

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр  
Освітній рівень


Система моніторингу енерговитрат у кіберфізичній системі «Розумний будинок»  
Назва теми

КВРКІ 220018.22.01.33 ПЗ  
Шифр


Галузь знань 12 «Інформаційні технології»  
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»  
Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»  
Назва

Виконав: студент III курсу, група KI2c-22-1  Роман КРАЧКОВСЬКИЙ  
Ініціали, прізвище

Керівник  Ольга ПАВЛОВА  
Ініціали, прізвище

Нормоконтролер  Тетяна КИСІЛЬ  
Ініціали, прізвище

До захисту допускаю:  
зав. кафедри комп'ютерної  
інженерії та інформаційних  
систем

  
Підпис

Ольга ПАВЛОВА  
Ініціали, прізвище

«05» червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2025 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

Роману КРАЧКОВСЬКОМУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Система моніторингу енерговитрат у кіберфізичній системі «Розумний будинок»

Керівник проекту (роботи) Ольга ПАВЛОВА, д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

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

Дослідження кіберфізичної системи «Розумний будинок» та постановка задачі

Проектування системи моніторингу енерговитрат в режимі реального часу у кіберфізичній системі «Розумний будинок» Програмно-апаратна реалізація та тестування системи моніторингу енерговитрат у кіберфізичній системі «Розумний будинок».

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Схеми апаратних з'єднань

Інтерфейси програмно-апаратного засобу

Блок-схеми програм

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КПС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КПС		

7. Дата видачі завдання « 10 » 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітки
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2025	виконано
4	Робота над розділом 2 – вибір компонентів для проектування системи адаптивного застосування моніторингових елементів розвідувального БПЛА	01.04.2025	виконано
5	Робота над розділом 3 – проектування системи адаптивного застосування моніторингових елементів розвідувального БПЛА	29.04.2025	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2025	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент

Підпис

Роман КРАЧКОВСЬКИЙ  
Ініціали, прізвище

Керівник роботи

Підпис

Ольга ПАВЛОВА  
Ініціали, прізвище



## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система моніторингу енерговитрат у кіберфізичній системі «Розумний будинок» на Arduino».

Автор роботи: Роман Крачковський.

Керівник роботи: Павлова Ольга Олександрівна.

Пояснювальна записка: 61 с., 24 рис., 5 табл., 40 джерел.

Графічна частина:

КІБЕРФІЗИЧНА СИСТЕМА, РОЗУМНИЙ БУДИНОК, ARDUINO, ЕЛЕКТРОЖИВЛЕННЯ, ЕНЕРГОСПОЖИВАННЯ, МІКРОСХЕМА, ІНТЕРНЕТ РЕЧЕЙ, СЕРВЕР, ACS712, ZMPT101B, NODE.

Об'єктом дослідження є система моніторингу енерговитрат у кіберфізичній системі «Розумний будинок». Предметом дослідження є методи збору, обробки та відображення даних про енергоспоживання з використанням апаратних та програмних засобів на базі платформи Arduino.

Метою роботи є розробка та дослідження системи моніторингу енерговитрат для кіберфізичної системи «Розумний будинок» на основі мікроконтролерів, яка дозволяє автоматизовано реєструвати, аналізувати та візуалізувати показники електроспоживання.

Практична цінність роботи полягає у можливості застосування розробленої системи для моніторингу енерговитрат у житлових і промислових приміщеннях. Це дозволяє отримувати інформацію про рівень енергоспоживання, що може бути використано для прийняття рішень щодо оптимізації використання електроенергії.

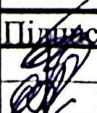
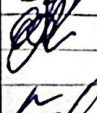

  
Підпис студента

30.05.2025

Дата

## ЗМІСТ

<b>СКРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ</b> .....	4
<b>ВСТУП</b> .....	5
<b>1 АНАЛІЗ СИСТЕМ МОНІТОРИНГУ ЕНЕРГОВИТРАТ У РОЗУМНОМУ БУДИНКУ</b> .....	7
1.1 Аналіз концепції «розумного будинку» та його енергоефективності ....	7
1.2 Огляд існуючих систем моніторингу енерговитрат .....	10
1.3 Аналіз апаратних та програмних платформ для автоматизованого контролю енергоспоживання.....	13
1.4 Порівняння підходів до збору, обробки та візуалізації енергоспоживанн. ....	20
1.5 Постановка завдання для розробки системи моніторингу енерговитрат	21
1.6 Висновки .....	22
<b>2 ПРОЄКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ЕНЕРГОВИТРАТ НА ОСНОВІ ARDUINO</b> .....	24
2.1 Вибір апаратного забезпечення для збору та передачі даних про енергоспоживання .....	24
2.2 Вибір програмного забезпечення для збору, обробки та відображення інформації .....	30
2.3. Вимоги до апаратного забезпечення системи.....	32
2.4 Вимоги до програмного забезпечення та архітектури системи.....	32
2.5 Архітектурна схема та алгоритм роботи системи моніторингу.....	34
2.6. Висновки .....	36
<b>3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ЕНЕРГОВИТРАТ</b> .....	37
3.1 Розробка апаратної частини системи та її інтеграція з Arduino.....	37

КВРКІ.220018.22.01.33 ПЗ								
Зм.	Арк.	№ док.ум.	Підпис	Дата	Підсистеми керування електроживленням та відслідковування енергоспоживання в режимі реального часу на платформі Arduino в кіберфізичній системі "Розумний будинок"	Літера	Арк.шіт	Арк.шітів
Виконав		Роман КРАЧКОВСЬКИЙ		03.05.25		у		
Перевір.		Ольга ПАВЛОВА		05.06.25		2		
Н.контр. Затвер.		Гетяна КИСІЛЬ Ольга ПАВЛОВА		18.06 06.06.25		ХНУ КІ2с-22-1		

3.2 Розробка програмного забезпечення для мікроконтролера та його алгоритмів збору даних .....	41
3.3 Розробка програмного забезпечення для веб-інтерфейсу моніторингу енергоспоживання.....	46
3.4 Взаємодія між апаратною та програмною частинами.....	49
3.5 Проведення тестування системи та аналіз отриманих результатів....	50
3.6. Висновки .....	53
<b>ВИСНОВКИ</b> .....	54
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ</b> .....	55
Додаток А.....	60
Додаток Б.....	61
Додаток В.....	62
Додаток Г.....	63
Додаток Д.....	66

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ЕМ – електромагнітний

MQTT – Message Queuing Telemetry Transport

IoT – Internet of Things

ПЛК – програмований логічний контролер

GSM – Global System for Mobile Communications

АЦП – аналого-цифровий перетворювач

I2C – Inter-Integrated Circuit

SPI – Serial Peripheral Interface

UART – Universal Asynchronous Receiver-Transmitter

CAN – Controller Area Network

HTTP – Hypertext Transfer Protocol

HTTPS – Hypertext Transfer Protocol Secure

МГц – мегагерц

IDE – Integrated Development Environment

CSS – Cascading Style Sheets

API – Application Programming Interface

HTML – HyperText Markup Language

JSON – JavaScript Object Notation

ОС – операційна система

MAC – Media Access Control

GND – ground

VCC – voltage common collector

					КВРКІ.220018.22.01.33 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

## ВСТУП

Кіберфізичні системи є комплексом програмних і апаратних засобів, які забезпечують автоматизоване управління фізичними об'єктами на основі зібраних даних. Однією з таких систем є «Розумний будинок», який включає датчики, контролери та виконавчі пристрої для моніторингу та керування параметрами житлових або промислових приміщень. Одним з аспектів функціонування «Розумного будинку» є контроль енергоспоживання з метою його оптимізації та раціонального використання ресурсів.

Системи моніторингу енерговитрат у кіберфізичних системах дозволяють отримувати дані про споживання електроенергії в режимі реального часу, аналізувати їх та приймати рішення щодо управління електроприладами. Дані системи можуть ґрунтуватися на різних апаратних платформах, серед яких популярними є мікроконтролери, такі як Arduino. Використання мікроконтролерів забезпечує можливість створення автономних рішень, які можуть інтегруватися в загальну архітектуру «Розумного будинку».

Об'єктом дослідження є система моніторингу енерговитрат у кіберфізичній системі «Розумний будинок». Предметом дослідження є методи збору, обробки та відображення даних про енергоспоживання з використанням апаратних та програмних засобів на базі платформи Arduino.

Метою роботи є розробка та дослідження системи моніторингу енерговитрат для кіберфізичної системи «Розумний будинок» на основі мікроконтролерів, яка дозволяє автоматизовано реєструвати, аналізувати та візуалізувати показники електроспоживання.

Для досягнення поставленої мети необхідно проаналізувати концепцію «Розумного будинку» та його енергоспоживання, розглянути існуючі системи моніторингу, визначити їхні технічні особливості, переваги та обмеження. На основі отриманих даних буде обґрунтовано вибір апаратного та програмного забезпечення, розроблено архітектурну схему та алгоритм функціонування системи.

					КВРКІ.220018.22.01.33 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

Проектована система включатиме апаратну частину, що складається з датчиків та мікроконтролера, програмне забезпечення для збору даних, а також веб-інтерфейс для візуалізації параметрів енергоспоживання.

У ході роботи буде проведено тестування розробленої системи з метою перевірки коректності збору та обробки даних. Аналіз результатів дозволить оцінити функціональність системи та можливість її використання у реальних умовах.

Практична цінність роботи полягає у можливості застосування розробленої системи для моніторингу енерговитрат у житлових і промислових приміщеннях. Це дозволяє отримувати інформацію про рівень енергоспоживання, що може бути використано для прийняття рішень щодо оптимізації використання електроенергії.

					КВРКІ.220018.22.01.33 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

# 1 АНАЛІЗ СИСТЕМ МОНІТОРИНГУ ЕНЕРГОВИТРАТ У РОЗУМНОМУ БУДИНКУ

## 1.1 Аналіз концепції «розумного будинку» та його енергоефективності

Концепція «розумного будинку» передбачає інтеграцію автоматизованих систем контролю та управління інженерними комунікаціями будівлі. Такі системи об'єднують електропостачання, опалення, вентиляцію, кондиціонування, водопостачання, освітлення, безпеку та інші підсистеми в єдину інформаційно-керуючу інфраструктуру. Автоматизований контроль здійснюється за допомогою датчиків, контролерів і програмного забезпечення, що забезпечують збирання, обробку та аналіз даних.

Функціональна структура «розумного будинку» включає декілька рівнів, серед яких рівень збору даних, рівень обробки інформації та рівень управління виконавчими пристроями. На рівні збору даних використовуються датчики температури, вологості, руху, освітленості, споживання електроенергії та інші сенсори. Отримані дані передаються на контролери або сервери для подальшої обробки.

Рівень обробки інформації містить алгоритми аналізу отриманих показників та прийняття рішень щодо коригування параметрів функціонування інженерних систем. Взаємодія з користувачем відбувається через графічні інтерфейси, мобільні додатки або веб-інтерфейси, що дозволяють контролювати стан обладнання та вносити зміни в налаштування [1].

На рисунку 1.1 представлено типову структуру «розумного будинку», що містить основні складові системи: сенсори, контролери, виконавчі пристрої та інтерфейси управління. Взаємодія між компонентами здійснюється через дротові або бездротові комунікаційні протоколи.

					КВРКІ.220018.22.01.33 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

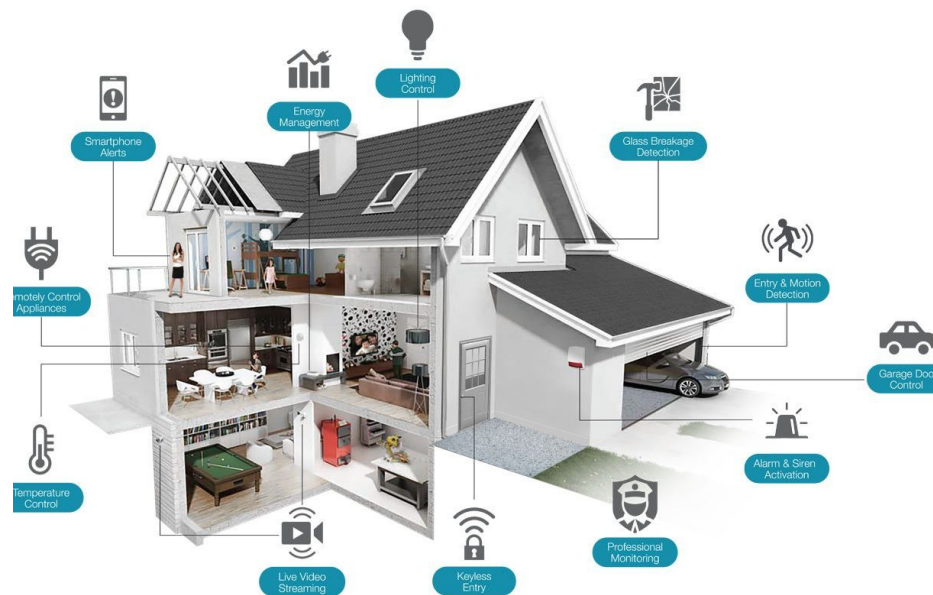


Рисунок 1.1 – Типова структура системи «Розумний будинок»[2]

Одна з особливостей «розумного будинку» полягає у можливості регулювання енергоспоживання за допомогою автоматичних сценаріїв, що адаптуються до умов експлуатації приміщень. Наприклад, система може коригувати інтенсивність освітлення залежно від природного освітлення або регулювати потужність опалювальних пристроїв відповідно до температурних змін [2].

Моніторинг енергоспоживання у «розумному будинку» здійснюється за допомогою спеціалізованих лічильників та датчиків, що реєструють параметри електричних навантажень у режимі реального часу. Інформація передається в обчислювальні модулі, які виконують аналіз витрат електроенергії та формують рекомендації щодо оптимізації використання ресурсів.

Для реалізації моніторингу використовуються різні технічні рішення, включаючи автономні датчики з бездротовим передаванням даних, централізовані контролери або розподілені обчислювальні системи, що взаємодіють із хмарними сервісами. Вибір апаратної платформи залежить від масштабу об'єкта та необхідних функціональних можливостей.

Інформація про енергоспоживання може бути представлена у вигляді графіків, таблиць або інтерактивних візуалізацій, що дозволяє користувачам

оцінювати рівень використання електроенергії та приймати рішення щодо коригування режимів роботи обладнання.

Окрім безпосереднього вимірювання споживання електроенергії, системи можуть використовувати прогнозні алгоритми, що базуються на історичних даних та аналітичних моделях. Такі алгоритми дозволяють прогнозувати пікові навантаження, виявляти нераціональне використання ресурсів та пропонувати коригувальні заходи.

Функціонування системи моніторингу енерговитрат може бути інтегроване з іншими підсистемами «розумного будинку», що дозволяє узгоджене управління різними інженерними комунікаціями. Наприклад, дані про енергоспоживання можуть використовуватися для автоматичного регулювання роботи електроприладів або зміни режимів роботи системи опалення.

Використання автоматизованих систем моніторингу дозволяє отримувати детальні відомості про структуру енергоспоживання, що сприяє підвищенню точності контролю та коригуванню параметрів електропостачання відповідно до актуальних потреб користувачів.

Кіберфізична система «Розумний будинок» може використовувати розподілену архітектуру, в якій локальні контролери обробляють первинні дані та передають їх до центрального сервера або хмарного сервісу. Це забезпечує можливість гнучкого масштабування системи та підвищення надійності її роботи [3].

Моніторинг енерговитрат у «розумному будинку» може здійснюватися як у локальному режимі, так і через віддалені сервери, що забезпечує доступ до даних з будь-якої точки. Віддалений доступ дозволяє користувачам переглядати статистику, змінювати налаштування та отримувати сповіщення про відхилення від заданих параметрів.

					КВРКІ.220018.22.01.33 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

## 1.2 Огляд існуючих систем моніторингу енерговитрат

Системи моніторингу енерговитрат призначені для збору, обробки та аналізу інформації про споживання електроенергії різними пристроями та системами. Вони можуть бути реалізовані у вигляді апаратно-програмних комплексів, що містять сенсори, контролери, комунікаційні модулі та програмне забезпечення для візуалізації даних.

На рисунку 1.2 представлена типова схема системи моніторингу, що складається декількох зон, передавальних модулів і централізованого сервера.

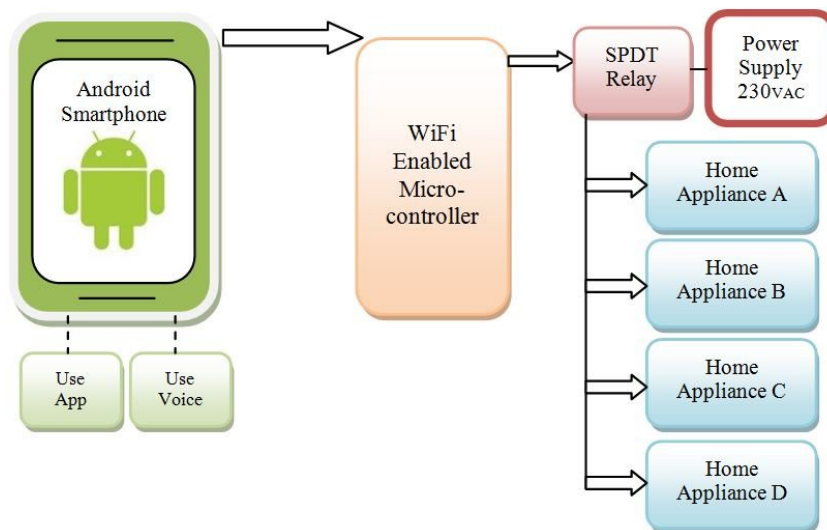


Рисунок 1.2 – Типова схема системи моніторингу [3]

Дані надходять у систему керування, де виконуються їх аналіз та обробка.

Існуючі системи моніторингу можна класифікувати за способом передачі даних, рівнем автономності та технологічною платформою. Вони можуть працювати автономно або інтегруватися у загальну систему автоматизованого керування будівлею.

Деякі пристрої здійснюють локальний збір інформації та передають дані через дротові канали зв'язку. Інші використовують бездротові технології, такі як Wi-Fi, Zigbee або LoRaWAN [4], що дозволяє інтегрувати їх у розподілені системи.

Одним із прикладів систем моніторингу є пристрої Shelly EM. Вони містять

вбудовані Wi-Fi модулі та працюють із мобільними застосунками для перегляду статистики в режимі реального часу. Візуалізація споживання відбувається у вигляді графіків та цифрових показників.

На рисунку 1.3 зображено включення пристрою Shelly EM [5], який підключається до електромережі через трансформатори струму та передає інформацію через Wi-Fi. Діапазон вимірювання дозволяє контролювати споживання приладів.



Рисунок 1.3 – Пристрій Shelly EM [5]

Іншим прикладом є система моніторингу Sonoff POW R3 [6]. Вона реалізує контроль енергоспоживання з можливістю віддаленого керування електроприладами. Передача даних здійснюється через Wi-Fi, а керування можливе за допомогою мобільного застосунку або через інтеграцію у системи автоматизації.

На рисунку 1.4 представлено архітектуру системи на базі Sonoff POW R3, що включає реле, вимірювальний модуль та хмарний сервер для обробки інформації. Дані можуть бути передані в системи розумного будинку через API.

					КВРКІ.220018.22.01.33 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

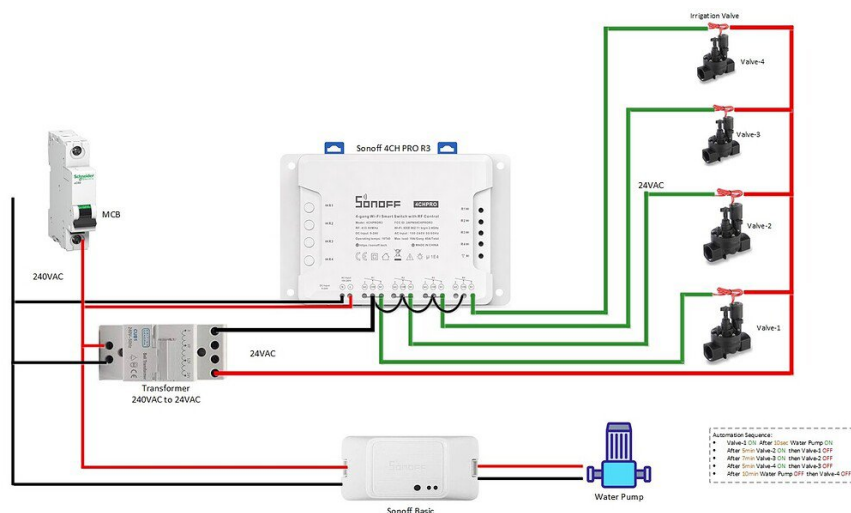


Рисунок 1.4 – Архітектура системи на базі Sonoff POW R3 [6]

Деякі системи моніторингу підтримують взаємодію з платформами IoT, такими як Home Assistant або OpenHAB [7]. Це дозволяє використовувати дані енергоспоживання для автоматизованого керування електроприладами залежно від заданих сценаріїв.

Вбудовані алгоритми можуть прогнозувати рівень споживання електроенергії на основі історичних даних та виконувати автоматичне відключення пристроїв у разі перевищення заданих порогів.

Окрім споживчих пристроїв, існують промислові системи моніторингу, які використовуються для контролю електропостачання у великих будівлях. Вони включають багатоканальні лічильники, системи архівування даних та інтерфейси для аналізу навантаження.

Деякі системи моніторингу використовують принцип розподіленого збору даних, коли інформація передається від кількох пристроїв до центрального сервера через бездротові мережі. Це дозволяє об'єднати інформацію про енергоспоживання всіх підключених пристроїв у єдиній базі даних.

Програмне забезпечення для моніторингу може включати інструменти аналітики, що дозволяють проводити розрахунок прогнозних показників, визначати пікові навантаження та оцінювати витрати на електроенергію.

Деякі системи підтримують можливість інтеграції з системами управління

будівлею через протоколи Modbus, MQTT або BACnet [8], що забезпечує їхню взаємодію з іншими інженерними підсистемами.

Мережеві пристрої моніторингу можуть працювати з хмарними сервісами, що дозволяє користувачам переглядати інформацію про енергоспоживання з будь-якого місця через веб-інтерфейс або мобільний застосунок.

Для запобігання перевантаженням електромережі деякі системи можуть автоматично регулювати розподіл навантаження між приладами або вмикати резервні джерела живлення у разі виходу параметрів за допустимі межі.

Додаткові функції, такі як автоматичні звіти про витрати електроенергії, можуть використовуватися для аналізу споживання за певні періоди часу та визначення тенденцій зміни навантаження.

Окремі моделі пристроїв підтримують можливість підключення до систем керування через API, що дозволяє інтегрувати їх у складні розподілені системи автоматизації.

Існують також системи моніторингу, що використовують енергозберігаючі алгоритми, які оптимізують роботу електроприладів залежно від поточного навантаження та режимів роботи будівлі.

Пристрої, що підтримують IoT-архітектуру, можуть взаємодіяти з іншими елементами розумного будинку, передаючи дані у глобальні системи обробки та аналізу інформації. Розподілені системи дозволяють інтегрувати моніторинг енергоспоживання в загальну екосистему автоматизованого керування інженерними мережами та обчислювальними ресурсами.

### 1.3 Аналіз апаратних та програмних платформ для автоматизованого контролю енергоспоживання

Апаратні та програмні платформи, що використовуються для автоматизованого контролю енергоспоживання, включають мікроконтролери, сенсори, комунікаційні модулі, серверні рішення та програмне забезпечення для збору, обробки й аналізу даних. Вибір технічних засобів визначається вимогами до

					КВРКІ.220018.22.01.33 ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

точності вимірювань, швидкості обробки інформації, можливостей інтеграції з іншими системами та засобами візуалізації.

Мікроконтролери виступають центральними елементами таких систем. Вони виконують функції зчитування показників із датчиків, обробки отриманих даних та передавання їх на сервер або кінцевий пристрій. Серед платформ, що використовуються для контролю енергоспоживання, можна виділити Arduino, ESP8266, ESP32, Raspberry Pi та промислові ПЛК.

Arduino є апаратною платформою, що дозволяє створювати автоматизовані системи контролю на основі мікроконтролерів. Її особливістю є підтримка як аналогових, так і цифрових входів, що дає змогу підключати широкий спектр сенсорів для вимірювання параметрів електромережі. Доступні моделі відрізняються за характеристиками, зокрема за кількістю входів/виходів, обсягом пам'яті та продуктивністю.

Популярними моделями є Arduino Uno, Arduino Mega та Arduino Nano. Arduino Uno має 14 цифрових входів/виходів і 6 аналогових входів, що дозволяє реалізовувати базові системи моніторингу. Arduino Mega відрізняється розширеною кількістю входів/виходів, що робить її придатною для більш складних проєктів. Arduino Nano має компактні розміри, що дозволяє інтегрувати її в пристрій з обмеженим простором для монтажу [9].

Платформа підтримує підключення додаткових модулів, зокрема Ethernet, Wi-Fi та GSM, що дає змогу передавати дані у локальні та глобальні мережі. Бібліотека програмного забезпечення, доступна в середовищі розробки Arduino IDE, спрощує інтеграцію датчиків та обробку даних. Завдяки цьому можна швидко створювати прототипи систему збору та аналізу даних. Гнучкість платформи дозволяє адаптувати її для різноманітних сценаріїв використання, від простих домашніх рішень до промислових застосувань. Зовнішній вигляд одного з представників платформи наведено на рисунку 1.5.

					КВРКІ.220018.22.01.33 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

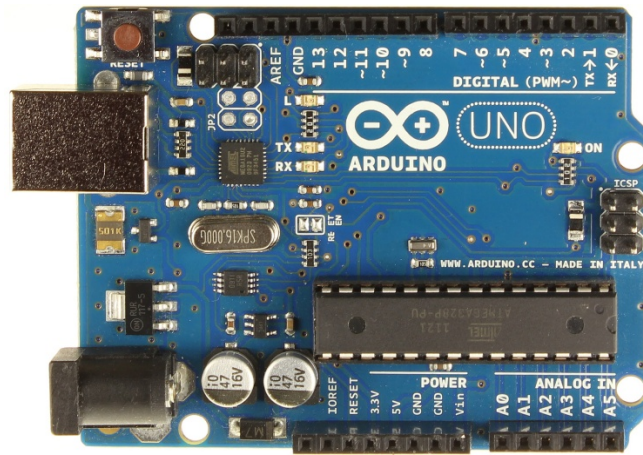
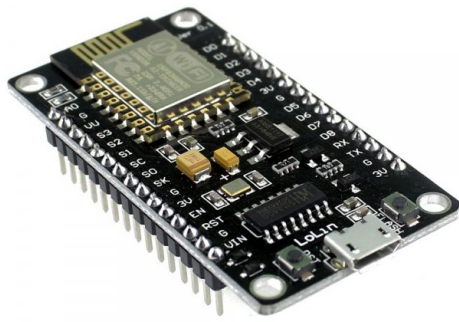


Рисунок 1.5 – Зовнішній вигляд платформи Arduino Uno R3 [9]

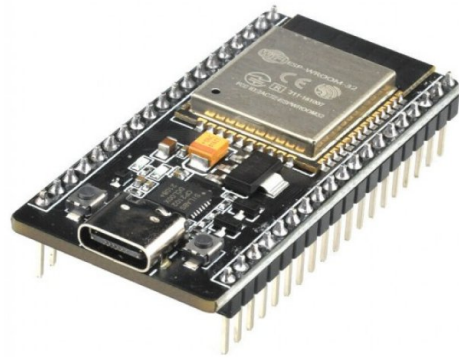
ESP8266 та ESP32 є мікроконтролерами, що мають вбудовані бездротові модулі Wi-Fi. Це дозволяє реалізовувати дистанційний контроль та передавання даних у локальні або хмарні сервери без використання додаткових модулів зв'язку. ESP32 додатково підтримує Bluetooth, що розширює можливості інтеграції з іншими пристроями, зокрема з мобільними телефонами та бездротовими датчиками.

ESP8266 має 11 цифрових входів/виходів, підтримку UART, I2C та SPI, а також 10-бітний АЦП, що дає змогу вимірювати напругу. Його продуктивність дозволяє виконувати базові обчислення та передавати дані в режимі реального часу. ESP32 має двоядерний процесор, підтримку кількох аналогових входів та розширені можливості енергозбереження [10].

Ці мікроконтролери підтримують підключення до MQTT-брокерів, що дозволяє інтегрувати їх у системи Інтернету речей. Вони сумісні з хмарними платформами, такими як Firebase та AWS IoT, що дає змогу реалізовувати розподілені системи моніторингу енергоспоживання. Зовнішній вигляд ESP8266 та ESP32 наведено на рисунку 1.6.



ESP8266



ESP32

Рисунок 1.6 – Зовнішній вигляд ESP8266 та ESP32 [10]

Raspberry Pi є одноплатним комп'ютером, що працює під управлінням операційної системи. Це дає змогу використовувати його для складних обчислювальних задач, роботи з базами даних та створення веб-інтерфейсів для візуалізації даних моніторингу.

Основними моделями є Raspberry Pi 3, Raspberry Pi 4 та Raspberry Pi Zero. Raspberry Pi 3 має чотириядерний процесор, 1 ГБ оперативної пам'яті та підтримку бездротових інтерфейсів Wi-Fi та Bluetooth. Raspberry Pi 4 має покращені характеристики, включно з 2, 4 або 8 ГБ оперативної пам'яті та підтримкою Gigabit Ethernet, що дозволяє обробляти великі обсяги даних у реальному часі. Raspberry Pi Zero є компактним варіантом з мінімальним енергоспоживанням, що підходить для вбудованих систем [11].

Платформа також має можливість працювати з різними операційними системами, такими як Ubuntu, Raspberry PI OS, та Windows, що розширює можливості налаштування системи під конкретні потреби.

Завдяки підтримці мов програмування Python, C++ та JavaScript, Raspberry Pi може взаємодіяти з сенсорами та серверами. Використання баз даних, зокрема SQLite та PostgreSQL [12], дозволяє зберігати великі масиви даних моніторингу та виконувати аналітичні розрахунки. Зовнішній вигляд одного з представників платформи наведено на рисунку 1.7.

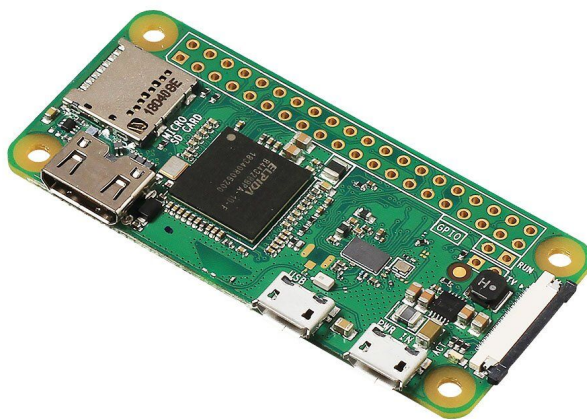


Рисунок 1.7 – Зовнішній вигляд Raspberry Pi Zero [11]

Програмовані логічні контролери (ПЛК) використовуються в промислових системах контролю енергоспоживання. Вони підтримують широкий набір протоколів зв'язку, зокрема Modbus, CAN та BACnet, що забезпечує сумісність із промисловим обладнанням та системами керування будівлями.

Основними виробниками ПЛК є Siemens, Schneider Electric та Allen-Bradley. Контролери Siemens SIMATIC мають розширені можливості для інтеграції у великі промислові системи та підтримують візуалізацію даних через SCADA-системи. Schneider Electric M221 [13] є компактним рішенням для автоматизації невеликих об'єктів, а Allen-Bradley CompactLogix використовується у складних промислових системах з підтримкою розподілених модулів вводу/виводу.

ПЛК можуть працювати у режимах реального часу, виконуючи аналіз енергоспоживання та автоматизоване керування навантаженнями. Вони підтримують підключення датчиків струму, напруги та температури через аналогові входи та комунікаційні інтерфейси.

Завдяки стійкості до промислових умов експлуатації, ПЛК застосовуються у виробничих комплексах, енергетичних системах та великих комерційних будівлях. Вони можуть працювати автономно або у складі мережесистем моніторингу, передаючи дані у центральні сервери для подальшої обробки.

Для збору даних про параметри електромережі використовуються датчики струму, напруги та потужності. Вони можуть бути реалізовані у вигляді

трансформаторів струму, шунтів або спеціалізованих вимірювальних мікросхем, таких як INA219 або ACS712.

Системи моніторингу можуть працювати як автономно, так і у складі комплексних рішень, що інтегруються з іншими пристроями автоматизації. Локальні платформи забезпечують обробку та збереження даних без необхідності використання зовнішніх серверів. Це дозволяє підвищити рівень автономності та зменшити залежність від інтернет-з'єднання.

Home Assistant надає широкі можливості для збирання інформації про споживання енергії. Система підтримує підключення різноманітних пристроїв, включаючи розумні розетки, лічильники та сенсори струму. Зібрані дані можуть використовуватись для аналізу та керування електроспоживанням, зокрема для автоматичного відключення приладів у певний час доби або за перевищенням встановлених порогів.

OpenHAB реалізує аналогічний підхід, дозволяючи інтегрувати різні протоколи та стандарти автоматизації. Платформа працює через спеціальні зв'язуючі модулі, що забезпечують взаємодію з контролерами, лічильниками та датчиками. Однією з можливостей є налаштування сценаріїв автоматизації, наприклад, регулювання роботи електричних приладів залежно від рівня навантаження в мережі.

Node-RED є графічним середовищем для побудови логіки обробки даних. Вона використовує блоки-вузли, які можна з'єднувати між собою, створюючи схеми обробки інформації. Це дозволяє отримувати дані від сенсорів, перетворювати їх у потрібний формат та надсилати у локальні або хмарні бази даних.

Хмарні сервіси відкривають можливості для дистанційного контролю та зберігання даних. Інформація про енергоспоживання передається на віддалені сервери, що дозволяє аналізувати її через веб-інтерфейси або мобільні додатки.

Blynk надає інструменти для візуалізації даних на мобільних пристроях. Платформа підтримує взаємодію з мікроконтролерами, що дозволяє отримувати

					КВРКІ.220018.22.01.33 ПЗ	Арк. 18
Зм.	Арк.	№ докум.	Підпис	Дата		

інформацію про споживання енергії та керувати пристроями через смартфон.

ThingsBoard використовується для побудови панелей моніторингу. Вона підтримує інтеграцію з IoT-пристроями через протоколи MQTT, HTTP або CoAP. Дані можуть оброблятися в режимі реального часу, а також архівуватися для подальшого аналізу.

Google Cloud IoT та AWS IoT пропонують розширені можливості для обробки даних у великих масштабах. Вони дозволяють використовувати алгоритми машинного навчання для прогнозування енергоспоживання та оптимізації навантаження [14].

Окремий напрямок - SCADA-системи, що застосовуються на промислових об'єктах. Вони дозволяють контролювати параметри електромережі, аналізувати історичні дані та формувати аналітичні звіти.

Для організації доступу до даних моніторингу можуть використовуватися веб-сервери. Одним із варіантів є реалізація сервера на базі Flask або Django, що дозволяє створювати веб-інтерфейс для перегляду поточних показників енергоспоживання.

Node.js з фреймворком Express забезпечує швидке створення серверних додатків для отримання, обробки та візуалізації даних. Веб-сервери на основі Express можуть взаємодіяти з базами даних, обробляти запити користувачів та відображати інформацію у вигляді динамічних веб-сторінок [15].

Для інтеграції із системами автоматизації часто використовують брокери MQTT, що забезпечують передачу даних між сенсорами, контролерами та серверами. Це дає змогу реалізувати веб-інтерфейси, які відображають дані про електроспоживання в реальному часі.

Застосування конкретної платформи залежить від вимог до обробки інформації, можливостей інтеграції з іншим обладнанням та способу збереження даних.

					КВРКІ.220018.22.01.33 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

#### 1.4 Порівняння підходів до збору, обробки та візуалізації енергоспоживання

У процесі збору, обробки та візуалізації даних про енергоспоживання можна використовувати різні методи передачі та обробки інформації. Один із підходів, який я застосував у своїй роботі, полягає у використанні плати, що знімає показники споживання енергії, а потім передає ці дані через HTTP запит на веб-сервер.

Зібрані дані про енергоспоживання, такі як потужність, струм, напруга, передаються за допомогою протоколу HTTP, що забезпечує їх доставку до веб-сервера для подальшої обробки. Використання HTTP дозволяє налагодити взаємодію між пристроєм, який збирає дані, і сервером, що їх обробляє, а також спрощує інтеграцію з іншими веб-технологіями. Веб-сервер приймає отримані дані та зберігає їх у базі даних для подальшого аналізу або одразу передає на веб-сторінку.

На веб-сторінці дані можуть бути представлені у вигляді графіків, таблиць або інших елементів, що дозволяють зручно відстежувати зміни в енергоспоживанні. Такий підхід забезпечує простоту доступу до даних і дозволяє користувачам в реальному часі відстежувати параметри енергоспоживання на зручному інтерфейсі. Візуалізація на веб-сторінці дає змогу швидко отримати необхідну інформацію, не вдаючись до складних додаткових налаштувань або спеціальних програм [16].

Передача даних через HTTP має кілька переваг. Зокрема, цей протокол є стандартом для веб-додатків і підтримується більшістю пристроїв та платформ. Це дозволяє легко інтегрувати різні компоненти системи, створюючи єдину мережу, яка здатна працювати з великим обсягом даних. HTTP дає змогу швидко і без зайвих складнощів передавати дані від сенсорів на сервер для обробки та зберігання.

При цьому передача даних по HTTP має й певні обмеження. Наприклад, якщо з'єднання з сервером переривається або виникають проблеми з мережею, це може призвести до втрати частини даних або неможливості їх передачі в реальному часі.

					КВРКІ.220018.22.01.33 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

Однак така проблема може бути вирішена за допомогою додаткових механізмів зберігання даних на локальному пристрої, поки з'єднання не буде відновлено.

Для забезпечення безпеки передаваної інформації через HTTP можна використовувати HTTPS, що дозволяє зашифрувати дані та захистити їх від несанкціонованого доступу під час передачі. Це необхідно, якщо дані містять чутливу інформацію про енергоспоживання, особливо в великих або комерційних системах.

Порівняно з іншими методами збору даних, такими як MQTT чи Modbus, HTTP є універсальним і широко використовуваним протоколом, що добре інтегрується в сучасні веб-додатки. Хоча HTTP може не підходити для систем, що потребують миттєвої передачі даних, для більшості задач, пов'язаних з моніторингом енергоспоживання, цього цілком достатньо [17].

Отже, використання передачі даних через HTTP для збору та візуалізації енергоспоживання є зручним і доступним методом для реалізації таких систем. Цей підхід дозволяє отримувати актуальну інформацію у реальному часі, забезпечуючи простоту інтеграції з іншими технологіями і зручність виведення даних для користувача.

### 1.5 Постановка завдання для розробки системи моніторингу енерговитрат

Метою розробки системи моніторингу енерговитрат є забезпечення можливості постійного контролю за споживанням електричної енергії в межах будинку або окремого вузла в будівлі. Система повинна здійснювати вимірювання споживаної потужності в реальному часі та забезпечувати її відображення на зручному інтерфейсі. Вона повинна бути спроектована таким чином, щоб мати можливість моніторити навантаження не менше ніж 5 А, що забезпечить її застосування в різних типах споживачів електричної енергії.

Основним завданням системи є збір даних про енергоспоживання, що здійснюється за допомогою сенсорів або вимірювальних пристроїв. Дані повинні передаватися бездротовим зв'язком на центральний сервер або безпосередньо на

					КВРКІ.220018.22.01.33 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

веб-сторінку для відображення. Для цього можна використовувати бездротові технології, що забезпечують стабільну передачу даних на відстань, зокрема Wi-Fi або Bluetooth, залежно від конкретних умов і вимог до системи.

Для відображення зібраної інформації передбачається використання веб-сторінки, яка буде виступати в якості монітора. Інтерфейс веб-сторінки має надавати користувачу можливість переглядати поточні значення потужності, а також переглядати історію енергоспоживання за визначений період часу. Веб-сторінка повинна бути адаптована для зручного перегляду на різних пристроях, таких як комп'ютери, планшети чи смартфони.

Система має забезпечити передачу даних на веб-сторінку в реальному часі. Це дозволить користувачам оперативно відстежувати зміни в споживанні енергії і здійснювати коригування, якщо це необхідно. Веб-сторінка повинна підтримувати оновлення даних без необхідності перезавантаження сторінки, щоб забезпечити безперервний моніторинг поточних показників.

При розробці системи також слід враховувати простоту її налаштування та використання, щоб кінцевий користувач міг легко встановити та налаштувати необхідні параметри без значних технічних знань. Передбачається, що система буде готова до розширення, що дозволить в подальшому додавати нові пристрої або змінювати конфігурацію відповідно до змін у структурі енергоспоживання [18].

Таким чином, система моніторингу енерговитрат повинна забезпечувати збір і передачу даних про енергоспоживання, відображати їх на веб-сторінці та дозволяти користувачу здійснювати моніторинг в реальному часі, надаючи можливість для подальшого аналізу та коригування енергоспоживання.

## 1.6 Висновки

На основі проведеного аналізу відомих комерційних рішень кіберфізичних систем «Розумний будинок» та огляду доступних засобів для розробки таких систем можна зробити висновок, що хоча сучасні технології пропонують широкі

					КВРКІ.220018.22.01.33 ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

можливості для моніторингу енергоспоживання, вартість готових рішень залишається високою, що може бути перешкодою для широкого застосування.

Крім того, більшість рішень вимагають втручання в проводку, що, хоч і необхідно для точного вимірювання споживаної енергії, не дозволяє уникнути складнощів при монтажі та налаштуванні.

На основі проведених досліджень було сформовано набір функцій, які повинні бути реалізовані в системах моніторингу енерговитрат. Система повинна підтримувати бездротову передачу даних, що дозволяє уникнути необхідності прокладати додаткові кабелі і полегшує монтаж. Також система має працювати в режимі реального часу, щоб можна було оперативно відстежувати зміни в параметрах споживання. Інформація повинна відображатися на зручному інтерфейсі, наприклад, на веб-сторінці, що дасть можливість користувачам отримувати актуальні дані за допомогою будь-якого пристрою з доступом до Інтернету.

Таким чином, для розробки системи моніторингу енерговитрат необхідно забезпечити простоту інтеграції з існуючою електричною мережею, зручність в експлуатації, а також можливість відображення даних у реальному часі через бездротові технології.

					КВРКІ.220018.22.01.33 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

## 2 ПРОЄКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ЕНЕРГОВИТРАТ НА ОСНОВІ ARDUINO

2.1 Вибір апаратного забезпечення для збору та передачі даних про енергоспоживання

Для створення системи моніторингу енергоспоживання була обрана апаратна платформа ESP32 ревізія Wroom. Плата ESP32 оснащена двоядерним процесором з частотою до 240 МГц, що дозволяє обробляти дані з високою швидкістю. Це дає можливість виконувати обчислення для вимірювання параметрів енергоспоживання, їх обробку та передачу даних на сервер або веб-сторінку [19].

Одним з факторів вибору ESP32 є наявність вбудованого модуля Wi-Fi, який дозволяє бездротово передавати дані на сервер або веб-сторінку. Це усуває необхідність додаткових плат для підключення до мережі, що спрощує реалізацію системи.

На рисунку 2.1 наведено розташування елементів на платі.

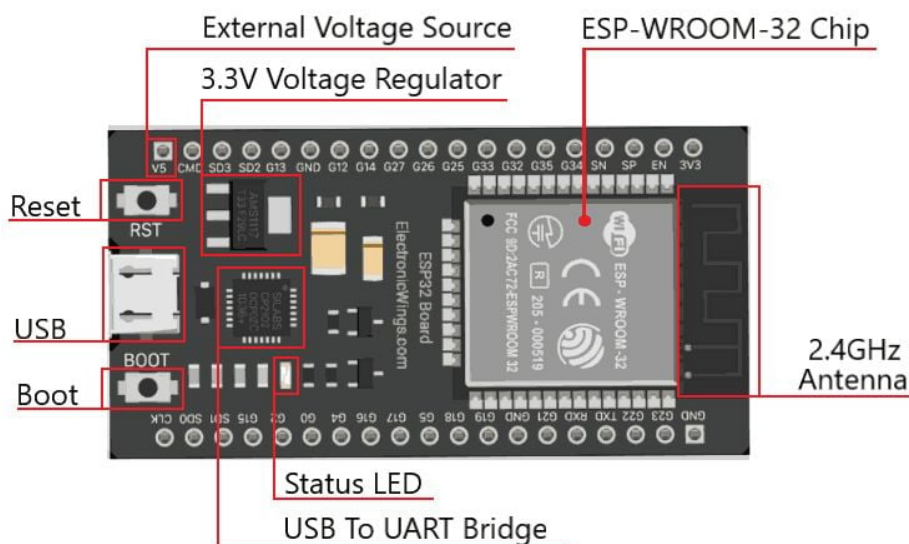


Рисунок 2.1 – Елементи ESP-Wroom-32 [20]

ESP32 має 34 цифрових пінів, з яких до 15 можна використовувати як аналогові входи. Це дозволяє підключати різні типи сенсорів для вимірювання

електричних параметрів, таких як напруга, струм і потужність. Плата підтримує такі інтерфейси, як SPI, I2C, UART [21] і PWM, що дає змогу додавати різні периферійні пристрої, такі як дисплеї, реле, датчики та інші компоненти. Це дозволяє створювати розширені системи моніторингу енергоспоживання з додатковими функціями управління та виведення інформації.

ESP32 має вбудований 12-бітний аналогово-цифровий перетворювач (АЦП), який дозволяє здійснювати вимірювання аналогових сигналів з роздільною здатністю 12 біт. Це дозволяє проводити точні вимірювання змінних напруг та струмів. Оскільки ESP32 працює на напрузі 3,3 В, вхідні аналогові сигнали, які подаються на пін, повинні бути в межах цього рівня. Плата здатна вимірювати сигнали від 0 до 3,3 В, і результат перетворення буде у вигляді цифрового значення, що відображає величину вхідної напруги у діапазоні від 0 до 4095 (12 біт). Таким чином, роздільна здатність перетворення дозволяє отримати точні дані при вимірюванні низьких та високих рівнів сигналу.

АЦП на ESP32 працює з кількома каналами одночасно, однак для забезпечення стабільних та точних вимірювань одночасно використовувати кілька каналів може бути складно через обмеження вхідної пропускну здатності при активному використанні Wi-Fi модуля. Коли Wi-Fi активно використовується для передачі даних, ресурси мікроконтролера можуть бути значною мірою завантажені, що може впливати на стабільність роботи АЦП, особливо при вимірюваннях на кількох каналах одночасно. Це може обмежити точність вимірювань або навіть призвести до затримок в обробці даних. Тому при реалізації системи моніторингу енергоспоживання з використанням Wi-Fi модуля треба враховувати можливість обмеженого використання каналів АЦП при високих вимогах до швидкості та точності передачі даних.

Крім того, існують додаткові параметри, які можуть вплинути на якість вимірювань АЦП. Наприклад, на точність вимірювань може впливати шум та інтерференція, що виникають через навколишні електричні пристрої, які можуть створювати електромагнітні перешкоди. Для мінімізації таких впливів можна

					КВРКІ.220018.22.01.33 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

використовувати екранування або додаткові фільтри на входах АЦП для поліпшення точності вимірювань. Також необхідно налаштувати пін перед підключенням сенсорів для мінімізації спотворень сигналу.

При підключенні зовнішніх сенсорів для вимірювання напруги чи струму до аналогових входів ESP32 необхідно враховувати, що на виході ці сенсори повинні генерувати сигнал в межах 0–3,3 В, відповідно до характеристик АЦП плати. Наприклад, при використанні сенсора ACS712 для вимірювання струму або ZMPT101B [22] для вимірювання напруги, сигнал, що подається на пін ESP32, має бути відповідно оброблений та відфільтрований для забезпечення коректного вимірювання.

ESP32 підтримує також функції низького енергоспоживання при використанні модуля Wi-Fi. Плата має кілька режимів сну, зокрема, глибокий режим сну, при якому споживання енергії може знижуватися до мікроскопічних значень, що дозволяє збільшити час роботи системи при автономному живленні, наприклад, від батареї. У режимі сну основні функції мікроконтролера відключаються, а Wi-Fi модуль або інші периферійні пристрої можуть бути виведені з роботи, що значно знижує енергоспоживання.

Враховуючи численні функціональні можливості ESP32, такі як наявність двоядерного процесора, вбудованого Wi-Fi модуля, багатьох пінів для підключення периферії та точного АЦП, ця плата є дуже гнучким рішенням для створення систем моніторингу енергоспоживання, де потрібно обробляти різноманітні сигнали і передавати дані в реальному часі, зберігаючи низьке споживання енергії. Це дозволяє розширювати систему в залежності від вимог конкретного проекту та забезпечує можливість інтеграції з іншими пристроями та мережами.

Для отримання потужності в системі моніторингу енергоспоживання до плати ESP32 підключаються два основних елементи: сенсор струму ACS712 та сенсор напруги ZMPT101B. Використання окремого датчика для вимірювання напруги є необхідним, оскільки напруга в електричній мережі може змінюватися, і для точного вимірювання її значення потрібен спеціалізований сенсор.

					КВРКІ.220018.22.01.33 ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

ACS712 є сенсором струму, який дозволяє вимірювати змінний або постійний струм. Він працює на принципі вимірювання зміщення напруги на виході пропорційного до сили струму, що проходить через провідник. Датчик має вбудоване внутрішнє перетворення аналогового сигналу в напругу, яку можна обробляти мікроконтролером. Діапазон вимірювань цього сенсора становить від  $\pm 5$  А до  $\pm 20$  А залежно від конкретної моделі, що дозволяє точно вимірювати силу струму в межах цього діапазону.

ACS712 використовує принцип Холла [23] для вимірювання сили струму. Зміна магнітного поля, яке виникає при проходженні струму через провідник, інтерпретується сенсором як зміна вихідної напруги. Напруга на виході пропорційна величині струму, що дозволяє отримувати точні дані про величину струму в електричній мережі.

Розташування пінів на модулі ACS712 для з'єднання з іншими пристроями наведено на рисунку 2.2.

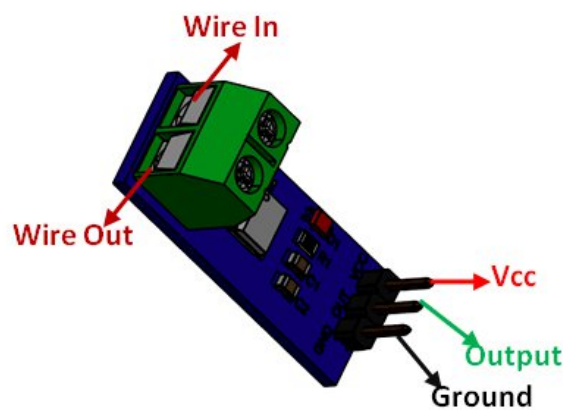


Рисунок 2.2 – Розташування виходів ACS712 [24]

Вихідний пін сенсора підключається до одного з аналогових входів плати ESP32, що дозволяє проводити аналіз та обробку сигналу. Для забезпечення точних вимірювань, сенсор потребує стабільного живлення, яке забезпечується від 5 В джерела живлення.

Технічні характеристики ACS712 включають точність вимірювання, яка визначається відхиленням вихідної напруги від ідеальної лінії. Діапазон роботи

					КВРКІ.220018.22.01.33 ПЗ	Арк. 27
Зм.	Арк.	№ докум.	Підпис	Дата		

сенсора також можна налаштувати в залежності від необхідних вимірів, що дозволяє використовувати його в різних умовах.

Для вимірювання напруги в мережі використовується сенсор ZMPT101B. Цей сенсор дозволяє вимірювати змінну напругу в мережі, перетворюючи її на пропорційну напругу, яку можна зчитувати за допомогою аналогового входу мікроконтролера. Використання окремого датчика для вимірювання напруги є необхідним, оскільки напруга в міських електричних мережах може коливатися, і для точного вимірювання її величини потрібен спеціалізований сенсор.

Елементи конструкції та розташування пінів сенсора ZMPT101B наведено на рисунку 2.3, де показані основні компоненти пристрою.

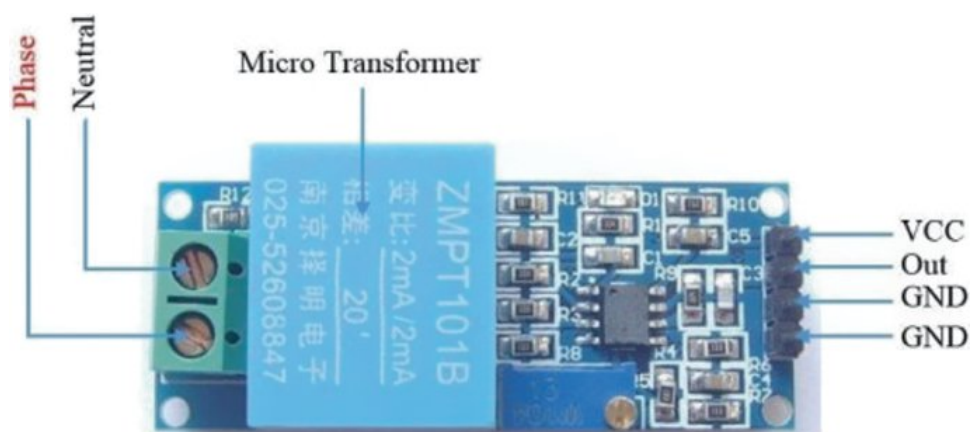


Рисунок 2.3 – Конструкція сенсора ZMPT101B [25]

Система моніторингу енергоспоживання буде підключатися до вузла, який у даному випадку буде реалізований через мережевий фільтр на 5 пристроїв, який наведено на рисунку 2.4.



## Рисунок 2.4 – Мережевий фільтр

Мережевий фільтр дозволяє зменшити вплив перешкод та коливань напруги, що можуть виникати в мережі живлення, на роботу підключених пристроїв. Це необхідно для забезпечення стабільного функціонування сенсорів і електроніки, яка може бути чутливою до змін у напрузі. Крім того, мережевий фільтр дозволяє зменшити вплив високочастотних електромагнітних завад, що можуть виникати внаслідок роботи пристроїв, підключених до мережі живлення.

Вузол також забезпечує підключення до п'яти різних пристроїв, що дозволяє здійснювати моніторинг енергоспоживання одразу на кількох точках, кожен з яких може бути підключений до свого окремого порту фільтра. Це дозволяє отримувати більш точні дані про споживану потужність на кожному з підключених пристроїв, що може бути потрібним для подальшого аналізу енергоспоживання та оптимізації роботи системи.

Завдяки такій архітектурі, система може здійснювати моніторинг енергоспоживання не лише на рівні одного пристрою, а й на кількох підключених до мережевого фільтра, що дозволяє забезпечити більшу гнучкість у вимірюваннях та зібранні даних для подальшої обробки.

### 2.2 Вибір програмного забезпечення для збору, обробки та відображення інформації

Для розробки програмної частини системи моніторингу енергоспоживання використовується Arduino IDE [26]. Це середовище підтримує мікроконтролери ESP32, дозволяє писати код мовою C++ та містить засоби для компіляції й завантаження прошивки на пристрій. Arduino IDE має можливість додавання бібліотек, що дозволяє працювати з різними периферійними модулями, у тому

					КВРКІ.220018.22.01.33 ПЗ	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		

числі сенсорами напруги та струму, а також мережевими інтерфейсами.

Програмна частина контролера реалізована мовою C++. Це дає змогу керувати апаратними ресурсами пристрою, працювати з вхідними та вихідними сигналами, а також взаємодіяти з модулями для зчитування електричних параметрів. Використання C++ дозволяє працювати безпосередньо з регістрами та пам'яттю мікроконтролера, що знижує затримку обробки сигналів та передачі даних.

Для зчитування значень сили струму використовується бібліотека ACS712 [27]. Вона містить набір функцій для перетворення аналогового сигналу з виходу датчика ACS712 у цифрове значення, яке потім використовується для обчислення потужності споживання. Використання цієї бібліотеки спрощує взаємодію з сенсором та зменшує обсяг коду, необхідного для роботи з аналоговими сигналами.

Для вимірювання напруги використовується бібліотека EmonLib [28]. Вона містить функції для обробки сигналів із датчика напруги та перетворення їх у цифрові значення. Використання цієї бібліотеки дозволяє контролеру отримувати дані про зміну напруги в електромережі та виконувати обчислення в реальному часі.

Передача даних реалізована через вбудований модуль Wi-Fi, який є частиною ESP32. Робота з мережею здійснюється за допомогою бібліотек WiFi та HTTPClient. Бібліотека WiFi відповідає за підключення пристрою до бездротової мережі, отримання IP-адреси, керування з'єднанням та підтримку зв'язку з мережею.

Бібліотека HTTPClient [29] дозволяє мікроконтролеру відправляти HTTP-запити методом POST або GET. Використання цього підходу забезпечує передачу отриманих даних на сервер, який їх зберігає та обробляє. Передача інформації відбувається з використанням стандартного протоколу HTTP, що спрощує взаємодію між контролером та сервером.

Обробка та візуалізація інформації виконується за допомогою веб-інтерфейсу, який розроблено з використанням HTML, CSS і JavaScript [30]. HTML

					КВРКІ.220018.22.01.33 ПЗ	Арк. 30
Зм.	Арк.	№ докум.	Підпис	Дата		

використовується для структурування веб-сторінки, CSS визначає зовнішній вигляд елементів інтерфейсу, а JavaScript забезпечує оновлення даних у режимі реального часу.

JavaScript використовується для отримання даних із сервера та їх динамічного оновлення на сторінці. Для цього застосовуються запити до API сервера, які дозволяють періодично отримувати нові дані та оновлювати їх на екрані без перезавантаження сторінки.

Для створення серверної частини використовується середовище Node.js. Це середовище дозволяє виконувати JavaScript-код на сервері та підтримує обробку запитів від клієнтів. Node.js забезпечує взаємодію між мікроконтролером та веб-інтерфейсом, приймаючи та зберігаючи отримані від нього дані.

Для розгортання веб-сервера використовується фреймворк Express. Він забезпечує створення API, яке приймає HTTP-запити та передає клієнтам необхідну інформацію. Express дозволяє визначати маршрути запитів, обробляти отримані дані та надсилати відповіді у форматі JSON або HTML [31].

Обробка запитів клієнтів виконується за допомогою відповідних маршрутів, визначених у Express. При надходженні запиту сервер перевіряє наявність нових даних, виконує необхідні обчислення та відправляє клієнту оновлену інформацію.

Веб-інтерфейс оновлює отримані дані за допомогою AJAX-запитів. Це дозволяє змінювати інформацію на сторінці без необхідності перезавантаження, що покращує інтерактивність інтерфейсу та зменшує навантаження на сервер.

Збереження отриманих даних може виконуватися у вигляді локальних файлів або у базі даних. Використання бази даних дає можливість зберігати історію змін параметрів енергоспоживання та аналізувати тенденції на основі збережених записів.

Поєднання зазначених технологій дозволяє реалізувати систему моніторингу енергоспоживання, що включає збір, обробку та відображення даних у режимі реального часу. Веб-інтерфейс надає можливість віддаленого доступу до інформації, а серверна частина забезпечує взаємодію між контролером та

					КВРКІ.220018.22.01.33 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

клієнтським пристроєм.

### 2.3. Вимоги до апаратного забезпечення системи

На основі проведеного аналізу апаратного забезпечення та вибору елементів для реалізації системи моніторингу енергоспоживання сформульовані вимоги до апаратних компонентів. Для забезпечення збору, обробки та передачі інформації про електричні параметри необхідно використовувати наступні елементи:

- 1) мікроконтролерна плата ESP-Wroom-32;
- 2) датчик струму ACS712;
- 3) датчик напруги ZMPT101B;
- 4) мережевий фільтр для підключення електропристроїв;
- 5) дротові з'єднання для комутації компонентів;
- 6) макетна плата для розміщення модулів та зручності з'єднання;
- 7) конектор для забезпечення підключення живлення та сигналів.

Вказані компоненти забезпечують функціональність системи моніторингу енергоспоживання, дозволяючи вимірювати параметри електричної мережі, передавати отримані дані та відображати їх у користувацькому інтерфейсі.

### 2.4 Вимоги до програмного забезпечення та архітектури системи

Для розробки програмного забезпечення використовується середовище розробки Arduino IDE, яке підтримує написання, компіляцію та завантаження коду на мікроконтролерні платформи. Arduino IDE є міжплатформним середовищем, доступним для операційних систем Windows, Linux та macOS. Код у цьому середовищі пишеться мовами програмування C та C++ із використанням спеціальних бібліотек, що спрощують роботу з периферійними пристроями.

Програма для ESP-Wroom-32 складається з двох основних функцій: `setup()` і `loop()`. У функції `setup()` виконується ініціалізація апаратних модулів, мережевих підключень та бібліотек. Основний алгоритм реалізується у функції `loop()`, яка

					КВРКІ.220018.22.01.33 ПЗ	Арк. 32
Зм.	Арк.	№ докум.	Підпис	Дата		

виконується циклічно та обробляє дані з датчиків, передає їх на сервер і забезпечує обмін інформацією.

Для роботи з датчиками струму та напруги використовується бібліотека ACS712, яка дозволяє отримувати значення сили струму, та бібліотека EmonLib, що застосовується для обчислення потужності. Дані з датчиків надходять на аналогові входи мікроконтролера, після чого обробляються алгоритмом вимірювання, що включає калібрування та згладжування значень.

Для забезпечення бездротової передачі даних використовується бібліотека WiFi, яка відповідає за підключення ESP-Wroom-32 до мережі Wi-Fi. Передача даних до веб-сервера здійснюється за допомогою HTTP-запитів, реалізованих через бібліотеку HTTPClient.

Веб-інтерфейс системи реалізовано з використанням HTML, CSS та JavaScript. Ці технології забезпечують представлення отриманих даних у зручному вигляді та взаємодію з користувачем через браузер.

Для обробки HTTP-запитів, збереження отриманих даних та забезпечення роботи веб-інтерфейсу використовується серверна частина, розроблена на платформі Node.js із застосуванням фреймворку Express. Веб-сервер приймає дані від ESP-Wroom-32, обробляє їх та передає клієнтському веб-додатку.

Таким чином, програмне забезпечення системи складається з двох частин: прошивки для ESP-Wroom-32, яка забезпечує збір даних і їхню передачу, та серверного веб-додатку, що обробляє отриману інформацію та виводить її у вигляді графіків і таблиць у веб-інтерфейсі.

## 2.5 Архітектурна схема та алгоритм роботи системи моніторингу

Система моніторингу складається з трьох основних компонентів: апаратної частини, веб-сервера та веб-клієнта. Кожен з цих компонентів виконує свою функцію у процесі збору, обробки, збереження та візуалізації даних.

Апаратна частина представлена мікроконтролером ESP32, до якого підключені датчики для вимірювання електричних параметрів. Використовуються

					КВРКІ.220018.22.01.33 ПЗ	Арк. 33
Зм.	Арк.	№ докум.	Підпис	Дата		

датчики ACS712 для вимірювання струму та модуль EmonLib для розрахунку напруги та потужності. ESP32 підключається до локальної Wi-Fi мережі та передає зібрані дані на веб-сервер.

Веб-сервер розгорнутий на платформі Node.js. Він виконує функцію отримання, збереження та обробки отриманих від мікроконтролера даних. Сервер обробляє HTTP-запити від пристроїв, реєструє їх активність та відповідає на запити від веб-клієнта. Дані про споживану потужність зберігаються у внутрішній структурі сервера, а інформація про пристрої оновлюється в реальному часі.

Для забезпечення безперебійної роботи системи реалізовано механізми обробки помилок та повторної передачі даних у разі втрати з'єднання.

Веб-клієнт реалізований як браузерний додаток, який отримує дані від веб-сервера та відображає їх у графічному вигляді. Інформація оновлюється у режимі реального часу, що дозволяє користувачеві стежити за станом пристроїв та динамікою споживання електроенергії. Графіки з потужністю та спожитою енергією оновлюються на основі отриманих значень.

Архітектурна схема системи представлена на рисунку 2.5. Вона відображає взаємозв'язки між апаратною частиною, сервером та веб-клієнтом, а також порядок обміну даними між ними.

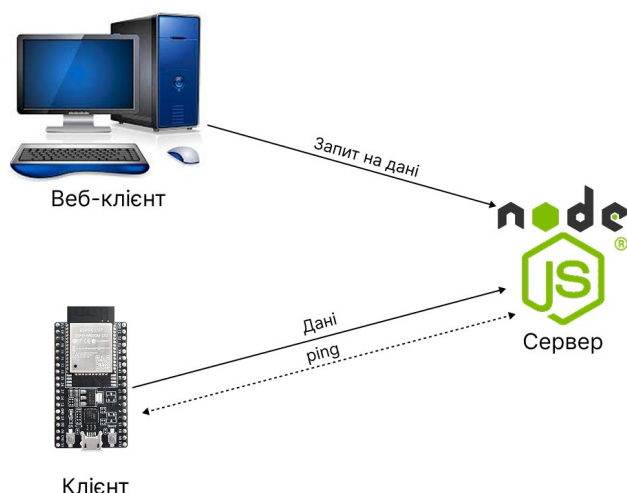


Рисунок 2.5 – Схематичне відображення архітектури системи

Взаємодія між компонентами здійснюється за допомогою протоколу HTTP.

					КВРКІ.220018.22.01.33 ПЗ	Арк. 34
Зм.	Арк.	№ докум.	Підпис	Дата		

Плата відправляє дані на сервер за допомогою GET-запиту [32], який містить параметри потужності та MAC-адресу пристрою. Сервер аналізує отриману інформацію, оновлює дані у своїй структурі та реєструє пристрій як активний. Веб-клієнт періодично надсилає запити до сервера для отримання актуальної інформації та її відображення.

Окремо реалізовано механізм перевірки з'єднання між пристроями. Сервер періодично перевіряє активність підключених пристроїв. Якщо пристрій не передає дані протягом заданого часу, він виключається зі списку активних. Це дозволяє контролювати наявність зв'язку та оперативно визначати втрату з'єднання.

Принцип роботи системи базується на циклічному зборі та передачі даних. Апаратна частина виконує ініціалізацію Wi-Fi-з'єднання, після чого здійснює зчитування показників струму та напруги. Отримані значення використовуються для обчислення потужності, яка відправляється на сервер. Сервер приймає значення, зберігає їх та оновлює інформацію про пристрої. Веб-клієнт періодично запитує сервер для отримання оновлених даних, що дозволяє відображати актуальний стан системи у вигляді графіків та числових показників.

Для реалізації перевірки підключень сервер запускає окремий процес, який кожну секунду аналізує останній час активності пристроїв. Якщо час бездіяльності перевищує встановлений поріг, пристрій виключається зі списку активних. Це дозволяє виявляти пристрої, які втратили зв'язок або припинили передавати дані.

Окремо передбачена можливість керування пристроями через веб-інтерфейс. Веб-клієнт надає функцію зміни імен підключених пристроїв, що дозволяє ідентифікувати їх за логічними назвами. Ця інформація синхронізується з сервером, що забезпечує її збереження та відображення в актуальному вигляді.

Блок-схеми алгоритмів роботи апаратної частини, сервера та веб-клієнта наведені у Додатку Б

## 2.6. Висновки

У цьому розділі було проведено аналіз існуючих апаратних засобів для

					КВРКІ.220018.22.01.33 ПЗ	Арк. 35
Зм.	Арк.	№ докум.	Підпис	Дата		

побудови системи моніторингу енергоспоживання у кіберфізичній системі «розумний будинок». Було розглянуто можливі варіанти апаратного забезпечення та обрано оптимальну конфігурацію на основі мікроконтролера ESP32, датчика ACS712 для вимірювання струму та датчика ZMPT101B для вимірювання напруги.

Для реалізації програмної частини було прийнято рішення використовувати середовище розробки Arduino IDE, що забезпечує зручну інтеграцію з обраним апаратним забезпеченням та підтримує необхідні бібліотеки для роботи з датчиками.

Серверна частина системи реалізована на основі веб-сервера Node.js, що дозволяє забезпечити обробку запитів від пристроїв, збереження отриманих даних та їх передачу клієнтам у режимі реального часу.

Для відображення отриманої інформації розроблено веб-клієнт на основі HTML та CSS, що дозволяє візуалізувати дані у вигляді графіків та таблиць.

Таким чином, у розділі визначено архітектуру системи моніторингу, обрано відповідне апаратне забезпечення та програмні засоби.

### **3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ МОНІТОРИНГУ ЕНЕРГОВИТРАТ**

#### **3.1 Розробка апаратної частини системи та її інтеграція з Arduino**

Для здійснення усіх з'єднань та повного апаратного підключення проекту необхідно провести такі з'єднання:

1. Підключити ZMPT101B до живлення й аналогового піна ESP32.
2. Підключити ACS712 до живлення й аналогового піна ESP32.
3. Підключити ACS712 послідовно в фазу.
4. Підключити ZMPT101B паралельно джерелу напруги.

Датчик ZMPT101B, призначений для вимірювання змінної напруги. Схема підключення ZMPT101B до плати ESP32 наведена на рисунку 3.1.

					КВРКІ.220018.22.01.33 ПЗ	Арк. 36
Зм.	Арк.	№ докум.	Підпис	Дата		

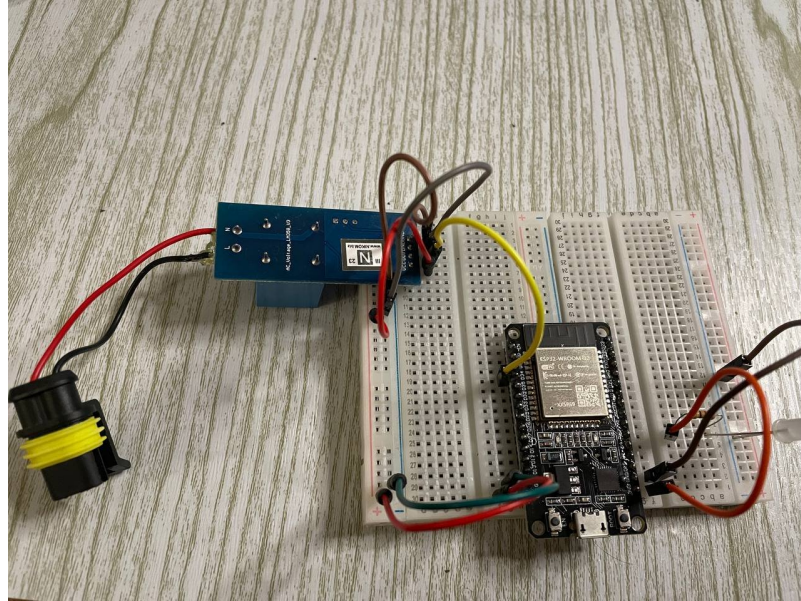


Рисунок 3.1 – Підключення ZMPT101B до ESP32

ZMPT101B живиться від 5 В, має два піни GND. Обидва ці піни повинні бути підключені до загальної землі (GND) з платою ESP32, що дозволить забезпечити коректну роботу датчика. Живлення датчика здійснюється через пін VIN плати ESP32, оскільки він забезпечує необхідну напругу для роботи ZMPT101B. Пін Out датчика підключається до аналогового піна 34 плати ESP32, що дозволяє зчитувати сигнал з датчика.

Для калібрування ZMPT101B застосовується потенціометр, що дозволяє здійснити еталонний вимір, забезпечуючи точність вимірювання змінної напруги. Калібрування є необхідним етапом для коректної роботи датчика в рамках моніторингу енергоспоживання.

Підключення датчика ACS712 для вимірювання струму здійснюється аналогічно. Однак, у цьому випадку було вирішено жити датчик від 3,3 В замість стандартних 5 В. Це рішення було прийнято для того, щоб при нульовому струмі (0 А) рівень вихідної напруги на піні Out склав 3,3 В. Хоча таке підключення знижує точність зчитувань з одного боку, однак, це не впливає на точність розрахунків, таких як визначення струму за допомогою дільника напруги. Пін Out датчика ACS712 підключається до аналогового піна 35 плати ESP32 для зчитування

значень струму. Схема підключення ACS712 до ESP32 показана на рисунку 3.2.

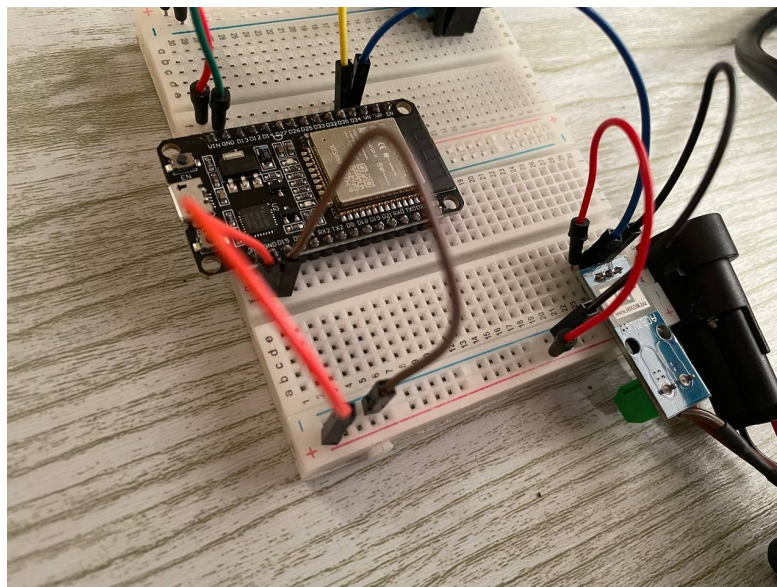


Рисунок 3.2 – Підключення ACS712 до ESP32

Для підключення датчиків до мережі 220 В необхідно коректно визначити кабелі фази та нейтралі в трьохжильному кабелі. Спочатку слід виділити фазу, яку потім підключають до високовольтного входу датчика ACS712. Цей датчик вимірює струм, тому для коректної роботи його потрібно підключати саме до фази мережі. З цією метою фазний провід з трьохжильного кабелю підключається до входу датчика ACS712, а з іншого боку датчика виводиться провід до місця підключення фази в мережевому фільтрі.

Для датчика ZMPT101В підключення здійснюється через конектор, який має два контакти: + і -. Позитивний контакт конектора (+) припаюється до фази мережевого фільтра, а негативний контакт (-) припаюється до нейтралі мережі. Це забезпечує коректну роботу датчика для вимірювання змінної напруги, оскільки він працює на основі підключення до джерела змінної напруги.

Варто зауважити, що кабелі для підключення розраховані на струм до 5 А. Це саме той максимальний струм, який може вимірювати ACS712, оскільки його найбільша допустима робоча величина струму складає 5 А. Таким чином, цей струм є мінімальним для коректної роботи датчика, і вимірювання будуть здійснюватися

в межах цієї величини.

Підключення до мережевого фільтра для обох датчиків (ACS712 та ZMPT101B) показано на рисунку 3.3.

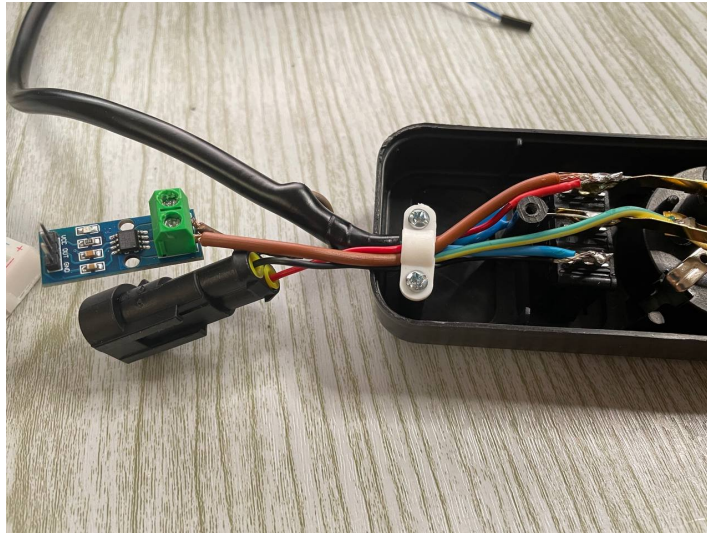


Рисунок 3.3 – Підключення до мережевого фільтру

Підключення всіх компонентів системи, включаючи датчики ACS712 і ZMPT101B, до мережевого фільтра, а також їх взаємодія, показано на реальному зображенні, яке наведено на рисунку 3.4. На фото відображено фізичне підключення всіх елементів системи в єдину конструкцію, що дозволяє візуально оцінити розташування компонентів та їх з'єднання.

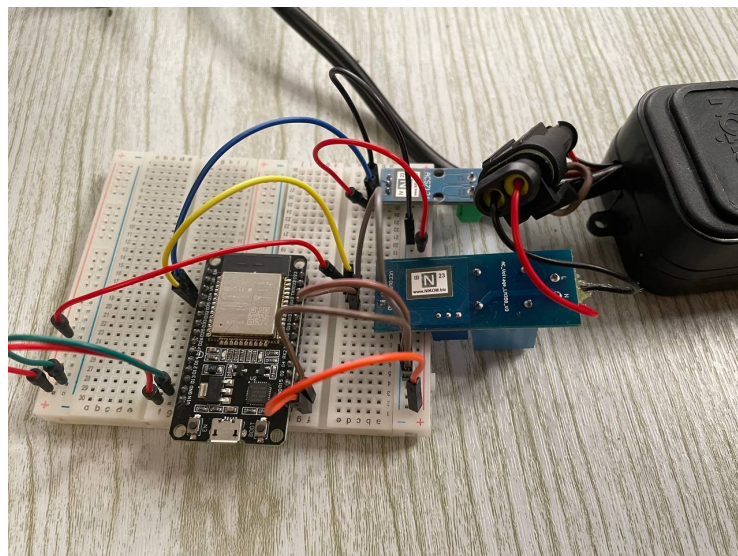


Рисунок 3.4 – Апаратні з'єднання проекту

На рисунку 3.5 зображена схема апаратних з'єднань, побудована в програмному середовищі Fritzing [33].

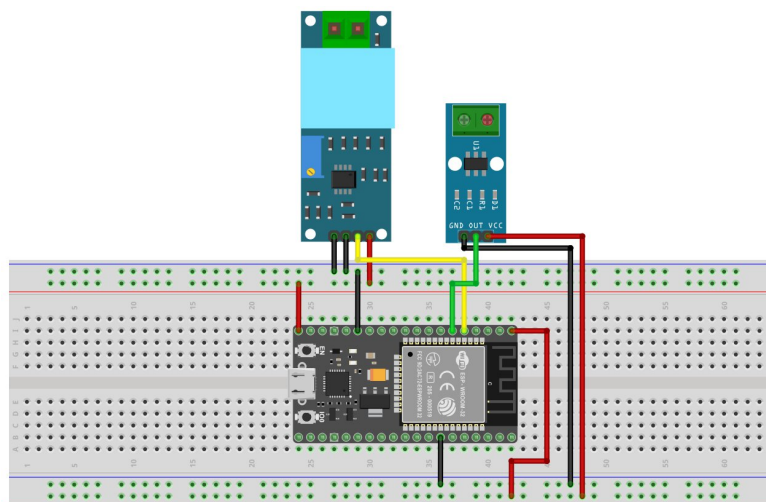


Рисунок 3.5 – Схема апаратних з'єднань

На рисунку 3.6 наведено принципову схему приладу, створену в програмному забезпеченні KiCad [34]. Ця схема демонструє взаємозв'язок всіх компонентів системи, включаючи підключення датчиків, а також їх інтеграцію з платою ESP32.

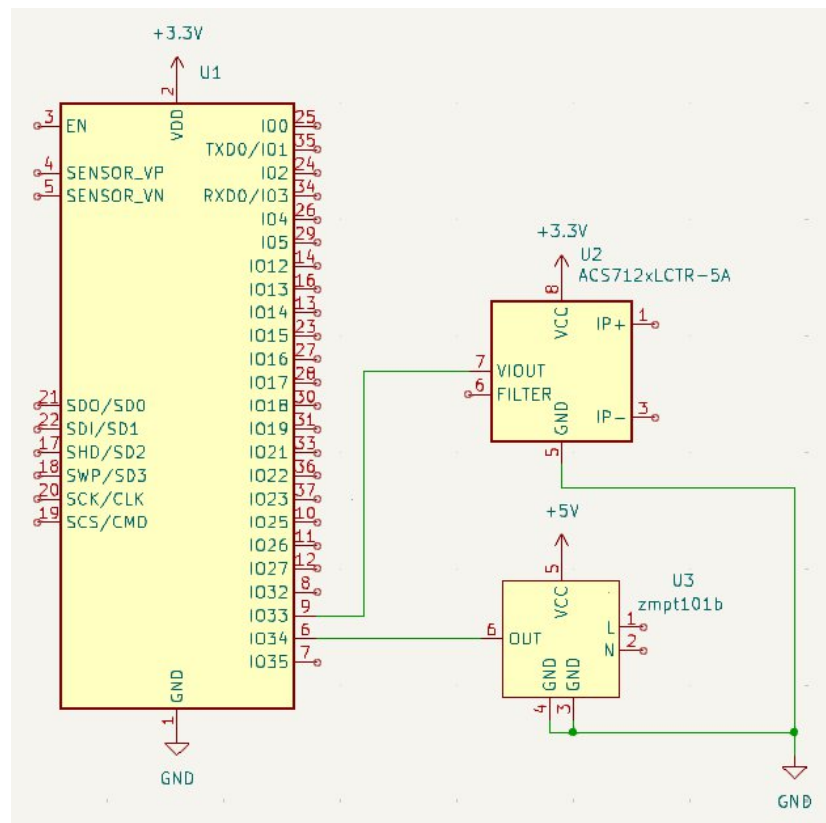


Рисунок 3.6 – Принципова схема приладу

### 3.2 Розробка програмного забезпечення для мікроконтролера та його алгоритмів збору даних

Для реалізації програмного забезпечення на мікроконтролері ESP32, що забезпечує збір і передачу даних з датчиків ACS712 та ZMPT101B на сервер, було використано мову програмування C++ та середовище Arduino IDE. Основне завдання полягає у вимірюванні струму та напруги, розрахунку потужності та відправці цих даних на сервер для подальшого аналізу.

Програмне забезпечення складається з кількох частин, кожна з яких виконує свою конкретну задачу в процесі роботи системи.

Спочатку необхідно підключити потрібні бібліотеки для роботи з датчиками та мережею. Це забезпечує коректне зчитування значень з датчиків і передачу даних через Wi-Fi. Код для підключення бібліотек:

```
#include "ACS712.h"
#include "EmonLib.h"
```

```
#include <WiFi.h>
#include <HTTPClient.h>
```

Після підключення бібліотек, потрібно визначити константи для підключення до бездротової мережі та адреси сервера, на який будуть відправлятися дані. Вони задаються наступним чином:

```
const char* ssid = "SterbenIoT";
const char* password = "zaqwerty123";
const          char*          serverIP          =
"http://192.168.0.108:3000/get_power";
```

Ці значення є статичними і вказують на точку доступу Wi-Fi та сервер, на який буде відправлятися інформація про потужність.

Для роботи з датчиками необхідно створити екземпляри класів для ACS712 [35] і EmonLib. Це дозволяє здійснювати вимірювання відповідно до налаштувань датчиків:

```
ACS712 ACS(35, 3.3, 4095, 185);
EnergyMonitor emon;
```

У цьому випадку:

- ACS712 ACS(35, 3.3, 4095, 185); — створюється об'єкт для датчика струму ACS712, де вказано пін, на якому підключений датчик, напруга живлення, максимальна розрядність і коефіцієнт чутливості.

- EnergyMonitor emon; — створюється об'єкт для енергомоніторингу, що використовує бібліотеку EmonLib.

Функція setup() виконується один раз при запуску мікроконтролера. У ній налаштовуються початкові параметри, підключення до Wi-Fi, ініціалізація датчиків та серійного монітора для відображення результатів.

```
void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
```

					КВРКІ.220018.22.01.33 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

```
    delay(500);  
    Serial.println("Підключення...");  
}
```

У цій функції:

1. Ініціалізується серійний монітор для виведення інформації.
2. Проводиться підключення до Wi-Fi, і чекається на успішне з'єднання.

Після цього виводиться IP-адреса пристрою.

3. Налаштовується роздільна здатність для аналогових зчитувань.
4. Виконується налаштування датчика ACS712 (задання середнього значення).

5. Ініціалізуються налаштування для вимірювання напруги за допомогою бібліотеки EmonLib.

Після налаштування всіх параметрів, реалізовано метод для отримання та розрахунку потужності. Це включає збір даних з датчиків, розрахунок потужності та відправку даних на сервер:

```
String macAddress = WiFi.macAddress();  
String url = String(serverIP) + "?power=" +  
String(power) + "&node=" + macAddress;  
Serial.println("Відправляється URL: " + url);  
HTTPClient http;  
http.begin(url);  
int httpCode = http.GET();
```

У цьому фрагменті:

- Отримується MAC-адреса пристрою, яка використовується для ідентифікації при передачі даних.
- Формується URL для запиту на сервер, куди додаються параметри з потужністю та MAC-адресою.
- Використовується бібліотека HTTPClient для виконання GET-запиту.

Опрацювання відповіді від сервера:

```
if (httpCode > 0) {
```

					КВРКІ.220018.22.01.33 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		

```

String payload = http.getString();
Serial.println("Відповідь сервера: " + payload);
} else {
Serial.println("Помилка відправки запиту: " +
String(httpCode));
}
http.end();

```

У цій частині коду:

- Перевіряється статус відповіді від сервера.
- Якщо відповідь успішна, виводиться текст відповіді на серійний монітор.

Якщо виникає помилка, виводиться код помилки.

Основний цикл програми виконується безперервно і здійснює основні операції - вимірювання значень, розрахунок потужності та відправку результатів на сервер.

```

void loop() {
emon.calcVI(20, 1000);
float voltage = emon.Vrms;
float amperage = getAmperage();
float power = voltage * amperage;
Serial.printf("Power: %.1f\n", power);
sendPowerToServer(power);
delay(1000);
}

```

В цьому циклі:

1. Викликається метод calcVI() [36], який здійснює обчислення значень напруги та струму.
2. Зчитуються значення напруги та струму, після чого розраховується потужність.
3. Потужність виводиться на серійний монітор для відображення.
4. Викликається функція sendPowerToServer(), яка відправляє отриману

					КВРКІ.220018.22.01.33 ПЗ	Арк. 44
Зм.	Арк.	№ докум.	Підпис	Дата		

потужність на сервер.

5. Після кожного вимірювання відбувається затримка в 1 секунду, щоб уникнути надмірного навантаження на систему.

Розрахунок споживаної потужності проводиться безпосередньо на сервері. Такий підхід дозволяє знизити навантаження на ресурси плати ESP32, оскільки обчислення енергоспоживання може вимагати значних обчислювальних потужностей, що може негативно вплинути на ефективність роботи мікроконтролера, особливо при великих обсягах даних.

Після написання коду для плати та її налаштування, було проведено тестування функціональності пристрою. На рисунку 3.7 наведено скріншот консолі з Arduino IDE, на якому відображено результати роботи системи при відключених приладах навантаження.

```
A: 0.0
Power: 0.0
Відправляється URL: http://192.168.0.108:3000/get_power?power=0.00&node=24:DC:C3:9F:C6:1C
Відповідь сервера: Power received from Кімната 1: 0.00 W
A: 0.0
Power: 0.0
Відправляється URL: http://192.168.0.108:3000/get_power?power=0.00&node=24:DC:C3:9F:C6:1C
Відповідь сервера: Power received from Кімната 1: 0.00 W
A: 0.0
Power: 0.0
Відправляється URL: http://192.168.0.108:3000/get_power?power=0.00&node=24:DC:C3:9F:C6:1C
Відповідь сервера: Power received from Кімната 1: 0.00 W
A: 0.0
Power: 0.0
Відправляється URL: http://192.168.0.108:3000/get_power?power=0.00&node=24:DC:C3:9F:C6:1C
Відповідь сервера: Power received from Кімната 1: 0.00 W
```

Рисунок 3.7 – Скріншот консолі з Arduino IDE при відключених пристроях

На рисунку 3.8 зображено результат роботи пристрою з підключеними навантаженнями.

```
A: 2.2
Power: 478.3
Відправляється URL: http://192.168.0.108:3000/get_power?power=478.29&node=24:DC:C3:9F:C6:1C
Відповідь сервера: Power received from Кімната 1: 478.29 W
A: 2.2
Power: 464.2
Відправляється URL: http://192.168.0.108:3000/get_power?power=464.19&node=24:DC:C3:9F:C6:1C
Відповідь сервера: Power received from Кімната 1: 464.19 W
A: 2.2
Power: 461.6
Відправляється URL: http://192.168.0.108:3000/get_power?power=461.60&node=24:DC:C3:9F:C6:1C
Відповідь сервера: Power received from Кімната 1: 461.60 W
A: 2.2
Power: 468.8
Відправляється URL: http://192.168.0.108:3000/get_power?power=468.84&node=24:DC:C3:9F:C6:1C
Відповідь сервера: Power received from Кімната 1: 468.84 W
```

Рисунок 3.8 – Скріншот консолі з Arduino IDE з підключеними пристроями

Як видно з результатів на екрані, розрахунок потужності показав середнє значення 465 Вт, що підтверджує правильність роботи системи і коректність вимірювань. Похибка складає 7%, оскільки прилад заявлено, що споживає 0,5 кВт. Однак така похибка є допустимою і може бути в межах похибки самого приладу, що споживає енергію. Це є нормальним для побутових приладів і не впливає суттєво на загальну точність вимірювання.

### 3.3 Розробка програмного забезпечення для веб-інтерфейсу моніторингу енергоспоживання

Програмне забезпечення для серверної частини веб-інтерфейсу створене на основі фреймворка Express.js. Основна мета цього програмного забезпечення - прийом даних від пристроїв (мікроконтролерів), збереження їх у пам'яті та надання можливості отримання цих даних через API [37].

На початковому етапі підключаються необхідні бібліотеки для роботи серверу:

```
const express = require('express');
const fs = require('fs');
const cors = require('cors');
const app = express();
const port = 3000;
```

- express: фреймворк для створення серверів на Node.js.

- fs: стандартна бібліотека для роботи з файловою системою, використовується для збереження даних.

- cors: бібліотека для налаштування CORS (Cross-Origin Resource Sharing) [38], що дозволяє налаштовувати доступ до серверу з різних доменів.

Для зручності управління даними пристроїв та їхнього збереження у файлі, визначаються такі константи:

```
const devicesFile = 'devices.json';  
let devices = [];  
let devicesMap = loadDevicesMap();
```

- devicesFile: шлях до файлу, в якому зберігаються дані про пристрої.  
- devices: масив, що містить інформацію про підключені пристрої.  
- devicesMap: об'єкт, який зберігає зв'язок між MAC-адресами пристроїв і їхніми іменами. Ініціалізується через функцію loadDevicesMap(), що завантажує інформацію з файлу або встановлює порожній об'єкт.

Для того, щоб веб-клієнти могли здійснювати запити до серверу без обмежень, налаштовується CORS:

```
app.use(cors());
```

Це дозволяє виконувати запити з будь-якого джерела, що є необхідним для веб-клієнта, розміщеного на окремому сервері.

Наступним кроком реалізуються маршрути для отримання даних про потужність та список підключених пристроїв.

Цей маршрут обробляє запити на /get\_power, отримуючи потужність та MAC-адресу пристрою [39] з параметрів запиту. Якщо значення потужності та MAC-адреси отримано, здійснюється пошук пристрою в масиві devices. Якщо пристрій ще не зареєстрований, додається новий елемент у список, а для кожного пристрою оновлюються показники потужності та час останньої активності. Після цього сервер відповідає повідомленням, що потужність отримано від певного пристрою.

Маршрут для отримання списку пристроїв та значень потужності:

```
app.get('/get_devices', (req, res) => {  
    res.json(devices);
```

					КВРКІ.220018.22.01.33 ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		

```
});
```

Цей маршрут обробляє запити на `/get_devices`, повертаючи список усіх підключених пристроїв у форматі JSON [40].

Для забезпечення актуальності даних про пристрої на сервері, створено функцію, яка видаляє пристрої, що не надсилали дані протягом певного часу:

```
function removeInactiveDevices() {
    const currentTime = Date.now();
    devices = devices.filter(device => {
        if (currentTime - device.lastActive <= 10000)
        {
            return true;
        } else {
            console.log(`Пристрій ${device.node}
            (${device.mac}) видалено зі списку активних.`);
            return false;
        }
    });
}
```

Ця функція перевіряє час останньої активності кожного пристрою. Якщо пристрій не оновлював дані більше ніж 10 секунд (10000 мс), він видаляється зі списку активних пристроїв.

Для забезпечення постійної перевірки підключених пристроїв і автоматичного видалення неактивних, налаштовується періодичний виклик функції `removeInactiveDevices`:

```
setInterval(removeInactiveDevices, 1000);
```

Це забезпечує виклик функції кожну секунду, що дозволяє актуалізувати список підключених пристроїв.

Для запуску самого серверу використовується команда:

```
app.listen(port, () => {
    console.log(`Сервер запущено на
```

					КВРКІ.220018.22.01.33 ПЗ	Арк. 48
Зм.	Арк.	№ докум.	Підпис	Дата		

```
http://localhost:${port}`);  
});
```

Це забезпечує запуск серверу на порту 3000, де він буде доступний для прийому запитів.

Створена серверна частина веб-інтерфейсу моніторингу енергоспоживання включає маршрути для отримання даних про потужність та список підключених пристроїв. Всі дані зберігаються на сервері, а інформація про неактивні пристрої видається автоматично. Використання таких технологій, як Express.js та CORS, дозволяє працювати з клієнтською частиною та зберігати актуальність даних без перевантаження серверу.

### 3.4 Взаємодія між апаратною та програмною частинами

Взаємодія між апаратною та програмною частинами реалізована через HTTP-запити по локальній мережі. Для отримання даних про пристрої веб-клієнт використовує маршрут `/get_devices`, який був описаний у попередньому розділі.

Функція, яка відповідає за отримання даних про підключені пристрої та оновлення інформації на веб-сторінці, виглядає наступним чином.

Спочатку виконується HTTP-запит за допомогою методу `fetch`. Він надсилає запит на сервер за адресою `http://localhost:3000/get_devices`, щоб отримати список активних пристроїв:

```
fetch('http://localhost:3000/get_devices')
```

Після цього сервер повертає дані у форматі JSON, і вони передаються в блок обробки. За допомогою методу `.json()` відповіді ці дані перетворюються на JavaScript-об'єкти, що дає можливість працювати з ними у зручному форматі:

```
.then(response => response.json())
```

Далі, після того як отримано список пристроїв, дані обробляються в циклі `forEach`. Для кожного пристрою виводиться в консоль його ім'я (`node`), MAC-адреса і потужність:

```
data.forEach(device => {
```

					КВРКІ.220018.22.01.33 ПЗ	Арк. 49
Зм.	Арк.	№ докум.	Підпис	Дата		

```
        console.log(`Пристрій:      ${device.node},      MAC:
${device.mac}, Потужність: ${device.power} W`);
    });
```

Якщо пристрій ще не додано до об'єкта `devicesData`, створюється новий запис для цього пристрою. У цьому записі містяться його MAC-адреса, ім'я (`node`), потужність, загальна енергія, максимальна потужність та кількість вимірювань. Якщо пристрій вже є в об'єкті, оновлюється його потужність, і перевіряється, чи не перевищує поточна потужність попередній максимум.

Після цього викликається функція `updateCharts()`, яка оновлює графічне відображення даних на веб-сторінці. Це забезпечує відображення поточної інформації про пристрої та їх потужність:

```
updateCharts();
```

В разі виникнення помилок при виконанні запиту до серверу (наприклад, якщо сервер недоступний), обробка помилок виконується за допомогою методу `.catch`, і помилка виводиться в консоль:

```
.catch(error => console.error("Помилка запиту:",
error));
```

Завдяки такій структурі функції, веб-клієнт отримує актуальні дані про пристрої та їх стан. Ці дані використовуються для оновлення інформації на сторінці, що дозволяє користувачеві отримувати актуальну інформацію про енергоспоживання.

### 3.5 Проведення тестування системи та аналіз отриманих результатів

Тестування системи проводилося в реальних умовах, з метою перевірки її працездатності та точності вимірювань. На початковому етапі було здійснено перевірку відображення даних на веб-сторінці, коли система працює без підключеного навантаження. У цьому випадку на веб-сторінці з'явилися два графіки: перший графік показував значення потужності, яка була зафіксована на момент тестування, і це значення дорівнювало нулю (рисунком 3.9), оскільки жоден

					КВРКІ.220018.22.01.33 ПЗ	Арк. 50
Зм.	Арк.	№ докум.	Підпис	Дата		

споживач енергії не був підключений. Другий графік відображав спожиту енергію в кіловат-годинах (кВт·год), що також було рівне нулю (рисунок 3.10), оскільки не було споживаної енергії.

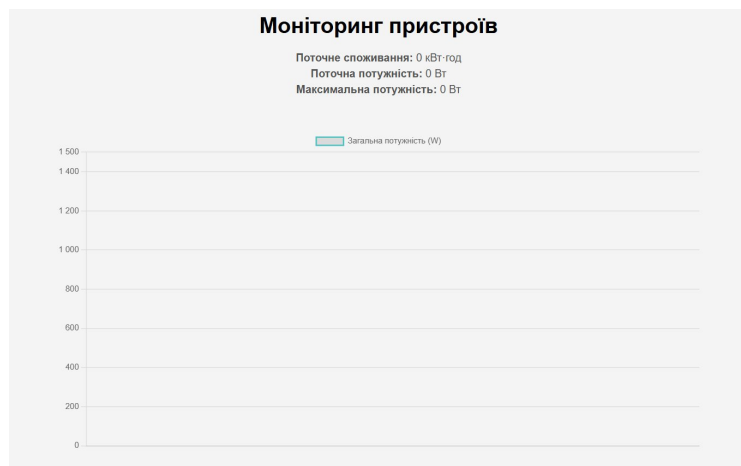


Рисунок 3.9 – Графік потужності без споживача

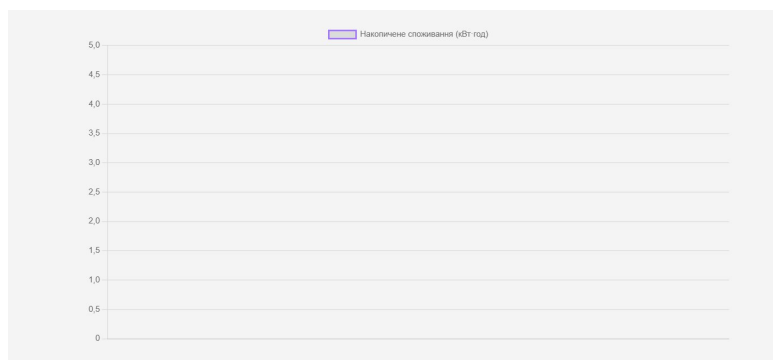


Рисунок 3.10 – Графік спожитої електроенергії без споживача

Після цього для подальшої перевірки функціональності системи було підключено навантаження - нагрівальний елемент, заявлена потужність якого становить 1 кВт. При цьому на веб-сторінці на першому графіку почала відображатися потужність, яку споживав підключений пристрій. Як видно з результатів на рисунку 3.11, в середньому, споживана потужність склала 820 Вт, що є меншою за заявлену потужність 1 кВт, але це може бути пов'язано з характеристиками самого нагрівального елемента. Другий графік (рисунок 3.12), який відповідає за відображення спожитої енергії, почав показувати загальну кількість енергії, що була витрачена пристроєм.



практично збігається із середнім показником 820 Вт, зафіксованим системою, що дає похибку приблизно 1%. Така точність вимірювання підтверджує коректність роботи пристрою.

Далі виконується розрахунок спожитої енергії. Відомо, що навантаження працювало протягом 80 секунд. Спожита енергія визначається за формулою 3.2.

$$E = P \times t \quad (3.2)$$

де  $E$  – енергія у ват-секундах (Вт/с),

$t$  – час у секундах.

Підставляючи розраховану потужність отримано 64960 Вт/с. Оскільки 1 кВт/година дорівнює 3600000 Вт/с, перетворимо отримане значення й отримаємо 0.018 кВт/год.

Порівняння з даними, отриманими системою, показує, що на веб-інтерфейсі зафіксоване значення 0,016 кВт/год. Відхилення становить 0,002 кВт/год, що є допустимою похибкою вимірювань

### 3.6. Висновки

У даному розділі було розроблено та реалізовано апаратну частину пристрою для моніторингу енергоспоживання, створено програмне забезпечення для мікроконтролера та сервера, а також забезпечено їхню взаємодію через HTTP-запити у локальній мережі.

Було протестовано коректність роботи системи шляхом вимірювання потужності та порівняння отриманих значень із розрахунковими даними. Виявлена похибка знаходиться в допустимих межах, що підтверджує правильність функціонування пристрою.

Також було розроблено веб-інтерфейс для візуалізації отриманих даних, що дозволяє в реальному часі контролювати потужність і спожиту енергію підключених пристроїв.

					КВРКІ.220018.22.01.33 ПЗ	Арк. 53
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

В ході виконання дослідження кваліфікаційної роботи на тему "Система моніторингу енерговитрат у кіберфізичній системі «Розумний будинок» на Arduino" було проведено комплексне вивчення методів збору, обробки та аналізу даних про енергоспоживання, розроблено апаратно-програмний комплекс, реалізовано алгоритми вимірювання та відображення даних, а також виконано тестування створеної системи.

У першому розділі здійснено аналіз концепції «Розумного будинку» та його енергоефективності. Було розглянуто існуючі рішення для контролю електроспоживання, проведено порівняння їх функціональних можливостей, точності вимірювань та варіантів реалізації. Детально проаналізовано апаратні та програмні платформи, які використовуються для побудови систем моніторингу енергоспоживання. Розглянуто методи збору, обробки та візуалізації інформації. На основі отриманих даних сформульовано вимоги до системи, що проєктується, та визначено її технічні характеристики.

У другому розділі проведено проєктування апаратної та програмної частини системи моніторингу. Апаратна частина базується на мікроконтролері ESP-Wroom-32, який забезпечує обробку та передачу даних. Для вимірювання напруги використано датчик ZMPT101B, а для вимірювання сили струму – сенсор ACS712. Дані з цих датчиків передаються на мікроконтролер, де відбувається їх первинна обробка.

Програмна частина включає серверну та клієнтську складові. Серверна частина реалізована на основі Node.js з використанням фреймворку Express.js, що забезпечує прийом, обробку та зберігання отриманих даних. Веб-інтерфейс системи дозволяє візуалізувати параметри електроспоживання в режимі реального часу.

Було розроблено структурну схему взаємодії компонентів, описано алгоритм функціонування системи та визначено механізми передачі даних через локальну мережу.

					КВРКІ.220018.22.01.33 ПЗ	Арк. 54
Зм.	Арк.	№ докум.	Підпис	Дата		

У третьому розділі виконано реалізацію системи, її тестування та аналіз отриманих результатів. Розроблена апаратна частина успішно інтегрована з програмним забезпеченням, що дозволило реалізувати функціональність збору та передачі інформації про енергоспоживання.

Система була протестована у реальних умовах. Початкові випробування проводилися без підключених навантажень, що дозволило перевірити коректність роботи веб-інтерфейсу та правильність відображення нульових значень споживаної потужності. Подальше тестування проводилося із підключенням навантаження – нагрівального елемента із заявленою потужністю 1 кВт.

Аналіз отриманих даних показав, що система визначила середнє значення потужності 820 Вт, а розрахункове значення на основі вимірної напруги (232 В RMS) та сили струму (3,5 А) склало 812 Вт. Таким чином, було підтверджено точність вимірювань із похибкою 1%.

Додатково було проведено розрахунок спожитої енергії за певний період часу. Пристрій працював 80 секунд, протягом яких система зафіксувала споживання 0,016 кВт·год, що відповідає розрахунковим теоретичним значенням.

Розроблена система є доступним рішенням для моніторингу енергоспоживання у кіберфізичних системах «Розумний будинок». Вона дозволяє в режимі реального часу отримувати інформацію про потужність підключених пристроїв, аналізувати енергоспоживання та приймати рішення щодо його оптимізації.

Система може бути використана як у житлових, так і в промислових приміщеннях для контролю споживання електроенергії, що сприяє підвищенню енергоефективності та зниженню витрат на електропостачання. Реалізоване рішення дозволяє масштабувати систему за рахунок підключення додаткових датчиків та впровадження аналітичних механізмів для прогнозування та автоматизованого керування енергоспоживанням.

					КВРКІ.220018.22.01.33 ПЗ	Арк. 55
Зм.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Оленич І., Павлик М., Марцінів М. Система прогнозування потужності сонячної батареї розумного будинку, м.Київ, 2022р. 123с. , URL: <https://ela.kpi.ua/server/api/core/bitstreams/8aa8adf4-e2de-42d7-81bd-403716bc7d28/content> (дата звернення 20.05.2025).
2. Розумний будинок. Smarthome. URL:<https://ikstelecom.net.ua/smarthome.html> (дата звернення 20.05.2025).
3. Manisha P. , Ajij D. IOT Base Home Automation, International Journal for Multidisciplinary Research, 2024, 9с. URL: <http://ijfmr.com/papers/2024/1/12996.pdf> (дата звернення 20.05.2025).
4. LPWAN, LoRa, LoRaWAN та Internet of Things. iotji. URL: <https://iotji.io/osoblyvosti-lorawan> (дата звернення 20.05.2025).
5. Shelly EM. Shelly Knowledge Base. URL: <https://kb.shelly.cloud/knowledge-base/shelly-em>. (дата звернення 20.05.2025).
6. Sonoff POWR3. SONOFF. URL: <https://sonoff.tech/product/diy-smart-switches/powr3> (дата звернення 20.05.2025).
7. Свєрдлюк Б. Home Assistant. Посібник для початківців. м.Київ, 2022, DOU. URL: <https://dou.ua/forums/topic/38947> (дата звернення 20.05.2025).
8. Modbus BACnet to MQTT IBMS ndustrial IoT Gateway. IoT Solution. URL: <https://www.iot-solution.com/industrial-iot-gateway-p00424p1.html> (дата звернення 20.05.2025).
9. Mahajan A. Getting started with Arduino. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/getting-started-with-arduino/> (дата звернення 20.05.2025).
10. Difference between ESP32 and ESP8266 Modules. ElProCus. URL: <https://www.elprocus.com/difference-between-esp32-and-esp8266> (дата звернення 20.05.2025).
11. Raspberry Pi Hardware. Raspberry Pi. URL: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html> (дата

					КВРКІ.220018.22.01.33 ПЗ	Арк. 56
Зм.	Арк.	№ докум.	Підпис	Дата		

звернення 20.05.2025).

12. PostgreSQL vs SQLite. Astera. URL: <https://www.astera.com/knowledge-center/postgresql-vs-sqlite> (дата звернення 20.05.2025).

13. Логічний контролер Modicon M221. SE URL: <https://www.se.com/ua/uk/product-range/62128-логічний-контролер-modicon-m221/> (дата звернення 20.05.2025).

14. Hasan M. The IoT Cloud. IoT Analytics. URL: <https://iot-analytics.com/iot-cloud> (дата звернення 20.05.2025).

15. Express Web Framework. MDN Web Docs. URL: [https://developer.mozilla.org/enUS/docs/Learn\\_web\\_development/Extensions/Server-side/Express\\_Nodejs](https://developer.mozilla.org/enUS/docs/Learn_web_development/Extensions/Server-side/Express_Nodejs) (дата звернення 20.05.2025).

16. Шибецький Б., Дорогий Я. Система моніторингу та аналізу споживання електроенергії в домогосподарствах на основі IoT-технологій. Наукові праці Донецького національного технічного університету. Серія: «Комп'ютерна інженерія та автоматизація». 2024. 134с. URL: <https://otanew.donntu.edu.ua/article/view/316438/307085> (дата звернення 22.05.2025).

17. Серета Ю.І. Засоби збору та візуалізації даних в системах управління. м. Тернопіль. 2020. 74с. URL: <http://dSPACE.wunu.edu.ua/bitstream/316497/40522/3/Серета%20МР%202020.pdf> (дата звернення 22.05.2025).

18. Полоус В.Ю. Інтелектуальна комп'ютерна система аналізу та прогнозування енергоспоживання. м.Харків. 2023. 66с. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/ae44ba9f-ed80-459a-b634-62580dd306eb/content> (дата звернення 22.05.2025).

19. ESP32–WROOM–32 Datasheet. Espressif Systems. URL: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf) (дата звернення 22.05.2025).

20. Introduction to ESP32. ElectronicWings. URL: <https://www.electronicwings.com/esp32/introduction-to-esp32> (дата звернення 22.05.2025).

					КВРКІ.220018.22.01.33 ПЗ	Арк. 57
Зм.	Арк.	№ докум.	Підпис	Дата		

21. Терещенко Т.О. Розподілені мікропроцесорні системи: Конспект лекцій. Київ. 2018. 192 с. URL: [https://eds.kpi.ua/wpcontent/uploads/2017/12/Конспект\\_лекцій-РМС-Б1.pdf](https://eds.kpi.ua/wpcontent/uploads/2017/12/Конспект_лекцій-РМС-Б1.pdf) (дата звернення 22.05.2025).

22. Датчик напруги ZMPT101B. IT Master. URL: <https://itmaster.biz.ua/directory/sensors/zmpt101b.html> (дата звернення 22.05.2025).

23. Що таке ефект Холла. FMUSER. URL: <https://uk.fmuser.net/content/?21020.html> (дата звернення 23.05.2025).

24. ACS712 current sensor module. Components101. URL: <https://components101.com/sensors/acs712-current-sensor-module> (дата звернення 23.05.2025).

25. Rabert T. What is voltage sensor. ForumElectrical. URL: <https://forumelectrical.com/what-is-a-voltage-sensor/> (дата звернення 23.05.2025).

26. Software. Arduino. URL: <https://www.arduino.cc/en/software> (дата звернення 23.05.2025).

27. Tillaart R. Arduino library for ACS current sensor. GitHub. URL: <https://github.com/RobTillaart/ACS712> (дата звернення 23.05.2025).

28. Electricity monitoring library. GitHub. URL: <https://github.com/openenergymonitor/EmonLib> (дата звернення 23.05.2025).

29. HttpClient. Arduino. URL: <https://docs.arduino.cc/libraries/httpclient/> (дата звернення 24.05.2025).

30. Василик І. HTML, CSS і JavaScript: основи веб-розробки. PRAVDA.IF.UA., м.Київ. 2023. 15с. URL: <https://pravda.if.ua/html-css-i-javascript-osnovy-veb-rozrobky/> (дата звернення 24.05.2025).

31. Зеленський К.К. Система керування внутрішнім середовищем розумного будинку на базі мікроконтролерів: магістерська робота. Тернопіль. 2024. 95 с. URL: [https://elartu.tntu.edu.ua/bitstream/lib/44850/1/2024\\_KRM\\_SNnm-61\\_Zelenskiy%20K.K.pdf](https://elartu.tntu.edu.ua/bitstream/lib/44850/1/2024_KRM_SNnm-61_Zelenskiy%20K.K.pdf) (дата звернення 24.05.2025).

32. HTTP Методи запиту. W3Schools українською. URL: [https://w3schoolsua.github.io/tags/ref\\_httpmethods.html#gsc.tab=0](https://w3schoolsua.github.io/tags/ref_httpmethods.html#gsc.tab=0) (дата звернення

					КВРКІ.220018.22.01.33 ПЗ	Арк. 58
Зм.	Арк.	№ докум.	Підпис	Дата		

24.05.2025).

33. Download and start. Fritzing. URL: <https://fritzing.org/> (дата звернення 24.05.2025).

34. Schematic capture & PCB design software. KiCad EDA. URL: <https://www.kicad.org/> (дата звернення 24.05.2025).

35. ACS712 with the ESP32 and Home Assistant. Arduino Forum. URL: <https://forum.arduino.cc/t/acs712-with-the-esp32-and-home-assistant/1280015> (дата звернення 24.05.2025).

36. AC voltage monitor faster approach with emonlib. Arduino Forum. URL: <https://forum.arduino.cc/t/ac-voltage-monitor-faster-approach-with-emonlib/947808> (дата звернення 24.05.2025).

37. What is an API (application programming interface). GeeksforGeeks. URL: <https://www.geeksforgeeks.org/what-is-an-api/> (дата звернення 24.05.2025).

38. CrossOrigin resource sharing. MDN Web Docs. URL: <https://developer.mozilla.org/enUS/docs/Web/HTTP/CORS> (дата звернення 24.05.2025).

39. Що таке MAC-адреса і для чого вона потрібна. Volia. URL: [https://my.volia.com/kyiv/uk/faq/article/chto-takoe-mac-adres-i-dlya-chego-on-nuzhen?partner=organic\\_search&utm\\_source=google&utm\\_medium=organic](https://my.volia.com/kyiv/uk/faq/article/chto-takoe-mac-adres-i-dlya-chego-on-nuzhen?partner=organic_search&utm_source=google&utm_medium=organic) (дата звернення 24.05.2025).

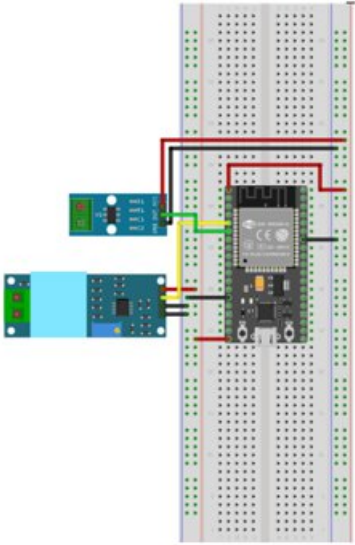
40. Introducing JSON. JSON. URL: <https://www.json.org/json-en.html> (дата звернення 24.05.2025).

					КВРКІ.220018.22.01.33 ПЗ	Арк. 59
Зм.	Арк.	№ докум.	Підпис	Дата		

# Додаток А (обов'язковий)

## КОПІЯ КРЕСЛЕННЯ «СХЕМИ АПАРАТНИХ З'ЄДНАНЬ»

Копія креслення «Схеми апаратних з'єднань»



КвРКІ 220018.22.01.33 Е8

№ п/п	№ документа	Назва документа	Дата	Вид
1	КвРКІ 220018.22.01.33 Е8	Схеми апаратних з'єднань	2018	Схематична
2	КвРКІ 220018.22.01.33 Е8	Схеми апаратних з'єднань	2018	Схематична
3	КвРКІ 220018.22.01.33 Е8	Схеми апаратних з'єднань	2018	Схематична
4	КвРКІ 220018.22.01.33 Е8	Схеми апаратних з'єднань	2018	Схематична
5	КвРКІ 220018.22.01.33 Е8	Схеми апаратних з'єднань	2018	Схематична
6	КвРКІ 220018.22.01.33 Е8	Схеми апаратних з'єднань	2018	Схематична
7	КвРКІ 220018.22.01.33 Е8	Схеми апаратних з'єднань	2018	Схематична
8	КвРКІ 220018.22.01.33 Е8	Схеми апаратних з'єднань	2018	Схематична
9	КвРКІ 220018.22.01.33 Е8	Схеми апаратних з'єднань	2018	Схематична
10	КвРКІ 220018.22.01.33 Е8	Схеми апаратних з'єднань	2018	Схематична

КвРКІ 220018.22.01.33 Е8

Схеми апаратних з'єднань

ХНУ/КІС-22-1

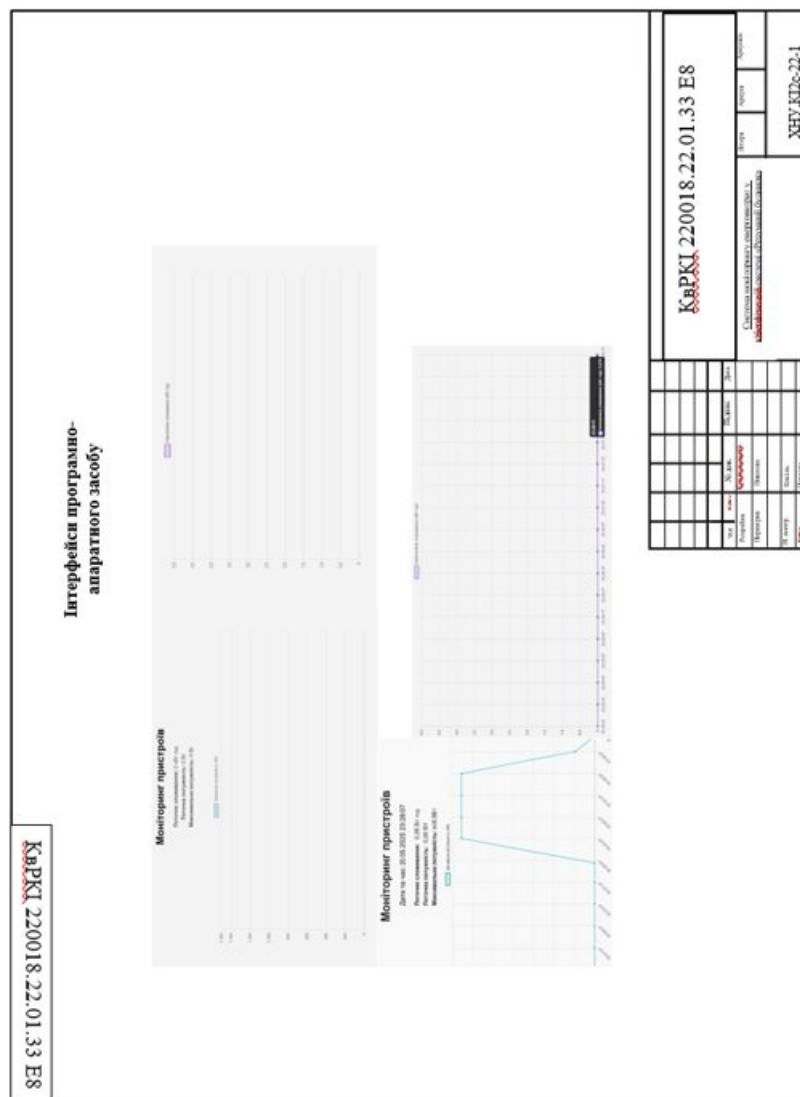
# Додаток Б (обов'язковий)

## КОПІЯ КРЕСЛЕННЯ «БЛОК-СХЕМИ ПРОГРАМ»



## Додаток В (обов'язковий)

### КОПІЯ КРЕСЛЕННЯ «ІНТЕРФЕЙСИ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ»



## Додаток Г (обов'язковий)

### ЛІСТИНГ КОДУ ДЛЯ НАДСИЛАННЯ ДАНИХ ДО СЕРВЕРА

```
#include "ACS712.h"
#include "EmonLib.h"
#include <WiFi.h>
#include <HTTPClient.h>

const char* ssid = "SterbenIoT";
const char* password = "zaqwerty123";

const char* serverIP = "http://192.168.0.108:3000/get_power";

ACS712 ACS(35, 3.3, 4095, 185);
EnergyMonitor emon;

// Функція для отримання значення струму
float getAmperage() {
    float average = 0;
    for (int i = 0; i < 100; i++) {
        average += ACS.mA_AC();
    }
    float A = (average / 100.0) / 1000.0;
    Serial.printf("A: %.1f\n", A);
    return max(0.0f, A - 0.2f);
}

void sendPowerToServer(float power) {
```

```

String macAddress = WiFi.macAddress();
String url = String(serverIP) + "?power=" + String(power) + "&node=" +
macAddress;
Serial.println("Відправляється URL: " + url);

HTTPClient http;
http.begin(url);

int httpCode = http.GET();

if (httpCode > 0) {
    String payload = http.getString();
    Serial.println("Відповідь сервера: " + payload);
} else {
    Serial.println("Помилка відправки запиту: " + String(httpCode));
}

http.end();
}

void setup() {
    Serial.begin(115200);

    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.println("Підключення...");
    }
    Serial.println("Підключено!");
}

```

```
Serial.print("IP-адреса: ");
Serial.println(WiFi.localIP());

analogReadResolution(12);
ACS.setMidPoint(2048);
emon.voltage(34, 135, 1.7);

delay(2000);
}

void loop() {
  emon.calcVI(20, 1000);
  float voltage = emon.Vrms;
  float amperage = getAmperage();
  float power = voltage * amperage;

  Serial.printf("V: %.2f\n", voltage);
  Serial.printf("Power: %.1f\n", power);

  sendPowerToServer(power);

  delay(1000);
}
```

## Додаток Д

(ОБОВ'ЯЗКОВИЙ)

## ЛІСТИНГ КОДУ СЕРВЕРА

```
const express = require('express');
const fs = require('fs');
const cors = require('cors');
const app = express();
const port = 3000;

const devicesFile = 'devices.json';
let devices = [];

function loadDevicesMap() {
  if (fs.existsSync(devicesFile)) {
    return JSON.parse(fs.readFileSync(devicesFile, 'utf8'));
  }
  return {};
}

function saveDevicesMap(devicesMap) {
  fs.writeFileSync(devicesFile, JSON.stringify(devicesMap, null, 2));
}

let devicesMap = loadDevicesMap();

function removeInactiveDevices() {
  const currentTime = Date.now();
  devices = devices.filter(device => {
    if (currentTime - device.lastActive <= 10000) {
```

```
    return true;
  } else {
    console.log(`Пристрій ${device.node} (${device.mac}) видалено зі списку
активних.`);
    return false;
  }
});
}
```

```
app.use(cors());
```

```
app.get('/get_power', (req, res) => {
```

```
  const power = req.query.power;
```

```
  const mac = req.query.node;
```

```
  if (power && mac) {
```

```
    let nodeName = devicesMap[mac];
```

```
    if (!nodeName) {
```

```
      nodeName = `Node${Object.keys(devicesMap).length + 1}`;
```

```
      devicesMap[mac] = nodeName;
```

```
      saveDevicesMap(devicesMap);
```

```
      console.log(`Додано новий пристрій: ${nodeName} (${mac})`);
```

```
    }
```

```
    let device = devices.find(d => d.mac === mac);
```

```
    if (!device) {
```

```
      devices.push({ mac, node: nodeName, power: power, lastActive: Date.now() });
```

```
    } else {
```

```

    device.power = power;
    device.lastActive = Date.now();
  }
  console.log("Список підключених пристроїв:", devices);
  console.log(`Отримано потужність від ${nodeName}: ${power} W`);
  res.send(`Power received from ${nodeName}: ${power} W`);
} else {
  res.send('Power or MAC address not received!');
}
});

app.get('/get_devices', (req, res) => {
  res.json(devices);
});

app.post('/rename_device', express.json(), (req, res) => {
  const { node, new_name } = req.body;

  console.log(node);
  console.log(new_name);

  if (node && new_name) {
    let deviceMac = Object.keys(devicesMap).find(mac => devicesMap[mac] ===
node);

    if (deviceMac) {
      devicesMap[deviceMac] = new_name;

      let device = devices.find(d => d.mac === deviceMac);

```

## Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Роман КРАЧКОВСЬКИЙ

Співавтор:

Назва: Крачковський\_Система моніторингу енерговитрат у кіберфізичній системі «Розумний будинок»

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:1.8%

Коефіцієнт подібності 2:0.4%

Мікропробіли: 57

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-02 11:16:35.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2025-06-02

Дата



Доцент Андрій Нічепорук

експерт

а

# Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 0.0%

Dictionaries check: en\_US, ru\_RU, ua\_UA. Errors in the documents: 11%

ID: 242772 Title: БКР Система моніторингу енерговитрат у кіберфізичній системі «Розумний будинок» Added in a DB: 2025-06-02 Authors: Роман КРАЧКОВСЬКИЙ Heads: Ольга ПАВЛОВА Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	74632	640	1258 (2%)	26 (4%)

## Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Крачковський Роман Андрійович

Тема: Система моніторингу енерговитрат у кіберфізичній системі «Розумний будинок» на Arduino

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 61

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розробка системи моніторингу енерговитрат у кіберфізичній системі «Розумний будинок»

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи проведено дослідження предметної області (проаналізовано системи моніторингу енерговитрат у кіберфізичних системах) та виконано постановку задачі дослідження. В другому розділі кваліфікаційної роботи проведено вибір компонентів та апаратно-програмне середовище для виконання завдання. В третьому розділі кваліфікаційної роботи виконано апаратну реалізацію системи моніторингу енерговитрат у кіберфізичній системі «Розумний будинок» на Arduino.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатня увага приділена деталізації при розробці системи

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному технічному рівні.

8. Інші зауваження: \_\_\_\_\_

9. Оцінка дипломної роботи: задовільно

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Верратюк

Леонід Петрович, д. фіз-мат. наук, професор.

зав. каф. ІІЗ

"04" серпня 2025 р.

 (підпис)

Завідувачу кафедри КІС  
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Романа КРАЧКОВСЬКОГО

---

ІІБ здобувача вищої освіти


ФІТ, 3 курсу, групи КІ2с-22-1

### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

 2025 року

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ**  
**КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ**  
**ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система моніторингу енерговитрат у кіберфізичній системі «Розумний будинок»

Автор: Роман КРАЧКОВСЬКИЙ

Спеціальність: 123– Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Ольга ПАВЛОВА, д.т.н, проф.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
- 4) в якості запозичень в окремих місцях системою зафіксовано послідовності чотирьохрозрядних двійкових кодів, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
- 5) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 2.67% і адресується до 25 першоджерела; та системою Anti-Plagiarism складає 11%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІІС



Ольга ПАВЛОВА

Андрій НІЧЕПОРУК

Ольга ПАВЛОВА

