Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Програмно-апаратне IoT-рішення для розумного керування живленням пристроїв від акумулятору
(тема)

Виконав:
здобувач 4 року навчання,
групи КІУКІ-21-5
Олексій БАТУРІН
(власне ім’я, прізвище)

Спеціальність
123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Комп’ютерна інженерія
(повна назва освітньої програми)

Керівник: ст. викл. Яна НІ
(посада, власне ім’я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім’я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп’ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Батуріну Олексію Олександровичу
(прізвище, ім’я, по батькові)

1. Тема роботи Програмно-апаратне IoT-рішення для розумного керування живленням пристроїв від акумулятору

затверджена наказом по університету від “ 26 ” травня 2025 р. № 424 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 17 червня 2025 р.

3. Вхідні дані до роботи 1) значення напруги та струму; 2)акумулятор для живлення; 3) віддалений доступ до веб-сторінок; 4) мікроконтролер ESP32; 5) компоненти: INA226, PZEM-004T, LM2596, OLED-дисплей; 6) сонячний інвертор; 7) порогові значення.

4. Перелік питань, що потрібно опрацювати у роботі
1) аналіз предметної області;
2) дослідження використаних модулів та технологій;
3) програмно-апаратна реалізація;
4) збір пристрою;
5) тестування програмно-апаратної частини;
6) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

Слайд-презентація – 14 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Ознайомлення з літературними	26.05.25-28.05.25	
	джерелами, аналіз та вибір методу		
	вирішення поставленої задачі		
2	Проектування алгоритмів рішення	28.05.25-1.06.25	
3	Розробка програмного забезпечення	1.06.25-7.06.25	
4	Тестування, виправлення програмних помилок	7.06.25-9.06.25	
5	Збір пристрою та тестування функціоналу	9.06.25-10.06.25	
6	Оформлення матеріалів кваліфікаційної роботи	13.06.25-15.06.25	
7	Подання кваліфікаційної роботи керівникові та	15.06.25-16.06.25	
	її попередній захист		

Дата видачі завдання " 26 " травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ ст. викл. Яна НІ
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 92 с., 23 рис., 2 дод., 25 джерел.

НАПРУГА, СТРУМ, ПОТУЖНІСТЬ, ESP32, PZEM004T, INA226, WI-FI, ВЕБ-ІНТЕРФЕЙС, HTTP, МІКРОКОНТРОЛЕР, СЕРВЕР.

Метою кваліфікаційної роботи є розробка та реалізація автономної системи моніторингу та управління електричною потужністю на базі мікроконтролера ESP32, що передбачає реалізацію точних і надійних вимірювань параметрів струму та напруги, забезпечує адаптивне керування навантаженням відповідно до встановлених порогів напруги й потужності та надає веб-інтерфейс для віддаленого моніторингу та конфігурації параметрів.

У ході виконання роботи проведено аналіз апаратних модулів PZEM004T та INA226 для вимірювання параметрів змінного та постійного струму; розроблено програмне забезпечення для ESP32, що забезпечує обробку даних, їхнє перетворення та передачу через HTTP-сервер. Створено веб-інтерфейс з можливістю перегляду в реальному часі показників напруги, струму, потужності та частоти, а також налаштування порогів умикання/вимикання навантаження. Реалізовано механізм автоматичного повторного підключення до Wi-Fi та роботу точки доступу для конфігурації мережевих параметрів. Проведено тестування системи в різних режимах роботи, оцінено точність вимірювань та швидкодію оновлення інтерфейсу. Отримані результати підтвердили надійну роботу рішення та його придатність для інтеграції в IoT-мережі енергоменеджменту.

ABSTRACT

Bachelor’s thesis: 92 pages, 23 figures, 2 appendices, 25 sources.

VOLTAGE, CURRENT, POWER, ESP32, PZEM004TV30, INA226, WI-FI, WEB INTERFACE, HTTP, MICROCONTROLLER, SERVER.

The major goal of this thesis is to develop and implement an autonomous electric power monitoring and control system based on the ESP32 microcontroller. The system provides accurate and reliable measurement of current and voltage parameters, adaptive load control according to user-defined voltage and power thresholds, and a web interface for remote monitoring and parameter configuration.

During the work, hardware modules PZEM004T and INA226 were analyzed for AC and DC measurements, and firmware was created that reads and processes sensor data and serves it over an HTTP server. A real-time web interface was designed to display voltage, current, power and frequency readings and allow threshold adjustments for automatic load switching. An automatic Wi-Fi reconnect mechanism and access-point mode for network configuration were implemented. System tests under various operating conditions demonstrated measurement accuracy and interface responsiveness. The results confirm the reliability of the solution and its suitability for integration into IoT energy-management networks.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ADC – аналого-цифровий перетворювач (англ. Analog-to-Digital Converter)

AP – режим точки доступу (англ. Access Point)

GPIO – універсальний вхід/вихід (англ. General-Purpose Input/Output)

HTTP – протокол передачі гіпертексту (англ. Hypertext Transfer Protocol)

I²C – шина двонаправленої послідовної передачі даних (англ. Inter-Integrated Circuit)

IoT – Інтернет речей (англ. Internet of Things)

JSON – легкий формат обміну даними (англ. JavaScript Object Notation)

MCU – мікроконтролерний пристрій (англ. Microcontroller Unit)

NTP – мережевий протокол часу (англ. Network Time Protocol)

SSR – твердотільне реле (англ. Solid-State Relay)

STA – режим станції (англ. Station)

UART – асинхронний послідовний інтерфейс передачі даних (англ. Universal Asynchronous Receiver/Transmitter)

Wi-Fi – технологія бездротової передачі даних (англ. Wireless Fidelity)

ВСТУП

У сучасному світі, коли стрімко розвиваються інформаційні технології, а увага до екологічної сталості стає все більшою, питання раціонального використання енергетичних ресурсів набувають особливої актуальності. Одним із ключових напрямів інновацій є створення розумних систем управління електроживленням, що дозволяють не лише оптимізувати споживання енергії, а й підвищити ефективність використання альтернативних джерел енергії, а також забезпечити надійність електропостачання.

Особливо гостро це питання стоїть в Україні, де внаслідок військових дій та супутніх руйнувань інфраструктури питання енергетичної безпеки стали надзвичайно важливими. Перебої з електропостачанням змушують багато домогосподарств шукати альтернативні джерела енергії – серед яких провідне місце займають сонячні панелі – а також впроваджувати ефективні методи зберігання електроенергії за допомогою акумуляторних батарей. Проте разом із встановленням такого обладнання постає складне завдання: як максимально ефективно використовувати накопичену енергію, уникаючи при цьому глибокого розряду акумуляторів, що значно скорочує їх термін служби та впливає на надійність всієї системи.

У цьому контексті надзвичайно важливо розробити інтелектуальні системи керування електроживленням, які автоматично регулюватимуть роботу побутових приладів, що не мають першочергового значення, але здатні оптимізувати споживання енергії. Зокрема, такі прилади, як бойлери чи кондиціонери, що характеризуються високим споживанням електроенергії, можна активувати лише у ті періоди, коли акумулятор має надлишковий заряд. Це дозволяє не лише підвищити загальну ефективність використання енергоресурсів, а й знизити навантаження на електромережу та захистити акумулятор від шкідливих розрядів.

Метою цієї роботи є створення програмно-апаратного IoT-рішення для розумного керування живленням пристроїв від акумуляторної батареї сонячної електростанції. Запропонована система забезпечить автоматичне включення та вимкнення навантаження залежно від рівня заряду акумулятора з можливістю віддаленого моніторингу та коригування параметрів. Завдяки цьому рішенням можна значно підвищити енергоефективність та продовжити термін експлуатації акумуляторів, зменшуючи кількість глибоких розрядів і, відповідно, знижуючи потребу в частій їх заміні.

Окрім цього, при експлуатації сонячних електростанцій є ризик виникнення помилок інвертора, коли акумулятор повністю заряджений і немає споживачів для розсіювання надлишкової енергії. В таких ситуаціях інвертор може перейти в аварійний режим або відключитися, що призводить до простою системи та втрати виробленої електроенергії. Запропоноване програмно-апаратне рішення, яке автоматично керує включенням необов’язкових навантажень, дозволить ефективно використовувати надлишок енергії, запобігаючи аварійним станам інвертора.

Крім технічних аспектів, дана тема має й соціально-економічний вимір, адже вона дозволяє користувачам зменшити витрати на електроенергію та підвищити комфорт життя навіть у складних умовах – таких, як воєнний стан або віддаленість від централізованих мереж. Запровадження таких систем створює передумови для розвитку енергетичної незалежності окремих домогосподарств і громад загалом.

У цій роботі буде здійснено аналіз сучасних підходів до управління живленням у системах з акумуляторами, описано особливості функціонування сонячних електростанцій в українських реаліях, а також детально представлено розроблене IoT-рішення, яке включає апаратну частину, алгоритми керування, засоби комунікації та інтерфейс користувача.

Таким чином, створене рішення має потенціал стати ефективним інструментом для підвищення надійності та автономності енергопостачання в українських домогосподарствах, сприяючи їхній енергетичній безпеці.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Глобальні тенденції енергоменеджменту

Упродовж останнього десятиліття енергетичний сектор переживає найшвидшу трансформацію з часів електрифікації. Частка відновлюваних джерел у світовому виробництві електроенергії зросла до 30 % у 2023 р. і, за прогнозом IEA, сягне 46 % вже до 2030 р. (рисунок 1.1) [10]. Зростання генерації з ВДЕ супроводжується різким підйомом попиту на електроенергію: до 2027 р. він збільшуватиметься майже на 4 % щороку завдяки електромобілям, які в сучасному світі стають все більш популярними, центрам обробки даних і тепловим насосам, що еквівалентно щорічному додатковому споживанню країни розміром із Японію [11].

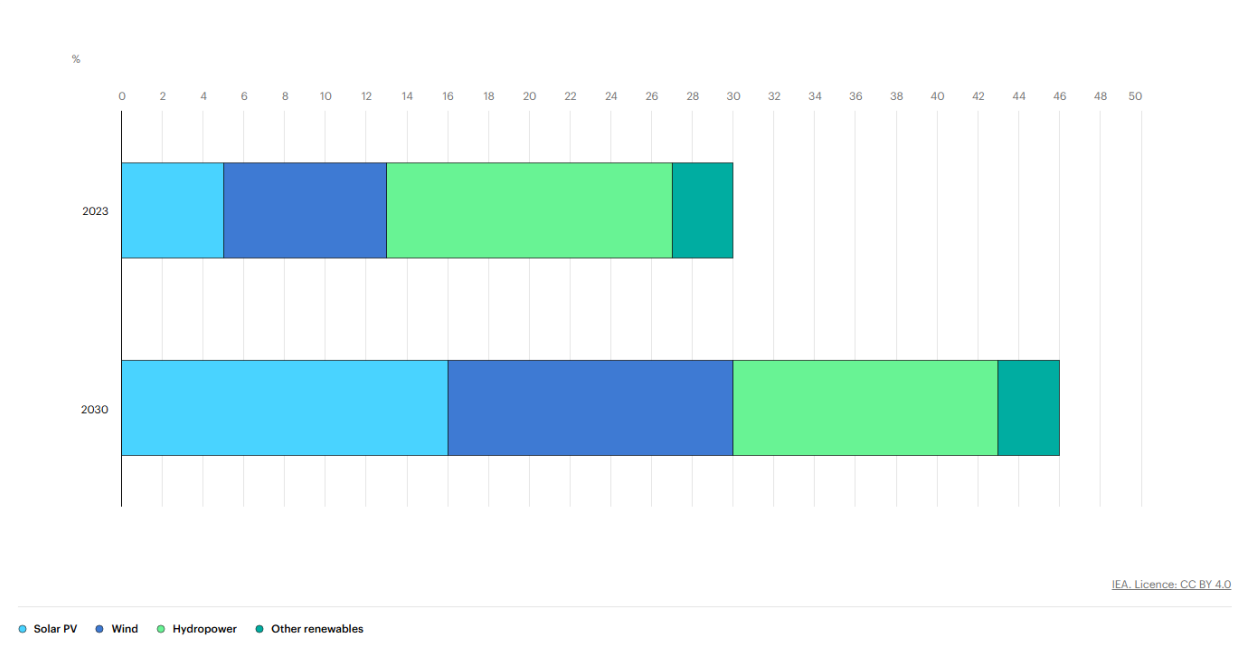

Рисунок 1.1 – Зростання попиту на відновлювану енергію за секторами

Нагальним стає питання гнучкості мереж: BloombergNEF прогнозує, що встановлена потужність систем зберігання енергії зросте до 137 ГВт / 442 ГВт·год уже у 2030 р., підтримуючи середньорічний темп 21 % [7]. Це робить

акумулятори ключовим елементом балансування нестабільних сонячних і вітрових потоків та основою нових бізнес-моделей “виробник – споживач”.

Цифровізація посилює ці зрушення. За оцінкою McKinsey, до кінця десятиліття понад 70 % нових енергетичних активів постачатимуться вже з вбудованими сенсорами та підключенням до хмари, що забезпечує безперервний моніторинг і оптимізацію роботи обладнання [14]. Паралельно швидко зростає ринок IoT-рішень для енергомереж: у 2025-2029 рр. його обсяг перевищить 58 млрд дол., а середньорічний приріст сягне 19 % [20].

Ключовим драйвером залишається політика декарбонізації. З ухваленням “Inflation Reduction Act” у США та “Зеленого курсу” в ЄС обсяги інвестицій у чисту енергетику вперше обігнали витрати на викопне паливо, що пришвидшує запровадження домашніх сонячних електростанцій з акумуляторами [13].

Україна відчуває всі ці глобальні тенденції особливо гостро. Після масових пошкоджень енергетичної інфраструктури в 2024 р. країна влітку стикнулася з дефіцитом генерації 2,3 ГВт при піковому попиті 12 ГВт; електроенергію подавали за графіком, подекуди лише кілька годин на добу [12]. Як відповідь, домогосподарства масово встановлюють приватні сонячні станції разом з накопичувачами: сумарна потужність таких установок перевищила 1,5 ГВт на початок 2025 р. [17]. Це підвищує енергетичну стійкість, проте створює нові технічні задачі – зокрема потребу захищати інвертори від перенапруги, а акумулятори – від глибокого розряду й одночасно ефективно розсіювати надлишкову енергію на другорядні навантаження [1].

Отже, глобальні тенденції – швидке зростання ВДЕ, електрифікація побуту, бум накопичувачів і цифрових платформ – формують чіткий запит на інтелектуальні системи, здатні в режимі реального часу координувати виробництво, зберігання й споживання енергії. Саме в такому контексті розробка програмно-апаратного IoT-рішення для керування домашніми навантаженнями набуває практичної й наукової цінності.

1.2 Домашні альтернативні енергосистеми

Переходячи від загальних концепцій енергозабезпечення та сучасних трендів у відновлюваній енергетиці, розглянемо практичні рішення для побуту: домашня альтернативна енергосистема зазвичай включає фотоелектричні модулі для перетворення сонячного випромінювання на електроенергію, акумуляторний блок для накопичення надлишків для вечірніх піків або аварійних відключень, а також інвертор/зарядний пристрій, який перетворює постійний струм у змінний для побутових приладів та контролює заряд батарей (рисунок 1.2), що в сукупності забезпечує покриття денного споживання і надійне резервне живлення.

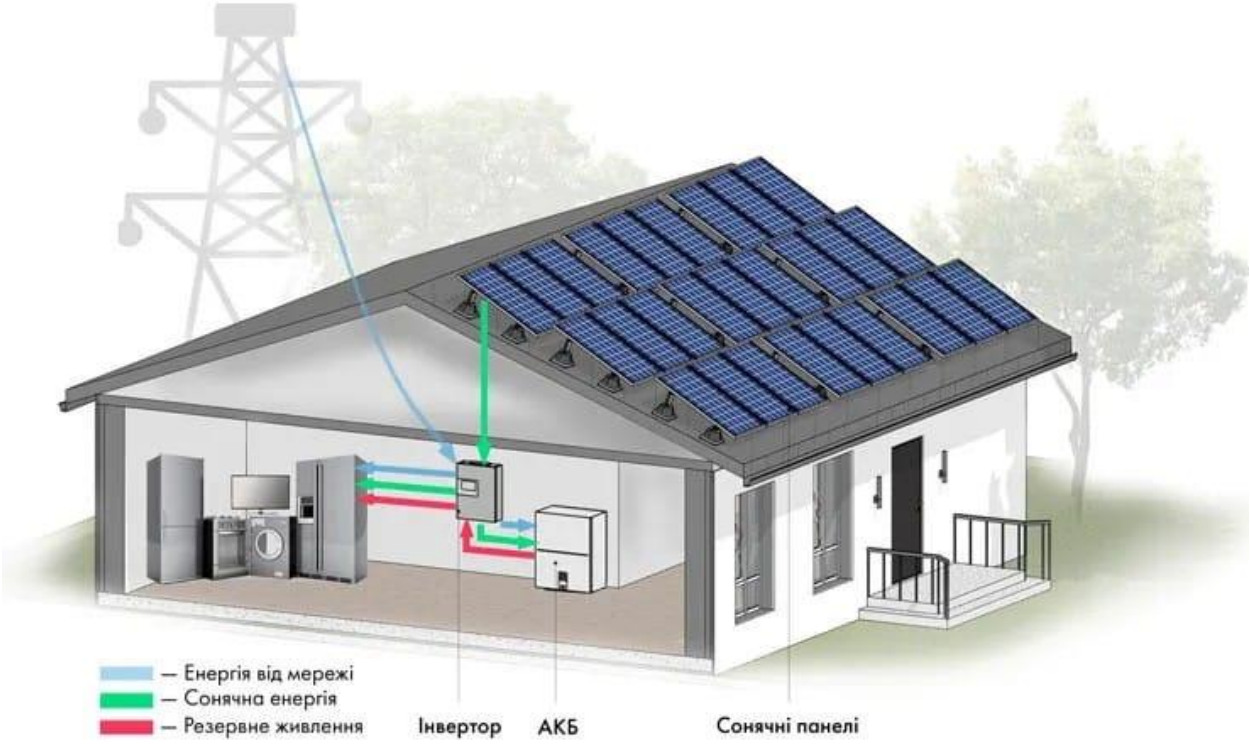

Рисунок 1.2 – Схема домашньої альтернативної енергосистеми

Стислий огляд ключових компонентів і типових проблем, що визначають вимоги до “розумного” керування:

- фотоелектричні модулі. Для дахових установок поширені кремнієві

панелі потужністю 300–550 Вт. У типових українських домах встановлюють масиви до 5 кВт, що дає річну генерацію приблизно 5 МВт·год, достатню для автономного електроживлення і часткового покриття потреб опалення тепловим насосом;

- акумуляторні батареї. Починаючи з 2021 р. основною хімією стаціонарних накопичувачів стала літій-залізо-фосфатна ($LiFePO_4$, LFP) через більшу циклову стабільність та пожежостійкість порівняно з NMC-елементами [15]. При роботі в межах 20–80 % SoC LFP-акумулятори витримують 4000–6000 циклів проти 1000–1500 у свинцево-кислотних, що у 8–10 разів зменшує витрати на заміну протягом життєвого циклу системи [5]. Щоб реалізувати такий ресурс, потрібен Battery Management System, який виконує балансування, вимірює температуру й напругу;

- інвертор/зарядний пристрій. Пристрій перетворює постійний струм АКБ у змінний приблизно 220 В і одночасно керує зарядом. Поширені гібридні моделі 3–5 кВт, які здатні заряджати батарею від сонця або і віддавати енергію в будинок. Найчастіші відмови пов’язані з перенапругою на шині постійного струму, перевантаженням вихідного інвертора та некоректним балансом напруги між ланками, що призводить до аварійних зупинок або виходу з ладу силових модулів [25]. Автоматичне підключення другорядної нагрузки (наприклад, бойлера) при підвищенні напруги на АКБ допомагає розрядити акумулятор до безпечного рівня й запобігти хибним “перевищенням шини”.

Таким чином, домашня альтернативна енергосистема – це взаємодія сонячних панелей, LFP-акумуляторів із BMS та інвертора. Ефективність і надійність такої системи визначає здатність “розумно” відстежувати напругу на батареї, балансувати осередки та підключати другорядні навантаження для розсіювання надлишкової енергії для забезпечення ефективності системи. Це безпосередньо формує технічні вимоги до програмно-апаратного IoT-модуля, що розробляється в роботі.

1.3 IoT-технології розумного керування електроживленням

У сучасних умовах зростання енергоспоживання, популярності автономних енергосистем та актуальності енергозбереження, зростає потреба у гнучких та інтелектуальних рішеннях для керування електроживленням. На ринку вже існує низка готових рішень для моніторингу і керування енергоспоживанням – це, зокрема, розумні розетки, системи розумного будинку, а також комплексні хмарні платформи на зразок Blynk, Home Assistant, Google Home.

Проте такі рішення мають низку обмежень, зокрема:

- залежність від зовнішніх серверів або хмарних сервісів, що знижує автономність системи;
- потреба у постійному підключенні до Інтернету, що створює ризики в разі його відсутності;
- висока вартість окремих компонентів або підписок.

Через це доцільно розробити власну IoT-систему, яка не залежить від сторонніх серверів, працює повністю автономно, забезпечує базову аналітику та гнучке керування живленням побутових пристроїв.

IoT– це концепція, у якій датчики, виконавчі пристрої та програмні сервіси поєднуються мережею і взаємодіють у режимі реального часу. У контексті автономних енергосистем IoT забезпечує три ключові функції: безперервний збір даних про стан обладнання, прийняття рішень згідно з наперед визначеною або адаптивною логікою та віддалений доступ користувача до всіх налаштувань і журналів.

Архітектурно типовий IoT-ланцюжок складається з рівня польових пристроїв, граничного контролера, транспортного протоколу і сервісного рівня (рисунок 1.3). Контролер збирає телеметрію через I²C- або UART-шину, серіалізує її і публікує.

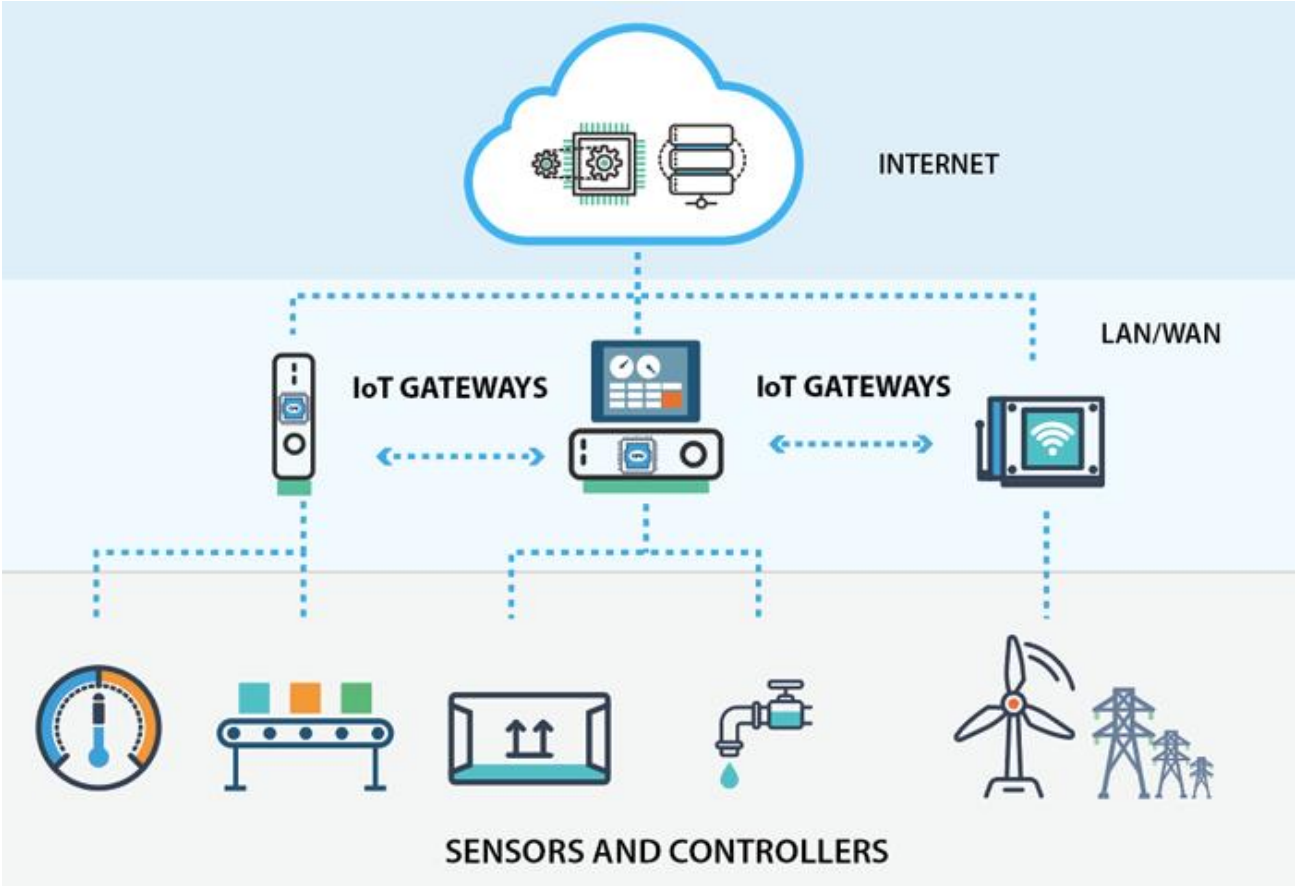

Рисунок 1.3 – Архітектура IoT

Розроблюване рішення реалізує ефективний IoT-підхід без використання зовнішніх серверів або хмар. Уся логіка і взаємодія реалізовані безпосередньо на мікроконтролері ESP32. Пристрій оснащено вбудованим веб-сервером, до якого можна під’єднатися напряму через Wi-Fi, набравши IP-адресу у браузері. Через цей інтерфейс користувач може переглянути поточну напругу, стан реле, а також змінити порогові значення, за якими пристрій вмикає або вимикає бойлер чи інше навантаження.

Контролер зчитує телеметрію (напругу, струм, температуру) через цифрові сенсори за допомогою інтерфейсу I²C та UART. Зібрані дані відображаються на дисплеї та передаються у веб-інтерфейс, де вони оновлюються в режимі реального часу. Передача відбувається за протоколом HTTP, що дозволяє бачити зміну значень без перезавантаження сторінки.

Логіка керування навантаженням базується на простій пороговій схемі з гістерезисом: коли напруга на акумуляторі перевищує встановлений рівень,

пристрій активує твердотільне реле і вмикає пристрій. Якщо напруга падає нижче рівня, реле вимикається. Такий підхід дозволяє уникнути частих перемикань і продовжити ресурс системи. У майбутньому можливе впровадження адаптивної логіки з прогнозуванням енерговиробництва або обліком історичних даних.

З точки зору безпеки, пристрій працює в локальній мережі, тому не потребує зовнішнього підключення до Інтернету. Водночас реалізована базова авторизація для захисту доступу до налаштувань.

Таким чином, пристрій реалізує практичну модель IoT-системи, що функціонує автономно, не залежить від зовнішніх серверів і дозволяє користувачеві керувати живленням побутових приладів у зручний спосіб – напряму через браузер.

1.4 Огляд існуючих рішень для керування живленням пристроїв

Світовий ринок уже пропонує низку рішень для моніторингу та керування домашніми системами “СЕС + акумулятор”, однак жодне з них не закриває повністю потреби типового українського користувача, який прагне недорогої, гнучко керованої та локалізованої системи. Найпопулярніші типи рішень:

- Комерційні інтегровані системи;
- Ринки DIY та open-source;
- Універсальні IoT-платформи;
- Нішеві українські рішення.

Щодо комерційних інтегрованих систем, то найбільш відомі Victron Energy ESS, SolarEdge StorEdge та Tesla Powerwall. Вони постачаються “під ключ”, мають сертифіковані BMS-модулі, веб-портали та мобільні застосунки. Переваги – висока надійність, служба підтримки, гарантії 10 років. Недоліки – ціна (від 900 € за кВт·год), закритий протокол і обмежена можливість підлаштування логіки під локальні тарифи чи нестандартні

навантаження [24]. Для України додатковою перешкодою є складність гарантійного сервісу і часта недоступність “білих” інверторів у роздрібних мережах .

На ринки DIY та open-source найпопулярніші проєкти – OpenEnergyMonitor (рисунок 1.4) [16], LibreSolar та DIYBMS. Вони відкривають код прошивок і схем, дозволяють використовувати недорогі плати ESP32/STM32, а дані публікують у форматах MQTT / InfluxDB. Сильний бік – повна кастомізація і вартість уп’ятеро нижча за комерційні набори. Але для введення в експлуатацію потрібні паяльник, 3D-друк корпусів і ґрунтовні знання електротехніки; документація англомовна, а питання безпеки (сертифікація, захист трансформаторів) лягають на користувача.

Рисунок 1.4 – OpenEnergyMonitor

Щодо універсальні IoT-платформи, то це Home Assistant, Node-RED та Blynk, які забезпечують зручний інтерфейс і автоматизацію “drag-and-drop”; вони підтримують понад 1900 інтеграцій, серед яких інвертори Victron,

GoodWe, Deye. Великий плюс – спільнота, швидкі оновлення та українська локалізація інтерфейсу. Водночас самих “залізних” рішень вони не містять: користувач має окремо купувати датчики, реле й мікроконтролери та самостійно налаштовувати MQTT-брокер, TLS-сертифікати й правила автоматизації.

Також є нішеві українські рішення, наприклад на ринку з’явились готові контролери “SUN UP 3k Smart” і “Solar Monitor UA” та інші девайси без назви, що вміють по Modbus зчитувати напругу інвертора й вмикати ТЕН бойлера. Вони відносно дешеві, однак мають закриту прошивку, суржиковий інтерфейс і не дозволяють змінювати пороги без підключення ПК через RS-485.

Таким чином, існуючі рішення поділяються на дорогі “коробкові” системи, які не дають гнучкості, і дешеві open-source-плати, що потребують високої кваліфікації користувача та є не дуже зручними для користування звичайними юзерами.

1.5 Висновки та формування вимог

По-перше, світове зростання відновлюваних джерел та накопичувачів відбувається на тлі різкого підйому попиту на електроенергію; це формує потребу в децентралізованих “розумних” системах, здатних балансувати виробництво й споживання в реальному часі [10; 12].

По-друге, в Україні після масових ушкоджень мереж домашні СЕС із $LiFePO_4$-акумуляторами стали основою енергетичної стійкості, але потребують захисту інверторів від перенапруги та запобігання глибокому розряду батарей.

По-третє, на ринку існує розрив між дорогими “коробковими” ESS-системами та DIY-проєктами, які вимагають високої кваліфікації; бракує недорогого, локалізованого й водночас безпечного контролера, що керує другорядними навантаженнями без складної серверної інфраструктури.

Функціональні вимоги до розроблюваного пристрою:

- безперервне вимірювання напруги, струму АКБ;
- логіка вмикання/вимкнення пристроїв, що не мають першочергового значення;
- ручне примусове ввімкнення/вимкнення через веб-інтерфейс.
- налаштування порогів через локальний веб-інтерфейс;
- збереження користувацьких налаштувань після оновлення;
- повна працездатність у локальній мережі.

Виконання цих вимог забезпечить стабільну роботу інвертора, продовжить ресурс акумуляторів і надасть користувачеві зручний інструмент для розподілу надлишкової енергії, що є критично важливим у сучасних українських реаліях.

2 АНАЛІЗ ВИКОРИСТАНИХ МОДУЛІВ ТА ТЕХНОЛОГІЙ

2.1 Основні використані модулі

У розробці програмно-апаратного рішення були використані доступні, перевірені на практиці модулі та компоненти, які забезпечують точний моніторинг, стабільну роботу і можливість масштабування у майбутньому. Основні використані модулі:

- мікроконтролер ESP32;
- датчик напруги INA226;
- перетворючав напруги DC-DC LM 2596;
- вимірювач споживаної енергії PZEM-004T;
- твердотільне реле;
- OLED-дисплей.

2.1.1 Мікроконтролер ESP32

Основою керуючого модуля в розробленій системі є мікроконтролер ESP32 від компанії Espressif Systems. Це високопродуктивний 32-бітний чип, побудований на основі двоядерного процесора Xtensa LX6, з тактовою частотою до 240 МГц, а також вбудованими модулями Wi-Fi і Bluetooth. Він має до 520 КБ SRAM, підтримку зовнішньої флеш-пам’яті, апаратне шифрування [19].

ESP32 підтримує численні периферійні інтерфейси, зокрема I²C, SPI, UART, PWM, ADC, DAC, що робить його надзвичайно гнучким і придатним для роботи як з аналоговими, так і цифровими сенсорами. Завдяки такій функціональності ESP32 забезпечує одночасне зчитування показників з датчиків, керування реле, збереження налаштувань та обслуговування веб-інтерфейсу (рисунок 2.1).

У контексті цієї розробки ESP32 виконує всі критично важливі функції. Він зчитує напругу та струм з сенсора INA226 через інтерфейс I²C, приймає рішення про вмикання чи вимикання навантаження на основі заданих порогів і керує твердотільним реле через цифровий вихід. Крім цього, мікроконтролер обслуговує локальний веб-сервер, через який користувач може переглядати поточні параметри системи, змінювати конфігурацію порогів спрацьовування та контролювати стан навантаження. Зміни в налаштуваннях зберігаються у вбудованій енергонезалежній пам’яті, що дозволяє зберігати параметри навіть після перезапуску пристрою.

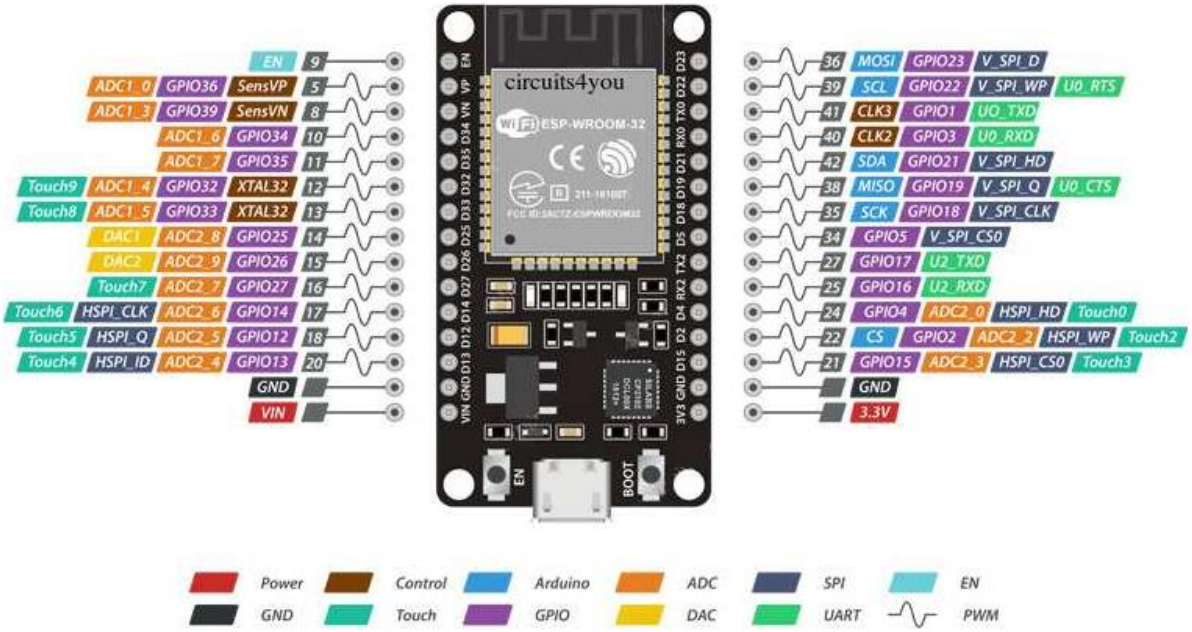

Рисунок 2.1 – Мікроконтролер ESP32

Мікроконтролер ESP32 обрано також через його широку підтримку серед розробників, відкриті бібліотеки, повну сумісність із Arduino IDE, простоту інтеграції з іншими модулями та документовану підтримку роботи з веб-технологіями. Завдяки цьому він забезпечує стабільну роботу системи, надає зручний спосіб взаємодії з користувачем і є гнучкою основою для подальшого розширення функціональності пристрою.

2.1.2 Датчик напруги INA226

Для точного вимірювання напруги, струму та потужності в системі використовується цифровий датчик INA226. Це високоточний монітор енергоспоживання, який підключається до мікроконтролера через інтерфейс I²C і дозволяє в реальному часі отримувати основні електричні параметри [2].

INA226 вимірює як напругу на шині живлення (до 36 В), так і падіння напруги на прецизійному шунті. На основі цих значень пристрій обчислює миттєву потужність, а також дозволяє оцінити середнє енергоспоживання за певний період часу. Точність перетворення становить 16 біт, що забезпечує високу роздільну здатність навіть для невеликих струмів. Це особливо важливо для систем із літієвими акумуляторами, де важливе точне керування зарядом і захист від глибокого розряду [22].

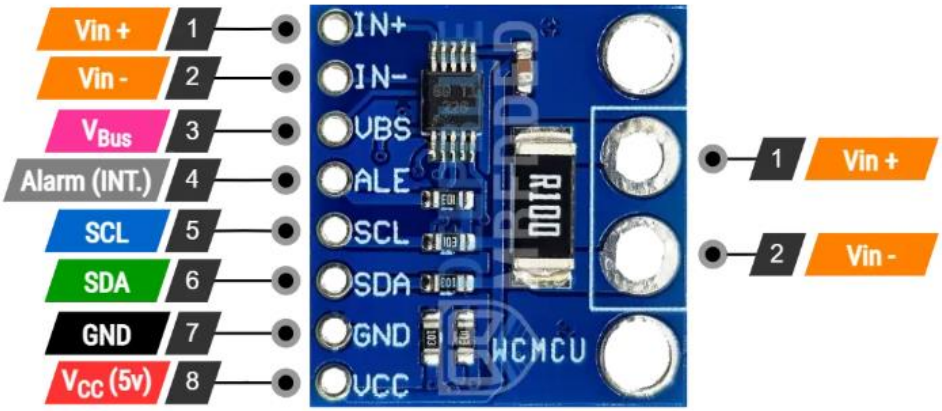

Рисунок 2.2 – Датчик напруги INA226

У рамках розроблюваного пристрою INA226 виконує роль основного сенсора телеметрії. Він надає мікроконтролеру інформацію про поточну напругу на акумуляторі, на основі якої приймається рішення щодо вмикання чи вимикання другого навантаження – бойлера. Додатково вимірюється струм, що відкриває можливість у майбутньому реалізувати підрахунок залишкової ємності акумулятора та вести базовий облік спожитої або збереженої енергії для зручності.

INA226 також підтримує апаратні порогові сповіщення: при перевищенні або зниженні напруги чи струму за межі визначених границь активується вивід ALERT, який можна під’єднати до GPIO-контролера як джерело переривання. Це дозволяє створити апаратну схему захисту або тригери подій незалежно від основного програмного циклу [2].

Базуючись на стабільній роботі, точності вимірювань і простоті інтеграції, INA226 є оптимальним вибором для побудови систем моніторингу в малопотужних автономних системах, особливо при роботі з $LiFePO_4$-акумуляторами, чутливими до перенапруги та глибокого розряду.

2.1.3 Перетворючав напруги DC-DC LM 2596

Для стабілізованого живлення мікроконтролера ESP32 від акумуляторної батареї в системі використовується понижувальний (step-down) імпульсний стабілізатор на базі мікросхеми LM2596. Цей модуль дозволяє перетворювати змінну вхідну напругу в діапазоні приблизно від 7 В до 40 В у стабільне постійне значення 5 В, необхідне для живлення логіки пристрою [23].

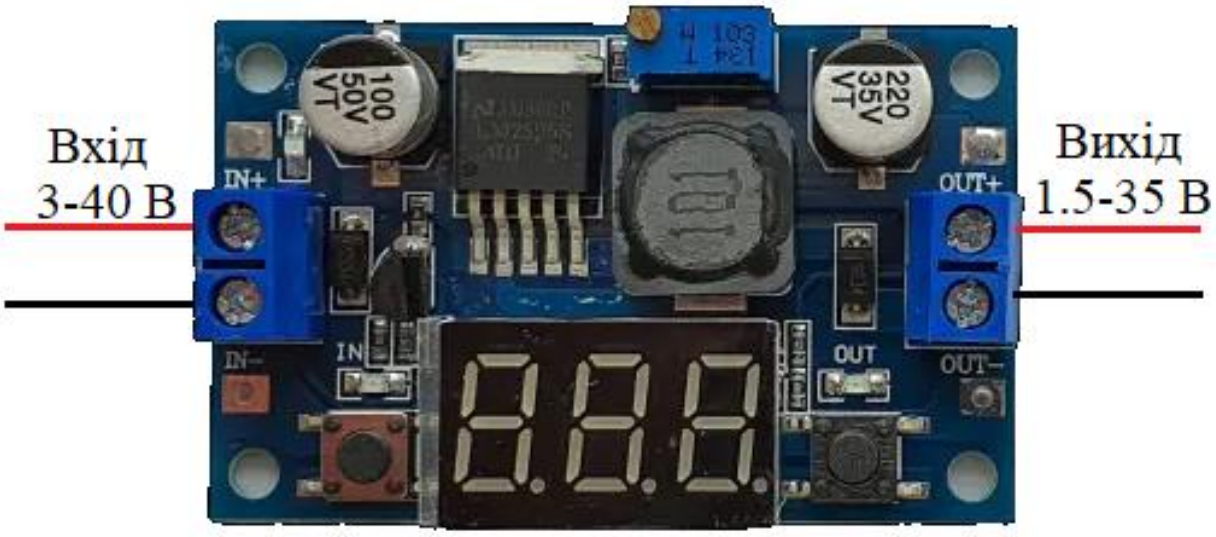

Рисунок 2.3 – Перетворючав напруги DC-DC LM 2596

Основною перевагою LM2596 є високий коефіцієнт корисної дії (до 75–85 % залежно від навантаження), що значно важливіше у системах, які живляться від акумулятора. На відміну від лінійних стабілізаторів, імпульсний перетворювач практично не нагрівається та не витрачає енергію даремно, навіть при великих перепадах вхідної напруги, що робить його дуже гарним рішенням у таких системах.

У розроблюваній системі LM2596 встановлюється між виходом акумуляторної батареї (яка може сягати 28–30 В при повному заряді) та входом ESP32, який вимагає стабільних 5 В для своєї роботи. Вбудований потенціометр дозволяє точно налаштувати вихідну напругу, а модульна версія з зазвичай містить також індикатор роботи, фільтрувальні конденсатори та захист від короткого замикання .

Важливо дотримуватись правильного теплового режиму: при струмі навантаження понад 1 А бажано передбачити мінімальний радіатор або забезпечити вентиляцію. Також для безпеки рекомендовано додати запобіжник або самовідновлюваний PTC-резистор у вхідне коло, щоб захистити стабілізатор від перенавантаження або короткого замикання для більш довгого використання.

LM2596 є оптимальним вибором для цієї системи завдяки простоті налаштування, доступності, низькій ціні та достатній надійності для побутового застосування з навантаженням до 2 А.

2.1.4 Вимірювач споживаної енергії PZEM-004T

Модуль PZEM-004T є ключовим елементом системи моніторингу електричної потужності, оскільки він забезпечує безпосереднє зчитування широкого спектру параметрів змінного струму (рисунок 2.4). Основними перевагами цього пристрою є компактність, відносно висока точність та підтримка стандартного протоколу Modbus-RTU, що спрощує інтеграцію з різними контролерами, зокрема ESP32.

У серці PZEM-004T-100A знаходиться трансформатор струму з коефіцієнтом перетворення, розрахованим на амплітуду до 100 А, а також вбудований аналогово-цифровий перетворювач високої роздільності для напруги (до 260 V) та струму. Завдяки цьому модуль одночасно вимірює миттєву напругу, струм, потужність, накопичену енергію, частоту мережі та коефіцієнт потужності [18].

Комунікація з модулем здійснюється через інтерфейс UART на 9600 бод. Для обміну даними з ESP32 використовуються готова бібліотеки, що реалізують функції повернення всіх необхідних полів.

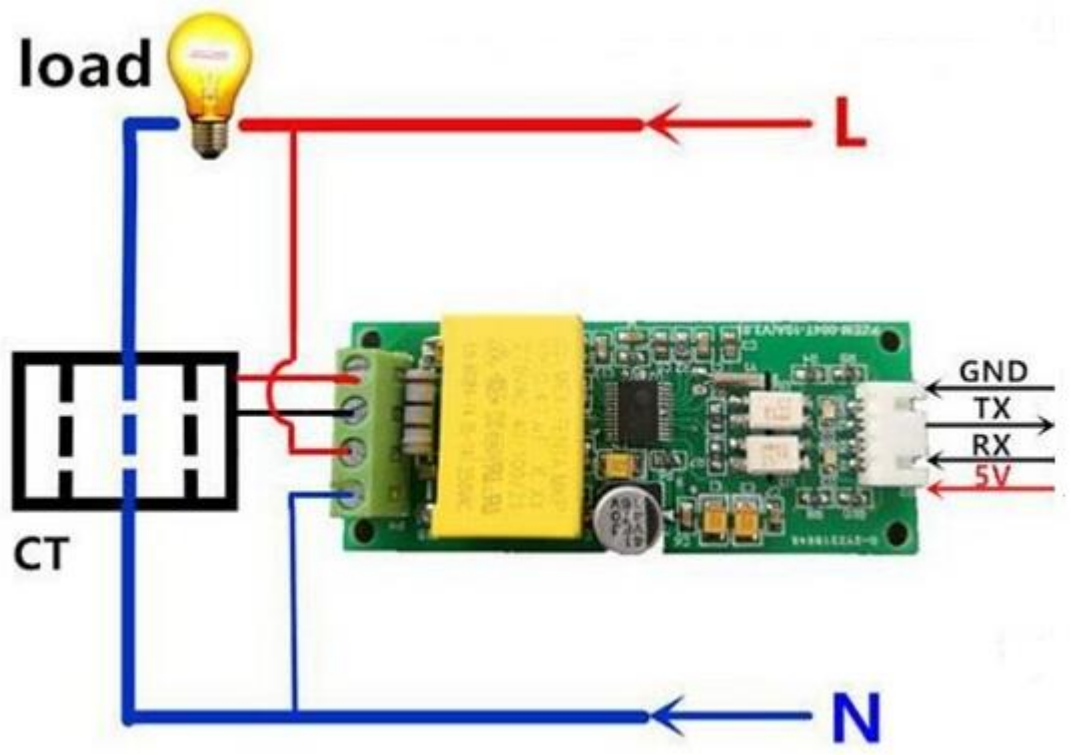

Рисунок 2.4 – Вимірювач споживаної енергії PZEM-004T-100A

У рамках аналізу альтернатив були розглянуті модулі аналогічного класу, але PZEM-004T-100A виявився найбільш збалансованим за ціною, доступністю та спільнотою розробників. Його відкритий вихідний код бібліотек, наявність документації та позитивний досвід інтеграції на інших платформах підтвердили правильність вибору для реалізації нашого проекту.

2.1.5 Твердотільне реле

Для безпечного та надійного керування змінним навантаженням у системі використовується твердотільне реле. На відміну від електромеханічних реле, SSR використовує напівпровідникові елементи і оптопари для гальванічної розв’язки керуючого та силового ланцюгів. Це значно підвищує ресурс роботи пристрою (до десятків мільйонів циклів увімкнення-вимкнення) та усуває вібрацію та електричні перенапруги, характерні для контактів [9].

Керування SSR здійснюється цифровим виходом ESP32, який подає низьковольтний сигнал (3,3 В) на оптопрохідну діодну ланку усередині реле (рисунок 2.5). Оптрон гальванічно відокремлює низьковольтну сторону контролера від високовольтного навантаження, що гарантує безпеку мікроконтролера навіть у разі короткого замикання на виході реле. Силова частина побудована на основі TRIAC-елемента, який спрацьовує лише після проходження синусоїди через нульову точку, що мінімізує електромагнітні перешкоди та плавно вмикає навантаження.

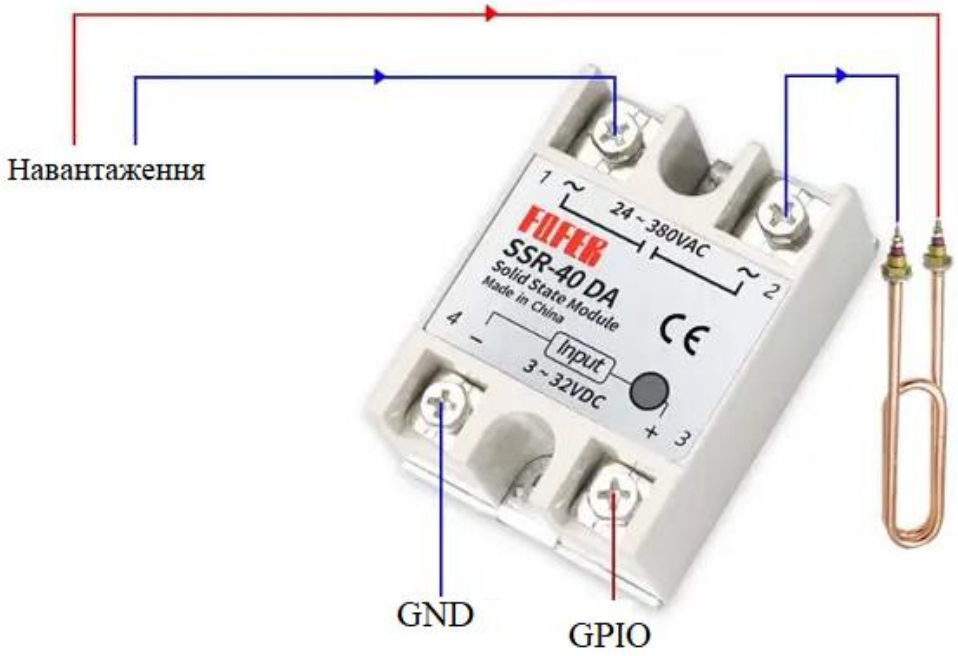

Рисунок 2.5 – Принцип підключення твердотільного реле

Однією з головних переваг SSR-реле є відсутність механічного зносу – реле не клацає, не створює іскор і може перемикатись тисячі разів без втрати ресурсу. Завдяки цьому реле особливо зручне для автоматичного керування навантаженням із частими перемиканнями, наприклад, у режимах з гістерезисом або адаптивною логікою. Водночас через внутрішнє тепловиділення (падіння напруги близько 1–1,5 В на навантаженні) при струмах понад 10 А потрібен радіатор або встановлення на теплопровідну поверхню з термопрокладкою.

Таким чином, застосування твердотільного реле у проєкті гарантує високу надійність і довговічність системи, відсутність контактного зносу та імовірності виникнення дугового розряду, а також дозволяє точно реалізувати стратегії адаптивного керування навантаженням в умовах змінного енергоспоживання.

2.1.6 Дисплей

Для локального відображення поточних параметрів системи – таких як напруга акумулятора, стан реле, встановлені пороги увімкнення та вимкнення, підключення до мережі – використовується мініатюрний дисплей на базі технології OLED. Було використано 0.96-дюймовий OLED-дисплей з роздільною здатністю 128×64 пікселі, що працює через інтерфейс I²C і легко інтегрується з мікроконтролером ESP32 (рисунок 2.6) [26].

OLED-дисплей має низку переваг:

- високу контрастність;
- відсутність підсвітки;
- дуже мале енергоспоживання;
- компактні розміри.

У режимі очікування або сну він споживає менше 0.06 мВт, а оновлення тексту виконується практично миттєво, що робить його ідеальним для застосування в енергоефективних системах.

У даній системі дисплей використовується як зручне доповнення до веб-інтерфейсу. Він дозволяє швидко перевірити поточний стан пристрою та параметрів без підключення до мережі.

Підключення дисплея до ESP32 виконується через стандартні виводи SCL та SDA. Часто використовуються готові бібліотеки, наприклад Adafruit_SSD1306, які мають простий API для виводу тексту.

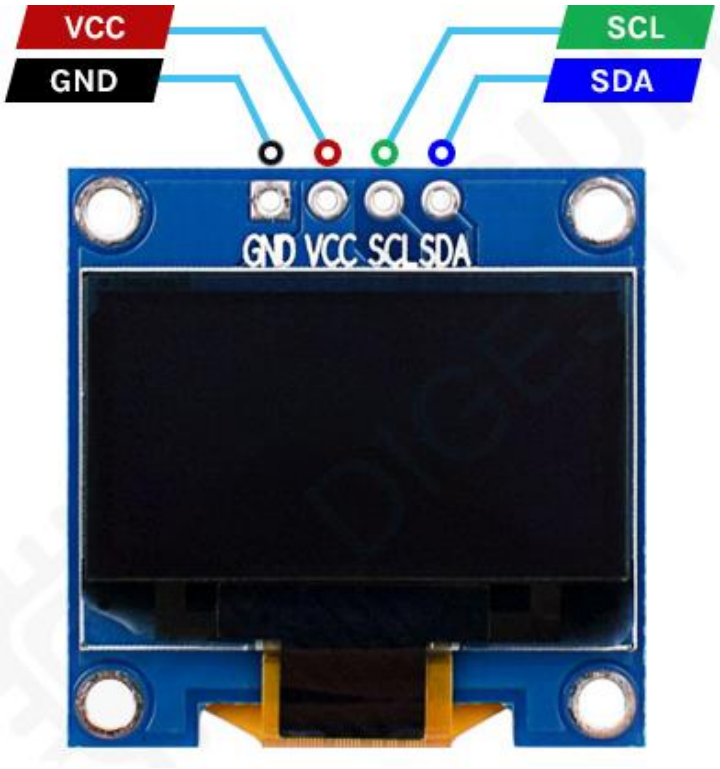

Рисунок 2.6 – 0.96-дюймовий OLED-дисплей

Завдяки простоті реалізації, надійності та низькому енергоспоживанню OLED-дисплей є доцільним рішенням для побутових автономних IoT-пристроїв, де потрібна індикація без зайвої складності.

2.2 Огляд мови програмування

У якості програмного забезпечення була обрана мова C++, яка поєднує в собі високу продуктивність і гнучкість. Це мова системного програмування, яка дозволяє керувати апаратними ресурсами на низькому рівні, одночасно

підтримуючи абстракції високого рівня, такі як класи, шаблони та об’єктно-орієнтоване програмування. Завдяки цьому C++ є основною мовою для розробки програмного забезпечення для вбудованих систем, зокрема мікроконтролерів[21].

У проєкті використовується середовище Arduino IDE, яке забезпечує платформу для компіляції та прошивки коду на мікроконтролері ESP32. Arduino реалізує спрощене застосування C++ з фреймворком для роботи з периферією, таймерами, інтерфейсами (I^2C, UART, SPI) та іншими ресурсами мікроконтролера. Такий підхід дозволяє ефективно розробляти прошивку, навіть при обмежених апаратних ресурсах [3].

Програма керує роботою пристрою: зчитує дані з датчика INA226, аналізує напругу й струм, приймає рішення про вмикання або вимикання бойлера через SSR-реле, залежно від встановлених порогів. Усі параметри можуть бути змінені через локальний веб-інтерфейс, який реалізовано за допомогою бібліотеки ESPAsyncWebServer, що дозволяє обслуговувати HTTP-запити.

Використання C++ дозволило реалізувати як базову логіку контролю, так і розширені можливості: журналювання подій, збереження налаштувань у пам’яті мікроконтролера, індикацію на OLED-дисплеї, обробку помилок. Потенційно можна впровадити алгоритми адаптивного керування або інтегрувати BLE-доступ.

Таким чином, C++ є ефективним інструментом для створення програмної частини енергоефективного IoT-пристрою, забезпечуючи гнучкість, масштабованість і сумісність із сучасною апаратною платформою.

2.3 HTTP-сервер та ESPAsyncWebServer.h

Для взаємодії користувача з пристроєм у режимі реального часу використовується вбудований HTTP-сервер, що працює безпосередньо на мікроконтролері ESP32. Це дозволяє організувати зручний веб-інтерфейс для

перегляду параметрів системи (напруга, струм, стан реле) та змінення налаштувань – зокрема порогів увімкнення та вимкнення навантаження.

Основою серверної частини є бібліотека ESPAsyncWebServer.h, яка дозволяє ефективно обслуговувати HTTP-запити без блокування основного циклу loop() мікроконтролера (тобто в асинхронному режимі). Це особливо важливо у проєктах реального часу, де паралельно працюють датчики, логіка керування реле, дисплей та інші елементи.

Бібліотека підтримує стандартні HTTP-методи (GET, POST), а також передачу даних у форматі JSON, що дає змогу інтегрувати інтерфейс не тільки з браузером, а й із мобільними додатками або REST-клієнтами. Наприклад, GET може повертати поточну напругу, а POST– зберігати нові пороги в пам’ять пристрою [8].

У межах пристрою HTTP-сервер дозволяє підключитися до контролера через локальну мережу (Wi-Fi) або у режимі точки доступу. На веб-сторінці користувач бачить поточні значення, може увімкнути/вимкнути реле вручну, або змінити налаштування.

Таким чином, бібліотека ESPAsyncWebServer.h є гнучким і потужним інструментом для реалізації веб-інтерфейсу на ESP32, що забезпечує зручне налаштування пристрою, моніторинг та підтримку безпечного з’єднання.

2.4 ArduinoIDE

Для написання, компіляції та завантаження програмного коду на мікроконтролер ESP32 у цьому проєкті використовується середовище розробки Arduino IDE. Це кросплатформна інтегрована розробницька система з відкритим кодом, яка широко застосовується у сфері вбудованих систем, навчальних проєктів, прототипування і побутової автоматизації.

Arduino IDE підтримує мову програмування на базі C++ з низкою спрощень, адаптованих до мікроконтролерів. Середовище забезпечує інтуїтивно зрозумілий інтерфейс, просте підключення зовнішніх бібліотек,

автозбереження, серійний монітор і можливість працювати з великою кількістю сумісних плат, серед яких – ESP32 [3].

Для підтримки ESP32 у середовищі Arduino необхідно встановити відповідну плату через Boards Manager. Це надає доступ до специфічних функцій ESP32, таких як керування GPIO, OTA-оновлення, Wi-Fi, SPIFFS, I^2C, а також використання енергоощадних режимів [4].

У контексті даного проєкту Arduino IDE забезпечує повний цикл розробки:

- написання коду з використанням бібліотек;
- компіляція з перевіркою на помилки;
- завантаження прошивки через USB або бездротовим способом.

Усі конфігураційні параметри, такі як швидкість порту, вибір плати, розділ пам’яті або вибір версії ядра, задаються безпосередньо у вікні середовища.

Завдяки великій спільноті, Arduino IDE має десятки тисяч прикладів, що значно спрощує реалізацію типових завдань. Це дозволяє розробнику зосередитися на логіці пристрою, не витрачаючи час на базову інфраструктуру.

Таким чином, Arduino IDE є ключовим інструментом для розробки програмного забезпечення у проєкті, що поєднує простоту використання, надійність і гнучкість у роботі з ESP32.

2.5 Висновки та формування вимог

Загалом, у цьому розділі було детально розглянуто використані апаратні та програмні модулі для створення системи моніторингу та керування навантаженням у домашніх енергосистемах. Вибір мікроконтролера ESP32 обґрунтований його високою продуктивністю, підтримкою Wi-Fi, наявністю численних інтерфейсів та відкритим програмним середовищем, що дозволяє гнучко налаштовувати систему

відповідно до потреб користувача.

Для вимірювання параметрів змінного і постійного струму обрано високоточні датчики INA226 та PZEM-004T, що забезпечують точне зчитування електричних показників у реальному часі. Встановлений перетворювач напруги DC-DC LM2596 гарантує стабільне живлення мікроконтролера, навіть при варіативності вхідної напруги. Твердотільне реле забезпечує надійне та безпечне керування навантаженнями, а OLED-дисплей дозволяє зручний локальний моніторинг стану системи.

Програмне забезпечення, розроблене на мові C++ в середовищі Arduino IDE, включає використання бібліотеки ESPAsyncWebServer для створення асинхронного веб-інтерфейсу, що дає змогу користувачеві в реальному часі змінювати параметри та отримувати актуальну інформацію про стан пристрою. Вибір цього підходу дозволяє досягти високої ефективності роботи системи, зберігаючи її гнучкість і масштабованість.

Завдяки використанню перевірених на практиці компонентів і сучасних технологій розроблена система забезпечує високу надійність, точність вимірювань і зручність в експлуатації, що робить її ефективним рішенням для керування енергетичними ресурсами в домашніх умовах.

3 РЕАЛІЗАЦІЯ ПРОГРАМНО-АПАРАТНОГО ІОТ-РІШЕННЯ

3.1 Апаратна частина

У розділі апаратної частини описано вибір і взаємне підключення основних модулів системи моніторингу та керування потужністю на базі ESP32. Ключовими компонентами є мікроконтролер ESP32, модуль вимірювання змінного струму PZEM-004T-100A, датчик постійної напруги INA226, OLED-дисплей SSD1306, твердотільне реле (SSR) для комутації навантаження й кнопка для локальної навігації інтерфейсом.

Для живлення та обміну даними ESP32 використовує VIN та шинні інтерфейси: UART2 (RX=16, TX=17) для зв’язку з PZEM-004T-100A, I²C (SDA=21, SCL=22) для INA226 і OLED. Твердотільне реле підключене до GPIO 2, що дозволяє безконтактно вмикати або вимикати навантаження напругою до 260 В. Кнопка підключена до GPIO 15 з резистором підтягування внутрішньої конфігурації, її натискання зафіксовує модуль Button із програмним дебаунсом.

Спочатку монтаж усіх компоненті був виконаний на макетній платі з урахуванням зручності доступу до вузлів з’єднання та охолодження. Такий підхід до проектування і монтажу апаратної частини гарантує компактність, надійність з’єднань та зручність подальшого обслуговування пристрою в експлуатації.

3.1.1 Проектування схеми пристрою

Для візуалізації всіх сигналів і живлення була розроблена принципова електрична схема у середовищі EasyEDA (рисунок 3.1). Джерелом живлення служить DC–DC конвертер LM2596, що перетворює вхідні 12-36 В на 5 В для VIN ESP32 і живлення PZEM-004T. Вихід 5 В від LM2596 також подається

на VCC модуля PZEM і на вхід Vin микроконтролера ESP32.

Мікроконтролер ESP32 використовує наступні інтерфейси:

- UART2 (TX2 → PZEM RX, RX2 ← PZEM TX) для зв’язку з PZEM-004T-100A;
- I²C-шину (SDA на D21, SCL на D22) для INA226 та OLED-дисплея SSD1306;
- GPIO 2 для керування твердотільним реле (SSR);
- GPIO 15 для підключення кнопки з підтягуючим резистором.

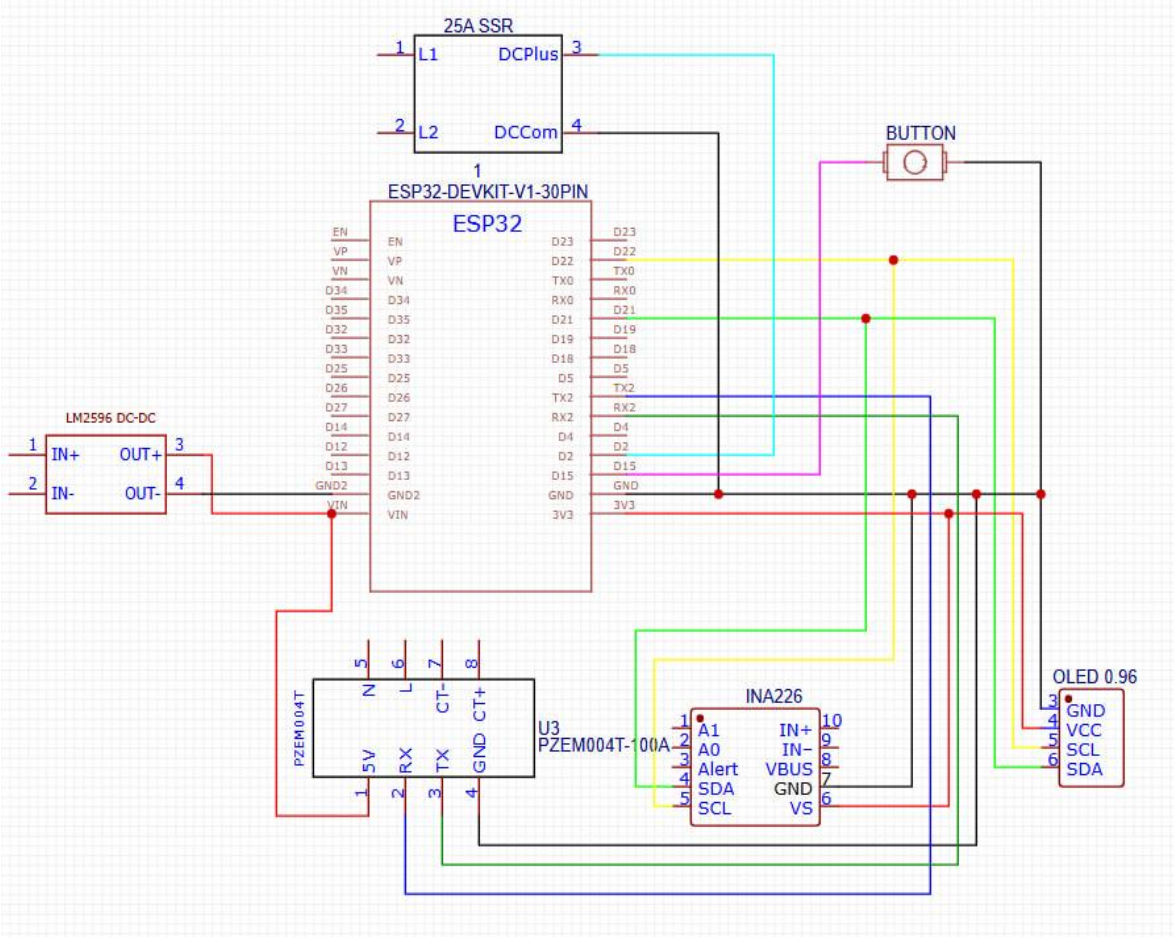

Рисунок 3.1 – Схема пристрою у середовищі EasyEDA

Модуль PZEM-004T-100A підключено на окремий роз’єм із лініями 5 В, GND і двома лініями CT для вимірювання струму. Датчик INA226, що вимірює напругу DC, підключено до шини I²C. OLED-дисплей 0.96″ з’єднано паралельно до тих же SDA/SCL і до 3.3 В від ESP32.

Твердотільне реле розміщене на виносному роз’ємі з L1/L2 у силовому ланцюзі навантаження та DCPlus/DCCom на керуючій стороні. Кнопка для перемикання екранних режимів з’єднана між GPIO 15 і GND.

Усі з’єднання позначені кольорами в схемі: червоне – живлення, чорне – маса, синє – UART, жовте – I²C, бірюзове – керування SSR, фіолетове – кнопка. Така структуризація полегшує трасування і дозволяє чітко розділити функціональні групи на друкованій платі.

3.1.2 Монтаж апаратної частини рішення

Монтаж апаратної частини виконано на універсальній перфоплаті, закріпленій усередині герметичного корпусу. На верхній стороні плати (рисунок 3.2) ліворуч встановлено DC–DC конвертер LM2596, який перетворює зовнішні 5-36 В на стабільні 5 В для живлення ESP32 та PZEM-004T-100A. Поруч розташована плата ESP32. Далі праворуч змонтовано модуль PZEM-004T. У правій частині плати закріплено твердотільне реле.

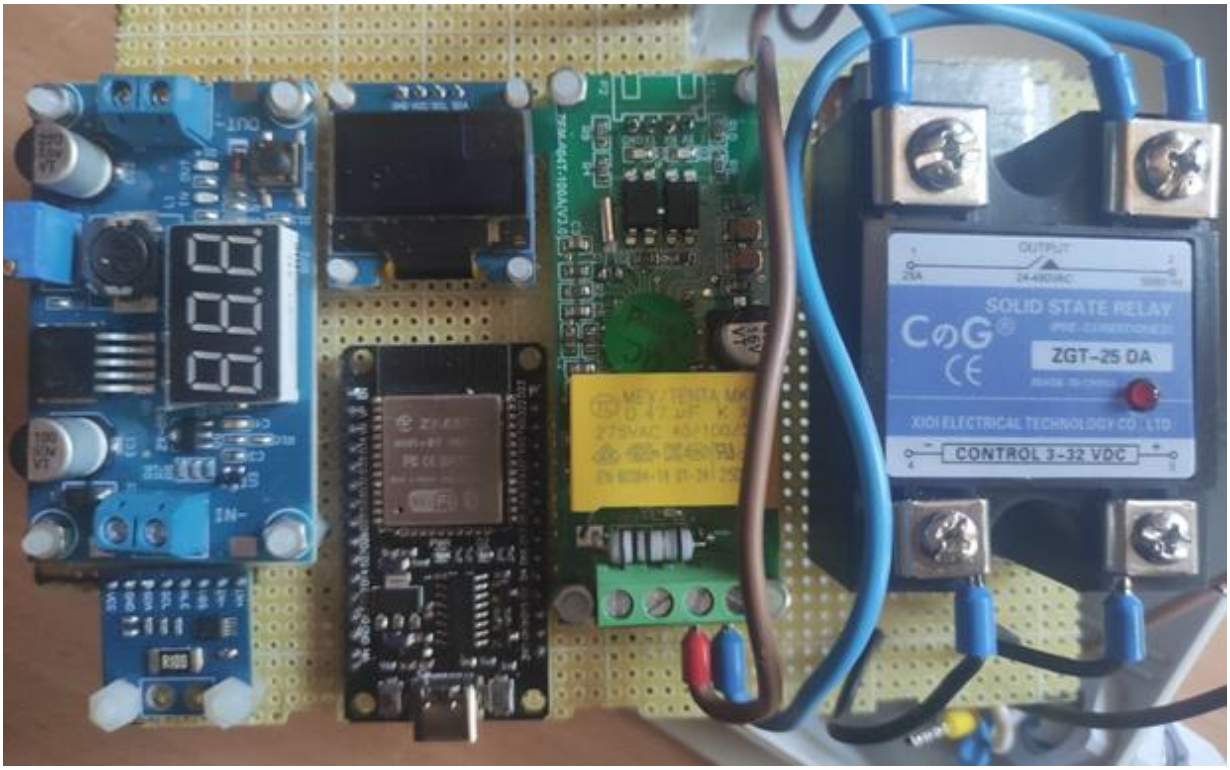

Рисунок 3.2 – Монтаж апаратної частини з передньої сторони

На зворотному боці (рисунок 3.3) видно акуратно прокладені групи проводів. Кожна функціональна зона ізольована, що мінімізує перехресні наводки й спрощує подальше налагодження.

- зелена зона – DC-DC конвертер LM2596;
- червона зона – мікроконтролер ESP32;
- жовта зона – вимірювач споживаної енергії PZEM-004T-100A;
- синя зона – датчик напруги INA226;
- помаранчева зона – OLED-дисплей;
- блакитна зона – твердотільне реле.

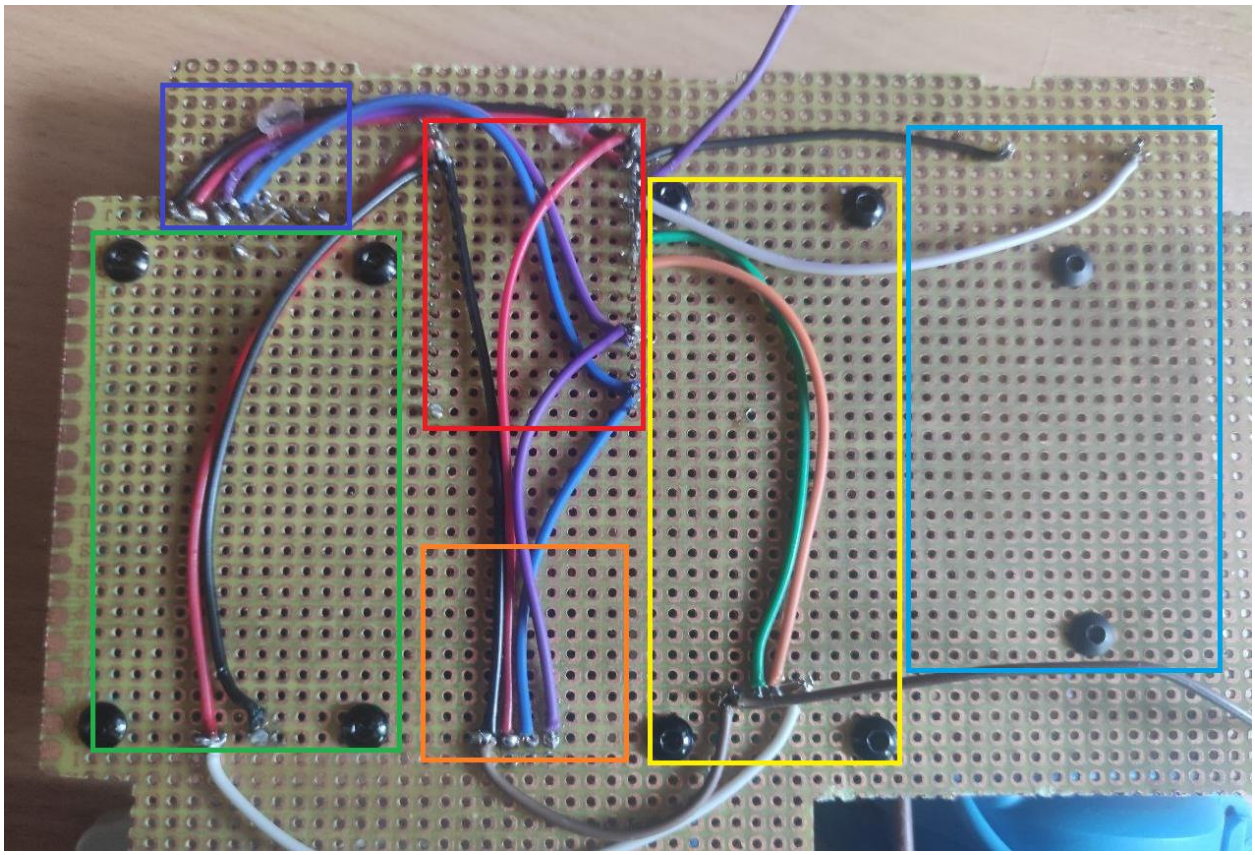

Рисунок 3.3 – Монтаж апаратної частини зі зворотньої сторони

Після завершення монтажу плата встановлена всередину водонепроникного корпусу (рисунок 3.4). Кабельні вводи в нижній частині забезпечують герметичне підведення як мережевого живлення, так і силових ліній до навантаження через SSR. Таке компонування гарантує надійну фіксацію всіх модулів і ефективний теплообмін для силового реле.

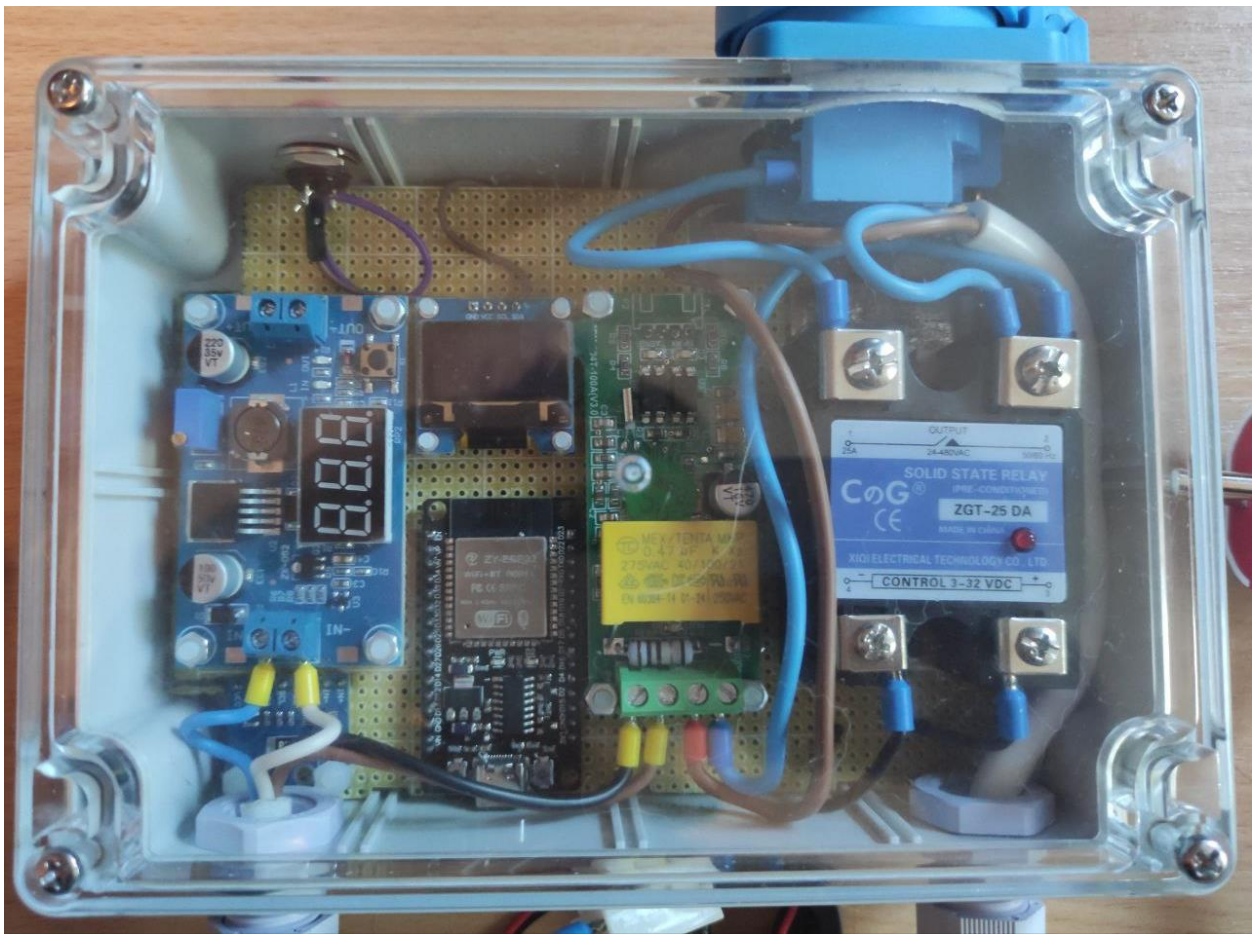

Рисунок 3.4 – Готовий пристрій у корпусі

3.2 Програмна частина

Першим кроком було налаштовано Arduino-IDE із встановленням бібліотек для ESP32, PZEM004Tv30, INA226, Adafruit_SSD1306, ESPAsyncWebServer. Далі наведена модульна архітектура застосунку: окремі компоненти Sensors, Device, Button, Screens, WiFiUtils, TimeUtils і WebServer реалізують зчитування даних, керування SSR, обробку натискань, локальний OLED-інтерфейс, підключення до Wi-Fi/ NTP, перевірку часових вікон та асинхронний HTTP-сервер. Логіка керування навантаженням реалізована в updateDeviceState(), яка порівнює миттєві показники напруги й потужності з заданими порогами та враховує часовий інтервал роботи. Веб-сервер відповідає на GET/POST запити шляхів /data, /borders, /window, /scan і /toggle, повертаючи JSON-дані або HTML-сторінки з динамічно підставленими

значеннями. Клієнтський JavaScript на боці браузера періодично оновлює показники та передає зміни конфігурації, забезпечуючи зручний віддалений контроль системи.

3.2.1 Налаштування середовища розробки

Для розробки ПЗ використовувалась Arduino IDE з доданим платіжним пакетом ESP32. Після встановлення IDE необхідно було через Boards Manager додати підтримку плати ESP32 та в налаштуваннях додати URL-адресу для мікроконтролеру.

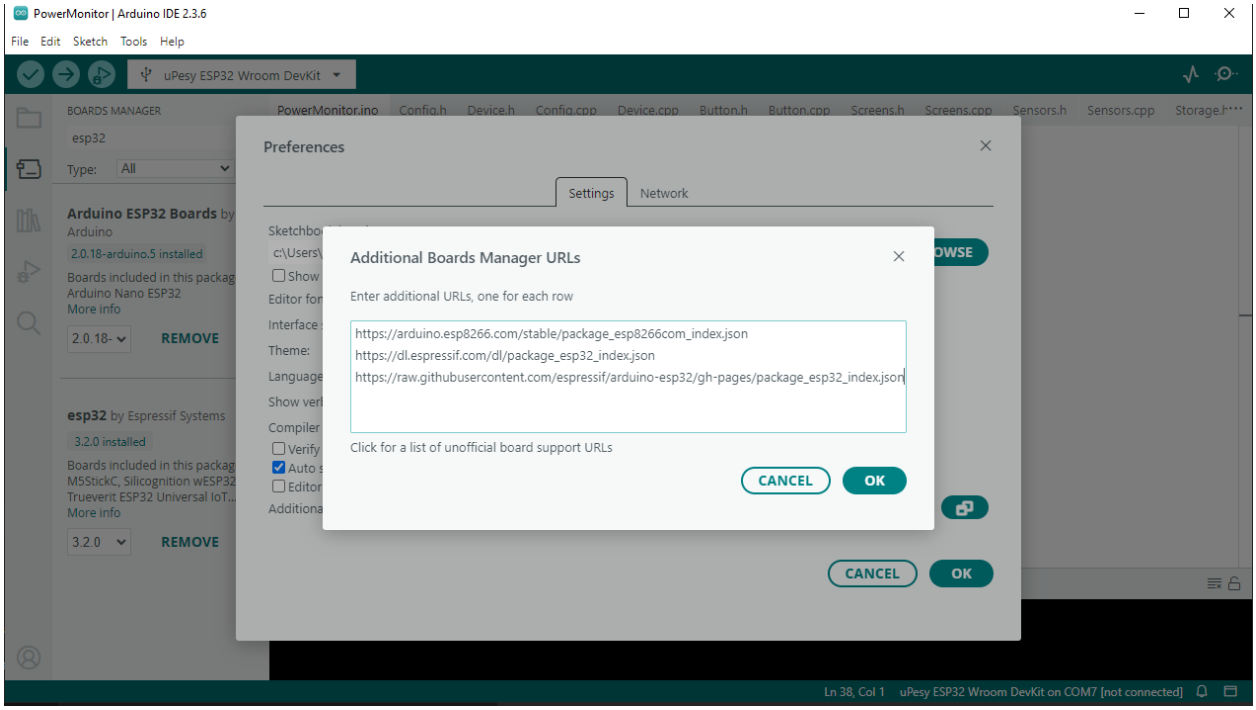

Рисунок 3.5 – Налаштування Arduino IDE

Далі через Library Manager інсталювали всі необхідні бібліотеки:

- PZEM004Tv30 для зв’язку з PZEM-004T-100A;
- INA226_WE для роботи з I²C-датчиком INA22;
- Adafruit_SSD1306 та Adafruit_GFX для OLED-дисплея;
- ESPAsyncWebServer для асинхронного HTTP-сервера;
- WiFi для роботи з мережою;

- Preferences для збереження конфігурації.

Таким чином, середовище було оптимізоване для швидкого компілювання, надійного з’єднання з ESP32 та зручного відлагодження всіх модулів системи.

3.2.2 Структура програмної частини застосунку

Застосунок побудований за модульним принципом, де кожен функціональний компонент винесено у власний набір файлів .h/.cpp (рисунок 3.6).

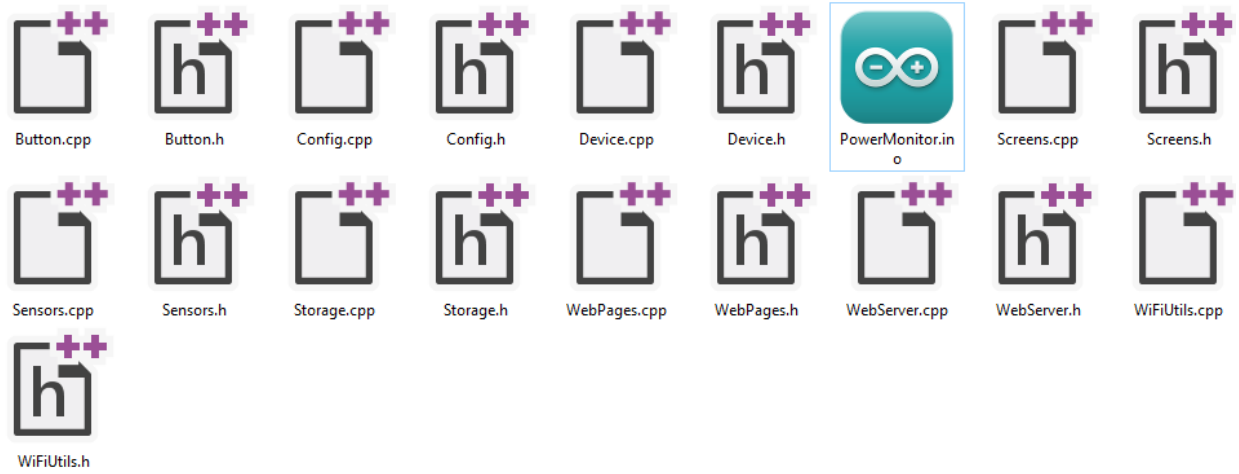

Рисунок 3.6 – Компоненти проекту

У кореневому файлі PowerMonitor.ino зосереджено лише ініціалізацію та основні функції: підключення шини I²C, запуск сенсорів через модуль Sensors, налаштування Wi-Fi і NTP в WiFiUtils та TimeUtils, ініціалізація OLED-дисплея, кнопки й реле через класи Button і Device, а також запуск асинхронного HTTP-сервера з модуля WebServer. Далі головний цикл loop() періодично викликає функцію зчитування з датчиків, оновлює стан реле за алгоритмом з модуля Device, перемикає екрани за натисканням кнопки та передає керівництво рендерингом у Screens (лістинг 3.1).

Лістинг 3.1 – Головний цикл

```
void loop() {
  readSensors();
  if(button.wasPressed()) { screenManager.next();
lastActivity=millis(); }
  if(millis()-lastActivity > IDLE_TIMEOUT) {
screenManager.reset(); lastActivity=millis(); }
  updateDeviceState();
  screenManager.show();
  delay(50);
}
```

Модуль Sensors об’єднує в собі підтримку PZEM-004T-100A і INA226, надаючи єдиний інтерфейс readSensors(), який одним викликом повертає всю потрібну інформацію (лістинг 3.2).

Лістинг 3.2 – Функція зчитування показників

```
void readSensors() {
  voltageSystem   = pzem.voltage();
  currentSystem   = pzem.current();
  powerSystem     = pzem.power();
  frequencySystem = pzem.frequency();
  voltageDC       = ina226.getBusVoltage_V() * 0.947;
}
```

Логіка керування навантаженням зосереджена у функції updateDeviceState() (лістинг 3.3), яка користується глобальними змінними поточної напруги і потужності та перевіряє їх проти збережених у Preferences порогів і часових інтервалів з TimeUtils.

Лістинг 3.3 – Функція для керування живлення пристрою

```
void updateDeviceState() {
  time_t now=time(nullptr);
  bool inOverride = overrideActive && now<overrideUntil;
  bool timeOK = inTimeWindow() || inOverride;
  if(device.isOn()) {
    if(!timeOK || powerSystem>maxPowerSystem ||
voltageDC<minVoltageDC){device.turnOff(); overrideActive=false;}
  } else {
    if(timeOK && voltageDC>maxVoltageDC &&
powerSystem<minPowerSystem)
      device.turnOn();}}
```

Веб-сервер з модуля WebServer працює реєструючи маршрути для отримання JSON-даних /data, збереження конфігурації /borders і /window, а також для відображення HTML-сторінок / та /wifi, шаблони яких збережені у флеш-пам’яті завдяки PROGMEM. Клієнтська JavaScript-логіка, вбудована в HTML, періодично опитує сервер через fetch і динамічно оновлює інтерфейс, не потребуючи перезавантаження сторінки. Така архітектура гарантує чітке розділення обов’язків між модулями, сприяє прозорості коду й полегшує додавання нових функцій.

3.2.3 Логіка керування навантаженням

Керування навантаженням у системі ESP32 Power Monitor здійснюється на основі порогових значень напруги та потужності, а також налаштованого часового інтервалу роботи. Після кожного циклу опитування датчиків система обчислює поточне значення напруги та потужність змінного струму, порівнює їх із мінімальними та максимальними межами і враховує поточний часовий проміжок, отриманий із синхронізованого NTP-годинника. Якщо всі умови задовольняються, реле вмикається. Якщо хоча б одна з умов виходить за встановлені межі або поточний час знаходиться поза налаштованим вікном, реле негайно вимикається.

У разі коли користувач вручну вмикає реле поза межами дозволеного інтервалу, система переводить його в режим тимчасового “override”, у якому реле залишається увімкненим до закінчення поточного періоду роботи. Після настання кінцевої позначки часового вікна або вимкнення реле через вихід за порогові значення режим “override” автоматично скидається, і подальша робота реле підпорядковується стандартним пороговим умовам.

Таким чином, поєднання перевірки електричних параметрів, часових обмежень і механізму “override” забезпечує ефективне та надійне управління навантаженням без необхідності постійного втручання користувача.

3.2.4 Веб-сервер

У системі ESP32 Power Monitor веб-сервер реалізовано на основі бібліотеки ESPAsyncWebServer, що дозволяє обробляти HTTP-запити асинхронно, без блокування основного циклу loop(). Під час ініціалізації відбувається налаштування маршрутизаторів для основних URL-шляхів.

Маршрут “/” відповідає за віддачу головної HTML-сторінки, у якій міститься шаблон веб-інтерфейсу моніторингу разом із кнопкою для керування реле (лістинг 3.4). Ця сторінка завантажується при переході користувача за адресою пристрою.

Лістинг 3.4 – GET-запит на доставку головної HTML-сторінки

```
server.on("/", HTTP_GET, [](auto *req){
    req->send_P(200,"text/html",index_html,processor);
  });
```

При зверненні до маршруту “/data” сервер повертає поточні показники в форматі JSON (лістинг 3.5). У відповіді містяться значення напруги, струму, потужності та інших змінних, необхідні для динамічного оновлення даних на клієнтській стороні.

Лістинг 3.5 – GET-запит на отримання поточних значень показників

```
server.on("/data", HTTP_GET, [](auto *req){
    String j =
String("{\"statusDevice\":")+(device.isOn()?"true":"false")+
      ",\"voltageSystem\":\""+String(voltageSystem,1)+
      "\",\"currentSystem\":\""+String(currentSystem,1)+
      "\",\"powerSystem\":\""+String(powerSystem,1)+
      "\",\"frequencySystem\":\""+String(frequencySystem,1)+
      "\",\"voltageDC\":\""+String(voltageDC,1)+"\"}";
    req->send(200,"application/json",j);
  });
```

Маршрут “/borders” реалізований двома методами. GET-запит повертає існуючі порогові значення напруги та потужності, тоді як POST-запит

приймає нові межі від користувача і зберігає їх у постійній пам’яті (лістинг 3.6). Це дозволяє гнучко змінювати умови вмикання і вимикання навантаження без перезавантаження пристрою.

Лістинг 3.6 – GET та POST запити на отримання та встановлення порогових значень

```
server.on("/borders", HTTP_GET, [](auto *r){
    r->send(200,"application/json",
      String("{\"minVoltageDC\":\"")+String(minVoltageDC,1)+
      "\",\"maxVoltageDC\":\""+String(maxVoltageDC,1)+
      "\",\"minPowerSystem\":\""+String(minPowerSystem,1)+

"\",\"maxPowerSystem\":\""+String(maxPowerSystem,1)+"\"}");
  });

  server.on("/borders", HTTP_POST, [](auto *r){
    auto setF=[&](const char* n,float &v){
      if(r->hasParam(n,true)) v = r->getParam(n,true)-
>value().toFloat();
    };
    setF("minV",minVoltageDC); setF("maxV",maxVoltageDC);
    setF("minP",minPowerSystem); setF("maxP",maxPowerSystem);
    saveThresholds();
    r->send(200,"application/json",
      String("{\"minVoltageDC\":\"")+String(minVoltageDC,1)+
      "\",\"maxVoltageDC\":\""+String(maxVoltageDC,1)+
      "\",\"minPowerSystem\":\""+String(minPowerSystem,1)+

"\",\"maxPowerSystem\":\""+String(maxPowerSystem,1)+"\"}");
  });
```

Аналогічно для часових інтервалів роботи передбачено маршрут “/window”. GET-запит повертає налаштовані початок і кінець дозволеного вікна роботи, а POST-запит дозволяє встановити нові часові межі, які система враховує при автоматичному керуванні реле (лістинг 3.7).

Лістинг 3.7 – GET та POST запити на отримання та встановлення часових інтервалів

```
server.on("/window", HTTP_GET, [](AsyncWebServerRequest *req){
    char buf[6];
    sprintf(buf,"%02u:%02u",window.fromH,window.fromM);
    String json = "{\"from\":\""; json += buf; json +=
```

```
"\",\"to\":\"";
    sprintf(buf,"%02u:%02u",window.toH,window.toM);
    json += buf; json += "\"}";
    req->send(200,"application/json",json);
  });

  server.on("/window", HTTP_POST, [](auto *r){
    auto getHM=[&](const char* p,uint8_t &h,uint8_t &m){
      if(r->hasParam(p,true)){ String v=r->getParam(p,true)-
>value();
        h=v.substring(0,2).toInt(); m=v.substring(3,5).toInt();
}
    };
    getHM("from",window.fromH,window.fromM);
    getHM("to",  window.toH,  window.toM);
    saveWindow(); r->send(200,"text/plain","OK");
});
```

Ручне вмикання або вимикання реле здійснюється через маршрут “/toggle” (лістинг 3.8). Клієнт відправляє POST-запит, і сервер негайно змінює стан реле, повертаючи в тілі відповіді поточний стан.

Лістинг 3.8 – POST-запит на зміну стану реле

```
server.on("/toggle", HTTP_POST, [](auto *req){
    if(device.isOn()){ device.turnOff(); overrideActive=false; }
    else {
      device.turnOn();
      if(!inTimeWindow()){ overrideActive=true;
overrideUntil=nextWindowEnd(); }}
    req->send(200,"application/json",
String("{\"statusDevice\":")+(device.isOn()?"true":"false")+"}")
;});
```

Сторінка налаштування Wi-Fi доступна за маршрутом “/wifi” (лістинг 3.9). При переході на цю адресу користувач бачить форму, де можна вибрати мережу та ввести пароль для підключення ESP32 до локальної точки доступу або маршрутизатора.

Лістинг 3.9 – GET-запит на HTML-сторінки з налаштування Wi-Fi

```
server.on("/wifi", HTTP_GET, [](AsyncWebServerRequest *req){
    req->send_P(200, "text/html", wifi_html);
});
```

Для асинхронного сканування доступних точок Wi-Fi використовується маршрут “/scan”, який повертає список SSID у вигляді JSON. Після вибору мережі та введення пароля клієнт надсилає POST-запит на “/save”, і сервер зберігає вказані налаштування у флеш-пам’яті та перезавантажується для підключення до мережі (лістинг 3.10).

Лістинг 3.10 – GET та POST запити на сканування та отримання доступних точок Wi-Fi

```
server.on("/scan", HTTP_GET, [](AsyncWebServerRequest *req){
    int16_t n = WiFi.scanNetworks(false, false);
    String  json = "[";
    for (int i = 0; i < n; ++i) {
      if (i) json += ',';
      json += '"' + WiFi.SSID(i) + '"';
    }
    json += "]";
    WiFi.scanDelete();
    req->send(200, "application/json", json);
  });

  server.on("/save", HTTP_POST, [](auto *r){
    if(r->hasParam("ssid",true)&&r->hasParam("pass",true)){
      Preferences p; p.begin("wifi",false);
      p.putString("ssid",r->getParam("ssid",true)->value());
      p.putString("pass",r->getParam("pass",true)->value());
      p.end(); r->send(200,"text/plain","OK"); delay(300);
ESP.restart();
    } else r->send(400,"text/plain","Bad request");
  });
```

HTML-шаблони (index_html та wifi_html) зберігаються в пам’яті PROGMEM, а динамічна підстановка значень відбувається через функцію processor(), яка замінює уміст плейсхолдерів на актуальні дані змінних. Для взаємодії з клієнтським JavaScript-кодом усі JSON-ендпоїнти повертають рядок у форматі application/json, що забезпечує простий парсинг за допомогою fetch() та оновлення DOM без перезавантаження сторінки.

Ключовою перевагою асинхронного підходу є те, що сервер може обслужити багато одночасних запитів без затримок опитування датчиків і відображення на OLED.

3.3 Тестування пристрою

Тестування системи ESP32 Power Monitor проводилося за основними сценаріями, що імітують реальні умови експлуатації. Перший сценарій передбачав роботу в режимі стаціонарного підключення до домашнього роутера (STA), коли прилад постійно отримує IP-адресу та стабільно обмінюється даними з мережею. У цьому випадку перевірялися точність опитування PZEM-004T і INA226, частота оновлення веб-інтерфейсу та швидкодія ручного перемикання реле через браузер. Другий сценарій передбачав різке відключення та поновлення Wi-Fi-з’єднання для перевірки механізму автоматичного реконекту й коректності відновлення роботи HTTP-серверу без втрати даних.

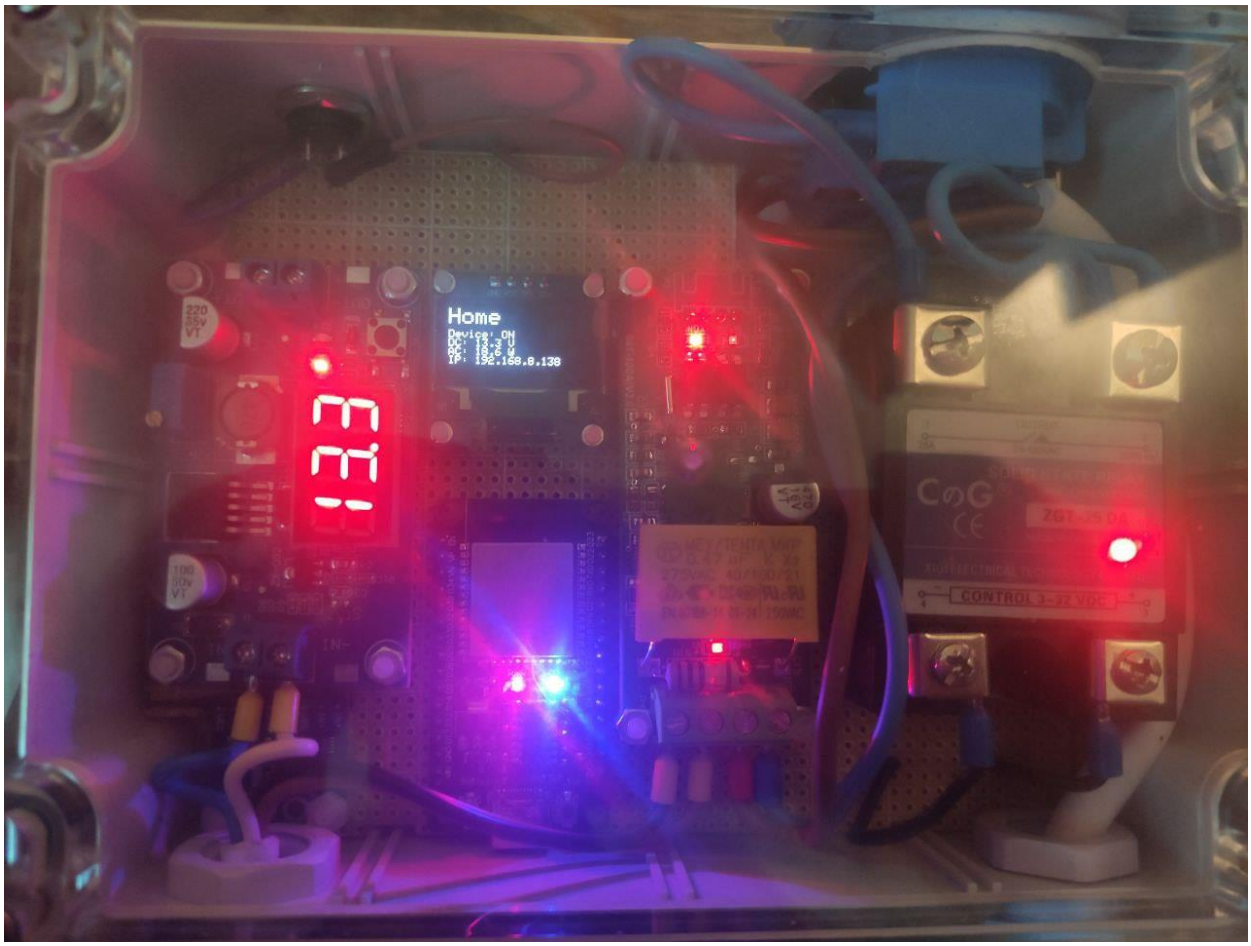

Рисунок 3.7 – Тестування пристрою

Під час тестів проводилася вимірювальна серія з використанням еталонного мультиметра. З’ясовано, що при використанні PZEM-004T усі показники були виміряні точно. Вимірювання напруги INA226 відрізнялися від еталону приблизно на 5%, тому було введено коефіцієнт множення. Після цього усі дані співпадали з мультиметром.

Отримані дані свідчать про високу стабільність та реальну придатність ESP32 Power Monitor для безперервного використання у домашніх та промислових умовах. Похибка вимірювань відповідає заявленим специфікаціям компонентів, а оптимізоване програмне забезпечення забезпечує своєчасний збір та передачу даних, без зайвих затримок і втрат у роботі системи.

4 ІНСТРУКЦІЯ КОРИСТУВАЧА

4.1 Огляд IoT-рішення

Розроблений ESP32 Power Monitor є повноцінним IoT-рішенням для моніторингу та управління електричною потужністю в реальному часі. Реалізація IoT-пристрою ESP32 Power Monitor поєднує в одному компактному корпусі вимірювальні модулі, елементи керування навантаженням та мережевий інтерфейс для віддаленого моніторингу. Фізичні розміри корпусу становлять 17 см у довжину, 14 см у ширину та 5,5 см у висоту.

Живлення пристрою здійснюється від джерела постійного струму (до 36 В) через регульований модуль LM2596, що стабілізує напругу до 5 В для живлення ESP32, OLED-дисплея та інших периферійних елементів. Для вимірювання сили струму на фазний провід після інвертору встановлюється струмовий трансформатор PZEM-004T-100A, а сам модуль під’єднується до стандартної мережевої розетки, що дозволяє зчитувати величину напруги змінного струму, обчислювати потужність та накопичену енергію. Моніторинг напруги постійного струму виконує сенсор INA226, підключений по шині I^2C, і результати виводяться на OLED-екран із кнопкою для локальної навігації між режимами відображення (рисунок 4.1).

Вбудований мікроконтролер ESP32 забезпечує одночасну роботу в режимах станції та точки доступу Wi-Fi, що дозволяє або підключатися до існуючої мережі Інтернет, або запускати власну точку доступу для первинного налаштування. Асинхронний HTTP-сервер передає в реальному часі дані у форматі JSON, а клієнтська частина веб-інтерфейсу дає змогу віддалено стежити за напругою, струмом, потужністю й частотою, встановлювати порогові значення та часові інтервали роботи реле, а також керувати навантаженням вручну.

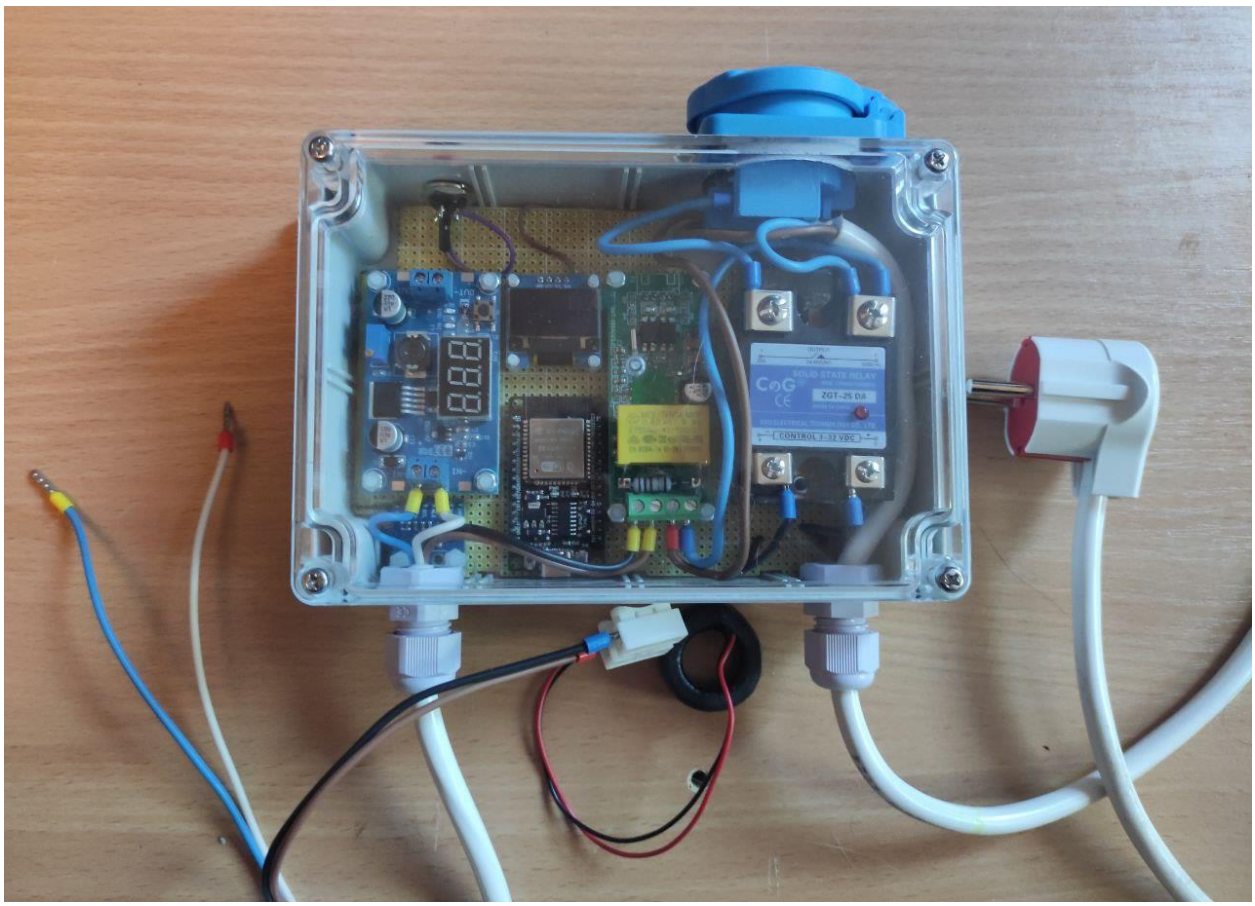

Рисунок 4.1 – Огляд фінальної версії IoT-рішення

4.2 Підключення та робота пристрою

Для підключення пристрою потрібно подати живлення від джерела постійного струму до 36 В на вхідний модуль на базі LM2596, який виконує стабілізацію напруги до 5 В для живлення ESP32, OLED-дисплея та логіки. Для цього потрібно блакитний провід під’єднати до “+” акумулятору, а білий провід до “-”.

Для вимірювання струму котушку потрібно встановити на фазний провід мережі змінного струму після інвертору, тобто пропустити тільки один провід крізь отвір датчика. Для вимірювання напруги прилад слід підключити до мережі, вставивши вилку в розетку (рисунок 4.2).

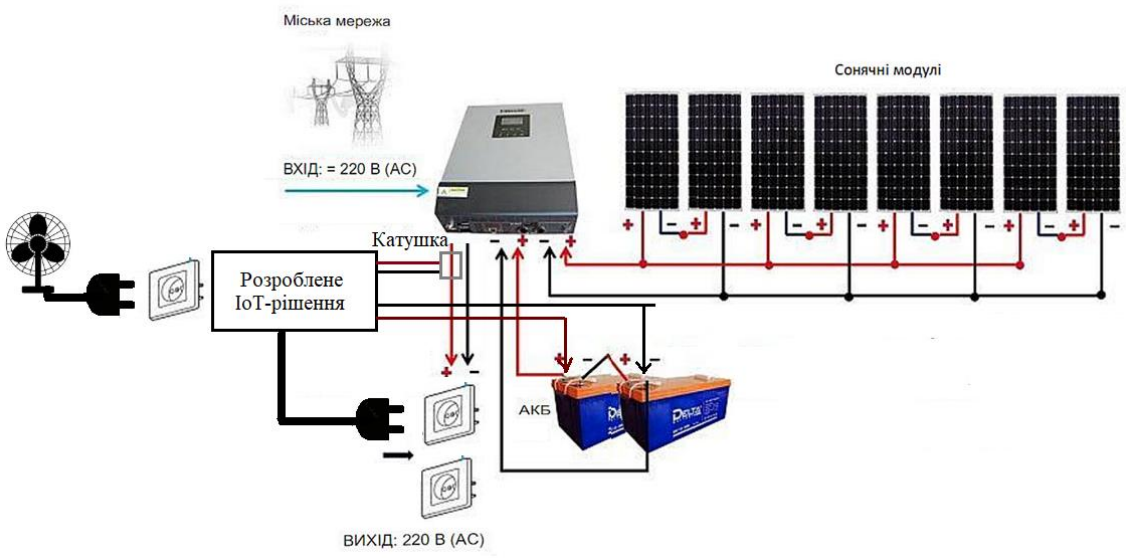

Рисунок 4.2 – Схема підключення

Після першого включення ESP32 автоматично переходить у режим точки доступу з ім’ям мережі Esp32_PowerMonitor і паролем 12345678 (рисунок 4.3).

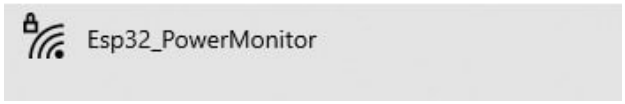

Рисунок 4.3 – Мережа Esp32_PowerMonitor

Щоб налаштувати Wi-Fi, спочатку підключіться до точки доступу пристрою та відкрийте в браузері адресу http://192.168.4.1. На головній сторінці вам відобразяться поточні параметри системи та встановлені межі роботи обладнання (рисунок 4.4). У верхньому куті є кнопка “Wi-Fi setup”. При натисканні на яку користувач переходить до іншої сторінки, де можна підключитися до мережі.

Після переходу пристрій автоматично просканує всі доступні мережі й покаже їх у випадаючому списку. Виберіть свою домашню мережу, введіть пароль і натисніть “Save & Reboot” (рисунок 4.5). ESP32 перезавантажиться у режимі станції та спробує приєднатися до обраної мережі. Якщо з’єднання не відбудеться, пристрій повернеться в режим точки доступу.

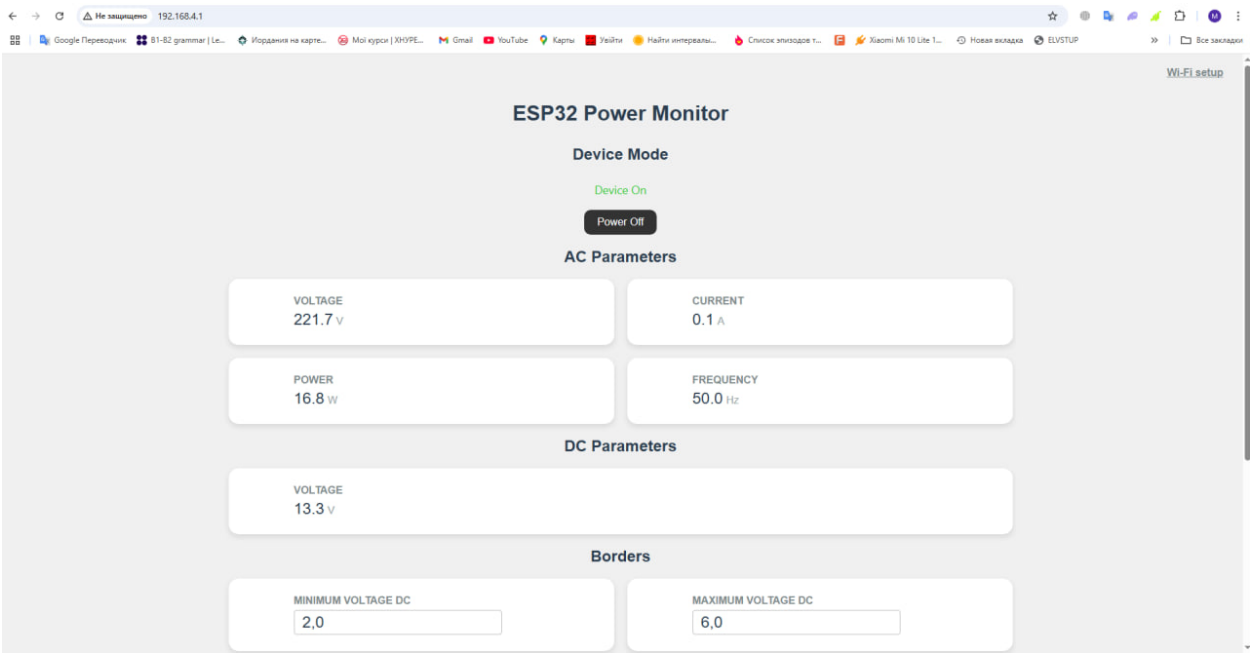

Рисунок 4.4 – Головна сторінка

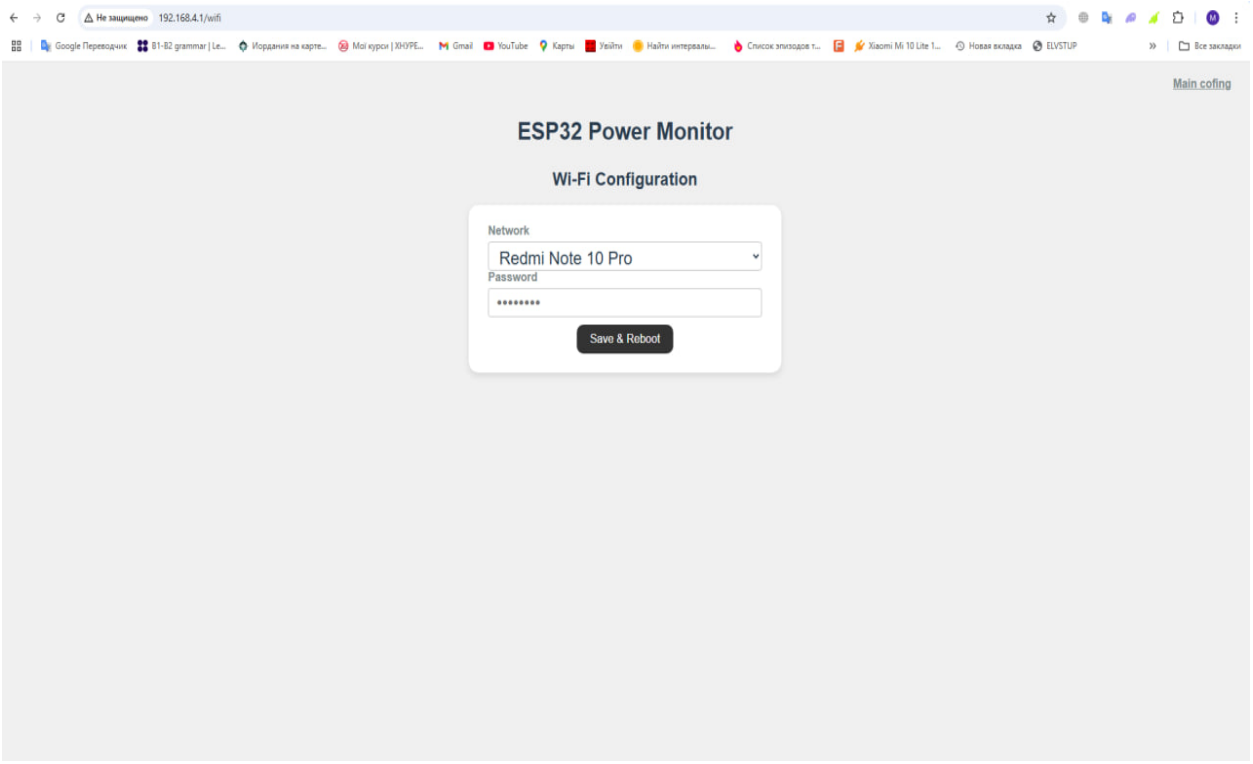

Рисунок 4.5 – Сторінка з налаштування Wi-Fi

Веб-інтерфейс доступний за адресою http://<IP-адреса> і складається з декількох розділів. Головна сторінка відображає стан реле, кнопку керування для ручного вмикання або вимикання навантаження. Секція параметрів

змінного струму показує поточні значення напруги, струму, потужності та частоти у вигляді карток. Розділ параметрів постійного струму виводить вимірювану DC-напругу. У розділі меж можна вводити нові порогові значення DC-напруги та AC-потужності і натискати кнопку “Save Config”, після чого вони зберігаються у внутрішній пам’яті ESP32. Розділ часової роботи дозволяє задати інтервал увімкнення пристрою в годинах і хвилинах та зберегти його для подальшого врахування в алгоритмі керування (рисунок 4.6).

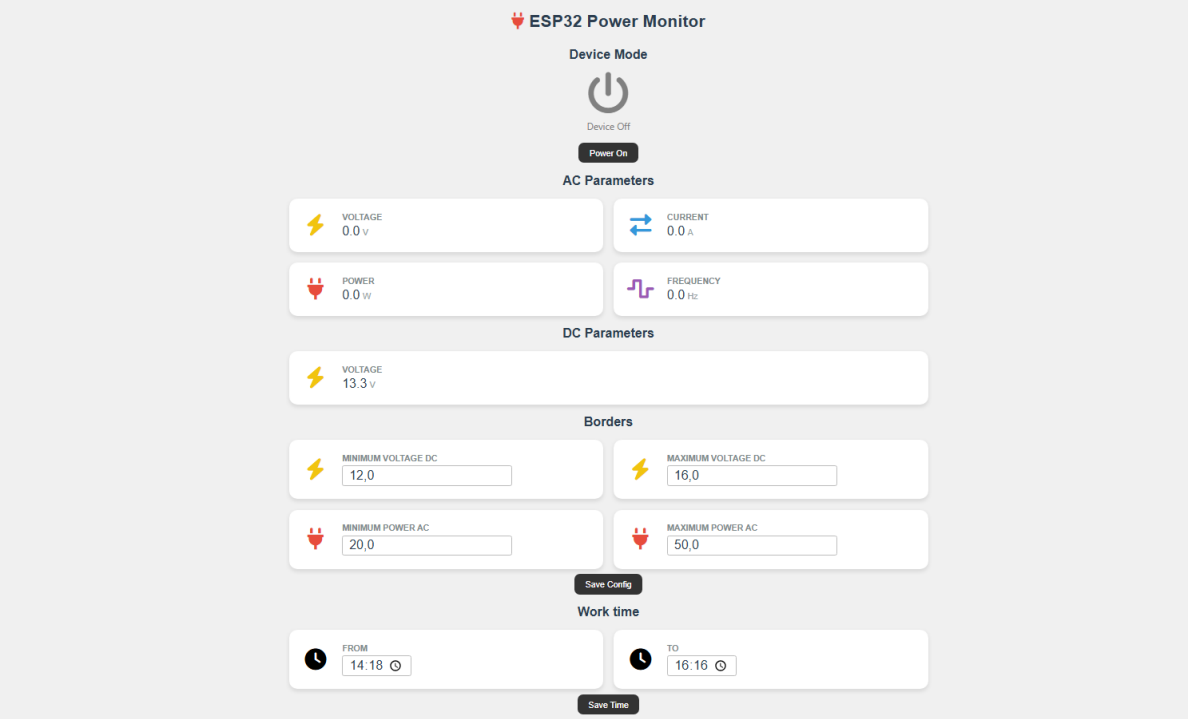

Рисунок 4.6 – Головна сторінка з обмеженнями

На OLED-дисплеї реалізовано чотири екрани, які перемикаються натисканням однієї кнопки. Перший екран показує поточний стан реле, значення DC-напруги і AC-потужності разом із IP-адресою пристрою. Другий екран відображає детальну інформацію по змінному струму: напругу, струм, потужність і частоту мережі. Третій екран призначено для перегляду встановлених меж: мінімальних та максимальних значень DC-напруги і AC-потужності, а також часових меж роботи пристрою. Четвертий екран

демонструє ім’я підключеної Wi-Fi мережі та поточну IP-адресу. Після 60 секунд бездіяльності інтерфейс автоматично повертається до першого екрана.

Логіка автоматичного керування навантаженням передбачає, що реле вмикається, коли напруга на DC-шіні перевищує встановлений максимум і потужність змінного струму не перевищує мінімум в рамках заданого часового інтервалу. Якщо напруга на акумуляторі падає нижче мінімального значення, або потужність системи перевищує максимальний поріг, або поза інтервалом роботи, реле автоматично вимикається. У разі ручного ввімкнення поза вікном система переходить у режим override і залишатиметься в цьому стані до завершення інтервалу, однак тільки при дотриманні інших обмежень.

4.3 Технічні обмеження системи та рекомендації

Система розрахована на роботу з навантаженням, яке не перевищує номінальний струм твердотільного реле. Перевищення цього значення може призвести до перегріву напівпровідникових елементів і виходу їх з ладу, тому не рекомендується підключати прилади з високим пусковим струмом без попередніх розрахунків та додаткового тепловідведення.

Рекомендований вхідний діапазон модуля живлення LM2596 становить від 5 В до 36 В. При підключенні джерела з напругою ближче до верхньої межі слід перевірити тепловиділення модуля та, за потреби, встановити радіатор.

Для вимірювання напруги та струму використовується модуль INA226. Діапазон вимірювання напруги шини обмежений 0…36 В, а максимальна напруга на вході датчика (VIN+) не повинна перевищувати 36 В.

Модуль PZEM-004T-100A допускає вимірювання змінної напруги до 260 В і струму до 100 А через зовнішній трансформатор струму.

Електронні компоненти, зокрема плати ESP32, модулі вимірювання та

дисплей, повинні працювати в сухому середовищі. Волога або конденсат можуть викликати короткі замикання та корозію контактів, тому корпус і місце монтажу слід захищати від попадання вологи та пилу.

Для коректного відображення веб-інтерфейсу необхідно переконатися у правильності IP-адреси пристрою та стабільності Wi-Fi-з’єднання. Якщо сторінка не відкривається, слід перевірити, чи ESP32 успішно підключена до мережі, та при необхідності перейти у режим точки доступу для повторного введення SSID і пароля. Вбудований механізм автоматичного повторного підключення полегшує цю процедуру, але в разі системних збоїв рекомендується виконати повне скидання мережевих налаштувань.

Застосування наведених рекомендацій дозволить уникнути передчасного зносу обладнання, забезпечить стабільність роботи в різних умовах експлуатації та гарантує довгий термін служби.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було створено та апробовано програмно-апаратне IoT-рішення для розумного керування навантаженнями від акумуляторної батареї сонячної електростанції. Практична реалізація на базі мікроконтролера ESP32 із застосуванням модулів PZEM-004T, INA226, перетворювача LM2596 та твердотільного реле дозволила забезпечити точне вимірювання напруги, струму й потужності змінного та постійного струму, а також гнучку логіку автоматичного включення і виключення другорядних приладів.

Розроблена структурна архітектура програмного забезпечення поділена на окремі модулі, що відповідають за зчитування даних із сенсорів, обробку часових та порогових умов, збереження налаштувань і обслуговування асинхронного HTTP-сервера. Це гарантує стабільну й надійну роботу навіть під час одночасної обробки кількох запитів клієнтів та мінімальне блокування основного циклу.

Впроваджений веб-інтерфейс дозволяє користувачеві віддалено переглядати показники в реальному часі, оперативно коригувати пороги напруги й потужності, а також задавати часові вікна роботи приладів без необхідності перезавантаження пристрою. Локальний OLED-дисплей із кнопкою навігації надає базову інформацію та зручний спосіб контролю без доступу до мережі.

Результати тестування підтвердили точність вимірювань INA226 та PZEM-004T-100A. Завдяки автоматичному перемиканню між режимами STA та AP і реалізації асинхронного сканування Wi-Fi скрипти керування продемонстрували швидке відновлення зв’язку та збереження працездатності сервера без жодних зависань. Відмова від зовнішніх хмарних сервісів зробила розв’язок незалежним від Інтернет-каналу та підвищила безпеку локального керування.

Соціально-економічний ефект від впровадження системи полягає у продовженні терміну служби акумуляторів за рахунок запобігання глибоким розрядам, зниженні витрат на електроенергію та зменшенні навантаження на центральні мережі під час пікових періодів споживання. Інструментальна цінність рішення проявиться у можливості ефективно керувати побутовими приладами в умовах енергетичної нестабільності чи обмеженого доступу до централізованих мереж.

Подальші дослідження можуть бути спрямовані на впровадження адаптивних алгоритмів прогнозування вироблення енергії з урахуванням погодних даних, інтеграцію з MQTT-брокером для централізованого моніторингу, а також розширення системи можливістю накопичення історії вимірювань на SD-карті чи сторонніх серверах. Реалізація таких доповнень зробить розроблене IoT-рішення ще більш гнучким, масштабованим та готовим до комерційного використання.

Підсумовуючи, виконана кваліфікаційна робота продемонструвала ефективність розробленого IoT-рішення на базі ESP32, яке поєднує високу точність вимірювань, гнучку модульну архітектуру програмного забезпечення та зручні локальні й віддалені інтерфейси. Запропонована система дозволяє автоматизовано керувати побутовими навантаженнями залежно від рівня заряду акумуляторів, утримувати енергоспоживання в заданих межах і забезпечувати безперебійну роботу як у зв’язку зі стабільною мережею, так і в автономному режимі точки доступу. Результати тестів підтвердили надійність і швидкодію пристрою, водночас економічний і екологічний ефект від його впровадження робить його перспективним інструментом для підвищення стійкості та незалежності енергозабезпечення в умовах сучасних викликів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Батурін О. О., Ні Я. С. ПРОГРАМНО-АПАРАТНЕ IoT-РІШЕННЯ ДЛЯ РОЗУМНОГО КЕРУВАННЯ ЖИВЛЕННЯМ ПРИСТРОЇВ: тези доп. 15-ї міжнар. наук.-техн. конф. (24–25 квіт. 2025 р., Баку – Харків – Жиліна). – Т. 1. – Харків; Баку; Жиліна, 2025. – С. 140.

2. Adafruit Learning System. Using the INA219/INA226 for High-Side Current Sensing. URL: https://learn.adafruit.com/adafruit-ina219-current-sensor-breakout/overview (дата звернення 12.05.2025).

3. Arduino Docs. Arduino Language Reference. URL: https://docs.arduino.cc/learn (дата звернення 17.05.2025).

4. Arduino Docs. Getting Started with Arduino IDE. URL: https://docs.arduino.cc/software/ide-v2 (дата звернення 14.05.2025).

5. Battery University. BU-808: How to Prolong Lithium-based Batteries. URL: https://batteryuniversity.com/article/bu-808-how-to-prolong-lithium-based-batteries (дата звернення 13.05.2025).

6. BloombergNEF. Global Energy Storage Market Records Biggest Jump Yet. URL: https://about.bnef.com/blog/global-energy-storage-market-records-biggest-jump-yet/?utm_source=linkedin&utm_medium=paid_social&utm_campaign=global_bnef_bssscorp_2024_ao&utm_content=blog_bnef-energ-stor&tactic=833883 (дата звернення 12.05.2025).

7. Electronics Notes. Solid State Relays (SSR) – What They Are and How They Work. URL: https://www.electronics-notes.com/articles/electronic_components/electrical-electronic-relay/solid-state-relay-switch.php (дата звернення 18.05.2025).

8. ESPAsyncWebServer. GitHub Repository. URL: https://github.com/me-no-dev/ESPAsyncWebServer (дата звернення 14.05.2025).

9. SSR-25DA Solid State Relay Datasheet. URL:

https://www.alldatasheet.com/view.jsp?Searchword=Ssr25da%20datasheet&gad_source=1&gad_campaignid=1437346752&gbraid=0AAAADcdDU91Lw8menyHMRBdjTITwqxfk&gclid=Cj0KCQjwucDBBhDxARIsANqFdr39wKBC8bqpZgPgKgYdFEEIWxI5oa1beGx9OmSY-OWgW0C7AnHFLLkaAoc1EALw_wcB (дата звернення 14.05.2025).

10. International Energy Agency. Growth in Global Electricity Demand Is Set to Accelerate. URL: https://www.iea.org/reports/global-energy-review-2025/electricity (дата звернення 13.05.2025).

11. International Energy Agency. Renewables 2024: Global Overview. URL: https://iea.org/reports/renewables-2024/global-overview (дата звернення 14.05.2025).

12. International Energy Agency. Ukraine’s Energy System Under Attack. URL: https://www.iea.org/reports/ukraines-energy-security-and-the-coming-winter/ukraines-energy-system-under-attack (дата звернення 12.05.2025).

13. International Energy Agency. World Energy Outlook 2024. URL: https://iea.org/reports/world-energy-outlook-2024 (дата звернення 15.05.2025).

14. McKinsey & Company. Global Energy Perspective 2024. URL: https://www.mckinsey.com/industries/energy-and-materials/our-insights/global-energy-perspective (дата звернення 16.05.2025).

15. National Renewable Energy Laboratory. 2024 ATB: Residential Battery Storage. URL: https://atb.nrel.gov/electricity/2024/residential_battery_storage (дата звернення 17.05.2025).

16. OpenEnergyMonitor. URL: https://openenergymonitor.org/ (дата звернення 13.05.2025).

17. PVKnowHow. Residential Solar Surpasses 1.5 GW in Ukraine. URL: https://www.pvknowhow.com/news/residential-solar-surpasses-1-5-gw-in-ukraine (дата звернення 14.05.2025).

18. PZEM004T-100A Serial AC Power Energy Monitor Datasheet. SainSmart. URL: https://www.sainsmart.com/products/pzem-004t-100a-serial-ac-power-energy-monitor (дата звернення 16.05.2025).

19. Random Nerd Tutorials. ESP32 Web Server – AsyncWebServer Library. URL: https://randomnerdtutorials.com/esp32-async-web-server-espasyncwebserver-library (дата звернення 17.05.2025).

20. Stroustrup B. The C++ Programming Language (4th Edition). Addison-Wesley, 2013.

21. Technavio. IoT in Energy Grid Management Market Size 2025–2029. URL: https://www.technavio.com/report/iot-market-in-energy-grid-management-industry-analysis#:~:text=The%20global%20IoT%20In%20Energy,19.4%25%20during%20the%20forecast%20period (дата звернення 19.05.2025).

22. Texas Instruments. INA226 Precision Digital Current and Power Monitor with I2C Interface (Datasheet). URL: https://www.ti.com/lit/ds/symlink/ina226.pdf (дата звернення 20.05.2025).

23. Texas Instruments. LM2596 SIMPLE SWITCHER® Power Converter Datasheet. URL: https://www.ti.com/lit/ds/symlink/lm2596.pdf (дата звернення 21.05.2025).

24. Tycorun Battery Blog. 10 Common Inverter Failures and Solutions. URL: https://tycorun.com/blogs/news/inverter-failure (дата звернення 16.05.2025).

25. Victron Energy. Energy Storage System (ESS) White Paper, 2024.

26. Waveshare. 0.96 inch OLED Display Module Specification. URL: https://www.waveshare.com/wiki/0.96inch_OLED_Module (дата звернення 18.05.2025).