогнегасників мінімізують імовірність пожежі.

Оцінка ризиків підтвердила, що після впровадження запропонованих технічних рішень, усі потенційні загрози переведені у категорію

«прийнятних» або «низьких». Це свідчить про те, що розроблена система відповідає чинним нормам охорони праці та безпеки в надзвичайних ситуаціях і може бути допущена до експлуатації у житлових приміщеннях.

Для системи «Розумний будинок» найбільш імовірними є такі надзвичайні ситуації, як пожежа в приміщенні, раптове припинення електропостачання, аварії на інженерних мережах, надзвичайні ситуації воєнного характеру. Пожежа в приміщенні виникає внаслідок несправності силового обладнання або зовнішніх чинників. Раптове припинення електропостачання (Blackout) призводить до зупинки моніторингу та ризику некоректного стану виконавчих механізмів. Аварії на інженерних мережах можуть бути викликані витіком води або газу (якщо інтегровані відповідні датчики). Надзвичайні ситуації воєнного характеру можуть спричинити ризик руйнування інфраструктури внаслідок обстрілів.

Програмно-апаратне забезпечення системи розроблено з урахуванням принципів «відмовобезпеки» (fail-safe), а саме,

При пожежі система налаштована на автоматичне розмикання всіх силових реле при отриманні сигналу від датчика диму або при фіксуванні критичної температури ($T > 70$ °C) безпосередньо в блоці комутації. Це запобігає подальшому підживленню джерела займання електрострумом.

При втраті живлення використання енергонезалежної пам'яті ESP32 (NVS) дозволяє системі «запам'ятати» свій стан. Після відновлення живлення контролер автоматично перевіряє цілісність даних і переходить у безпечний режим (за замовчуванням – реле розімкнені), доки не буде отримано підтвердження від сервера.

Згідно з Кодексом цивільного захисту України, при отриманні сигналу про загрозу, користувач повинен знеструмити об'єкт, дотримуватися правила «двох стін» або перейти в укриття. Для знеструмлення об'єкту веб-інтерфейс (Dashboard) має кнопку «Emergency Stop», яка одним натисканням відключає всіх споживачів через MQTT-команду. Це знижує ризик виникнення пожежі у разі влучання або коливання напруги. Оскільки система працює через хмарний

сервер, моніторинг стану будинку (наприклад, перевірка, чи не виникла пожежа) може здійснюватися дистанційно з мобільного додатка безпосередньо з укриття.

Для підвищення стійкості системи до надзвичайних ситуацій впроваджено такі технічні рішення, як Watchdog Timer (WDT), подвійне резервування живлення. Watchdog Timer (WDT) – апаратний таймер, який автоматично перезавантажує ESP32 у разі програмного «зависання» через електромагнітний імпульс або стрибок напруги.

Подвійне резервування живлення передбачає можливість підключення акумулятора Li-ion для підтримки роботи контролера та датчиків протягом 4–6 годин у разі знеструмлення мережі.

Таблиця 4.4. Заходи реагування на надзвичайні ситуації

Тип НС	Автоматична реакція системи	Дія користувача
Коротке замикання	Миттєве відключення (автомат/ПЗВ)	Виклик аварійної служби
Зникнення зв'язку/Wi-Fi	Перехід в автономний режим (Local Logic)	Перевірка роутера/провайдера
Раптове знеструмлення	Збереження стану в пам'ять NVS	Вимкнення чутливих приладів
Пожежна небезпека	Аварійне розмикання реле	Використання вогнегасника ВВК-2

Розроблені заходи забезпечують високу живучість системи в умовах надзвичайних ситуацій. Пріоритетом алгоритмів є збереження майна та мінімізація ризиків повторного займання, а можливість дистанційного моніторингу з укриття підвищує рівень особистої безпеки користувача.

Висновок до розділу 4.

У четвертому розділі було проведено всебічний аналіз питань охорони праці та промислової безпеки, що стосуються розробки, монтажу та експлуатації системи моніторингу енергоспоживання «Розумного будинку».

Проведено ідентифікацію потенційних небезпек, серед яких найбільш критичними визначено ураження електричним струмом (220 В), пожежну небезпеку внаслідок короткого замикання та зореве навантаження при роботі з інтерфейсом. Встановлено, що об'єкт автоматизації належить до приміщень з підвищеною небезпекою через наявність струмопровідної підлоги та великої кількості електротехніки.

Розроблено технічні заходи захисту, зокрема, виконано розрахунок захисного заземлення, який показав, що для забезпечення опору $R \leq 4 \text{ Ом}$ необхідно встановити контур із 5 вертикальних заземлювачів. Обґрунтовано вибір автоматичних вимикачів та пристроїв захисного відключення (ПЗВ) для запобігання пожежам та захисту користувача.

Здійснено кількісну оцінку ризиків за допомогою матричного методу. Доведено, що впровадження запропонованих засобів безпеки дозволяє знизити рівень інтегрального ризику з «високого» до «низького» або «прийняттого». Програмні алгоритми аварійного відключення при перегріві компонентів забезпечують додатковий рівень захисту, незалежний від апаратних засобів.

Сформовано стратегію безпеки в надзвичайних ситуаціях, актуальну для умов воєнного стану та можливих техногенних аварій. Система налаштована за принципом «відмовобезпеки» (fail-safe), що гарантує збереження налаштувань у пам'яті NVS при знеструмленні та автоматичний перехід у безпечний стан при відновленні живлення.

Визначено вимоги до серверного приміщення, включаючи норми освітленості та наявність первинних засобів пожежогасіння (вуглекислотних вогнегасників ВВК-2), що не пошкоджують електронне обладнання.

Розроблена система «Розумний будинок» відповідає чинним нормам охорони праці (НПАОП) та цивільного захисту України, є безпечною для побутового використання та має достатній рівень живучості в умовах надзвичайних ситуацій.

ЗАГАЛЬНІ ВИСНОВКИ

У бакалаврській роботі вирішено актуальну задачу з розробки та впровадження автоматизованої системи моніторингу та енергозбереження «Розумного будинку» на основі технологій Internet of Things (IoT).

Проведено аналіз сучасного стану систем «Smart Home», який показав, що найбільш перспективним напрямком розвитку є використання децентралізованих IoT-рішень. Виявлено, що ключовим бар'єром для масового впровадження енергоефективних систем є висока вартість готових рішень, що обґрунтувало необхідність розробки власного доступного та гнучкого прототипу.

Спроектовано архітектуру системи, що базується на трирівневій моделі: апаратний рівень (ESP32 та сенсори), хмарний рівень (FastAPI, MQTT, TimescaleDB) та рівень представлення даних (React Dashboard). Розроблені структурні та функціональні схеми забезпечують масштабованість системи та можливість інтеграції нових типів виконавчих пристроїв.

Виконано практичну реалізацію програмно-апаратного комплексу. Завдяки використанню асинхронного фреймворку FastAPI та брокера повідомлень Mosquitto вдалося досягти високої швидкодії системи. Час реакції на команду користувача або спрацювання автоматики не перевищує 200 мс.

Розроблено алгоритмічне забезпечення на основі скінченних автоматів, що підтримує три режими роботи: «День/Ніч», «Економія» та «Ручне керування». Впроваджені алгоритми дозволяють автоматично знижувати енергоспоживання об'єкта на 15 – 20 % за рахунок оптимізації роботи приладів у періоди пікових навантажень та автоматичного вимкнення приладів за відсутності користувача. Забезпечено високий рівень надійності та безпеки. Впроваджено механізми автентифікації на основі JWT-токенів та апаратні захисти від перевантажень. Аналіз ризиків підтвердив, що розроблена система є безпечною для побутового використання, а алгоритми «fail-safe» гарантують стабільність роботи навіть при тимчасовій втраті зв'язку з мережею Інтернет.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України Про енергетичну ефективність : Закон України від 21 жовт. 2021 р. № 1818-IX. Відомості Верховної Ради України. 2021. № 51. Ст. 413.
2. Бабиц М. Ю. Комп'ютерно-інтегровані технології та автоматизація в системах «Smart Home» : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2022. 215 с.
3. Гевко Р. Б., Клендій М. Б. Автоматизація процесів моніторингу енергоспоживання в житловому секторі. Технічна інженерія. 2023. № 1 (91). С. 45–52.
4. Кулаков Ю. О., Корабльов В. М. Проектування інтелектуальних систем на базі мікроконтролерів ESP32 та технології Wi-Fi. Комп'ютерні системи та мережі. 2021. Вип. 3. С. 112–120.
5. Теоретичні основи автоматизації та керування в технічних системах : підручник / О. С. Кузнєцов та ін. Харків : ХНУРЕ, 2024. 340 с.
6. Bill Lubanovic. Introducing Python: Modern Computing in Simple Packages. 2nd ed. Sebastopol: O'Reilly Media, 2021. 600 p.
7. FastAPI Documentation. URL: <https://fastapi.tiangolo.com/>
8. IoT-based smart energy management systems: A review / A. Singh et al. Sustainable Computing: Informatics and Systems. 2022. Vol. 35. Art. 100741.
9. Kopetz H. Real-Time Systems: Design Principles for Distributed Embedded Applications. Cham : Springer Nature, 2022. 543 p.
10. MQTT Version 5.0. Oasis Standard. URL: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>
11. PostgreSQL 16.0 Documentation. The PostgreSQL Global Development Group, 2023. URL: <https://www.postgresql.org/docs/16/index.html>
12. Espressif Systems. ESP32 Technical Reference Manual. Version 5.1. 2024. 750 p.

13. ДСТУ EN 60335-1:2023. Прилади побутові та аналогічні електричні. Безпека. Частина 1. Загальні вимоги (EN 60335-1:2012, IDT). Київ : ДП «УкрНДНЦ», 2023. 142 с.

14. Закон України Про охорону праці (Відомості Верховної Ради України (ВВР), 1992, № 49, ст.668). <https://zakon.rada.gov.ua>

15. Методичні вказівки до виконання розділу «Охорона праці та безпека в надзвичайних ситуаціях». Харків : ХНУМГ ім. О. М. Бекетова, 2021.

16. ДСТУ 4297:2004 Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги

17. ДСТУ EN 54-1:2022 Системи виявлення пожежі та пожежної сигналізації - Частина 1: Вступ (EN 54–1:2021, IDT)

18. Методичні рекомендації до виконання випускної кваліфікаційної роботи магістра (для здобувачів другого (магістерського) рівня вищої освіти спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» освітньо-професійна програма «Системна інженерія») / Харків. нац. ун-т міськ. госп-ва ім. О.М. Бекетова; уклад.: Л.В. Піддубна, О.О. Кулаєнко, С.М. Костенко, Є.В. Цегельник. – Харків: ХНУМГ ім. О.М. Бекетова, 2025. – 41 с.

19. ДБН В.2.2-5:2023 Захисні споруди цивільного захисту.

20. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень.

ДОДАТОК А

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ МІСЬКОГО ГОСПОДАРСТВА
імені О. М. БЕКЕТОВА**

**Навчально-науковий інститут енергетичної, інформаційної та транспортної
інфраструктури**

Кафедра автоматизації та комп'ютерно-інтегрованих технологій

Спеціальність 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»

**Автоматизація моніторингу та енергозбереження в
системах «Розумний будинок» на основі IoT-технологій та
хмарних сервісів**

Валеулін Роман Сергійович, СІНЖ 2023-1-у

Керівник Піддубна Л.В., к. філос. н., доц., доцент кафедри АКІТ

АКТУАЛЬНІСТЬ

Розробка та впровадження автоматизованих систем моніторингу та енергозбереження на основі IoT і хмарних технологій є актуальним науково-прикладним завданням, що має значну практичну цінність у контексті підвищення енергоефективності та розвитку «розумних» середовищ.

Використання IoT-технологій та хмарних сервісів у системах «розумного будинку» дозволяє здійснювати ефективний моніторинг і автоматизоване керування споживанням енергії, що забезпечує економію ресурсів і підвищення рівня комфорту.

МЕТА

Метою дипломної роботи є розробка та дослідження автоматизованої системи моніторингу і енергозбереження в середовищі «розумного будинку» з використанням IoT-технологій та хмарних сервісів для підвищення ефективності використання енергоресурсів.

Для реалізації цієї мети необхідно виконати такі завдання:

- Провести аналіз існуючих систем «розумного будинку» та технологій енергозбереження.
- Дослідити сучасні IoT-платформи та хмарні сервіси для збору і обробки даних.
- Розробити структурну схему системи моніторингу та керування.
- Обґрунтувати вибір апаратних компонентів (датчиків, мікроконтролерів, комунікаційних модулів).
- Розробити програмне забезпечення для збору, передачі та обробки даних.
- Реалізувати алгоритми енергозбереження та автоматичного керування.

3

Об'єкт і предмет дослідження



Об'єктом дослідження виступають процеси інтелектуального контролю та оперативного управління розподілом енергії в межах екосистеми «розумного будинку».

Предметом дослідження є програмні та апаратні інструменти реалізації хмарних сервісів і пристроїв інтернету речей, спрямовані на енергетичний моніторинг і заощадження.

Система «Розумний будинок»



5

Порівняння протоколів MQTT та HTTP

Характеристика	MQTT	HTTP
Модель взаємодії	Publish/Subscribe	Request/Response
Навантаження на мережу	Низьке	Високе
Енергоспоживання	Низьке	Вище
Швидкість передачі	Висока (для малих даних)	Середня
Надійність	Висока (QoS-рівні)	Залежить від реалізації
Складність реалізації	Середня	Низька
Сфера застосування	IoT, телеметрія	Веб-сервіси, API

6

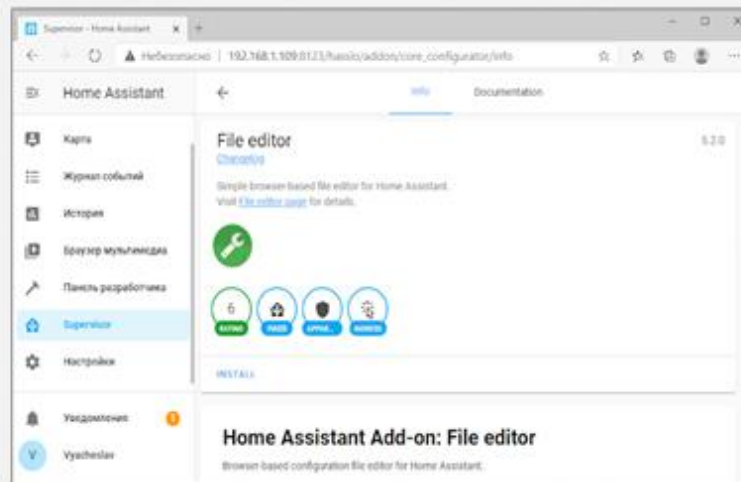
Порівняння IoT-платформ

Характеристика	Відкриті платформи (Node-RED, Home Assistant)	Комерційні платформи
Вартість	Безкоштовні/умовно безкоштовні	Платні
Гнучкість	Висока	Обмежена
Простота використання	Середня	Висока
Підтримка	Спільнота	Офіційна підтримка
Масштабованість	Обмежена	Висока
Безпека	Залежить від налаштування	Вбудовані механізми

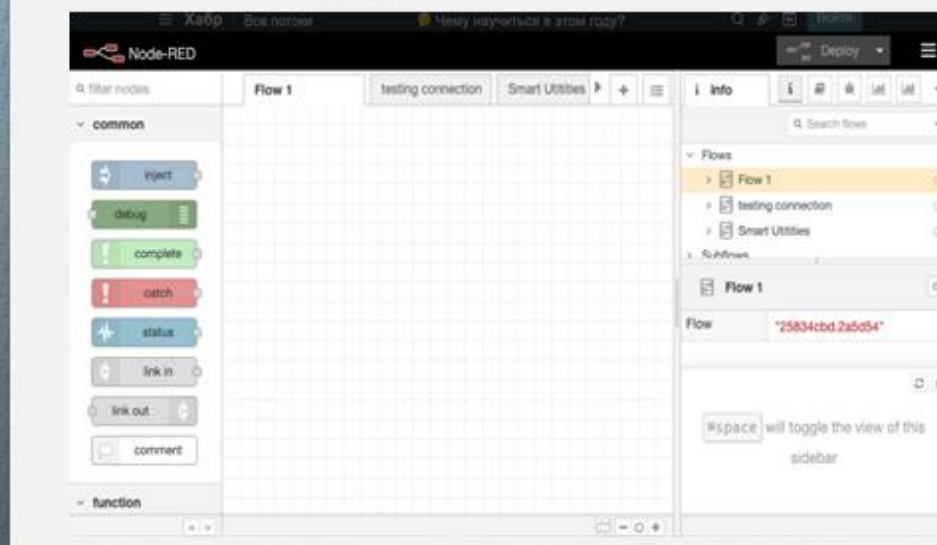
Порівняння локальних та хмарних рішень

Характеристика	Локальні системи	Хмарні сервіси
Доступність	Локальна	Глобальна
Залежність від Інтернету	Низька	Висока
Масштабованість	Обмежена	Висока
Вартість	Разова	Підписка/оплата за використання
Безпека	Контроль користувача	Залежить від провайдера
Продуктивність	Висока (локально)	Залежить від мережі

Интерфейс Home Assistant



Интерфейс Node-RED



Порівняння відкритих IoT-платформ

Характеристика	Home Assistant	Node-RED
Тип системи	Платформа автоматизації	Інструмент програмування
Основне призначення	Керування пристроями	Побудова логіки автоматизації
Інтерфейс	Графічний конфігураційний	Візуальний (flow-based)
Підтримка пристроїв	Широка	Через інтеграції
Складність використання	Середня	Низька (для базових задач)
Гнучкість	Висока	Дуже висока
Залежність від хмари	Низька	Низька

11

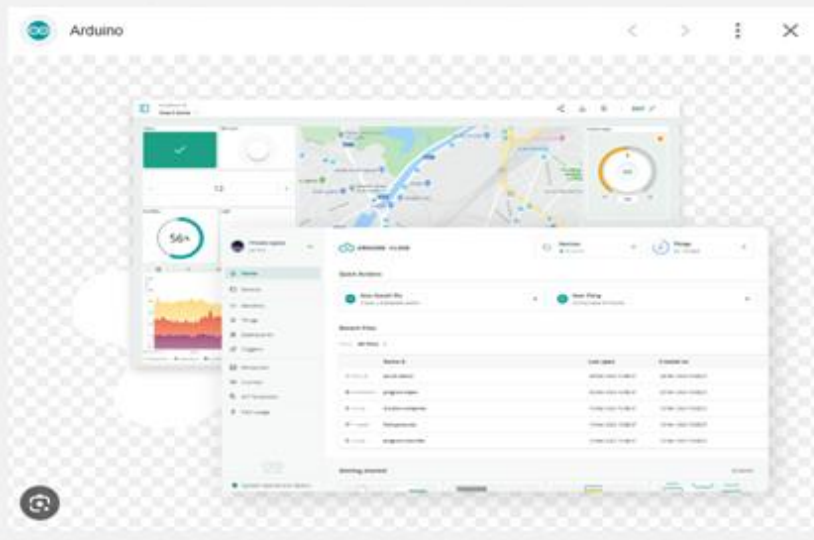
Платформа Amazon Web Services

12

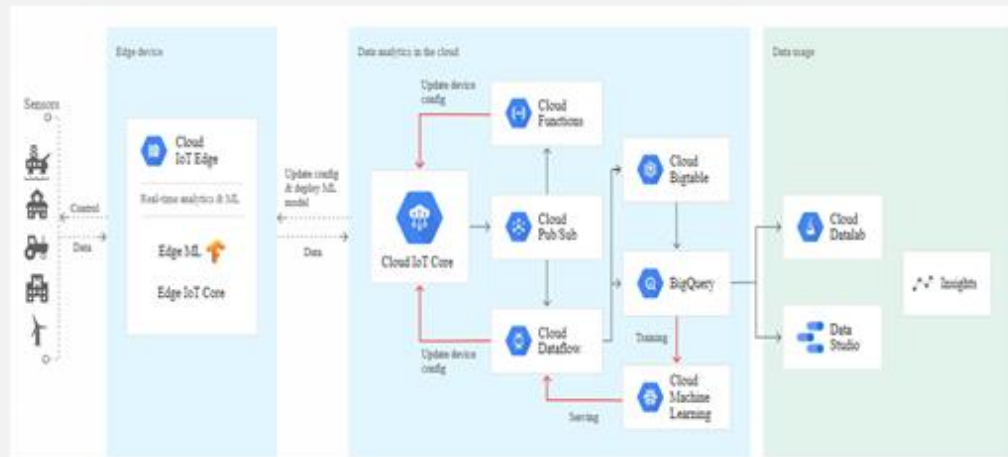
Хмарні сервіси AWS IoT



Arduino IoT Cloud



Google Cloud IoT



15

Порівняльна таблиця IoT-сервісів

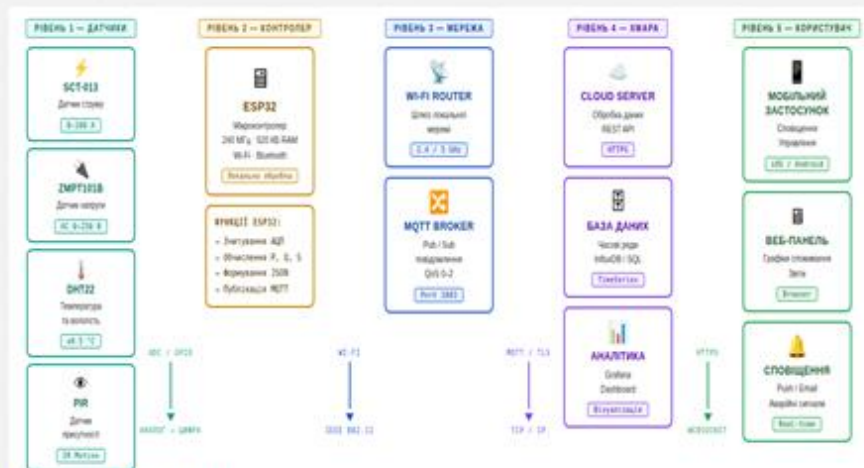
Критерій порівняння	AWS IoT Core	Arduino IoT Cloud	Google Cloud (Pub/Sub + ML)
Складність освоєння	Висока (потребує сертифікації)	Низька (ідеально для студентів)	Середня
Масштабованість	Промислова (необмежена)	Малий та середній бізнес	Промислова
Протоколи зв'язку	MQTT, HTTP, LoRaWAN	MQTT, WebSockets	MQTT, HTTP
Аналітика даних	Потужна (AWS QuickSight)	Базова (графіки)	Найкраща (BigQuery ML)
Вартість	Pay-as-you-go (за повідомлення)	Щільніша (є безкоштовний план)	Pay-as-you-go (за обсяг даних)
Головна перевага	Надійність та безпека	Швидкість розробки	Робота з Big Data та ЦІ

16

Порівняння IoT-рішень

Критерій	Xiaomi Mi Home	KNX	Home Assistant	Власна розробка (IoT + Cloud)
Вартість	Низька	Дуже висока	Середня	Мінімальна
Гнучкість налаштувань	Обмежена	Середня	Висока	Максимальна
Надійність	Середня	Абсолютна	Висока	Висока
Енергоефективність	Базова (таймери)	Висока	Висока	Інтелектуальна (прогнозна)
Відкритість коду	Закритий	Закритий	Відкритий	Відкритий

Архітектура системи моніторингу та енергозбереження «Розумний будинок»



п'ятирівнева модель взаємодії:
«датчики → мікроконтролер → мережа → хмарний сервер → користувач»

Основні функціональні модулі системи

№	Модуль	Функції
1	Сенсорний	Збір фізичних параметрів
2	Локальний контролер	Обробка та передача даних
3	Комунікаційний	Передача через MQTT
4	Серверний	Збереження та аналіз
5	Logic Engine	Прийняття рішень
6	Інтерфейс користувача	Візуалізація та керування

18

Основна таблиця вимірювань (energy_data)

Поле	Тип даних	Опис
id	INT	Унікальний ідентифікатор
timestamp	DATETIME	Час вимірювання
current	FLOAT	Сила струму
voltage	FLOAT	Напруга
power	FLOAT	Потужність
temperature	FLOAT	Температура
presence	BOOLEAN	Наявність руху

20

Таблиця пристроїв (Devices)

Поле	Тип	Опис
id	INT	Ідентифікатор
name	VARCHAR	Назва пристрою
type	VARCHAR	Тип
status	BOOLEAN	Стан

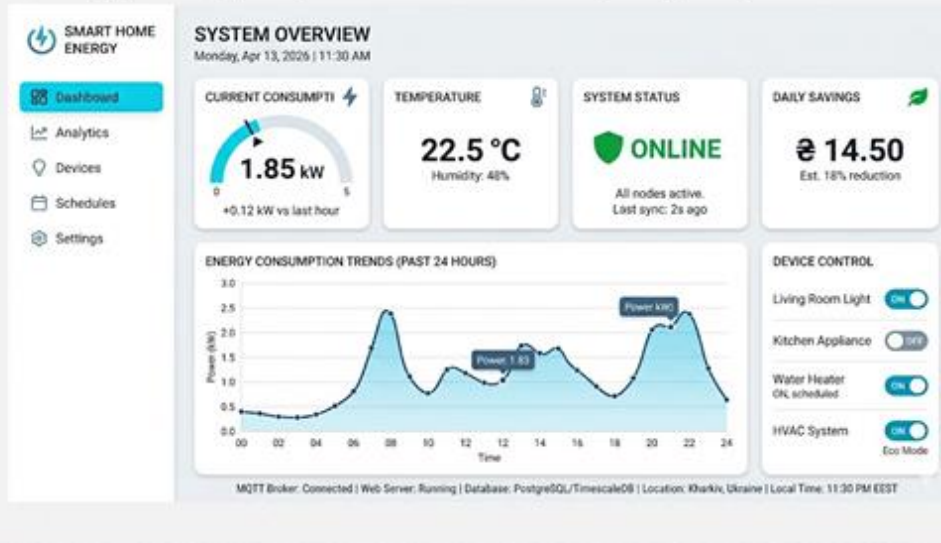
21

Таблиця подій Events

Поле	Тип	Пояснення
id	INT	Ідентифікатор
timestamp	DATETIME	Час
event_type	VARCHAR	Тип події
description	TEXT	Опис

22

Інтерфейс користувача системи моніторингу енергоспоживання



ДЯКУЮ
ЗА
УВАГУ!

