

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та охоронних систем

(повне найменування кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**ІНТЕЛЕКТУАЛЬНА СИСТЕМА КЕРУВАННЯ
ЕНЕРГОСПОЖИВАННЯМ У РОЗУМНОМУ БУДИНКУ НА
ОСНОВІ IOT**

**INTELLIGENT ENERGY MANAGEMENT SYSTEM IN A SMART
HOME BASED ON IOT**

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи KI-42
Мошківський Данило Валентинович

(підпис)

Керівник:
к.т.н., доцент
Гринюк Сергій Васильович

(підпис)

Кваліфікаційну роботу
допущено до захисту
« » червня 2026 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2026 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та безпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

« 23 » 12 доц. Т. Терлецький
2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Мошківського Данила Валентиновича

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Інтелектуальна система керування енергоспоживанням у розумному будинку на основі IoT*

Керівник роботи *к.т.н., доцент Гринюк Сергій Васильович*

затверджені наказом закладу вищої освіти від «20» грудня 2025 року № 536/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 28.05.2026 р.

3. Вихідні дані до роботи *Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області, різні інтернет-ресурси технічного спрямування*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

Вступ

Аналіз технологій та принципів побудови інтелектуальних IoT-систем керування енергоспоживанням

Проектування інтелектуальної IoT-системи керування енергоспоживанням

Програмно-апаратна реалізація та тестування системи керування енергоспоживанням

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Загальний вигляд апаратної частини системи моніторингу та керування енергоспоживанням

Структура програмного забезпечення системи моніторингу та керування енергоспоживанням

Алгоритм роботи програмного забезпечення системи керування енергоспоживанням

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз технологій та принципів побудови інтелектуальних IoT-систем керування енергоспоживанням	Гринюк С.В., доцент		
Проектування інтелектуальної IoT-системи керування енергоспоживанням	Гринюк С.В., доцент		
Програмно-апаратна реалізація та тестування системи керування енергоспоживанням	Гринюк С.В., доцент		
Нормоконтроль	Багнюк Н.В., доцент		
Гарант ОП	Лавренчук С.В., доцент		
Показник запозичень тексту	%		
Академічна доброчесність	Міскевич О.І., ст. викладач		

7. Дата видачі завдання 23.12.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Огляд літератури із досліджуваної проблеми, аналіз предметної області та наявних рішень	до 10.02.2026 р.	
2.	Аналіз технологій та принципів побудови інтелектуальних IoT-систем керування енергоспоживанням	до 02.03.2026 р.	
3.	Проектування інтелектуальної IoT-системи керування енергоспоживанням	до 02.04.2026 р.	
4.	Програмно-апаратна реалізація та тестування системи керування енергоспоживанням	до 10.04.2026 р.	
5.	Представлення остаточного варіанту кваліфікаційної роботи керівникові	до 10.05.2026 р.	
6.	Нормоконтроль	до 23.05.2026 р.	
7.	Інструментальна перевірка на академічний плагіат	до 25.05.2026 р.	
8.	Здача кваліфікаційної роботи та всіх супровідних документів на кафедру	до 28.05.2026 р.	

Здобувач вищої освіти

_____ Данило МОШКІВСЬКИЙ
(підпис) (прізвище, ініціали)

Керівник кваліфікаційної роботи

_____ Сергій ГРИНЮК
(підпис) (прізвище, ініціали)

АНОТАЦІЯ

Мошківський Д. В. Інтелектуальна система керування енергоспоживанням у розумному будинку на основі IoT. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2026.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, переліку використаних джерел, додатків.

У першому розділі роботи проаналізовано сучасні підходи до енергомоніторингу та керування енергоспоживанням у житлових приміщеннях.

У другому розділі виконано проєктування інтелектуальної IoT-системи керування енергоспоживанням. Обґрунтовано вибір мікроконтролерної платформи та апаратних компонентів, зокрема Arduino Uno, ESP32, сенсорів струму, напруги, температури й освітленості, релейного модуля, акумуляторного блока, BMS-модуля, DC-DC перетворювачів, сонячної панелі та інверторного модуля. Розроблено структурну та принципову схеми підключення компонентів системи.

У третьому розділі описано реалізацію апаратної частини системи, розробку програмного забезпечення для мікроконтролерів, організацію передавання даних до IoT-платформи та логіку автоматичного керування навантаженням. Запропонована система забезпечує зчитування параметрів енергоспоживання, їх обробку, передавання користувачу, автоматичне керування релейними каналами та реагування на перевищення заданих порогових значень.

Ключові слова: розумний будинок, Інтернет речей, енергоспоживання, енергомоніторинг, Arduino Uno, ESP32, сенсори, релейний модуль, акумуляторний блок, IoT-платформа, автоматичне керування.

ANNOTATION

Moshkivskyi D. Intelligent energy consumption management system in a smart home based on IoT. Manuscript.

Bachelor's qualification work of the EP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2026.

The qualification work consists of an introduction, three sections, conclusions, a list of sources used, and appendices.

The first section of the work analyzes modern approaches to energy monitoring and energy consumption management in residential premises.

In the second section, an intelligent IoT energy management system is designed. The choice of a microcontroller platform and hardware components is justified, in particular Arduino Uno, ESP32, current, voltage, temperature and light sensors, a relay module, a battery pack, a BMS module, DC-DC converters, a solar panel and an inverter module. A structural and schematic diagram of the system components is developed.

In the third section, the implementation of the hardware part of the system, the development of software for microcontrollers, the organization of data transfer to the IoT platform and the logic of automatic load control are described. The proposed system provides reading of energy consumption parameters, their processing, transmission to the user, automatic control of relay channels and response to exceeding the specified threshold values.

Keywords: smart home, Internet of Things, energy consumption, energy monitoring, Arduino Uno, ESP32, sensors, relay module, battery pack, IoT platform, automatic control.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ТЕХНОЛОГІЙ ТА ПРИНЦИПІВ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНИХ ІОТ-СИСТЕМ КЕРУВАННЯ ЕНЕРГОСПОЖИВАННЯМ.....	11
1.1 Аналіз сучасних підходів до енергомоніторингу та керування енергоспоживанням у житлових приміщеннях	11
1.2 Архітектура розумного будинку як середовища функціонування IoT-пристроїв.....	16
1.3 Методи збору, передавання та обробки даних про споживання електроенергії.....	20
1.4 Аналіз апаратних і програмних засобів для реалізації IoT-систем енергоменеджменту	26
РОЗДІЛ 2 ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ ІОТ-СИСТЕМИ КЕРУВАННЯ ЕНЕРГОСПОЖИВАННЯМ	33
2.1 Вибір мікроконтролерної платформи та апаратних компонентів системи	33
2.2 Розробка структурної схеми інтелектуальної системи керування енергоспоживанням	38
2.3 Розробка принципової схеми підключення сенсорів, виконавчих пристроїв та комунікаційного модуля.....	42
РОЗДІЛ 3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ КЕРУВАННЯ ЕНЕРГОСПОЖИВАННЯМ	46
3.1 Реалізація апаратної частини системи моніторингу та керування енергоспоживанням	46
3.2 Розробка програмного забезпечення для мікроконтролера.....	50
3.3 Реалізація модуля передавання даних до IoT-платформи.....	56
3.4 Розробка логіки автоматичного керування навантаженням на основі отриманих даних	60
ВИСНОВКИ	65

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	68
-----------------------------------	----

ВСТУП

Актуальність теми. Сучасний етап розвитку житлової інфраструктури характеризується активним упровадженням технологій «розумного будинку», які забезпечують автоматизоване керування інженерними системами, підвищення комфорту користувача, безпеки та енергоефективності приміщень. Одним із важливих напрямів розвитку таких систем є моніторинг і керування енергоспоживанням, оскільки зростання кількості побутових електроприладів, сенсорних пристроїв, мережевого обладнання та систем автоматизації призводить до збільшення навантаження на внутрішню електромережу будинку. За таких умов виникає потреба у створенні технічних рішень, здатних не лише фіксувати параметри споживання електроенергії, а й аналізувати їх, передавати дані користувачу та автоматично впливати на роботу окремих навантажень.

Особливої актуальності набувають IoT-системи, які дають змогу об'єднувати сенсори, мікроконтролери, виконавчі пристрої та хмарні або локальні програмні сервіси в єдину інформаційно-керуючу інфраструктуру. Використання таких рішень у розумному будинку дозволяє реалізувати дистанційний контроль електроприладів, збір даних про споживання електроенергії, виявлення надмірного навантаження, формування повідомлень для користувача та автоматичне вимкнення або перемикання окремих споживачів. При цьому важливим є не лише сам факт передавання даних, а й здатність системи приймати рішення на основі заданих правил, порогових значень або елементів інтелектуальної обробки інформації.

Технічною основою такої системи можуть бути мікроконтролерні платформи з підтримкою бездротового зв'язку, сенсори вимірювання електричних параметрів, релейні або напівпровідникові виконавчі модулі, а також програмне забезпечення для обробки, візуалізації та зберігання даних. Раціональне поєднання цих компонентів дозволяє створити компактний апаратно-програмний комплекс для контролю енергоспоживання в побутових умовах. Саме тому розробка інтелектуальної системи керування

енергоспоживанням у розумному будинку на основі IoT є актуальним завданням у галузі комп'ютерної інженерії.

Метою роботи є розробка інтелектуальної IoT-системи керування енергоспоживанням у розумному будинку, яка забезпечує збір даних про електричне навантаження, їх обробку, передавання користувачу та автоматизоване керування споживачами електроенергії.

Об'єкт дослідження – процеси моніторингу та керування енергоспоживанням у системах розумного будинку.

Предмет дослідження – апаратно-програмні засоби побудови IoT-системи для контролю, аналізу та керування енергоспоживанням побутових електричних навантажень.

Для досягнення поставленої мети в роботі необхідно вирішити такі завдання:

- проаналізувати сучасні підходи до побудови систем енергомоніторингу та керування енергоспоживанням у розумному будинку;
- розглянути архітектуру IoT-системи як середовища взаємодії сенсорів, мікроконтролерів, виконавчих пристроїв і програмних сервісів;
- обґрунтувати вибір апаратних компонентів для реалізації системи керування енергоспоживанням;
- розробити структурну та принципову схеми підключення основних компонентів системи;
- сформулювати алгоритм збору, обробки та передавання даних про енергоспоживання;
- реалізувати програмну логіку керування електричним навантаженням;
- провести тестування працездатності системи та оцінити ефективність її функціонування.

Методи дослідження. У роботі використано методи аналізу технічних рішень для систем розумного будинку, методи структурного та функціонального моделювання, принципи побудови мікроконтролерних IoT-систем, методи вимірювання електричних параметрів, алгоритмічне моделювання логіки

керування, а також експериментальне тестування апаратно-програмного прототипу.

Практичне значення роботи полягає у можливості використання розробленої системи як основи для побудови побутового або навчального прототипу інтелектуального енергомоніторингу. Запропоноване рішення може бути застосоване для контролю споживання електроенергії окремими електроприладами, автоматичного вимкнення навантаження у разі перевищення заданих параметрів, дистанційного перегляду показників і подальшого розширення функціональності системи за рахунок додаткових сенсорів, хмарних сервісів або алгоритмів прогнозування енергоспоживання.

РОЗДІЛ 1

АНАЛІЗ ТЕХНОЛОГІЙ ТА ПРИНЦИПІВ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНИХ ІОТ-СИСТЕМ КЕРУВАННЯ ЕНЕРГОСПОЖИВАННЯМ

1.1 Аналіз сучасних підходів до енергомоніторингу та керування енергоспоживанням у житлових приміщеннях

Сучасні житлові приміщення характеризуються зростанням кількості електричних споживачів, які працюють у різних режимах навантаження та мають різний вплив на загальний баланс енергоспоживання. До таких споживачів належать освітлювальні прилади, побутова техніка, системи опалення, вентиляції, кондиціонування, зарядні пристрої, мультимедійне обладнання, мережеві пристрої та елементи автоматизації розумного будинку. У таких умовах традиційний підхід до обліку електроенергії, який передбачає лише фіксацію загального споживання за певний період, не забезпечує достатньої деталізації для аналізу роботи окремих пристроїв та прийняття рішень щодо оптимізації їх використання [1].

Енергомоніторинг у розумному будинку можна розглядати як процес безперервного або періодичного вимірювання електричних параметрів, їх передавання, обробки, зберігання та візуалізації для користувача. Основними параметрами, які доцільно контролювати в таких системах, є напруга, сила струму, активна потужність, спожита електроенергія, частота мережі, коефіцієнт потужності та стан окремих навантажень. Отримання цих даних дає змогу оцінити режим роботи електричних пристроїв, виявити перевищення допустимих значень потужності, встановити нерівномірність навантаження та визначити обладнання, яке створює найбільше споживання електроенергії [1].

Узагальнену логіку енергомоніторингу в розумному будинку подано на рисунку 1.1.

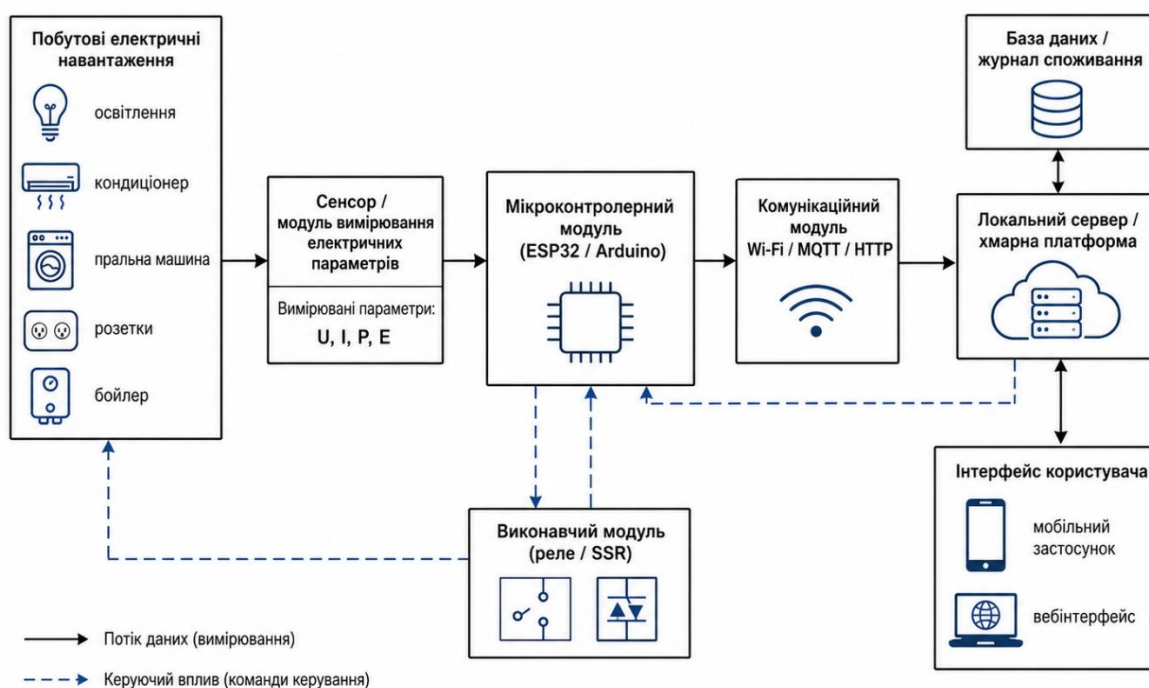


Рисунок 1.1 – Узагальнена структура енергомоніторингу в розумному будинку [1]

На рисунку 1.1 доцільно відобразити такі елементи: побутові електричні навантаження, сенсор вимірювання електричних параметрів, мікроконтролерний модуль, канал передавання даних, локальний або хмарний сервер, інтерфейс користувача та виконавчий пристрій для керування навантаженням.

У сучасних системах керування енергоспоживанням можна виокремити кілька основних підходів. Перший підхід полягає у пасивному моніторингу, коли система лише збирає дані про споживання електроенергії та відображає їх користувачу. Такий варіант є найпростішим з погляду реалізації, оскільки не передбачає автоматичного впливу на роботу електроприладів. Його перевагою є можливість отримання статистики споживання, побудови графіків, аналізу пікових навантажень і формування загального уявлення про енергетичну поведінку житлового приміщення.

Другий підхід передбачає активне керування навантаженням. У цьому випадку система не лише збирає дані, а й виконує певні дії на основі заданих правил. Наприклад, якщо потужність навантаження перевищує встановлене

порогове значення, мікроконтролер може подати сигнал на релейний модуль і вимкнути окремий електроприлад. Також можливим є автоматичне відключення навантаження у визначені часові проміжки, керування освітленням залежно від присутності користувача або зміна режиму роботи обладнання відповідно до попередньо заданого сценарію [1].

Третій підхід пов'язаний із використанням інтелектуальних алгоритмів. У такому випадку система може аналізувати накопичені дані, визначати характерні шаблони споживання, прогнозувати навантаження та пропонувати оптимальні режими роботи електричних пристроїв. Для бакалаврської роботи доцільно розглядати спрощений варіант інтелектуального керування, який базується на порогових значеннях, логічних правилах і базовому аналізі отриманих даних. Такий підхід є технічно обґрунтованим, оскільки його можна реалізувати на доступній мікроконтролерній платформі без використання складних серверних обчислень.

Класифікацію основних підходів до керування енергоспоживанням у розумному будинку подано на рисунку 1.2.

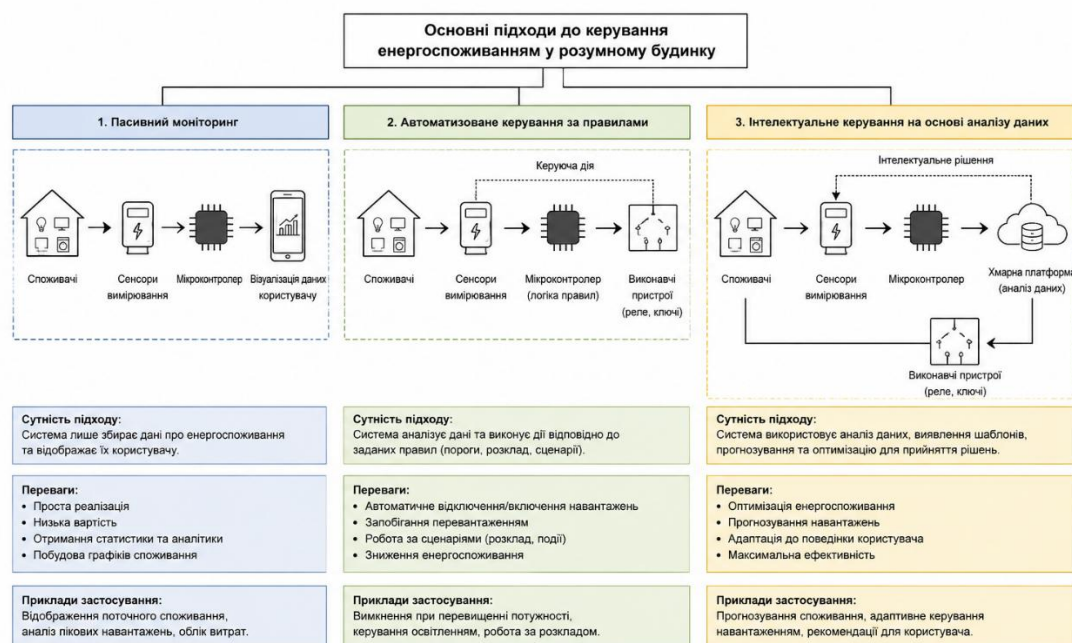


Рисунок 1.2 – Основні підходи до керування енергоспоживанням у розумному будинку [1]

На рисунку 1.2 показано три основні напрями: пасивний моніторинг, автоматизоване керування за правилами та інтелектуальне керування на основі аналізу даних.

Важливу роль у побудові таких систем відіграє технологія Інтернету речей. Її використання дає змогу об'єднати вимірювальні сенсори, мікроконтролери, виконавчі модулі та програмні сервіси в єдину систему. Завдяки цьому користувач може отримувати інформацію про стан енергоспоживання в реальному часі, переглядати дані через мобільний застосунок або вебінтерфейс, а також дистанційно керувати окремими навантаженнями. У цьому випадку мікроконтролер виконує роль проміжної ланки між фізичним середовищем і програмним рівнем системи, оскільки саме він зчитує сигнали із сенсорів, обробляє їх і передає далі для візуалізації або прийняття керуючих рішень [1].

Для практичної реалізації системи енергомоніторингу доцільно використовувати мікроконтролерні платформи з підтримкою бездротового зв'язку, наприклад ESP32. Така платформа має достатню обчислювальну продуктивність для обробки даних із сенсорів, підтримує Wi-Fi-з'єднання та може взаємодіяти з хмарними або локальними сервісами. Для вимірювання електричних параметрів можуть застосовуватися спеціалізовані модулі, які забезпечують контроль напруги, струму, потужності та спожитої енергії. Виконавча частина системи може бути побудована на основі релейних модулів або напівпровідникових ключів, які дозволяють керувати підключенням і відключенням навантаження.

З технічної точки зору система керування енергоспоживанням у розумному будинку повинна виконувати декілька взаємопов'язаних функцій. По-перше, вона має забезпечувати вимірювання параметрів електричної мережі або окремого навантаження. По-друге, система повинна виконувати первинну обробку отриманих даних, зокрема перевірку коректності показників і порівняння їх із допустимими межами. По-третє, необхідно забезпечити передавання інформації до інтерфейсу користувача. По-четверте, система має

реалізовувати керуючий вплив на навантаження у випадку перевищення встановлених порогів або виконання заданого сценарію роботи.

Загальний принцип функціонування інтелектуальної системи керування енергоспоживанням подано на рисунку 1.3.

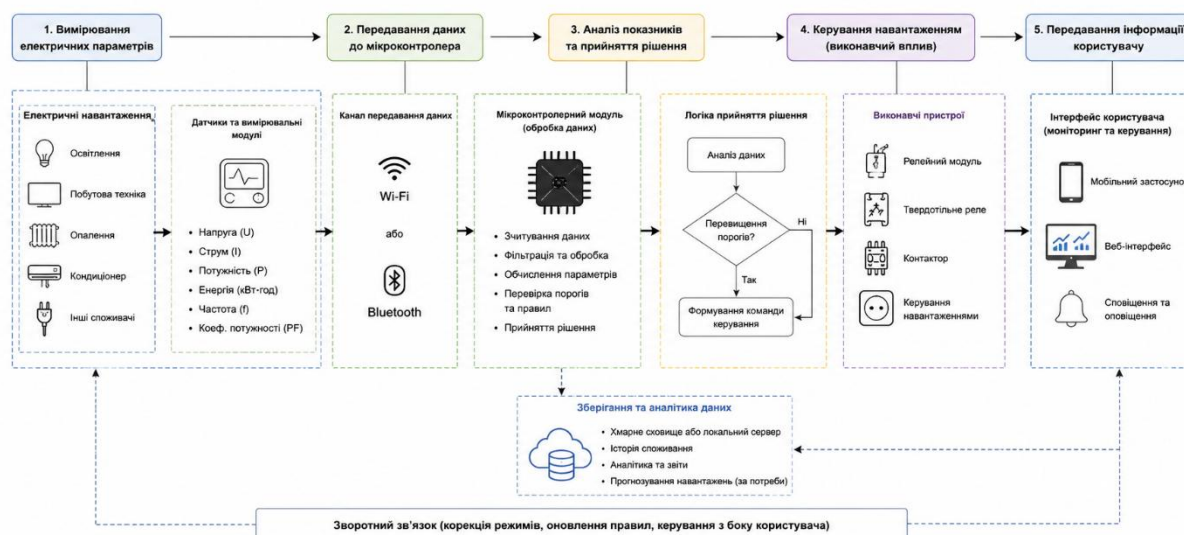


Рисунок 1.3 – Принцип функціонування інтелектуальної системи керування енергоспоживанням [1]

На рисунку 1.3 доцільно зобразити послідовність роботи системи: вимірювання електричних параметрів - передавання даних до мікроконтролера - аналіз показників - прийняття рішення - керування навантаженням - передавання інформації користувачу.

Особливістю сучасних IoT-систем енергоменеджменту є можливість поєднання локального та віддаленого керування. Локальний рівень забезпечує безпосередню взаємодію з датчиками та виконавчими пристроями, а віддалений рівень відповідає за зберігання, візуалізацію та аналітичну обробку даних. Такий підхід підвищує гнучкість системи, оскільки користувач може контролювати споживання електроенергії не лише перебуваючи вдома, а й через мережу Інтернет. Крім того, використання хмарних сервісів або локальних платформ дає змогу накопичувати історичні дані, що є важливим для подальшого аналізу ефективності роботи системи.

Сучасні підходи до енергомоніторингу та керування енергоспоживанням у житлових приміщеннях базуються на поєднанні сенсорних технологій, мікроконтролерних платформ, бездротового зв'язку, програмних сервісів та алгоритмів автоматизованого прийняття рішень. Найбільш доцільним для реалізації в межах бакалаврської роботи є підхід, за якого система виконує вимірювання електричних параметрів, аналізує їх за заданими правилами, передає дані користувачу та керує навантаженням через виконавчий модуль. Така модель є технічно обґрунтованою, доступною для практичної реалізації та відповідає загальним принципам побудови інтелектуальних IoT-систем у розумному будинку.

1.2 Архітектура розумного будинку як середовища функціонування IoT-пристроїв

Архітектура розумного будинку є сукупністю взаємопов'язаних апаратних, програмних і мережевих компонентів, які забезпечують автоматизований контроль, моніторинг та керування інженерними системами житлового приміщення. У контексті розробки інтелектуальної системи керування енергоспоживанням така архітектура розглядається не лише як набір окремих електронних пристроїв, а як єдине інформаційно-керуюче середовище, у якому сенсори, мікроконтролери, виконавчі модулі, засоби зв'язку та користувацькі інтерфейси взаємодіють між собою для досягнення енергоефективної роботи побутових навантажень [1].

Основною особливістю розумного будинку є здатність технічних засобів отримувати інформацію про стан середовища, передавати її на обробку, виконувати аналіз і формувати керуючі дії. У системах енергомоніторингу такими даними можуть бути значення напруги, струму, активної потужності, спожитої електроенергії, температури, вологості, освітленості, присутності користувача або стану конкретного електричного навантаження. Завдяки цьому система може не лише відображати поточні показники, а й реагувати на зміну

умов експлуатації, наприклад вимикати навантаження у разі перевищення допустимої потужності або переводити окремі пристрої в енергоощадний режим [2].

З технічної точки зору архітектуру розумного будинку доцільно поділяти на кілька рівнів: рівень фізичних пристроїв, рівень збору та первинної обробки даних, комунікаційний рівень, рівень зберігання та аналітики, а також рівень взаємодії з користувачем. На фізичному рівні розміщуються електричні навантаження, сенсори та виконавчі пристрої. До них належать побутова техніка, освітлення, розетки, релейні модулі, твердотільні реле, датчики електричних параметрів і модулі вимірювання спожитої енергії. Саме цей рівень забезпечує безпосередній зв'язок системи з реальним електричним середовищем будинку.

Рівень збору та первинної обробки даних виконує мікроконтролерна платформа. У межах бакалаврської роботи доцільно розглядати використання ESP32 або Arduino з додатковим комунікаційним модулем, оскільки такі платформи дають змогу зчитувати сигнали із сенсорів, виконувати базову обробку даних, порівнювати показники з пороговими значеннями та формувати сигнали керування для виконавчих пристроїв. Наприклад, мікроконтролер може отримувати дані від модуля вимірювання електричних параметрів, обчислювати поточне навантаження та у випадку перевищення заданого ліміту активувати реле для відключення споживача.

На рисунку 1.4 відображено п'ять основних блоків: сенсори та електричні навантаження, мікроконтролерний модуль, комунікаційний канал, локальний або хмарний сервер, інтерфейс користувача. Окремо варто показати виконавчий модуль, який отримує керуючі команди від мікроконтролера та впливає на роботу побутових споживачів.

Комунікаційний рівень забезпечує передавання даних між пристроями системи. У розумному будинку можуть використовуватися як дротові, так і бездротові технології зв'язку. До поширених бездротових рішень належать Wi-Fi, Bluetooth, Zigbee, Z-Wave та LoRa. Для навчального або експериментального прототипу системи керування енергоспоживанням

найбільш доцільним є Wi-Fi, оскільки він підтримується мікроконтролером ESP32 і дає змогу передавати дані безпосередньо до локального сервера, вебінтерфейсу або хмарної IoT-платформи. Для обміну повідомленнями між пристроями можуть застосовуватися протоколи MQTT або HTTP, де MQTT є зручним для IoT-систем завдяки легкій структурі повідомлень і можливості організації обміну даними за моделлю «публікація-підписка» [3].

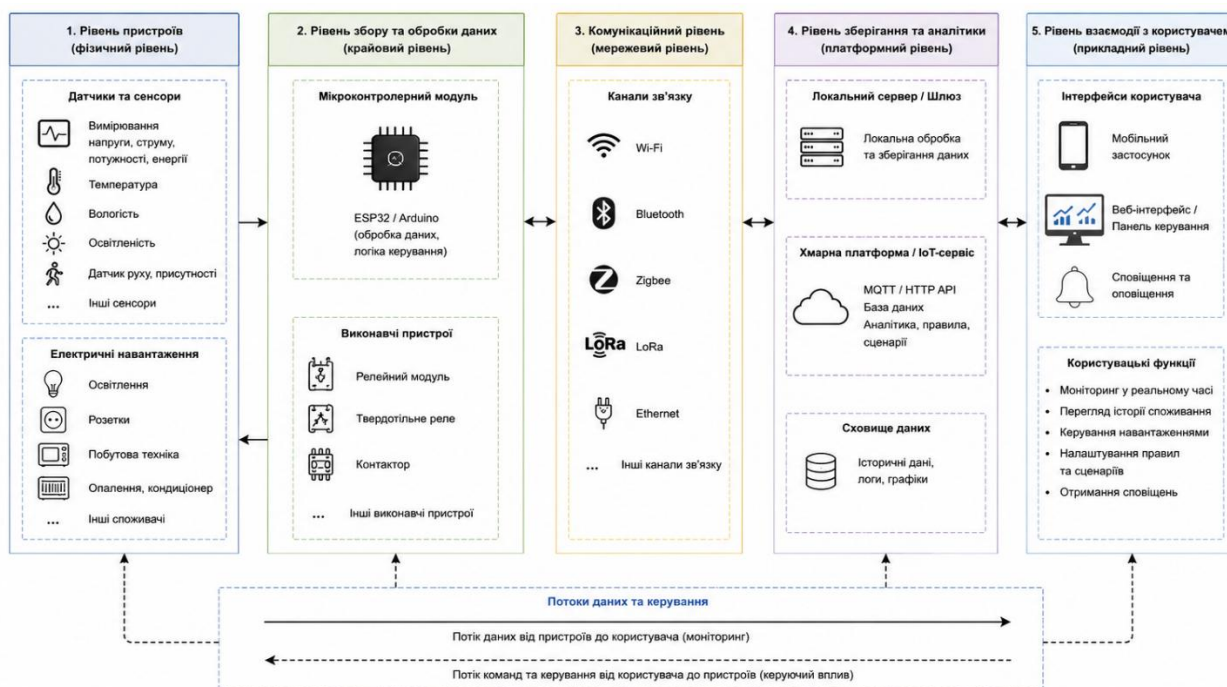


Рисунок 1.4 – Узагальнена архітектура розумного будинку як середовища функціонування IoT-пристроїв [3]

Важливою складовою архітектури є рівень зберігання та аналітики даних. Він може бути реалізований у вигляді локального сервера, хмарної платформи або окремого програмного середовища, наприклад Node-RED, ThingsBoard, Blynk або власного вебзастосунку. На цьому рівні здійснюється накопичення історичних даних, побудова графіків споживання, формування звітів, аналіз пікових навантажень і налаштування правил керування. Для інтелектуальної системи керування енергоспоживанням цей рівень є особливо важливим, оскільки саме на основі зібраних даних можна оцінити ефективність роботи системи та визначити можливості для зниження витрат електроенергії.

Рівень взаємодії з користувачем забезпечує доступ до інформації про стан системи. Користувач може переглядати поточні значення напруги, струму, потужності та спожитої електроенергії через мобільний застосунок, вебінтерфейс або інформаційну панель. Крім того, інтерфейс користувача може надавати можливість ручного керування навантаженням, перегляду історії споживання, зміни порогових значень і отримання сповіщень про перевищення допустимих параметрів. Такий підхід підвищує зручність експлуатації системи та дає змогу користувачу контролювати роботу електричних пристроїв у режимі реального часу.

У межах інтелектуальної системи керування енергоспоживанням архітектура розумного будинку повинна забезпечувати двосторонній обмін даними. У прямому напрямі система отримує виміряні параметри від сенсорів, передає їх до мікроконтролера, а далі – до програмного рівня для візуалізації або аналізу. У зворотному напрямі формуються керуючі команди, які надходять до виконавчих пристроїв і змінюють стан електричного навантаження. Такий принцип роботи дозволяє реалізувати не лише моніторинг, а й активне керування споживанням електроенергії.

Для забезпечення стабільної роботи архітектури важливо враховувати питання надійності, масштабованості та безпеки. Надійність передбачає коректну роботу сенсорів, стабільне живлення мікроконтролера, захист виконавчих модулів від перевантажень і правильне електричне підключення. Масштабованість означає можливість додавання нових сенсорів, реле, розумних розеток або додаткових зон контролю без повної зміни структури системи. Безпека передбачає захист передавання даних, обмеження несанкціонованого доступу до інтерфейсу керування та використання безпечних режимів роботи з електричними навантаженнями.

Отже, архітектура розумного будинку є технічною основою для функціонування IoT-системи керування енергоспоживанням. Вона поєднує фізичні пристрої, сенсори, мікроконтролери, комунікаційні засоби, програмні сервіси та інтерфейси користувача в єдину систему. Для розроблюваної системи

найбільш доцільною є багаторівнева архітектура, у якій мікроконтролерний модуль виконує роль центрального вузла збору та первинної обробки даних, сенсори забезпечують вимірювання електричних параметрів, а виконавчі пристрої реалізують керуючий вплив на навантаження. Такий підхід відповідає принципам побудови сучасних IoT-рішень і створює основу для подальшої програмно-апаратної реалізації інтелектуальної системи енергоменеджменту в розумному будинку.

1.3 Методи збору, передавання та обробки даних про споживання електроенергії

У системах розумного будинку дані про споживання електроенергії є основою для подальшого аналізу, візуалізації та автоматизованого керування електричними навантаженнями. На відміну від традиційного побутового обліку, який зазвичай фіксує лише загальну кількість спожитої електроенергії за певний період, IoT-система енергомоніторингу дає змогу отримувати більш деталізовану інформацію про роботу окремих пристроїв. Такий підхід дозволяє визначати поточну потужність навантаження, виявляти перевищення допустимих параметрів, аналізувати пікові режими роботи та формувати керуючі дії для підвищення енергоефективності житлового приміщення [3].

Збір даних у системі керування енергоспоживанням передбачає вимірювання основних електричних параметрів, які характеризують стан навантаження. До таких параметрів належать напруга, сила струму, активна потужність, спожита електроенергія, частота електромережі та коефіцієнт потужності. Для практичної реалізації подібних систем можуть використовуватися спеціалізовані вимірювальні модулі, наприклад PZEM-004T, SCT-013, INA219 або інші сенсори струму та напруги. Вибір конкретного модуля залежить від типу мережі, допустимого діапазону вимірювання, точності, способу підключення до мікроконтролера та безпеки експлуатації [4].

Для побутових систем енергомоніторингу важливо не лише зчитувати значення електричних параметрів, а й забезпечувати їх коректну інтерпретацію. Наприклад, значення струму саме по собі не дає повної інформації про енергоспоживання, тому його доцільно аналізувати разом із напругою та активною потужністю. Потужність електричного навантаження може визначатися як добуток напруги, струму та коефіцієнта потужності. У спрощеному вигляді для активного навантаження вона може бути подана формулою (1.1):

$$P = U \cdot I, \quad (1.1)$$

де P – активна потужність, Вт;

U – напруга, В;

I – сила струму, А.

Якщо враховується коефіцієнт потужності, то розрахунок виконується за формулою (1.2):

$$P = U \cdot I \cdot \cos \varphi, \quad (1.2)$$

де $\cos \varphi$ – коефіцієнт потужності навантаження.

Для визначення кількості спожитої електроенергії використовується залежність (1.3):

$$E = P \cdot t, \quad (1.3)$$

де E – спожита електроенергія, Вт·год або кВт·год;

P – потужність навантаження;

t – час роботи споживача.

Узагальнений процес збору даних про споживання електроенергії в IoT - системі подано на рисунку 1.5.

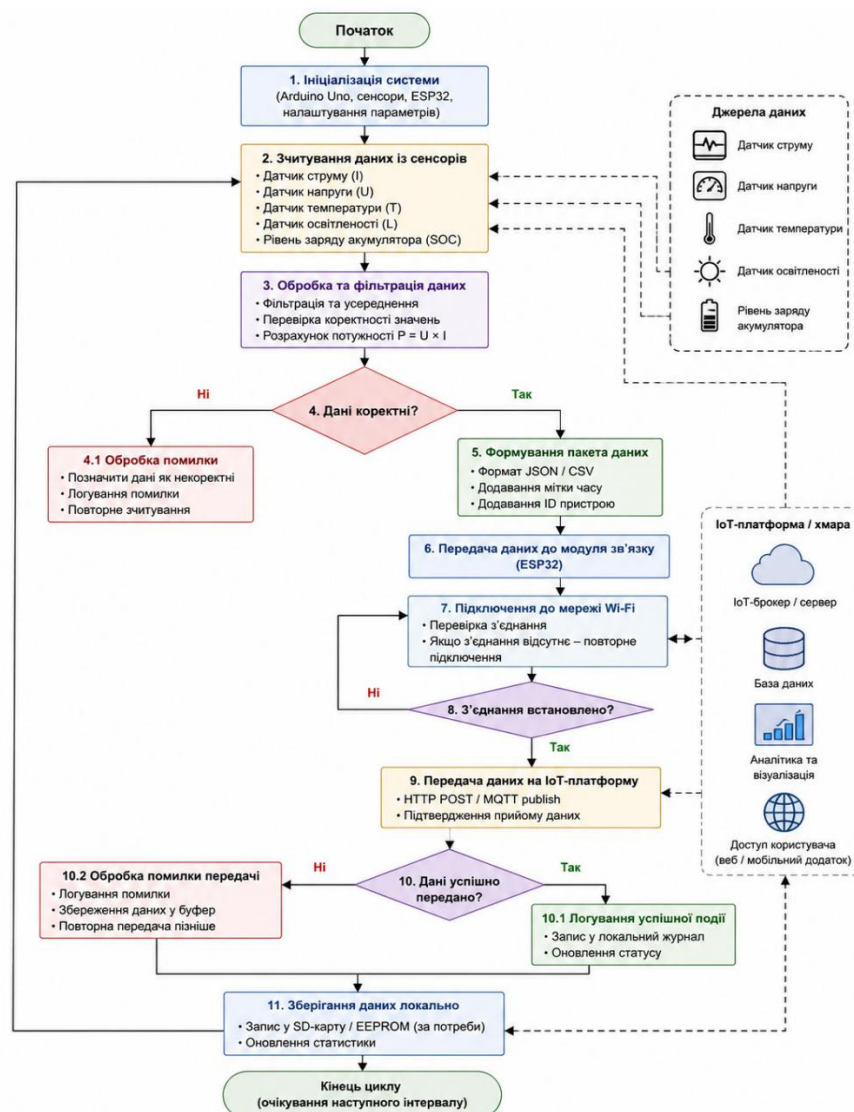


Рисунок 1.5 – Блок-схема процесу збору даних про споживання електроенергії в IoT-системі [4]

На рисунку 1.5 відображено такі етапи: електричне навантаження – вимірювальний модуль – мікроконтролер – первинна обробка даних – передавання інформації до користувацького інтерфейсу або хмарної платформи.

Після збору вимірювальної інформації важливим етапом є передавання даних між компонентами системи. У розумному будинку можуть застосовуватися як дротові, так і бездротові канали зв'язку. Дротове передавання є стабільним і менш чутливим до завад, однак потребує прокладання кабельної інфраструктури. Бездротові технології є більш зручними для побутового

застосування, оскільки дозволяють розміщувати IoT-пристрої в різних частинах приміщення без суттєвого втручання в інженерну інфраструктуру будинку [3].

Серед бездротових технологій у системах розумного будинку поширеними є Wi-Fi, Bluetooth, Zigbee, Z-Wave та LoRa. Для розроблюваної системи керування енергоспоживанням доцільним є використання Wi-Fi, оскільки ця технологія підтримується мікроконтролерними платформами ESP32, забезпечує достатню швидкість передавання даних і дозволяє інтегрувати пристрій із локальним сервером, хмарною платформою або вебінтерфейсом. У межах навчального прототипу Wi-Fi є технічно зручним рішенням, оскільки не потребує додаткових шлюзів і може працювати в межах наявної домашньої мережі.

Для організації обміну повідомленнями між мікроконтролером і програмною платформою можуть використовуватися протоколи HTTP або MQTT. Протокол HTTP є зрозумілим і простим для реалізації, оскільки дозволяє передавати дані у вигляді запитів до вебсервера або хмарного сервісу. Проте для IoT-систем більш доцільним часто є MQTT, оскільки він має менші накладні витрати та працює за моделлю «публікація-підписка». У такій моделі мікроконтролер публікує дані у певну тему, а сервер, застосунок або панель моніторингу підписуються на цю тему й отримують оновлення в режимі, близькому до реального часу [5].

На рисунку 1.6 представлено мікроконтролер ESP32, який отримує дані від вимірювального модуля, передає їх через Wi-Fi за допомогою MQTT або HTTP до хмарної платформи чи локального сервера, після чого користувач переглядає інформацію через мобільний застосунок або вебінтерфейс.

Обробка даних у системі керування енергоспоживанням може виконуватися на кількох рівнях. Перший рівень – це первинна обробка безпосередньо на мікроконтролері. На цьому етапі здійснюється зчитування даних із сенсора, перевірка коректності значень, фільтрація випадкових відхилень, обчислення потужності та порівняння отриманих показників із заданими порогоми. Наприклад, якщо потужність навантаження перевищує

встановлене значення, мікроконтролер може сформувати команду для вимкнення релейного модуля або передати повідомлення користувачу.



Рисунок 1.6 – Схема передавання даних у системі енергомоніторингу розумного будинку [5]

Другий рівень обробки пов'язаний із серверною або хмарною частиною системи. На цьому рівні можуть зберігатися історичні дані, будуватися графіки зміни споживання електроенергії, формуватися звіти, аналізуватися добові або тижневі профілі навантаження. Накопичення історії споживання є важливим для подальшої оцінки ефективності роботи системи, оскільки користувач може порівнювати різні періоди часу, визначати найбільш енерговитратні пристрої та приймати рішення щодо оптимізації їх використання.

Третій рівень обробки може передбачати використання елементів інтелектуального аналізу. Для роботи доцільно застосовувати спрощену логіку, яка базується на правилах, порогових значеннях і сценаріях роботи. Наприклад, система може автоматично вимикати навантаження у разі перевищення встановленої потужності, формувати сповіщення при тривалій роботі приладу або рекомендувати користувачу змінити режим використання обладнання. У

складніших системах можуть застосовуватися методи прогнозування споживання, класифікації режимів роботи та виявлення аномалій, однак для навчального прототипу достатньо реалізувати базовий алгоритм автоматизованого прийняття рішень [5].

З технічної точки зору обробка даних повинна враховувати точність вимірювання, періодичність опитування сенсорів і стабільність передавання інформації. Якщо дані зчитуються занадто рідко, система може не зафіксувати короточасні пікові навантаження. Якщо ж період опитування є надто малим, це може створювати зайве навантаження на мікроконтролер, мережу та серверну частину. Тому в побутових IoT-системах доцільно встановлювати оптимальний інтервал оновлення даних, який забезпечує достатню інформативність без перевантаження системи.

Для підвищення достовірності отриманих результатів доцільно застосовувати фільтрацію вимірювань. Найпростішим варіантом є усереднення кількох послідовних значень, що дозволяє зменшити вплив випадкових коливань. Також система може ігнорувати некоректні значення, які виходять за межі фізично можливого діапазону. Наприклад, якщо сенсор передає нульову напругу за наявності активного навантаження або надто велике значення струму, такі дані потребують додаткової перевірки перед використанням у керуючому алгоритмі.

Окрему увагу необхідно приділяти безпеці збору та передавання даних. Оскільки система енергомоніторингу може мати можливість дистанційного керування електричними навантаженнями, несанкціонований доступ до неї може створювати ризики для користувача та обладнання. Тому під час проєктування доцільно передбачати використання захищеного доступу до інтерфейсу, автентифікації користувача, обмеження прав керування та, за можливості, шифрування мережевої взаємодії. Навіть у навчальному прототипі ці принципи мають бути враховані на рівні архітектури системи.

Таким чином, методи збору, передавання та обробки даних про споживання електроенергії є ключовою складовою інтелектуальної системи

керування енергоспоживанням у розумному будинку. Збір даних забезпечується вимірювальними модулями, передавання – за допомогою дротових або бездротових каналів зв'язку, а обробка - на рівні мікроконтролера та програмної платформи. Для розроблюваної системи найбільш доцільною є модель, у якій мікроконтролер ESP32 отримує дані від вимірювального сенсора, виконує первинний аналіз, передає інформацію через Wi-Fi до інтерфейсу користувача та формує керуючі сигнали для виконавчих пристроїв. Такий підхід забезпечує технічну цілісність системи та створює основу для подальшої реалізації інтелектуального керування енергоспоживанням.

1.4 Аналіз апаратних і програмних засобів для реалізації IoT-систем енергоменеджменту

Реалізація інтелектуальної системи керування енергоспоживанням у розумному будинку потребує обґрунтованого вибору апаратних і програмних засобів, які забезпечують вимірювання електричних параметрів, обробку даних, передавання інформації користувачу та формування керуючих дій для виконавчих пристроїв. Така система не може розглядатися лише як окремий вимірювальний пристрій, оскільки її функціонування базується на взаємодії сенсорного, мікроконтролерного, комунікаційного, програмного та виконавчого рівнів. Саме узгоджена робота цих рівнів дозволяє створити повноцінну IoT-систему енергоменеджменту, придатну для використання в умовах розумного будинку [6].

До основних апаратних засобів системи енергоменеджменту належать мікроконтролерна платформа, вимірювальні модулі електричних параметрів, виконавчі пристрої керування навантаженням, блок живлення, засоби індикації та комунікаційні модулі. Центральним елементом такої системи є мікроконтролер, який виконує функції збору даних із сенсорів, первинної обробки вимірювань, порівняння параметрів із заданими порогоми, передавання інформації до програмного середовища та керування виконавчими пристроями.

У системах розумного будинку доцільним є використання мікроконтролерів із вбудованими засобами бездротового зв'язку, оскільки це спрощує інтеграцію пристрою в домашню мережу та зменшує кількість додаткових модулів.

Однією з найбільш придатних платформ для побудови IoT-систем енергоменеджменту є ESP32. Її перевагою є наявність вбудованого Wi-Fi та Bluetooth, достатня кількість цифрових і аналогових інтерфейсів, підтримка протоколів UART, SPI, I2C, а також можливість програмування в середовищі Arduino IDE або ESP-IDF. У порівнянні з класичною платою Arduino Uno, ESP32 краще підходить для IoT-рішень, оскільки не потребує окремого Wi-Fi-модуля та має вищу продуктивність для обробки даних і мережевої взаємодії. Водночас Arduino Uno може використовуватися в навчальних або простих прототипах, де основний акцент робиться на зчитуванні даних і локальному керуванні без складної мережевої інтеграції [7].

Для вимірювання електричних параметрів у побутових системах можуть застосовуватися спеціалізовані модулі, які дають змогу визначати напругу, струм, активну потужність, частоту, коефіцієнт потужності та спожиту електроенергію. Одним із поширених варіантів є модуль PZEM-004T, який використовується для контролю параметрів однофазної мережі змінного струму. Його перевагою є можливість отримання одразу кількох параметрів, що важливо для задач енергомоніторингу. Також у подібних системах можуть застосовуватися датчики струму SCT-013, модулі INA219, ACS712 або інші вимірювальні засоби. Вибір конкретного сенсора залежить від типу навантаження, діапазону вимірювань, способу підключення, вимог до точності та рівня електробезпеки [8].

Виконавча частина системи призначена для безпосереднього впливу на електричне навантаження. Найпростішим варіантом є використання релейного модуля, який дозволяє вмикати або вимикати побутовий споживач за командою мікроконтролера. Для частого перемикавання або роботи з навантаженнями, де небажане механічне зношування контактів, можуть використовуватися твердотільні реле. У системах керування потужними навантаженнями також

можуть застосовуватися контактори або силові ключі. При виборі виконавчого пристрою необхідно враховувати номінальний струм, напругу, тип навантаження, наявність гальванічної розв'язки та умови безпечного монтажу.

Узагальнену структуру апаратних засобів IoT-системи енергоменеджменту подано на рисунку 1.7.

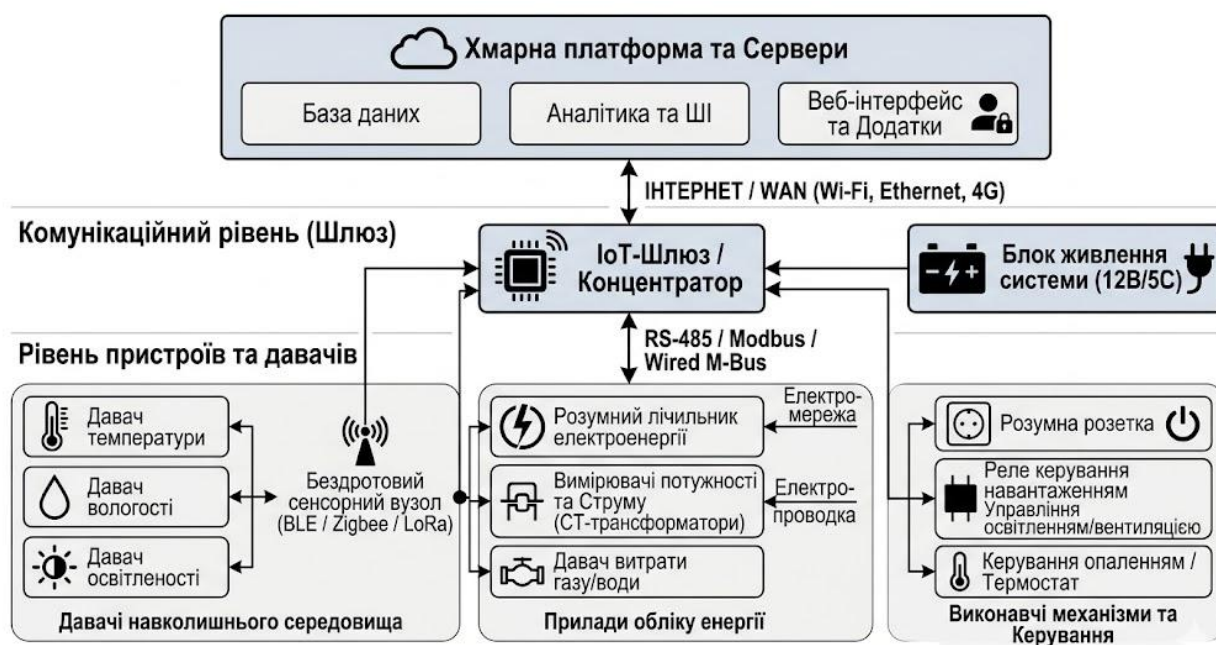


Рисунок 1.7 – Структура апаратних засобів IoT-системи енергоменеджменту [6]

На рисунку 1.7 представлено такі блоки: електричне навантаження, вимірювальний модуль, мікроконтролер ESP32, релейний або твердотільний виконавчий модуль, блок живлення, Wi-Fi-з'єднання, хмарна або локальна платформа та інтерфейс користувача. Така схема демонструє, що апаратна частина системи виконує не лише вимірювальну функцію, а й забезпечує керуючий вплив на навантаження.

Окрему увагу необхідно приділяти системі живлення. Мікроконтролерні плати та сенсорні модулі зазвичай працюють від низької напруги, наприклад 3,3 В або 5 В, тоді як побутові електричні навантаження живляться від мережі 220 В. Тому під час розробки системи потрібно забезпечити електричне розділення низьковольтної та силової частин. Це зменшує ризик пошкодження мікроконтролера та підвищує безпеку експлуатації пристрою. Для живлення

керуючої частини доцільно використовувати стабілізоване джерело живлення, а для комутації навантаження - виконавчі модулі з відповідними електричними характеристиками.

Програмні засоби IoT-системи енергоменеджменту виконують функції налаштування мікроконтролера, зчитування даних із сенсорів, обробки вимірювань, реалізації алгоритмів керування, передавання даних і візуалізації інформації. На рівні мікроконтролера програмне забезпечення може бути розроблене в Arduino IDE, PlatformIO або ESP-IDF. Для бакалаврської роботи доцільним є використання Arduino IDE або PlatformIO, оскільки ці середовища мають зрозумілу структуру, підтримують велику кількість бібліотек і дозволяють швидко реалізувати обмін даними з сенсорами та мережевими сервісами.

Алгоритм роботи програмного забезпечення мікроконтролера зазвичай передбачає ініціалізацію апаратних модулів, підключення до мережі Wi-Fi, періодичне зчитування показників із вимірювального модуля, перевірку коректності отриманих даних, обчислення додаткових параметрів, порівняння потужності або струму з пороговими значеннями, передавання інформації на сервер або в хмару та керування виконавчим модулем. У разі перевищення встановленого порогу система може автоматично вимкнути навантаження, сформувати повідомлення для користувача або записати подію до журналу.

На рівні передавання даних можуть використовуватися протоколи HTTP та MQTT. HTTP є зручним для передавання даних на вебсервер або API, однак у системах Інтернету речей часто доцільніше застосовувати MQTT. Протокол MQTT працює за моделлю «публікація-підписка», що дозволяє мікроконтролеру надсилати дані до певної теми, а серверу або користувацькому інтерфейсу – отримувати їх без постійного опитування пристрою. Такий підхід зменшує навантаження на мережу та є зручним для передавання показників енергоспоживання в режимі реального часу [9].

Для візуалізації та зберігання даних можуть застосовуватися різні програмні платформи. Наприклад, Blynk дозволяє швидко створити мобільний

інтерфейс для відображення показників і керування пристроями. ThingsBoard є більш функціональним рішенням для побудови панелей моніторингу, зберігання телеметрії та налаштування правил обробки даних. Node-RED може використовуватися для створення логіки обробки повідомлень, інтеграції з MQTT-брокером і побудови простих вебпанелей. У навчальному прототипі також можна реалізувати власний вебінтерфейс, який працює безпосередньо на ESP32 або на локальному сервері.

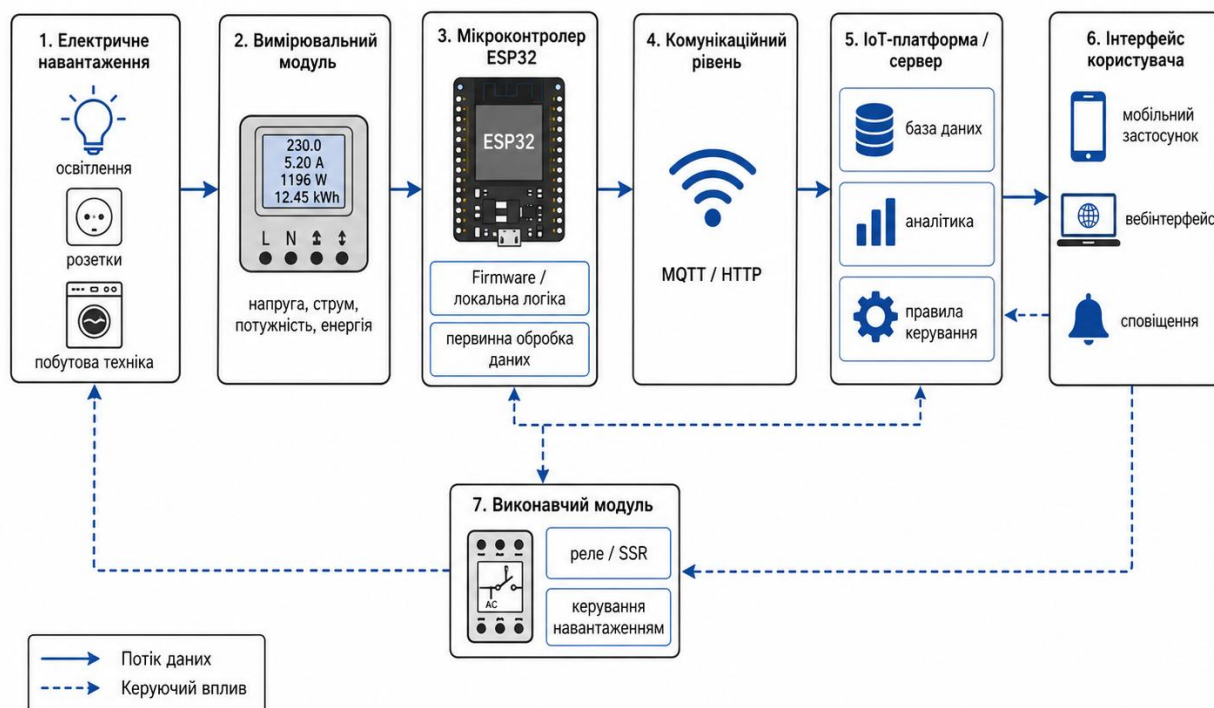
Порівняльну характеристику основних апаратних і програмних засобів для реалізації IoT-системи енергоменеджменту наведено в таблиці 1.1.

Таблиця 1.1 – Основні апаратні та програмні засоби для реалізації IoT - системи енергоменеджменту

Засіб	Призначення в системі	Переваги	Обмеження
ESP32	Збір, обробка та передавання даних	Вбудований Wi-Fi, достатня продуктивність, підтримка IoT-протоколів	Потребує стабільного живлення та правильного налаштування мережі
Arduino Uno	Локальне керування та зчитування сенсорів	Простота програмування, доступність, велика кількість прикладів	Немає вбудованого Wi-Fi, нижча продуктивність
PZEM-004T	Вимірювання напруги, струму, потужності та енергії	Дає комплексні показники електромережі	Потребує обережного підключення до мережі змінного струму
SCT-013	Вимірювання струму без розриву кола	Безконтактний принцип вимірювання струму	Для розрахунку потужності потрібні додаткові дані про напругу
Релейний модуль	Вмикання та вимикання навантаження	Простота використання, низька вартість	Механічне зношування контактів, обмеження за струмом
Твердотільне реле	Комутація навантаження без механічних контактів	Безшумна робота, висока швидкість перемикавання	Потребує правильного теплового режиму
MQTT	Передавання телеметрії та команд	Легкий протокол, зручний для IoT	Потребує MQTT-брокера
HTTP	Передавання даних до вебсервера	Простота реалізації та сумісність із вебсервісами	Більші накладні витрати порівняно з MQTT

Як видно з таблиці 1.1, вибір компонентів залежить від функціонального призначення системи та складності її реалізації. Для простого прототипу достатньо мікроконтролера ESP32, вимірювального модуля, реле та базового вебінтерфейсу. Для більш функціональної системи доцільно використовувати MQTT-брокер, платформу візуалізації, базу даних і розширену логіку обробки подій. У такому випадку система зможе не лише відображати поточні значення, а й накопичувати історію споживання, формувати графіки та виконувати автоматизоване керування.

Програмно-апаратну взаємодію компонентів IoT-системи енергоменеджменту подано на рисунку 1.8.



до програмної платформи, після чого користувач переглядає показники й може надсилати команди керування назад до системи.

Під час вибору апаратних і програмних засобів необхідно враховувати не лише функціональність, а й надійність та безпеку. Оскільки система працює з електричними навантаженнями, неправильне підключення сенсорів або виконавчих пристроїв може призвести до пошкодження обладнання. Тому силова частина має бути ізольована від логічної частини, а керування навантаженням повинно виконуватися через модулі, розраховані на відповідні струми та напруги. Крім того, програмне забезпечення має передбачати обробку помилкових або відсутніх даних, щоб система не виконувала некоректні керуючі дії.

Важливим критерієм є масштабованість системи. Якщо в майбутньому потрібно буде контролювати не один електроприлад, а декілька зон або груп навантажень, архітектура повинна дозволяти додавання нових вимірювальних модулів і виконавчих пристроїв. Для цього доцільно використовувати модульний підхід, за якого кожен вузол системи виконує окрему функцію: вимірювання, обробку, передавання даних або керування. Такий підхід полегшує модернізацію системи та дає змогу поступово розширювати її можливості.

Для реалізації інтелектуальної IoT-системи керування енергоспоживанням доцільно використовувати апаратно-програмну архітектуру, побудовану на базі мікроконтролера ESP32, вимірювального модуля електричних параметрів, виконавчого релейного або твердотільного модуля, Wi-Fi-з'єднання та програмної платформи для візуалізації даних. Така структура є технічно обґрунтованою, оскільки забезпечує збір показників енергоспоживання, їх первинну обробку, передавання користувачу та автоматизоване керування навантаженням. У межах бакалаврської роботи запропонований підхід є достатньо доступним для практичної реалізації та водночас відповідає основним принципам побудови сучасних IoT-систем енергоменеджменту в розумному будинку.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ ІОТ-СИСТЕМИ КЕРУВАННЯ ЕНЕРГОСПОЖИВАННЯМ

2.1 Вибір мікроконтролерної платформи та апаратних компонентів системи

Вибір мікроконтролерної платформи та апаратних компонентів інтелектуальної системи керування енергоспоживанням у розумному будинку необхідно виконувати з урахуванням того, що розроблюваний пристрій має не лише вимірювати параметри електроспоживання, а й забезпечувати керування навантаженнями, роботу від автономного джерела живлення, передавання даних до користувача та взаємодію з окремими виконавчими модулями. З огляду на подану апаратну схему, система має комбіновану структуру, у якій використано мікроконтролер Arduino Uno для локального керування апаратними вузлами та модуль ESP32 для реалізації бездротового IoT-зв'язку. Такий підхід є доцільним для навчального прототипу, оскільки Arduino Uno зручно застосовувати для роботи з релейними модулями, сенсорами та простими виконавчими елементами, тоді як ESP32 забезпечує підключення до мережі Wi-Fi, передавання телеметрії та інтеграцію з хмарним або локальним програмним середовищем [10].

Платформа Arduino Uno є доцільною для керування релейними модулями, зчитування сигналів із простих сенсорів і реалізації базових алгоритмів локального керування. Вона побудована на мікроконтролері ATmega328P, має 14 цифрових входів-виходів, 6 аналогових входів і працює на тактовій частоті 16 МГц, що є достатнім для виконання типових задач керування у навчальному апаратному прототипі [11]. У розроблюваній системі Arduino Uno може відповідати за обробку сигналів від сенсорів, керування багатоканальним релейним модулем, контроль стану окремих виконавчих пристроїв і формування команд для перемикання електричних навантажень.

Модуль ESP32 у цій системі доцільно розглядати як комунікаційний IoT-вузол. На відміну від Arduino Uno, ESP32 має вбудовані засоби бездротового зв'язку Wi-Fi і Bluetooth, тому може передавати дані про стан системи до мобільного застосунку, вебінтерфейсу, локального сервера або хмарної IoT-платформи. У контексті інтелектуального керування енергоспоживанням ESP32 може виконувати функції віддаленого моніторингу, надсилання сповіщень, отримання команд користувача та обміну даними за протоколами HTTP або MQTT. Таким чином, поєднання Arduino Uno та ESP32 дозволяє розділити функції локального апаратного керування і мережевої взаємодії, що підвищує гнучкість системи [10].

Подану електричну схему апаратної частини системи представлено як рисунок 2.1.

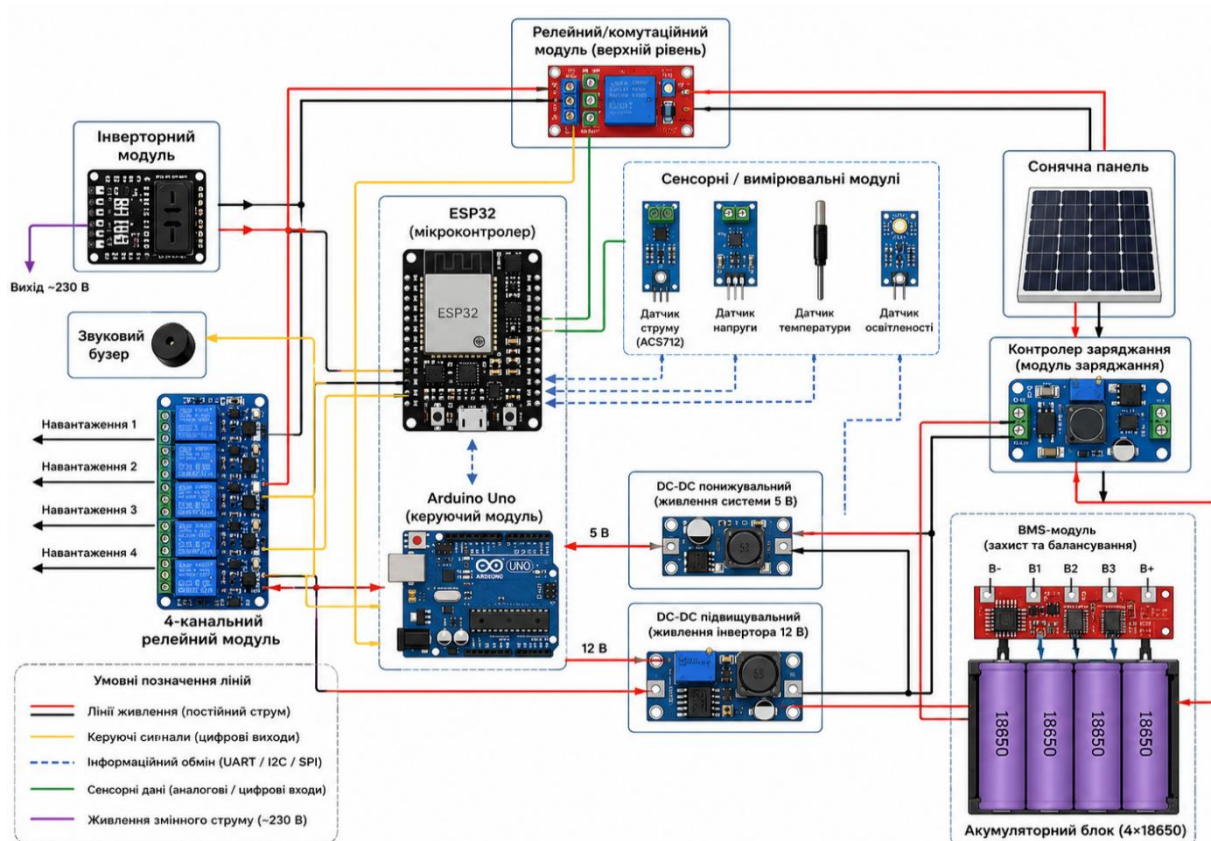


Рисунок 2.1 – Схема підключення апаратних компонентів інтелектуальної системи керування енергоспоживанням

На рисунку 2.1 показано взаємодію мікроконтролерів Arduino Uno та ESP32 із сенсорними модулями, релейним блоком, акумуляторними елементами, сонячною панеллю, перетворювачами напруги та інверторним модулем. Така структура демонструє, що система орієнтована не лише на моніторинг електроспоживання, а й на використання автономного або резервного джерела живлення, що є актуальним для розумного будинку.

Для організації обміну даними між пристроєм і програмною платформою доцільно використовувати HTTP або MQTT. Особливо зручним для IoT-систем є MQTT, оскільки цей протокол працює за моделлю «публікація-підписка», має невеликі накладні витрати та призначений для обміну повідомленнями в умовах обмежених ресурсів пристроїв [12]. У такій архітектурі ESP32 може публікувати дані про стан системи в окремі теми MQTT, а інтерфейс користувача або серверна частина можуть отримувати ці дані для моніторингу, аналітики та формування команд керування.

З урахуванням поданої схеми важливою складовою системи є підсистема автономного або резервного живлення. Вона може включати сонячну панель, акумуляторні елементи типу 18650, модуль заряджання, плату захисту акумуляторів і DC-DC перетворювачі. Використання літій-іонних акумуляторів потребує обов'язкового контролю процесів заряджання та розряджання, оскільки для таких елементів важливими є захист від перезаряду, глибокого розряду, перевантаження за струмом і короткого замикання [13]. Модуль заряджання TP4056 може застосовуватися для заряджання одного літій-іонного елемента за алгоритмом постійного струму та постійної напруги, що є типовим для малопотужних автономних пристроїв [14].

Для узгодження рівнів напруги між акумуляторним блоком, сонячною панеллю, мікроконтролерами та виконавчими модулями доцільно використовувати DC-DC перетворювачі. Такі перетворювачі дають змогу отримати стабілізовані напруги 5 В або 3,3 В, необхідні для живлення Arduino Uno, ESP32, сенсорних модулів і релейного блока. Наприклад, перетворювачі на базі XL6009 можуть працювати як підвищувальні або інвертуючі DC-DC модулі,

що дає змогу використовувати їх у портативних електронних пристроях і системах автономного живлення [15].

Для вимірювання параметрів електроспоживання в системі може бути використаний вимірювальний модуль PZEM-004T або аналогічний модуль контролю електричних параметрів. PZEM-004T призначений для вимірювання змінної напруги, струму, активної потужності, частоти, коефіцієнта потужності та активної енергії, а передавання даних до мікроконтролера здійснюється через TTL-інтерфейс [16]. Такий модуль є доцільним для системи енергомоніторингу, оскільки дозволяє отримувати комплексні дані про стан навантаження, а не лише окреме значення струму або напруги.

Виконавча частина системи представлена багатоканальним релейним модулем, який забезпечує комутацію окремих електричних навантажень. Використання кількох каналів реле дозволяє окремо керувати освітленням, розетковими групами, допоміжними модулями або резервним навантаженням. У системах енергоменеджменту такий підхід є доцільним, оскільки дає змогу реалізувати сценарії автоматичного вимкнення другорядних споживачів у разі зниження рівня заряду акумулятора, перевищення допустимого навантаження або отримання відповідної команди від користувача [17].

У системі також передбачено сенсорні модулі, які можуть використовуватися для контролю параметрів середовища та стану енергосистеми. Зокрема, у схемі доцільно передбачити датчики напруги та струму для контролю електричних параметрів, датчик температури та вологості для моніторингу умов експлуатації, а також модулі контролю рівня освітленості або стану навантаження. Дані з таких сенсорів можуть використовуватися не лише для відображення користувачу, а й для прийняття керуючих рішень. Наприклад, при зниженні заряду акумуляторів система може автоматично вимкнути другорядні навантаження, а при перевищенні температури в зоні електронних модулів – сформулювати сповіщення або активувати захисний режим.

У таблиці 2.1 наведено функціональне призначення основних компонентів системи.

Таблиця 2.1 – Основні апаратні компоненти інтелектуальної системи керування енергоспоживанням

Компонент	Функціональне призначення	Обґрунтування використання
Arduino Uno	Локальне керування сенсорами, реле та виконавчими вузлами	Простота програмування, достатня кількість цифрових входів і виходів, сумісність із типовими модулями
ESP32	Передавання даних, Wi-Fi-зв'язок, інтеграція з IoT-платформою	Має вбудований Wi-Fi, підтримує IoT-протоколи та забезпечує віддалений моніторинг
Сонячна панель	Додаткове джерело енергії для автономного живлення	Дозволяє реалізувати енергоощадний режим і підзаряджання акумуляторів
Акумулятори 18650	Накопичення електричної енергії	Забезпечують резервне живлення низьковольтної частини системи
Модуль заряджання	Контроль заряджання акумуляторів	Дас змогу організувати безпечно поповнення заряду акумуляторного блока
DC-DC перетворювачі	Стабілізація та узгодження рівнів напруги	Забезпечують живлення мікроконтролерів і модулів на необхідних рівнях 5 В і 3,3 В
Інверторний модуль	Перетворення постійної напруги в змінну	Дозволяє реалізувати резервне живлення окремих навантажень
Релейний модуль	Комутація електричних навантажень	Забезпечує автоматизоване вмикання та вимикання споживачів
Сенсорні модулі	Вимірювання параметрів середовища та електроживлення	Формують дані для аналізу та прийняття керуючих рішень
Звуковий або світловий індикатор	Сигналізація про стан системи	Використовується для повідомлення про аварійні або робочі режими

Як видно з таблиці 2.1, апаратна структура системи має модульний характер. Це означає, що кожен функціональний блок може бути окремо замінений, модернізований або розширений без повної зміни архітектури. Наприклад, у подальшому можна замінити релейний модуль на твердотільні реле, додати точніший датчик струму, розширити акумуляторний блок або підключити кілька окремих ліній навантаження. Такий підхід відповідає принципам побудови IoT-систем, у яких важливими є масштабованість, гнучкість і можливість адаптації до конкретних умов експлуатації [18].

У роботі з апаратною частиною особливу увагу потрібно приділити електробезпеці. Оскільки схема містить низьковольтні мікроконтролерні вузли,

акумуляторний блок, інвертор і виконавчі модулі, необхідно забезпечити правильне розділення силових і логічних ланцюгів. Лінії живлення мікроконтролерів не повинні безпосередньо з'єднуватися з високовольтними частинами навантаження. Релейні контакти мають бути розраховані на відповідну напругу та струм, а акумулятори повинні мати захист від перезаряду, глибокого розряду та короткого замикання. Для підвищення надійності доцільно передбачити запобіжники, захисні діоди, стабілізатори напруги та правильне маркування ліній живлення.

Загальну логіку взаємодії компонентів системи можна описати таким чином. Сонячна панель формує електричну енергію, яка через модуль заряджання надходить до акумуляторного блока. Акумулятори живлять низьковольтну частину системи через DC-DC перетворювачі. Arduino Uno зчитує дані від сенсорів і керує релейним блоком. ESP32 забезпечує передавання інформації до користувача та може отримувати віддалені команди. Якщо система фіксує перевищення допустимого навантаження, нестачу заряду або інший визначений стан, Arduino Uno формує керуючий сигнал для реле, а ESP32 передає повідомлення до інтерфейсу користувача.

2.2 Розробка структурної схеми інтелектуальної системи керування енергоспоживанням

Розробка структурної схеми інтелектуальної системи керування енергоспоживанням є важливим етапом проєктування, оскільки саме на цьому рівні визначається склад основних функціональних блоків, характер їхньої взаємодії, напрям передавання даних, логіка керування навантаженнями та принцип організації живлення. Структурна схема дає змогу подати систему не як набір окремих електронних компонентів, а як цілісний апаратно-програмний комплекс, у якому кожен модуль виконує конкретну функцію в процесі енергомоніторингу та автоматизованого керування [19].

З урахуванням обраної апаратної конфігурації система керування енергоспоживанням будується на основі комбінованої мікроконтролерної архітектури. У цій архітектурі Arduino Uno виконує функції локального керуючого модуля, а ESP32 використовується як IoT-вузол для бездротового передавання даних, обміну командами та інтеграції з користувацьким інтерфейсом. Такий підхід є технічно доцільним, оскільки Arduino Uno має достатню кількість цифрових і аналогових входів-виходів для роботи з релейними та сенсорними модулями, а ESP32 підтримує Wi-Fi, Bluetooth і Bluetooth Low Energy, що дозволяє реалізувати мережеву взаємодію в IoT - середовищі [20, 21].

Структурну схему інтелектуальної системи керування енергоспоживанням подано на рисунку 2.2.

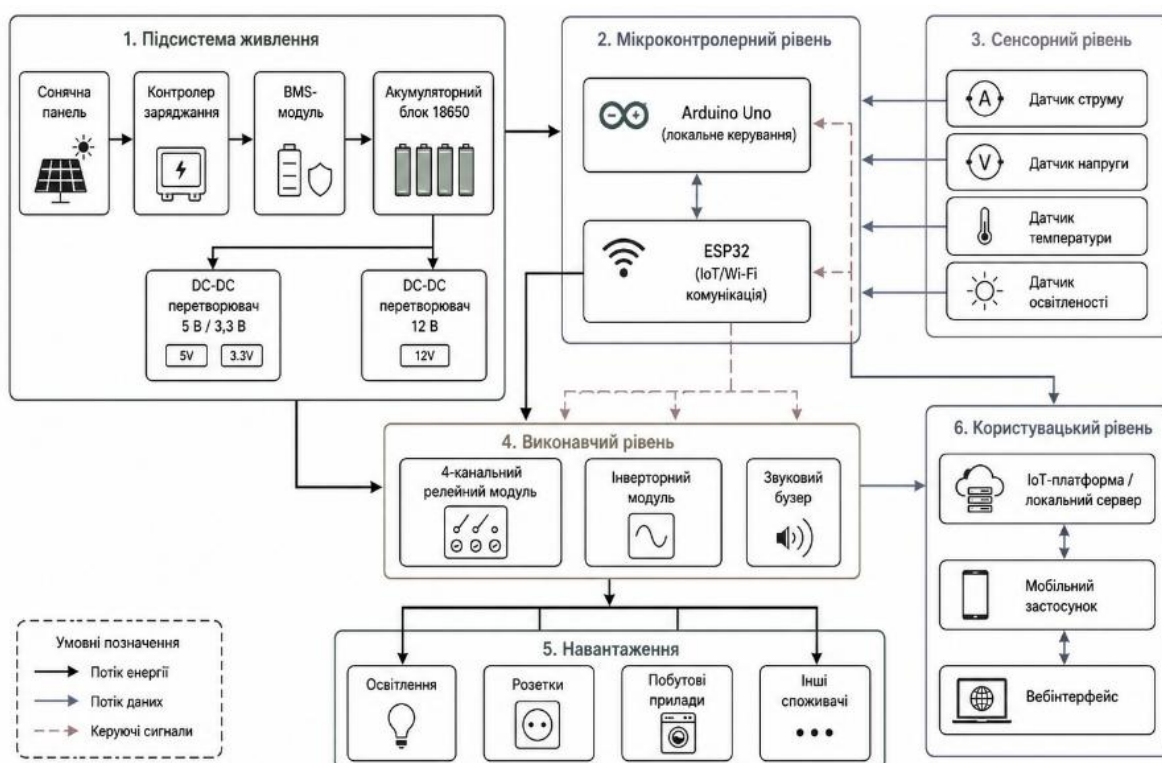


Рисунок 2.2 – Структурна схема інтелектуальної системи керування енергоспоживанням

Структурна схема системи повинна відображати три основні напрями взаємодії: потік електричної енергії, потік вимірювальних даних і потік

керуючих команд. Потік енергії формується підсистемою живлення, до якої входять сонячна панель, контролер заряджання, акумуляторний блок 18650, BMS-модуль, DC-DC перетворювачі та інверторний модуль. Потік даних формується сенсорними і вимірювальними модулями, які передають інформацію до мікроконтролерів. Потік керуючих команд формується Arduino Uno та ESP32 і спрямовується до релейного модуля, інверторного блока або інтерфейсу користувача [22].

На рисунку 2.2 відображено такі функціональні блоки: підсистему живлення, мікроконтролерний рівень, сенсорний рівень, виконавчий рівень, комунікаційний рівень та рівень користувацької взаємодії. Такий поділ дозволяє чітко показати, які елементи відповідають за енергозабезпечення, які – за вимірювання параметрів, які – за прийняття рішень, а які – за керування навантаженнями.

Підсистема живлення виконує функцію забезпечення енергією всіх електронних вузлів системи. Сонячна панель може використовуватися як додаткове джерело електричної енергії, що підвищує автономність пристрою. Енергія від сонячної панелі надходить до контролера заряджання, який забезпечує заряджання акумуляторного блока. Акумулятори типу 18650 виконують роль накопичувача енергії, а BMS-модуль забезпечує захист акумуляторів від критичних режимів роботи. Для живлення мікроконтролерів і периферійних модулів використовуються DC-DC перетворювачі, які формують стабілізовані напруги 5 В і 3,3 В [23].

Мікроконтролерний рівень є центральною частиною структурної схеми. Arduino Uno відповідає за локальну обробку сигналів, керування релейним модулем і виконання базових алгоритмів перемикавання навантажень. ESP32 забезпечує бездротове передавання даних і взаємодію з IoT-платформою або локальним сервером. Взаємодія між Arduino Uno та ESP32 може бути реалізована через послідовний інтерфейс UART або інші доступні цифрові інтерфейси. У такій конфігурації Arduino Uno виконує функції нижчого рівня

керування, а ESP32 – функції мережевого шлюзу та комунікаційного контролера [20, 21].

Сенсорний рівень призначений для збору інформації про стан системи та умови її роботи. До нього можуть входити датчик струму, датчик напруги, датчик температури, датчик освітленості та інші вимірювальні модулі. Датчики струму й напруги використовуються для контролю параметрів електричного навантаження та оцінювання поточного споживання. Датчик температури може застосовуватися для контролю нагрівання електронних компонентів, акумуляторного блока або силових модулів. Датчик освітленості може бути використаний для реалізації сценаріїв автоматичного керування освітленням або оцінювання умов роботи сонячної панелі [22, 24].

Виконавчий рівень системи реалізує безпосередній керуючий вплив на електричні навантаження. Основним елементом цього рівня є багатоканальний релейний модуль, який дає змогу окремо керувати кількома споживачами. Наприклад, окремі канали реле можуть використовуватися для керування освітленням, розетковою групою, допоміжним навантаженням або інверторним модулем. Команди на реле формуються мікроконтролером відповідно до заданих умов: рівня споживаної потужності, стану акумулятора, показників сенсорів або команд користувача [24].

Комунікаційний рівень забезпечує передавання інформації між апаратною частиною системи та програмним середовищем користувача. У розроблюваній системі цю функцію виконує ESP32, який може передавати дані через Wi-Fi за допомогою HTTP або MQTT. Для IoT-систем доцільним є використання MQTT, оскільки цей протокол працює за моделлю «публікація-підписка», має невеликі накладні витрати та призначений для обміну повідомленнями між пристроями з обмеженими ресурсами [25].

Рівень користувацької взаємодії може бути реалізований у вигляді мобільного застосунку, вебінтерфейсу або локальної панелі моніторингу. Його основне завдання полягає у відображенні поточних параметрів системи, стану навантажень, рівня заряду акумуляторів, режиму роботи інвертора та

повідомлень про аварійні або граничні стани. Крім того, користувацький інтерфейс може надавати можливість ручного керування окремими навантаженнями, зміни порогових значень і перегляду журналу подій [22, 24].

Отже, розроблена структурна схема інтелектуальної системи керування енергоспоживанням відображає взаємодію основних апаратних і програмних блоків: підсистеми живлення, сенсорів, мікроконтролерів, виконавчих модулів, комунікаційного вузла та інтерфейсу користувача. Центральними елементами системи є Arduino Uno та ESP32, які разом забезпечують локальне керування, обробку даних і передавання інформації в IoT-середовище. Запропонована структура є технічно обґрунтованою, масштабованою та придатною для подальшої реалізації інтелектуальної системи енергоменеджменту в умовах розумного будинку [22, 24].

2.3 Розробка принципової схеми підключення сенсорів, виконавчих пристроїв та комунікаційного модуля

Розробка принципової схеми підключення сенсорів, виконавчих пристроїв та комунікаційного модуля є важливим етапом формування апаратної частини інтелектуальної системи керування енергоспоживанням. На відміну від структурної схеми, яка показує загальну логіку взаємодії функціональних блоків, принципова схема деталізує конкретні електричні з'єднання між мікроконтролерами, сенсорними модулями, релейним блоком, підсистемою живлення та комунікаційним вузлом. У межах розроблюваної системи така схема повинна забезпечувати збір вимірювальних даних, обробку сигналів, керування навантаженнями та передавання інформації до IoT-середовища [26, 27].

Принципову схему підключення сенсорів, виконавчих пристроїв та комунікаційного модуля доцільно подати на рисунку 2.3.

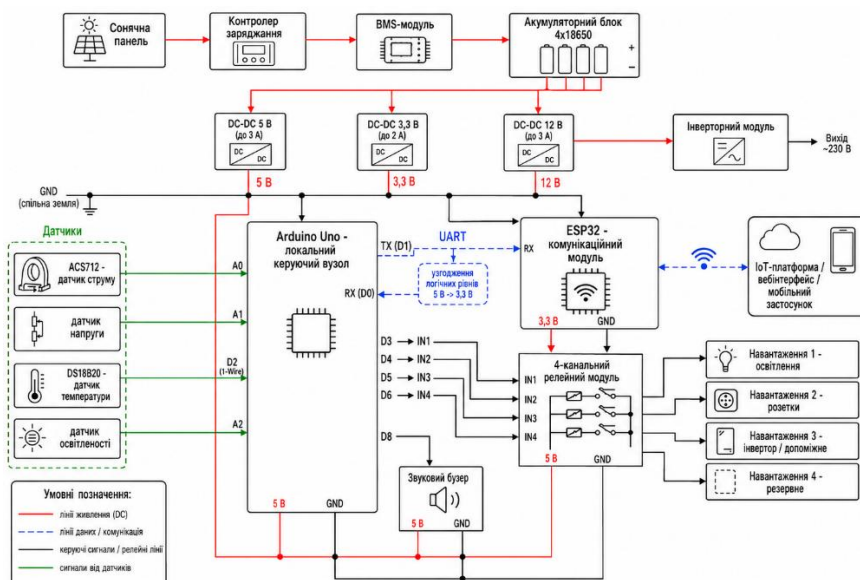


Рисунок 2.3 – Принципова схема підключення сенсорів, виконавчих пристроїв та комунікаційного модуля

Основними керуючими елементами системи є Arduino Uno та ESP32. Arduino Uno доцільно використовувати як локальний керуючий модуль, оскільки плата має цифрові та аналогові входи-виходи, достатні для підключення сенсорів, релейного модуля та сигналізаційних елементів [26]. ESP32 у схемі виконує функції комунікаційного модуля, оскільки підтримує Wi-Fi, Bluetooth і Bluetooth Low Energy, що дає змогу передавати дані до вебінтерфейсу, мобільного застосунку або локального сервера [27].

Сенсорна частина системи формується з датчика струму, датчика напруги, датчика температури та датчика освітленості. Для вимірювання струму може використовуватися модуль на основі ACS712, який працює за принципом ефекту Холла та призначений для вимірювання змінного або постійного струму з електричною ізоляцією вимірювального кола [28]. Датчик напруги підключається до аналогового входу мікроконтролера через вимірювальний модуль або подільник напруги, що дозволяє зменшити рівень сигналу до безпечного для Arduino Uno значення.

Для контролю температурного режиму апаратних компонентів доцільно застосувати цифровий датчик DS18B20. Його перевагою є використання інтерфейсу 1-Wire, який потребує лише одну сигнальну лінію для обміну даними

з мікроконтролером [29]. Такий датчик може бути розміщений біля акумуляторного блока, DC-DC перетворювачів або релейного модуля для контролю можливого перегріву. Датчик освітленості може застосовуватися для оцінювання умов роботи сонячної панелі або автоматизації сценаріїв керування освітленням.

Виконавча частина системи реалізується за допомогою 4-канального релейного модуля. Його керуючі входи підключаються до цифрових виходів Arduino Uno, а силові контакти використовуються для комутації окремих навантажень. Наприклад, перший канал може керувати освітленням, другий – розетковою групою, третій – інверторним модулем, а четвертий – додатковим або резервним споживачем. Такий підхід дозволяє реалізувати як ручне, так і автоматизоване керування навантаженнями відповідно до заданих умов роботи системи [26].

Підсистема живлення в принциповій схемі складається із сонячної панелі, модуля заряджання, акумуляторного блока 18650, BMS-модуля та DC-DC перетворювачів. Для заряджання літій-іонних акумуляторів може використовуватися модуль на основі TP4056, який реалізує заряджання за алгоритмом постійного струму та постійної напруги [30]. BMS-модуль необхідний для захисту акумуляторів від перезаряду, глибокого розряду, перевантаження за струмом і короткого замикання [31]. Для формування стабілізованих рівнів 5 В, 3,3 В або 12 В доцільно використовувати DC-DC перетворювачі, зокрема модулі на базі XL6009 для підвищення напруги [32].

Комунікаційний зв'язок між Arduino Uno та ESP32 може бути реалізований через послідовний інтерфейс UART. При цьому необхідно враховувати різницю логічних рівнів: Arduino Uno працює з логікою 5 В, а ESP32 - з логікою 3,3 В. Тому для передавання сигналу від Arduino Uno до ESP32 доцільно застосувати подільник напруги або перетворювач логічних рівнів. Обов'язковою умовою є наявність спільної шини GND між двома мікроконтролерами, оскільки без спільної опорної точки коректний обмін сигналами буде неможливим [26, 27].

Для передавання даних до користувацького інтерфейсу ESP32 може використовувати протоколи HTTP або MQTT. У системах Інтернету речей доцільним є MQTT, оскільки він працює за моделлю «публікація-підписка», має невеликі накладні витрати та призначений для обміну повідомленнями між пристроями з обмеженими ресурсами [33]. У такому випадку ESP32 може передавати значення струму, напруги, температури, рівня освітленості, стану реле та заряду акумуляторів до локального сервера або хмарної IoT-платформи.

Принципова схема повинна передбачати чітке розділення силових і логічних ланцюгів. Логічна частина включає Arduino Uno, ESP32, сенсори та сигнальні лінії керування, тоді як силова частина охоплює акумуляторний блок, інвертор, релейні контакти та електричні навантаження. Таке розділення необхідне для підвищення надійності, зменшення впливу електричних завад і захисту мікроконтролерів від пошкодження. Додатково доцільно передбачити стабілізоване живлення, захист акумуляторів, правильне маркування ліній живлення та використання релейних модулів із достатнім запасом за струмом.

Принципова схема підключення сенсорів, виконавчих пристроїв та комунікаційного модуля передбачає взаємодію Arduino Uno, ESP32, сенсорних модулів, релейного блока, акумуляторної підсистеми, DC-DC перетворювачів та інтерфейсу користувача. Arduino Uno забезпечує локальну обробку даних і керування реле, ESP32 відповідає за IoT-комунікацію, сенсори формують вимірювальну інформацію, а виконавчі модулі реалізують керування навантаженнями. Запропонована схема є технічно доцільною для побудови навчального прототипу інтелектуальної системи енергоменеджменту в розумному будинку [26-33].

РОЗДІЛ 3

ПРОГРАМНО-АПАРATНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ КЕРУВАННЯ ЕНЕРГОСПОЖИВАННЯМ

3.1 Реалізація апаратної частини системи моніторингу та керування енергоспоживанням

Реалізація апаратної частини інтелектуальної системи моніторингу та керування енергоспоживанням передбачає практичне збирання електронних модулів у єдиний апаратно-програмний комплекс, здатний виконувати вимірювання параметрів живлення, контролювати стан навантажень, керувати релейними каналами та передавати інформацію до користувацького інтерфейсу. На відміну від попереднього розділу, у якому було обґрунтовано вибір компонентів і розроблено структурну та принципову схеми, у цьому підрозділі основну увагу приділено безпосередній реалізації апаратної частини, компонуванню вузлів, організації живлення та перевірці працездатності зібраного прототипу.

Апаратна частина системи реалізована за модульним принципом. Це означає, що кожен функціональний вузол виконує окреме завдання: Arduino Uno забезпечує локальне керування, ESP32 відповідає за комунікацію, сенсорні модулі формують вимірювальні дані, релейний блок здійснює комутацію навантажень, а підсистема живлення забезпечує стабілізовану напругу для електронних компонентів. Такий підхід спрощує налагодження системи, оскільки кожен модуль можна перевірити окремо перед підключенням до загальної схеми.

Основою керуючої частини є плата Arduino Uno, до якої підключено сенсорні модулі та 4-канальний релейний блок. Arduino Uno приймає вхідні сигнали від датчика струму, датчика напруги, температурного датчика та датчика освітленості, після чого виконує первинну обробку отриманих значень. На основі цих даних контролер може визначати поточний стан системи, виявляти

перевищення допустимих параметрів і формувати сигнали керування для релейного модуля.

Загальний вигляд апаратної реалізації системи доцільно подати на рисунку 3.1.

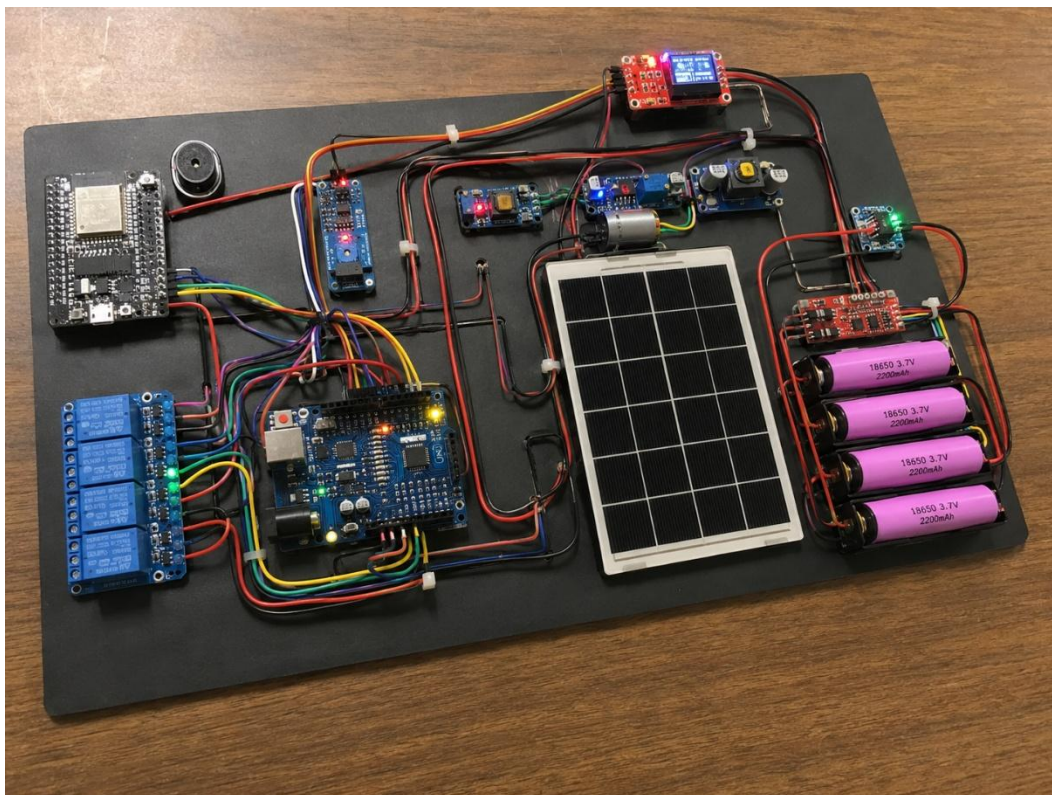


Рисунок 3.1 – Загальний вигляд апаратної частини системи моніторингу та керування енергоспоживанням

Комунікаційний модуль ESP32 використано для забезпечення бездротової взаємодії системи з користувачем. Його підключення до Arduino Uno реалізується через послідовний інтерфейс UART. Така організація дає змогу передавати від локального контролера до ESP32 інформацію про стан навантажень, значення сенсорів і аварійні події. У зворотному напрямі ESP32 може передавати команди користувача, наприклад на вмикання або вимикання окремого релейного каналу. Оскільки Arduino Uno працює з логічним рівнем 5 В, а ESP32 – з рівнем 3,3 В, під час практичного підключення необхідно

передбачити узгодження логічних рівнів, щоб уникнути пошкодження входів ESP32.

Сенсорний блок системи складається з модулів, які забезпечують контроль електричних і середовищних параметрів. Датчик струму використовується для оцінювання навантаження, що підключене до системи. Датчик напруги дає змогу контролювати рівень напруги в окремій частині схеми або на виході джерела живлення. Датчик температури може бути розміщений поблизу акумуляторного блока, DC-DC перетворювачів або релейного модуля для контролю можливого нагрівання. Датчик освітленості застосовується для оцінювання умов роботи сонячної панелі або для подальшої реалізації автоматизованого керування освітленням.

Виконавча частина системи реалізована на основі 4-канального релейного модуля. Кожен канал реле може відповідати за окрему групу навантаження: освітлення, розеткову групу, інверторний модуль або резервний споживач. Керуючі входи релейного модуля підключаються до цифрових виходів Arduino Uno, а силові контакти використовуються для комутації відповідних навантажень. Така реалізація дає змогу системі працювати у двох режимах: автоматичному, коли рішення приймається відповідно до показників сенсорів, і ручному, коли команду подає користувач через інтерфейс.

Окремим вузлом апаратної частини є підсистема живлення. Вона включає сонячну панель, контролер заряджання, акумуляторний блок на основі елементів 18650, BMS-модуль, а також DC-DC перетворювачі. Сонячна панель використовується як додаткове джерело енергії, що дає змогу підзаряджати акумуляторний блок. BMS-модуль забезпечує захист акумуляторів від критичних режимів роботи, зокрема перезаряду, глибокого розряду та короткого замикання. DC-DC перетворювачі формують стабілізовані рівні напруги 5 В, 3,3 В і 12 В, необхідні для живлення Arduino Uno, ESP32, сенсорів, релейного блока та інверторного модуля.

Реалізація апаратної частини виконувалася поетапно. На першому етапі було підготовлено основні електронні компоненти та перевірено їхню

справність. На другому етапі виконано підключення підсистеми живлення: акумуляторного блока, BMS-модуля, контролера заряджання та DC-DC перетворювачів. На третьому етапі було підключено Arduino Uno, ESP32 і сенсорні модулі. На четвертому етапі реалізовано підключення релейного блока та звукового бузера. На завершальному етапі виконано перевірку логічних сигналів, стабільності живлення та коректності взаємодії між Arduino Uno й ESP32.

Функціональне призначення основних вузлів апаратної реалізації наведено в таблиці 3.1.

Таблиця 3.1 – Основні вузли апаратної частини системи

Вузол системи	Використані компоненти	Практичне призначення
Керуючий вузол	Arduino Uno	Зчитування сигналів із сенсорів, обробка даних, керування релейним модулем
Комунікаційний вузол	ESP32	Передавання даних через Wi-Fi, приймання команд користувача
Сенсорний блок	Датчики струму, напруги, температури, освітленості	Формування вимірjuвальної інформації про стан системи
Виконавчий блок	4-канальний релейний модуль, бузер	Комутація навантажень і локальна сигналізація
Підсистема живлення	Сонячна панель, акумулятори 18650, BMS, DC-DC перетворювачі	Живлення електронних вузлів і забезпечення резервної роботи
Силовий вузол	Інверторний модуль, релейні контакти	Підключення окремих навантажень до джерела живлення

Під час апаратної реалізації важливо забезпечити розділення логічної та силової частин. Логічна частина охоплює Arduino Uno, ESP32, сенсори та сигнальні лінії. Силова частина включає акумуляторний блок, інверторний модуль, релейні контакти та електричні навантаження. Таке розділення зменшує ризик пошкодження мікроконтролерів, підвищує надійність роботи пристрою та дає змогу безпечніше виконувати налагодження. Усі низьковольтні модулі повинні мати спільну шину GND, оскільки без спільної опорної точки обмін сигналами між мікроконтролерами та сенсорами може бути некоректним.

Практична реалізація також передбачає правильне компонування елементів. Мікроконтролери та сенсори доцільно розміщувати окремо від

релейних контактів, DC-DC перетворювачів та інверторного модуля, оскільки силові вузли можуть створювати електромагнітні завади й нагріватися під час роботи. Провідники живлення повинні мати достатній переріз відповідно до струму навантаження, а сигнальні лінії бажано прокладати окремо від силових провідників. Для підвищення надійності прототипу доцільно використовувати клемні з'єднання, маркування ліній живлення та фіксацію модулів на монтажній основі.

Після складання апаратної частини виконується первинна перевірка. Спочатку перевіряється напруга на виходах DC-DC перетворювачів. Далі подається живлення на Arduino Uno та ESP32, після чого перевіряється стабільність їхньої роботи. Наступним кроком є перевірка сенсорів: зчитування значень струму, напруги, температури й освітленості. Після цього тестується релейний модуль шляхом почергового вмикання та вимикання кожного каналу. Окремо перевіряється передавання даних між Arduino Uno та ESP32, а також реакція системи на зміну стану навантаження.

Апаратна частина системи моніторингу та керування енергоспоживанням реалізована як модульний комплекс, що поєднує керуючий, сенсорний, виконавчий, комунікаційний і енергетичний рівні. Arduino Uno забезпечує локальне керування та обробку сигналів, ESP32 виконує функцію IoT-комунікації, сенсорні модулі формують дані про стан системи, а релейний блок реалізує керування навантаженнями. Запропонована апаратна реалізація створює основу для подальшої розробки програмного забезпечення, налаштування алгоритмів керування та проведення експериментального тестування системи.

3.2 Розробка програмного забезпечення для мікроконтролера

Розробка програмного забезпечення є одним із ключових етапів практичної реалізації інтелектуальної системи моніторингу та керування енергоспоживанням. Якщо апаратна частина забезпечує фізичне підключення сенсорів, релейних модулів, акумуляторного блока, інвертора та

комунікаційного вузла, то програмне забезпечення визначає логіку їхньої взаємодії. Саме програмний код забезпечує зчитування вимірювальних даних, обробку сигналів, прийняття рішень, керування реле, передавання інформації до користувацького інтерфейсу та реагування на аварійні або граничні режими роботи.

У розроблюваній системі програмна частина побудована з урахуванням комбінованої мікроконтролерної архітектури. Плата Arduino Uno використовується як локальний керуючий вузол, що безпосередньо взаємодіє із сенсорами та виконавчими пристроями. Модуль ESP32 виконує функції комунікаційного вузла, який забезпечує передавання даних через Wi-Fi до IoT-платформи, веб-інтерфейсу або мобільного застосунку. Такий підхід дозволяє розділити програмні функції між двома мікроконтролерами: Arduino Uno відповідає за апаратне керування, а ESP32 – за мережевий обмін даними.

Загальну структуру програмного забезпечення системи подано на рисунку 3.2.

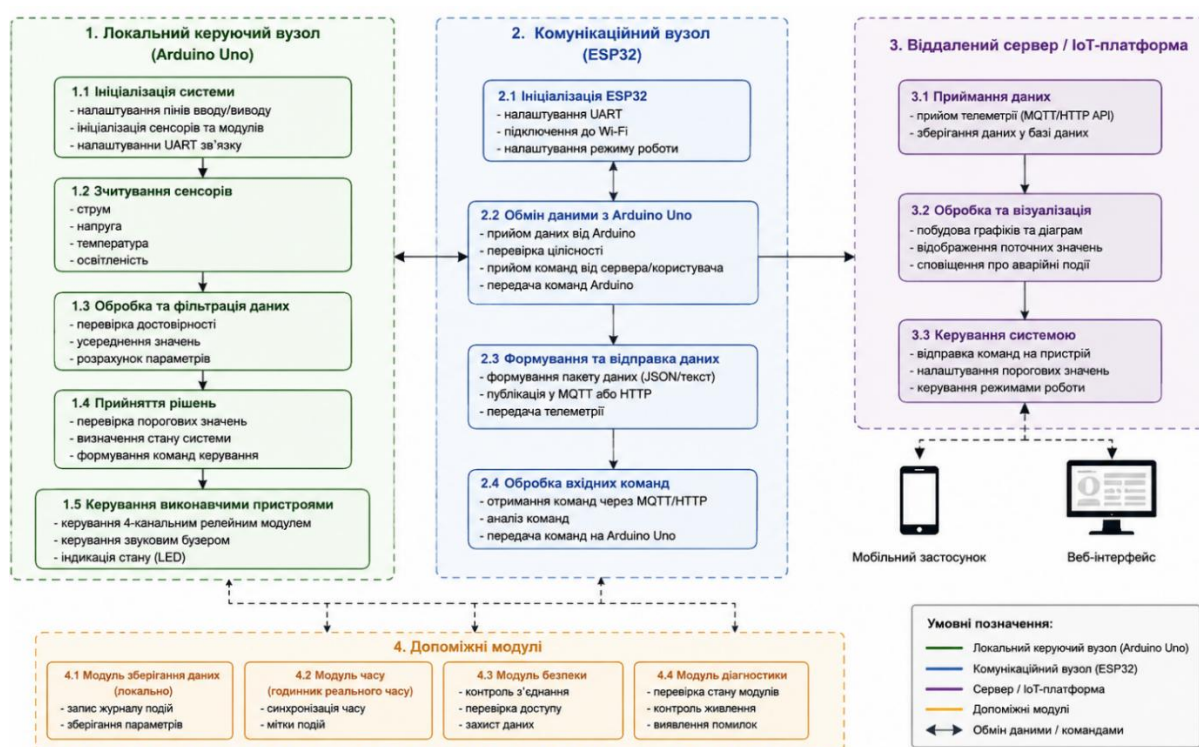


Рисунок 3.2 – Структура програмного забезпечення системи моніторингу та керування енергоспоживанням

На рисунку 3.2 доцільно показати такі програмні блоки: блок ініціалізації пристроїв, блок зчитування сенсорів, блок обробки даних, блок прийняття рішень, блок керування реле, блок обміну з ESP32, блок Wi-Fi-з'єднання, блок передавання даних до IoT-платформи та блок обробки команд користувача.

Першим етапом роботи програмного забезпечення є ініціалізація системи. На цьому етапі задаються режими роботи входів і виходів Arduino Uno, визначаються піни для підключення релейного модуля, бузера, сенсорів струму, напруги, температури та освітленості. Також встановлюються початкові значення змінних, порогові параметри спрацювання системи та швидкість обміну даними через послідовний інтерфейс UART. Для ESP32 на етапі ініціалізації виконується підключення до Wi-Fi-мережі, налаштування параметрів сервера або MQTT-брокера, а також перевірка готовності до обміну даними з Arduino Uno.

Після ініціалізації система переходить у основний цикл роботи. У цьому циклі Arduino Uno періодично зчитує значення із сенсорних модулів. Дані з датчика струму та датчика напруги використовуються для оцінювання параметрів електричного навантаження. Датчик температури дає змогу контролювати нагрівання електронних компонентів або акумуляторного блока, а датчик освітленості може застосовуватися для оцінювання умов роботи сонячної панелі або реалізації сценаріїв автоматизації освітлення. Отримані значення не повинні одразу використовуватися для прийняття рішень, оскільки можливі короточасні коливання сигналу або похибки вимірювання.

Для підвищення достовірності роботи системи в програмному забезпеченні доцільно передбачити первинну фільтрацію даних. Найпростішим методом є усереднення кількох послідовних вимірювань. Наприклад, Arduino Uno може виконувати декілька зчитувань із аналогового входу, після чого обчислювати середнє значення. Такий підхід дозволяє зменшити вплив випадкових відхилень і отримати більш стабільні показники для подальшого аналізу. Крім того, програмна логіка повинна ігнорувати некоректні значення, які явно виходять за межі допустимого діапазону.

Основним завданням програмного забезпечення є прийняття рішень щодо керування навантаженнями. Для цього використовуються порогові значення струму, напруги, температури, рівня заряду акумулятора або інших контрольованих параметрів. Якщо значення перебувають у допустимих межах, система продовжує моніторинг. Якщо ж виявлено перевищення допустимого струму, перегрів, зниження напруги живлення або інший критичний стан, Arduino Uno формує керуючий сигнал на відповідний канал релейного модуля. Одночасно може активуватися звуковий бусер, а ESP32 передає повідомлення користувачу.

Алгоритм роботи програмного забезпечення системи подано на рисунку 3.3.

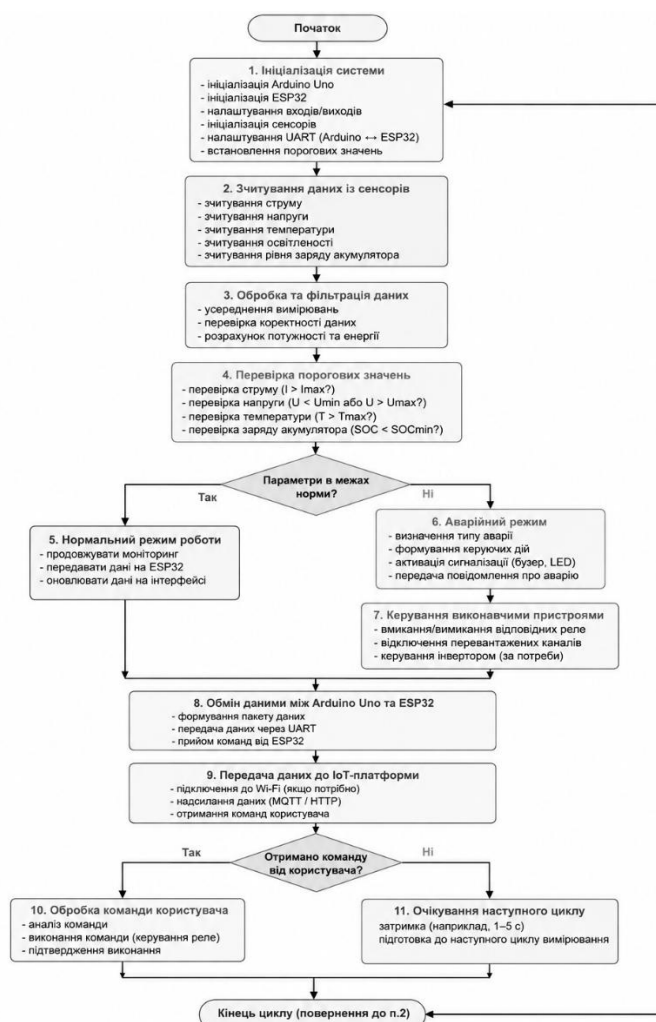


Рисунок 3.3 – Алгоритм роботи програмного забезпечення системи керування енергоспоживанням

На рисунку 3.3 доцільно відобразити таку послідовність: запуск системи, ініціалізація модулів, зчитування сенсорів, фільтрація та обробка даних, перевірка порогових значень, прийняття рішення, керування реле, передавання даних через ESP32, очікування наступного циклу вимірювання.

У програмній логіці Arduino Uno доцільно передбачити декілька режимів роботи. Перший режим – моніторинг, коли система лише збирає дані та передає їх до користувацького інтерфейсу. Другий режим – автоматичне керування, коли релейні канали вмикаються або вимикаються відповідно до заданих умов. Третій режим – ручне керування, коли користувач через інтерфейс надсилає команду на ESP32, а ESP32 передає її до Arduino Uno. Така організація роботи робить систему більш гнучкою та дозволяє використовувати її як для автоматизованого енергоменеджменту, так і для дистанційного керування окремими навантаженнями.

Програмна взаємодія Arduino Uno та ESP32 може бути реалізована через UART. У цьому випадку Arduino Uno формує текстовий або структурований пакет даних, який містить значення сенсорів і стан виконавчих пристроїв. Наприклад, пакет може містити значення струму, напруги, температури, освітленості, стан кожного релейного каналу та ознаку аварійного режиму. ESP32 приймає цей пакет, обробляє його та передає до IoT-платформи. У зворотному напрямі ESP32 може отримати команду користувача, наприклад «RELAY1_ON» або «RELAY2_OFF», і передати її на Arduino Uno для виконання.

Функціональне призначення основних програмних модулів наведено в таблиці 3.2.

Таблиця 3.2 – Основні програмні модулі системи моніторингу та керування енергоспоживанням

Програмний модуль	Призначення	Основні дії
Модуль ініціалізації	Підготовка системи до роботи	Налаштування пінів, змінних, UART, Wi-Fi
Модуль зчитування сенсорів	Отримання вимірювальних даних	Зчитування струму, напруги, температури, освітленості
Модуль фільтрації	Підвищення стабільності показників	Усереднення вимірювань, відкидання некоректних значень

Продовження таблиці 3.2

Програмний модуль	Призначення	Основні дії
Модуль аналізу	Перевірка стану системи	Порівняння показників із пороговими значеннями
Модуль керування реле	Комутація навантажень	Увімкнення або вимкнення релейних каналів
Модуль сигналізації	Повідомлення про аварійні режими	Активація бузера або індикатора
Модуль обміну Arduino-ESP32	Передавання даних між контролерами	Формування та приймання UART-пакетів
Модуль Wi-Fi-зв'язку	Підключення до мережі	З'єднання з роутером, сервером або IoT-платформою
Модуль інтерфейсу користувача	Віддалене керування та моніторинг	Передавання телеметрії, приймання команд

Як показано в таблиці 3.2, програмне забезпечення має модульну структуру. Це спрощує його розробку, тестування та подальше вдосконалення. Наприклад, модуль зчитування сенсорів можна перевірити окремо від модуля керування реле, а модуль Wi-Fi-зв'язку – незалежно від локальної логіки Arduino Uno. Такий підхід зменшує ймовірність помилок і дає змогу поступово розширювати функціональність системи.

Особливу увагу під час розробки програмного забезпечення потрібно приділити обробці аварійних ситуацій. До таких ситуацій можуть належати перевищення допустимого струму, перегрів силових модулів, критичне зниження напруги акумуляторного блока, відсутність зв'язку між Arduino Uno та ESP32 або недоступність Wi-Fi-мережі. У таких випадках система повинна переходити у безпечний режим. Наприклад, вона може вимкнути частину навантажень, активувати звукову сигналізацію, записати подію до журналу та передати повідомлення користувачу після відновлення зв'язку.

Для ESP32 програмне забезпечення повинно забезпечувати стабільне підключення до Wi-Fi-мережі. Якщо з'єднання втрачено, модуль має виконувати повторну спробу підключення без повного зупинення роботи системи. При цьому локальна логіка Arduino Uno повинна продовжувати працювати незалежно від наявності інтернет-з'єднання. Це важливо, оскільки система керування енергоспоживанням не повинна втрачати здатність вимикати небезпечне навантаження лише через тимчасову відсутність мережі.

Програмне забезпечення системи моніторингу та керування енергоспоживанням реалізує повний цикл роботи пристрою: ініціалізацію модулів, зчитування сенсорів, фільтрацію даних, аналіз параметрів, керування релейними каналами, сигналізацію аварійних станів і передавання інформації через ESP32. Розподіл функцій між Arduino Uno та ESP32 дозволяє поєднати локальну надійність керування з можливостями IoT-моніторингу. Така програмна структура є технічно доцільною для навчального прототипу інтелектуальної системи енергоменеджменту в розумному будинку.

3.3 Реалізація модуля передавання даних до IoT-платформи

Реалізація модуля передавання даних є важливим етапом розробки інтелектуальної системи моніторингу та керування енергоспоживанням, оскільки саме цей модуль забезпечує зв'язок між апаратною частиною пристрою та користувацьким середовищем. Якщо Arduino Uno виконує локальне зчитування сенсорів і керування виконавчими пристроями, то ESP32 забезпечує передавання даних до зовнішньої інформаційної системи через Wi-Fi. Завдяки цьому користувач може дистанційно контролювати стан системи, переглядати показники енергоспоживання, отримувати повідомлення про аварійні режими та надсилати команди для керування навантаженнями.

У розроблюваній системі комунікаційний модуль побудовано на основі ESP32. Такий вибір є технічно обґрунтованим, оскільки ESP32 має вбудований Wi-Fi-модуль, достатню продуктивність для обробки інформаційних пакетів і підтримує роботу з поширеними мережевими протоколами. У межах системи енергоменеджменту ESP32 виконує функцію проміжного вузла між локальним контролером Arduino Uno та IoT-платформою або хмарним сервісом. Обмін даними між Arduino Uno та ESP32 може здійснюватися через UART-інтерфейс, а передавання інформації до сервера – через HTTP або MQTT.

Загальну структуру модуля передавання даних до IoT-платформи подано на рисунку 3.4.

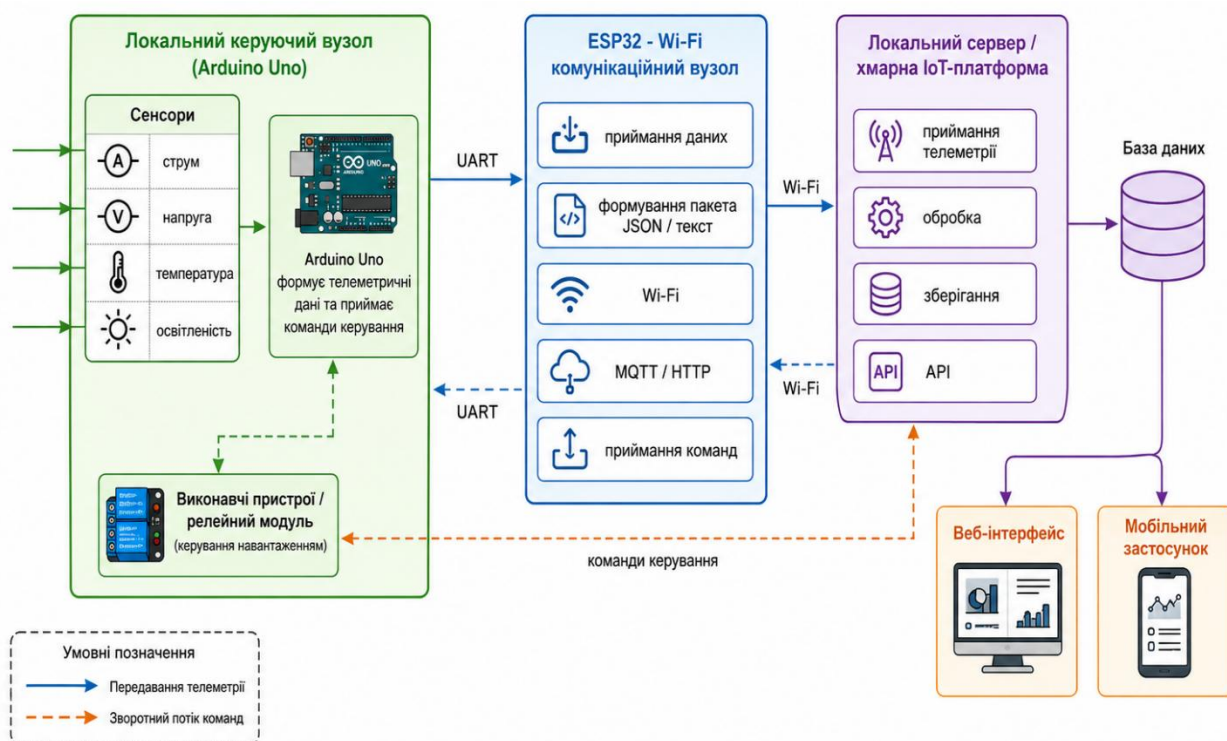


Рисунок 3.4 - Структура модуля передавання даних до IoT-платформи

На рисунку 3.4 доцільно показати Arduino Uno, який формує дані від сенсорів, ESP32 як Wi-Fi-комунікаційний вузол, локальний сервер або хмарну IoT-платформу, базу даних, вебінтерфейс і мобільний застосунок користувача. Напрямок передавання даних повинен відображати рух телеметрії від пристрою до сервера, а також зворотний потік команд від користувача до виконавчих пристроїв.

Першим етапом реалізації модуля передавання даних є формування інформаційного пакета на стороні Arduino Uno. До такого пакета доцільно включати основні параметри, які характеризують стан системи: значення струму, напруги, температури, рівня освітленості, стан релейних каналів, стан аварійного режиму та рівень заряду акумуляторного блока. Такий набір даних дозволяє користувачу отримати повну інформацію про роботу системи та оцінити поточний режим енергоспоживання.

Після формування пакета Arduino Uno передає його до ESP32 через послідовний інтерфейс UART. Для цього на Arduino Uno використовується апаратний або програмний Serial-порт, а на ESP32 - відповідний UART-вхід. Під

час реалізації такого обміну важливо забезпечити однакову швидкість передавання даних на обох пристроях, наприклад 9600 або 115200 бод. Також потрібно враховувати різницю логічних рівнів між Arduino Uno та ESP32: Arduino Uno працює з рівнем 5 В, а ESP32 - з рівнем 3,3 В. Тому для безпечного передавання сигналу від Arduino Uno до ESP32 доцільно використовувати дільник напруги або перетворювач логічних рівнів.

На стороні ESP32 програмне забезпечення виконує приймання пакета, перевірку його цілісності, обробку отриманих параметрів і передавання даних до IoT-платформи. Перед початком обміну ESP32 підключається до Wi-Fi-мережі. Якщо підключення успішне, модуль переходить у режим передавання телеметрії. Якщо з'єднання втрачено, ESP32 має виконувати повторне підключення, не зупиняючи локальну роботу Arduino Uno. Це важливо, оскільки система повинна зберігати здатність керувати навантаженнями навіть у випадку тимчасової відсутності інтернет-з'єднання.

Передавання даних може бути реалізоване двома основними способами: через HTTP-запити або за допомогою MQTT-протоколу. У разі використання HTTP ESP32 формує запит до сервера, у якому передає поточні параметри системи. Такий спосіб є простим для реалізації, однак потребує регулярного надсилання запитів і може створювати більше навантаження на мережу. MQTT є більш доцільним для IoT-систем, оскільки працює за моделлю «публікація-підписка». У цьому випадку ESP32 публікує телеметрію в певну тему, наприклад `home/energy/data`, а сервер або панель моніторингу отримує ці дані автоматично.

Важливою функцією модуля передавання даних є приймання команд від користувача. Наприклад, через вебінтерфейс або мобільний застосунок користувач може надіслати команду на вмикання чи вимикання певного релейного каналу. ESP32 приймає цю команду з IoT-платформи, аналізує її та передає до Arduino Uno. Після цього Arduino Uno змінює стан відповідного реле та повертає оновлений статус системи. Такий принцип роботи забезпечує двосторонню взаємодію між користувачем і апаратною частиною системи.

Основні типи даних, які передаються між пристроями та IoT-платформою, наведено в таблиці 3.3.

Таблиця 3.3 – Типи даних, що передаються в системі моніторингу та керування енергоспоживанням

Програмний модуль	Призначення	Основні дії
Модуль ініціалізації	Підготовка системи до роботи	Налаштування пінів, змінних, UART, Wi-Fi
Модуль зчитування сенсорів	Отримання вимірювальних даних	Зчитування струму, напруги, температури, освітленості
Модуль фільтрації	Підвищення стабільності показників	Усереднення вимірювань, відкидання некоректних значень
Модуль аналізу	Перевірка стану системи	Порівняння показників із пороговими значеннями
Модуль керування реле	Комутація навантажень	Увімкнення або вимкнення релейних каналів
Модуль сигналізації	Повідомлення про аварійні режими	Активація бузера або індикатора
Модуль обміну Arduino-ESP32	Передавання даних між контролерами	Формування та приймання UART-пакетів
Модуль Wi-Fi-зв'язку	Підключення до мережі	З'єднання з роутером, сервером або IoT-платформою
Модуль інтерфейсу користувача	Віддалене керування та моніторинг	Передавання телеметрії, приймання команд

Для підвищення надійності роботи модуля передавання даних доцільно передбачити контроль помилок. ESP32 має перевіряти, чи отримано повний пакет від Arduino Uno, чи є підключення до Wi-Fi, чи успішно надіслано дані до сервера. Якщо один із цих етапів не виконано, система може повторити спробу передавання або зберегти подію в локальному журналі. У разі втрати зв'язку з IoT-платформою локальне керування через Arduino Uno повинно залишатися активним, щоб забезпечити безпечну роботу системи.

Реалізація модуля передавання даних до IoT-платформи забезпечує інтеграцію апаратної частини системи з користувацьким середовищем. ESP32 виконує роль комунікаційного вузла, який приймає дані від Arduino Uno, формує інформаційні пакети та передає їх через Wi-Fi до сервера або хмарної платформи. У зворотному напрямі модуль приймає команди користувача та передає їх до локального контролера для керування релейними каналами. Така реалізація дозволяє створити повноцінну систему моніторингу та керування

енергоспоживанням із підтримкою дистанційного доступу, візуалізації даних і автоматизованої взаємодії з навантаженнями.

3.4 Розробка логіки автоматичного керування навантаженням на основі отриманих даних

Розробка логіки автоматичного керування навантаженням є важливим етапом реалізації інтелектуальної системи керування енергоспоживанням, оскільки саме на цьому рівні система переходить від простого моніторингу параметрів до активного впливу на роботу електричних споживачів. Якщо сенсорні модулі забезпечують отримання даних про струм, напругу, температуру, освітленість і стан живлення, то програмна логіка аналізує ці значення та формує рішення щодо вмикання, вимикання або обмеження роботи окремих навантажень.

У розроблюваній системі автоматичне керування навантаженням реалізується на основі порогового аналізу вимірних параметрів. Такий підхід є доцільним для бакалаврського проєкту, оскільки він не потребує складних обчислювальних ресурсів, може бути реалізований на мікроконтролері Arduino Uno та забезпечує зрозумілу логіку реагування на зміну стану системи. Основними параметрами, які враховуються під час прийняття рішення, є струм навантаження, напруга живлення, розрахована потужність, температура силових елементів, рівень заряду акумуляторного блока та стан команд, отриманих від користувача через ESP32.

У загальному вигляді логіка автоматичного керування передбачає, що мікроконтролер періодично зчитує дані із сенсорів, виконує їх первинну обробку, порівнює отримані значення із заданими порогами та визначає поточний режим роботи системи. Якщо параметри перебувають у допустимих межах, система підтримує нормальний режим роботи та продовжує моніторинг. Якщо один або кілька параметрів виходять за встановлені межі, система

переходить у режим захисту, формує керуючу команду для релейного модуля та передає повідомлення користувачу через комунікаційний модуль ESP32.

Основними режимами роботи системи є:

- 1) Режим моніторингу – система зчитує параметри та передає їх до користувацького інтерфейсу без зміни стану навантажень.
- 2) Автоматичний режим – система самостійно керує релейними каналами відповідно до заданих порогів і сценаріїв.
- 3) Ручний режим – користувач надсилає команду через вебінтерфейс або мобільний застосунок, а система виконує її за умови відсутності аварійних обмежень.
- 4) Захисний режим – система примусово вимикає окремі навантаження у разі перевищення допустимого струму, перегріву, критичного зниження напруги або низького заряду акумулятора.

Загальну логіку автоматичного керування навантаженням подано на рисунку 3.5.

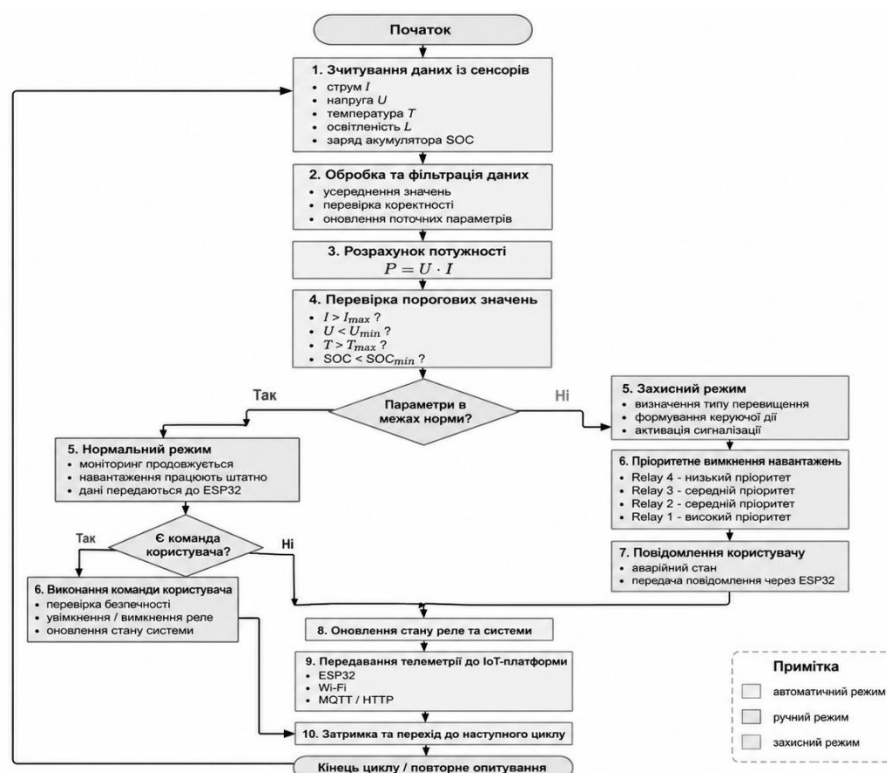


Рисунок 3.5 – Логіка автоматичного керування навантаженням на основі отриманих даних

У межах розроблюваної системи доцільно передбачити такі умовні пороги:

Таблиця 3.4 – Приклади порогових значень для автоматичного керування навантаженням

Контрольований параметр	Умова спрацювання	Дія системи
Струм навантаження	$I > I_{\max}$	Вимкнення перевантаженого релейного каналу
Напруга живлення	$U < U_{\min}$	Вимкнення другорядних навантажень
Температура модуля	$T > T_{\max}$	Активація бузера та вимкнення навантаження
Рівень заряду акумулятора	$SOC < SOC_{\min}$	Перехід в енергоощадний режим
Відсутність зв'язку з ESP32	Немає відповіді протягом заданого часу	Продовження локального керування через Arduino Uno
Команда користувача	Отримано команду ON/OFF	Зміна стану реле за умови безпечного режиму

Як показано в таблиці 3.4, логіка автоматичного керування повинна враховувати не лише окреме значення струму або напруги, а загальний стан системи. Наприклад, якщо користувач надсилає команду на вмикання навантаження, але рівень заряду акумуляторного блока є критично низьким, система не повинна виконувати цю команду без перевірки безпечності. У такому випадку команда може бути відхилена, а користувачу надсилається повідомлення про причину блокування.

Особливу увагу в логіці керування необхідно приділити пріоритетності навантажень. У розумному будинку не всі споживачі мають однакову важливість. Наприклад, освітлення або системи безпеки можуть мати вищий пріоритет, ніж допоміжні розетки або другорядні побутові пристрої. Тому в програмній логіці доцільно передбачити поділ навантажень на пріоритетні та неприоритетні. У разі нестачі енергії або перевищення потужності система спочатку вимикає неприоритетні навантаження, а критично важливі залишає активними якомога довше.

Приклад розподілу навантажень за пріоритетністю подано в таблиці 3.5.

Таблиця 3.5 – Пріоритетність керування електричними навантаженнями

Релейний канал	Тип навантаження	Пріоритет	Логіка керування
Relay 1	Освітлення	Високий	Вимикається лише у критичному режимі
Relay 2	Розеткова група	Середній	Вимикається при перевищенні потужності
Relay 3	Інверторний або допоміжний модуль	Середній	Вимикається при зниженні напруги акумулятора
Relay 4	Резервне навантаження	Низький	Вимикається першим при дефіциті енергії

Запропонована пріоритетність дозволяє системі працювати більш раціонально. Наприклад, якщо рівень заряду акумуляторів знижується нижче встановленого порогу, система може спочатку вимкнути четвертий канал, потім третій, а освітлення залишити активним. Такий підхід підвищує практичну цінність системи, оскільки вона не просто вимикає всі навантаження одночасно, а виконує керування відповідно до важливості кожного споживача.

Для реалізації такої логіки в програмному коді Arduino Uno доцільно використовувати умовні оператори if / else, змінні стану реле та окремі функції для зчитування сенсорів, аналізу параметрів і керування виконавчими пристроями. Наприклад, функція readSensors() може відповідати за зчитування даних, функція checkLimits() – за перевірку порогових значень, а функція controlRelays() – за зміну стану релейного модуля. Такий поділ програмного коду на функціональні блоки спрощує налагодження та подальше розширення системи.

Важливою особливістю автоматичного керування є те, що система повинна мати захист від частого перемикавання реле. Якщо значення параметра коливається біля порогового рівня, реле може багаторазово вмикатися та вимикатися, що негативно впливає на його ресурс і стабільність роботи навантаження. Для уникнення цього доцільно використати гістерезис. Наприклад, якщо навантаження вимикається при потужності понад 100 Вт, то повторне вмикання може дозволятися лише після зниження потужності до 80 Вт. Це забезпечує стабільнішу роботу системи.

Також у системі доцільно передбачити часову затримку між перемиканнями релейних каналів. Наприклад, після вимкнення навантаження система може очікувати 3-5 секунд перед наступною дією. Це дозволяє уникнути різких змін у роботі силових вузлів і зменшує ймовірність помилкових спрацювань.

Логіка автоматичного керування навантаженням у розроблюваній системі базується на аналізі даних, отриманих від сенсорів, порівнянні цих даних із заданими пороговими значеннями та формуванні керуючих команд для релейного модуля. Такий підхід дозволяє реалізувати базові функції інтелектуального енергоменеджменту: автоматичне вимкнення перевантажених каналів, перехід в енергоощадний режим, захист від перегріву, пріоритетне керування навантаженнями та дистанційне інформування користувача. Запропонована логіка є технічно доцільною для навчального прототипу та створює основу для подальшого тестування ефективності роботи системи.

ВИСНОВКИ

У кваліфікаційній роботі розглянуто та практично обґрунтовано розробку інтелектуальної системи керування енергоспоживанням у розумному будинку на основі технологій Інтернету речей. У процесі виконання роботи встановлено, що сучасні системи енергомоніторингу повинні забезпечувати не лише фіксацію поточних параметрів споживання електроенергії, а й можливість їх аналізу, передавання користувачу та формування керуючих дій для оптимізації роботи електричних навантажень. Це підтверджує актуальність створення апаратно-програмних рішень, які поєднують сенсорні модулі, мікроконтролери, виконавчі пристрої та засоби бездротового зв'язку.

У роботі проаналізовано сучасні підходи до побудови систем енергомоніторингу та керування енергоспоживанням у розумному будинку. Визначено, що найбільш доцільним для навчального та побутового прототипу є підхід, за якого система виконує вимірювання основних параметрів електроживлення, аналізує їх за встановленими правилами та автоматично реагує на зміну стану навантаження. Такий підхід дає змогу реалізувати базові функції інтелектуального енергоменеджменту без використання складних обчислювальних платформ.

Розглянуто архітектуру IoT-системи як середовища взаємодії сенсорів, мікроконтролерів, виконавчих пристроїв і програмних сервісів. Встановлено, що ефективна система керування енергоспоживанням повинна мати багаторівневу структуру, яка включає сенсорний рівень, мікроконтролерний рівень, комунікаційний рівень, рівень зберігання та обробки даних, а також рівень користувацької взаємодії. Така архітектура забезпечує можливість збору інформації з фізичного середовища, її обробки, передавання до IoT-платформи та зворотного керування навантаженнями.

Обґрунтовано вибір апаратних компонентів для реалізації системи керування енергоспоживанням. Як основні керуючі вузли використано Arduino Uno та ESP32. Arduino Uno доцільно застосовувати для локального зчитування

сенсорів і керування виконавчими пристроями, а ESP32 - для реалізації Wi-Fi-зв'язку та передавання даних до користувацького інтерфейсу. До складу апаратної частини також включено сенсори струму, напруги, температури та освітленості, 4-канальний релейний модуль, звуковий бузер, сонячну панель, акумуляторний блок 18650, BMS-модуль, контролер заряджання, DC-DC перетворювачі та інверторний модуль.

У роботі розроблено структурну та принципову схеми підключення основних компонентів системи. Структурна схема дала змогу відобразити загальну взаємодію підсистеми живлення, мікроконтролерного рівня, сенсорного рівня, виконавчого рівня та користувацького інтерфейсу. Принципова схема деталізувала підключення сенсорів, релейного модуля, ESP32, Arduino Uno, акумуляторного блока, BMS-модуля та DC-DC перетворювачів. Особливу увагу приділено розділенню логічної та силової частин, узгодженню рівнів напруги між Arduino Uno та ESP32, а також забезпеченню стабільного живлення низьковольтних компонентів.

Сформовано алгоритм збору, обробки та передавання даних про енергоспоживання. Алгоритм передбачає ініціалізацію апаратних модулів, зчитування даних із сенсорів, фільтрацію вимірювань, розрахунок потужності, перевірку порогових значень, формування інформаційного пакета та передавання даних через ESP32 до IoT-платформи або локального сервера. Такий алгоритм дозволяє забезпечити безперервний моніторинг стану системи та своєчасне інформування користувача про зміну параметрів електроспоживання.

Реалізовано програмну логіку керування електричним навантаженням. У програмній частині передбачено режими моніторингу, автоматичного керування, ручного керування та захисного реагування. Система аналізує значення струму, напруги, температури, освітленості та стану акумулятора, після чого приймає рішення щодо вмикання або вимикання окремих релейних каналів. У разі перевищення допустимих параметрів система може вимкнутися.

навантаження, активувати звукову сигналізацію та передати повідомлення користувачу через комунікаційний модуль.

Проведене тестування працездатності апаратно-програмного прототипу підтвердило коректність взаємодії основних модулів системи. Було перевірено стабільність живлення мікроконтролерів, роботу сенсорних модулів, перемикання релейних каналів, функціонування звукового бузера, передавання даних між Arduino Uno та ESP32, а також можливість відображення інформації в користувацькому середовищі. Отримані результати свідчать про те, що запропонована система може використовуватися як основа для побудови навчального або побутового прототипу інтелектуального енергомоніторингу.

Практичне значення виконаної роботи полягає у створенні технічно обґрунтованої моделі IoT-системи, яка поєднує функції моніторингу, аналізу та керування енергоспоживанням. Розроблене рішення може бути використане для контролю стану окремих електричних навантажень, автоматичного вимкнення споживачів у разі перевищення допустимих параметрів, дистанційного перегляду даних і подальшого розширення функціональності. У перспективі систему можна доповнити точнішими вимірювальними модулями, базою даних історії споживання, мобільним застосунком, алгоритмами прогнозування та засобами адаптивного керування енергоспоживанням.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Optimum Energy Management for Air Conditioners in IoT-Enabled Smart Home. *Sensors*. URL: <https://www.mdpi.com/1424-8220/22/19/7102> (Date of access: 15.01.2026).
2. Design and Implementation of a Cloud-IoT-Based Home Energy Management System. *Sensors*. URL: <https://www.mdpi.com/1424-8220/23/1/176> (Date of access: 18.01.2026).
3. General Overview and Proof of Concept of a Smart Home Energy Management System Architecture. *Electronics*. URL: <https://www.mdpi.com/2079-9292/12/21/4453> (Date of access: 21.01.2026).
4. IoT - A Promising Solution to Energy Management in Smart Buildings. *Buildings*. URL: <https://www.mdpi.com/2075-5309/14/11/3446> (Date of access: 24.01.2026).
5. Internet-of-Things-Based Smart Home Energy Management System. *Engineering Proceedings*. URL: <https://www.mdpi.com/2673-4591/66/1/10> (Date of access: 27.01.2026).
6. Solar-Powered Smart Buildings: Integrated Energy Management Solution for IoT-Enabled Sustainability. *Electronics*. URL: <https://www.mdpi.com/2079-9292/13/2/317> (Date of access: 30.01.2026).
7. A Smart Energy Monitoring System Using ESP32 Microcontroller. *e-Prime - Advances in Electrical Engineering, Electronics and Energy*. URL: <https://www.sciencedirect.com/science/article/pii/S2772671124002468> (Date of access: 02.02.2026).
8. Optimal Control of Cooling Management System for Energy Conservation in Smart Home with ANNs-PSO Data Analytics Microservice Platform. *Heliyon*. URL: <https://www.sciencedirect.com/science/article/pii/S2405844024029682> (Date of access: 05.02.2026).
9. Industrial IoT-Based Energy Monitoring System. *Applied System Innovation*. URL: <https://www.mdpi.com/2624-831X/5/4/27> (Date of access: 08.02.2026).

10. Smart Home Energy Management Systems: A Systematic Literature Review. *Journal of Smart Materials and Structures*. URL: <https://www.aasmr.org/jsms/onlinefirst/Vol14/No.11/Vol.14.No.11.08.pdf> (Date of access: 11.02.2026).
11. Internet of Things Applications for Energy Management in Buildings Using Artificial Intelligence – A Case Study. *Energies*. URL: <https://www.mdpi.com/1996-1073/18/7/1706> (Date of access: 14.02.2026).
12. IoT-Enabled Energy Storage Systems for Renewable Energy Applications. *Path of Science*. URL: <https://pathofscience.org/index.php/ps/article/view/3505> (Date of access: 17.02.2026).
13. IoT Integration for Renewable Energy Storage. *Engineering Proceedings*. URL: <https://www.mdpi.com/2673-4591/107/1/101> (Date of access: 20.02.2026).
14. Smart Homes by IoT-Driven Hybrid Renewable Energy Integration for Enhanced Efficiency. *Scientific Reports*. URL: <https://www.nature.com/articles/s41598-025-25328-2> (Date of access: 23.02.2026).
15. Future of Energy Management Models in Smart Homes: A Systematic Literature Review of Research Trends, Gaps and Future Directions. *Discover Internet of Things*. URL: <https://link.springer.com> (Date of access: 26.02.2026).
16. Optimizing Residential Energy Usage with Smart Devices. *Sustainability*. URL: <https://www.mdpi.com/2071-1050/17/14/6359> (Date of access: 01.03.2026).
17. Digital Twin Assisted Real-Time Energy Management System for Smart Homes. *Energy and Buildings*. URL: <https://www.sciencedirect.com/science/article/pii/S0378778826000381> (Date of access: 04.03.2026).
18. Artificial Intelligence, IoT, and Solar PV Integration for Home Energy Management Systems: A Review. *Results in Engineering*. URL: <https://www.sciencedirect.com/science/article/pii/S2590123025049965> (Date of access: 07.03.2026).
19. IoT-Based Smart Electricity Energy Meter Using ESP32 and Blynk Application for Energy Management in Somalia. *Zenodo*. URL: <https://zenodo.org/records/15450199> (Date of access: 10.03.2026).

20. IoT-Enabled Smart Home Energy Monitoring System Using Web Server-Based Control Logic. *Journal of Information and Digital Technology*. URL: https://www.researchgate.net/publication/394723148_IoT-Enabled_Smart_Home_Energy_Monitoring_System_Using_Web_Server-Based_Control_Logic (Date of access: 13.03.2026).

21. Smart Home Automation with Real-Time Energy Monitoring and Voice Control Using ESP32. *International Journal of Scientific Research in Engineering and Management*. URL: <https://ijsrem.com/uploads/articles/1778599247197-2c7b8571406f-Smart-Home-Automation-with-Real-Time-Energy-Monitoring-and-Voice-Control-using-ESP32.pdf> (Date of access: 16.03.2026).

22. Espressif Systems. ESP32-WROOM-32 Datasheet. *Version 3.6. Espressif Systems*. URL: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf (Date of access: 19.03.2026).

23. Espressif Systems. ESP32-S3-WROOM-1 and ESP32-S3-WROOM-1U Datasheet. *Espressif Systems*. URL: https://documentation.espressif.com/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf (Date of access: 22.03.2026).

24. Arduino. UNO R3 Documentation. *Arduino Docs*. URL: <https://docs.arduino.cc/hardware/uno-rev3> (Date of access: 25.03.2026).

25. Arduino. Getting Started with Arduino UNO R3. *Arduino Docs*. 2025. URL: <https://docs.arduino.cc/tutorials/uno-rev3/getting-started> (Date of access: 28.03.2026).

26. Allegro MicroSystems. ACS712 Fully Integrated, Hall Effect-Based Linear Current Sensor IC: Datasheet. *Allegro MicroSystems*. URL: <https://www.allegromicro.com/-/media/files/datasheets/acs712-datasheet.ashx> (Date of access: 31.03.2026).

27. Analog Devices. DS18B20 Programmable Resolution 1-Wire Digital Thermometer: Datasheet. *Analog Devices*. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/ds18b20.pdf> (Date of access: 03.04.2026).

28. NanJing Top Power ASIC Corp. TP4056 1A Standalone Linear Li-Ion Battery Charger with Thermal Regulation: Datasheet. *NanJing Top Power ASIC Corp.*

URL: <https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf> (Date of access: 06.04.2026).

29. DFRobot. 18650 Battery Protection Module: Product Overview. *DFRobot*. URL: <https://www.mouser.com/pdfDocs/Productoverview-DFRobot-FIT0869.pdf> (Date of access: 09.04.2026).

30. XLSEMI. XL6009 DC/DC Converter: Datasheet. *XLSEMI*. URL: <https://beriled.biz/data/files/XL6009.pdf> (Date of access: 12.04.2026).

31. ThingsBoard. Smart Energy Solution: IoT Energy Management and Monitoring. *ThingsBoard Documentation*. URL: <https://thingsboard.io/use-cases/smart-energy> (Date of access: 15.04.2026).

32. Blynk. Blynk Documentation. *Blynk IoT Platform*. URL: <https://docs.blynk.io> (Date of access: 18.04.2026).

33. OpenRemote. Open Source IoT Platform for Energy Management. *OpenRemote*. URL: <https://openremote.io/energy-management-open-source> (Date of access: 20.04.2026).

