

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Радіотехнічний факультет

Кафедра прикладної радіоелектроніки

До захисту допущено:

Зав. кафедри

 Андрій МОВЧАНЮК

«__» _____ 20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньою програмою «Інтелектуальні технології радіоелектронної техніки», за спеціальністю 172 «Телекомунікації та радіотехніка»

на тему: Система захищеного віддаленого керування домашньою системою електроживлення

Виконав:

Студент IV курсу, групи РЕ-22

Крамаренко Михайло Олександрович

Прізвище, ім'я, по батькові

Керівник: ст. викл. PhD Єзерський Нікіта Валерійович

Посада, науковий ступінь, вчене звання

Прізвище, ім'я, по батькові

Рецензент: ст. викл., к.т.н., каф. РІ Булашенко Андрій

Васильович

Посада, науковий ступінь, вчене звання,

Прізвище, ім'я, по батькові

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент (-ка)  _____

Київ – 2026 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Радіотехнічний факультет
Кафедра прикладної радіоелектроніки

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 172 «Телекомунікації та радіотехніка»

Освітня програма «Інтелектуальні технології радіоелектронної техніки»

ЗАТВЕРДЖУЮ

Зав. кафедри

_____ Андрій МОВЧАНЮК

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студента

Крамаренка Михайла Олександровича

1. Тема проєкту «Система захищеного віддаленого керування домашньою системою електроживлення»

2. Керівник проєкту Єзерський Нікіта Валерійович, затверджені наказом по університету від « 29 » травня 2026 р. №1989-с

Строк подання студентом проєкту 08 червня 2026 року.

3. Вихідні дані до проєкту: Напруга живлення 5 В, струм 3 А, захищене віддалене керування через VPN, надсилання сповіщень в Telegram, інформативний інтерфейс керування з поточною ємністю, поточним споживанням, та можливістю керування інверторами станції.

4. Зміст пояснювальної записки 1) Огляд існуючих рішень, 2) Обґрунтування вибору програмного забезпечення 3) Практична реалізація проєкту.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) UML діаграма послідовності, HTML коди повідомлень, презентація 17 сторінок.

6. Дата видачі завдання 14 квітня 2026 року.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Огляд існуючих рішень	14.04.26-22.04.26	Виконав
2	Обґрунтування вибору програмного забезпечення	23.04.26-25.04.26	Виконав
3	Налаштування мікрокомп'ютера	26.04.26-27.04.26	Виконав
4	Інтегрування станції зі шлюзом	28.04.26-29.04.26	Виконав
5	Налаштування захищеного каналу віддаленого керування	30.04.26-01.05.26	Виконав
6	Розробка Telegram Bot та налаштування сповіщень	02.05.26-09.05.26	Виконав
7	Розробка графічного інтерфейсу	10.05.26-17.05.26	Виконав
8	Експериментальне дослідження енергоспоживання шлюзу	18.05.26-26.05.26	Виконав
9	Оформлення текстової та графічної документації	27.05.26-08.06.26	Виконав

Студент



Крамаренко Михайло Олександрович

Керівник



Єзерський Нікіта Валерійович

АНОТАЦІЯ

Дипломний проєкт складається з пояснювальної записки обсягом 74 сторінки, що мають в собі 42 ілюстрацію, 2 таблиці, 5 додатків, 48 бібліографічних посилань.

Метою проєкту є розробка, практична реалізація та експериментальне дослідження системи збору телеметрії та захищеного віддаленого керування домашньою системою електроживлення.

У проєкті використано загальнодоступні сервіси, такі як Home Assistant, бібліотеки Github, Telegram Bot. Реалізовано захищений віддалений доступ до панелі керування через L2TP VPN. Розроблено зручний для користувача дашборд.

Ключові слова: Bluetti, Home Assistant, Github, VPN, керування, кіберзахист, Mikrotik, Raspberry Pi 4 Model B

ANNOTATION

The diploma project consists of a report of 74 pages, which includes 42 illustrations, 2 tables, 5 appendices, and 48 bibliographic references.

The aim of the project is to develop, implement, and conduct experimental research on a system for collecting telemetry data and providing secure remote control of a home power supply system.

The project utilizes publicly available services such as Home Assistant, GitHub libraries, and Telegram Bot. Secure remote access to the control panel via L2TP VPN has been implemented. A user-friendly dashboard has been developed.

Keywords: Bluetti, Home Assistant, GitHub, VPN, control, cybersecurity, Mikrotik, Raspberry Pi 4 Model B

Пояснювальна записка
до дипломного проєкту
на тему: «Система захищеного віддаленого керування
домашньою системою електроживлення»

Київ — 2026 року

ЗМІСТ

Перелік скорочень.....	4
Вступ.....	5
1 Аналіз Існуючого стану Об’єкту проектування.....	8
1.1 Аналіз існуючих технічних рішень на ринку.....	8
1.1.1 Модернізація апаратної бази шляхом придбання дорожчого обладнання	8
1.1.2 Функціональний аналіз штатного програмного забезпечення Bluetti App	10
1.1.3 Професійні системи моніторингу на прикладі платформи Victron Energy VRM	12
1.1.4 Готові апаратні Bluetooth шлюзи екосистем Xiaomi та TuYa..	13
1.1.5 Підсумки проведеного аналізу та обґрунтування концепції власної розробки	15
2 Обґрунтування напрямків дослідження та проектних рішень	18
2.1 Визначення вимог до апаратної бази	18
2.2 Обґрунтування вибору програмного забезпечення середовища управління.....	20
2.3 Вибір оптимального методу інтеграції периферійного пристрою в екосистему	21
2.4 Організація захищеного віддаленого доступу та вибір мережевого протоколу	23
2.4.1 Концепція застосування менеджерів паролів	24

					РЕ22.461142.001 ПЗ			
ЗМ.	Лис	№ докум.	Підпис	Дата	Система захищеного віддаленого керування ДСЕ	Лім.	Лист	Листів
Розробив	Крамаренко							
Перевірів	Єзерський						1	
Н. Контр.	Єзерський						РЕ-22, РТФ, НТУУ «КПІ ім. Ігоря Сікорського»	
Затвер-	Єзерський							

2.5	Проектування системи оперативного інформування та моніторингу подій.....	26
2.6	Проектування інтерфейсу користувача	28
2.7	Висновки до другого розділу	30
3	Практична реалізація програмно-апаратного шлюзу моніторингу	34
3.1	Розгортання та первинне налаштування операційного середовища Home Assistant на базі мікрокомп'ютера Raspberry PI 4.....	34
3.1.1	Підготовка завантажувального носія.....	34
3.1.2	Запис операційної системи Home Assistant.....	39
3.1.3	Конфігурація бездротового інтерфейсу мережі Wi-Fi.....	43
3.2	Інтеграція джерел безперервного живлення	45
3.3	Розгортання захищеного каналу віддаленого керування на базі технології L2TP VPN	49
3.4	Інтеграція та налаштування модуля сповіщень на базі платформи Telegram.....	54
3.5	Розробка графічного інтерфейсу користувача та побудова панелі керування	55
3.6	Експериментальне дослідження енергоспоживання системи моніторингу	57
	Висновки	63
	Перелік джерел посилання	65
	Додаток А. HTML код тригера на вимкнення живлення.....	70
	Додаток Б. HTML код тригера на увімкнення живлення	71

					РЕ22.461142.001 ПЗ			
					Система захищеного віддаленого керування ДСЕ	Лім.	Лист	Листів
ЗМ.	Лис	№ докум.	Підпис	Дата			2	
Розробив	Крамаренко							
Перевірів	Єзерський							
Н. Контр.	Єзерський							
Затвер-	Єзерський					РЕ-22, РТФ, НТУУ «КПІ ім. Ігоря Сікорського»		

Додаток В. HTML код тригера на поточний заряд.....	72
Додаток Г. UML діаграма послідовності.....	73
Додаток Д. ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ	74

					РЕ22.461142.001 ПЗ				
ЗМ.	Лист	№ докум.	Підпис	Дата	Система захищеного вiддаленого керування ДСЕ		Лiт.	Лист	Листiв
Розробив	Крамаренко								
Перевірів	Єзерський							3	
Н. Контр.	Єзерський						РЕ-22, РТФ, НТУУ «КПІ ім. Ігоря Сікорського»		
Затвер-	Єзерський								

ПЕРЕЛІК СКОРОЧЕНЬ

ДСЕ — Домашня Система Електроживлення

UML — Unified Modeling Language

VPN — Virtual Private Network

Wi-Fi — Wireless Fidelity

GSM — Global System for Mobile Communications

API — Application Programming Interface

VRM — Victron Remote Monitoring

MQTT — Message Queuing Telemetry Transport

L2TP — Layer 2 Tunneling Protocol

HACS — Home Assistant Community Store

USB — Universal Serial Bus

SMS — Short Message Service

BLE — Bluetooth Low Energy

EEPROM — Electrically Erasable Programmable Read-Only Memory

ISAKMP — Internet Security Association and Key Management Protocol

NAT — Network Address Translation

HTTP — Hypertext Transfer Protocol

BMS — Battery Management System

IOS — iPhone Operating System

HTML — HyperText Markup Language

					<i>PE22.461142.001 ПЗ</i>	Лис
						4
Зм.	Лис	№ докум.	Підпис	Дата		

ВСТУП

Нинішній стан енергетичного сектору України характеризується безпрецедентними викликами, які зумовлені тривалою повномасштабною збройною агресією росії, та систематичними масованими атаками на об'єкти критичної енергетики, а також на локальні розподільчі станції. Для мирних користувачів, як наслідком руйнівних дій, став стійкий дефіцит генеруючих потужностей, який в свою чергу призвів до аварійних відключень та впровадження жорстких графіків обмеження електроспоживання. Щонайбільше до двох годин світла на добу. У таких реаліях критичним завданням став перехід до концепції децентралізації енергомереж та масове впровадження локальних автономних систем відновлюваної та накопичувальної енергетики.

Аналіз вітчизняного ринку свідчить про стрімке насичення споживчого та промислового секторів малогабаритними портативними системами енергозабезпечення. Проте, значна частина масових у виробництві та постачанні портативних джерел живлення з метою оптимізації собівартості виробництва позбавлені інтегрованих мережевих інтерфейсів дальньої дії, таких як Wi-Fi або GSM-модулі. Єдиним штатним засобом комунікації з такими пристроями є бездротовий інтерфейс локального радіуса дії Bluetooth Low Energy.

Це створює серйозну практичну проблему: за умов відсутності користувача в безпосередній близькості до об'єкта повністю втрачається можливість оперативного контролю параметрів енергосистеми, та дистанційного керування силовими інверторами пристрою. За умов тривалих відключень світла відсутність такого моніторингу суттєво підвищує ризики раптового знеструмлення підключеного критичного обладнання через непередбачуваний розряд батареї. Існуючі комерційні хмарні рішення від виробників часто мають закриту архітектуру та залежать від сторонніх серверів, або вимагають дорогої модернізації обладнання. Таким чином виникає гостра необхідність розробки та впровадження проміжного програмно-апаратного шлюзу на базі відкритих технологій, який виконуватиме роль інтелектуального містка між пристроєм та користувачем.

					РЕ22.461142.001 ПЗ	Лис
						5
Зм.	Лис	№ докум.	Підпис	Дата		

Для вирішення поставленої проблеми запропоновано комплексний підхід, що базується на інтеграції сучасних мікропроцесорних засобів, платформ автоматизації з відкритим вихідним кодом та передових технологій мережевої безпеки. Апаратною основою шлюзу обрано одноплатний мікрокомп'ютер Raspberry Pi 4 Model B [1]. Його обчислювальна потужність, наявність вбудованого апаратного модуля Bluetooth, низьке енергоспоживання та підтримка повноцінних операційних систем на базі ядра Linux дозволяють розгорнути стійку локальну службу безперервного опитування зарядної станції, здатну обробляти криптографічні алгоритми шифрування без затримок.

Програмною платформою було обрано модульне середовище Home Assistant, яке забезпечує гнучкість, наявність розвиненої екосистеми інтеграцій та можливість повної локальної ізоляції системи від публічних хмарних сервісів. Для оперативного інформування розроблено односторонню інтеграцію з месенджером Telegram за допомогою захищеного API-бота. Оскільки віддалене керування енергоживленням є критично вразливою точкою для несанкціонованого доступу, у проєкті задіяно професійне мережеве обладнання компанії MikroTik. Безпека передачі даних реалізується шляхом побудови захищених віртуальних приватних мереж VPN типу L2TP. Повна ізоляція технологічного трафіку від публічного інтернету, використання стійких алгоритмів шифрування та протоколів аутентифікації гарантують конфіденційність, цілісність даних та захист системи від зовнішнього втручання.

Об'єктом дослідження є процес дистанційного моніторингу та криптозахищеного керування параметрами автономних малогабаритних систем енергозабезпечення в умовах обмеженості штатних інтерфейсів зв'язку. Предметом дослідження виступають методи, програмно-апаратні засоби, протоколи бездротового зв'язку, мережеві конфігурації та криптографічні алгоритми захисту даних, що використовуються для створення шлюзу моніторингу на базі платформи Home Assistant та мікрокомп'ютера Raspberry Pi 4. Метою проєкту є розробка, конфігурування та практична реалізація захищеного віддаленого програмно-апаратного шлюзу для забезпечення безперервного дистанційного

					<i>PE22.461142.001 ПЗ</i>	Лис
						6
Зм.	Лис	№ докум.	Підпис	Дата		

контролю і керування параметрами зарядної станції Bluetti AC180, що дозволить підвищити рівень автономності, безпеки та енергоефективності підключених об'єктів.

Створена система має універсальну архітектуру і може бути масштабована чи адаптована для використання у різних галузях. У сфері моніторингу та автоматизації об'єктів критичної інфраструктури, розробка забезпечує стабільне функціонування, контроль залишку ємності та віддалений перезапуск комутаційного чи серверного обладнання на віддалених або важкодоступних технологічних майданчиках.

У секторі цивільного захисту та приватних домогосподарств система дозволяє інтегрувати портативні джерела живлення в загальний контур розумного будинку в зонах підвищеного ризику, здля створення гнучких сценаріїв автоматизації, таких як, наприклад, автоматичне відсікання другорядних споживачів чи оптимізація заряджання під час появи мережі. Для автономних військових та відомчих систем зв'язку застосування розробки в польових умовах, командно-штабних пунктах або мобільних вузлах зв'язку завдяки повній ізоляції трафіку через маршрутизатори MikroTik та шифруванню VPN-каналів задовольняє суворі вимоги до конфіденційності.

					РЕ22.461142.001 ПЗ	Лис
						7
Зм.	Лис	№ докум.	Підпис	Дата		

1 АНАЛІЗ ІСНУЮЧОГО СТАНУ ОБ'ЄКТУ ПРОЄКТУВАННЯ

1.1 Аналіз існуючих технічних рішень на ринку

Дослідження теперішнього стану енергетичного сектору України вказує на стрімку зміну парадигми споживання. Масовані удари росії по об'єктах генерації спричинили гостру кризу в енергетиці. Це змусило майже всю країну шукати альтернативні варіанти для забезпечення автономності. Наслідком цього процесу стало масоване насичення внутрішнього ринку портативними джерелами живлення різної потужності, і, відповідно, різної цінової категорії. Проте детальне вивчення ринку готових пристроїв показало серйозну інженерну проблему. На ринку відсутні універсальні та доступні засоби для організації віддаленого моніторингу систем малої енергетики, тобто дешевих станцій, які б одночасно гарантували високий рівень криптографічного захисту передачі даних.

Більшість існуючих систем автоматизації та контролю від відомих світових брендів спираються на використання пропрієтарних протоколів зв'язку. Такі протоколи мають повністю закриту архітектуру. Це унеможлиблює їх пряму інтеграцію в сторонні комплекси управління розумним будинком. Користувач фактично стає заручником програмної екосистеми одного конкретного виробника заліза. Будь-яка спроба розширити функціонал або підключити пристрої інших брендів стикається з повною несумісністю інтерфейсів на програмному рівні.

1.1.1 Модернізація апаратної бази шляхом придбання дорожчого обладнання

Перший з найбільш очевидних способів вирішення проблеми віддаленого доступу полягає у первинному виборі дорожчого класу обладнання. Мова йде про придбання промислових або преміальних зарядних комплексів типу Bluetti EP500 (рис. 1.1) або EcoFlow Delta Pro (рис. 1.2). Такі системи комплектуються інтегрованими мережевими інтерфейсами бездротового зв'язку Wi-Fi безпосередньо на заводі виробника.

					РЕ22.461142.001 ПЗ	Лис
						8
Зм.	Лис	№ докум.	Підпис	Дата		



Рисунок 1.1 — Bluetti EP500



Рисунок 1.2 — Ecoflow Delta Pro

					РЕ22.461142.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		9

Зазначені пристрої дозволяють здійснювати повноцінний дистанційний моніторинг усіх електричних параметрів через мережу інтернет. Безпосередньо через екосистему виробника. Безкоштовно, або в деяких випадках за підписку на сервіс. Користувач отримує доступ до фірмового графічного інтерфейсу та хмарної інфраструктури з будь-якої точки світу. Однак такий підхід має кілька критичних недоліків, які роблять його неприйнятним для багатьох сценаріїв.

По-перше — низька економічна ефективність даного рішення. Фінансова вартість професійних енергетичних станцій із вбудованими мережевими платами в кілька разів перевищує сумарний бюджет розгортання гнучкої системи на базі доступної станції Bluetti AC180 та мікрокомп'ютера Raspberry Pi 4. Для масового впровадження в умовах кризи такий підхід є фінансово недоцільним.

По-друге — повна втрата конфіденційності інформації. Навіть найдорожчі преміальні моделі здійснюють передачу телеметрії через публічні сервери виробника. Часто ці сервери фізично розташовані в інших країнах. Такий принцип обміну даними є категорично неприпустимим для об'єктів критичної інфраструктури, військових підрозділів або відомчих установ. На подібних об'єктах весь технологічний трафік повинен суворо залишатися всередині локального криптозахищеного периметра безпеки.

По-третє — ефект технологічної прив'язки до одного постачальника. Це унеможлиблює створення єдиного консолідованого центру диспетчеризації у майбутньому. Якщо виникне потреба додати в загальну мережу дешевші акумулятори чи інвертори інших марок, об'єднати їх в один інтерфейс буде неможливо.

1.1.2 Функціональний аналіз штатного програмного забезпечення Bluetti App

Переважає більшість сучасних портативних джерел живлення середнього цінового сегменту поставляються на ринок із базовим програмним забезпеченням. Типовим прикладом є фірмовий мобільний застосунок Bluetti App (рис. 1.3). Він розроблений для швидкого налаштування пристрою користувачем.

					РЕ22.461142.001 ПЗ	Лис
						10
Зм.	Лис	№ докум.	Підпис	Дата		

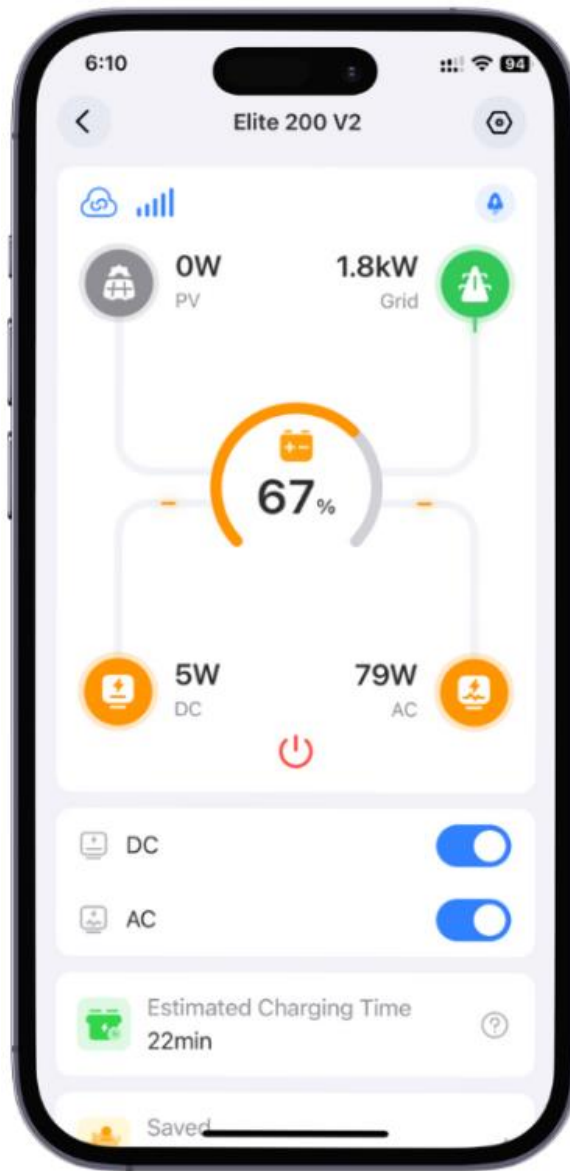


Рисунок 1.3 — Інтерфейс мобільного застосунку Bluetti App

Дане програмне рішення забезпечує контроль усіх доступних параметрів станції та дозволяє безпосередньо керувати силовими реле. Комунікація відбувається через бездротовий інтерфейс локального радіуса дії Bluetooth Low Energy. Радіус стабільного покриття за такого підключення обмежується відстанню до 10–15 метрів у межах прямої видимості.

Головним недоліком штатного програмного забезпечення є повна відсутність віддаленого доступу. У базових і бюджетних моделях станцій фізично відсутній Wi-Fi модуль. Через це дистанційний контроль параметрів за межами квартири чи офісу стає технічно неможливим. Користувач повністю втрачає інформацію про стан енергосистеми одразу після виходу з приміщення.

					<i>PE22.461142.001 ПЗ</i>	Лис
						11
Зм.	Лис	№ докум.	Підпис	Дата		

Крім того, додаток має жорстку хмарну залежність для авторизації профілю. Навіть у тих модифікаціях, де присутній Wi-Fi, весь потік даних обов'язково проходить через зовнішні закордонні сервери компанії. Це створює загрозу повної втрати контролю над живленням об'єкта у разі відключення зовнішнього інтернету чи збоїв на серверах розробника.

Але у цього варіанта також є один великий плюс. Він повністю сумісний з усіма Bluetti станціями, шифрованими і ні. Немає необхідності вигадувати, як обійти шифрування станції.

1.1.3 Професійні системи моніторингу на прикладі платформи Victron Energy VRM

У промисловому секторі та професійній сонячній енергетиці широкого поширення набули спеціалізовані екосистеми дистанційного керування. Найбільш відомим представником цього класу є хмарна платформа Victron Remote Management від компанії Victron Energy (рис. 1.4).

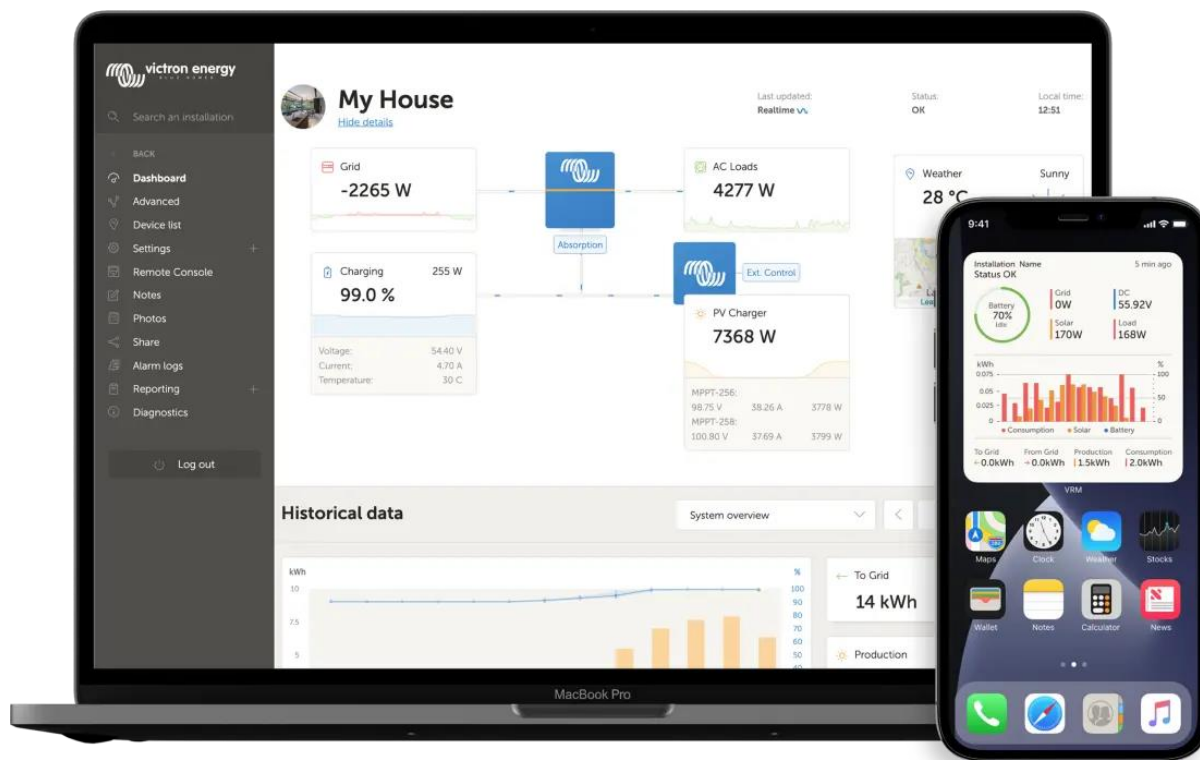


Рисунок 1.4 — Робочий простір професійної платформи Victron Remote Management

До очевидних переваг даного комплексу слід віднести дуже високу надійність роботи. Система надає максимально детальну телеметрію та підтримує

					РЕ22.461142.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		12

велику кількість промислових протоколів передачі даних на великі відстані. Платформа дозволяє будувати складні графіки та аналізувати стан великих акумуляторних масивів у реальному часі.

Основним стримуючим фактором для впровадження рішень від Victron Energy є надзвичайно висока вартість апаратних модулів та супутніх компонентів. Сама архітектура екосистеми є повністю закритою для стороннього коду. Інтеграція дешевших побутових пристроїв інших виробників у цю систему вимагає значних зусиль або є взагалі неможливою через програмні обмеження.

1.1.4 Готові апаратні Bluetooth шлюзи екосистем Xiaomi та Tuuya

На споживчому ринку існують готові компактні рішення Bluetooth шлюзів. Мова йде про апаратні Bluetooth шлюзи та хаби від відомих брендів масового сегменту розумного будинку, таких як Xiaomi (рис. 1.5) або Tuuya Smart (рис.1.6). Ці пристрої є досить доступними за ціною. Їхня вартість варіюється в межах від 400 до 2500 гривень, залежно від конкретної моделі та від наявності додаткових інтерфейсів.



Рисунок 1.5 —Bluetooth шлюз Xiaomi

					РЕ22.461142.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		13



Рисунок 1.6 —Bluetooth шлюз TuYa

Основним призначенням таких «хабів» є збір даних з локальних Bluetooth датчиків та подальша трансляція цієї інформації в мережу через домашній роутер. Проте аналіз можливостей даних пристроїв показав їх повну непридатність для виконання завдань дипломного проєкту. Головний мінус полягає в тому, що всі подібні шлюзи жорстко розраховані на роботу всередині своєї конкретної комерційної екосистеми. Платформа TuYa не здатна розпізнати специфічні пропріетарні BLE пакети даних від зарядної станції компанії Bluetti. Сумісність між цими пристроями повністю відсутня на рівні базового коду прошивки.

Другим критичним недоліком є питання безпеки та надійності. Всі зібрані дані такі шлюзи відправляють виключно в публічну хмару виробника, зазвичай китайську. Робота без постійного доступу до глобальної мережі інтернет для них не передбачена конструктивно. Передача інформації про стан резервного живлення критичного об'єкта через сторонні китайські сервери не може вважатися надійним чи безпечним рішенням. У випадку кібератаки на сервери або під час звичайного зникнення зв'язку з провайдером користувач повністю втрачає контроль над системою автоматизації.

					РЕ22.461142.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		14

1.1.5 Підсумки проведеного аналізу та обґрунтування концепції власної розробки

Детальний аналіз існуючих комерційних та побутових рішень дозволяє нам зробити дуже важливий і технічно правильний висновок. Жоден із доступних на ринку варіантів не здатний одночасно забезпечити повну автономність від сторонніх хмарних провайдерів та високий рівень криптографічного захисту технологічного трафіку. Професійні системи автоматизації мають надмірну вартість для масового розгортання. Доступні побутові пристрої обмежені малим радіусом дії Bluetooth та вразливі через архітектурну прив'язку до закордонних інтернет-серверів. Безперечно, навіть попри значну кількість недоліків та невелику кількість готових рішень, вони мають і плюси. Легке розгортання, але в межах своєї інфраструктури. Легкість досягається тим, що користувачу не потрібно нічого вигадувати, включив і воно працює. Не треба налаштовувати MQTT чи щось ще.

Рішення, яке використовується в цьому дипломному проєкті, пропонує принципово інший підхід до вирішення актуальної проблеми. Використання одноплатного мікрокомп'ютера Raspberry Pi 4, як локального інтелектуального шлюзу дозволяє безпосередньо знімати телеметрію зі станції Bluetti AC180 по локальному Bluetooth каналу. Встановлений Home Assistant на Дані обробляються повністю всередині локального контуру без задіяння сторонніх серверів. Так, є свої недоліки, наприклад, оскільки Raspberry Pi 4 має лише один Bluetooth модуль, можна керувати лише однією станцією.

Для виконання поставленої задачі у вигляді передачі інформації на великі відстані використовується мережеве обладнання компанії MikroTik. Дане високотехнологічне обладнання забезпечує надійне транспортування трафіку(паролі, команди, відгуки) через спеціально налаштований криптозахищений L2TP VPN тунель. Такий підхід повністю ізолює технологічний трафік від публічного інтернету. Це гарантує абсолютну конфіденційність та захист системи від зовнішнього втручання. Саме в поєднанні відкритої гнучкої платформи автоматизації з професійними мережевими технологіями захисту і полягає

					PE22.461142.001 ПЗ	Лис
						15
Зм.	Лис	№ докум.	Підпис	Дата		

головна новизна та практична цінність розробленого програмно-апаратного шлюзу.

Для наочного порівняння та комплексного оцінювання можливостей усіх проаналізованих технічних підходів доцільно звести їхні ключові інженерні та економічні характеристики в єдину порівняльну матрицю. Це дозволить чітко продемонструвати переваги розробленого програмно-апаратного шлюзу на тлі існуючих ринкових альтернатив.

Таблиця 1.1 — Порівняльна характеристика існуючих рішень

	Преміальні станції	Штатний застосунок Bluetti App	Професійні системи Victron VRM	Готові шлюзи Xiaomi або TuYa
Наявність віддаленого доступу	Повноцінний через інтернет	Відсутній (лише Bluetooth)	Повноцінний через інтернет	Повноцінний через інтернет
Залежність від сторонніх хмарних серверів	Повна	Відсутня (локальна робота)	Повна	Повна
Робота системи при відключенні інтернету	Блокується віддалений контроль	Працює в радіусі Bluetooth	Блокується збір метрик	Блокується автоматизація
Економічна доступність розгортання	Низька	Максимальна	Низька	Висока
Захист технологічного трафіку	Базове ари	Локальний	Пропріетарне хмарне шифрування	Базове хмарне шифрування

Обхід вбудованого BLE шифрування	Не потрібен	Нативний	Немає	Немає
Пряме керування інвертором та реле	Доступно через хмару	Доступно в радіусі Bluetooth	Доступно через спецмодулі	Немає
Інтеграція пристроїв різних виробників	Неможлива	Неможлива	Обмежена промисловими протоколами	Обмежена екосистемою бренду
Ризик блокування з боку вендора	Високий	Високий	Високий	Високий
Створення гнучких сценаріїв автоматизації	Базова в межах додатку	Відсутня	Є	Базова в межах додатку
Миттєві push сповіщення	Через хмарний додаток	Відсутні	Через електронну пошту	Через хмарні push-сервіси
Необхідність стороннього софту на телефонії	Вимагає фірмовий додаток	Вимагає фірмовий додаток	Вимагає фірмовий додаток	Вимагає фірмовий додаток

2 ОБГРУНТУВАННЯ НАПРЯМКІВ ДОСЛІДЖЕННЯ ТА ПРОЄКТНИХ РІШЕНЬ

2.1 Визначення вимог до апаратної бази

Процес проектування та створення ефективного програмно-апаратного шлюзу вимагає ретельного підходу до вибору центрального обчислювального елемента. Спочатку планувалось здешевити виріб і використати шлюз на базі ESP32 (рис. 2.1).

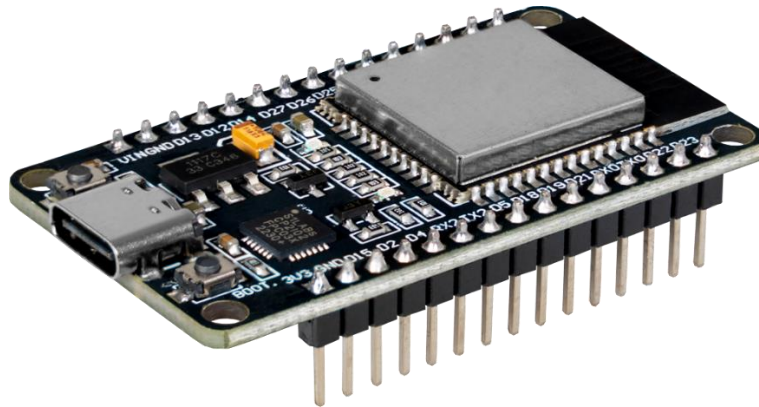


Рисунок 2.1 – ESP32

Проте, головною перепоною у виборі архітектури полягає в особливостях обміну даними із зарядною станцією Bluetti AC180. Сучасні версії фірмових прошивок використовують зашифровані пакети та протоколи авторизації в рамках бездротової технології Bluetooth Low Energy. Офіційно дешифратором виступає додаток Bluetti App, проте цього нам не вистачить. На етапі концептуального проектування розглядалися два протилежних варіанти побудови апаратного містка. Першим варіантом було використання ультрабюджетного мікроконтролера типу ESP32 із вбудованими модулями зв'язку. Другим варіантом виступало застосування повноцінного одноплатного мікрокомп'ютера типу Raspberry Pi 4 Model B (рис. 2.2).

					РЕ22.461142.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		18



Рисунок 2.2 — Raspberry Pi 4 Model B

Застосування мікроконтролера ESP32 має суттєві переваги з точки зору мінімального енергоспоживання, миттєвого старту операційної системи та низької фінансової вартості заліза. Проте детальний аналіз обчислювальних можливостей архітектури даного чіпа виявив його повну непридатність для виконання поставлених завдань. Обмежений об'єм оперативної пам'яті та низька частота процесора не дозволяють у реальному часі виконувати складні математичні операції дешифрації Bluetooth трафіку від декількох потоків даних. На базі мікроконтролера неможливо підняти повноцінний локальний стек служб безпеки.

Вибір одноплатного мікрокомп'ютера Raspberry Pi 4 Model B забезпечує систему значним запасом продуктивності. Наявність потужного чотириядерного процесора та великого об'єму оперативної пам'яті дозволяє без затримок обробляти криптографічні алгоритми. Це створює запас для майбутнього розширення екосистеми. Вбудований апаратний радіомодуль з підтримкою сучасних стандартів бездротового зв'язку гарантує високу чутливість при роботі з Bluetooth периферією. Додатковим практичним чинником вибору даного обладнання стала його безпосередня наявність. Це дозволило повністю виклю-

чити додаткові фінансові витрати на закупівлю заліза та прискорити етап прототипування. На основі цих технічних та економічних аргументів для реалізації проекту було обрано платформу Raspberry Pi 4 Model B.

2.2 Обґрунтування вибору програмного забезпечення середовища управління

Опісля остаточного визначення апаратної бази постало завдання вибору програмного середовища для реалізації логіки збору метрик та автоматизації. На етапі системного аналізу розглядалися два різні інженерні підходи. Першим варіантом було розгортання на мікрокомп'ютері Android-емулятора для запуску оригінального мобільного застосунку виробника. Другим, більш прогресивним варіантом, виступало встановлення спеціалізованої серверної платформи Home Assistant з відкритим вихідним кодом.

Використання Android-емулятора в умовах одноплатного комп'ютера є повністю неефективним. Процес емуляції мобільної операційної системи вимагає трансляції інструкцій та постійної роботи віртуального графічного процесора, що створює колосальне навантаження на залізо. Це призводить до постійного перевантаження оперативної пам'яті, сильного перегріву центрального процесора та значного підвищення споживання електричної енергії. Крім того, архітектура мобільних додатків розрахована на сенсорний екран смартфона, тому інтегрувати такий інтерфейс у серверне рішення без графічної оболонки технічно неможливо.

Такий підхід повністю виключає можливість побудови власних алгоритмів автоматизації або зв'язку з іншим мережевим обладнанням. Користувач залишається обмеженим базовим інтерфейсом фірмового додатка без жодної можливості масштабування системи в майбутньому. Також виникають серйозні інженерні труднощі з налаштуванням захищеного віддаленого доступу та організацією миттєвого надсилання сповіщень про критичні події, так званих алертів, оскільки емулятор часто блокує фонові мережеві процеси.

Впровадження серверної платформи Home Assistant забезпечує мінімальне використання системних ресурсів завдяки оптимізованому коду ядра та

					PE22.461142.001 ПЗ	Лис
						20
Зм.	Лис	№ докум.	Підпис	Дата		

повній відсутності важких графічних оболонок на серверній стороні. Це програмне забезпечення має гнучку модульну структуру і підтримує запуск у контейнеризованому середовищі на базі Docker. Використання контейнерів дозволяє повністю ізолювати процеси платформи від основної операційної системи мікрокомп'ютера, що суттєво підвищує загальний рівень відмовостійкості та безпеки шлюзу. Платформа дозволяє збирати, структурувати та довгостроково зберігати історичні дані про роботу станції в локальній базі даних. Це дає можливість аналізувати графіки навантаження, відстежувати деградацію акумулятора та прогнозувати час автономної роботи об'єкта.

Головною інженерною перевагою Home Assistant є реалізація концепції Cloud-free, тобто повної незалежності від зовнішніх хмарних провайдерів. Весь рушій автоматизацій та логіка обробки даних працюють виключно локально всередині пам'яті Raspberry Pi 4. Система залишається повністю працездатною навіть під час тотального зникнення підключення до глобальної мережі інтернет або у разі збоїв та кібератак на сервери виробника зарядної станції. Платформа підтримує роботу з величезною кількістю мережевих протоколів та відкритих інтерфейсів, що дозволяє легко масштабувати комплекс. Виходячи з міркувань високої швидкодії, архітектурної гнучкості, безпеки та стабільності в екстремальних умовах, для реалізації проєкту було обрано систему Home Assistant.

2.3 Вибір оптимального методу інтеграції периферійного пристрою в екосистему

Задля забезпечення взаємодії між операційною системою шлюзу та внутрішніми регістрами зарядної станції необхідно було визначити метод програмної інтеграції пристрою. У ході дослідження аналізувалися два основні інженерні підходи. Перший підхід базувався на використанні універсального легкого протоколу передачі повідомлень MQTT через відповідний локальний брокер. Другий підхід передбачав встановлення спеціалізованих кастомних компонентів з відкритого репозиторію GitHub за допомогою системи розширення функціоналу HACS.

					PE22.461142.001 ПЗ	Лис
						21
Зм.	Лис	№ докум.	Підпис	Дата		

Протокол MQTT являє собою спеціалізований мережевий протокол передачі даних між пристроями, який працює за принципом видавця та підписника. Він розроблений спеціально для систем інтернет-речей та міжмашинної взаємодії в умовах обмеженої пропускну здатності каналів зв'язку. Центральним елементом такої мережі є MQTT брокер, який приймає повідомлення від одних пристроїв та перенаправляє їх іншим за певними текстовими темами.

Використання протоколу MQTT є класичним рішенням для систем інтернет-речей завдяки мінімальному розміру пакетів, низьким вимогам до апаратної частини та високій швидкості передачі даних в локальній мережі. Однак стандартні загальнодоступні профілі MQTT для роботи з обладнанням Bluetti часто обмежені виключно функціями пасивного моніторингу. Вони дозволяють зчитувати поточні показники напруги чи заряду, але в даній інтеграції не мають зворотного каналу зв'язку для керування силовими реле. Це повністю суперечить меті проекту, яка вимагає реалізації віддаленого управління інверторами. Я пов'язую це з особливістю зашифрованого зв'язку станції.

Альтернативний підхід передбачає використання екосистеми розробників навколо веб-сервісу GitHub, який є найбільшою у світі платформою для хостингу ІТ-проектів та їх спільної розробки. Цей ресурс дозволяє програмістам з усього світу публікувати відкритий вихідний код своїх рішень, відстежувати помилки та спільно модернізувати програмні продукти. Саме у відкритих репозиторіях цієї платформи ентузіастами було створено спеціалізовані програмні модулі для прямої взаємодії з процесорами портативних станцій.

Для зручної інтеграції таких сторонніх рішень у середовище автоматизації використовується інструмент HACS, який розшифровується як кастомний магазин додатків для Home Assistant. Цей інструментарій виступає в ролі менеджера пакетів і дозволяє користувачам встановлювати, оновлювати та керувати елементами, які офіційно не входять у базову збірку операційної системи розумного будинку, але доступні на платформі GitHub.

Встановлення кастомного компонента з репозиторію GitHub за допомогою системи HACS дозволяє отримати повний доступ до низькорівневого коду взаємодії з процесором станції. Цей програмний модуль підтримує повноцінну

					PE22.461142.001 ПЗ	Лис
						22
Зм.	Лис	№ докум.	Підпис	Дата		

двосторонню комунікацію. Він транслює в інтерфейс Home Assistant не лише інформаційні сенсори, а й активні органи управління. До них відносяться перемикачі режимів роботи, активація виходів змінного та постійного струму, а також зміна параметрів швидкості заряджання батареї. На основі цього було прийнято рішення використовувати спеціалізовану інтеграцію з відкритого репозиторію.

2.4 Організація захищеного віддаленого доступу та вибір мережевого протоколу

Питання безпечного підключення до інтерфейсу управління з глобальної мережі є найважливішою частиною проєкту, оскільки несанкціонований доступ до енергетичної системи може призвести до її виведення з ладу. Першим варіантом організації каналу зв'язку був розгляд програмних мереж типу ZeroTier. Ця технологія дозволяє створювати віртуальні локальні мережі поверх публічного інтернету, об'єднуючи віддалені пристрої так, ніби вони підключені до одного комутатора. Програмні рішення типу ZeroTier приваблюють простотою налаштування та можливістю обходу складних мережевих конфігурацій чи провайдерського блокування без необхідності купівлі статичної IP-адреси.

Проте робота таких мереж критично залежить від сторонніх центральних серверів координації компанії-розробника. Це відображається на загальній стабільності, створює додаткову точку відмови та потенційну вразливість системи для цілеспрямованих кібератак на хмарну інфраструктуру. Крім того, встановлення додаткового клієнтського програмного забезпечення на мобільні пристрої користувача ускладнює повсякденну експлуатацію технологічного контуру.

В якості альтернативи також детально розглядалося популярне сучасне рішення на базі протоколу WireGuard. Цей протокол відрізняється високою швидкістю роботи, мінімальним кодом та сучасними алгоритмами шифрування. Однак використання WireGuard має такий самий суттєвий недолік, як і

					<i>РЕ22.461142.001 ПЗ</i>	Лис
						23
Зм.	Лис	№ докум.	Підпис	Дата		

ZeroTier. Він вимагає обов'язкового встановлення та постійної роботи окремого фірмового додатка на мобільному телефоні користувача. В умовах повсякденного використання це є мінусом, оскільки на пристрої користувача може банально не вистачати вільного місця в пам'яті, або операційна система смартфона може жорстко закривати фоновий додаток задля економії заряду батареї.

Через ці обмеження було прийнято рішення орієнтуватися на використання вбудованого VPN-клієнта, який інтегрований в операційну систему мобільного телефону за замовчуванням. Побудова захищеного каналу на базі високопродуктивного маршрутизатора MikroTik дозволяє використовувати нативні технології операційної системи RouterOS. Під час аналізу розглядалися різні типи вбудованих тунелів, включаючи IKEv2, IPsec та L2TP. Враховуючи широку підтримку рідних інтерфейсів у мобільних операційних системах, остаточний вибір було зроблено на користь протоколу L2TP.

Даний протокол має певні архітектурні недоліки, серед яких дещо вища надлишковість трафіку та більша чутливість до блокування окремих портів провайдерами. Проте він забезпечує достатній рівень безпеки для приватного контуру та є значно простішим у процесі первинного налаштування, ведення системних логів і діагностики можливих помилок на стороні маршрутизатора MikroTik. Це забезпечує високу надійність зв'язку та швидке відновлення сесії без задіяння стороннього софту.

2.4.1 Концепція застосування менеджерів паролів

Під час розгортання та практичного підняття мережових віртуальних тунелів критично важливим аспектом безпеки стає захист від методів автоматичного перебору облікових даних. Протокол L2TP для забезпечення надійного шифрування вимагає використання складних та довгих паролів як для автентифікації конкретного користувача, так і для спільного секретного ключа IPsec. Використання простих комбінацій символів створює пряму загрозу успішного зламу системи злоумисниками. Пам'ятати велику кількість складних

					РЕ22.461142.001 ПЗ	Лис
						24
Зм.	Лис	№ докум.	Підпис	Дата		

криптографічних ключів для людини є неможливим завданням, тому виникає гостра необхідність впровадження спеціалізованих засобів захисту інформації.

Для вирішення цієї проблеми в рамках проєкту було обґрунтовано обов'язкове використання менеджера паролів. Менеджер паролів являє собою спеціалізоване програмне забезпечення, призначене для безпечного генерації, зберігання та систематизації зашифрованих облікових даних. Всі паролі зберігаються у спеціальній базі даних, яка захищена за допомогою стійкого алгоритму симетричного шифрування AES-256. Доступ до всієї бази здійснюється за допомогою одного унікального майстер-пароля, який знає лише безпосередній адміністратор системи.

Зазначений алгоритм AES-256 являє собою передовий міжнародний стандарт блочного симетричного шифрування з довжиною ключа 256 біт. Принцип його дії базується на розбитті вхідного масиву інформації на фіксовані 128-бітні блоки з їх наступним перетворенням у двовимірні матриці байтів. Процес захисту даних складається з 14 послідовних раундів складних математичних та логічних трансформацій, які включають нелінійну заміну байтів, циклічні зсуви рядків, перемішування стовпчиків за правилами полів Галуа та додавання підключей за допомогою операції виключного або. Астрономічна кількість можливих комбінацій ключа повністю виключає можливість успішного проведення атак методом простого перебору за допомогою сучасних обчислювальних комплексів чи перспективних квантових комп'ютерів. Завдяки такій безпрецедентній стійкості алгоритм офіційно сертифікований на урядовому рівні для захисту інформації з найвищим ступенем таємності.

В якості конкретного інструменту для реалізації цього завдання було обрано кросплатформений менеджер паролів Bitwarden. Ця програма підтримує повноцінне функціонування на різних типах пристроїв. Існують офіційні версії як для персональних комп'ютерів у вигляді окремих додатків чи розширень браузерів, так і мобільні клієнти для смартфонів під управлінням різних операційних систем. Це забезпечує миттєву синхронізацію даних між комп'ютером адміністратора та телефоном користувача під час підключення до мережі.

					<i>РЕ22.461142.001 ПЗ</i>	Лис
						25
Зм.	Лис	№ докум.	Підпис	Дата		

Дане програмне забезпечення є повністю безпечним для використання у критичних системах. Безпека Bitwarden гарантується архітектурою нульового розголошення, за якої шифрування та дешифрування бази даних відбувається виключно на локальному пристрої користувача. Хмарний сервер отримує лише повністю зашифрований масив інформації. Це унеможливорює витік даних навіть у разі повного злому серверів компанії-розробника. Додатковим фактором надійності є повністю відкритий вихідний код, що дозволяє світовій спільноті незалежних експертів з безпеки постійно тестувати систему на відсутність прихованих уразливостей.

Система Bitwarden дозволяє зручно та безпечно зберігати в одному зашифрованому сховищі всі автентифікаційні дані проекту. Сюди відносяться секретні ключі та паролі від VPN-тунелів маршрутизатора MikroTik, а також облікові записи для доступу до внутрішнього веб-інтерфейсу серверної платформи Home Assistant. Це дозволяє генерувати випадкові паролі максимальної довжини для кожного окремого сервісу, суттєво підвищуючи загальний рівень захищеності розробленого програмно-апаратного шлюзу.

2.5 Проектування системи оперативного інформування та моніторингу подій

Для підтримки постійної обізнаності про стан віддаленого об'єкта без необхідності безперервного утримання активної сесії віртуальної приватної мережі була спроектована комплексна підсистема сповіщень. На етапі розробки архітектури аналізувалися декілька технічних каналів доставки інформації та спеціалізованих платформ моніторингу. Зокрема розглядалися варіанти використання професійного серверного комплексу моніторингу MikroTik The Dude, класична поштова служба на базі протоколу SMTP, відправка прямих мобільних SMS-повідомлень, а також інтеграція локальних засобів звукового та світлового оповіщення.

Використання спеціалізованого програмного комплексу MikroTik The Dude розглядалося як варіант для глибокого контролю мережевої інфраструк-

					<i>PE22.461142.001 ПЗ</i>	Лис
						26
Зм.	Лис	№ докум.	Підпис	Дата		

тури. Даний безкоштовний інструмент від виробника маршрутизаторів призначений для автоматичного сканування мереж, побудови інтерактивних карт зв'язків та моніторингу працездатності окремих хостів за протоколами ICMP, SNMP або DNS. Система The Dude є надзвичайно ефективною для великих корпоративних мереж та інтернет-провайдерів, оскільки дозволяє бачити загальну архітектуру та виявляти аварійні вузли на ранніх етапах. Проте для даного проєкту ця платформа виявилася надмірною та складною. Вона вимагає розгортання окремого серверного процесу, сильно завантажує обчислювальні ресурси та не має зручних штатних засобів для зчитування специфічних параметрів акумуляторів через Bluetooth інтерфейс, оскільки орієнтована суто на мережеві хости.

Другим класичним інструментом виступала поштова служба сповіщень на базі протоколу SMTP. Використання поштових повідомлень є надійним світовим стандартом для серверних систем, але воно має занадто великі затримки у часі доставки критично важливої інформації. В умовах енергетичної кризи, коли ситуація на об'єкті може кардинально змінитися протягом кількох хвилин, затримка листа на транзитних поштових серверах є неприпустимою. Також стандартний поштовий інтерфейс не дозволяє оперативно реалізувати зручні механізми зворотного керування системою за допомогою текстових команд чи швидких інтерактивних кнопок.

Третім варіантом було проектування шлюзу для відправки SMS-повідомлень безпосередньо на мобільний телефон користувача через стільникову мережу. Головною перевагою SMS є повна незалежність від наявності мобільного інтернету, оскільки повідомлення доставляються через базові голосові канали зв'язку GSM. Проте впровадження такого рішення вимагає обов'язкового підключення до мікрокомп'ютера додаткового апаратного USB-модема або використання платних хмарних SMS-сервісів. Це суттєво здорожує загальну вартість системи та створює фінансову залежність від тарифів операторів мобільного зв'язку, що знижує економічну доцільність розробки.

					РЕ22.461142.001 ПЗ	Лис
						27
Зм.	Лис	№ докум.	Підпис	Дата		

Додатково розглядалася можливість підключення локальних фізичних сповіщувачів, таких як світлодіодні матриці або звукові п'єзоелектричні сирени, безпосередньо до платформи автоматизації. Такі пристрої забезпечують миттєву реакцію та гучний звуковий сигнал у разі аварії, якщо користувач перебуває всередині приміщення. Однак цей підхід є повністю неефективним, коли людина знаходиться поза межами будівлі або у від'їзді. Локальний звуковий сигнал у порожньому приміщенні не здатний попередити власника про критичний розряд акумулятора.

Через виявлені недоліки існуючих альтернатив остаточний вибір було зроблено на користь створення інтелектуального бота для сучасного месенджера Telegram. Цей крок забезпечує миттєву доставку важливих сповіщень на мобільний телефон користувача за допомогою безкоштовної технології push-повідомлень. Платформа Home Assistant має готові інтегровані інструменти для прямої взаємодії з офіційним Telegram API. Це дозволяє гнучко налаштувати структуру текстових повідомлень, автоматично додавати графіки споживання енергії у реальному часі та формувати звіти. Важливою інженерною перевагою є можливість реалізації інтерактивного меню з кнопками всередині чату. Це дозволяє користувачеві швидко вимикати другорядне навантаження, змінювати струм заряджання або перевіряти поточний статус станції безпосередньо з вікна месенджера, повністю виключаючи необхідність відкривати важкий веб-інтерфейс системи автоматизації.

2.6 Проектування інтерфейсу користувача

Важливою складовою частиною проектування системи є розробка графічного інтерфейсу користувача. Він повинен забезпечувати максимальну наочність, мінімальний час відгуку та зручність сприйняття інформації в умовах обмеженого часу або стресових ситуацій під час відключення світла. На даному етапі заплановано створення спеціалізованого інтерактивного дашборду в середовищі Home Assistant на базі модульних карток системи. Головним завданням цього інтерфейсу з точки зору концепції UI/UX є чітка анімована візуалізація поточних енергопотоків між зовнішньою промисловою мережею,

					<i>PE22.461142.001 ПЗ</i>	Лис
						28
Зм.	Лис	№ докум.	Підпис	Дата		

акумулятором зарядної станції Bluetti AC180 та кінцевими споживачами електричного струму. Користувач повинен з першого погляду оцінювати залишок часу автономної роботи, поточне навантаження в кіловатах та загальну ефективність системи.

Під час проєктування апаратної схеми та інтерфейсу було закладено важливу технічну умову: одноплатний мікрокомп'ютер Raspberry Pi 4 Model B житиметься безпосередньо від постійного струму через вбудований DC-вихід зарядної станції. Таке рішення дозволяє уникнути подвійного перетворення енергії через інвертор змінного струму та суттєво підвищує загальний коефіцієнт корисної дії шлюзу. Проте така схема живлення створює серйозний архітектурний ризик. Якщо користувач випадково натисне кнопку вимкнення DC-виходу в інтерфейсі Home Assistant, мікрокомп'ютер миттєво знеструмиться. Це призведе до повної втрати віддаленого доступу до системи без можливості дистанційного відновлення зв'язку.

Для нівелювання цієї загрози на етапі проєктування інтерфейсу було розроблено спеціальний програмний захист. У коді дашборду для елемента керування DC-виходом впроваджується обов'язкова функція підтвердження дії. При спробі натискання на перемикач система не виконує команду миттєво, а викликає додаткове спливаюче діалогове вікно з попередженням про ризик знеструмлення шлюзу. Тільки після повторного свідомого підтвердження користувачем команда передається на станцію. Крім того, на рівні рушія автоматизацій Home Assistant створюється захисний сценарій, який повністю блокує відправку команди вимкнення DC-порту, якщо система фіксує активну сесію моніторингу або роботу від акумулятора, що повністю виключає людський фактор випадкового обриву зв'язку.

Програма тестування на етапі проєктування передбачає обов'язкову перевірку працездатності всього комплексу при підключенні через захищений віддалений канал зв'язку. Необхідно провести точні вимірювання мережевих затримок передачі пакетів даних і оцінити коректність відображення динамічних графіків на екранах мобільних пристроїв. Окремим важливим інженерним завданням у рамках цього етапу є експериментальне дослідження споживання

					РЕ22.461142.001 ПЗ	Лис
						29
Зм.	Лис	№ докум.	Підпис	Дата		

електричної енергії самим обчислювальним модулем. Буде проведено порівняльний аналіз роботи плати Raspberry Pi 4 в умовах використання активного охолодження за допомогою додаткової плати з вентиляторами та при повністю пасивному відведенні тепла через радіатори. Це дозволить оцінити доцільність використання механічного охолодження, знайти баланс між тепловим режимом процесора та оптимізувати автономність усієї системи від акумулятора під час тривалого блекауту.

2.7 Висновки до другого розділу

У ході виконання другого розділу дипломного проєкту було проведено детальний аналіз, порівняння та обґрунтування всіх ключових архітектурних, апаратних та програмних рішень. На основі проведених інженерних розрахунків та оцінки обчислювальної потужності для реалізації шлюзу було остаточно обрано однокплатний мікрокомп'ютер Raspberry Pi 4 Model B. Програмною основою для збору метрик, логування та автоматизації затверджено платформу Home Assistant, розгорнуту в ізольованому контейнеризованому середовищі на базі Docker. Для забезпечення взаємодії з реєстрами зарядної станції по Bluetooth каналах обрано кастомний двосторонній компонент з відкритого репозиторію GitHub.

Для організації безпечного віддаленого доступу до технологічного контуру було відкинуто сторонні хмарні мережі та WireGuard через необхідність встановлення клієнтського софту на смартфоні. Замість цього обрано побудову нативного віртуального приватного тунелю за протоколом L2TP на базі професійного маршрутизатора MikroTik. Для забезпечення максимального захисту від перебору паролів упроваджено кросплатформений менеджер Bitwarden з алгоритмом симетричного шифрування AES-256.

Для підсистеми оперативного інформування було спроектовано інтелектуального чат-бота в месенджері Telegram. При цьому архітектурно закладено, що через Telegram-бота реалізується виключно пасивний моніторинг та отримання текстових статусів. Оскільки Telegram є публічним месенджером і теоретично піддається ризикам компрометації чи зламу облікових записів, через

					<i>PE22.461142.001 ПЗ</i>	Лис
						30
Зм.	Лис	№ докум.	Підпис	Дата		

нього повністю заборонено надсилання активних команд керування інверторами чи зміни налаштувань станції. Будь-які дії з перемикання реле та зміни режимів вимагають обов'язкового підключення до захищеного VPN-тунелю MikroTik та авторизації у веб-інтерфейсі Home Assistant. Такий розподіл прав гарантує, що навіть у разі повної компрометації чат-бота зловмисники не зможуть нашкодити обладнанню чи знеструмити об'єкт, що повністю задовольняє висунуті вимоги безпеки.

Для забезпечення системного підходу до конфігурування програмно-апаратного шлюзу всі розглянуті альтернативи та критерії їх вибору були зведені в єдину технічну матрицю прийняття проєктних рішень.

Таблиця 2.1 — Матриця обґрунтування та прийняття проєктних рішень

	Розглянуті альтернативні варіанти	Критерії порівняння та оцінки	Остаточне обране інженерне рішення	Обґрунту- вання фінального вибору
Вибір апаратного обчислювального модуля	Мікроконтролер ESP32 або мікрокомп'ютер Raspberry Pi 4	Обчислювальна потужність, об'єм пам'яті, наявність заліза	Мікрокомп'ютер Raspberry 4 Model B	Достатня продуктивність для дешифрації BLE та наявність мікрокомп'ютера у мене
Вибір базового програмного середовища	Android-емуляція застосунку або платформа Home Assistant	Навантаження на процесор, гнучкість, енергоефективність	Серверна платформа Home Assistant	Мінімальне споживання ресурсів та потужний рушій автоматизацій

Програмна інтеграція периферійного пристрою	Універсальний MQTT або кастомний модуль з GitHub	Двосторонній контроль, управління інвертором станції	Кастомний компонент з GitHub через HACS	Забезпечення повного контролю силових реле та перемикачів
Організація захищеного віддаленого доступу	Мережі ZeroTier, WireGuard, тунелі Mikrotik	Необхідність стороннього софту, хмарна незалежність	Нативний VPN-тунель за протоколом L2TP	Використання вбудованого в мобільні ОС клієнта без необхідності завантаження стороннього ПЗ
Захист автентифікаційних даних проєкту	Локальне запам'ятовування або менеджер паролів Bitwarden	Стійкість до перебору паролів, безпека, кросплатформеність.	Менеджер паролів Bitwarden	Архітектура нульового розголошення та захист від brute-force атак. Виключення забування пароля користувачем
Вибір каналу оперативного інформування	Mikrotik Dude, SMTP пошта, SMS-шлюз, Telegram бот	Швидкість доставки, легкість налаштування	Інтелектуальний чат-бот у месенджері Telegram	Миттєві push-сповіщення

Зм.	Лис	№ докум.	Підпис	Дата

PE22.461142.001 ПЗ

Організація живлення обчислювального вузла	Живлення від мережі 220 вольт або від DC-виходу станції	Коефіцієнт корисної дії, подвійне перетворення	Пряме підключення до DC-порту постійного струму	Максимальна енергоефективність шлюзу під час блекауту
Безпека користувачького інтерфейсу	Пряме виконання команд або захист кнопок підтвердженням	Ризик випадкового знеструмлення, людський фактор	Програмний захист із викликом діалогового вікна	Виключення ризику випадкового вимкнення DC-виходу

З ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОГРАМНО-АПАРАТНОГО ШЛЮЗУ МОНІТОРИНГУ

3.1 Розгортання та первинне налаштування операційного середовища Home Assistant на базі мікрокомп'ютера Raspberry Pi 4

Для забезпечення відмовостійкості, автономності та безперервності моніторингу елементів малої енергомережі як апаратну платформу було обрано мікрокомп'ютер Raspberry Pi. Центральним програмним забезпеченням системи виступає операційна система Home Assistant, що дозволяє локально обробляти дані, мінімізувати затримки та забезпечити високий рівень безпеки без обов'язкового використання сторонніх хмарних сервісів.

Процес підготовки апаратного забезпечення та розгортання операційної системи складається з кількох послідовних етапів: підготовки завантажувального носія, конфігурації мікропрограми Raspberry Pi, ініціалізації ОС та налаштування мережевих параметрів на рівні маршрутизатора.

3.1.1 Підготовка завантажувального носія

Для запису операційної системи на флеш-носій типу MicroSD використовується спеціалізоване утилітарне програмне забезпечення Raspberry Pi Imager (рис. 3.1.1). [2-5]

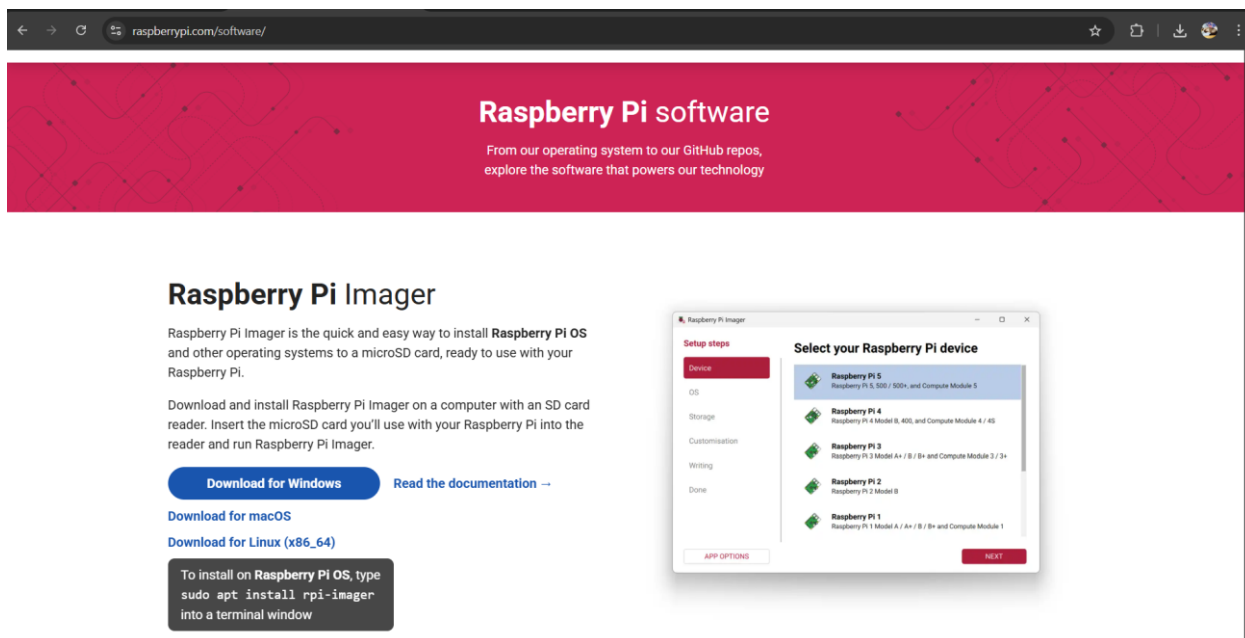


Рисунок 3.1.1 — Завантаження Raspberry Pi Imager.

Зм.	Лис	№ докум.	Підпис	Дата

РЕ22.461142.001 ПЗ

Лис
34

Ця утиліта дозволяє автоматично завантажувати актуальні та стабільні збірки ОС безпосередньо з офіційних репозиторіїв.

Послідовність дій при підготовці носія:

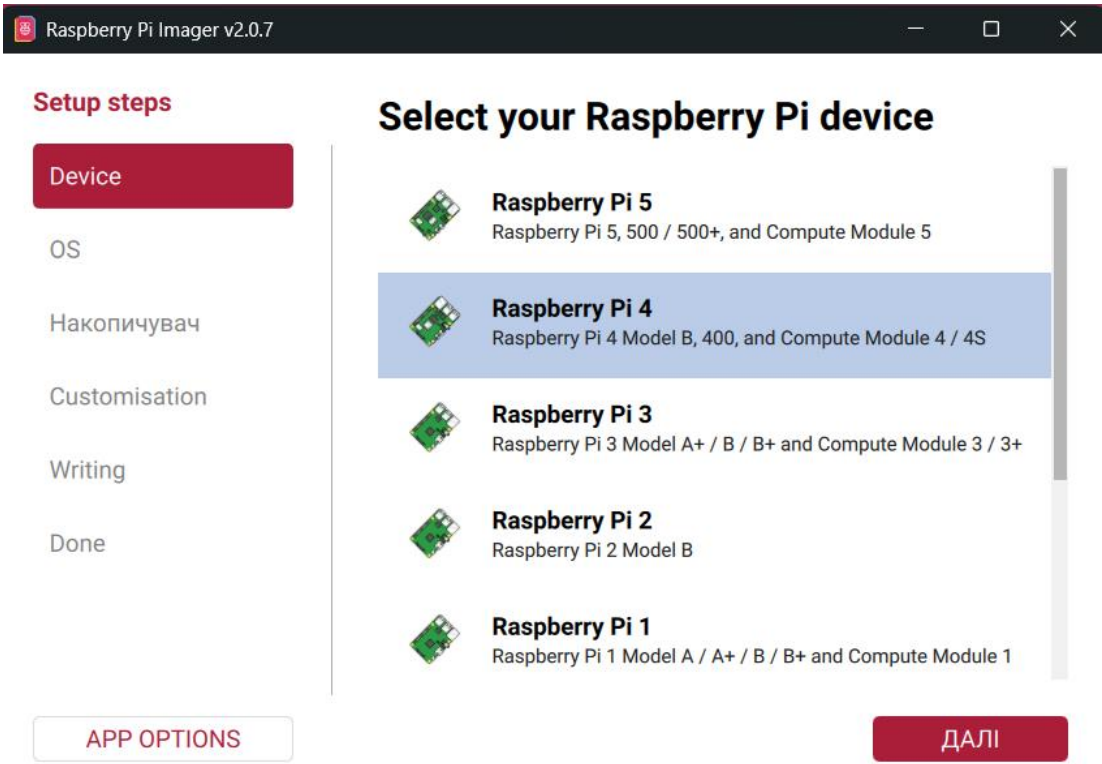


Рисунок 3.1.2 — Обираємо Raspberry Pi 4 .

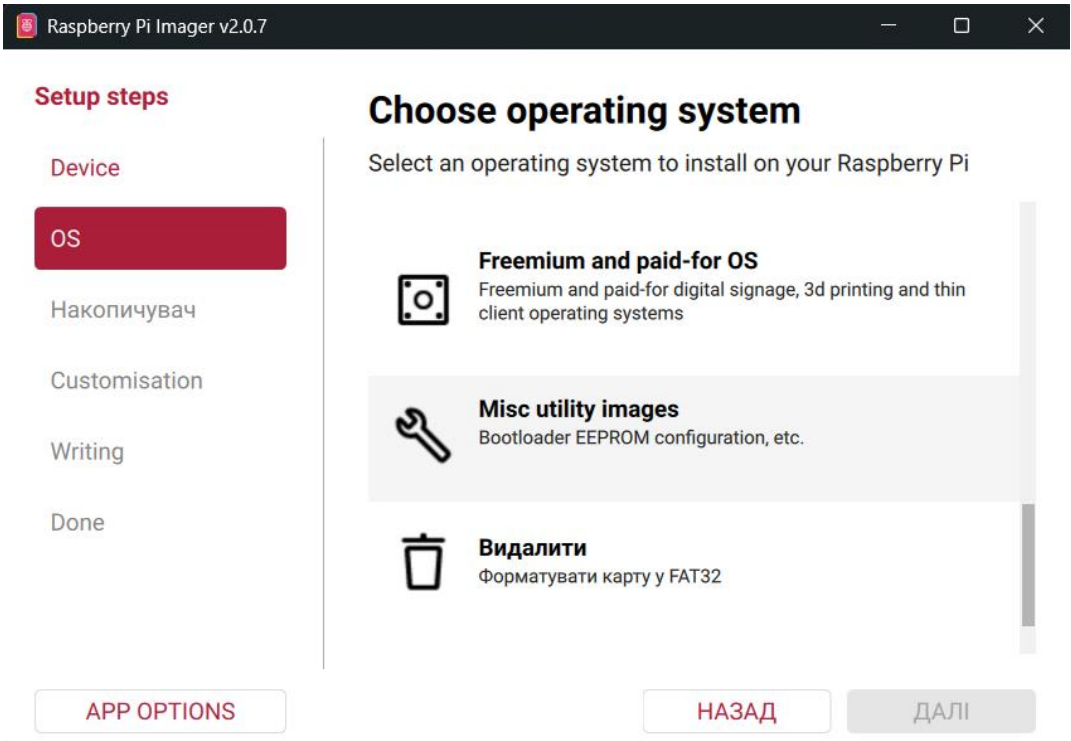


Рисунок 3.1.3 — Обираємо Misc utility imager.

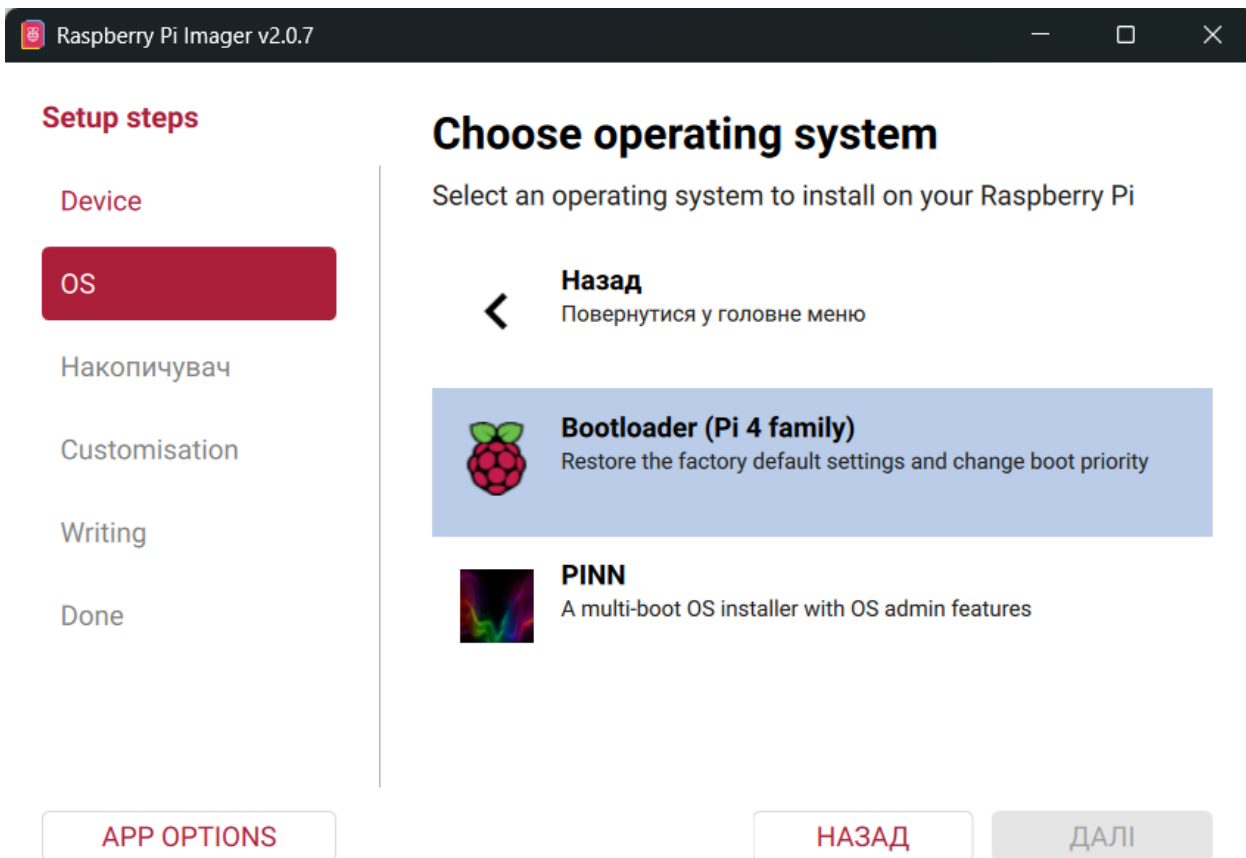


Рисунок 3.1.4 — Обираємо операційну систему.

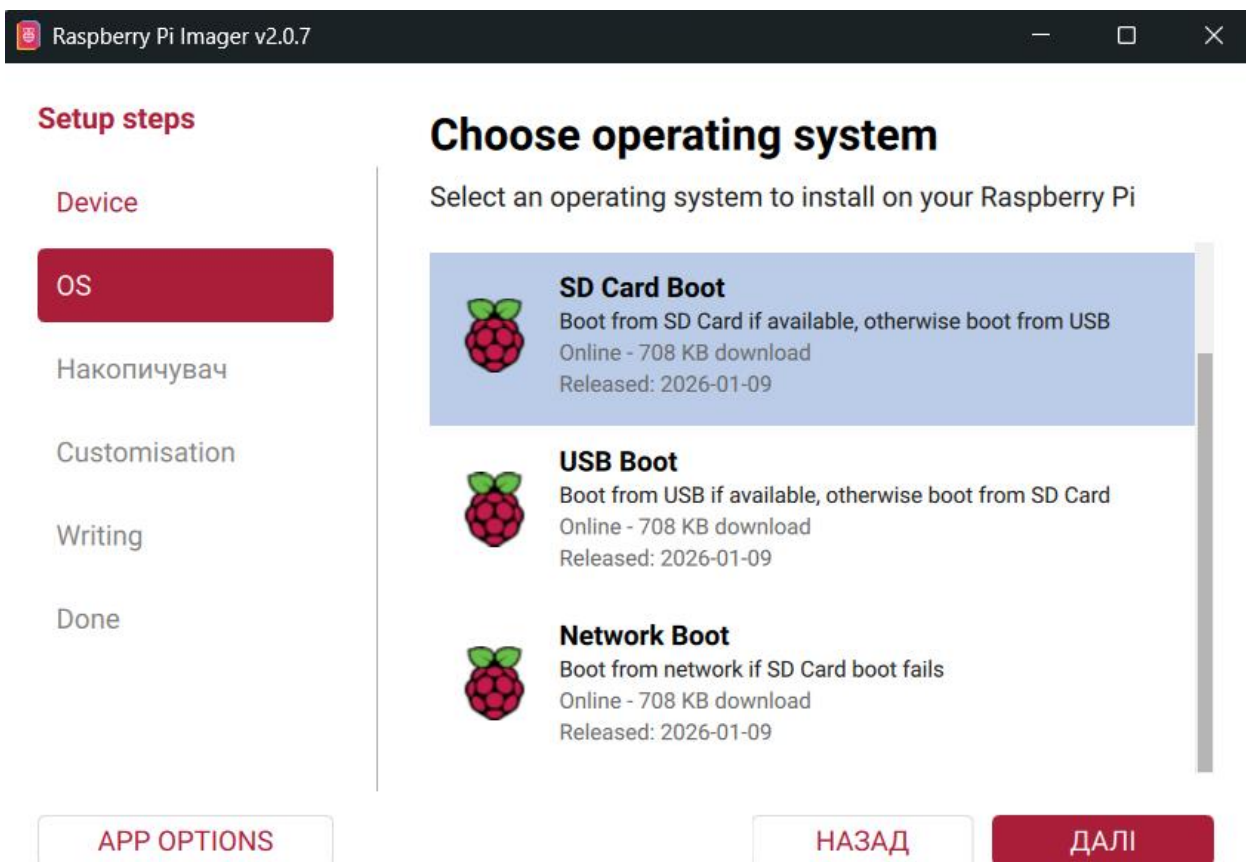


Рисунок 3.1.5 — Обираємо завантаження з SD-карти.

					РЕ22.461142.001 ПЗ	Лис
						36
Зм.	Лис	№ докум.	Підпис	Дата		

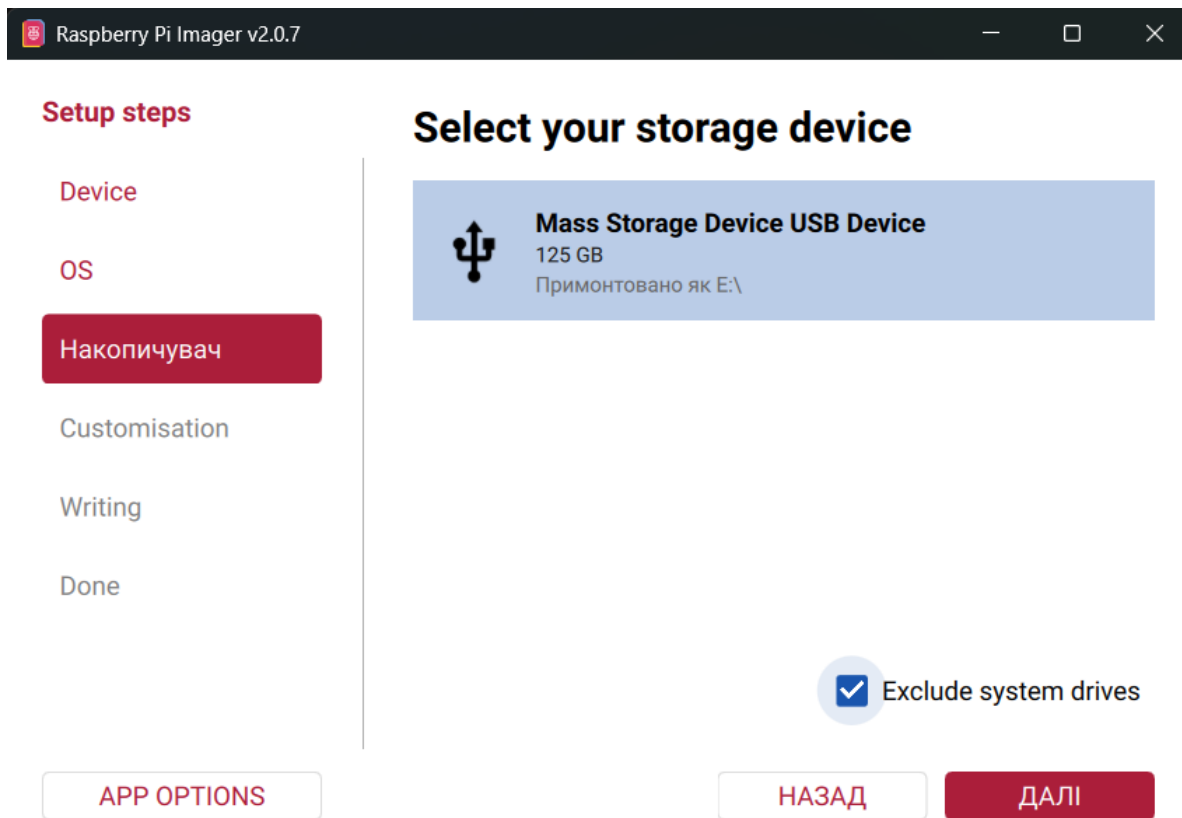


Рисунок 3.1.6 — Обираємо саму SD-карту.

Тут бажано обрати галочку «Exclude system drives» для виключення зі списку системних дисків. Це вбереже нас від випадкового перезаписування робочого диску комп'ютера

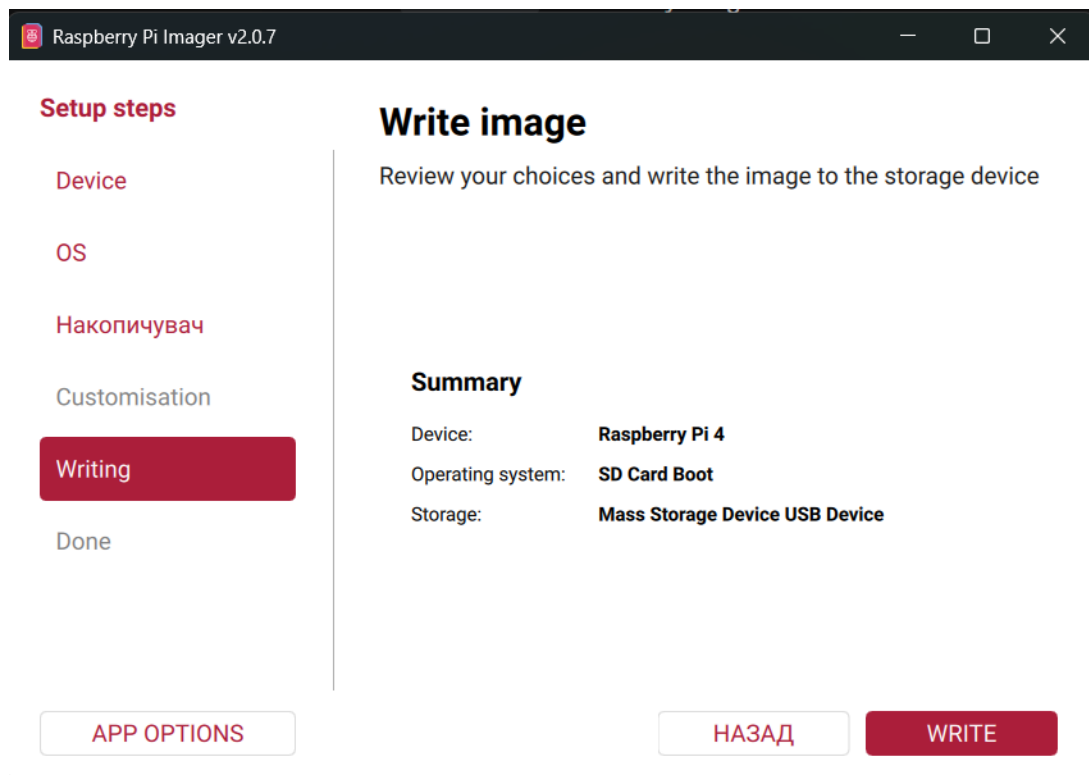


Рисунок 3.1.7 — Перевіряємо правильність нашого вибору.

					<i>PE22.461142.001 ПЗ</i>	Лис
						37
Зм.	Лис	№ докум.	Підпис	Дата		

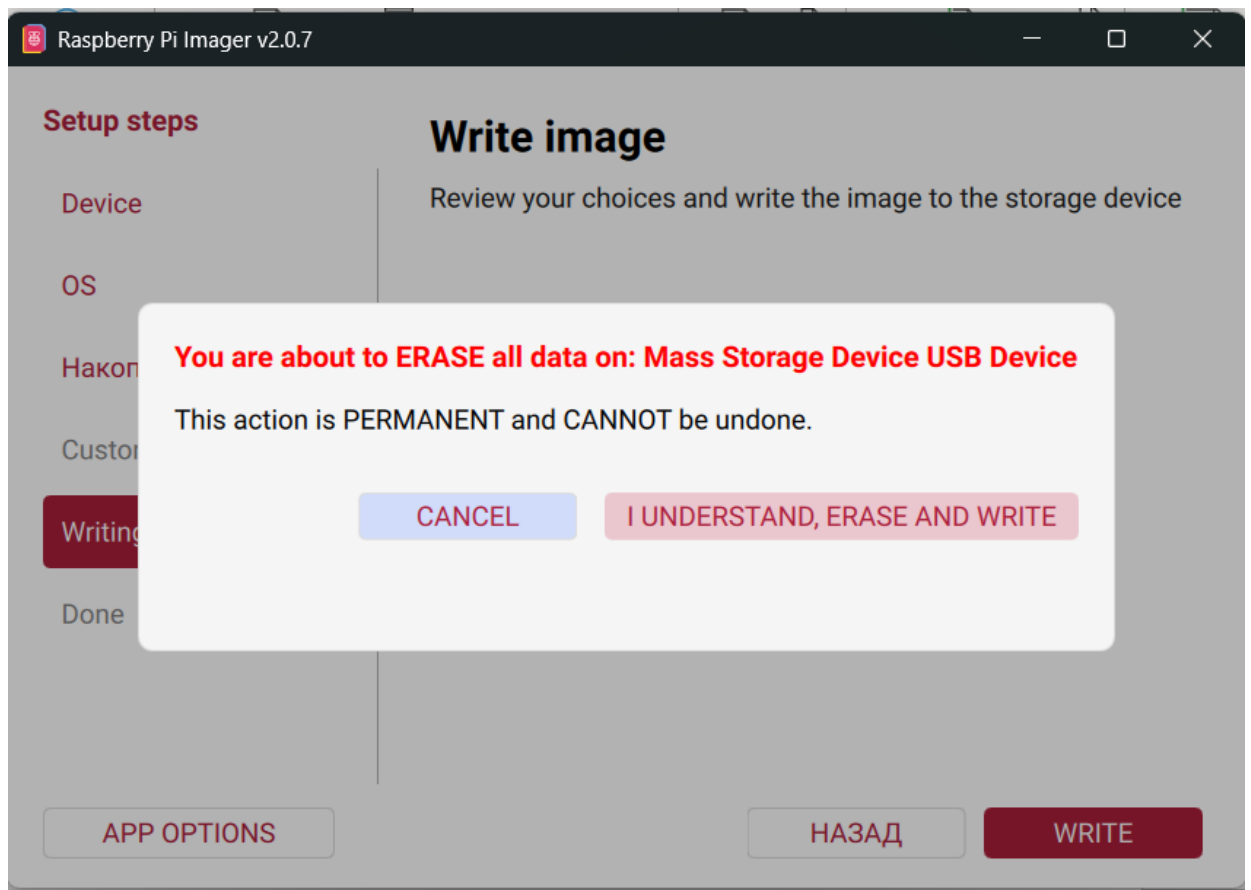


Рисунок 3.1.8 — Підтверджуємо зміни.

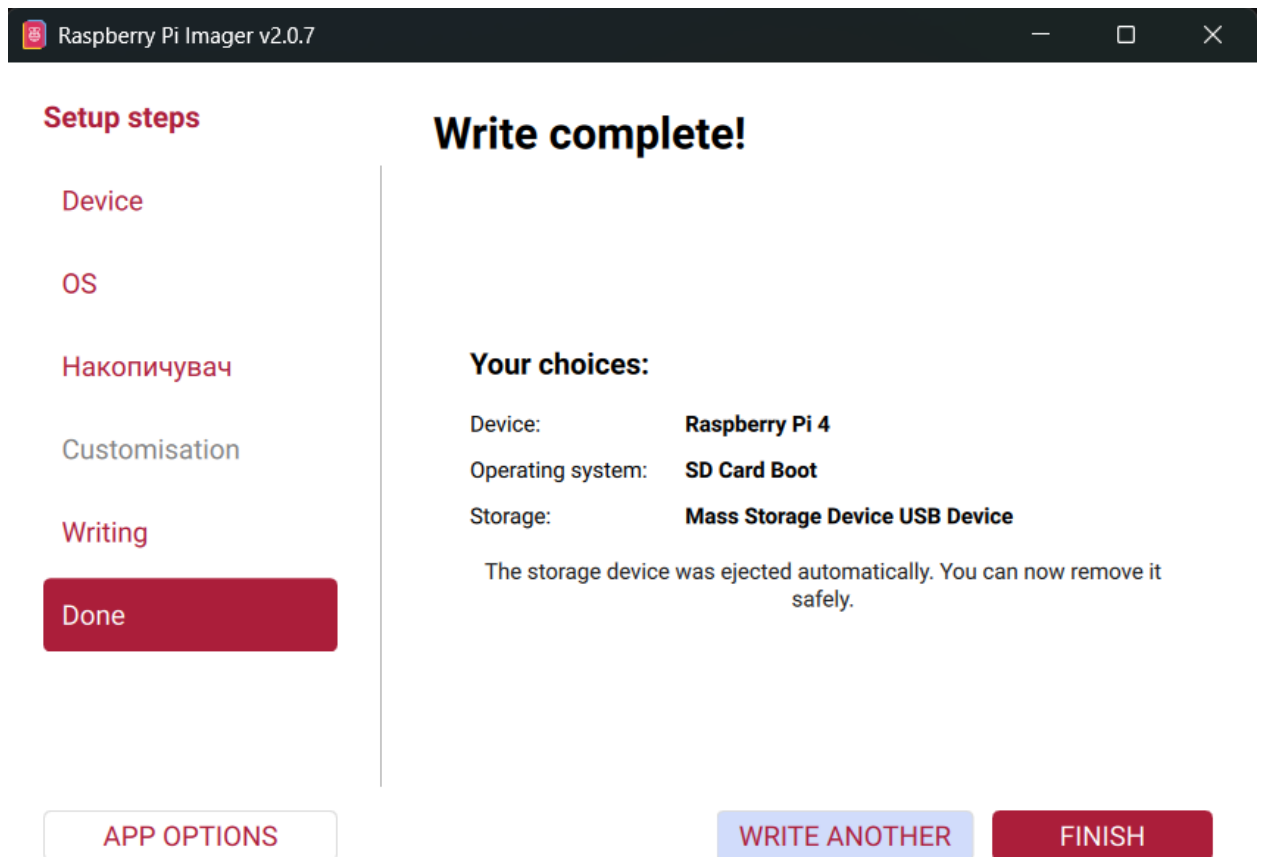


Рисунок 3.1.9 — Повідомлення про успішне форматування картки.

					РЕ22.461142.001 ПЗ	Лис
						38
Зм.	Лис	№ докум.	Підпис	Дата		

Після завершення запису утилітарного образу завантажувача накопичувач MicroSD вилучається з карт-рідера персонального комп'ютера та встановлюється у відповідний слот знеструмленого мікрокомп'ютера Raspberry Pi 4. Наступним кроком є подача напруги живлення на плату пристрою. Згідно з технічною документацією виробника, для забезпечення стабільної роботи Raspberry Pi 4 необхідне джерело живлення з вихідними параметрами 5 В та 3 А (загальна потужність 15 Вт).

Контроль процесу оновлення енергонезалежної пам'яті (EEPROM) здійснюється за допомогою апаратної світлодіодної індикації на платі мікрокомп'ютера за таким алгоритмом: червоний світлодіод, який відповідає за індикацію інтегрального живлення, повинен світитися безперервно, що свідчить про відсутність просідання напруги та достатню потужність джерела; зелений світлодіод, який сигналізує про активність системного завантажувача та реєстрацію помилок за допомогою серій спалахів, має спочатку зафіксувати постійне світіння, а потім перейти в режим циклічного блимання з однаковою частотою.

Стабільне та рівномірне блимання зеленого індикатора впродовж 30 секунд свідчить про успішне завершення процедури перепрошивки EEPROM. Після цього мікрокомп'ютер успішно ініціалізує нову конфігурацію, згідно з якою першочерговим пріоритетом завантаження визначено інтерфейс SD-карти. По завершенню вказаного інтервалу часу живлення плати вимикається, картка MicroSD повторно вилучається та підключається через карт-рідер до персонального комп'ютера для запуску утиліти Raspberry Pi Imager та запису основного образу операційної системи.

3.1.2 Занес операційної системи Home Assistant

Після успішного оновлення мікропрограми завантажувача (EEPROM) мікрокомп'ютера Raspberry Pi 4 виконується безпосередній процес підготовки основного робочого середовища системи моніторингу. Для цього накопичувач MicroSD повторно підключається до персонального комп'ютера адміністратора для запису образу операційної системи Home Assistant OS (HAOS).

					PE22.461142.001 ПЗ	Лис
						39
Зм.	Лис	№ докум.	Підпис	Дата		

Процедура флешування та первинного запуску реалізується за таким алгоритмом:

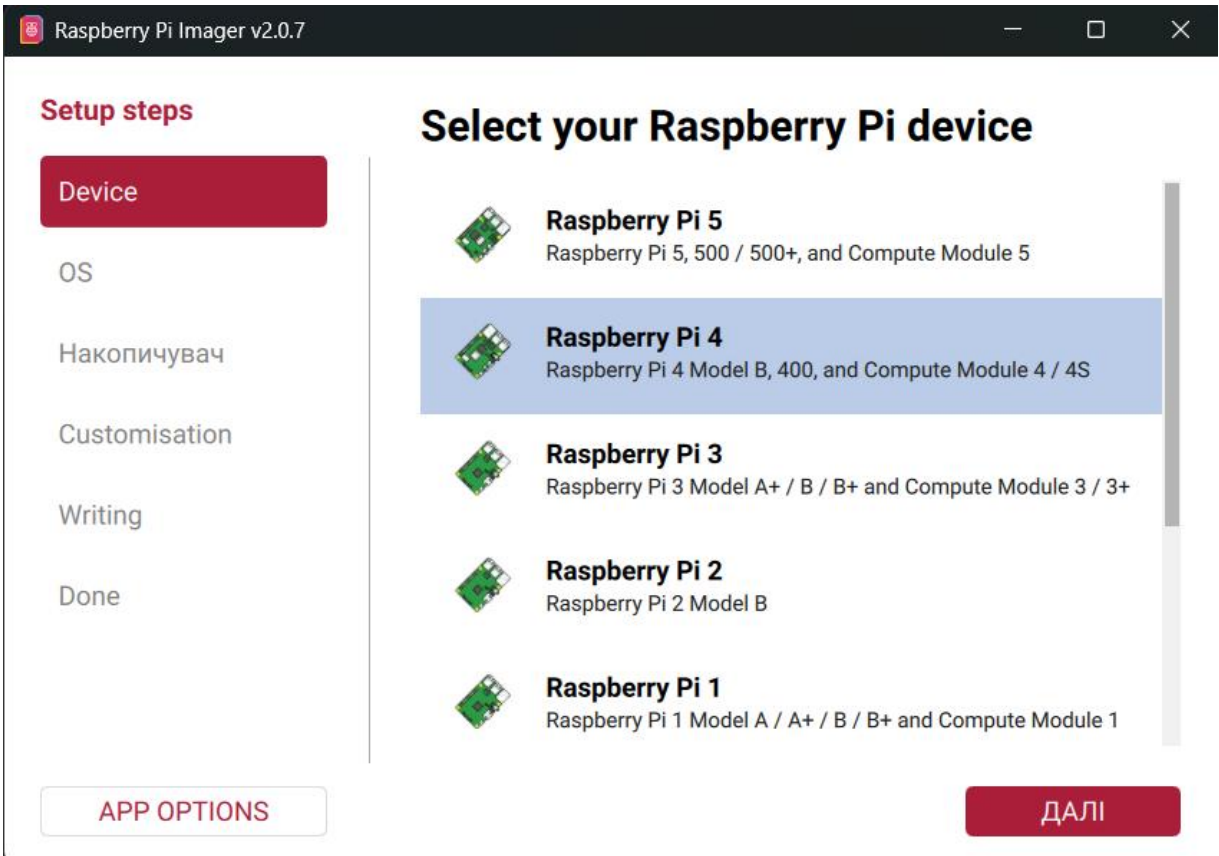


Рисунок 3.1.10 — Знову відкриваємо Imager, обираємо Raspberry Pi 4

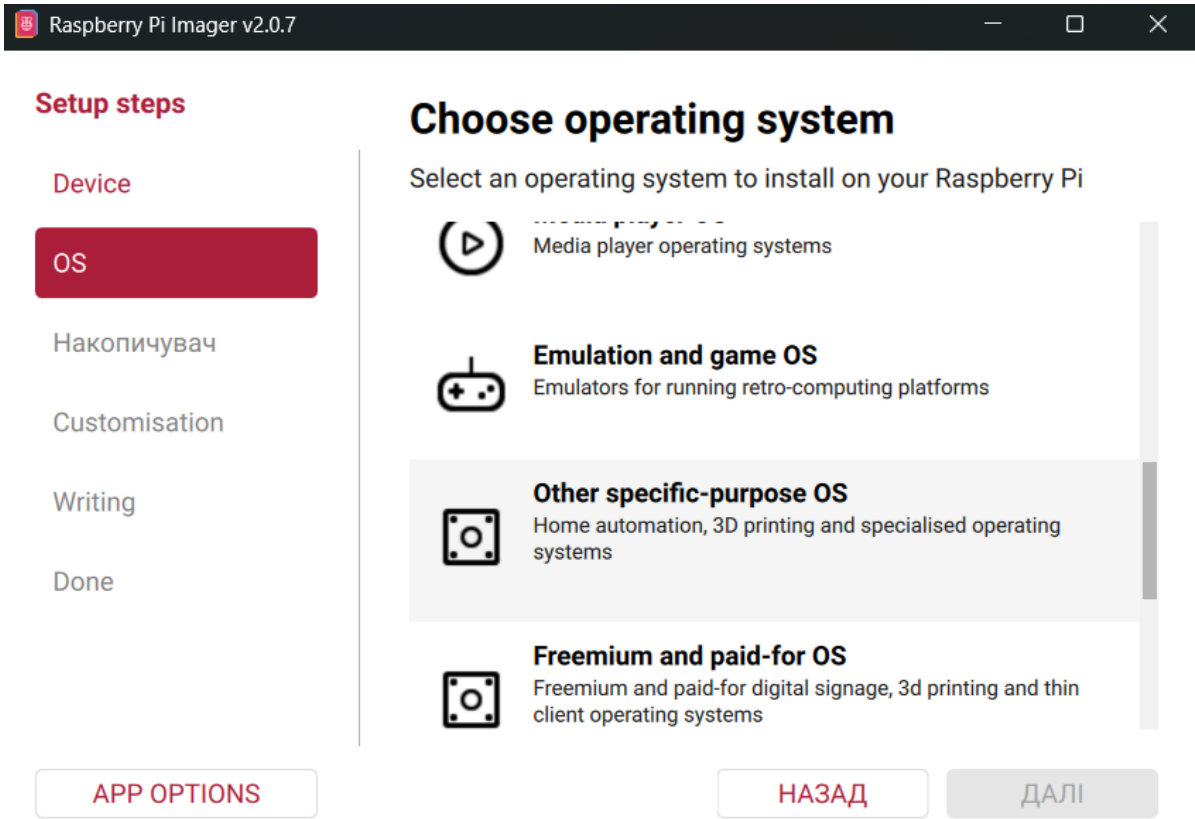


Рисунок 3.1.11 — Обираємо іншу специфічну операційну систему

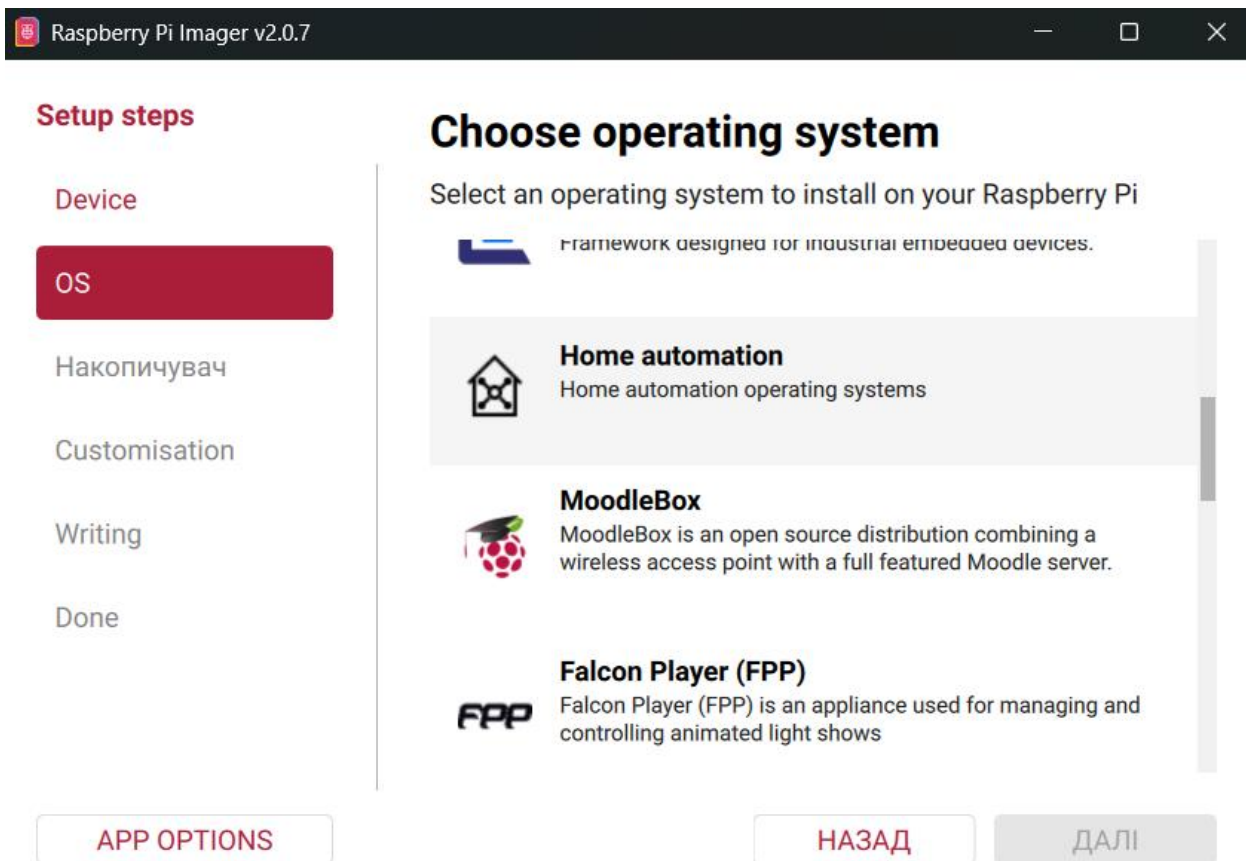


Рисунок 3.1.12 — Обираємо Home automation

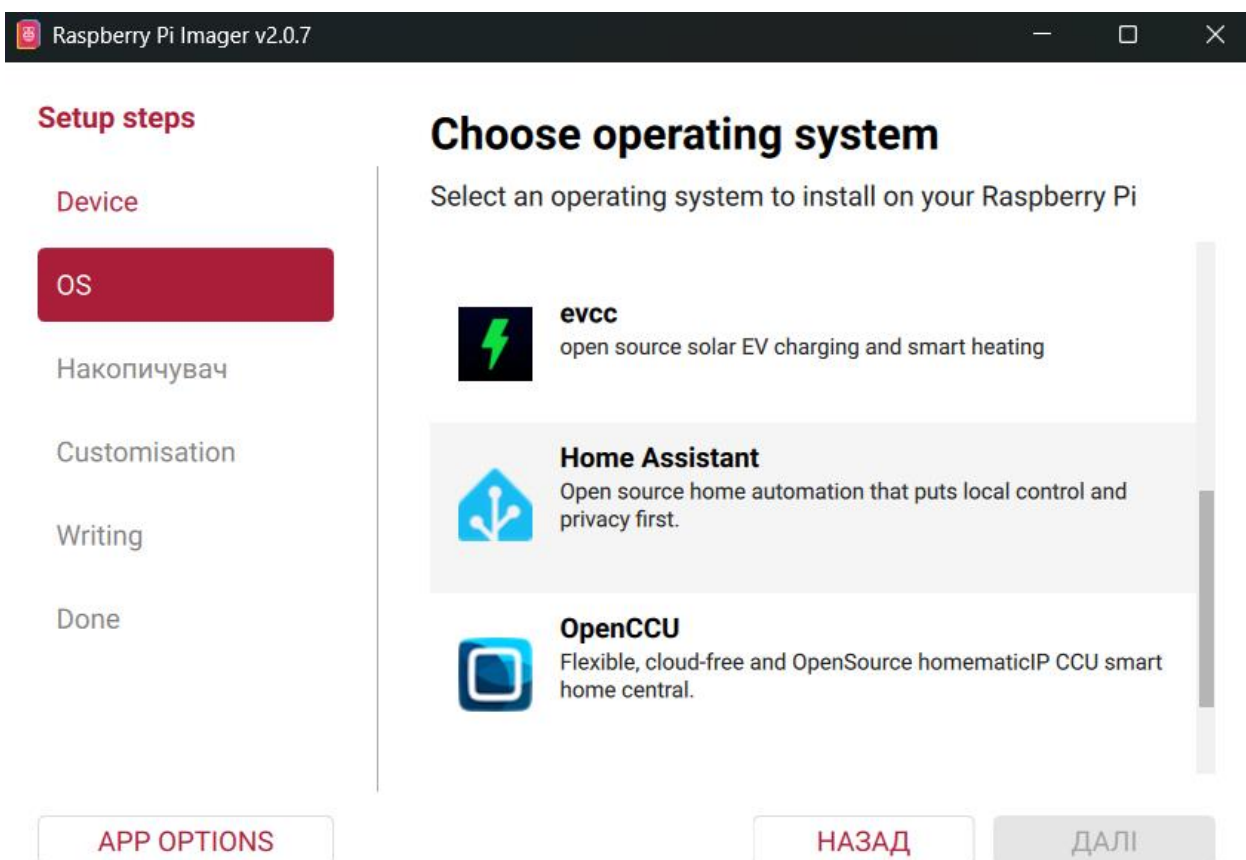


Рисунок 3.1.12 — Обираємо Home Assistant

					РЕ22.461142.001 ПЗ	Лис
						41
Зм.	Лис	№ докум.	Підпис	Дата		

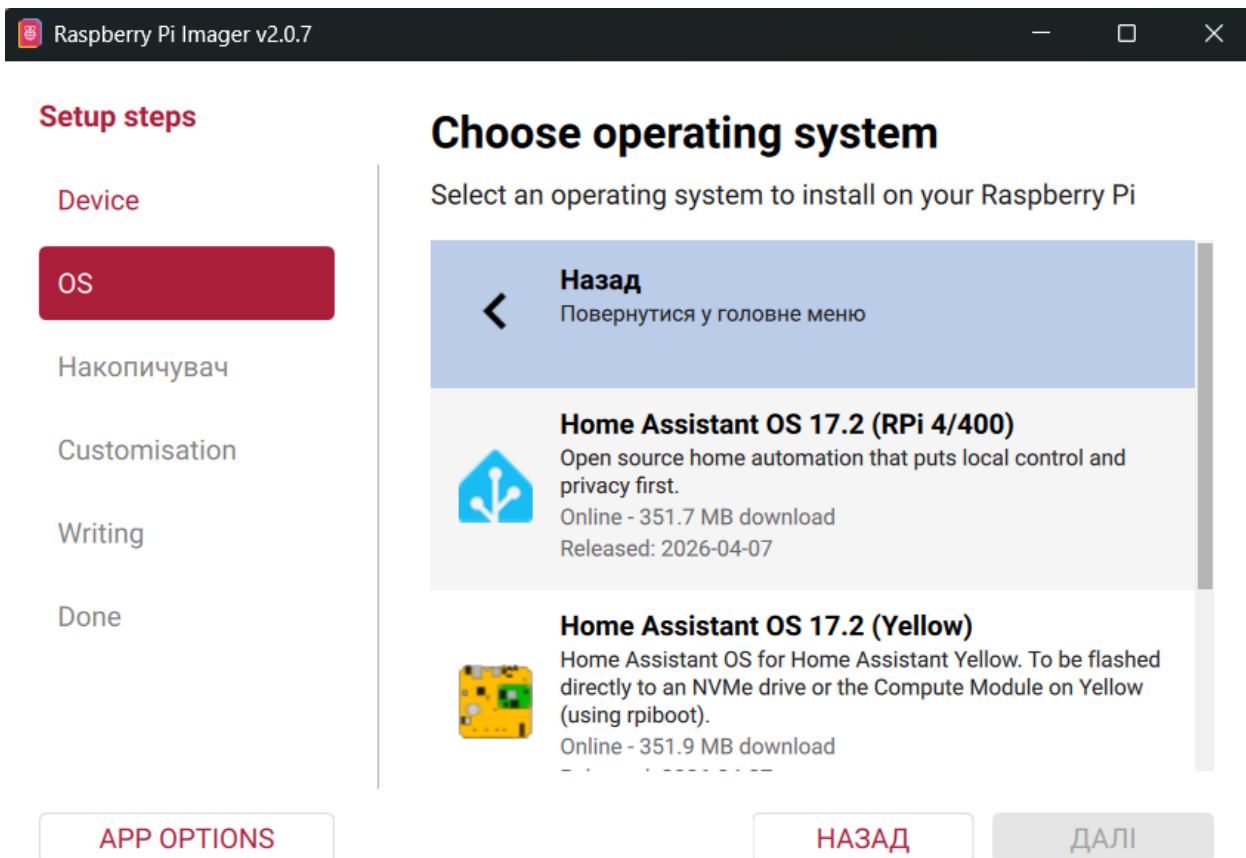


Рисунок 3.1.13 — Тут обираємо «синій» Home Assistant

Далі процедура така ж, як на рисунках 3.1.6 – 3.1.9.

На цьому етапі я зіткнувся з дуже цікавою проблемою. НА не запустився. Я довго чекав, але підключитись на нього так і не зміг. Довелось підключити монітор і клавіатуру. Проблема виявилась в тому, що Docker контейнери не запускались. Вони не встигали запускатись. Повільна швидкість флешки. Коли встановлюється зображення, в Raspberry Pi Imager показує швидкість завантаження (рис. 3.1.14) 16 MB/s. Перша SD-карта була 12 MB/s, на ній контейнери не встигали запускатись і ядро НА падало в помилку. На другій 16 MB/s все запустилось.[6-12]

					РЕ22.461142.001 ПЗ	Лис
						42
Зм.	Лис	№ докум.	Підпис	Дата		

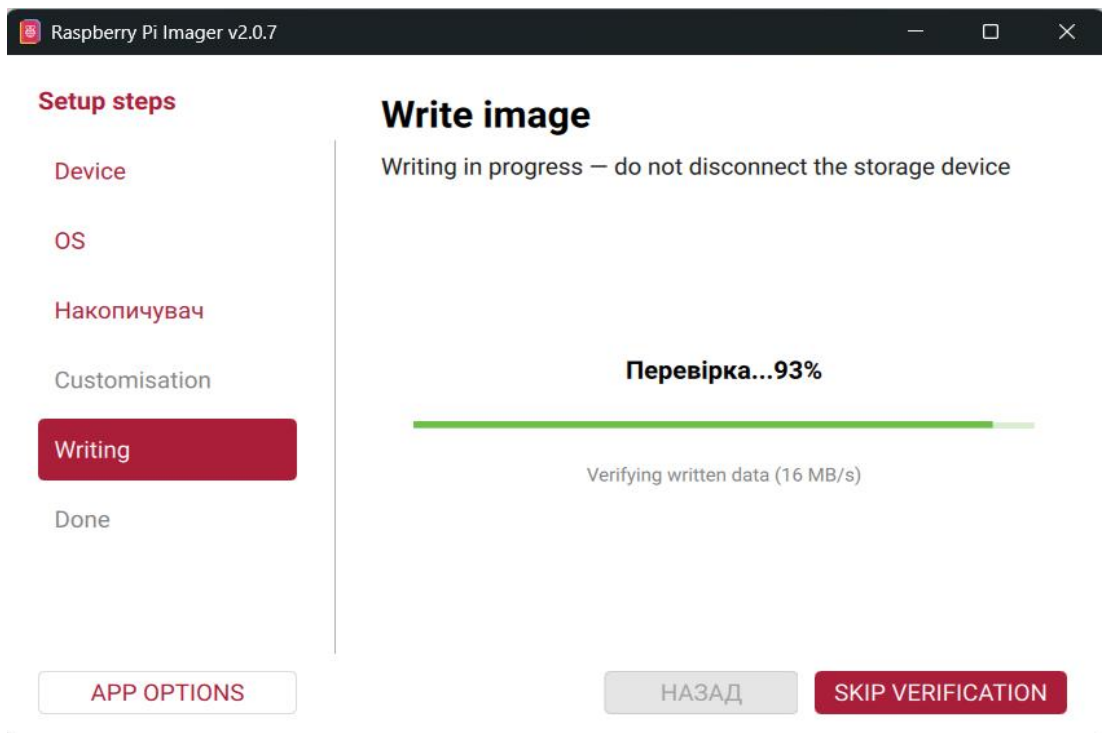


Рисунок 3.1.14 — Процес запису зображення

3.1.3 Конфігурація бездротового інтерфейсу мережі Wi-Fi

По завершенню процесу верифікації картка MicroSD вилучається з комп'ютера та встановлюється в апаратний слот мікрокомп'ютера. Для виключення потенційних завад у бездротовому середовищі та забезпечення гарантованого завантаження необхідних системних компонентів під час першого запуску, пристрій інтегрується в локальну мережу за допомогою дротового з'єднання. Патч-корд категорії Cat5e підключається до гігабітного мережевого інтерфейсу Ethernet (RJ-45) Raspberry Pi 4 та комутується у вільний LAN-порт маршрутизатора MikroTik.

Після підключення мережевого кабелю на плату подається номінальна напруга живлення (5 В, 3 А).

Спочатку треба зайти в НА. Для цього можна зайти на Mikrotik, та подивитись, яку айпі отримав НА, та одразу зробити її статичною(рис. 3.1.14)

					РЕ22.461142.001 ПЗ	Лис
						43
Зм.	Лис	№ докум.	Підпис	Дата		

DHCP Lease <192.168.0.183,192.168.0.183>

GeneralActive

Active Address:192.168.0.183

Active MAC Address:DC:A6:32:30:26:B5

Active Client ID:1:dc:a6:32:30:26:b5

Active Host Name:homeassistant

Active Class ID:

Active Agent Circuit Id:

Active Agent Remote Id:

Active Server:defconf

Bridge Port:wlan1

Src. MAC Address:

Expires After:00:05:37

Last Seen:00:04:23

Age:

Reconfigure Key:

Last Sent Counter:

Reconfigure Status:

OK

Cancel

Apply

Disable

Comment

Copy

Remove

Send Reconfigure

Ping

Check Status

Рисунок 3.1.14 — Статична IP на НА

Праворуч має бути кнопка «Make static». Після натискання, побачите те, що на рисунку 3.1.14.

Тепер з браузера можна зайти за айпі адресою (рис. 3.1.15). При першому вході НА запропонує створити користувача. Створюємо. Пароль раджу взяти з Bitwarden. Хоча б 14 символів з використанням букв, цифр та спеціальних знаків.

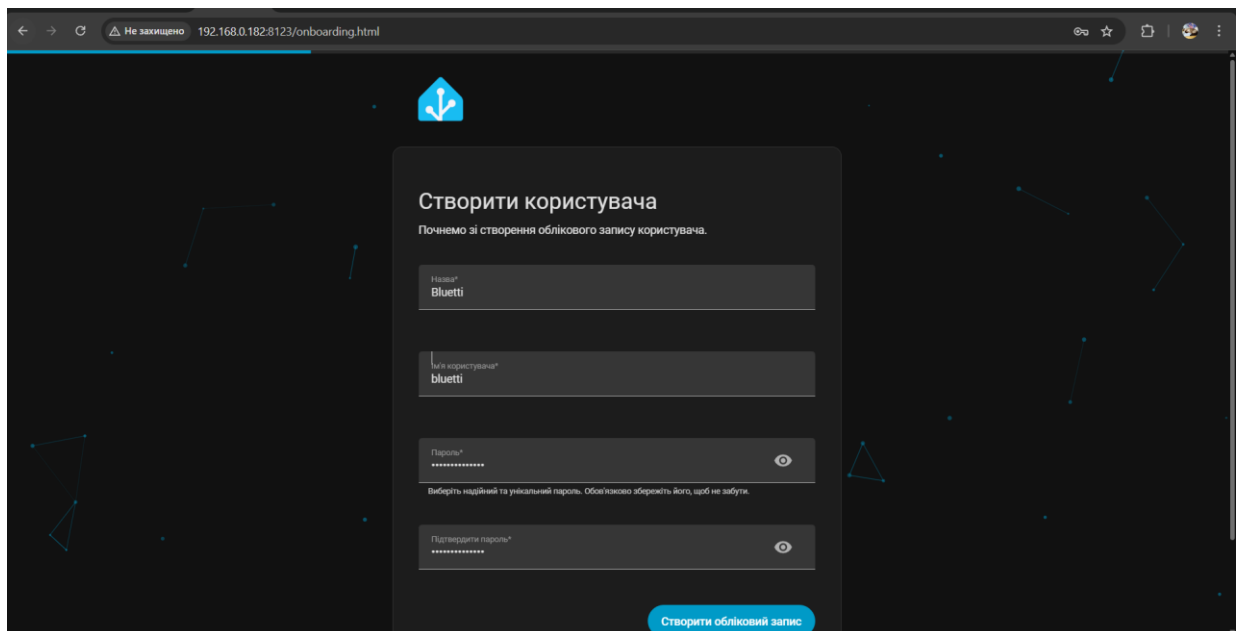


Рисунок 3.1.15 — Створення акаунта НА.

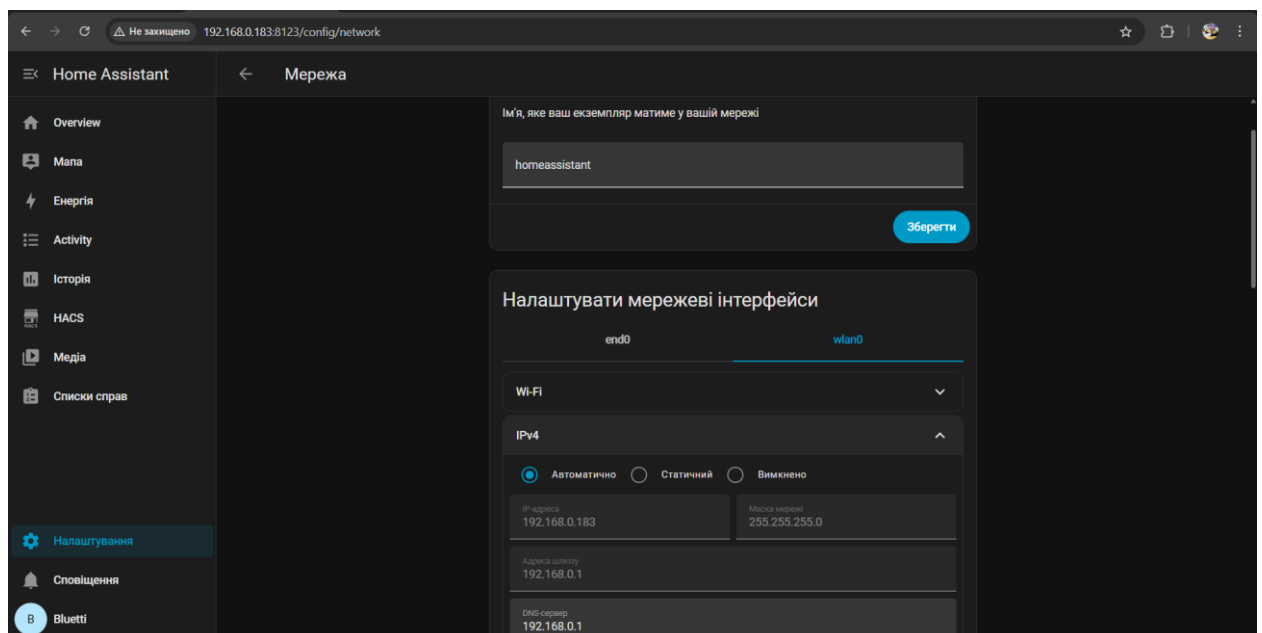


Рисунок 3.1.16 — Налаштування Wi-Fi модуля

3.2 Інтеграція джерел безперервного живлення

Для реалізації моніторингу та керування параметрами портативної зарядної станції необхідно забезпечити локальний канал зв'язку між мікрокомп'ютером Raspberry Pi та контролером обладнання Bluetti. Оскільки базове ядро операційної системи Home Assistant не підтримує роботу з цим типом обладнання «з коробки», першочерговим завданням є розширення можливостей сервера шляхом інтеграції альтернативного каталогу додатків HACS.[13][14]

					РЕ22.461142.001 ПЗ	Лис
						45
Зм.	Лис	№ докум.	Підпис	Дата		

Весь процес розгортання та зв'язування компонентів виконується послідовно через графічний інтерфейс системи.

На початковому етапі, після завантаження базових інсталяційних скриптів HACS у внутрішню файлову систему, необхідно виконати обов'язковий повний перезапуск усього сервера Home Assistant. Це здійснюється через головне меню налаштувань системи шляхом натискання кнопки «Перезапустити Home Assistant» (рис. 3.2.1).

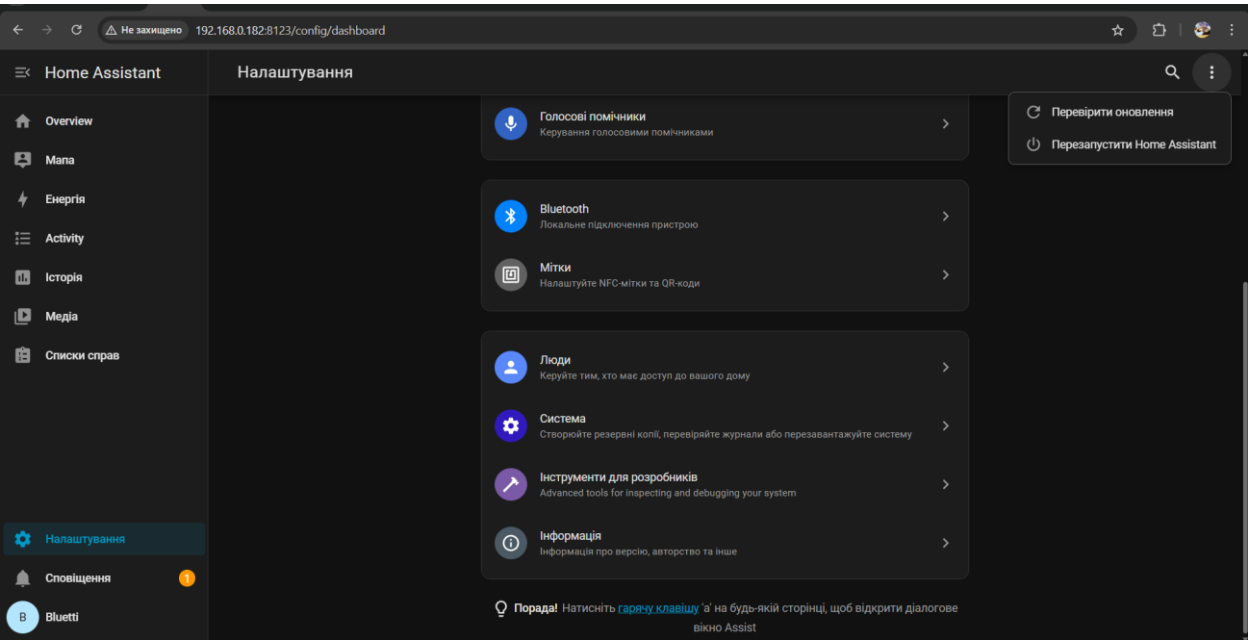


Рисунок 3.2.1 — Перезапуск НА

Цей крок є критично важливим для того, щоб операційна система оновила свої залежності, очистила тимчасовий кеш та ініціалізувала завантажені програмні бібліотеки в оперативній пам'яті мікрокомп'ютера.

Після успішного завершення циклу перезавантаження сервера, повертаємось до розділу конфігурації, обираємо пункт «Пристрої та сервіси» та ініціюємо додавання нової інтеграції. У пошуковому рядку вікна вибору брендів вводиться назва каталогу HACS (рис. 3.2.2).

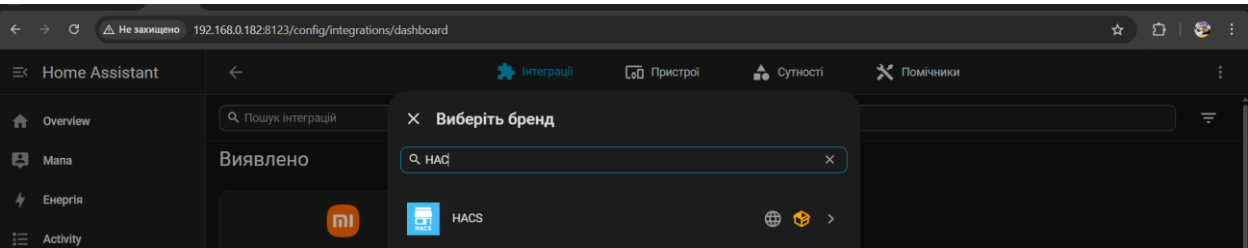


Рисунок 3.2.2 — Інтеграція HACS

					РЕ22.461142.001 ПЗ	Лис
						46
Зм.	Лис	№ докум.	Підпис	Дата		

Для того щоб браузер коректно відобразив новий компонент і не використовував застарілі збережені елементи інтерфейсу, на цьому етапі рекомендується виконати примусове оновлення кешу сторінки за допомогою комбінацій клавіш Ctrl+F5.

Наступним кроком є проходження процедури ліцензійного ознайомлення, де користувач підтверджує згоду з чотирма основними положеннями роботи кастомного магазину додатка. Після цього відкривається вікно авторизації, яке вимагає зв'язування локального сервера з глобальною платформою для розробників GitHub[15-17]. Це необхідно для того, щоб каталог HACS мав офіційний доступ до віддалених репозиторіїв додатків, зокрема до офіційного сховища доповнень HACS Addons, та міг безперешкодно викачувати необхідні кодові бази і стежити за їхніми оновленнями. Авторизація реалізована за захищеним протоколом безпеки: Home Assistant видає одноразовий код, який адміністратор вводить на сторінці активації пристроїв у своєму профілі GitHub.

Після підтвердження прав доступу з боку GitHub[18], у лівій робочій панелі Home Assistant з'являється повноцінний окремий розділ HACS. Перейшовши в цей каталог, виконуємо внутрішній пошук серед доступних світових репозиторіїв інтеграцій. Вводимо запит для пошуку спеціалізованого модуля «Bluetti BT»(рис. 3.2.3).

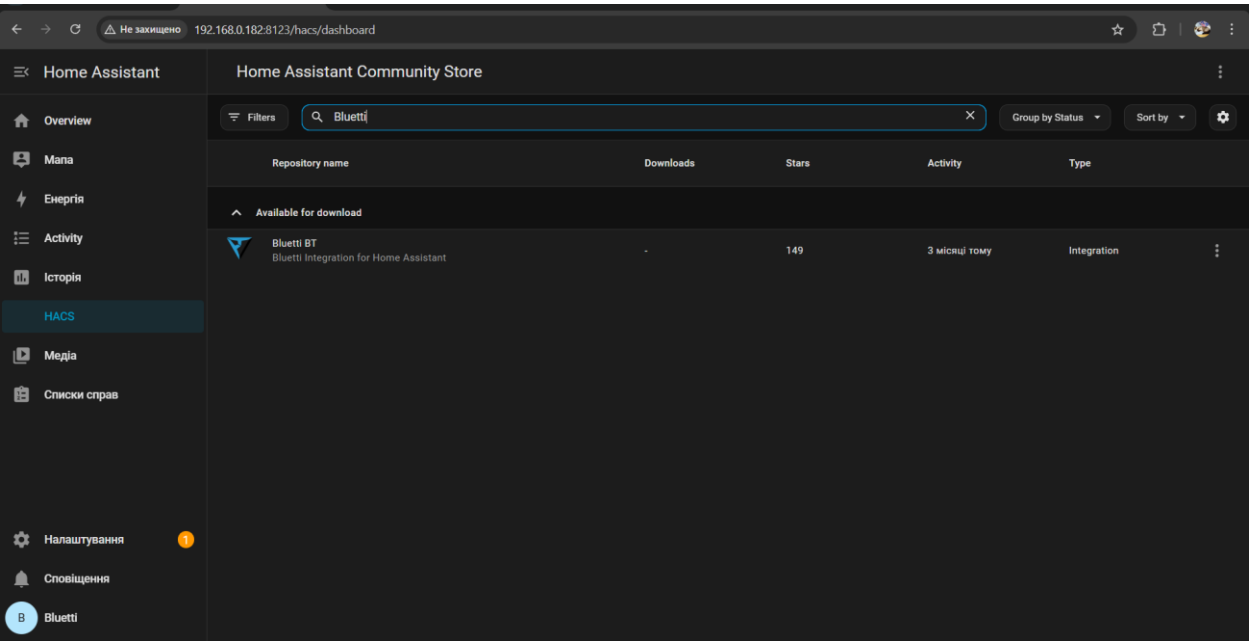


Рисунок 3.2.3 — Інтеграція Bluetti BT

Цей кастомний драйвер призначений для прямої взаємодії з зарядною станцією та завантажується безпосередньо з інтегрованих сховищ розробників.

Після завершення скачування пакета «Bluetti BT» виконується фінальний перезапуск сервера для остаточного впровадження драйвера в операційну систему. Принцип роботи даного компонента базується на використанні вбудованого апаратного модуля Bluetooth Low Energy мікрокомп'ютера Raspberry Pi. Модуль автоматично сканує бездротовий ефір, виявляє унікальний Bluetooth-передавач зарядної станції(рис. 3.2.4).

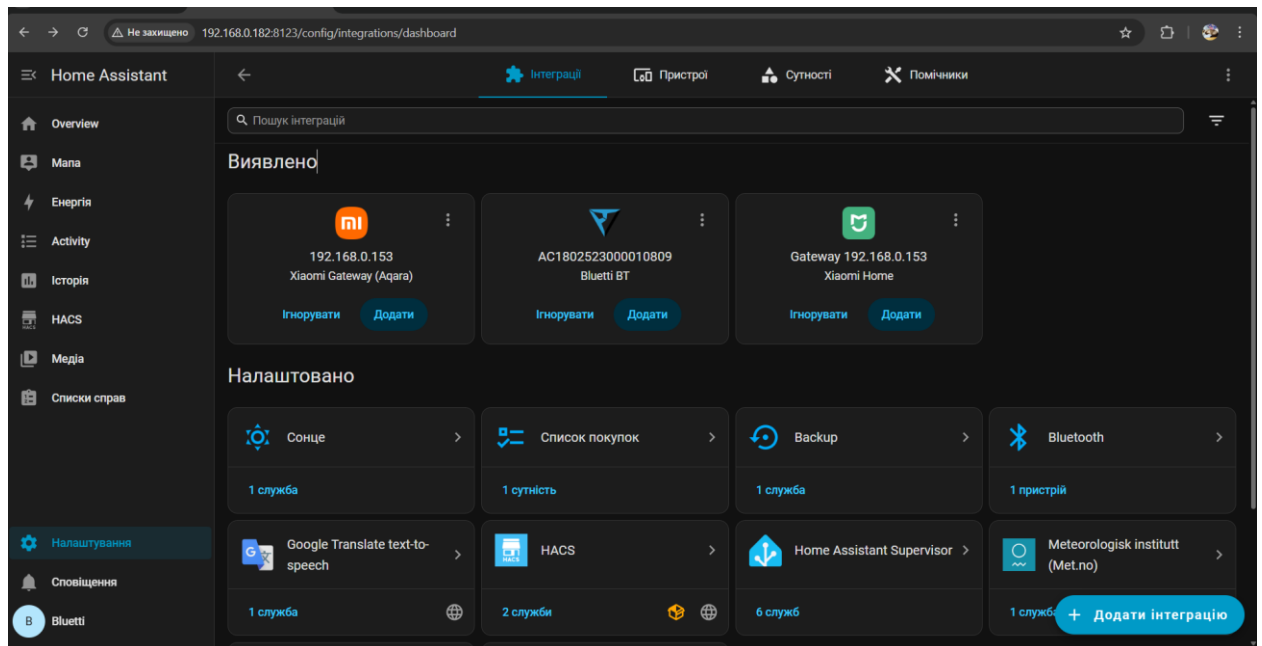


Рисунок 3.2.4 — Успішно з'явилась станція. Натискаємо «Додати»

Головною перевагою такого підходу є повна незалежність системи від наявності зовнішнього інтернету чи працездатності хмарних серверів виробника. Як тільки бездротовий зв'язок по Bluetooth успішно стабілізується, в інтерфейсі Home Assistant автоматично генерується повний набір сутностей для даного пристрою(рис. 3.2.5).

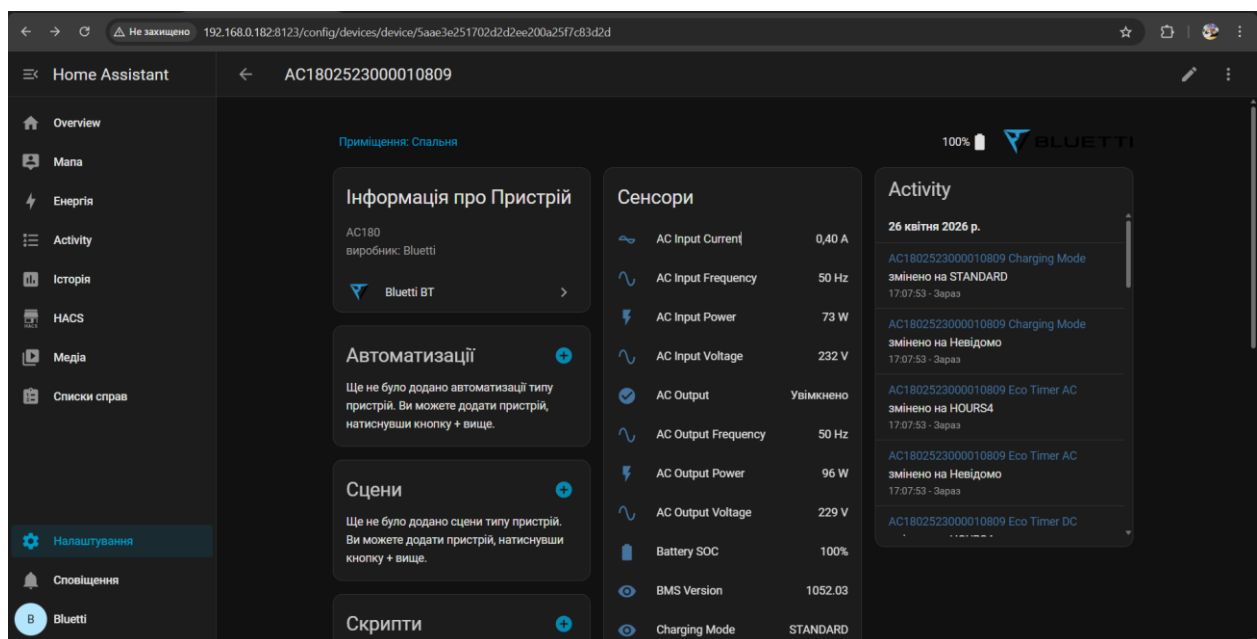


Рисунок 3.2.5 — Всі наявні сенсори і перемикачі станції

Користувач отримує інформаційні віртуальні сенсори, наприклад, рівень заряду акумулятора у відсотках, вхідна потужність сонячних панелей, поточне навантаження, та виконавчі перемикачі, які дозволяють дистанційно керувати інвертором станції, вмикати або вимикати виходи змінного (AC) та постійного (DC) струму, забезпечуючи надійну основу для подальшої автоматизації системи безпеки та енергоживлення.

Переглянуто було дуже багато репозиторіїв Github, в пошуках того самого, де можна буде керувати станцією. Бібліотека від Patrick762 [20] є основою. Вона добре допомагає підключити Bluetti до HA, але в ній не можна керувати шифрованими станціями. Я переглянув багато форків (так називається бібліотека, коли хтось бере за основу бібліотеку від Patric762 і міняє буквально пару рядків коду) [21-34]. І зупинився на форку від atiweb [34]

3.3 Розгортання захищеного каналу віддаленого керування на базі технології L2TP VPN

Для забезпечення надійного та безпечного віддаленого доступу до системи моніторингу енергоживлення з-за меж локальної мережі є неприпустимим пряме відкриття портів сервера Home Assistant (Port Forwarding) у зовнішню мережу Інтернет. Такий підхід створює критичні вразливості та відкри-

					PE22.461142.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		49

ває потенційному зловмиснику вектори для проведення кібератак на інтегральні компоненти системи. З метою захисту телеметрії та виконавчих команд керування, у даному проекті реалізовано архітектуру захищеного тунелювання на базі маршрутизатора MikroTik під керуванням операційної системи RouterOS з використанням комбінованого протоколу L2TP у поєднанні з набором протоколів шифрування IPsec.[35-44]

Процес налаштування безпечного віддаленого доступу реалізується безпосередньо на мережевому шлюзі об'єкта. Спочатку в утиліті WinBox виконується конфігурація локального пулу IP-адрес, які будуть динамічно виділятися віддаленим клієнтам (наприклад, мобільному телефону адміністратора) при встановленні захищеного з'єднання. Тут я трошки схитрив. У мене DHCP який видає IP всім девайсам має пул 192.168.0.150-192.168.0.254. Для користувача L2TP при під'єднанні видається 192.168.0.101. Відповідно, користувач може звертатись до пристроїв в мережі, але його IP жоден пристрій не займе. Наступним кроком є активація та налаштування самого L2TP-сервера. У параметрах сервера в обов'язковому порядку вмикається обов'язкове використання протоколу IPsec (Use IPsec: yes) та задається стійкий спільний ключ (IPsec Secret), який виступає першим рівнем автентифікації та захищає тунель від спроб перехоплення трафіку на етапі ініціалізації сесії.

Для детальної ідентифікації користувача у розділі PPP Secrets створюється унікальний обліковий запис віддаленого адміністратора, де прописується складний пароль, визначається тип сервісу (l2tp) та фіксуються статичні локальна та віддалена IP-адреси всередині тунелю. Це дозволяє чітко розмежувати права доступу та забезпечити стабільну маршрутизацію пакетів (рис. 3.3.1).

					РЕ22.461142.001 ПЗ	Лис
						50
Зм.	Лис	№ докум.	Підпис	Дата		

PPP Secret <Mishka-iPhone>

Name: Mishka-iPhone

Password: dcs^%dfv21sc0*fv

Service: l2tp

Caller ID:

Profile: profile1

Local Address: 192.168.0.100

Remote Address: 192.168.0.101

Remote IPv6 Prefix:

Routes:

IPv6 Routes:

Limit Bytes In:

Limit Bytes Out:

Last Logged Out: Jun/08/2026 11:35:46

Last Caller ID: 5.248.184.77

Last Disconnect Reason: hung up

enabled

OK

Cancel

Apply

Disable

Comment

Copy

Remove

Рисунок 3.3.1 — Інтерфейс утиліти WinBox на етапі конфігурації параметрів L2TP сервера та облікових записів

PPP

Interface PPPoE Servers OVPN Servers Secrets Profiles Active Connections L2TP Ether

+ - ✓ ✗ 📁 🗑️ PPP Scanner PPTP Server SSTP Server L2TP Server

	Name	Type	Actual MTU	L2 MTU	Tx
R	«» Mishka-lphone	L2TP Server Binding	1400		92

Interface <Mishka-lphone>

General Status Traffic

Name: Mishka-lphone

Type: L2TP Server Binding

Actual MTU: 1400

VRF: main

User: Mishka-iPhone

OK

Cancel

Apply

Disable

Comment

Copy

Рисунок 3.3.2 — Створення статичного інтерфейсу L2TP

Зм.	Лис	№ докум.	Підпис	Дата

PE22.461142.001 ПЗ

Лис
51

Також, важливим пунктом є створення статичного інтерфейсу (рис. 3.3.2). Це полегшить налаштування Firewall. Без цього, інтерфейс створюється лише коли користувач підключився. Відповідно, коли користувач відключається, інтерфейс пропадає, що в свою чергу ламає Firewall, і унеможливорює повторне підключення.

Важливим етапом побудови захищеної мережевої інфраструктури є адаптація політики міжмережевого екрана. Оскільки маршрутизатор MikroTik за умовчанням блокує всі неініційовані вхідні підключення із зовнішнього інтерфейсу (WAN), у таблиці Firewall Rules створюються спеціальні дозволяючі правила для пропуску VPN-трафіку. Оскільки протокол L2TP сумісно з IPsec використовує для встановлення зв'язку та шифрування пакети типу UDP, міжмережевий екран налаштовується на відкриття портів 500, для обміну ключами ISAKMP, 4500 для NAT-Traversal та 1701 безпосередньо для трафіку L2TP. При цьому всі інші спроби несанкціонованого сканування або підбору паролів з боку зовнішньої мережі жорстко відсікаються базовими правилами фільтрації (рис 3.3.3).

Firewall											
Filter Rules											
NAT Mangle Raw Service Ports Connections Address Lists Layer7 Protocols											
+ - [icon] [icon] [icon] [icon] [icon] [icon] [icon] [icon] [icon] [icon] [icon] [icon]											
Reset Counters Reset All Counters											
#	Action	Chain	Src. Address	Dst. Address	Protocol	Dst. Port	In. Interface	In. Interf...	Bytes	Packets	Comment
0 D	passthrough	forward							86.3 GiB	99 830 644	special dummy rul...
1	fasttrack connection	forward							29.7 GiB	32 194 332	Fasttrack
2	accept	forward							29.7 GiB	32 194 332	Accept establishe...
3	drop	forward							2470.1 KiB	41 309	Drop invalid
4	accept	forward		192.168.0.183			Mishka-Iphone		374 KiB	687	L2TP to All network
5	drop	forward						WAN	0 B	0	defconf: drop all fr...
6	accept	input							51.5 MiB	241 843	Accept establishe...
7	drop	input							1317.6 KiB	6 395	Drop invalid
8	accept	input			6 (tcp)	8291		LAN	120 B	3	Winbox LAN
9	accept	input						LAN	19.5 MiB	169 777	LAN input allow
10 X	accept	input			6 (tcp)	8291		WAN	0 B	0	
11	accept	input	192.168.0.101		6 (tcp)	8291	Mishka-Iphone		104 B	2	Allow from L2TP
12	accept	input			50 (ipsec-esp)			WAN	0 B	0	L2TP-ipsec
13	accept	input			17 (udp)	500,4500,1701		WAN	26.8 KiB	158	L2TP-ipsec
14	drop	input							11.9 MiB	218 617	Drop input

Рисунок 3.3.3 — Конфігурація дозволяючих правил фільтрації міжмережевого екрана в операційній системі RouterOS

Кінцевим етапом реалізації криптозахищеного каналу є налаштування клієнтського терміналу на мобільному пристрої адміністратора під керуванням операційної системи Android або iOS. У системних налаштуваннях мережі створюється новий профіль VPN типу «L2TP». У відповідні поля конфігурації вноситься зовнішня статична IP-адреса маршрутизатора MikroTik, загальний

ключ шифрування IPsec, а також логін і пароль створеного раніше PPP-користувача.

Після активації тунелю мобільний телефон встановлює шифроване з'єднання з маршрутизатором, проходячи двоетапну перевірку автентичності, та отримує внутрішню IP-адресу. Завдяки цьому користувач отримує повноцінний безпечний доступ до локальних ресурсів мережі, зокрема до веб-панелі Home Assistant за її внутрішньою адресою, з будь-якої точки світу, при цьому весь трафік керування надійно захищений від перехоплення чи модифікації.

Сервер це публічна айпі мікротіка, «Запис» це логін користувача, «Пароль» це пароль користувача, «Секрет» це спільний ключ IPsec.

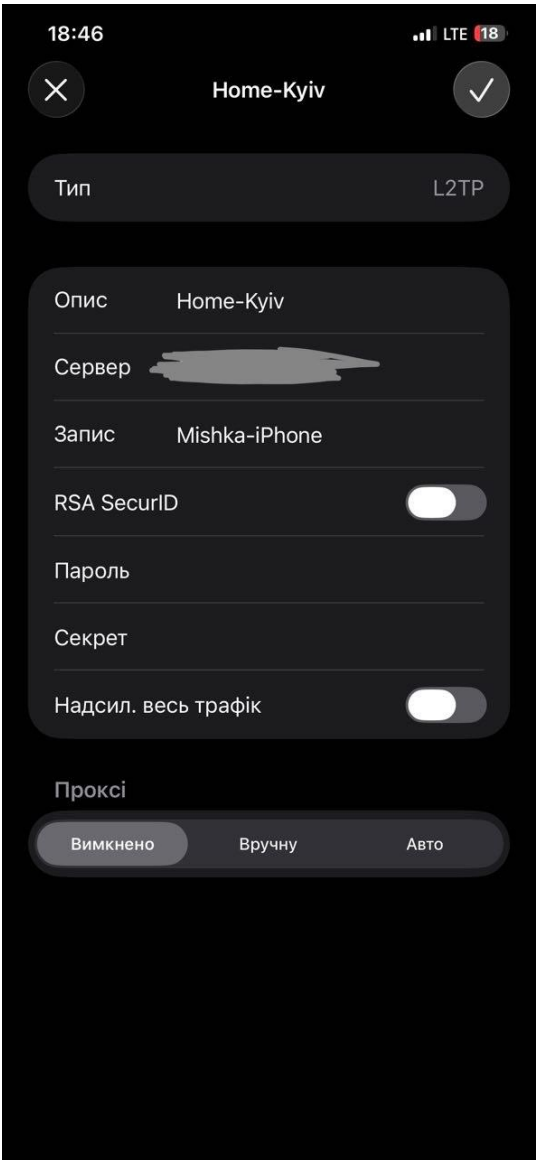


Рисунок 3.9 — Графічний інтерфейс налаштування клієнтського VPN тунелю на мобільному пристрої

3.4 Інтеграція та налаштування модуля сповіщень на базі платформи Telegram

Для забезпечення оперативного інформування оператора про поточний стан енергопостачання об'єкта та критичні зміни параметрів акумуляторних батарей у систему впроваджено модуль віддалених сповіщень. Як комунікаційну платформу обрано месенджер Telegram, що дозволяє безкоштовно та з мінімальними затримками передавати текстові й графічні повідомлення через захищені протоколи зв'язку.[45]

Процес створення інтеграційного каналу починається на рівні сервісів розробника Telegram за допомогою спеціалізованого робота BotFather. Шляхом відправки відповідних команд ініціюється реєстрація нового унікального бота системи моніторингу, для якого платформа генерує захищений цифровий токен доступу типу NTTP API.

Основна інженерна цінність даної підсистеми полягає у логіці роботи автоматизацій, які безперервно аналізують показники раніше створених сенсорів. У межах проекту розроблено сценарії реагування на три ключові події, пов'язані зі зміною режимів роботи та ємності джерела живлення. Перший сценарій контролює знеструмлення об'єкта та відновлення живлення, де головним тригером виступає зміна стану сенсора напруги або вхідної мережі, після чого бот миттєво надсилає повідомлення про перехід системи в автономний режим або повернення до штатної схеми живлення.

Другий сценарій розрахований на досягнення верхнього порогу розряду акумулятора, за якого при виснаженні батареї станції Bluetti до рівня 85% система надсилає перше попереджувальне сповіщення, що сигналізує про необхідність оптимізації поточного споживання. Третій сценарій активується при досягненні критичного енергетичного порогу, коли у разі падіння рівня заряду до 70%. Фінальний результат функціонування розроблених алгоритмів сповіщення та реальні приклади повідомлень, які приходять на мобільний термінал адміністратора у режимі реального часу, відображено на рисунку 3.4.

HTML коди тригерів наведено в додатках А, Б та В.

					<i>PE22.461142.001 ПЗ</i>	Лис
						54
Зм.	Лис	№ докум.	Підпис	Дата		

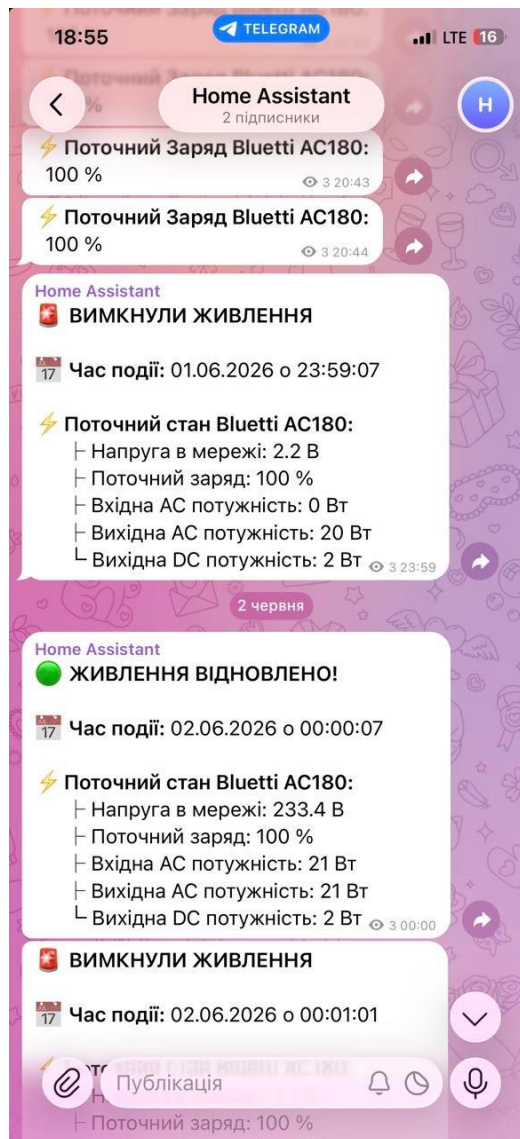


Рисунок 3.4 — Інтерфейс чат-бота Telegram із результатами генерації автоматичних сповіщень системи

3.5 Розробка графічного інтерфейсу користувача та побудова панелі керування

Завершальним етапом практичної реалізації системи моніторингу є проектування та візуалізація графічного інтерфейсу користувача. Головною метою розробки панелі керування є наочне представлення складних технічних параметрів малої енергосистеми у вигляді, зручному для швидкого аналізу, а також надання інструментів для зворотного зв'язку та активного керування силовим обладнанням.[46]

					РЕ22.461142.001 ПЗ	Лис
						55
Зм.	Лис	№ докум.	Підпис	Дата		

Графічний інтерфейс побудовано за модульним принципом з використанням стандартних та кастомних карток відображення компонентів. На головному екрані панелі реалізовано інформаційні віджети для трансляції телеметрії зі станції Bluetti, де за допомогою та лінійних індикаторів відображається поточний рівень ємності батареї, а також цифрові графіки миттєвої потужності генерації сонячних панелей та споживання підключеного навантаження.[47]

Окрім пасивного спостереження, на дашборд винесено інтерактивні елементи активного керування, які взаємодіють із сутностями типу switch кастомного драйвера. Основним виконавчим елементом виступає графічна кнопка-перемикач для дистанційного увімкнення та вимкнення силового інвертора змінного струму станції. Команда на перемикання шифрується всередині локальної мережі й передається по протоколу Bluetooth Low Energy безпосередньо на контролер станції, забезпечуючи миттєву реакцію обладнання на дії оператора.[48]



Рисунок 3.5.1 — Візуальне представлення розробленого графічного інтерфейсу користувача в Home Assistant

					РЕ22.461142.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		56

Загальний вигляд розробленої графічної панелі моніторингу та кнопки віддаленого керування представлено на рисунку 3.5.1, а функціонал блокування кнопки на рисунку 3.5.2.

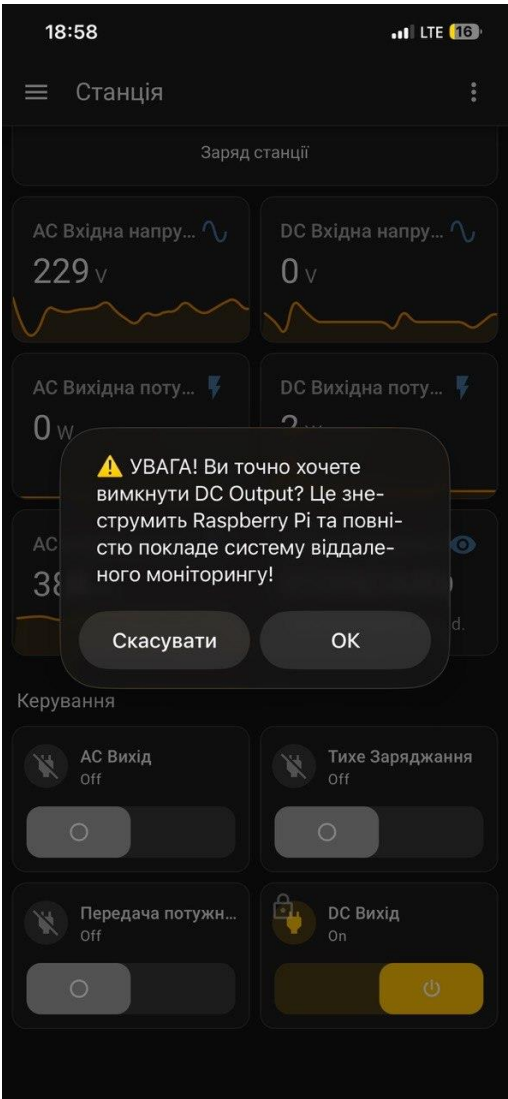


Рисунок 3.5.2 — Візуальне представлення попереджувального сповіщення при натисканні кнопки «DC Вихід»

Загальну схему, як відбувається зв’язок користувача зі станцією та логіку роботи системи продемонстровано в додатку Г.

3.6 Експериментальне дослідження енергоспоживання системи моніторингу

Для оцінки реальної автономності та енергетичного балансу розробленого програмно-апаратного комплексу було проведено довготривалий практичний експеримент. Перед початком вимірювань акумуляторну батарею портативної зарядної станції Bluetti номінальною ємністю 1 кВт·год було повністю

розряджено, а потім заряджено до максимального рівня 100%. До DC-виходу станції через один кабель було підключено мікрокомп'ютер Raspberry Pi 4 Model B із розгорнутим операційним середовищем Home Assistant OS. Фіксація даних здійснювалася при вимкненому дисплеї зарядної станції за допомогою створених віртуальних сенсорів телеметрії.

У процесі моніторингу було зафіксовано чотири контрольні часові точки зміни залишку заряду акумулятора. Стартова точка зафіксована 28 квітня о 23:08 при рівні заряду 100% (рис. 3.6.1).

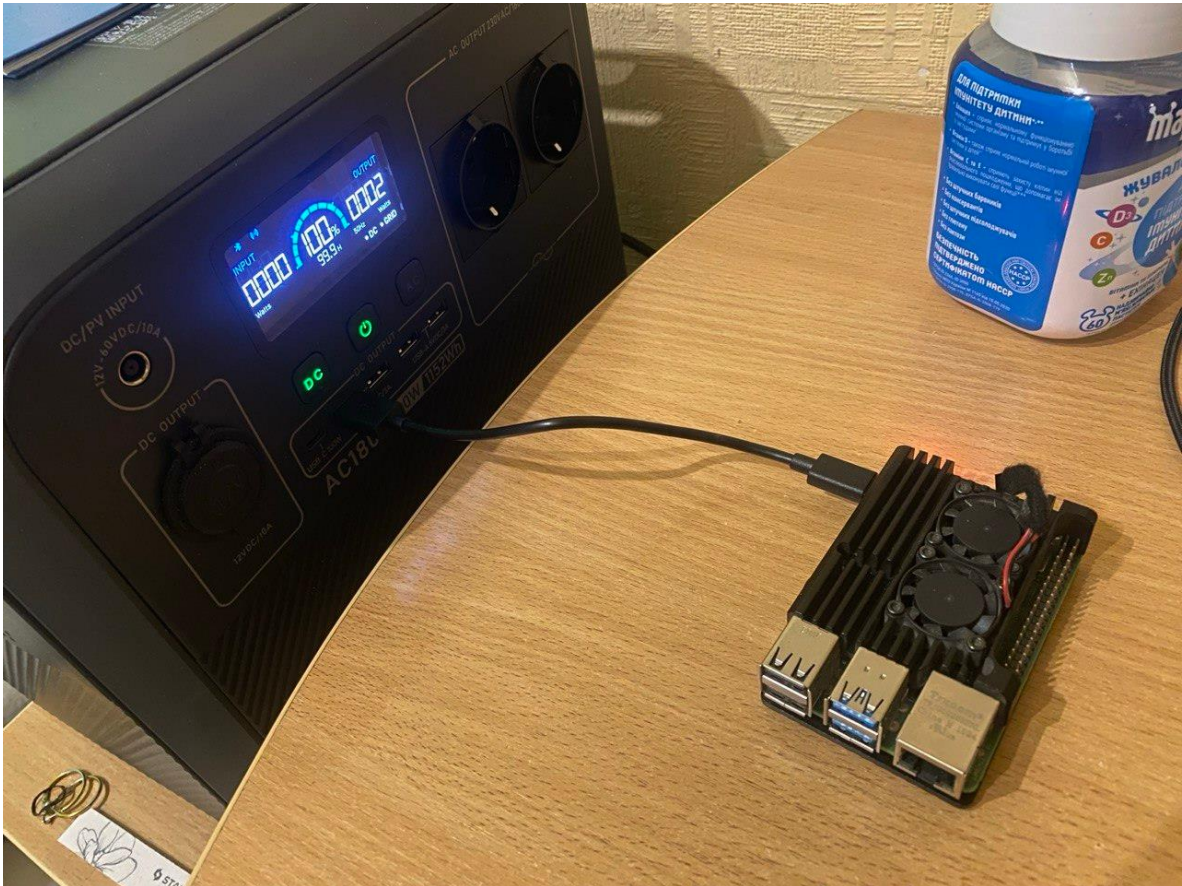


Рисунок 3.6.1 — Фіксація заряду станції 28 квітня 2026 року о 23:08

Наступні заміри показали поступове лінійне зниження ємності акумулятора зарядної станції. Так, 29 квітня о 22:39 рівень заряду становив 81% (рис. 3.6.2), а 1 травня о 17:43 вказаний показник знизився до 50% (рис. 3.6.3). Завершальний замір, проведений 2 травня о 16:34, зафіксував 34% залишку заряду батареї (рис. 3.6.4). Загальна тривалість дослідження від першої до четвертої точки склала 89 годин 26 хвилин, за які система витратила 66% загальної ене-

ргії станції. Аналіз математичної залежності дозволив встановити, що швидкість лінійного розряду в середньому становить 1,35 години (близько 81 хвилини) на кожен 1% ємності акумулятора..



Рисунок 3.6.2 — Фіксація заряду станції 29 квітня 2026 року о 22:39

Зм.	Лис	№ докум.	Підпис	Дата

PE22.461142.001 ПЗ



Рисунок 3.6.3 — Фіксація заряду станції 1 травня 2026 року о 17:43

					РЕ22.461142.001 ПЗ	Лис
Зм.	Лис	№ докум.	Підпис	Дата		60



Рисунок 3.6.4 — Фіксація заряду станції 2 травня 2026 року о 16:34

Головна інженерна особливість отриманих результатів полягає в розбіжності між теоретичним та фактичним споживанням. При чистому споживанні плати Raspberry Pi 4 на рівні 2 Вт теоретичні витрати за перші дві доби мали становити близько 133 Вт·год, проте фактичні втрати ємності станції за лічильником BMS склали 50% (приблизно 500 Вт·год). Окрім власного споживання внутрішньої плати керування станції (BMS та модуля Bluetooth), ключовим фактором такої розбіжності є просадка напруги на вихідному шлюзі та інтерфейсному кабелі під постійним навантаженням. Оскільки алгоритми станції вираховують залишок відсотків ємності на основі поточного вольтажу акумуляторних комірок, динамічне падіння напруги призводить до того, що система відображає занижений відсоток реальної місткості, хоча фактичний

Зм.	Лис	№ докум.	Підпис	Дата

PE22.461142.001 ПЗ

Лис
61

енергетичний резерв залишається вищим. Таким чином, з урахуванням апаратних похибок вимірювання та внутрішніх витрат, сумарне розрахункове споживання комплексу склало близько 7,5 Вт/год.

					РЕ22.461142.001 ПЗ	Лис
						62
Зм.	Лис	№ докум.	Підпис	Дата		

ВИСНОВКИ

У дипломному проекті успішно вирішено актуальну науково-практичну задачу з розробки, програмно-апаратної реалізації та захисту системи автоматизованого моніторингу й керування малими джерелами енергопостачання. Отримані результати повністю відповідають сучасному рівню технічних знань у галузі децентралізованих інтелектуальних мереж та концепції Інтернету речей. Використання платформи з відкритим кодом Home Assistant на базі мікрокомп'ютера Raspberry Pi 4 Model B дозволило створити повністю локальне, автономне рішення, яке функціонує без використання сторонніх комерційних хмарних сервісів.

В ході експериментальної перевірки повністю доведено практичку працездатність та високу ефективність створеного проєкту. Розроблена система демонструє високу швидкість реагування на аварійну втрату основного живлення об'єкта, яка не перевищує меж від 6 до 8 секунд від моменту аварії до відправки повідомлення. Завдяки оптимізації бездротового каналу Bluetooth Low Energy досягнуто пристойного часу відгуку, за якого мікрокомп'ютер здійснює циклічне опитування внутрішніх регістрів контролера зарядної станції кожні 6 секунд, забезпечуючи високу точність телеметрії.

Технічна цінність та безпека комплексу підтверджується впровадженням захищеного криптографічного тунелю L2TP VPN на рівні маршрутизатора MikroTik, що повністю убезпечило локальну адмін-панель від несанкціонованого зовнішнього доступу без прямого відкриття портів. Додатково реалізовано безпечний метод інформування користувача за допомогою інтеграції з API Telegram, де створений чат-бот налаштований виключно на одностороннє приймання екстрених сповіщень від сервера.

Аналіз використання апаратних ресурсів сервера показав, що споживання потужності мікрокомп'ютера Raspberry Pi 4 залишається стабільно низьким, фіксуючи в пікових режимах навантаження лише 8% завантаженості центрального процесора та 24% об'єму оперативної пам'яті. Проведені енергетичні

					РЕ22.461142.001 ПЗ	Лис
						63
Зм.	Лис	№ докум.	Підпис	Дата		

тести підтвердили стабільний лінійний розряд системи зі швидкістю 1,35 години на 1% ємності, а також дозволили виявити системний ефект просадки напруги на вихідному шлюзі, який впливає на фіксацію залишку заряду інтегрованим контролером BMS. Створений програмно-апаратний комплекс повністю готовий до впровадження на автономних об'єктах та має величезний ресурсний запас для майбутніх інженерних досліджень.

					РЕ22.461142.001 ПЗ	Лис
						64
Зм.	Лис	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Raspberry Pi 4 Model B specifications. URL: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/> (дата звернення: 24.04.2026).
2. Raspberry Pi - Home Assistant Installation Documentation. URL: <https://www.home-assistant.io/installation/raspberrypi/> (дата звернення: 24.04.2026).
3. Google Gemini App. URL: <https://gemini.google.com/app> (дата звернення: 24.04.2026).
4. How to Install Home Assistant on a Raspberry Pi (Beginners Guide) : веб-сайт / Smart Home Theory. URL: <https://www.youtube.com/watch?v=xSqopd0eARI> (дата звернення: 24.04.2026).
5. Home Assistant on Raspberry Pi: perfect first project. Reddit. URL: https://www.reddit.com/r/raspberry_pi/comments/1olkvg9/home_assistant_on_raspberry_pi_perfect_first/ (дата звернення: 24.04.2026).
6. How to troubleshoot Raspberry Pi crashing. Home Assistant Community. URL: <https://community.home-assistant.io/t/how-to-troubleshoot-raspberry-pi-crashing/703179> (дата звернення: 24.04.2026).
7. Boot message: Waiting for the Home Assistant CLI to be ready. Home Assistant Community Forum. URL: <https://community.home-assistant.io/t/boot-message-waiting-for-the-home-assistant-cli-to-be-ready/814558/2> (дата звернення: 24.04.2026).
8. Fixed: Home Assistant CLI not starting for new installations. Reddit. URL: https://www.reddit.com/r/homeassistant/comments/1e0j1cf/fixed_home_assistant_cli_not_starting_for_new/ (дата звернення: 24.04.2026).
9. Waiting for the Home Assistant CLI troubleshooting guide. ThreatResearcher. URL: <https://blog.threatresearcher.com/waiting-for-the-home-assistant-cli/> (дата звернення: 24.04.2026).
10. Boot message: Waiting for the Home Assistant CLI to be ready. Post 8. Home Assistant Community Forum. URL: <https://community.home->

					PE22.461142.001 ПЗ	Лис
						65
Зм.	Лис	№ докум.	Підпис	Дата		

assistant.io/t/boot-message-waiting-for-the-home-assistant-cli-to-be-ready/814558/8 (дата звернення: 24.04.2026).

11. First install attempt results in WARN: Home Assistant CLI not starting, jump into emergency console. Home Assistant Community. URL: <https://community.home-assistant.io/t/first-install-attempt-results-in-warn-home-assistant-cli-not-starting-jump-into-emergency-console/740859> (дата звернення: 26.04.2026).

12. HA installation problem: CLI issues on initial boot. Home Assistant Community. URL: <https://community.home-assistant.io/t/ha-installation-problem/911684/3> (дата звернення: 28.05.2026).

13. HACS Download and Installation Guide. Home Assistant Community Store. URL: <https://www.hacs.xyz/docs/use/download/download/> (дата звернення: 28.05.2026).

14. Common Tasks: Installing a third-party app repository. Home Assistant Documentation. URL: <https://www.home-assistant.io/common-tasks/os/#installing-a-third-party-app-repository> (дата звернення: 28.05.2026).

15. Bluetti Official Home Assistant Integration Repository. GitHub. URL: <https://github.com/bluetti-official/bluetti-home-assistant> (дата звернення: 28.05.2026).

16. Bluetti is finally getting an official smart home integration. Reddit. URL: https://www.reddit.com/r/homeassistant/comments/1oiysir/bluetti_is_finally_getting_a_official_smart_home/ (дата звернення: 28.05.2026).

17. Hass.io Bluetti Bluetooth Old Driver Repository / 0x4E4448. GitHub. URL: <https://github.com/0x4E4448/hassio-bluetti-bt-old> (дата звернення: 28.05.2026).

18. Bluetti Bluetooth Python Library: Supported Powerstations and Data / Patrick762. GitHub. URL: <https://github.com/Patrick762/bluetti-bt-lib?tab=readme-ov-file#supported-powerstations-and-data> (дата звернення: 28.05.2026).

					<i>РЕ22.461142.001 ПЗ</i>	Лис
						66
Зм.	Лис	№ докум.	Підпис	Дата		

19. Форум DOU: Обговорення розгортання та інтеграції систем моніторингу енергопостачання. DOU. URL: <https://dou.ua/forums/topic/51385/> (дата звернення: 28.05.2026).

20. Hass.io Bluetti Bluetooth Integration / Patrick762. GitHub. URL: <https://github.com/Patrick762/hassio-bluetti-bt> (дата звернення: 28.05.2026).

21. Bluetti2MQTT Bridge Gateway Component / semitop7. GitHub. URL: <https://github.com/semitop7/bluetti2mqtt/tree/main/bluetti2mqtt> (дата звернення: 28.05.2026).

22. Bluetti2MQTT Issue #54: Connection timeouts during Bluetooth polling / SSMCD. GitHub. URL: <https://github.com/SSMCD/bluetti2mqtt/issues/54> (дата звернення: 28.05.2026).

23. Bluetti MQTT Core Service Integration / ftrueck. GitHub. URL: https://github.com/ftrueck/bluetti_mqtt/tree/main (дата звернення: 28.05.2026).

24. Bluetti2MQTT Project Issues Tracker / SSMCD. GitHub. URL: <https://github.com/SSMCD/bluetti2mqtt/issues> (дата звернення: 28.05.2026).

25. Bluetti MQTT Driver implementation / warhammerkid. GitHub. URL: https://github.com/warhammerkid/bluetti_mqtt (дата звернення: 28.05.2026).

26. Bluetti MQTT Driver Issue #135: Communication glitches on low battery states / warhammerkid. GitHub. URL: https://github.com/warhammerkid/bluetti_mqtt/issues/135 (дата звернення: 28.05.2026).

27. Problem with new AC180P power station monitoring. Bluetti Power Community Forum. URL: <https://community.bluettipower.com/t/problem-with-new-ac180p/28425/112?page=7> (дата звернення: 28.05.2026).

28. Patrick762 Projects: Issue Tracker Item #146514322. GitHub Projects. URL: <https://github.com/users/Patrick762/projects/4/views/1?pane=issue&itemId=146514322&issue=Patrick762%7Cbluetti-bt-lib%7C2> (дата звернення: 28.05.2026).

29. Bluetti MQTT Driver Issue #141: Encryption and authentication protocol changes / warhammerkid. GitHub. URL:

					<i>PE22.461142.001 ПЗ</i>	Лис
						67
Зм.	Лис	№ докум.	Підпис	Дата		

https://github.com/warhammerkid/bluetti_mqtt/issues/141 (дата звернення: 28.05.2026).

30. Bluetti MQTT Driver Issue #129: Bluetooth LE state machine errors / warhammerkid. GitHub. URL: https://github.com/warhammerkid/bluetti_mqtt/issues/129 (дата звернення: 28.05.2026).

31. Hass.io Bluetti BT custom component alternative branch / cmorgannorris. GitHub. URL: <https://github.com/cmorgannorris/hassio-bluetti-bt> (дата звернення: 28.05.2026).

32. Bluetti HA support including encryption handling. Home Assistant Community. URL: <https://community.home-assistant.io/t/bluetti-ha-support-including-encryption/997610> (дата звернення: 28.05.2026).

33. Hass.io Bluetti Bluetooth component main branch / 0x4E4448. GitHub. URL: <https://github.com/0x4E4448/hassio-bluetti-bt> (дата звернення: 28.05.2026).

34. Hass.io Bluetti Bluetooth Integration / atiweb. GitHub. URL: <https://github.com/atiweb/hassio-bluetti-bt> (дата звернення: 28.05.2026)

35. ZeroTier Virtual Networking Platform. URL: <https://www.zerotier.com/> (дата звернення: 28.05.2026).

36. WireGuard Secure VPN Tunnel Protocol. URL: <https://www.wireguard.com/> (дата звернення: 28.05.2026).

37. Back To Home (BTH) VPN Service Architecture. MikroTik RouterOS Solutions. URL: <https://mikrotik.com/bth> (дата звернення: 28.05.2026).

38. What does this built-in VPN do? iOS implementation details. Reddit. URL: https://www.reddit.com/r/iphone/comments/1huoxpv/what_does_this_built_in_vpn_do_does_it_have_all/ (дата звернення: 28.05.2026).

39. Apple Deployment Guide: VPN settings overview for Apple devices. Apple Support. URL: <https://support.apple.com/uk-ua/guide/deployment/depaec3d361d0/web> (дата звернення: 28.05.2026).

40. Apple Platform Security: VPN security architecture reference. Apple Support. URL: <https://support.apple.com/uk-ua/guide/security/sec802e8ab55/web> (дата звернення: 28.05.2026).

					<i>РЕ22.461142.001 ПЗ</i>	Лис
						68
Зм.	Лис	№ докум.	Підпис	Дата		

41. Wireguard setup with MikroTik and your smartphone : веб-сайт / MikroTik. URL: <https://www.youtube.com/watch?v=vn9ky7p5ESM> (дата звернення: 28.05.2026).

42. Wireguard and iOS connection persistent issues. MikroTik Community Forum. URL: <https://forum.mikrotik.com/t/wireguard-and-ios/177383/6> (дата звернення: 28.05.2026).

43. MikroTik iOS App can't connect over WireGuard using IP. MikroTik Community Forum. URL: <https://forum.mikrotik.com/t/mikrotik-ios-app-cant-connect-over-wireguard-using-ip/182647> (дата звернення: 28.05.2026).

44. How to setup L2TP VPN connection on iPhone and iPad manually : веб-сайт / S4M VPN Setup. URL: <https://www.youtube.com/watch?v=QzOkBInNP5s> (дата звернення: 28.05.2026).

45. Telegram Bots: Developers Tutorial and API Introduction. Telegram Core Documentation. URL: <https://core.telegram.org/bots/tutorial> (дата звернення: 28.05.2026).

46. Dashboards: Actions configuration and Tap Action features. Home Assistant Core Documentation. URL: <https://www.home-assistant.io/dashboards/actions/#tap-action> (дата звернення: 28.05.2026).

47. Confirmation option on switches and buttons etc. Post 10. Home Assistant Community. URL: <https://community.home-assistant.io/t/confirmation-option-on-switches-and-buttons-etc/665213/10> (дата звернення: 29.05.2026).

48. Dashboards: Creating a new dashboard. Home Assistant Core Documentation. URL: <https://www.home-assistant.io/dashboards/dashboards/#creating-a-new-dashboard> (дата звернення: 29.05.2026).

					<i>РЕ22.461142.001 ПЗ</i>	Лис
						69
Зм.	Лис	№ докум.	Підпис	Дата		

ДОДАТОК А. HTML КОД ТРИГЕРА НА ВИМКНЕННЯ ЖИВЛЕННЯ

 ВИМКНУЛИ ЖИВЛЕННЯ

 Час події: {{ now().strftime('%d.%m.%Y о %H:%M:%S') }}

 Поточний стан Bluetti AC180:

└ Напруга в мережі: {{
states('sensor.ac1802523000010809_ac_input_voltage') }} В

└ Поточний заряд: {{ states('sensor.ac1802523000010809_battery_soc')
}} %

└ Вхідна АС потужність: {{
states('sensor.ac1802523000010809_ac_input_power') }} Вт

└ Вихідна АС потужність: {{
states('sensor.ac1802523000010809_ac_output_power') }} Вт

└ Вихідна DC потужність: {{
states('sensor.ac1802523000010809_dc_output_power') }} Вт

					РЕ22.461142.001 ПЗ	Лис
						70
Зм.	Лис	№ докум.	Підпис	Дата		

ДОДАТОК Б. HTML КОД ТРИГЕРА НА УВІМКНЕННЯ ЖИВЛЕННЯ

 **ЖИВЛЕННЯ ВІДНОВЛЕНО!**

 **Час події:** {{ now().strftime('%d.%m.%Y о %H:%M:%S') }}

 **Поточний стан Bluetti AC180:**

└ Напруга в мережі: {{
states('sensor.ac1802523000010809_ac_input_voltage') }} В

└ Поточний заряд: {{ states('sensor.ac1802523000010809_battery_soc')
}} %

└ Вхідна AC потужність: {{
states('sensor.ac1802523000010809_ac_input_power') }} Вт

└ Вихідна AC потужність: {{
states('sensor.ac1802523000010809_ac_output_power') }} Вт

└ Вихідна DC потужність: {{
states('sensor.ac1802523000010809_dc_output_power') }} Вт

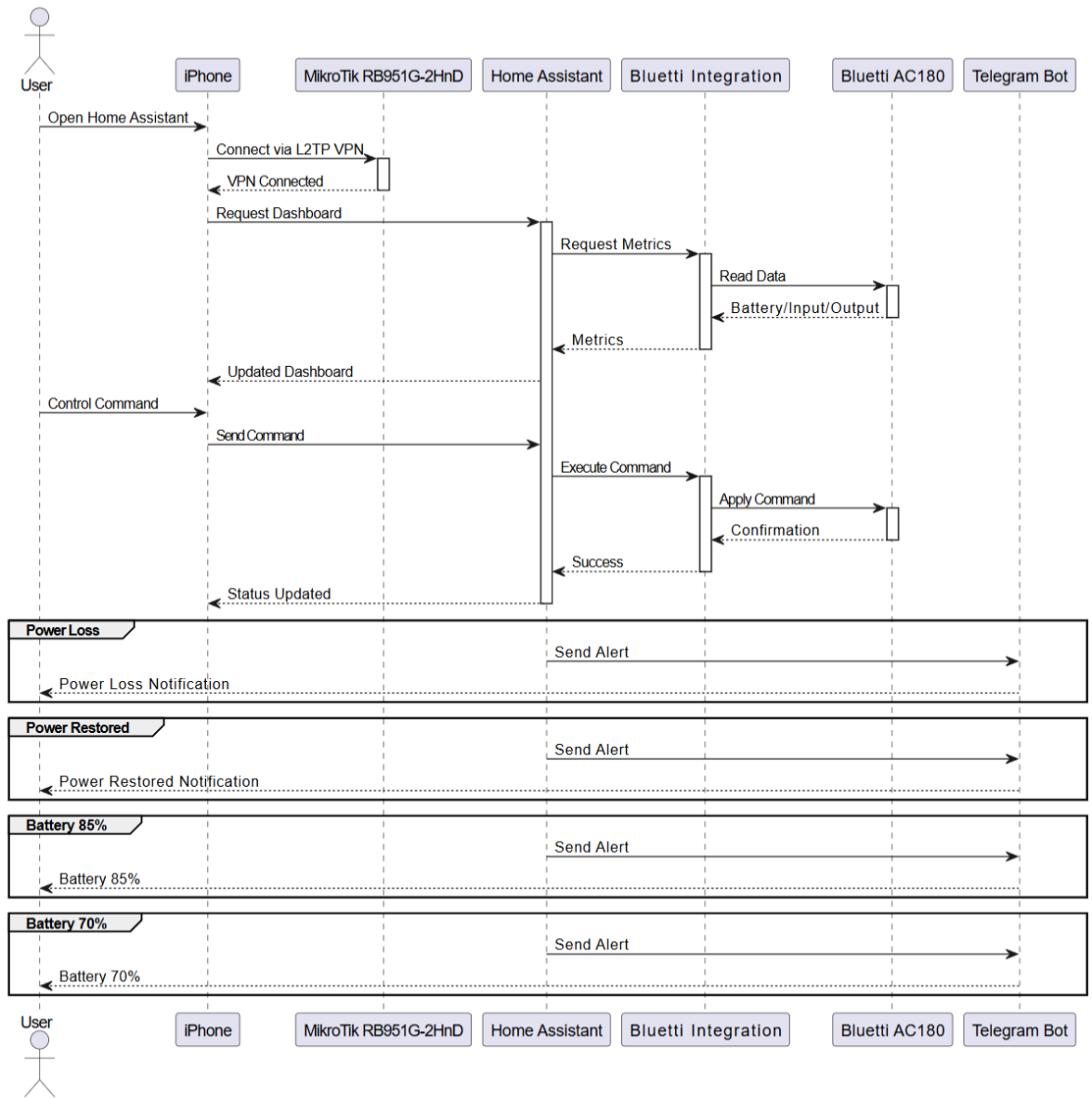
					РЕ22.461142.001 ПЗ	Лис
						71
Зм.	Лис	№ докум.	Підпис	Дата		

ДОДАТОК В. HTML КОД ТРИГЕРА НА ПОТОЧНИЙ ЗАРЯД

 Поточний Заряд Bluetti AC180: {{
states('sensor.ac1802523000010809_battery_soc') }} %

					РЕ22.461142.001 ПЗ	Лис
						72
Зм.	Лис	№ докум.	Підпис	Дата		

ДОДАТОК Г. UML ДІАГРАМА ПОСЛІДОВНОСТІ



ДОДАТОК Д. ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

[illegible]

				РЕ22.461142.001		
	ПБ	Підп.	Дата	Система Захищеного віддаленого керування ДСЕ	Лист	Листів
Розробн.	Крамаренко М.О.				1	1
Керівн.	Єзерський Н.В.				КП ім. Ігоря Сікорського Каф. ПРЕ, Гр. РЕ-22	
Н/контр.						
Зав.каф.						

					РЕ22.461142.001 ПЗ	Лис
						74
Зм.	Лис	№ докум.	Підпис	Дата		