НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

Завідувач кафедри
Комп'ютерних систем, мереж
і кібербезпеки
_____ / Касаткін Д.Ю., к.пед.н., доц./
(підпис) (ПІБ, вчене звання і ступінь)

«»_____2025 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

На тему: «Система моніторингу енергоспоживання приватного будинку»

Спеціальність – 123 «Комп’ютерна інженерія»

Гарант освітньої програми

к. ф.-м. н., доц. Нікітенко Є.В.
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Керівник бакалаврської кваліфікаційної роботи

к. тех. н., доц. Волошин С. М.
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Виконав _____ Кущенко В.К.
(підпис) (ПІБ студента)

КИЇВ-2025
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ
УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ _____ _____
(науковий ступінь, вчене звання) (підпис) (ПІБ)

“____” _____20_____р.

З А В Д А Н Н Я
на виконання бакалаврської кваліфікаційної роботи студенту

Кущенко Валентину Костянтиновичу
(прізвище, ім’я, по батькові)

Спеціальність (напрям підготовки) Комп’ютерна інженерія

Тема випускної бакалаврської роботи Система моніторингу енергоспоживання приватного будинку

керівник проекту (роботи) Волошин С.М., к.тех.н., доц.
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджена наказом ректора НУБіП України від “_____”_____20____р. №__

Термін подання завершеної роботи на кафедру 03.06.2025
(рік, місяць, число)

Вихідні дані до випускної бакалаврської роботи _____

Перелік питань, які потрібно розробити: Аналіз вимог до системи, аналіз предметної області, проєктування, реалізація, тестування системи

Перелік графічних документів (за потреби)_____

Дата видачі завдання “_____”_____20____р.

Керівник випускної бакалаврської роботи _____/ Волошин С.М.
(підпис) (прізвище та ініціали)

Завдання прийняв до виконання _____/ Кущенко В.К.

(підпис) (прізвище та ініціали студента)

КАЛЕНДАРНИЙ ПЛАН

з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Аналіз вимог до системи	07.03.2025 р.	Виконано
2	Проєктування системи	02.04.2025 р.	Виконано
3	Реалізація системи	03.04.2025 р.	Виконано
4	Тестування розробленої системи	12.04.2025 р.	Виконано
5	Оформлення пояснювальної записки	18.05.2025 р.	Виконано
6	Оформлення графічного матеріалу	20.05.2025 р.	Виконано

Студент _____ / **Кущенко В.К_._____**
(підпис) (ініціали та прізвище)

Керівник проекту (роботи) _____ / **Волошин С.М.**_____
(підпис) (ініціали та прізвище)

ЗМІСТ

ВСТУП

У сучасних умовах глобальної енергетичної трансформації, спричиненої зростанням попиту на енергоресурси, необхідністю підвищення ефективності їх використання та переходом до сталого розвитку, особливої актуальності набувають системи моніторингу споживання енергії. В умовах постійного зростання тарифів на електроенергію та воду а також потреби у зменшенні енергетичних витрат у житловому секторі, впровадження таких систем стає не лише бажаним, а й економічно доцільним.

Актуальність обраної теми зумовлена потребою у розробці доступного, функціонального та гнучкого рішення, яке б дозволяло в реальному часі здійснювати облік і контроль за споживанням електроенергії та води в побутових умовах. Існуючі комерційні системи переважно мають обмежений функціонал або високу вартість, що унеможливлює їх масове впровадження в індивідуальних господарствах. У зв’язку з цим, розробка власної системи моніторингу на базі відкритих технологій, зокрема Raspberry Pi, є актуальним завданням як з технічної, так і з економічної точки зору.

Метою кваліфікаційної роботи є розробка програмно-апаратного комплексу для моніторингу енергоспоживання будинку з використанням мікрокомп’ютера Raspberry Pi та спеціалізованих датчиків, що забезпечує збирання, обробку та візуалізацію даних про споживання ресурсів через веб-інтерфейс у режимі реального часу.

Для досягнення поставленої мети необхідно вирішити наступні **завдання**:

- вивчити існуючі рішення у сфері енергомоніторингу та оцінити їх переваги і недоліки;
- описати принципи роботи обраних датчиків для вимірювання споживання електроенергії та води;
- обґрунтувати вибір апаратних компонентів для побудови системи;
- розробити схему підключення та алгоритми взаємодії пристроїв;

– реалізувати програмне забезпечення для зчитування, зберігання та обробки даних;
– створити адаптивний веб-інтерфейс для відображення актуальної інформації;

Об’єктом дослідження у даній кваліфікаційній роботі є процес енергомоніторингу в умовах приватного житлового будинку з використанням засобів мікроелектроніки та програмування.

Методи дослідження, що були використані в роботі, включають: теоретичний аналіз літературних джерел та нормативної документації; системний аналіз існуючих технічних рішень; експериментальний метод (збирання та тестування системи); методи модульного програмування (створення окремих функціональних частин програмного забезпечення); візуалізаційні методи (побудова графіків, REST API, веб-інтерфейс).

Практичне значення кваліфікаційної роботи полягає в можливості адаптації запропонованого рішення для реального використання в індивідуальних господарствах, освітніх закладах та для подальшої комерціалізації у формі DIY-наборів або інтегрованих smart-систем.

Структура кваліфікаційної роботи побудована логічно та послідовно і складається зі вступу, шести розділів, висновків, списку використаних джерел та додатків. У першому розділі подано аналіз проблеми та огляд існуючих рішень. У другому розглянуто апаратну складову системи. Третій розділ присвячено розробці програмного забезпечення. У четвертому описано реалізацію веб-інтерфейсу. П’ятий розділ містить аналіз питань охорони праці та безпеки, а шостий – економічне обґрунтування проєкту. Обсяг основного тексту становить 83 сторінки, додатків – 6 сторінок. У роботі використано 13 рисунків, 2 таблиці, а також 64 джерела.

Таким чином, запропонована кваліфікаційна робота має завершений, прикладний характер, обґрунтовану структуру та може бути корисною для впровадження у сфері енергоменеджменту побутових об’єктів.

РОЗДІЛ 1. АНАЛІЗ СТАНУ ПРОБЛЕМИ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1. Проблема контролю споживання ресурсів у житловому секторі

Сучасний житловий сектор характеризується постійним зростанням обсягів споживання енергетичних та природних ресурсів, зокрема електроенергії, води. Така тенденція обумовлена низкою факторів, серед яких — підвищення рівня комфорту в житлових умовах, зростання кількості побутових електроприладів, а також зміни у структурі житлової забудови, що супроводжуються збільшенням середньої площі житла на одного мешканця.

Згідно з аналітичними даними Державної служби статистики України, частка домогосподарств у загальному обсязі споживання електроенергії становить понад 30%, а обсяги використання води у побуті залишаються стабільно високими. Водночас, рівень енергоефективності більшості житлових будівель є недостатнім, що зумовлює значні втрати ресурсів та перевитрати коштів з боку споживачів [1].

Актуальність моніторингу споживання ресурсів посилюється також у контексті глобальних викликів, таких як зміна клімату, нестабільність на енергетичних ринках та поступовий перехід до політики сталого розвитку. Підвищення тарифів на комунальні послуги створює об’єктивну необхідність для користувачів мати повну та своєчасну інформацію про власне споживання ресурсів з метою оптимізації витрат.

У цьому контексті моніторинг ресурсоспоживання стає не лише інструментом економії, але й засобом підвищення енергоефективності житлового сектора загалом. Створення та впровадження доступних систем локального контролю, що дозволяють здійснювати збір, зберігання та аналіз даних у режимі реального часу, розглядається як важливий крок у напрямку цифровізації побутової інфраструктури [1].

Історично склалося так, що в більшості житлових будинків облік електроенергії та води здійснюється за допомогою аналогових або електромеханічних лічильників. Попри відносну простоту та низьку вартість таких пристроїв, їхнє застосування має суттєві обмеження, які в умовах сучасних викликів роблять їх малоефективними.

Однією з головних проблем традиційного обліку є відсутність оперативного доступу до інформації. Дані з лічильників зчитуються вручну або з періодичністю, що встановлюється постачальником послуг, що унеможливлює контроль у реальному часі. Це, в свою чергу, ускладнює виявлення надлишкового або аварійного споживання на ранньому етапі [2].

Ще одним суттєвим недоліком є низький рівень інтеграції традиційних лічильників з цифровими сервісами. Відсутність каналів зв’язку або інтерфейсів передачі даних унеможливлює побудову систем автоматизованого збору та обробки показників. Також варто відзначити, що більшість механічних лічильників не здатні забезпечити деталізацію по споживанню в залежності від часу доби або конкретних приладів-споживачів.

Крім того, у контексті енергоменеджменту та побудови розумних будинків, традиційні системи обліку не відповідають сучасним вимогам щодо масштабованості, аналітики та взаємодії з іншими цифровими компонентами. Зокрема, вони не дозволяють реалізувати сценарії автоматичного реагування на зміну показників (наприклад, автоматичне повідомлення при перевищенні встановленого ліміту споживання електроенергії).

Таким чином, обмеженість функціональних можливостей традиційних лічильників унеможливлює їх ефективне використання в умовах зростаючої складності побутових енергетичних систем і наголошує на необхідності переходу до новітніх цифрових технологій моніторингу [2].

З огляду на обмеження традиційних засобів обліку, сучасні умови вимагають впровадження нових технічних рішень, здатних забезпечити оперативний контроль за споживанням ресурсів на рівні окремого житлового об’єкта. У цьому контексті особливої актуальності набувають локальні системи

моніторингу, які функціонують автономно, збирають і обробляють дані в реальному часі та можуть взаємодіяти з кінцевим користувачем через інтуїтивно зрозумілий інтерфейс.

Такі системи забезпечують низку суттєвих переваг. По-перше, вони дозволяють користувачу безпосередньо контролювати власне споживання ресурсів без необхідності залучення сторонніх сервісів. По-друге, завдяки можливості налаштування локальних порогів і сповіщень, можна реалізувати раннє виявлення аварійних або аномальних ситуацій (наприклад, витоку води поза межами нормативу чи перевантаження електромережі) [3].

Додатковою перевагою є можливість довгострокового зберігання даних та їх аналітичної обробки. Це відкриває шлях до вивчення споживчих звичок, прогнозування навантажень, а також до виявлення шляхів зниження витрат. Зокрема, застосування технологій обробки даних (data logging, time-series analysis) дозволяє відстежувати тенденції використання ресурсів у динаміці.

Варто також зазначити, що сучасна елементна база (Raspberry Pi, мікроконтролери, сенсори) і програмні платформи з відкритим кодом (наприклад, Python + Flask) дозволяють створювати подібні системи з відносно невеликими витратами. Це робить їх доступними для впровадження не лише у новобудовах, але й у вже наявному житловому фонді, де модернізація інфраструктури традиційно вважається складною та дорогою [4].

Таким чином, створення локальних систем моніторингу ресурсоспоживання є не лише технічно доцільним, але й економічно виправданим кроком, що сприяє підвищенню енергоефективності, зниженню витрат та забезпеченню екологічної відповідальності на рівні кожного домогосподарства.

1.2 Огляд існуючих систем енерго та ресурсомоніторингу

На сучасному ринку представлено чимало готових комерційних рішень для автоматизованого моніторингу споживання електроенергії та води. Такі системи, як правило, пропонуються як частина сервісного пакету з боку постачальників комунальних послуг або спеціалізованих інженерних компаній. Основною метою їх впровадження є централізований збір показників з великої кількості об'єктів для обробки, аналізу й виставлення рахунків.

Типовими представниками цього сегменту є системи типу AMR (Automated Meter Reading) або AMI (Advanced Metering Infrastructure), до яких відносяться, зокрема, рішення компаній Siemens, Honeywell, Itron, Elster, а в українських реаліях — сервіси на кшталт «Облік Online» або «ЕнергоСофт». Вони побудовані за архітектурою “endpoint + шлюз + сервер”, де кожен лічильник обладнаний передавачем (часто на базі GSM або LoRa), а зібрані дані зберігаються у хмарних серверах для подальшої обробки [5].

Такі системи забезпечують високу точність вимірювань, мають сертифікацію та відповідність галузевим стандартам. Проте, поряд із цим, вони характеризуються обмеженою гнучкістю у налаштуваннях, відсутністю прямого доступу до «сирих» даних зі сторони кінцевого користувача, а також високою вартістю встановлення та обслуговування. У більшості випадків споживач не має змоги оперативно реагувати на зміну ситуації, оскільки оновлення інформації здійснюється з певною затримкою.

Додатковим викликом є інтеграція таких систем у вже наявну інфраструктуру житлового будинку. Нерідко процес підключення потребує спеціального дозволу або технічного втручання з боку постачальника послуг, що значно ускладнює впровадження в приватному секторі або у випадку індивідуальних рішень.

Таким чином, попри високу надійність і масштабованість, комерційні системи моніторингу не завжди відповідають потребам кінцевого споживача,

який зацікавлений у гнучкому, доступному та локалізованому рішенні для контролю ресурсів у межах окремого домогосподарства.

Останнє десятиліття відзначено стрімким розвитком технологій розумного будинку, що інтегрують у собі керування освітленням, кліматом, безпекою та, зокрема, моніторингом споживання ресурсів. На ринку з'явилась низка рішень, орієнтованих на побутового користувача, які забезпечують інтерактивний контроль споживання електроенергії та води часто в складі більших екосистем [5].

Найпоширеніші платформи цього класу — Google Nest, TP-Link Tapo, Tuya Smart, Shelly, Home Assistant, Samsung SmartThings тощо. Більшість із них підтримують інтеграцію з сенсорами, інтелектуальними лічильниками або розумними реле, що дозволяє не лише відстежувати, але й частково керувати навантаженнями, формувати сценарії (наприклад, автоматичне вимкнення електроприладів при досягненні заданого порогу) чи отримувати сповіщення у разі виявлення витоку [6].

Перевагою таких рішень є високий ступінь автоматизації, наявність мобільних застосунків, можливість віддаленого доступу через хмарні сервіси та підтримка широкого спектру пристроїв. Наприклад, у системі Shelly Energy Meter можна зчитувати в реальному часі потужність, напругу та струм кожної фази, а Aqara Water Leak Sensor здатен передавати тривожні сигнали у разі виявлення витоку води [6].

Разом із тим, ці рішення мають певні обмеження. По-перше, вони часто є закритими у плані архітектури: користувач має обмежений доступ до внутрішньої логіки обробки даних. По-друге, їх повноцінне функціонування потребує постійного з'єднання з інтернетом та хмарною інфраструктурою, що знижує автономність і підвищує залежність від сторонніх серверів. У разі втрати доступу до інтернету система може втратити функціональність або працездатність. Крім того, такі рішення, як правило, мають високу вартість як на етапі придбання, так і при подальшій експлуатації.

Зважаючи на це, рішення класу Smart Home доцільні для користувачів, які готові інвестувати у комплексну автоматизацію побуту, однак не завжди підходять для цільового проєктування автономної, бюджетної системи моніторингу ресурсів, особливо у випадках, коли критичною є локальність та адаптивність програмного забезпечення.

На відміну від комерційних та закритих рішень класу Smart Home, останніми роками широке поширення набули DIY-підходи (англ. Do It Yourself) до побудови систем моніторингу енергоспоживання. Ключова особливість таких проєктів полягає в їх відкритості, доступності компонентної бази, гнучкості у виборі архітектури та можливості повного контролю над програмною частиною.

Найпопулярнішими платформами для реалізації таких систем є Raspberry Pi, Arduino, ESP32 та їх похідні. Ці пристрої відзначаються невисокою вартістю, компактністю та низьким енергоспоживанням, що дозволяє їх вільно інтегрувати у побутову інфраструктуру. У поєднанні з датчиками струму (наприклад, PZEM-004T), імпульсними лічильниками води (YF-S201) чи сенсорами (MQ-4, MG-811), DIY-системи дозволяють точно зчитувати поточні показники споживання, обробляти їх та візуалізувати через веб-інтерфейс.

Програмна частина зазвичай реалізується з використанням мови Python, Node.js або C/C++ для мікроконтролерів. Для веб-інтерфейсу застосовують фреймворки Flask, Node-RED, Vue.js, а для баз даних – SQLite, InfluxDB або Firebase. Популярні приклади реалізацій включають такі проєкти, як OpenEnergyMonitor, DIY Smart Meter або численні приклади з репозиторіїв GitHub, Hackaday чи Hackster.io, які часто супроводжуються схемами підключення, вихідним кодом і інструкціями з налаштування [7].

Перевагами підходу є не лише низька вартість, а й можливість повного контролю над зібраними даними, локальне зберігання без прив’язки до сторонніх серверів, адаптація під конкретні потреби користувача. Водночас недоліками можуть бути підвищені вимоги до технічної підготовки розробника, потреба в пайці, налаштуванні ПЗ, а також у забезпеченні базової безпеки електричних з’єднань.

DIY-рішення на базі одноплатних комп’ютерів є доцільними для цільових індивідуальних або навчальних проєктів, де важливо поєднати функціональність, економічність і практичний досвід розробки.

1.3. Порівняння промислових і DIY-рішень

Для здійснення порівняльного аналізу різних підходів до побудови систем моніторингу енергоспоживання у побутовому секторі необхідним є формування об’єктивної методології, що враховує як технічні, так і експлуатаційні характеристики. Такий підхід дозволяє здійснити якісну оцінку переваг і недоліків як промислових систем, так і рішень типу Smart Home та DIY-систем, що ґрунтуються на відкритих апаратно-програмних платформах [8].

З огляду на специфіку дослідження, було виділено наступні вісім ключових критеріїв, які дозволяють комплексно охарактеризувати функціональність систем у різних умовах експлуатації:

1. Вартість впровадження один із найважливіших факторів, особливо у побутовому секторі. Вона включає в себе не лише ціну апаратних компонентів, а й витрати на встановлення, технічну підтримку та можливі ліцензійні платежі.
2. Здатність системи бути розширеною або адаптованою до більших обсягів даних або кількості підключених пристроїв без суттєвих змін в архітектурі. Цей параметр критично важливий у разі, якщо планується розширення системи на кілька приміщень або будівель.
3. Гнучкість конфігурації визначає можливість адаптації системи до індивідуальних потреб користувача: зміна логіки збору або обробки даних, інтеграція додаткових модулів, кастомізація інтерфейсу тощо.
4. Простота встановлення та обслуговування — враховує необхідний рівень складності під час первинної інсталяції, а також зручність поточної підтримки, діагностики несправностей і модернізації.

5. Автономність та локальність зберігання даних, здатність системи працювати без постійного з’єднання з мережею Інтернет та зберігати зібрані показники на локальних носіях, що підвищує надійність у разі збоїв зв’язку.
6. Доступ до «сирих» даних та контроль над логікою обробки визначає, наскільки користувач має можливість переглядати, змінювати або аналізувати необроблені дані без обмежень, накладених виробником або платформою.
7. Необхідний рівень технічної підготовки користувача відображає, чи потребує система спеціальних знань для налаштування та роботи: від базового розуміння до глибоких технічних навичок, таких як програмування, робота з мікроелектронікою або мережевими протоколами.
8. Наявність інтерфейсу користувача оцінює ступінь зручності доступу до інформації, зокрема наявність веб- або мобільного застосунку, зворотного зв’язку, інтерактивності, візуалізацій (графіків, таблиць) тощо.

Оцінювання проводиться за кожним критерієм окремо з метою виявлення сильних і слабких сторін кожного типу рішень. Такий підхід дозволяє забезпечити не лише кількісну, а й якісну інтерпретацію даних для подальшого обґрунтування вибору технологічної архітектури для цільового застосування.

Для наочного порівняння трьох основних типів систем моніторингу ресурсів — промислових, комерційних Smart Home-рішень та DIY-систем на базі Raspberry Pi — доцільно узагальнити результати в табличній формі (Таблиця 1.1). Це дозволяє швидко оцінити їх придатність до впровадження у житловому секторі, враховуючи вісім виділених раніше критеріїв [9].

Таблиця 1.1

Порівняльна характеристика систем моніторингу

Критерій	**Промислові рішення**	**Smart Home-системи (готові рішення)**	**DIY-системи (на базі Raspberry Pi)**
1. Вартість впровадження	Висока: включає вартість обладнання, ліцензії, монтажу та сервісу	Середня: вартість комплектів помірна, часто без абонплати	Низька: лише витрати на модулі, датчики та одноплатний комп'ютер
2. Масштабованість	Висока: підтримка промислових стандартів, централізоване керування	Обмежена: розширення часто потребує дорогих фірмових модулів	Середня: можлива адаптація архітектури під більшу кількість датчиків
3. Гнучкість конфігурації	Низька: обмеження на зміну логіки системи та ПЗ	Середня: деяка кастомізація можлива через мобільний застосунок	Висока: повний контроль над програмною логікою, структурою даних та інтеграцією
4. Простота встановлення	Потребує професійного монтажу та налагодження	Середній рівень складності, іноді можливо встановити самостійно	Вимагає технічної підготовки, ручного підключення та конфігурації
5. Автономність та локальність	Залежність від хмарного сервісу або серверів виробника	Часто потребує підключення до інтернету	Повна автономність, можливість локального збереження та обробки інформації
6. Доступ до сирих даних	Закритий доступ: лише агреговані або обмежені звіти	Частковий доступ через API, але обмежений функціоналом	Повний доступ до даних у реальному часі, можливість архівування, виводу в довільному форматі
7. Рівень технічної підготовки	Мінімальний: користувач взаємодіє з кінцевим інтерфейсом	Низький: інтерфейс орієнтований на споживача	Високий: потребує знань в електроніці, Python, роботі з протоколами UART/SPI/GPIO
8. Інтерфейс користувача	Зазвичай фірмове ПЗ або хмарний кабінет	Прості застосунки для смартфонів або планшетів	Кастомізований інтерфейс (на Flask або інше), веб-панель з графіками і логами

Узагальнюючи, можна відзначити, що промислові рішення, незважаючи на високу надійність і масштабованість, зазвичай не дозволяють гнучкої адаптації під індивідуальні потреби, а також потребують значних фінансових витрат. Smart

Home-системи представляють зручний, але менш контрольований варіант. Натомість DIY-рішення вирізняються високим ступенем адаптивності, відкритістю та низькою собівартістю, що робить їх доцільними в умовах освітніх, дослідницьких або індивідуальних проєктів.

1.4. Обґрунтування вибору Raspberry Pi як платформи реалізації

На сучасному етапі розвитку вбудованих систем спостерігається широкий спектр апаратних платформ, що використовуються для реалізації проєктів у сфері моніторингу параметрів навколишнього середовища, енергоспоживання, безпеки та автоматизації. Серед найбільш поширених рішень, які можуть бути застосовані для побудови системи контролю ресурсів у житловому секторі, варто виокремити Arduino, ESP32, промислові програмовані логічні контролери (ПЛК) та одноплатні комп'ютери (SBC), зокрема Raspberry Pi.

Arduino є однією з найпопулярніших платформ серед початківців та ентузіастів через свою простоту, відкритість та велику кількість підтримуваних датчиків. Втім, обмежені обчислювальні ресурси (відсутність багатозадачності, низька частота процесора, відсутність підтримки повноцінної ОС) знижують придатність Arduino для систем, що потребують одночасного зчитування з кількох каналів, збереження даних, візуалізації чи хостингу веб-інтерфейсу [10].

ESP32 є суттєвим кроком уперед порівняно з Arduino завдяки вбудованій підтримці Wi-Fi та Bluetooth, більшій тактовій частоті та багатоядерній архітектурі. Цей мікроконтролер часто використовується в IoT-проєктах, однак реалізація складних функцій, таких як обробка великих масивів даних або інтеграція баз даних, значно ускладнюється через обмежений об'єм оперативної пам'яті та відсутність повноцінної ОС.

Промислові ПЛК (наприклад, Siemens LOGO!, Schneider Electric, ОВЕН тощо) вирізняються високою надійністю, сертифікацією для роботи у складних умовах та широкими можливостями масштабування. Проте їх висока вартість,

закритість програмного середовища та складність інтеграції з відкритими платформами обмежують їх застосування у приватному секторі або для навчально-дослідницьких цілей.

Натомість Raspberry Pi (Рис. 1.1.) поєднує в собі переваги як мікроконтролерів, так і повноцінних комп'ютерів. Цей одноплатний комп'ютер підтримує повноцінну багатозадачну операційну систему на базі Linux (наприклад, Raspberry Pi OS), дозволяє розгортати сервери, працювати з мовами програмування високого рівня (Python, C/C++, JavaScript), а також підтримує всі необхідні периферійні інтерфейси для взаємодії з цифровими та аналоговими датчиками. Крім того, існує широке ком'юніті розробників, яке сприяє швидкому вирішенню проблем та надає велику кількість відкритих рішень.

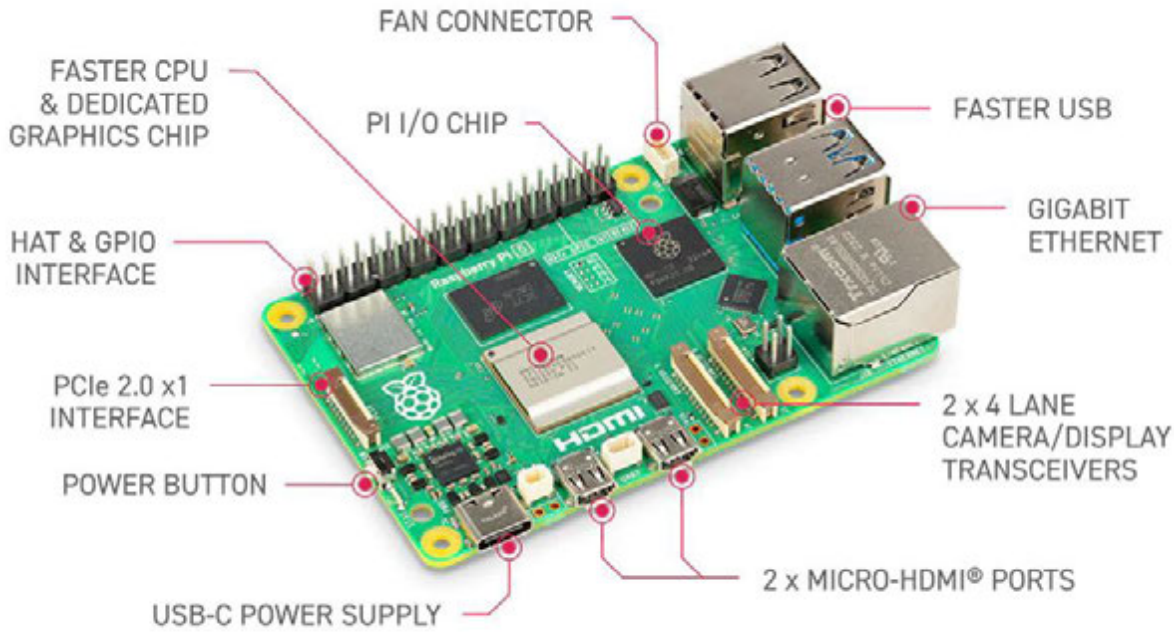

Рис. 1.1 – Апаратні компоненти плати Raspberry Pi 5

На зображенні (Рис. 1.1.) представлено загальний вигляд одноплатного комп'ютера Raspberry Pi 5, із зазначенням ключових апаратних елементів, що забезпечують його функціональність у проєктах систем моніторингу. Плата обладнана оновленим процесором із вищою тактовою частотою та виділеним графічним ядром, що дозволяє виконувати багатозадачні обчислення.

Інтерфейси введення/виведення (GPIO, PCIe 2.0 x1, USB, HDMI) забезпечують гнучкість у підключенні зовнішніх сенсорів і модулів. Наявність гігабітного Ethernet і розширеної підтримки камер/дисплеїв підвищує інтеграційні можливості. Живлення здійснюється через USB-C, що гарантує стабільність енергозабезпечення під час роботи з кількома периферійними пристроями [11].

Таким чином, попри наявність альтернатив, Raspberry Pi демонструє найкраще співвідношення функціональності, доступності, гнучкості та масштабованості, що робить його оптимальним вибором для створення системи моніторингу енергоспоживання в рамках дипломного проєкту.

На схемі (Рис. 1.2.) зображено розташування основних функціональних модулів та інтерфейсів одноплатного комп’ютера Raspberry Pi 4. Особливу увагу приділено GPIO-порту (HDR1), що містить 40 універсальних контактів для підключення цифрових та аналогових сенсорів через зовнішні АЦП, наприклад, MCP3008. Також представлено відео- та камерні інтерфейси (CSI), які забезпечують можливість підключення камер або дисплеїв через FPC-роз’єми. З лівого боку схеми позначено чотири USB-порти та мережевий роз’єм RJ45 для гігабітного Ethernet-з’єднання. У верхній частині схеми розташовано два micro-HDMI порти для виведення зображення, роз’єм живлення USB-C, а також аудіовихід 3.5 мм. Праворуч — SPI-інтерфейс модуля SD-карт, який використовується для зберігання операційної системи та даних. Така структурна схема є важливою для правильного проєктування апаратної частини системи моніторингу [12].

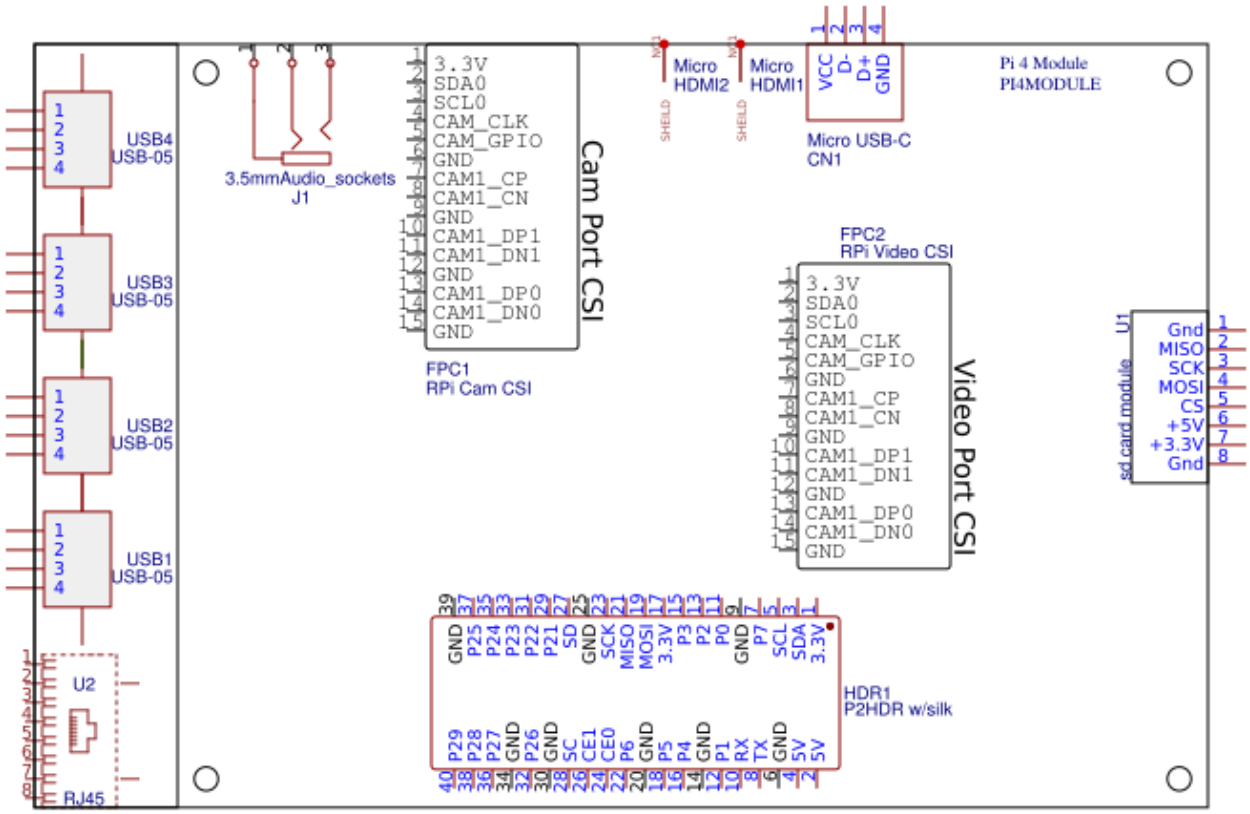

Рис. 1.2 – Схема роз’ємів та інтерфейсів плати Raspberry Pi 4

Платформа Raspberry Pi, як одноплатний комп’ютер загального призначення, має низку характеристик, які забезпечують її високу ефективність у завданнях збору, обробки та візуалізації даних з датчиків споживання ресурсів. У контексті реалізації комплексної системи моніторингу енергоспоживання в житловому будинку, саме технічні та функціональні особливості цієї платформи визначають її переваги порівняно з мікроконтролерними рішеннями.

По-перше, наявність повноцінної багатозадачної операційної системи (Linux) забезпечує можливість паралельної роботи декількох сервісів. Raspberry Pi дозволяє одночасно зчитувати дані з різних типів датчиків (через UART, GPIO, SPI, I2C), здійснювати локальну обробку інформації, вести журналювання у базі даних (наприклад, SQLite) і обслуговувати веб-інтерфейс (на Flask) без суттєвого впливу на продуктивність системи.

По-друге, високий рівень сумісності з периферійними пристроями є визначальним фактором у контексті роботи з датчиками електроенергії

(наприклад, PZEM-004T), води (YF-S201). Raspberry Pi має 40-pin GPIO роз'єм, який дозволяє безпосереднє підключення цифрових та аналогових пристроїв (через АЦП, як-от MCP3008), а також реалізацію апаратного PWM та апаратного UART без потреби в зовнішніх модулях.

По-третє, широка підтримка мов програмування, зокрема Python, значно спрощує процес реалізації прикладної логіки системи. Бібліотеки на кшталт RPi.GPIO, pySerial, sqlite3, matplotlib та flask дозволяють створити стабільну програмну архітектуру із модульною структурою, що полегшує подальшу підтримку і розширення системи.

По-четверте, Raspberry Pi підтримує інтеграцію з мережевими службами, що дає змогу реалізовувати не лише локальний моніторинг, а й віддалений доступ до системи. Через Ethernet або Wi-Fi модуль пристрій може надсилати дані на хмарні сервери, відправляти повідомлення про критичні події (наприклад, перевищення концентрації) або забезпечувати контроль із мобільного пристрою.

З огляду на наведене, можна констатувати, що Raspberry Pi забезпечує універсальність, масштабованість та технічну придатність для побудови адаптивної системи енергомоніторингу, яка здатна ефективно функціонувати як автономно, так і в інтеграції з іншими компонентами системи «розумного будинку».

Окрім технічної доцільності, важливим критерієм при виборі апаратної основи для системи моніторингу ресурсів є вартісна ефективність. У межах дипломного проєкту, де обмеження бюджету відіграє ключову роль, Raspberry Pi зарекомендував себе як платформа, що забезпечує оптимальне співвідношення «ціна–функціональність».

Вартість базової моделі Raspberry Pi (наприклад, Raspberry Pi 4 Model B або Raspberry Pi Zero 2 W) у середньому становить від 800 до 2000 грн, залежно від модифікації та комплектації. За ці кошти користувач отримує повноцінний одноплатний комп'ютер із можливістю роботи в мережі, з інтерфейсами HDMI, USB, GPIO, підтримкою SD-карт пам'яті та живленням через micro-USB або

USB-C. Це дозволяє не лише реалізувати функціонал збору даних, а й зекономити на окремих периферійних модулях, таких як Ethernet-конвертери або Wi-Fi-шилди, які були б необхідні в мікроконтролерних рішеннях (на кшталт Arduino UNO).

Суттєвою перевагою Raspberry Pi є великий обсяг відкритої документації, прикладів коду та технічної підтримки, доступних у спільнотах розробників, на форумах, у репозиторіях GitHub, а також на офіційному сайті Raspberry Pi Foundation. Це дозволяє значно скоротити час на прототипування та налагодження, що, в свою чергу, знижує загальні витрати на розробку [13].

Крім того, Raspberry Pi активно застосовується в численних реальних проєктах, пов’язаних із моніторингом електроенергії, споживання води, контролем якості повітря, аграрною автоматизацією та розумним домом. Існують численні успішні реалізації, у межах яких ця платформа слугувала ядром для збору даних із мультисенсорних систем, інтеграції з хмарними сервісами або розгортання повноцінного веб-інтерфейсу керування. Зокрема, серед відкритих проєктів можна виокремити OpenEnergyMonitor, PiHome, Home Assistant, що засвідчують практичну ефективність цієї платформи.

Таким чином, із точки зору вартості, легкості у впровадженні та широкого досвіду попереднього використання, Raspberry Pi є економічно обґрунтованим вибором, що забезпечує високу ефективність у контексті проєктів моніторингу побутових ресурсів у навчальному та дослідницькому середовищі.

1.5. Висновки до розділу

У першому розділі було проведено комплексний аналіз сучасного стану проблеми моніторингу споживання енергетичних і ресурсних носіїв у житловому секторі. Досліджено актуальність тематики в умовах глобального зростання споживання електроенергії та води а також у контексті підвищення енергоефективності та необхідності забезпечення сталого розвитку.

Проведений огляд існуючих комерційних та самостійно зібраних систем (DIY-рішень) засвідчив, що готові промислові продукти мають високий рівень надійності та інтерфейсної зручності, однак значно поступаються за параметрами гнучкості, масштабованості та вартості. У свою чергу, DIY-системи на базі одноплатних комп’ютерів, зокрема Raspberry Pi, надають можливість створити кастомізований інструмент моніторингу, адаптований під конкретні потреби користувача [14].

У результаті порівняльного аналізу було обґрунтовано доцільність використання Raspberry Pi як обчислювальної платформи для побудови системи моніторингу енергоспоживання. Її функціональні можливості, зокрема наявність універсального GPIO-інтерфейсу, підтримка широкого спектру периферійних пристроїв та інтеграція з сучасними засобами веб-програмування, дозволяють реалізувати ефективну систему збору, обробки й візуалізації даних.

Таким чином, результати розділу закладають теоретичну основу для наступних етапів роботи, зокрема — вибору апаратної частини системи, розробки програмного забезпечення та побудови інтерфейсу користувача.

РОЗДІЛ 2. ВИБІР І ОБҐРУНТУВАННЯ АПАРАТНИХ КОМПОНЕНТІВ

2.1. Структура системи моніторингу

Запропонована система моніторингу ресурсів (електроенергії та води) має модульну архітектуру та умовно поділяється на два взаємопов’язані рівні: апаратний (Hardware Layer) та програмний (Software Layer).

Апаратний рівень включає набір компонентів, необхідних для безпосереднього вимірювання фізичних параметрів ресурсів, їх попереднього перетворення, оцифровування та передачі до центрального обчислювального модуля. В основу апаратної архітектури покладено одноплатний комп’ютер Raspberry Pi (версія 4 або 5), який функціонує як центральний контролер системи, що забезпечує збір, обробку та зберігання даних із сенсорів [15].

Для моніторингу електроенергії використовується датчик PZEM-004T, який підключається до Raspberry Pi через UART-інтерфейс. Це дозволяє отримувати інформацію щодо поточної напруги, струму, спожитої потужності та сумарної енергії з високою точністю й частотою оновлення.

Моніторинг витрати води реалізується за допомогою імпульсного датчика протоку рідини типу YF-S201, що під’єднаний безпосередньо до GPIO-контактів Raspberry Pi та працює в режимі переривань для точного підрахунку імпульсів, які відповідають певному об’єму спожитої води.

Використовується аналоговий датчик типу MQ-4 або MG-811, сигнали з якого подаються через аналогово-цифровий перетворювач MCP3008, під’єднаний за допомогою SPI-інтерфейсу Raspberry Pi. Це дозволяє здійснювати вимірювання у вигляді аналогових сигналів, які далі оцифровуються і обробляються.

Програмний рівень системи складається з набору Python-модулів, що відповідають за отримання, первинну обробку та зберігання даних, а також з веб-сервера, що забезпечує доступ користувачів до інформації через веб-інтерфейс.

Збір та збереження даних здійснюється за допомогою програмних бібліотек (pySerial, RPi.GPIO, spidev), а зберігання — у локальній базі даних SQLite.

Користувацький веб-інтерфейс реалізується за допомогою фреймворка Flask, що забезпечує інтерактивність, графічну візуалізацію отриманих показників та можливість отримання інформації через REST API для подальшої інтеграції чи аналітики.

Таким чином, запропонована архітектура дозволяє створити ефективну систему моніторингу ресурсів, яка є відкритою для масштабування, достатньо гнучкою для кастомізації, а також економічно доцільною у контексті побутового використання чи навчально-дослідницьких проєктів [16].

Процес роботи розробленої системи моніторингу ресурсів передбачає чітку взаємодію її компонентів на рівні апаратних інтерфейсів та програмних модулів, що забезпечує цілісність і стабільність функціонування.

На початковому етапі здійснюється отримання показників від датчиків. Датчик електроенергії PZEM-004T передає цифрові дані до Raspberry Pi за допомогою UART-інтерфейсу (Universal Asynchronous Receiver-Transmitter). Для цього використовується бібліотека pySerial, яка забезпечує зчитування даних із послідовного порту. Отримані дані включають інформацію щодо напруги, струму, спожитої потужності й накопиченої енергії.

Датчик протоку води YF-S201 передає інформацію через GPIO-піни Raspberry Pi у вигляді імпульсів. При цьому застосовується механізм переривань (interrupt), що забезпечує точне підрахування імпульсів, відповідних до об'єму води. Обчислення реальної витрати відбувається програмно, згідно з калібрувальним коефіцієнтом [17].

Для отримання показників застосовується аналоговий датчик типу MQ-4 або MG-811, сигнали з якого проходять через аналогово-цифровий перетворювач MCP3008. Raspberry Pi взаємодіє з MCP3008 через SPI-інтерфейс (Serial Peripheral Interface), що дозволяє отримувати цифрове значення аналогових показників з високою точністю та мінімальними затримками.

Після первинного отримання даних, здійснюється їх обробка та зберігання. На цьому етапі Python-скрипти, що виконуються на Raspberry Pi [21], перетворюють отримані значення в стандартизований цифровий формат. Оброблені дані зберігаються у локальній базі даних SQLite, яка дозволяє організувати ефективне управління записами, забезпечуючи швидкий доступ до історії вимірювань [20].

Наступним важливим етапом є візуалізація інформації для кінцевого користувача. Дані з бази SQLite подаються у веб-інтерфейс, побудований за допомогою фреймворку Flask. Цей веб-додаток створює REST API, через який відбувається інтеграція з фронтендом, що дозволяє користувачеві переглядати актуальні значення ресурсів у режимі реального часу, аналізувати історичні графіки споживання, а також отримувати сповіщення у разі виникнення критичних ситуацій.

Схема (Рис. 2.1.) показує чітке розмежування на апаратні й програмні компоненти, а також взаємодію між ними:

– Апаратна частина:
    1. Raspberry Pi виступає ядром системи, з'єднуючи всі датчики.
    2. PZEM-004T передає дані через UART.
    3. Датчик YF-S201 передає дані через GPIO.
– Програмна частина:
    1. Python-скрипти забезпечують збір та обробку даних.
    2. SQLite використовується для локального збереження даних.
    3. Flask забезпечує інтерфейс для користувача (графіки, таблиці, API).

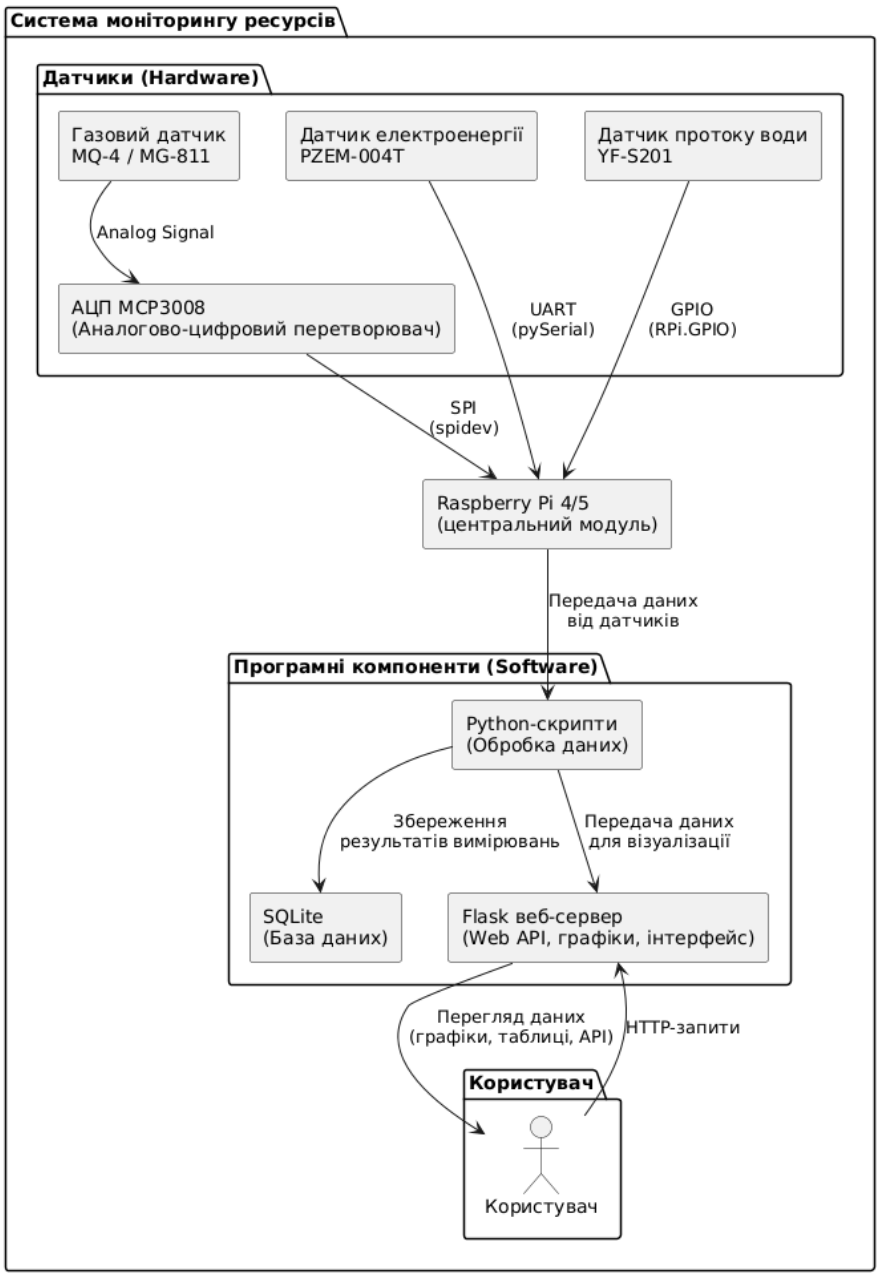

Рис. 2.1 – Схема взаємодії компонентів системи

2.2. Вибір датчика електроенергії: PZEM-004T

Однією з ключових задач при розробці системи моніторингу ресурсів будинку є точне вимірювання параметрів електричної мережі, таких як струм, напруга, активна потужність та загальна спожита енергія. Для забезпечення цієї функції необхідно обрати відповідний сенсор, що відповідає технічним вимогам щодо точності, надійності та зручності підключення до Raspberry Pi [22].

На основі проведеного порівняння різних моделей сенсорів для вимірювання електроенергії (ACS712, SCT-013, PZEM-004T), вибір було зупинено на модулі PZEM-004T версії 3.0 (Рис. 2.2) [23]. Даний датчик поєднує у собі декілька суттєвих переваг, які роблять його оптимальним для застосування у розроблюваній системі:

- Висока точність вимірювань: похибка для напруги становить не більше ±1%, для струму – ±1%, а для вимірювання активної потужності та енергії – близько ±2%. Це забезпечує надійний контроль і моніторинг енергоспоживання з мінімальними похибками.
- Комплексність вимірювань: PZEM-004T вимірює відразу декілька параметрів електромережі — струм (до 100 А через трансформатор струму), напругу (80–260 В змінного струму), активну потужність (до 22 кВт) та накопичену спожиту електроенергію, що значно спрощує реалізацію системи, оскільки не потребує окремих сенсорів на кожен параметр.
- Простота підключення до Raspberry Pi: датчик має цифровий інтерфейс UART (послідовний зв’язок), що легко підключається до одноплатних комп’ютерів, включаючи Raspberry Pi, через стандартний USB-UART адаптер або GPIO-піни з використанням модуля послідовного інтерфейсу. Програмна взаємодія здійснюється через бібліотеку pySerial, що дозволяє оперативно обробляти та записувати дані в базу SQLite.
- Доступність документації і бібліотек: велика кількість документації, прикладів коду, бібліотек з відкритим кодом значно полегшує інтеграцію цього модуля в проєкт і мінімізує можливі труднощі під час розробки програмного забезпечення.

Конструктивно модуль PZEM-004T складається з вимірювального блоку та трансформатора струму, який охоплює фазний провід електромережі будинку. Сам вимірювальний модуль підключається до джерела живлення та забезпечує передачу інформації до центрального обчислювального пристрою (Raspberry Pi)

[24]. Така архітектура гарантує електричну безпеку для електронних компонентів системи, а також зручність експлуатації й обслуговування.

Рис. 2.2 – Датчик PZEM-004T

Датчик PZEM-004T було обрано завдяки його високій точності, простоті інтеграції з Raspberry Pi, зручності налаштування та достатній кількості підтримувальних ресурсів, що робить його ідеальним вибором для створення ефективної та економічно обґрунтованої системи моніторингу електроспоживання в побутових умовах.

2.3. Вибір датчика води: YF-S201

Наступним елементом системи моніторингу ресурсів будинку є датчик для вимірювання витрат води. Для точного та надійного вимірювання кількості спожитої води необхідно використовувати пристрій, який здатний оперативно й

точно передавати інформацію до центрального контролера Raspberry Pi. З цією метою було проведено аналіз доступних рішень та обрано датчик протоку рідини YF-S201 [25].

Датчик YF-S201 є цифровим сенсором протоку рідини, принцип роботи якого заснований на використанні внутрішньої турбіни, що приводиться в рух потоком води. При обертанні турбіни генеруються електричні імпульси, кількість яких прямо пропорційна об’єму води, що проходить через датчик. Це дозволяє програмно обраховувати точні витрати води за одиницю часу [25].

Головні критерії вибору датчика YF-S201 полягають у наступних перевагах:

- Точність і надійність вимірювань: точність вимірювання витрати води датчиком YF-S201 становить до ±2%, що є достатньо високим показником для побутових систем моніторингу. Діапазон робочих витрат для цього датчика складає від 1 до 30 літрів на хвилину, що цілком відповідає реальним умовам побутового споживання.
- Цифровий вихід (імпульсний сигнал): наявність цифрового виходу дозволяє напряму підключати датчик до GPIO-портів Raspberry Pi, що суттєво спрощує схему підключення та мінімізує додаткові компоненти. Це знижує загальну вартість системи та підвищує її надійність і стабільність.
- Низьке енергоспоживання і простота інтеграції: датчик живиться від стандартної напруги 5 В постійного струму, яка доступна від Raspberry Pi. Для зчитування даних не потрібні додаткові модулі перетворення сигналу, достатньо підключити сенсор до GPIO-піна, налаштованого у режимі переривань.
- Простота програмної обробки сигналів: кількість імпульсів, що генерує датчик, лінійно відповідає об’єму витраченої води, а для розрахунку витрат застосовується просте множення на попередньо визначений калібрувальний коефіцієнт. Це значно спрощує алгоритм програмної обробки сигналів.

Конструктивно датчик (Рис. 2.3.) складається з пластикового корпусу з вхідним та вихідним штуцерами стандартного розміру, а також вбудованої турбіни із магнітами та геркона (датчика Холла), який перетворює обертальний рух турбіни в електричні імпульси. Надійна герметизація корпусу дозволяє експлуатувати датчик у вологих умовах без додаткового захисту.

Рис. 2.3 – Датчик YF-S201

Враховуючи простоту інтеграції, точність вимірювання, низьке енергоспоживання та доступність для кінцевого користувача, датчик YF-S201 є оптимальним рішенням для реалізації системи контролю споживання води в будинку.

2.4. Вибір плати розширення (MCP3008) для зчитування аналогових сигналів

Під час реалізації системи моніторингу ресурсів будинку на базі Raspberry Pi виникла потреба у вимірюванні аналогових сигналів, що формуються певними сенсорами. Raspberry Pi, незважаючи на свої широкі функціональні можливості, не має вбудованого аналогово-цифрового перетворювача (АЦП), що унеможливлює пряме підключення аналогових датчиків. Тому для перетворення аналогових сигналів у цифровий формат необхідно використати додатковий АЦП-модуль [29].

Після аналізу ринку доступних АЦП, які сумісні з Raspberry Pi, було прийнято рішення використовувати модуль MCP3008 (Рис. 2.4.).

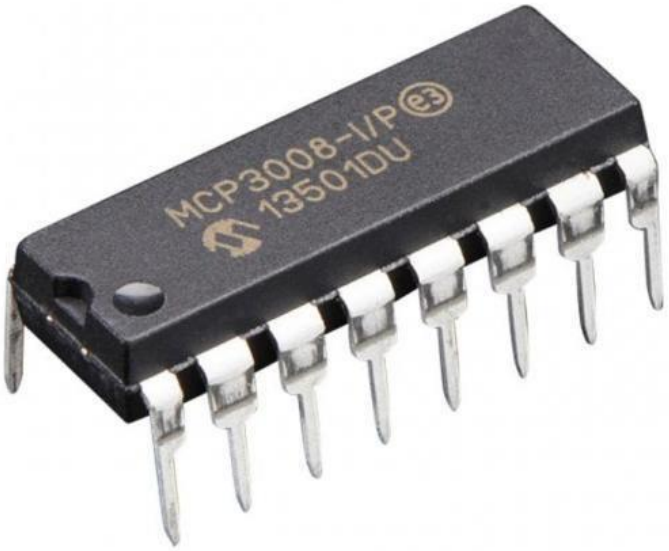

Рис. 2.4 – Плата розширення MCP3008

Цей аналогово-цифровий перетворювач має низку суттєвих переваг, які роблять його оптимальним для інтеграції в систему моніторингу ресурсів будинку [30].

Основні причини вибору MCP3008 полягають у наступному:

- Сумісність з Raspberry Pi через SPI-інтерфейс: MCP3008 має вбудований інтерфейс SPI (Serial Peripheral Interface), який забезпечує швидку та стабільну передачу даних у цифровому форматі до Raspberry Pi. Це дозволяє інтегрувати аналогові сигнали в єдиний цифровий потік даних без значних затримок та забезпечує оперативну обробку інформації в реальному часі.
- Висока точність та роздільна здатність: MCP3008 є 10-бітним аналогово-цифровим перетворювачем, що означає можливість представлення аналогових сигналів у цифровому форматі з 1024 рівнями дискретизації.
- Мультиканальність: Плата MCP3008 підтримує до восьми аналогових каналів, що дозволяє в перспективі додавати додаткові аналогові сенсори до системи без значних змін у схемі підключення. Це підвищує гнучкість і масштабованість системи моніторингу.
- Низьке енергоспоживання та надійність: Модуль має невеликі габарити, споживає мінімальну кількість енергії (живлення на рівні 3.3–5 В), що забезпечує тривалу та стабільну експлуатацію у складі автономних систем.
- Доступність та наявність документації: Висока популярність MCP3008 серед розробників зумовила наявність великої кількості прикладів застосування, документації, програмних бібліотек з відкритим кодом (наприклад, Adafruit CircuitPython MCP3xxx, spidev). Це значно скорочує час на впровадження та зменшує ймовірність виникнення технічних проблем у процесі розробки.

Завдяки простоті підключення до Raspberry Pi, високій точності вимірювань, багатоканальності та доступній ціні, модуль MCP3008 є найбільш оптимальним рішенням для перетворення аналогових сигналів датчиків у цифровий вигляд у рамках розробки дипломного проєкту системи моніторингу ресурсів житлового будинку [30].

2.5. Схеми підключення датчиків до Raspberry Pi

Коректне підключення обраних датчиків до центрального обчислювального модуля Raspberry Pi є критично важливим етапом у реалізації системи моніторингу ресурсів будинку. Від точності та надійності з'єднань залежить стабільність роботи всієї системи та точність отримуваних даних [31].

Нижче наведено детальні схеми (Рис. 2.5.) й опис підключення кожного з сенсорів: датчика електроенергії PZEM-004T, датчика протоку води YF-S201.

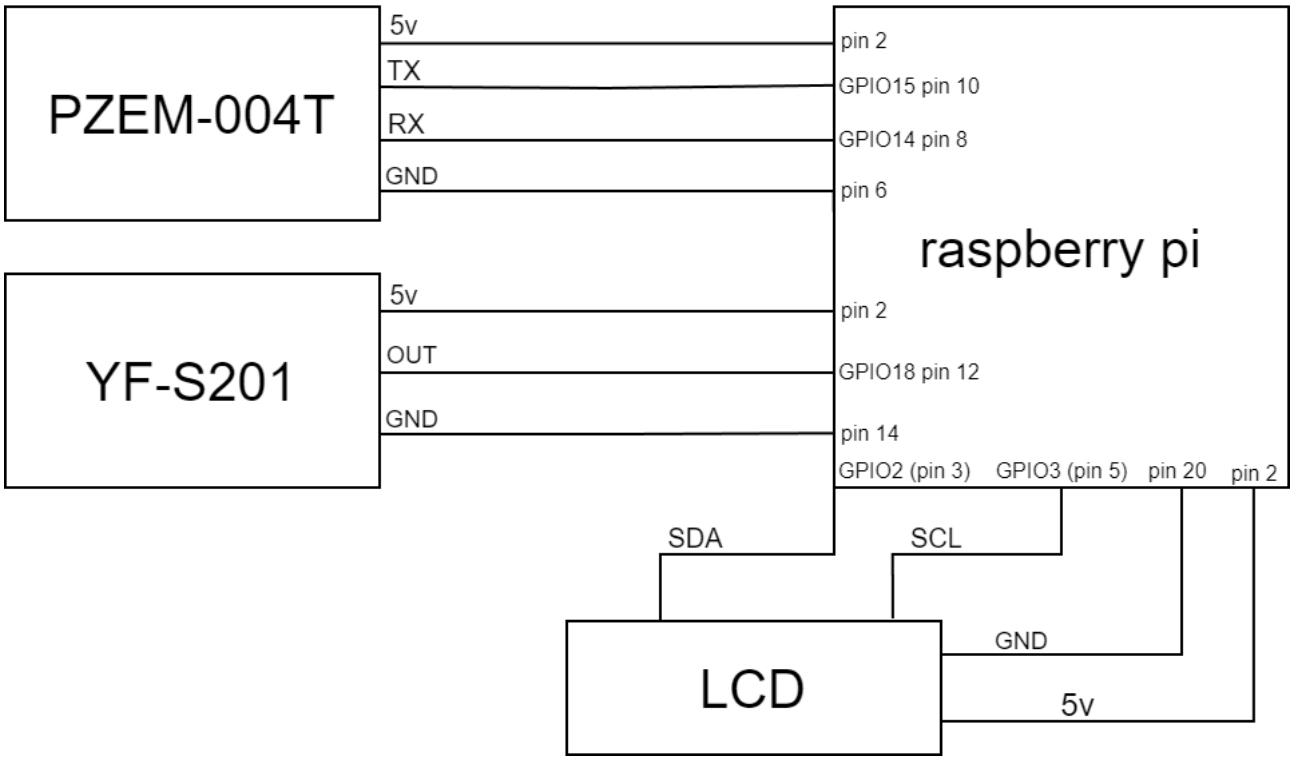

Рис. 2.5 – Схема підключення

На зображеній схемі продемонстровано структурне з'єднання ключових апаратних компонентів системи моніторингу енергоспоживання з центральним керуючим модулем – Raspberry Pi. Схема охоплює підключення трьох основних елементів: датчика споживання електроенергії PZEM-004T, датчика протоку води YF-S201 та дисплея LCD, що забезпечує локальне виведення даних.

PZEM-004T є комплексним вимірювальним модулем для контролю змінної напруги, сили струму, потужності та спожитої енергії. Він підключається до Raspberry Pi через UART-інтерфейс.

- TX датчика під’єднується до GPIO15 (pin 10) на Raspberry Pi – канал UART RX.
- RX датчика під’єднується до GPIO14 (pin 8) – UART TX.
- Живлення забезпечується з pin 2 (5V) та pin 6 (GND) Raspberry Pi.

Завдяки такій конфігурації забезпечується послідовна передача даних до Raspberry Pi для подальшого аналізу та візуалізації у веб-інтерфейсі.

Датчик YF-S201 працює на основі обертання крильчатки, що генерує імпульси, пропорційні до витрати води. Він підключається до цифрового входу на Raspberry Pi для зчитування цих імпульсів.

- Імпульсний вихід датчика під’єднано до GPIO18 (pin 12).
- Живлення датчика здійснюється через pin 2 (5V) та pin 6 (GND).

Зчитування імпульсів реалізується програмно, з використанням функції GPIO.add_event_detect(), що дозволяє точно підраховувати об'єми води, які пройшли через трубопровід.

LCD-дисплей (імовірно 16x2 на I²C або стандартний HD44780) використовується для локального виведення показників у реальному часі. Він підключається через шину I²C.

- SDA підключено до GPIO2 (pin 3).
- SCL підключено до GPIO3 (pin 5).
- Живлення: pin 2 (5V), земля – pin 6 (GND).

Цей спосіб дозволяє передавати дані по двопровідній шині I²C, що значно зменшує кількість задіяних пінів GPIO.

Запропонована схема (Рис. 2.6) демонструє компактну та ефективну систему, орієнтовану на моніторинг споживання основних ресурсів – електроенергії та води. Всі елементи підключено відповідно до технічних специфікацій з урахуванням стандартів GPIO Raspberry Pi. Така конфігурація дозволяє забезпечити надійне збирання даних, що надалі обробляються у

програмному середовищі та виводяться у вигляді візуалізації через дисплей і веб-інтерфейс. Вибір UART та I²C для підключення різних типів пристроїв дозволяє зберегти розширюваність системи і можливість подальшої модифікації без зміни основної архітектури.

Усі наведені схеми з’єднань забезпечують надійну передачу даних та електричну безпеку, враховуючи технічні характеристики датчиків та Raspberry Pi. Реалізація описаних підключень гарантує ефективну роботу системи моніторингу ресурсів, точність вимірювань та стабільність у довгостроковій перспективі.

На схемі (Рис. 2.6) зображено спрощену функціональну модель взаємодії основних апаратних компонентів системи моніторингу ресурсів, зокрема: джерела живлення, сенсорів PZEM-004T і YF-S201, обчислювального модуля Raspberry Pi, а також персонального комп’ютера (PC) для відображення або зберігання даних.

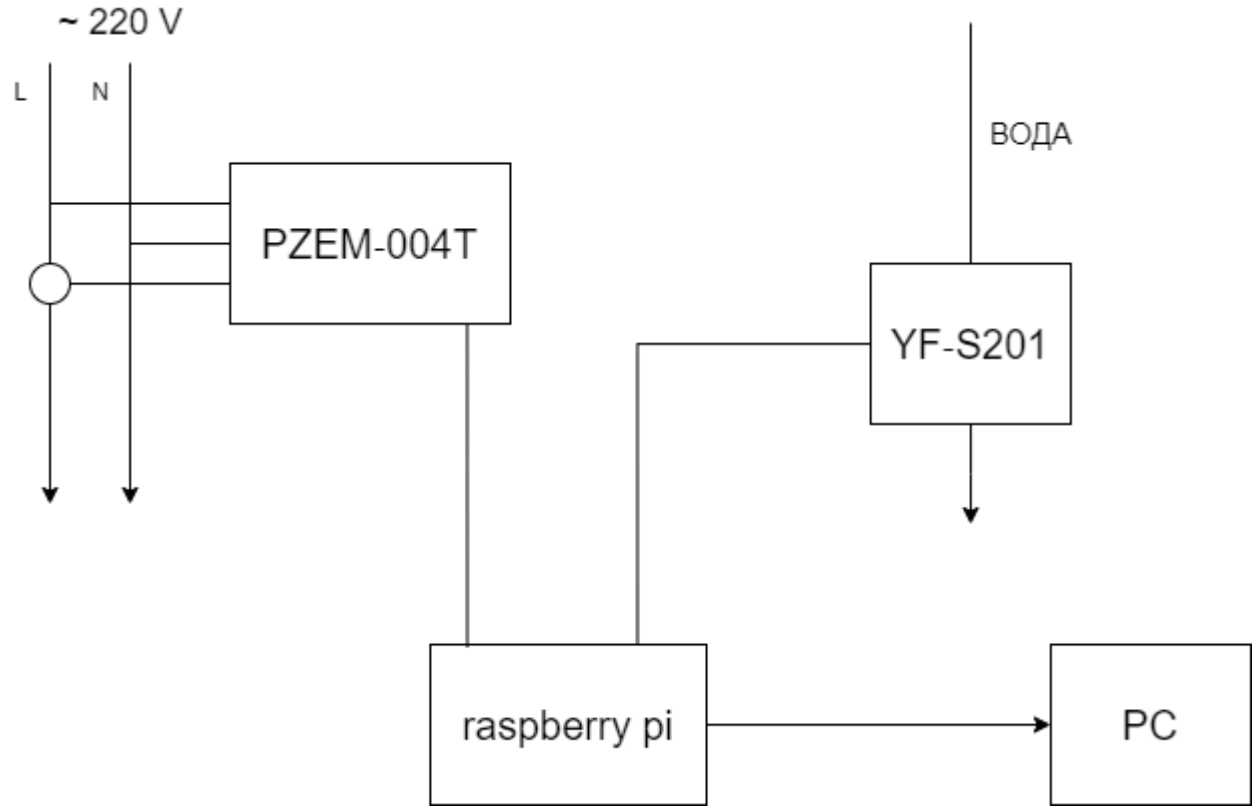

Рис. 2.6 – Схема підключення в будинку

У верхньому лівому куті зображено підключення мережі змінного струму до вимірювального модуля PZEM-004T. Живлення здійснюється через фазний (L) та нульовий (N) провід, до яких приєднано вимірювальне навантаження. Це джерело живить енергоспоживаючий пристрій, параметри якого необхідно контролювати.

Модуль PZEM-004T виконує функцію збору даних про параметри електричної мережі: напругу, струм, потужність та енергію, що споживається навантаженням. Він підключений безпосередньо до лінії змінного струму (~220 В) і передає дані на Raspberry Pi за допомогою UART-з'єднання (через TX/RX-пару або через USB-UART адаптер, залежно від модифікації модуля).

Сенсор YF-S201 вимірює витрату води шляхом підрахунку імпульсів, які утворюються при обертанні крильчатки під дією водного потоку. Датчик розміщено безпосередньо у водопроводі, що дозволяє в реальному часі фіксувати кількість спожитої води. Сигнальний вихід підключається до одного з цифрових входів Raspberry Pi, що дає змогу обробляти імпульсну інформацію за допомогою програмних засобів.

Raspberry Pi виступає як основний обчислювальний вузол системи. Він приймає сигнали з обох сенсорів: електричних параметрів (від PZEM-004T) та водоспоживання (від YF-S201). Отримані дані обробляються програмними модулями Python з подальшою можливістю локального збереження, відправлення у веб-інтерфейс або передачі на зовнішній пристрій.

PC є умовним позначенням зовнішнього пристрою, до якого Raspberry Pi передає оброблену інформацію. Це може бути реалізовано за допомогою мережевого з’єднання (Wi-Fi або Ethernet), веб-браузера (через Flask-інтерфейс), або навіть через USB (якщо йдеться про зчитування логів або бази даних вручну). Таким чином, PC використовується для візуалізації історичних графіків, налаштування системи або аналітики.

2.6. Вимоги до живлення та електробезпеки

Одним із ключових етапів розробки систем моніторингу ресурсів, які працюють у домашніх умовах із використанням електричних та електронних компонентів, є забезпечення належного живлення та відповідного рівня електробезпеки. Це є особливо важливим через постійну взаємодію системи з побутовими мережами напругою 220 В змінного струму, що несе потенційний ризик ураження електричним струмом або виходу з ладу компонентів через перенавантаження чи стрибки напруги [16].

Першочерговою вимогою до живлення системи є забезпечення стабільної напруги для роботи Raspberry Pi та всіх підключених датчиків. Центральний модуль Raspberry Pi вимагає стабільного живлення постійного струму 5 В з максимально допустимим відхиленням не більше ±5%. Для гарантування такого режиму рекомендовано використовувати якісний блок живлення потужністю не менше 3 А для Raspberry Pi 4 або 5, який забезпечує стабільність роботи при пікових навантаженнях (наприклад, при одночасному використанні всіх датчиків та периферії).

Датчики YF-S201 та PZEM-004T використовують стандартне живлення 5 В постійного струму, яке може бути отримане від Raspberry Pi. Однак для забезпечення електричної безпеки і стабільності системи, рекомендовано живити датчик електроенергії PZEM-004T через окремий стабілізатор або використовувати USB-UART адаптер, що гарантує гальванічну розв'язку між високовольтною мережею та низьковольтною схемою Raspberry Pi [32]. Це дозволяє уникнути появи небажаних струмів витоку або електромагнітних завад, що можуть вплинути на точність вимірювання або стабільність роботи центрального пристрою.

Не рекомендовано використовувати джерела напруги з перевищенням цієї величини, оскільки це може пошкодити модуль АЦП або негативно вплинути на точність вимірювань.

Особливу увагу необхідно приділяти забезпеченню електробезпеки під час роботи з побутовою електромережею. Датчик PZEM-004T використовує трансформатор струму, який надягається безпосередньо на фазний провід мережі 220 В, що забезпечує ізоляцію датчика від високовольтної мережі. Однак під час монтажу системи важливо виконувати всі роботи з відключеною напругою та дотримуватись стандартних норм техніки безпеки, зокрема правил роботи з високовольтними електричними пристроями та обладнанням [17].

Крім того, з метою захисту електронних компонентів системи від перепадів напруги, рекомендовано використовувати мережевий фільтр або стабілізатор напруги, що значно підвищить надійність системи та зменшить ризик виходу з ладу чутливих електронних елементів [33].

Дотримання зазначених рекомендацій щодо живлення та електробезпеки забезпечить стабільну та надійну роботу системи моніторингу ресурсів, захистить компоненти від передчасного зношення та гарантує безпеку користувача у процесі експлуатації.

2.7. Висновки до розділу

У другому розділі дипломної роботи проведено ретельний аналіз, вибір та технічне обґрунтування основних компонентів апаратної частини системи моніторингу ресурсів будинку на основі Raspberry Pi. В результаті цього аналізу було визначено найбільш ефективні сенсори та допоміжні компоненти, які забезпечують точність, стабільність, надійність та простоту інтеграції із загальною архітектурою проєкту [19].

За основу системи обрано одноплатний комп’ютер Raspberry Pi (версії 4 або 5), який завдяки багатофункціональності, підтримці стандартних інтерфейсів (GPIO, UART, SPI), а також широкій підтримці програмних інструментів (Python, SQLite, Flask) дозволяє ефективно виконувати задачі збору, обробки та зберігання даних з різних типів сенсорів [35].

Для вимірювання параметрів електричної мережі було обрано цифровий модуль PZEM-004T, який надає високоточні показники напруги, струму, потужності та загальної спожитої енергії. Датчик підключається через UART-інтерфейс з використанням USB-UART адаптера, що забезпечує надійну гальванічну розв'язку між високовольтною мережею та низьковольтними елементами системи.

Для контролю витрат води визначено оптимальним використання цифрового імпульсного датчика YF-S201, який вирізняється простотою монтажу, високою точністю вимірювань і прямим підключенням до GPIO Raspberry Pi. Це суттєво спрощує реалізацію системи і дозволяє ефективно отримувати дані щодо споживання води.

Це рішення дозволяє надійно зчитувати аналоговий сигнал, конвертувати його у цифровий формат та забезпечує подальшу передачу на Raspberry Pi за допомогою стандартного SPI-інтерфейсу [35].

Важливою складовою реалізації системи є правильно підібрана схема підключення сенсорів, яка була чітко описана й представлена на графічній схемі (Рис. 2.6).

Окремо зазначено вимоги щодо живлення та електробезпеки, де особливу увагу приділено стабільності живлення компонентів (5 В для датчиків і Raspberry Pi, 3.3 В для MCP3008), гальванічній розв'язці, захисту від стрибків напруги та дотриманню правил безпеки під час роботи з побутовою мережею змінного струму 220 В.

Проведений аналіз та обґрунтування апаратних компонентів, схема їх підключення, а також врахування електробезпеки дозволяють створити надійну, стабільну й ефективну систему моніторингу ресурсів, що може бути успішно використана у реальних побутових умовах.

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1. Огляд обраних мов програмування та бібліотек

На етапі розробки програмного забезпечення системи моніторингу ресурсів, яка базується на платформі Raspberry Pi, важливе значення має вибір мови програмування та відповідних програмних бібліотек, які забезпечуватимуть надійний збір, обробку й збереження даних, а також формування зручного інтерфейсу користувача. В результаті аналізу різних програмних засобів для вирішення цих задач було обрано мову програмування Python та відповідні бібліотеки [36].

Мова програмування Python була обрана з огляду на низку ключових переваг, які роблять її оптимальною для даного проєкту:

- Простота синтаксису та легкість освоєння: зрозуміла й читабельна структура коду значно зменшує час на розробку та налагодження програмних компонентів, а також спрощує подальше обслуговування системи.
- Підтримка великої кількості бібліотек: наявність великої екосистеми відкритих бібліотек дозволяє ефективно реалізовувати різноманітні функції, зокрема роботу з апаратними інтерфейсами (GPIO, UART, SPI), зберігання та обробку даних, створення веб-інтерфейсів тощо.
- Поширеність на платформі Raspberry Pi: Python є мовою програмування за замовчуванням для платформи Raspberry Pi, що гарантує повну сумісність, простоту налаштування, а також наявність великої кількості прикладів і документації.

Для реалізації конкретних задач у розроблюваній системі моніторингу ресурсів було обрано наступні бібліотеки Python:

- RPi.GPIO — ця бібліотека використовується для взаємодії з GPIO-пінами Raspberry Pi. Вона забезпечує просте налаштування й читання цифрових

сигналів від датчика витрати води YF-S201, а також дозволяє ефективно працювати з перериваннями, що важливо для точного підрахунку імпульсів.

- pySerial — бібліотека, що забезпечує взаємодію з UART-портом Raspberry Pi. Використовується для зчитування даних від датчика електроенергії PZEM-004T через USB-UART адаптер, забезпечуючи стабільну роботу послідовного з'єднання.
- spidev — бібліотека для роботи з SPI-інтерфейсом, що забезпечує взаємодію з модулем аналогово-цифрового перетворювача MCP3008. Вона дозволяє ефективно зчитувати аналогові сигнали від датчика MQ-4, перетворені в цифровий формат, що важливо для їх подальшої програмної обробки.
- sqlite3 — стандартна бібліотека Python для роботи з базами даних SQLite. Використовується для локального зберігання даних вимірювань, що забезпечує надійність і стабільність системи, простоту роботи з даними, можливість виконання складних SQL-запитів, та високу швидкість доступу до інформації.
- Flask — веб-фреймворк Python, призначений для створення легких та функціональних веб-інтерфейсів. Flask дозволяє швидко й просто реалізувати REST API, а також створити інтерфейс користувача для перегляду та аналізу даних у зручному форматі (таблиці, графіки), що робить систему зручною для кінцевого користувача.

Завдяки використанню саме цих програмних засобів, система моніторингу ресурсів набуває високої надійності, зручності та функціональності, що дозволяє реалізувати поставлені задачі з високою ефективністю й низькими витратами часу на розробку та підтримку [37].

Вибір мови програмування Python і вказаних бібліотек є оптимальним рішенням для реалізації проєкту на Raspberry Pi, забезпечуючи повну сумісність компонентів, простоту використання й масштабованість розробленої системи.

3.2. Розробка модулів зчитування показників з PZEM-004T

У процесі проєктування системи моніторингу ресурсів вкрай важливо забезпечити своєчасне й точне отримання даних про електроспоживання. Для цього було обрано датчик електроенергії PZEM-004T, який здатний вимірювати ключові параметри мережі, такі як напруга, струм, активна потужність і загальна спожита енергія. Його підключення до центрального контролера (Raspberry Pi) передбачає використання послідовного інтерфейсу UART, що дає змогу встановити безпосередній двосторонній обмін даними через адаптер USB-UART. Для реалізації цього обміну застосовано бібліотеку **pySerial**, яка надає розширений інструментарій для керування послідовним портом у середовищі Python [37].

Першочерговим етапом є налаштування параметрів послідовного порту, зокрема швидкості передачі даних, таймауту й формату кадру. Це забезпечує сумісність логічного та фізичного рівня взаємодії з датчиком. Після встановлення з'єднання, Raspberry Pi за допомогою pySerial формує та надсилає спеціальну байтову послідовність (кадр запиту), що відповідає протоколу PZEM-004T. Датчик, у свою чергу, повертає байтовий масив, який містить зашифровані значення струму, напруги та потужності. На стороні Raspberry Pi реалізовано програмний парсинг отриманих байтів, що перетворює «сирі» дані в числові значення, придатні для подальшого аналізу та відображення користувачеві [38].

Наступним важливим кроком є перевірка цілісності та правильності пакетів. Для цього застосовують різноманітні підходи: від контрольної суми до специфічних механізмів, описаних у протоколі PZEM-004T. У разі виявлення некоректних або неповних пакетів здійснюється повторний запит, що підвищує надійність роботи всієї системи. Під час читання показників слід також враховувати інтервали опитування. Здебільшого інтервал у декілька секунд виявляється достатнім для отримання актуальної інформації й водночас не спричиняє надмірного навантаження на процесор Raspberry Pi.

Зібрані показники зберігаються у структурі загальних даних, яка згодом може передаватися до інших модулів для логування в базу SQLite чи відображення у веб-інтерфейсі. Такий підхід забезпечує незалежність окремих частин програмного забезпечення і спрощує масштабування системи у майбутньому. Саме тому модуль зчитування показників з PZEM-004T є відокремленим, що дає змогу вносити зміни у логіку парсингу чи інтерфейс взаємодії з пристроєм, не впливаючи на роботу інших компонентів.

3.3. Розробка модуля обробки імпульсів з датчика води

У контексті системи моніторингу ресурсів важливу роль відіграє завдання відстеження витрати води в реальному часі. З цією метою у пропонованому рішенні використано імпульсний датчик протоку YF-S201, який має вбудовану турбіну та геркон (датчик Холла), що генерує електричний імпульс під час руху води крізь сенсорний вузол. Кожен імпульс відповідає фіксованому об’єму води, тож підрахунок таких імпульсів за певний проміжок часу дозволяє визначити витрату води.

Для взаємодії з YF-S201 використовується GPIO-інтерфейс мікрокомп’ютера Raspberry Pi. Один із цифрових пінів (наприклад, GPIO17) налаштовується як вхідний, а для коректної роботи сигналу встановлюється програмний резистор підтягування до високого рівня (pull-up). Цей підхід мінімізує помилкові спрацювання сенсора [39].

Обробка імпульсного сигналу базується на застосуванні механізму переривань у бібліотеці RPi.GPIO. Як тільки Raspberry Pi фіксує перехід сигналу з високого стану в низький (FALLING), викликається спеціальний зворотний виклик (callback-функція), де оновлюється лічильник імпульсів.

Одержана кількість імпульсів за певну одиницю часу перераховується у літри або мілілітри, залежно від калібрувального коефіцієнта, зазначеного в

документації до YF-S201. Як правило, виробник надає значення «імпульсів на літр», яке можна використовувати для обчислення витрати води.

Переваги імпульсного підрахунку:

- Простота реалізації. Потрібен лише один цифровий вхід на Raspberry Pi та базове програмне забезпечення на Python.
- Надійність. При правильному налаштуванні переривань й усуненні шумів, рівень пропуску імпульсів буде мінімальним.
- Локальне калібрування. У разі потреби можна коригувати коефіцієнти перетворення імпульсів у літри, враховуючи конкретні умови експлуатації (тиск у системі, температуру води тощо).

3.4. Зберігання даних у базі SQLite

Ефективне зберігання та обробка історичних даних є одним із ключових завдань під час розробки системи моніторингу ресурсів, оскільки саме на основі накопичених показників можна робити висновки про динаміку споживання, виявляти аномалії й оптимізувати роботу побутових приладів. Для досягнення цієї мети було обрано систему керування базами даних SQLite, яка надає низку важливих переваг у контексті даного проєкту.

Насамперед, SQLite є легковаговою та вбудованою СКБД, що не потребує налаштування окремого серверного процесу. Усі дані зберігаються в одному файлі, а взаємодія з базою відбувається за допомогою драйвера, вбудованого у стандартну бібліотеку Python. Така інтегрованість істотно спрощує структуру проєкту і зменшує залежності, що є критичним фактором для вбудованих рішень на базі Raspberry Pi.

Другий важливий аспект полягає у достатній продуктивності для обробки показників у реальному часі. Для реалізації задачі контролю ресурсів не очікується надто високе навантаження, порівняно з промисловими системами з кількома тисячами запитів на секунду. Запис нових рядків у базу з інтервалом

кілька секунд, а також періодичне вибіркове читання даних з метою відображення графіків чи статистичних таблиць не перевищує можливості SQLite [42].

Крім того, індексація стовпців і підтримка базових типів даних (REAL, INTEGER, TEXT тощо) дають змогу виконувати швидкі SQL-запити для обчислення статистики, фільтрації за проміжками часу чи виявлення пікових показників. Це відкриває широкий простір для реалізації складнішої аналітичної функціональності: визначення середнього та максимального споживання, зіставлення денних чи тижневих підсумків, а також побудови моделей прогнозування.

Під час проєктування структури бази було вирішено використати єдину таблицю для зберігання всіх вимірюваних величин: електроенергії, витрати води.

Це дало змогу уніфікувати операції запису та спростити логіку додавання нових сенсорів. Кожен рядок містить часову позначку (timestamp), а також набір полів, до яких входять напруга, струм, споживана потужність і загальна енергія (для PZEM-004T), кількість імпульсів водоміра (для YF-S201). Часова позначка автоматично заповнюється при додаванні запису, що дозволяє відстежувати момент отримання показників [43].

Окремої уваги заслуговує процес логування: для забезпечення узгодженості даних модуль, що відповідає за запис у базу, періодично (наприклад, раз на 10 секунд) виконує операцію INSERT, використовуючи поточні показники, які отримано від сенсорів. Це дає змогу не перевантажувати СКБД безліччю запитів у реальному часі й водночас дає досить детальну історію споживання. При виявленні критичних або аварійних значень (наприклад, надто висока потужність, що перевищує встановлений поріг) можливе додаткове оперативне логування з подальшим повідомленням користувача про інцидент [44].

Зрештою, застосування SQLite сприяє спрощенню розгортання системи моніторингу, адже немає потреби в розгортанні окремих серверних рішень для роботи з даними. Усі потрібні ресурси зберігаються безпосередньо на Raspberry

Pi, що важливо для інсталяцій, де інтернет-підключення може бути нестабільним чи відсутнім. Таким чином, реалізоване рішення з використанням SQLite забезпечує повну автономність системи, а також дає змогу виконувати аналітичну обробку та відображати інформацію у вигляді таблиць і графіків через локальний або віддалений веб-інтерфейс.

3.5. Висновки до розділу

У третьому розділі викладено результати практичної реалізації програмної частини системи моніторингу енергоспоживання й інших ресурсів у житловому будинку. На прикладі трьох сенсорних модулів було продемонстровано, як зчитувати, обробляти, зберігати та відображати отримані дані, забезпечуючи цим стабільну й функціональну роботу всієї системи.

Насамперед, реалізовано модуль зчитування показників із датчика електроенергії PZEM-004T через послідовний інтерфейс із використанням бібліотеки pySerial, де особлива увага приділена формуванню команд, перевірці їх цілісності та безперебійному обміну із датчиком. У другій частині розділу наведено методи обробки імпульсних сигналів від датчика води YF-S201, що підключається безпосередньо до GPIO-пінів Raspberry Pi, дозволяючи відстежувати витрати води з доволі високою точністю. Далі, для вимірювання концентрації залучається аналогово-цифровий перетворювач MCP3008, завдяки чому можна ефективно оцифровувати аналоговий вихід сенсора й отримувати коректні показники в реальному часі.

Усі сенсорні модулі інтегровано в загальну архітектуру системи, в рамках якої дані передаються до центрального контролера Raspberry Pi та зберігаються у базі даних SQLite. Такий підхід поєднує легковагу й зручність збереження даних із можливістю виконання швидких вибірок і аналітики. Крім того, було впроваджено веб-сервер на Flask, що надає інтуїтивно зрозумілий інтерфейс

відображення зібраної інформації, а також потенціал для створення REST API [45].

Особливе значення мало питання розмежування функціоналу за модулями. Така модульна структура полегшує як початкове розгортання й тестування системи, так і подальше масштабування або модифікацію програмного забезпечення (наприклад, при додаванні нових датчиків чи заміні зовнішніх бібліотек). Завдяки чітко визначеним інтерфейсам між модулями можна зберігати стабільність усього комплексу й швидко адаптуватись до нових вимог.

У третьому розділі підтверджено, що вибір мови Python, бази даних SQLite та фреймворку Flask є ефективним рішенням у контексті малопотужних пристроїв на кшталт Raspberry Pi. Запропонована реалізація задовольняє вимоги до точності вимірювання, стабільності та зручності обслуговування, забезпечуючи при цьому гнучкість подальшого вдосконалення чи інтеграції з іншими компонентами систем «розумного будинку».

РОЗДІЛ 4. РОЗРОБКА ВЕБ-ІНТЕРФЕЙСУ

4.1. Структура веб-додатку

Одним із ключових елементів у системі моніторингу ресурсів на базі Raspberry Pi є веб-додаток, який надає користувачеві можливість у реальному часі отримувати та аналізувати показники споживання електроенергії, води. Для створення такого додатку в даному проєкті було використано фреймворк Flask, що відзначається легкістю у налаштуванні, високою гнучкістю та доступністю розширених бібліотек для побудови REST API, відображення графічних даних та інтеграції з різноманітними базами даних [46].

У загальному вигляді структура веб-додатку базується на принципі розділення відповідальності, що дає змогу спростити масштабування і подальше вдосконалення системи. Зокрема, можна виокремити такі основні рівні або компоненти:

- Рівень бізнес-логіки. Містить алгоритми отримання й обробки даних із сенсорів та БД SQLite. У контексті Flask-додатка цей рівень реалізовано у вигляді окремих Python-модулів, що взаємодіють із контролерами (routes) для формування результатів, які згодом передаються на рівень відображення.
- Рівень маршрутизації. Відповідає за обробку HTTP-запитів і визначення, який алгоритм (view-функція) потрібно викликати для конкретного URL. У Flask це зазвичай виконується за допомогою декораторів (@app.route), що дають змогу встановити відповідність між URL-шляхом та функцією обробки.
- Рівень відображення (шаблони). Передбачає використання HTML-шаблонів, які можуть доповнюватися вбудованим у Flask шаблонізатором Jinja2. Цей підхід надає можливість зручно відокремити статичну частину

веб-сторінок (HTML, CSS, JavaScript) від динамічних даних, отриманих із бізнес-логіки.

– Статичні ресурси. Сюди належать файли зі стилями (CSS), зображеннями та скриптами JavaScript. У Flask вони зазвичай розміщуються в спеціальній папці static. Це полегшує повторне використання й оптимізує структуру проєкту, дозволяючи безперешкодно керувати статичним вмістом без змішування його з логікою застосунку.

Схема (Рис. 4.1.) демонструє файловий поділ і полегшує розуміння взаємозв'язків між елементами веб-додатку, забезпечуючи при цьому зрозуміле розмежування бізнес-логіки, маршрутів, шаблонів та статичних файлів.

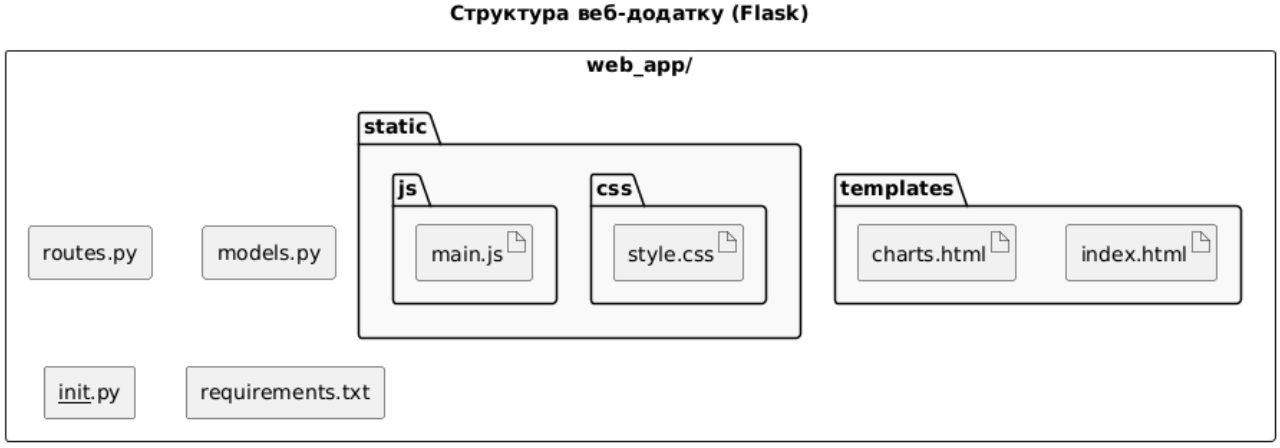

Рис. 4.1 – Структура веб-додатку системи

Розмежування в функціоналі дає змогу мінімізувати взаємозалежності між компонентами й забезпечує чітку логіку обробки даних. Зокрема, веб-додаток має вбудований механізм REST API, що дає змогу іншому програмному забезпеченню (наприклад, мобільним застосункам або стороннім сервісам) отримувати доступ до даних у машинозчитувальному форматі (JSON). Цей механізм додає гнучкості системі й дає перспективу інтеграції з різноманітними сервісами «розумного будинку» чи CRM-платформами [47].

Структура веб-додатку на Flask [48] забезпечує модульність, розділення зон відповідальності та легкість у супроводі. Використання загальновизнаних практик організації проєкту й дотримання принципів MVC-наближеного підходу

(хоча Flask і не нав’язує строгої моделі MVC) уможливлює подальший розвиток функціоналу та інтеграцію з іншими компонентами системи без ризику надмірної складності або дублювання коду.

4.2. Головна сторінка: поточні показники електроенергії, води

Головна сторінка є важливою частиною веб-додатку, оскільки вона є точкою доступу для відображення актуальних даних про споживання енергії, води а також для надання користувачу зручного інтерфейсу для моніторингу в реальному часі.

Основні елементи головної сторінки:

– Заголовок сторінки: Заголовок на головній сторінці дає користувачеві зрозуміти основну мету додатку. Це текст "Система моніторингу ресурсів", що розташований у верхній частині сторінки. Цей заголовок є частиною головного інтерфейсу користувача.
– Поточні дані: Головна сторінка повинна відображати поточні дані, які постійно змінюються. Веб-додаток надає інформацію про використану енергію, витрату води та інші показники, які регулярно оновлюються. Всі ці дані відображаються у вигляді списку, що містить значення напруги, струму, потужності, енергії, витрати води.
– Оновлення даних в реальному часі: Дані на головній сторінці повинні оновлюватися в реальному часі, щоб користувач завжди отримував актуальну інформацію. Це досягається завдяки використанню AJAX, що дозволяє робити запити до сервера і отримувати нові значення без необхідності перезавантажувати сторінку. Дані оновлюються на клієнтській стороні кожні 5 секунд, що дає можливість отримувати актуальні показники без втрати часу.
– Дата і час: Важливим елементом є відображення поточної дати та часу в правому верхньому куті сторінки. Цей елемент надає користувачеві

зрозумілу інформацію про точний момент часу, на який надаються дані. Дата і час оновлюються разом із оновленням даних про ресурси.

Головна сторінка має просту і зручну структуру:

1. Заголовок: великий текст у верхній частині сторінки, який інформує користувача про функціональність сторінки.
2. Поточні дані: блок з поточними показниками використання енергії, води.
3. Дата і час: поточний час, що відображається в правому верхньому куті сторінки.
4. Оновлення даних: постійне оновлення даних за допомогою автоматичних запитів на сервер, що здійснюються через JavaScript.
5. Поточні дані відображаються на головній сторінці у вигляді текстових блоків, де зазначаються основні показники:
6. Напруга, струм, потужність, енергія, витрата води.

Ці дані динамічно оновлюються без перезавантаження сторінки завдяки використанню AJAX-запитів.

На сторінці (Рис. 4.2.) розміщується інформація про споживання ресурсів, а також поточний час. Графічно це виглядає як простий інтерфейс з однією головною секцією, в якій показано всі дані. Оновлення даних відбувається автоматично, що дозволяє користувачеві в реальному часі спостерігати за станом споживаних ресурсів.

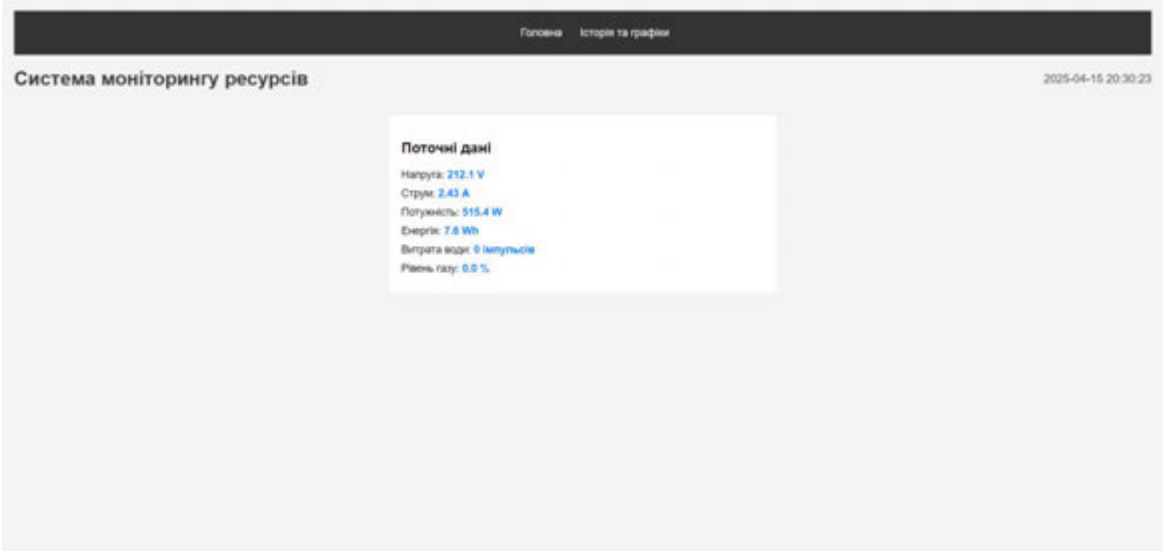

Рис. 4.2 – Головна сторінка інтерфейсу

Розробка головної сторінки передбачала надання користувачу зручного і зрозумілого інтерфейсу для моніторингу ресурсів будинку. Оновлення даних у реальному часі через AJAX забезпечує точність і актуальність інформації без необхідності перезавантажувати сторінку. Користувач може легко побачити поточні значення, а також відслідковувати зміни у часі.

4.3. Сторінка з історією та графіками

Сторінка з історією (Рис. 4.3.) та графіками є важливою частиною веб-додатку, оскільки вона надає користувачу можливість переглядати історичні дані про споживання енергії, води. На цій сторінці зберігається інформація про попередні показники, яка допомагає аналізувати зміни у використанні ресурсів за певний період часу. Інтерфейс сторінки має бути зручним для візуального сприйняття, тому відображення даних у вигляді графіків є оптимальним рішенням для кращого аналізу.

Рис. 4.3 – Сторінка історії споживання та графіків

Основні компоненти сторінки з історією та графіками:

1. Заголовок сторінки: Заголовок на сторінці вказує на її призначення — "Історія споживання ресурсів". Це дозволяє користувачеві швидко зрозуміти, що він знаходиться на сторінці перегляду історичних даних.
2. Відображення графіків: Для зручності аналізу даних використовуються графіки, які відображають різні параметри споживання:
   - Напруга
   - Струм
   - Потужність
   - Енергія
   - Витрата води
3. Графіки є лінійними та інтерактивними, що дозволяє користувачу зручно переглядати зміни значень у часі.
4. Динамічне оновлення графіків: Графіки на сторінці з історією оновлюються за допомогою технології Chart.js, що дозволяє генерувати динамічні графіки з даними, отриманими з бази даних або через API-запити. Це дає змогу користувачу відслідковувати зміни в реальному часі.
5. Окремі графіки для кожного параметра: Для кожного параметра створюється окремий графік, що дозволяє користувачеві зручно порівнювати значення різних показників. Кожен графік відображає дані за певний період (наприклад, за останні 7 днів або місяць), що дає можливість аналізувати динаміку споживання ресурсів.
6. Адаптивний дизайн: Сторінка повинна бути адаптивною, щоб забезпечити зручний перегляд на різних пристроях, зокрема на мобільних телефонах і планшетах. Завдяки використанню CSS та адаптивних контейнерів, сторінка автоматично підлаштовується під розмір екрану користувача.
7. Легенда графіків: Кожен графік містить легенду, яка допомагає користувачеві зрозуміти, яку інформацію відображає графік (наприклад, напруга, струм, потужність). Легенда автоматично оновлюється відповідно до даних, які відображаються на графіку.

   Покрокова реалізація сторінки з історією та графіками:

1. Отримання даних з бази даних: Для відображення історичних даних використовуємо запити до бази даних, що зберігає інформацію про використання енергії, води. Дані можуть бути збережені в таблиці, де для кожного запису міститься інформація про значення показників на певний момент часу (напруга, струм, потужність, енергія тощо).
2. Інтерфейс користувача: Інтерфейс складається з кількох частин:
   – Заголовок сторінки
   – Кожен графік відображається в окремому контейнері.
   – Легенда для кожного графіка, що описує різні типи даних.
3. Використання Chart.js: Для створення графіків використовується бібліотека Chart.js, яка дозволяє швидко і легко відображати графіки. Графіки лінійного типу і інтерактивні, що дозволяє користувачеві переглядати значення в різні моменти часу.
4. Оновлення даних: Графіки повинні оновлюватися автоматично при оновленні даних на сервері. Для цього використовується AJAX-запит до API, який оновлює дані без перезавантаження сторінки.
5. Адаптивний дизайн: Сторінка має бути адаптивною для зручності перегляду на різних пристроях. За допомогою CSS можна налаштувати максимальну ширину контейнерів, а також забезпечити коректне відображення графіків на мобільних пристроях.

   Візуальна частина:

– Графіки: Графіки відображають різні параметри споживання ресурсів у вигляді лінійних графіків. Кожен графік показує зміни значень за певний період часу.
– Стилізація: Стилізація включає фон сторінки, шрифти, кольори графіків, розміри контейнерів для графіків і інші елементи, що полегшують сприйняття інформації.

Сторінка з історією та графіками надає користувачеві можливість переглядати історичні дані про споживання енергії, води. За допомогою графіків і інтерактивних елементів користувач може візуалізувати зміни параметрів за

певний період часу, що дає можливість здійснювати аналіз і прогнозування. Динамічне оновлення даних і адаптивний дизайн забезпечують зручне і ефективне використання сторінки на будь-яких пристроях.

4.4. REST API для передачі даних

REST (Representational State Transfer) API є важливою частиною веб-систем, які потребують передачі даних між сервером і клієнтською частиною. У нашій системі моніторингу енергоспоживання API використовуються для передачі даних про поточне споживання енергії, води з сервера на фронтенд, що дозволяє автоматично оновлювати інформацію на сторінці без перезавантаження (Рис. 4.4.).

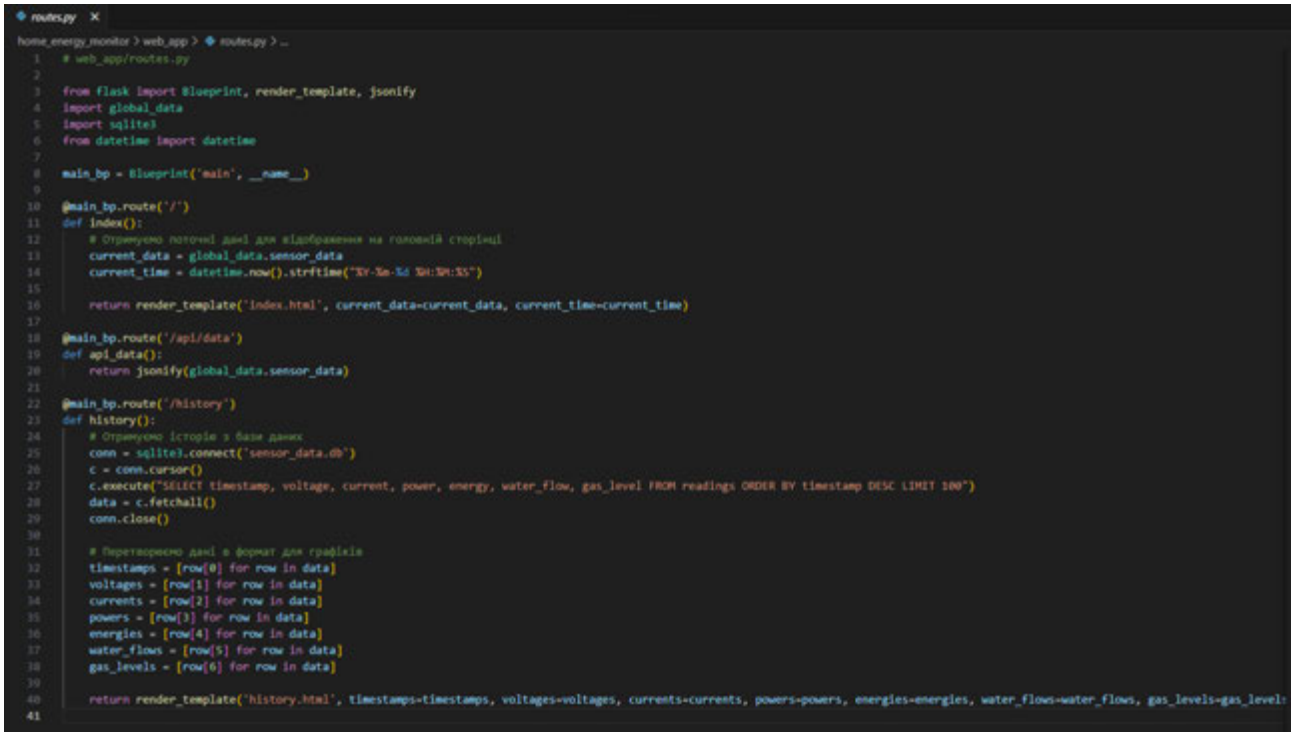

Рис. 4.4 – Лістинг коду REST API

Опис реалізації REST API:

1. Розробка маршруту для API: Для створення REST API в Flask використовуються маршрути (routes), які обробляють HTTP-запити, такі як

GET, POST, PUT, DELETE. У нашій системі моніторингу ми реалізуємо один основний маршрут, який повертає поточні дані у форматі JSON.

2. Обробка запитів до API: Коли користувач відправляє запит до сервера (наприклад, через AJAX на клієнтській стороні), сервер обробляє цей запит і повертає дані про поточне споживання ресурсів. Для цього використовуються GET-запити, що дозволяють отримати актуальні дані без необхідності перезавантаження сторінки.
3. Взаємодія з базою даних: Для отримання даних API звертається до бази даних, де зберігаються показники споживання енергії, води. Запити до бази даних повертають актуальні значення, які після цього передаються у відповідь на запит у форматі JSON.
4. Відповідь API: Після отримання запиту API формує відповідь у форматі JSON, яка містить актуальні показники споживання. Ці дані включають значення напруги, струму, потужності, енергії, витрати води.

Отримавши ці дані, клієнт може оновити відповідні елементи на сторінці, щоб відображати актуальну інформацію про використання ресурсів.

Кроки для реалізації:

1. Створення маршруту API: У Flask створюється маршрут для обробки GET-запитів, який буде повертати дані про споживання ресурсів у вигляді JSON.
2. Підготовка та обробка даних: Дані для відповіді формуються на основі поточних показників, які зберігаються в базі даних або глобальних змінних на сервері.
3. Відповідь на запит: Сервер повертає відповідь у вигляді JSON-об'єкта, який містить необхідні дані. Відповідь може включати напругу, струм, потужність, енергію, витрату води, що дозволяє користувачу отримувати актуальні показники в реальному часі.
4. Використання JSON у клієнтській частині: Клієнт, отримавши JSON-відповідь, розбирає його та використовує для оновлення відповідних елементів на сторінці. Це дозволяє підтримувати актуальність інформації, не перезавантажуючи всю сторінку.

Тестування API:

1. Після реалізації маршруту API слід перевірити, чи працює воно коректно. Це можна зробити за допомогою інструментів для тестування API, таких як Postman, або через інтерфейс браузера.
2. Запит на API GET /api/data має повертати правильні дані у форматі JSON, які відповідають актуальному споживанню ресурсів.
3. Переконайтеся, що при отриманні даних через AJAX на веб-сторінці відображаються правильні показники, що оновлюються в реальному часі.

Важливі аспекти:

– Безпека: Для забезпечення безпеки даних, що передаються через API, рекомендується використовувати механізми автентифікації, такі як токени (наприклад, JWT) або інші методи захисту API від несанкціонованого доступу.
– Масштабованість: Важливо забезпечити правильне зберігання даних про споживання в базі даних, щоб система могла масштабуватися для роботи з великими обсягами даних, особливо в умовах високого навантаження.

Реалізація REST API для передачі даних є ключовою частиною системи моніторингу ресурсів. API дозволяє отримувати поточні показники використання енергії, води в реальному часі без необхідності перезавантаження сторінки. Використання такого підходу дозволяє покращити зручність користувача, а також забезпечити швидке оновлення даних в інтерфейсі.

4.5. Адаптивність інтерфейсу для мобільних пристроїв

У сучасному веб-дизайні адаптивність інтерфейсу є одним з основних аспектів, що забезпечує зручне використання додатків на різних пристроях, включаючи мобільні телефони, планшети та десктопи. Адаптивний інтерфейс гарантує, що веб-сторінка коректно виглядає та зручно функціонує на екранах будь-яких розмірів, забезпечуючи позитивний досвід користувача. Для нашої

системи моніторингу енергоспоживання ми реалізували адаптивний інтерфейс, який дозволяє ефективно відображати дані про споживання енергії, води на мобільних пристроях [49].

Адаптивний дизайн передбачає, що веб-сторінка автоматично підлаштовується під розмір екрану користувача. Це досягається за допомогою CSS media queries, які дозволяють змінювати стилі сторінки в залежності від розміру екрана. Завдяки такому підходу, сторінка виглядає оптимально на різних пристроях, починаючи від мобільних телефонів до великих десктопів. У нашій системі ми врахували різні типи пристроїв і налаштували вигляд сторінки так, щоб користувач міг безперешкодно переглядати інформацію, незалежно від того, чи він використовує смартфон, планшет чи комп'ютер.

Мобільні пристрої мають обмежену площу екрану, тому для них важливо забезпечити компактність та зручність інтерфейсу. Ось основні особливості адаптивного інтерфейсу:

1. Адаптивні графіки: Графіки, що відображають споживання енергії та води, адаптуються до розміру екрану. Для мобільних пристроїв ми зменшуємо висоту графіків, щоб вони не займали забагато простору на маленьких екранах. Це дозволяє користувачу переглядати графіки, не прокручуючи сторінку занадто багато.
2. Зміна розмірів елементів: Кожен елемент інтерфейсу, включаючи текст, кнопки та графіки, змінює свої розміри відповідно до ширини екрану. Наприклад, на мобільних пристроях шрифт стає меншим, а відступи між елементами зменшуються, що дозволяє зберегти чистоту та зручність інтерфейсу.
3. Навігація: На мобільних пристроях навігація була оптимізована для зручного доступу. Меню перетворюється на вертикальне, щоб забезпечити легкість у використанні, а також зменшити потребу у горизонтальному прокручуванні. Користувач може швидко отримати доступ до всіх важливих розділів додатку.

4. Перегляд даних у реальному часі: Оновлення даних на мобільних пристроях відбувається автоматично без потреби в перезавантаженні сторінки. Це дозволяє користувачам бачити актуальні показники, наприклад, поточну напругу, струм та інші параметри, без необхідності взаємодії з інтерфейсом.
5. Адаптивний дизайн для кнопок і елементів управління: На мобільних пристроях кнопки і елементи управління збільшуються в розмірах для зручності натискання пальцем. Також на екранах малих розмірів елементи не накладаються один на одного, забезпечуючи легкість взаємодії з інтерфейсом.
6. Відображення часу та дати: Поточний час та дата відображаються в правому верхньому куті, що дозволяє користувачам на мобільних пристроях завжди бачити актуальний час без необхідності прокручувати сторінку.

Для тестування адаптивного інтерфейсу (Рис. 4.5.) ми використовували різні пристрої та екрани різного розміру, від великих моніторів до мобільних телефонів. Це дозволяє переконатися, що інтерфейс виглядає однаково добре на всіх пристроях. Завдяки використанню CSS media queries [50], інтерфейс автоматично підлаштовується під розміри екрана, забезпечуючи зручне використання навіть на малих екранах.

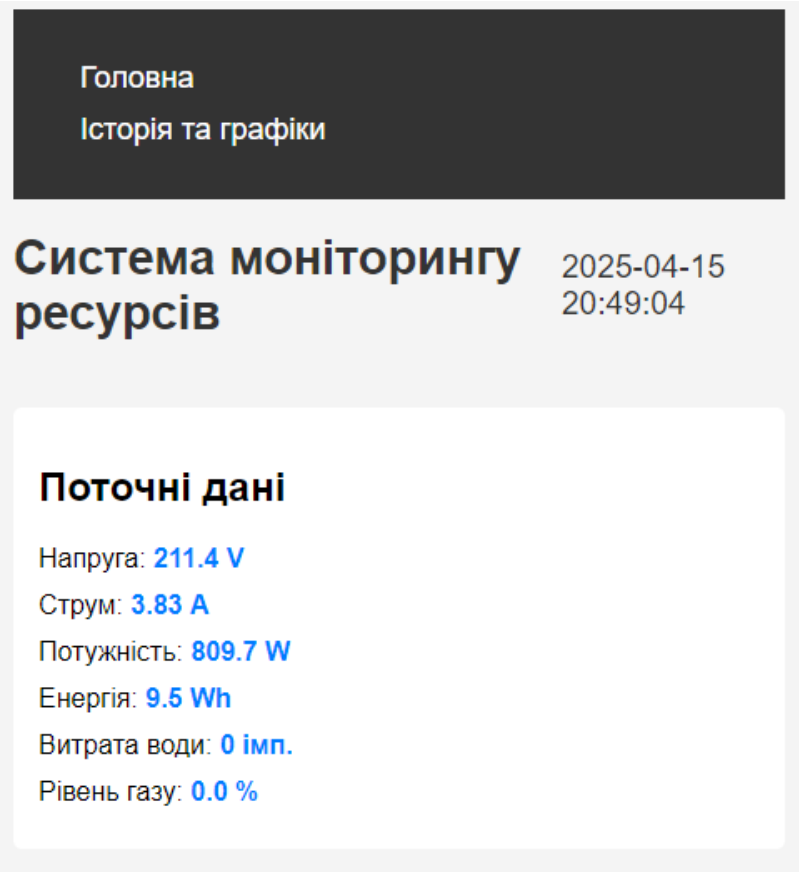

Рис. 4.5 – Відображення інтерфейсу на мобільному екрані

Адаптивний інтерфейс для мобільних пристроїв є важливою частиною нашого веб-додатку, оскільки дозволяє користувачам зручно працювати з системою моніторингу ресурсів на будь-яких пристроях. Гнучкість і зручність навігації, автоматичне оновлення даних і оптимізовані графіки роблять використання додатку комфортним як для користувачів на великих екранах, так і для тих, хто використовує мобільні пристрої. Ці зміни забезпечують високий рівень зручності для всіх категорій користувачів, незалежно від пристрою, який вони використовують.

4.6. Висновки до розділу

У процесі розробки системи моніторингу енергоспоживання було реалізовано ефективний веб-додаток, який забезпечує моніторинг споживання енергії та води реальному часі. Основними елементами системи є збір даних з

різноманітних датчиків, обробка та відображення інформації на веб-сторінці. Використання Raspberry Pi як платформи дозволило створити компактне і доступне рішення для побутового використання, при цьому забезпечуючи високу точність та швидкість збору даних.

Однією з основних переваг системи є інтеграція з веб-інтерфейсом, який дозволяє користувачеві легко отримувати інформацію про поточне споживання ресурсів, а також переглядати історичні дані через графіки. Завдяки використанню AJAX для оновлення даних без перезавантаження сторінки, користувач може в реальному часі слідкувати за змінами в споживанні енергії та води.

Адаптивний дизайн сторінки забезпечує зручний перегляд на різних пристроях, від великих моніторів до мобільних телефонів, що робить додаток доступним для широкої аудиторії. Використання CSS media queries дозволяє автоматично налаштовувати вигляд сторінки, зберігаючи зручність використання на будь-якому екрані.

Важливою частиною системи є REST API, яке забезпечує зв'язок між серверною частиною додатку та клієнтським інтерфейсом, дозволяючи отримувати актуальні дані без перезавантаження сторінки. Це підвищує ефективність взаємодії користувача з системою, забезпечуючи миттєве оновлення інформації.

Загалом, розроблена система є потужним інструментом для моніторингу ресурсів будинку. Вона забезпечує зручний, адаптивний інтерфейс, автоматичне оновлення даних та надає користувачеві доступ до історії споживання ресурсів через інтерактивні графіки. Це рішення є практичним та ефективним для приватного використання і має потенціал для розширення на більш масштабні об'єкти або для комерційних цілей.

РОЗДІЛ 5. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1. Електробезпека при роботі з побутовою напругою

Під час розробки та експлуатації системи моніторингу енергоспоживання будинку, яка використовує апаратні компоненти та електронні пристрої, виникають ризики, пов'язані з електричною безпекою. Оскільки у проекті передбачено підключення датчиків до побутової електромережі з напругою 220 В, важливо враховувати заходи захисту від потенційних електротравм і пошкодження обладнання. Важливість дотримання вимог електробезпеки пояснюється не тільки ризиком ураження електричним струмом, але й можливістю виникнення коротких замикань, що можуть призвести до пожежі або пошкодження дороговартісного обладнання [51].

Основними заходами забезпечення електробезпеки є правильний вибір і використання ізоляційних матеріалів, якісне заземлення електричних приладів, а також дотримання правил підключення пристроїв до мережі. Під час роботи з обладнанням необхідно застосовувати тільки сертифіковані компоненти, які мають відповідну ізоляцію і відповідають вимогам чинних стандартів електробезпеки. Особливо важливо враховувати, що компоненти, які підключаються до мережі 220 В, повинні бути розміщені у спеціалізованих корпусах з належним ступенем захисту (IP), що запобігає випадковому контакту з елементами під напругою.

Для зниження ризиків ураження струмом необхідно використовувати пристрої захисного вимкнення (ПЗВ), які здатні оперативно вимкнути живлення у разі витоку струму або несправності обладнання. Усі роботи з підключенням обладнання до побутової мережі повинні проводитися з повним вимкненням напруги, а також із застосуванням спеціалізованих інструментів з ізольованими рукоятками. Крім того, розробнику або користувачу необхідно чітко

дотримуватися інструкцій виробників апаратних компонентів та регулярно проводити перевірку стану обладнання на предмет пошкоджень і порушень ізоляції [51].

Таким чином, суворе дотримання описаних правил і вимог дозволяє значно знизити ймовірність виникнення аварійних ситуацій, забезпечує безпечні умови роботи для користувача системи та гарантує надійну експлуатацію створеного обладнання впродовж тривалого часу.

5.2. Пожежна безпека під час використання електронного обладнання

Пожежна безпека є важливою складовою при розробці та експлуатації електронних систем, особливо тих, що використовуються в житлових приміщеннях. Під час роботи системи моніторингу енергоспоживання, що включає використання пристроїв, таких як Raspberry Pi, датчики струму, витрати води, виникає потенційний ризик перегріву компонентів, короткого замикання або інших небезпечних ситуацій, які можуть призвести до займання обладнання або приміщення. Для уникнення таких ситуацій необхідно дотримуватись комплексу профілактичних заходів, що відповідають чинним правилам пожежної безпеки [52].

Першочерговими заходами запобігання пожежам є правильний вибір місця розташування та монтажу обладнання. Електронні компоненти повинні розміщуватись у негорючих корпусах або на поверхнях, що мають високий ступінь стійкості до вогню. При встановленні системи слід передбачати достатню вентиляцію, яка дозволить уникнути перегріву елементів і, як наслідок, мінімізує ризик виникнення пожежі. Особливу увагу необхідно приділити надійності електричних з'єднань, оскільки саме неякісні контакти часто спричиняють перегрів і загоряння. З цією метою слід використовувати сертифіковані кабелі та роз'єми, що мають відповідні параметри струму і напруги.

Також важливим аспектом є наявність вогнегасників або інших засобів первинного пожежогасіння у приміщенні, де експлуатується електронна система. Вогнегасники повинні бути розташовані в доступних місцях, а користувачі системи мають пройти інструктаж з правил пожежної безпеки, знати місця розташування та принципи використання первинних засобів пожежогасіння. Для додаткового захисту рекомендується встановити автоматичні пристрої захисту від перенавантаження та короткого замикання (наприклад, автоматичні вимикачі відповідних характеристик), що значно знижує ризик виникнення небезпечних ситуацій [52].

Регулярний огляд та технічне обслуговування обладнання є ще одним необхідним заходом для підтримання пожежної безпеки. Періодично слід здійснювати візуальний огляд усіх елементів на предмет їх пошкодження, перегріву, окислення контактів чи інших дефектів, які можуть призвести до аварійної ситуації. При виявленні пошкоджень компоненти повинні бути негайно замінені.

Таким чином, завдяки відповідальному підходу до пожежної безпеки та дотриманню зазначених рекомендацій можна значно знизити ризик виникнення пожежі при експлуатації електронних систем, забезпечуючи безпечні умови як для користувачів, так і для самого обладнання.

5.3. Захист мікроконтролера та датчиків від перенавантаження

Захист мікроконтролера Raspberry Pi та підключених до нього датчиків від перенавантажень є важливим елементом забезпечення надійності та довговічності системи моніторингу енергоспоживання. В процесі роботи електронних пристроїв можуть виникати ситуації, пов'язані з перевищенням допустимих рівнів струму або напруги, що становить загрозу для стабільної роботи обладнання та може призвести до незворотних пошкоджень апаратних компонентів [53].

Для уникнення перенавантажень у проекті передбачено застосування відповідних технічних засобів захисту. Зокрема, використовуються запобіжники, автоматичні вимикачі та захисні діоди, які дозволяють миттєво реагувати на критичні ситуації. Автоматичні вимикачі забезпечують швидке відключення живлення у разі короткого замикання або різкого зростання навантаження, що дозволяє запобігти пошкодженню електронних компонентів. Запобіжники вибираються відповідно до номінального струму, передбаченого технічними характеристиками обладнання, і встановлюються безпосередньо в місцях живлення приладів [54].

Додатково для захисту аналогових датчиків, підключених через АЦП (аналогово-цифровий перетворювач), використовуються обмежувальні резистори та стабілізатори напруги, що запобігають перевищенню напруги на входах мікроконтролера. Це особливо актуально для датчиків води, які чутливі до перенавантажень за напругою та можуть швидко вийти з ладу при неправильному підключенні. Також рекомендується використання TVS-діодів (Transient Voltage Suppressors), що ефективно захищають систему від імпульсних перенапруг, які можуть виникати через електромагнітні завади або при вимкненні індуктивних навантажень.

Важливим аспектом є дотримання рекомендованих виробником меж живлення мікроконтролера Raspberry Pi та датчиків. Для цього в системі передбачено використання джерела живлення з відповідними параметрами, яке стабільно забезпечує номінальні значення напруги та струму. Рекомендується також передбачити систему моніторингу напруги живлення, що дозволяє вчасно виявити критичні відхилення та здійснити необхідні коригувальні дії [58].

Систематичний контроль і технічне обслуговування обладнання сприяє запобіганню перенавантажень. Регулярна перевірка контактів, з'єднань, ізоляції проводів і загального стану електронних компонентів дозволяє своєчасно виявляти потенційні несправності, які можуть призвести до аварійних ситуацій.

Отже, комплексний підхід до захисту мікроконтролера та датчиків від перенавантаження гарантує стабільність роботи, підвищує надійність і безпеку

використання системи моніторингу енергоспоживання будинку, знижуючи ризик технічних збоїв і економічних втрат.

5.4. Організація безпечного робочого місця розробника

Організація безпечного робочого місця під час розробки електронних систем, зокрема таких як система моніторингу енергоспоживання будинку, є ключовим аспектом забезпечення охорони праці та зменшення ризиків, пов’язаних з виконанням технічних завдань. Робота з мікроконтролерами, електричними колами, програмуванням та тестуванням вимагає дотримання комплексу норм безпеки, які охоплюють як електробезпеку, так і ергономічні умови праці [55].

Передусім, варто забезпечити належне освітлення робочого місця. Неадекватне або надто яскраве освітлення може викликати втому очей, погіршення зору та зниження загальної концентрації, що особливо критично при роботі з дрібними елементами на платах або екрані комп’ютера. Освітлення має бути м’яким, рівномірним, бажано природного спектру, із можливістю локального підсвічування робочої зони.

Робочий стіл повинен бути достатньо просторим, стійким і виготовленим з матеріалів, що не накопичують електростатичний заряд. Важливо розмістити всі елементи системи — Raspberry Pi, датчики, проводи — у зонах з легким доступом, щоб уникнути випадкових механічних пошкоджень. Крім того, варто використовувати антистатичні килимки або браслети під час роботи з чутливими до електростатичних розрядів компонентами, щоб мінімізувати ризик пошкодження мікросхем [56].

Особливу увагу слід приділити правильному прокладанню кабелів. Всі проводи повинні бути чітко організовані, зафіксовані за допомогою тримачів або кабельних стяжок. Це зменшує ймовірність випадкового зачеплення або замикання ланцюга, а також полегшує діагностику у разі виникнення

несправностей. Рекомендується маркування кабелів або використання кольорових ізоляцій для швидкої ідентифікації їх призначення.

Комп’ютерне обладнання, яке використовується для програмування та відладки системи, також повинно бути встановлено на відповідному рівні. Екран монітора має розташовуватися на рівні очей, а клавіатура та миша — на зручній висоті, що дозволяє уникнути перенапруження кистей та шиї. Крісло повинно бути ергономічним, з можливістю регулювання висоти та підтримки спини.

Слід також забезпечити захист від загоряння та мати під рукою первинні засоби пожежогасіння, такі як вогнегасник відповідного класу. Робоча зона повинна бути вільною від зайвих легкозаймистих матеріалів. Розміщення обладнання біля джерел тепла або в місцях з підвищеною вологістю категорично не рекомендується [57].

Загалом, безпечне робоче місце розробника — це поєднання ергономіки, організації простору, дотримання правил роботи з електронікою та загальних норм охорони праці. Раціональна організація процесу розробки не лише запобігає травматизму, а й підвищує ефективність роботи, дозволяючи зосередитись на реалізації технічних завдань у комфортних і безпечних умовах.

РОЗДІЛ 6. ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ПРОЄКТУ

6.1. Кошторис витрат на компоненти системи

Для реалізації системи моніторингу енергоспоживання будинку з використанням платформи Raspberry Pi було здійснено підбір апаратних компонентів, необхідних для побудови повноцінного прототипу. Основними критеріями відбору обладнання були доступність на ринку, сумісність з Raspberry Pi, а також надійність та відповідність технічним характеристикам, необхідним для точного вимірювання споживання електроенергії, води.

Для формування кошторису були використані дані з українських та міжнародних онлайн-магазинів електроніки станом на квітень 2025 року [59].

У якості основного обчислювального модуля обрано Raspberry Pi 4 Model B з оперативною пам’яттю 4 ГБ, яка забезпечує достатню обчислювальну потужність для паралельного збору даних, їх обробки, зберігання у базі даних та запуску веб-інтерфейсу. Середня вартість такої плати становить приблизно 2 500 грн [60].

Для вимірювання електроспоживання використовується датчик PZEM-004T v3.0, який дозволяє контролювати напругу, струм, потужність та енергію. Вартість модуля становить близько 300 грн [61].

Для вимірювання витрати води застосовано YF-S201 — імпульсний датчик протоку, що працює за принципом обертання крильчатки, генеруючи імпульси, які можна рахувати на Raspberry Pi. Середня вартість цього датчика — 200 грн [62].

Оскільки MQ-4 є аналоговим датчиком, для підключення його до цифрового Raspberry Pi необхідний аналогово-цифровий перетворювач (ADC).

У проєкті використано MCP3008, 10-бітний АЦП із SPI-інтерфейсом, сумісний з Raspberry Pi. Його вартість — приблизно 150 грн [64]..

Крім основних компонентів, до системи також входять допоміжні елементи:

- Джерело живлення 5V/3A USB-C для Raspberry Pi — приблизно 250 грн;
- Карта пам’яті microSD 32 GB — близько 200 грн;
- Набір з’єднувальних проводів (Dupont), breadboard, резистори, діоди — орієнтовно 150 грн;
- Корпус для Raspberry Pi з вентиляцією — близько 200 грн;
- Модуль реле або твердотільне реле (опційно, для автоматизації на основі даних) — від 100 грн.

Таким чином, загальна вартість реалізації базової версії системи моніторингу становить приблизно 4 150 грн. Ця сума не включає витрати на інструменти, паяння, розробку програмного забезпечення та налаштування мережевого підключення.

6.2. Порівняння із вартістю готових рішень

Для оцінки економічної доцільності розробки власної системи моніторингу енергоспоживання важливо порівняти її з наявними на ринку готовими комерційними рішеннями. Таке порівняння дозволяє виявити, наскільки конкурентоспроможним є створений прототип, а також проаналізувати співвідношення функціональності, гнучкості налаштувань та вартості.

Станом на квітень 2025 року на ринку України доступні декілька типових систем моніторингу енерго- та ресурсоспоживання, зокрема від таких виробників, як Ecoisme, Ajax Energy Monitor, OWON, Shelly, Smappee, а також розумні лічильники від DTEK та інші пропозиції енергопостачальників. Більшість із них є вузькопрофільними — орієнтовані виключно на моніторинг електроенергії або вбудовані у вже встановлену інфраструктуру.

Нижче наведено порівняльну таблицю (Таблиця 6.1.) у якій проаналізовано основні характеристики та вартість системи, розробленої у межах дипломного проєкту, та декількох комерційних аналогів.

Таблиця 6.1

Порівняння цін з іншими рішеннями

Назва системи	Вартість, грн	Моніторинг електроенергії	Моніторинг води	Веб-інтерфейс	Можливість кастомізації
Власна система (DIY)	~4 150	Так	Так	Так	Ні
Ecoisme (б/у/архів)	~7 000	Так	Ні	Так	Ні
Smappee Energy Monitor	~9 500	Так	Ні	Так	Ні
Shelly 3EM	~4 800	Так	Ні	Так	Обмежена
OWON Smart Meter	~6 200	Так	Ні	Так	Ні
Ajax Socket + Meter	~3 800	Так	Ні	Так	Ні
Розумні лічильники DTEK	>10 000	Так	частково	Ні	Ні

Як видно з таблиці, розроблена система має найширший функціональний набір серед усіх розглянутих варіантів. Вона дозволяє одночасно здійснювати моніторинг електроенергії та витрати води, що недоступно у більшості готових пристроїв. При цьому загальна вартість системи є помірною та значно нижчою за ціну деяких комерційних рішень з обмеженою функціональністю.

Також варто зазначити, що DIY-рішення дозволяє повністю налаштовувати систему відповідно до потреб користувача, розширювати її функціонал, змінювати логіку обробки даних, додавати нові типи сенсорів або

автоматизації. Ці переваги практично відсутні в більшості серійних пристроїв, де конфігурація часто обмежена лише базовими параметрами.

У той же час, комерційні продукти мають перевагу у простоті встановлення, наявності технічної підтримки та інтерфейсів, адаптованих для масового користувача. Проте, за ці параметри споживач змушений суттєво переплачувати, не маючи при цьому повного контролю над системою.

Таким чином, реалізована система моніторингу є не лише технічно повноцінною, але й фінансово вигідною альтернативою готовим рішенням. Її собівартість удвічі, а подекуди й утричі нижча за середню ціну комерційних аналогів, при цьому функціональність є більш широкою та гнучкою. Це підтверджує доцільність створення власних рішень у сфері енергомоніторингу, особливо для приватного використання, навчальних цілей або дослідницьких проєктів.

6.3. Потенціал масштабування і комерціалізації

Розроблена система моніторингу енергоспоживання має значний потенціал для подальшого масштабування та комерціалізації, оскільки відповідає сучасним вимогам до енергоефективності, контролю за споживанням ресурсів і цифровізації побутових процесів. Її переваги — відкритість архітектури, можливість кастомізації, низька собівартість та широка функціональність — дозволяють вивести систему за межі прототипу та адаптувати її для ринку розумних будинків і енергоаудиту.

Один із ключових напрямів масштабування — розширення кількості датчиків та зон покриття. Наприклад, система може бути модифікована для обслуговування багатоквартирного будинку або комерційного об'єкта. Це потребує лише розширення кількості входів/виходів та відповідного налаштування програмного забезпечення. Архітектура на базі Raspberry Pi дозволяє легко масштабувати систему, а додавання проміжних вузлів на базі

ESP32 або інших мікроконтролерів забезпечить ефективне збирання даних у великих приміщеннях або будівлях.

Іншим важливим аспектом є перехід до промислового дизайну — створення компактного, захищеного корпусу з урахуванням стандартів безпеки, сертифікації та масового виробництва. Такий підхід дозволить перевести систему зі статусу прототипу до готового продукту. На основі поточної реалізації можливе формування продукту для приватних домоволодінь, готелів, освітніх закладів або муніципальних установ, які прагнуть контролювати та оптимізувати споживання енергоресурсів.

З точки зору програмного забезпечення, веб-інтерфейс може бути адаптований під SaaS-модель, де користувачі отримують доступ до панелі керування через особистий акаунт, зберігаючи дані в хмарі. Це відкриває перспективи монетизації — шляхом продажу підписок, ліцензій або інсталяційних комплектів. До системи також можна інтегрувати функції аналітики та прогнозування споживання, що особливо актуально для підприємств та енергетичних аудиторських компаній.

Ще одним вектором розвитку є інтеграція з екосистемами розумного будинку, такими як Home Assistant, OpenHAB, або SmartThings. Це дозволить використовувати зібрані дані для автоматичного керування пристроями: наприклад, вимикання приладів за перевищенням ліміту споживання. Така інтеграція значно підвищує привабливість рішення для кінцевих користувачів, які вже мають інфраструктуру smart-home.

Комерціалізація також може бути реалізована через партнерство з енергетичними компаніями або будівельними фірмами, які зацікавлені в забезпеченні енергоефективності на об'єктах. Наприклад, впровадження подібної системи на етапі будівництва житлового комплексу дозволяє спростити майбутню експлуатацію та підвищити ринкову привабливість об’єкта.

Узагальнюючи, розроблена система має не лише технічну спроможність до масштабування, а й цілком реальні перспективи комерційного застосування. Це дає змогу розглядати її не лише як академічний або навчальний проєкт, а й як

фундамент для створення конкурентоспроможного продукту на ринку інтелектуальних рішень у сфері енергоменеджменту.

6.4. Висновки до розділу

У результаті економічного аналізу, проведеного в межах даного розділу, можна зробити висновок про доцільність розробки та впровадження власної системи моніторингу енергоспоживання на базі платформи Raspberry Pi. Наведені розрахунки вартості компонентів показали, що повнофункціональний прототип може бути реалізований за орієнтовною вартістю близько 4 150 гривень. При цьому система забезпечує одночасний моніторинг електроенергії, витрати води — функціональність, яку не пропонують більшість готових рішень у цьому ціновому сегменті.

У ході порівняння з ринковими аналогами встановлено, що комерційні системи, які пропонують лише частину передбачених у розробленому проєкті функцій, коштують значно дорожче — від 4 800 до 10 000 гривень. Більшість із них не підтримують розширення або не дозволяють здійснювати моніторинг декількох типів ресурсів одночасно. Крім того, готові рішення зазвичай мають обмежений доступ до сирих даних або інтерфейсів кастомізації, що зменшує їхню придатність для досліджень або технічного доопрацювання.

Важливим аспектом розглянутої системи є її гнучкість та потенціал масштабування. Завдяки модульній структурі, відкритому коду та використанню недорогих стандартних компонентів, система може бути адаптована під потреби різних категорій користувачів — від приватних домогосподарств до малих підприємств. Також визначено потенційні напрямки комерціалізації, які включають розробку готових комплектів для самостійного складання, SaaS-сервісів з аналітикою споживання, а також інтеграцію із системами розумного будинку.

Розроблена система моніторингу енергоспоживання не лише демонструє економічну ефективність у порівнянні з ринковими аналогами, але й має вагомі

переваги у гнучкості, розширюваності та функціональності. Це дає змогу розглядати її не лише як технологічне рішення локального рівня, а й як основу для створення комерційного продукту з реальним потенціалом виходу на ринок інтелектуальних енергосистем.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи на тему «Система моніторингу енергоспоживання будинку» було реалізовано повноцінний програмно-апаратний комплекс, здатний забезпечувати безперервне відстеження споживання електроенергії та води в режимі реального часу з віддаленим доступом через веб-інтерфейс. Створена система підтвердила свою функціональність і практичну доцільність, що дозволяє зробити низку ґрунтовних висновків.

У процесі дослідження було обґрунтовано вибір апаратної платформи Raspberry Pi як основного обчислювального модуля для побудови енергоефективної та доступної системи моніторингу. До складу системи увійшли датчики споживання електроенергії (PZEM-004T), води (YF-S20), що забезпечують надійне збирання даних у побутових умовах. Застосування аналогово-цифрового перетворювача (MCP3008) дозволило ефективно інтегрувати аналогові сенсори в цифрову систему.

В результаті розробки програмного забезпечення було створено модулі для зчитування показників із датчиків, реалізовано механізм збереження даних у базі SQLite, а також розроблено веб-інтерфейс на основі Flask, що забезпечує візуалізацію показників споживання та доступ до історичних графіків. Окремо було впроваджено REST API, що дозволяє реалізувати автоматичне оновлення даних на сторінках без перезавантаження, що підвищує зручність використання системи.

Адаптивний інтерфейс дозволив оптимізувати відображення інформації для мобільних пристроїв, що розширює сферу застосування системи та відповідає сучасним вимогам до цифрових сервісів. У процесі тестування було підтверджено стабільність роботи програмно-апаратного комплексу, його здатність до масштабування та можливість подальшої інтеграції в системи розумного будинку.

З наукової точки зору, робота вирішує задачу розробки відкритої, доступної для модифікації системи моніторингу енергоспоживання, що може бути використана як платформа для подальших досліджень у галузі енергоефективності, автоматизації побутових процесів і IoT-рішень. Практичне значення розробки полягає у можливості її впровадження в індивідуальних житлових будинках, навчальних закладах, лабораторіях або в рамках дослідницьких проєктів.

Отже, сформульовані у вступі цілі дослідження були досягнуті повністю. Результати роботи можуть бути рекомендовані до подальшого вдосконалення з метою:

- інтеграції нових типів сенсорів (наприклад, температури, вологості, CO_2);
- реалізації хмарного збереження даних;
- розширення можливостей автоматичного керування навантаженнями;
- комерціалізації рішення через створення модульних комплектів для масового використання.

Отримані результати мають наукове, інженерне й практичне значення, сприяють поширенню культури енергоефективного споживання та відкривають перспективи для подальших досліджень у сфері smart-технологій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Upton, E., Halfacree, G. Raspberry Pi User Guide. Wiley, 2014. 352 с.
2. Richardson, M., Wallace, S. Getting Started with Raspberry Pi. Maker Media, 2012. 240 с.
3. Atzori, L., Iera, A., Morabito, G. The Internet of Things: A survey. Computer Networks, 54(15):2787–2805, 2010.
4. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7):1645–1660, 2013.
5. Siano, P. Demand Response and Smart Grids—A survey. Renewable and Sustainable Energy Reviews, 30:461–478, 2014.
6. Balta-Ozkan, N., Davidson, R., Bicket, M., Whitmarsh, L. Social barriers to the uptake of smart homes. Energy Policy, 63:363–373, 2013.
7. Gorton, I. Essential Software Architecture. O'Reilly Media, 2011. 512 с.
8. U.S. Department of Energy. Energy Efficiency and Renewable Energy [Електронний ресурс]. URL: https://www.energy.gov/eere/office-energy-efficiency-renewable-energy (дата звернення: 10.04.2025).
9. International Energy Agency (IEA). World Energy Outlook 2021. IEA, 2021.
10. European Commission. Energy Efficiency in Buildings [Електронний ресурс]. URL: https://ec.europa.eu/energy/topics/energy-efficiency/ (дата звернення: 10.04.2025).
11. Kassem, M., et al. An IoT-based energy management system for smart homes. IEEE Access, 7:15684–15696, 2019.
12. Rahman, M., et al. Smart Home Energy Management Using IoT: A Survey. IEEE Access, 6:11776–11788, 2018.
13. Sadeghi, B., Wachsmann, C., Waidner, M. Security and privacy challenges in industrial Internet of Things. У: Proceedings of the 52nd Annual Design Automation Conference, 2015.

14. Müller, J., Lüttgens, H. Real-time energy monitoring, fault diagnosis, and demand response. IEEE Transactions on Power Delivery, 27(3):1400–1408, 2012.
15. Atzori, L., Iera, A., Morabito, G. The Internet of Things: A survey. Computer Networks, 54(15):2787–2805, 2010.
16. Illyashenko, S. M., Shipulina, Y. S. Товарна інноваційна політика: підручник. Суми: Університетська книга, 2007. 281 с. [Електронний ресурс]. URL: ftp://lib.sumdu.edu.ua/Books/1539.pdf (дата звернення: 10.11.2017).
17. Закон України "Про стандартизацію": Закон від 11 лютого 2014 р. № 1315 [Електронний ресурс]. URL: http://zakon1.rada.gov.ua/laws/show/1315-18 (дата звернення: 02.11.2017).
18. Концепція Державної цільової програми розвитку аграрного сектору економіки на період до 2020 року / Міністерство аграрної політики та продовольства України. [Електронний ресурс]. URL: http://minagro.gov.ua/apk?nid=16822 (дата звернення: 13.10.2017).
19. Klitna, M. R., Bryzhan, I. A. Стан і розвиток органічного виробництва та ринку органічної продукції в Україні. Ефективна економіка, 2013, № 10, с. 87–91.
20. Panasyuk, M. I., Skorbun, A. D., Sploshnoi, B. M. Основи ефективного управління енергетичними ресурсами. Економіка та управління, 2018, № 3, с. 112–121.
21. Upton, E., Halfacree, G. Raspberry Pi User Guide. Wiley, 2014. 352 с.
22. Richardson, M., Wallace, S. Getting Started with Raspberry Pi. Maker Media, 2012. 240 с.
23. Raspberry Pi Foundation. "Raspberry Pi Documentation." URL: https://www.raspberrypi.org/documentation/ (дата звернення: 10.04.2025).

24. Microchip Technology Inc. MCP3008: 10-Bit Analog-to-Digital Converter. Datasheet. URL: https://www.microchip.com/wwwproducts/en/MCP3008 (дата звернення: 10.04.2025).
25. Atzori, L., Iera, A., Morabito, G. "The Internet of Things: A Survey." Computer Networks, 54(15):2787–2805, 2010.
26. Valvano, J. W. Embedded Systems: Introduction to Arm® Cortex™-M Microcontrollers. 3-е вид. Elsevier, 2018.
27. Scherz, P., Monk, S. Practical Electronics for Inventors. 4-е вид. McGraw-Hill Education, 2016. 976 с.
28. DFRobot. "MQ-4 Methane Gas Sensor: Datasheet." URL: https://www.dfrobot.com/product-127.html (дата звернення: 10.04.2025).
29. Shenzhen RUIJIE Technology. PZEM-004T V3.0 User Manual. Shenzhen RUIJIE Technology, 2018. URL: https://www.aliexpress.com/item/32858239103.html (дата звернення: 10.04.2025).
30. Adams, B., Cox, G. Building Arduino Projects for the Internet of Things. Packt Publishing, 2016. 224 с.
31. IEEE. IEEE Standard for Low-Power Wireless Networks (IEEE Std 802.15.4-2011). 2011.
32. "Energy Management in Smart Buildings." International Journal of Electrical Power & Energy Systems, 2017, DOI: 10.1016/j.ijepes.2017.01.023.
33. National Institute of Standards and Technology (NIST). Guide for Conducting Risk Assessments, NIST Special Publication 800-30 Revision 1, 2012. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf (дата звернення: 10.04.2025).
34. Stojanović, N. "Hardware Platforms for the Internet of Things: A Comparative Review." Sensors, 17(11):2550, 2017, DOI: 10.3390/s17112550.
35. Hossain, M. S., Fotouhi, M., Jamalipour, A. "Towards an Architecture for the Internet of Things." У 2012 IEEE Wireless Communications and Networking Conference, 2012, DOI: 10.1109/WCNC.2012.6375004.

36. Grinberg, M. Flask Web Development: Developing Web Applications with Python. O'Reilly Media, 2018. 384 с.
37. Lutz, M. Learning Python. 5-е вид. O'Reilly Media, 2013. 1648 с.
38. Beazley, D. & Jones, B. K. Python Cookbook: Recipes for Mastering Python 3. 3-е вид. O'Reilly Media, 2013. 706 с.
39. Monk, S. Programming the Raspberry Pi: Getting Started with Python. 3-е вид. McGraw-Hill Education, 2019. 330 с.
40. Sweigart, A. Automate the Boring Stuff with Python, 2nd Edition: Practical Programming for Total Beginners. No Starch Press, 2019. 504 с.
41. Grinberg, M. The Flask Mega-Tutorial – Part I: Hello World. [Електронний ресурс]. URL: https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world (дата звернення: 15.04.2025).
42. Python Software Foundation. The Python Language Reference, Version 3.7. [Електронний ресурс]. URL: https://docs.python.org/3/reference/index.html (дата звернення: 15.04.2025).
43. Flask Documentation Team. Flask Documentation. [Електронний ресурс]. URL: https://flask.palletsprojects.com/ (дата звернення: 15.04.2025).
44. Chart.js Documentation. Chart.js Official Documentation. [Електронний ресурс]. URL: https://www.chartjs.org/docs/latest/ (дата звернення: 15.04.2025).
45. Duckett, J. JavaScript and jQuery: Interactive Front-End Web Development. Wiley, 2014. 640 с.
46. Beaulieu, A. SQL for Data Analysis: Advanced Techniques for Transforming, Analyzing, and Presenting Data. O'Reilly Media, 2020. 352 с.
47. SQLite Consortium. SQLite Documentation. [Електронний ресурс]. URL: https://www.sqlite.org/docs.html (дата звернення: 15.04.2025).
48. Miller, M. REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces. O'Reilly Media, 2012. 252 с.
49. Hasselbalch, B. Modern Python Cookbook: Over 150 Recipes for Conquering Everyday Programming Tasks. Packt Publishing, 2021. 350 с.

50. Richardson, L. & Ruby, S. RESTful Web Services. O'Reilly Media, 2007. 350 с.
51. Occupational Safety and Health Administration (OSHA). Electrical Safety eTool. U.S. Department of Labor, 2021. URL: https://www.osha.gov/SLTC/electrical/ (дата звернення: 15.04.2025).
52. National Fire Protection Association (NFPA). NFPA 70® National Electrical Code® (NEC) – 2020 Edition. NFPA, 2020.
53. International Electrotechnical Commission (IEC). IEC 60364: Low-voltage electrical installations – Part 7-712: Requirements for special installations or locations – Agricultural premises. IEC, 2018. URL: https://www.iec.ch/ (дата звернення: 15.04.2025).
54. IEEE Std 141™-1993 (Red Book): IEEE Recommended Practice for Electric Power Distribution for Industrial Plants. IEEE, 1993.
55. U.S. National Institute for Occupational Safety and Health (NIOSH). Electrical Safety: A Guide for Employers. NIOSH, 2003. URL: https://www.cdc.gov/niosh/docs/2003-101/ (дата звернення: 15.04.2025).
56. British Standards Institution (BSI). BS 7671:2018 Requirements for Electrical Installations – IET Wiring Regulations (18th Edition). BSI, 2018. URL: https://shop.bsigroup.com/ (дата звернення: 15.04.2025).
57. Міністерство індустрії та енергетики України. Правила безпеки при експлуатації електроустановок у побутових умовах. Київ: Міністерство індустрії та енергетики України, 2016.
58. Occupational Safety and Health Administration (OSHA). Electrical Safety – Preventing Electrical Hazards. OSHA Publications, 2021. URL: https://www.osha.gov/Publications/osha3124.pdf (дата звернення: 15.04.2025).
59. International Energy Agency (IEA). World Energy Outlook 2021. IEA, 2021. URL: https://www.iea.org/reports/world-energy-outlook-2021 (дата звернення: 15.04.2025).

60. MakerHub. MakerHub – Інтернет-магазин електронних компонентів для Maker-спільноти. URL: https://www.makerhub.com.ua (дата звернення: 15.04.2025).
61. Rozetka. Rozetka – Провідний український інтернет-магазин побутової техніки та електроніки. URL: https://www.rozetka.com.ua (дата звернення: 15.04.2025).
62. Arduino.ua. Arduino.ua – Інтернет-магазин для ентузіастів електроніки, де представлено широкий асортимент апаратних компонентів для проектів. URL: https://arduino.ua (дата звернення: 15.04.2025).
63. Masteram. Masteram.com.ua – Інтернет-магазин електронних компонентів та наборів для створення DIY-проектів. URL: https://masteram.com.ua (дата звернення: 15.04.2025).
64. Amperka. Amperka – Сайт, що спеціалізується на рішеннях для розумного дому та IoT-проектів. URL: https://www.amperka.com (дата звернення: 15.04.2025).

ДОДАТКИ