

**Міністерство освіти і науки України**

**Луцький національний технічний університет**

(повне найменування закладу вищої освіти)

**Факультет комп'ютерних та інформаційних технологій**

(повне найменування факультету)

**Кафедра комп'ютерної інженерії та безпеки**

(повне найменування кафедри)

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

**СИСТЕМА БАГАТОКАНАЛЬНОГО МОНІТОРИНГУ ЗАРЯДУ  
АКУМУЛЯТОРНОЇ БАТАРЕЇ**

**MULTI-CHANNEL BATTERY CHARGE MONITORING SYSTEM**

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти

групи КІ-42

Місюра Дмитро Романович

(підпис)

Керівник:

к.т.н., доцент

Кайдик Олег Леонтійович

(підпис)

Кваліфікаційну роботу

допущено до захисту

« 04 » червня 2025 р.

Гарант освітньої програми:

к.т.н., доцент

Лавренчук Світлана Василівна

(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та безпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц. Тарас ТЕРЛЕЦЬКИЙ

« 10 » 01 2025 р.

ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

*Місюрі Дмитру Романовичу*

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Система багатоканального моніторингу заряду акумуляторної батареї

Керівник роботи к.т.н., доц. Кайдик Олег Леонтійович

затвержені наказом закладу вищої освіти від «04» січня 2025 року № 11/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 10.06.2025р.

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

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

Вступ

Аналіз предметної області та наявних рішень

Вибір апаратної та програмної бази для проєкту

Проектування та тестування системи моніторингу заряду батареї

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Електрична принципова схема пристрою

Інтерфейс пристрою

Тестування системи багатоканального моніторингу

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз проблеми за темою роботи та постановка завдань дослідження</i>	<i>Кайдик О.Л., доцент</i>		
<i>Вибір апаратної та програмної бази для проєкту</i>	<i>Кайдик О.Л., доцент</i>		
<i>Практична реалізація об'єкта проєктування</i>	<i>Кайдик О.Л., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>	_____ %		
<i>Академічна доброчесність</i>	<i>Міскевич О.І., ст. викладач</i>		

7. Дата видачі завдання 10.01.2025 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Огляд літератури із досліджуваної проблеми, аналіз предметної області та наявних рішень</i>	до 10.02.2025 р.	Виконано
2.	<i>Апаратна реалізація системи</i>	до 02.03.2025 р.	Виконано
3.	<i>Проєктування та тестування системи моніторингу заряду батареї</i>	до 02.04.2025 р.	Виконано
4.	<i>Висновки та пропозиції</i>	до 10.04.2025 р.	Виконано
5.	<i>Формування списку використаних джерел</i>	до 15.04.2025 р.	Виконано
6.	<i>Формування додатків</i>	до 02.05.2025 р.	Виконано
7.	<i>Оформлення ілюстративного матеріалу</i>	до 10.05.2025 р.	Виконано
8.	<i>Представлення остаточного варіанту кваліфікаційної роботи бакалавра керівникові</i>	до 15.05.2025 р.	Виконано
9.	<i>Нормоконтроль</i>	до 30.05.2025 р.	Виконано
10.	<i>Інструментальна перевірка на академічний плагіат</i>	до 03.06.2025 р.	Виконано
11.	<i>Здача кваліфікаційної роботи та всіх супровідних документів на кафедрі</i>	до 10.06.2025 р.	Виконано

Здобувач вищої освіти

\_\_\_\_\_ (підпис)

Місюра Д.Р.

\_\_\_\_\_ (прізвище, ініціали)

Керівник кваліфікаційної роботи

\_\_\_\_\_ (підпис)

Кайдик О.Л.

\_\_\_\_\_ (прізвище, ініціали)

## АНОТАЦІЯ

Місюра Д.Р. Система багатоканального моніторингу заряду акумуляторної батареї. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2025.

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

Перший розділ присвячено огляду предметної області, тут наведено загальні відомості про акумуляторні батареї, їх склад та розміри, основні електричні та експлуатаційні параметри. Особливу увагу приділено питанням та викликам багатоканального моніторингу, також здійснено огляд подібних промислових пристроїв та аналіз джерел літератури в цьому напрямку.

В другому розділі обґрунтовано вибір мікроконтролера зв контексті поставленого завдання, також описано монтаж пристрою з урахуванням особливостей будови мікроконтролера.

Третій розділ присвячено алгоритмам створення прошивки мікроконтролера системи багатоканального моніторингу, описано два варіанти прошивки, наведено їх порівняння, також описано тестування розробленого пристрою.

Ключові слова: ESP-WROOM-32, акумулятор, дільник напруг, АЦП, Arduino IDE.

## ANNOTATION

Misiura D. Multi-channel battery charge monitoring system. Manuscript.

Qualifying work of a bachelor of EP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

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

The first chapter is devoted to an overview of the subject area, where general information about batteries, their composition, and dimensions, as well as basic electrical and operational parameters, is provided. Particular attention is paid to the issues and challenges of multi-channel monitoring, a review of similar industrial devices, and an analysis of literature sources in this direction are also carried out.

The second chapter justifies the choice of a microcontroller in the context of the task and also describes the installation of the device, taking into account the features of the microcontroller structure.

The third chapter is devoted to the algorithms for creating the firmware of the microcontroller of the multi-channel monitoring system. Two firmware options are described, their comparison is provided, and testing of the developed device is also described.

Keywords: ESP-WROOM-32, battery, voltage divider, ADC, Arduino IDE.

## ЗМІСТ

ВСТУП .....	7
РОЗДІЛ 1 ТЕОРЕТИЧНІ ЗАСАДИ ТА СУЧАСНІ ПІДХОДИ ДО МОНІТОРИНГУ АКУМУЛЯТОРНИХ БАТАРЕЙ.....	9
1.1 Загальні відомості про акумуляторні батареї.....	9
1.1.1 Типи акумуляторних батарей .....	11
1.1.2 Ключові електричні та експлуатаційні параметри батарей.....	12
1.1.3 Основні поняття та визначення в контексті моніторингу .....	15
1.2 Проблеми та виклики багатоканального моніторингу.....	16
1.3 Огляд існуючих рішень .....	18
РОЗДІЛ 2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ СИСТЕМИ БАГАТОКАНАЛЬНОГО МОНІТОРИНГУ .....	22
2.1 Обґрунтування вибору мікроконтролера та його характеристики .....	22
2.2 Аналогово-цифровий перетворювач.....	26
2.3 Монтаж пристрою .....	28
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ МОНІТОРИНГУ .....	32
3.1 Початкові налаштування середовища програмування.....	32
3.2 Налаштування та калібрування АЦП для діапазону напруги .....	35
3.3 Створення прошивки мікроконтролера .....	38
3.4 Тестування система багатоканального моніторингу.....	40
ВИСНОВКИ.....	42
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	43
ДОДАТКИ.....	46

## ВСТУП

Акумуляторні батареї відіграють важливу роль у сучасному житті, адже вони використовуються у багатьох сферах: портативна електроніка (смартфони, планшети, ноутбуки), електротранспорт, системи резервного живлення та резервного освітлення [17], промислові пристрої (сенсори, вимірювальне обладнання, роботи тощо), домашні побутові пристрої, акумуляторні електроінструменти, дрони, відеокамери, тощо. В багатьох випадках ефективний моніторинг заряду таких батарей є критично важливим, як для їхньої довговічності, так і для оптимальної роботи, а часто і для безпеки.

Сучасні акумуляторні батареї часто складаються з багатьох елементів (комірок), з'єднаних послідовно та паралельно (наприклад, у автономних IoT-пристроях, електротранспорті, IT-інфраструктурі, телекомунікаційному обладнанні та ін.). Важливою задачею є моніторинг заряду кожної комірки окремо, що дасть можливість балансування напруги, уникнути перезаряду чи перерозряду та виявити несправності на ранніх стадіях. Тому звичайні методи моніторингу (наприклад, вимірювання загальної напруги) є недостатніми для безпечної та ефективної роботи складних систем, що використовують велику кількість акумуляторних елементів. Багатоканальний моніторинг дозволить контролювати кожен елемент акумуляторної системи окремо, що суттєво підвищить надійність роботи, безпеку експлуатації та довговічність батарей. Тому обрана тема кваліфікаційної роботи є актуальною та матиме практичне значення.

Мета роботи – розробити систему багатоканального моніторингу заряду акумуляторної батареї з використанням засобів комп'ютерної інженерії для забезпечення вимірювання та відображення параметрів окремих елементів акумуляторного блоку в режимі реального часу.

Об'єкт дослідження – акумуляторна батарея як складова частина автономної системи живлення.

Предмет дослідження – методи, засоби та алгоритми комп'ютерної інженерії для вимірювання, збору, обробки та передавання даних про стан окремих елементів акумуляторної батареї.

Завдання, які необхідно виконати:

- проаналізувати існуючі підходи до моніторингу заряду акумуляторних батарей;

- обґрунтувати вибір апаратної платформи (мікроконтролера) та сенсорних модулів;

- розробити електричну схему системи багатоканального моніторингу;

- написати програмне забезпечення для збору та обробки даних з декількох каналів;

- реалізувати інтерфейс для виведення інформації користувачеві;

- протестувати розробку.

Матеріали кваліфікаційної роботи обговорювалися на міжнародній науково-практичній конференції [22] (додаток А) в травні 2025 року. За результатами дослідження опубліковано тези [23] (додаток Б).

# РОЗДІЛ 1

## ТЕОРЕТИЧНІ ЗАСАДИ ТА СУЧАСНІ ПІДХОДИ ДО МОНІТОРИНГУ АКУМУЛЯТОРНИХ БАТАРЕЙ

### 1.1 Загальні відомості про акумуляторні батареї

Акумуляторна батарея – це автономне джерело живлення, яке здатне накопичувати, зберігати та віддавати енергію. В основі функціонування – багаторазове (цикли заряду-розряду) перетворення хімічної енергії на електричну, що забезпечує повторне використання таких видів батарей.

Першу свинцево-кислотну акумуляторну батарею, здатну до багаторазової перезарядки й конструктивно близьку до сучасних аналогів, було створено в другій половині XIX століття французьким ученим Гастоном Планте [18]. У своїй конструкції він використав дві свинцеві пластини, занурені в розчин сірчаної кислоти, між якими була розміщена ізоляція (гумова стрічка). Такий пристрій дозволяв заряджати акумулятор за допомогою зовнішнього джерела напруги, а потім віддавати накопичену енергію у вигляді електричного струму. Першу модель Plante використовували для живлення освітлення у вагонах під час зупинки [21]. На той час цей винахід дав великий поштовх для розвитку автомобільної галузі та резервного енергозабезпечення.

У 1881 році французький інженер-хімік Каміль Альфонс Фор модифікував початкову конструкцію, він у свинцеву решітку втискав пасту оксиду свинцю, утворюючи таким чином двокомпонентну пластину. Можна було поєднувати кілька таких пластин для кращої продуктивності. Таку конструкцію легше виготовляти, що дало змогу започаткувати її серійне виробництво [21].

У середині XX століття зростала потреба в надійних та невеликих батареях для живлення різних електронних пристроїв [13]. Канадський хімік Льюїс Уррі розробив лужну батарею, щоб задовольнити цей попит і у 1959 році отримав патент на лужно-марганцево-діоксидну батарею. На той час такі батареї випускала фірма, яка сьогодні відома як Energizer [8]. Це дало поштовх до розвитку портативної електроніки, яка популярна і зараз.

Друга половина ХХ століття була насичена різними спробами розвитку в напрямку легких та ємнісних акумуляторів (таблиця 1.1 створена на основі статті [15]).

Таблиця 1.1 – Історія розвитку акумуляторів на основі літію: ключові етапи та учасники створення літій-іонної батареї

Рік	Автор	Основний внесок	Проблеми	Результат
1972	M. Stanley Whittingham (Exxon)	розробка першої літієвої батареї з $TiS_2$ катодом і Li анодом	нестабільність літію, слабкий ринок, відсутність інтересу	перша літієва міжшарова батарея
1980	John B. Goodenough (Oxford)	замінив $TiS_2$ на $LiCoO_2$ , підвищив напругу до 4 В	відсутність партнерів, університет не профінансував патент	запатентовано через AERE, сам не отримав прибутку
1985-1986	Akira Yoshino (Asahi Chemical)	використав вуглецевий анод (нафтовий кокс), підвищив безпеку	Asahi не мала досвіду у виробництві батарей	побудовано лабораторний прототип
1986	Battery Engineering (CША)	побудова перших прототипів батарей за замовленням Asahi	секретність, обмежені ресурси	створено 200 дослідних зразків
1987	Isao Kuribayashi (Asahi)	представив батарею Sony	Asahi відмовилась від партнерства	Sony надихнулась на власну розробку
1987–1989	Sony	комерціалізувала технологію: удосконалення конструкції батареї	патент ще не ліцензовано, багато роботи над масштабуванням	розробка товарної літій-іонної батареї
1989	Sony & AERE (Велика Британія)	Sony ліцензує патент Goodenough через AERE	AERE спершу не усвідомила цінності патенту	укладено ліцензійну угоду
1991	Sony	випуск першої комерційної літій-іонної батареї	–	старт ери літій-іонних акумуляторів
1993	Asahi Chemical + Toshiba	почали виробництво батарей	затримка через нерішучість керівництва	вхід до ринку літій-іонних батарей
2019	Whittingham, Goodenough, Yoshino	отримали Нобелівську премію з хімії	–	–

Як бачимо з таблиці 1.1, в 1991 році фірма Sony випустила перший комерційний літій-іонний акумулятор. Це стимулювало розвиток портативної електроніки, адже цей вид акумулятора має високу енергетичну щільність, для

нього характерна відсутність ефекту пам'яті. Спочатку такі акумулятори використовувалися у відеокамерах, потім в мобільних телефонах [15].

«Згідно з даними, у 1999 році продажі свинцево-кислотних батарей становили близько 40-50 % від загального обсягу продажів батарей у світі, що свідчить про їхню популярність на ринку» [18]. Достатньо популярними вони є й зараз, адже з'являється все більше смарт-пристроїв, більшість з яких потребує автономного живлення.

### 1.1.1 Типи акумуляторних батарей

Для різних потреб існують різні види акумуляторів [26] (таблиця 1.2).

Таблиця 1.2 – Порівняльний аналіз акумуляторних батарей різних типів

Тип батареї	Склад	Напруга (В)	Щільність енергії (Вт·год/кг)	Переваги	Недоліки	Основні сфери застосування
свинцево-кислотна	свинець та сірчана кислота	2,0	30-50	низька вартість, простота обслуговування, надійність	велика вага, обмежений термін служби, токсичність	автомобілі, резервне живлення, телекомунікації
нікель-кадмієва (Ni-Cd)	нікель оксид та кадмій	1,2	40-60	стійкість до глибокого розряду, тривалий термін експлуатації	ефект пам'яті, токсичність кадмію	авіація, медична техніка, промисловість
нікель-металгідридна (Ni-MH)	нікель оксид та металгідрид	1,2	60-120	вища ємність, менша токсичність порівняно з Ni-Cd	схильність до саморозряду, обмежена кількість циклів	побутова електроніка, портативні пристрої
літій-іонна (Li-ion)	літій-кобальт-оксид / графіт	3,6–3,7	150-250	висока щільність енергії, відсутність ефекту пам'яті	вибухонебезпечність при пошкодженні, висока вартість	смартфони, ноутбуки, електромобілі
літій-залізо-фосфатна (LiFePO <sub>4</sub> )	літій-залізо-фосфат	3,2	90-160	термостійкість, безпечність, тривалий ресурс	нижча енергетична щільність ніж в Li-ion	електротранспорт, системи накопичення енергії

Як бачимо, в таблиці 1.2 представлено порівняльний аналіз п'яти основних типів акумуляторних батарей, що широко використовуються в сучасних технологіях. Порівняння здійснюється за такими критеріями: хімічний склад, номінальна напруга, щільність енергії, переваги, недоліки та сфери застосування.

Особливу увагу приділено технічним характеристикам та практичній ефективності кожного типу. Наприклад, літій-іонні батареї мають найвищу щільність енергії, що робить їх оптимальними для мобільної електроніки, у той час як свинцево-кислотні залишаються популярними завдяки низькій вартості. Літій-залізо-фосфатні акумулятори вирізняються високою термостійкістю та безпечністю, що робить їх привабливими для енергетичних систем.

Таблиця 1.2 дає змогу комплексно оцінити доцільність застосування конкретного типу батарей у різних галузях техніки.

#### 1.1.2 Ключові електричні та експлуатаційні параметри батарей

Різні параметри акумуляторів визначають їх придатність для конкретного застосування. У таблиці 1.3 систематизовано основні електричні та експлуатаційні параметри, до основних належать: номінальна напруга, ємність, струми заряду та розряду, внутрішній опір, а також енергетична щільність, що критично важлива для портативних пристроїв. Параметри кількості циклів, саморозряду та робочої температури визначають довговічність та стабільність батареї в умовах експлуатації.

Таблиця 1.3 – Електричні та експлуатаційні параметри акумуляторних батарей

Параметр	Одиниця вимірювання	Опис
Номінальна напруга	Вольт (В)	Середнє значення напруги при розряді батареї у стандартному режимі
Напруга відсічення	Вольт (В)	Мінімальна напруга, після якої подальший розряд шкідливий або неможливий
Ємність	мА·год / А·год	Кількість електричного заряду, що може бути віддана при повному розряді
Максимальний струм розряду	Ампер (А)	Граничний струм, який можна безпечно знімати з батареї
Максимальний струм заряду	Ампер (А)	Найвищий допустимий струм, яким можна заряджати батарею

Продовження таблиці 1.3

Параметр	Одиниця вимірювання	Опис
Внутрішній опір	Ом	Опір всередині елемента, що впливає на втрати енергії при роботі
Енергетична щільність	Вт·год/кг або Вт·год/л	Відношення запасу енергії до маси або об'єму батареї
Кількість циклів заряду-розряду	Цикли	Кількість повних циклів, які батарея витримує до зниження ємності
Саморозряд	% за місяць	Втрата ємності за час зберігання без навантаження
Температурний діапазон експлуатації	°C	Мінімальні та максимальні температури для нормальної роботи батареї
Температурний діапазон зберігання	°C	Межі температур, при яких батарея може зберігатися без пошкодження
Час заряду	Години (год)	Орієнтовний час, необхідний для повного заряду
Безпека	Якісна характеристика	Наявність захистів: від перенапруги, перегріву, короткого замикання тощо
Розміри та форм-фактор	–	Стандартизовані (AA, 18650, CR2032) або спеціальні
Маса	Грам (г) / Кілограм (кг)	Фізична вага батареї, залежить від типу та ємності
Термін служби	Роки або кількість циклів	Залежить від технології та умов експлуатації

За допомогою таблиці 1.3 можна оцінювати батареї для оптимального вибору для побутового або промислового використання.

Зараз існує велика різноманітність форм та розмірів акумуляторних батарейок. Побутові споживачі зазвичай обирають акумулятор за його типорозміром.

Типорозміри акумуляторних батарей – це стандартні розміри, які використовуються для живлення різних пристроїв. Вони стандартизовані і поділяються на циліндричні та прямокутні, із різними формами.

У таблиці 1.4 наведено поширені типорозміри акумуляторних батарей відповідно до міжнародних стандартів ІЕС (International Electrotechnical Commission) або галузевих позначень. Параметри включають геометричні розміри [25], номінальну напругу, середню ємність, форму, а також типові сфери застосування.

Таблиця 1.4 – Типорозміри акумуляторних батарей та їхні основні характеристики

Типорозмір	Позначення стандарту	Форма та габарити (мм)	Номинальна напруга (В)	Середня ємність (мА·год)	Сфера застосування
AAA	IEC: R03	циліндрична: діаметр 10,5, висота 44,5	1,2 (Ni-MH) / 1,5 (Alkaline)	600-1200	пульти, годинники, портативна електроніка
AA	IEC: R6	циліндрична: діаметр 14,5, висота 50,5	1,2 / 1,5	1300-3000	фотоапарати, дитячі іграшки, мишки
C	IEC: R14	циліндрична: діаметр 26,2, висота 50	1,2 / 1,5	3000-8000	ліхтарі, музичні пристрої
D	IEC: R20	циліндрична: діаметр 34,2, висота 61,5	1,2 / 1,5	8000-20000	потужні ліхтарі, медичні пристрої
9V	IEC: 6F22	прямокутний паралелепіпед: 26,5 на 17,5 на 48,5	9	120-600	датчики диму, радіопристрої
CR2032	IEC: CR2032	монета: діаметр 20, товщина 3,2	3	200-240	материнські плати, ваги, брелоки
18650	Li-ion (форм-фактор)	циліндрична: діаметр 18, висота 65	3,6-3,7	1800-3500	павербанки, ноутбуки, електровелосипеди
21700	Li-ion (форм-фактор)	циліндрична: діаметр 21, висота 70	3,6-3,7	4000-5000	електромобілі, системи накопичення енергії
Pouch (гнучкий)	індивідуальні розміри	плоска, залежить від виробника	3,6-3,7	до 10000+	смартфони, планшети, дрони
Prismatic	стандарт залежно від серії	залежить від серії	3,2-3,7	до 20000+	електротранспорт, системи зберігання енергії

На основі таблиці 1.4 можемо зробити висновок, що типорозміри AAA, AA, C і D широко використовуються у побуті і належать до традиційних циліндричних елементів. Формати 18650 і 21700 мають літій-іонні батареї високої енергетичної щільності, які застосовуються в потужних портативних пристроях та електротранспорті. Пакетні (pouch) і призматичні акумулятори

мають нестандартні розміри, вони використовуються у виробках, де важливими є компактність, легкість та висока енергетична щільність.

### 1.1.3 Основні поняття та визначення в контексті моніторингу

Досить часто від ефективної оцінки стану батареї залежить ефективність та безпека тої чи іншої системи. Моніторинг заряду акумулятора є дуже важливим у таких галузях, як електротранспорт, альтернативна енергетика, побутова електроніка, системи IoT, тощо.

Моніторинг заряду – це процес вимірювання, оцінки та прогнозування рівня заряду, стану здоров'я батареї та інших параметрів для забезпечення ефективного і безпечного використання джерела живлення. В таблиці 1.5 наведено основні терміни та визначення, які використовують у цьому контексті.

Таблиця 1.5 – Основні терміни та визначення, які використовуються для моніторингу стану заряду акумуляторної батареї

Термін	Визначення	Приклад
State of Charge (SoC)	поточний заряд батареї як відсоток від повної ємності	SoC = 80 % означає, що батарея заряджена на 80 %
State of Health (SoH)	відношення залишкової ємності до нової	SoH = 90 % означає, що батарея втратила 10 % від початкової ємності
Depth of Discharge (DoD)	величина розряду в процентах від повної ємності	DoD = 40 % означає, що батарея розряджена на 40 %
C-rate	відносна швидкість заряджання/розряджання	1C для 2000 mAh = 2 A
Open Circuit Voltage (OCV)	напруга без навантаження	Li-ion при 100 % $\approx$ 4,2 В
Terminal Voltage	напруга під навантаженням	Нижча за OCV через внутрішній опір
Internal Resistance	внутрішній електричний опір батареї	впливає на нагрівання
Cycle Count	кількість завершених циклів заряд-розряд	500 циклів – типове значення для Li-ion
Battery Management System (BMS)	система керування батареєю	відповідає за балансування, захист, оцінку SoC/SoH

Методи оцінки SoC:

– вольтметричний метод базується на вимірюванні напруги холостого ходу, використовується для батарей із добре вираженою залежністю напруга-заряд, чутливий до впливу температури та струму;

– кулонметричний метод передбачає обчислення заряду шляхом інтегрування струму за формулою (1.1):

$$Q(t) = Q_0 - \int_0^t I(t) dt \quad (1.1)$$

За формулою (1.1) визначається кількість електрики, яку потрібно витратити на окислення або електролітичне відновлення акумулятора. Цей метод достатньо точний при калібруванні, але з часом накопичується похибка;

– метод імпедансної спектроскопії використовує частотні характеристики батареї (вимірюється змінний опір батареї на різних частотах). Метод дає високу точність, але його технічно складно виконувати;

– алгоритмічні методи (наприклад, Фільтр Калмана, нейромережі, моделі) забезпечують точнішу оцінку SoC та SoH у реальному часі за рахунок комбінування методів, згаданих вище, та математичних моделей. Саме такі методи використовуються в сучасних BMS для електромобілів (наприклад, Tesla, BYD).

Приклад роботи системи моніторингу (BMS) наведено на рисунку 1.1.

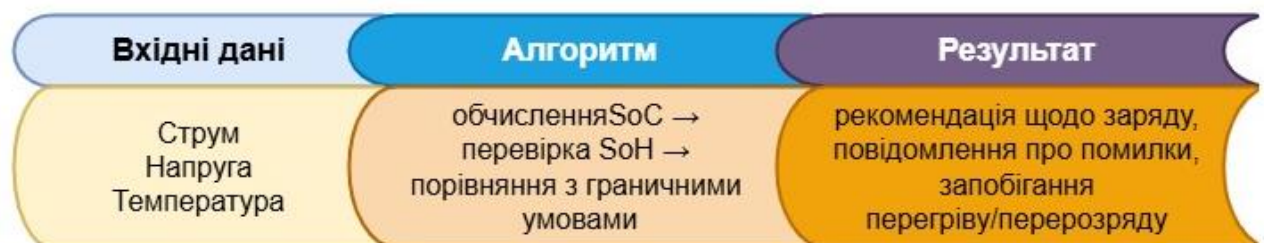


Рисунок 1.1 – Типові функції системи моніторингу стану батареї

## 1.2 Проблеми та виклики багатоканального моніторингу

Багатоканальний моніторинг – це контроль параметрів кількох елементів батарейного блока одночасно. Він використовується в електромобілях, дронах,

накопичувачах енергії, медичному обладнанні, але є багато моментів, які потребують вирішення:

- коли дуже багато елементів у батарейному модулі, то ускладнюється підключення, синхронізація та обробка даних;
- при масштабуванні виникає потреба у модульній архітектурі з локальними контролерами (наприклад, Smart BMS);
- у системах з великою кількістю дротів та каналів можуть виникати електромагнітні завади;
- потрібне екранування, фільтрація сигналів, захист від імпульсних перешкод;
- одночасне зчитування параметрів (напруга, струм, температура) з усіх каналів потребує точного таймування;
- невчасна або асинхронна вибірка призводить до хибних оцінок SoC/SoH;
- деякі алгоритми оцінки SoC/SoH (наприклад, на основі фільтру Калмана або із застосуванням штучного інтелекту) потребують великої обчислювальної потужності, особливо в реальному часі;
- великі обсяги передачі даних від багатьох елементів створюють навантаження на шину зв'язку (CAN, UART, SPI) і сховища;
- ідентифікація зіпсованого або відхиленого елемента (наприклад, з підвищеним внутрішнім опором) у великому масиві даних – складне завдання;
- постійне вимірювання, передача та обробка даних можуть споживати значну частину енергії батареї;
- потрібні енергоефективні контролери й сенсори;
- додаткові електронні модулі можуть призводити до локального нагріву, що небажано для елементів батареї.
- в умовах вібрації, вологи та температурних змін (наприклад, у дроні чи транспорті) можуть виникати проблеми з контактами;
- у багатоканальних системах з ієрархічною структурою можливі затримки в передачі даних або втрати пакетів;

– калібрування кожного каналу потребує часу та спеціального обладнання.

### 1.3 Огляд існуючих рішень

На ринку є готові пристрої для контролю заряду акумулятора. Одним з таких прикладів є Монітор BMV-700 [24], який рахує витрачені ампер-години та оцінює стан заряду акумулятора. Перевагою його є здатність вимірювати напругу, струм розряду, потужність і стан зарядженості акумулятора, під'єднуватися до гаджетів через Bluetooth (рисунок 1.2). Недоліки – одноканальність, тобто здатність працювати лише з одним акумулятором; висока ціна (близько 7000 грн станом на весну 2025 року) як для побутового використання. Діапазон вимірювання напруги – 6,5-95 Вольт, струму – до 500 А, точність вимірювання 10 мА для струму та 0,01 В для напруги. Це дає можливість використання для досить потужних акумуляторів.



Рисунок 1.2 – Монітор BMV-700 [24]

Подібний за функціоналом пристрій – Renogy 500A Battery Monitor (рисунок 1.3), проте він в 2,5 рази дешевший (вартує приблизно 90 доларів США), має дисплей, здатний вимірювати потужність, ємність, залишок заряду; діапазон напруг – від 10 до 120 В, струму – до 500 А [1].

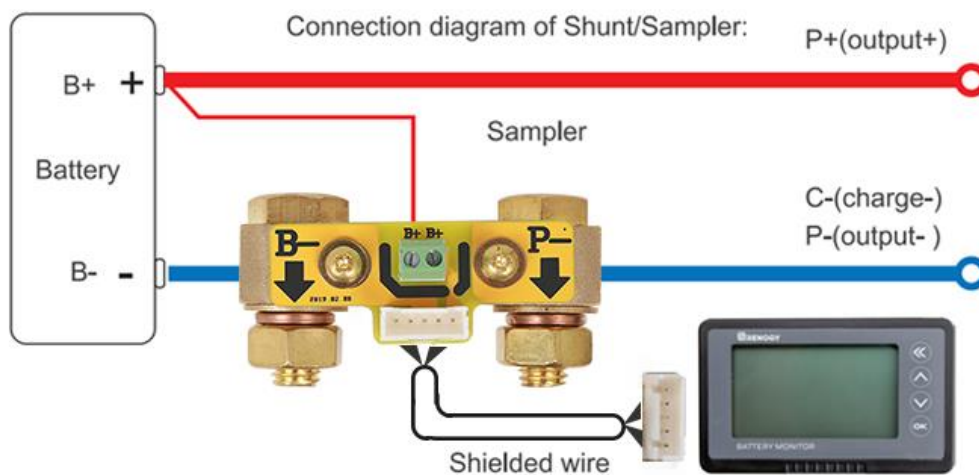


Рисунок 1.3 – 500A Battery Monitor With Shunt [1]

Компанія Arbin [5] пропонує різні пристрої та лабораторні установки, які здатні досить точно визначати характеристики акумуляторів та працюють в багатоканальному режимі (рисунок 1.4). Цей пристрій дає змогу протестувати та порівняти 8 акумуляторів, швидко вимірює струм саморозряду акумулятора, має високу точність (приблизно 0,02 %).



Рисунок 1.4 – Паралельний диференціальний тестер акумуляторів з інтегрованою камерою [5]

Трохи більш ніж на 1000 грн [19] (станом на початок 2025 року) можна купити кулонометр (рисунок 1.5) – індикатор ємності та рівня заряду, який використовують як вбудований елемент і різні системи (в сонячних електростанціях, електромобілях, тощо) для моніторингу стану заряду акумулятора.



Рисунок 1.5 – Вимірювач ємності та рівня заряду [19]

У праці [7] розглядається BMS (система керування акумуляторним блоком), особлива увага приділена алгоритмам балансування комірок та методам оптимізації, виділено важливі параметри, які залучені до процесу балансування комірок. Ці ж параметри підлягають моніторингу.

Стаття [4] присвячена акумуляторним блокам в транспорті (електромобілі, БПЛА, тощо), запропоновано цифровий двійник батареї, який точно відображає зміну стану заряду (SOC) та працездатності (SOH) в режимі реального часу. Модель SOH оцінює втрату ємності, а модель SOC враховує вплив старіння і динамічно перенавчається. Ефективність підходу перевірено на відкритому наборі даних.

У роботі [9] описується система моніторингу, створена на основі методів IoT, яка відповідає за збір даних з різних датчиків, встановлених в електромобілях, для моніторингу стану акумуляторів, оскільки їх енергетична ємність з часом зменшується, що в результаті впливає на експлуатаційні характеристики транспортного засобу.

Використання мікроконтролера STM32 для оцінки стану заряду акумулятора італійськими та португальськими вченими описано в праці [3]. Автори запропонували алгоритм на основі використання фільтру Калмана, використавши мову програмування Python.

Як бачимо, існують дорогі спеціалізовані системи моніторингу стану акумуляторів, які призначені для лабораторних досліджень чи промислового використання або ж інтегровані в спецобладнання (наприклад, електромобілі), тому створення недорогого, але функціонального пристрою для моніторингу стану акумулятора в побутових умовах є актуальним завданням. Крім того, на відміну від закритих систем, власна розробка дасть можливість в майбутньому не лише фіксувати, але й зберігати дані про стан баратей для подальшого їх використання та аналізу.

## РОЗДІЛ 2

### РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ СИСТЕМИ БАГАТОКАНАЛЬНОГО МОНІТОРИНГУ

#### 2.1 Обґрунтування вибору мікроконтролера та його характеристики

Оскільки ми не плануємо використовувати дисплей для відображення стану заряду акумуляторів, а передавати дані по мережі, то нам потрібен або мікроконтролер з вбудованим Wi-Fi модулем або ж додатково доєднувати такий модуль в проєкт. Також нам потрібні аналого-цифрові перетворювачі для кожного каналу. Тому такі мікроконтролери як Raspberry Pi Pico [10], STM32F103 [12] та Arduino Uno [14], нам не підійдуть, оскільки в них немає вбудованого Wi-Fi. Найкращим вибором буде мікроконтролер від фірми espressif [16].

Тому для нашого проєкту було обрано налагоджувальний модуль ESP32 Dev Module, який є комплексним рішенням на основі мікромодуля ESP-WROOM-32. Він добре підходить для вирішення нашої задачі та забезпечує великий запас в напрямку подальшого розвитку та модернізації проєктованої системи.

ESP32 є наступним поколінням після популярного ESP8266 і значно перевершує його за більшістю параметрів. Основні характеристики двох модулів від фірми espressif [16] та їх порівняння наведено в таблиці 2.1.

Таблиця 2.1 – Порівняння популярних мікроконтролерів від фірми ESP

Характеристика	ESP8266MOD (на базі ESP8266)	ESP-WROOM-32 (на базі ESP32)	Примітки
ядра процесора	одноядерний	двоядерний	ESP32 краще для багатозадачності
архітектура	32-біт Tensilica LX106	32-біт Tensilica LX6	ESP32 має потужніший процесор
тактова частота	до 160 МГц (зазвичай 80)	до 240 МГц	ESP32 швидший
Wi-Fi	802.11 b/g/n	802.11 b/g/n	є в обох
Bluetooth	ні	так (Classic + BLE)	велика перевага ESP32

Продовження таблиці 2.1

Характеристика	ESP8266MOD (на базі ESP8266)	ESP-WROOM-32 (на базі ESP32)	Примітки
GPIO піни	менше (близько 10-12 доступно)	більше (близько 30-36 доступно)	більше гнучкості з ESP32
аналогові входи (ADC)	1 (досить обмежений)	багато (10-16 каналів)	ESP32 набагато кращий для аналогових сигналів
аналогові виходи (DAC)	немає	2 канали	можливість генерації аналогових сигналів в ESP32
сенсорні піни	немає	є	для ємнісних сенсорних кнопок/поверхонь
датчик Холла	немає	є	для вимірювання магнітного поля
периферія	SPI, I2C, UART, PWM (програмний)	SPI, I2C, UART, CAN, Ethernet MAC, ШІМ (апаратний), I2S	ESP32 має значно багатший набір периферії
оперативна пам'ять (RAM)	менше	більше	ESP32 може виконувати складніші програми
енергоспоживання	зазвичай менше в активному режимі Wi-Fi	вище в активному режимі, але краще керування живленням та низьке споживання BLE	залежить від режиму роботи
ціна	дешевший	дорожчий	ESP8266 більш бюджетний
спільнота	велика, давно існує	дуже велика, швидко зростає	обидва мають чудову підтримку

Як бачимо з таблиці 2.1, єдиною перевагою ESP8266 є його ціна (ще, можливо, нижче енергоспоживання, залежно від режиму роботи, але з огляду на додаткові ускладнення в архітектурі проєкту, ця перевага є сумнівною).

На вибір для нашого проєкту саме ESP-WROOM-32 (ESP32) вплинули наступні факти та функціональні задачі:

- потрібна більша обчислювальна потужність та можливість виконувати кілька завдань одночасно (багатозадачність + двоядерний процесор);
- потрібні аналогові входи/виходи ADC – більше одного, оскільки ми проєктуємо багатоканальну систему;

- потрібно мати можливість одночасно опрацювати дані ADC та передавати їх по WiFi;
- потрібно більше оперативної пам'яті для опрацювання отриманих даних у вигляді багатовимірних масивів;
- проєкт має потенціал до розширення та досить велику складність;
- бюджет не дуже відрізняється для обох модулів, якщо врахувати вартість усієї системи, а не лише мікроконтролера.

Отже, ESP-WROOM-32 (на базі ESP32) (рисунок 2.1) є значно потужнішим і функціональнішим модулем, що пропонує більше можливостей та в майбутньому дозволить підключити більше периферії. Він краще підходить для нашого багатofункціонального проєкту.

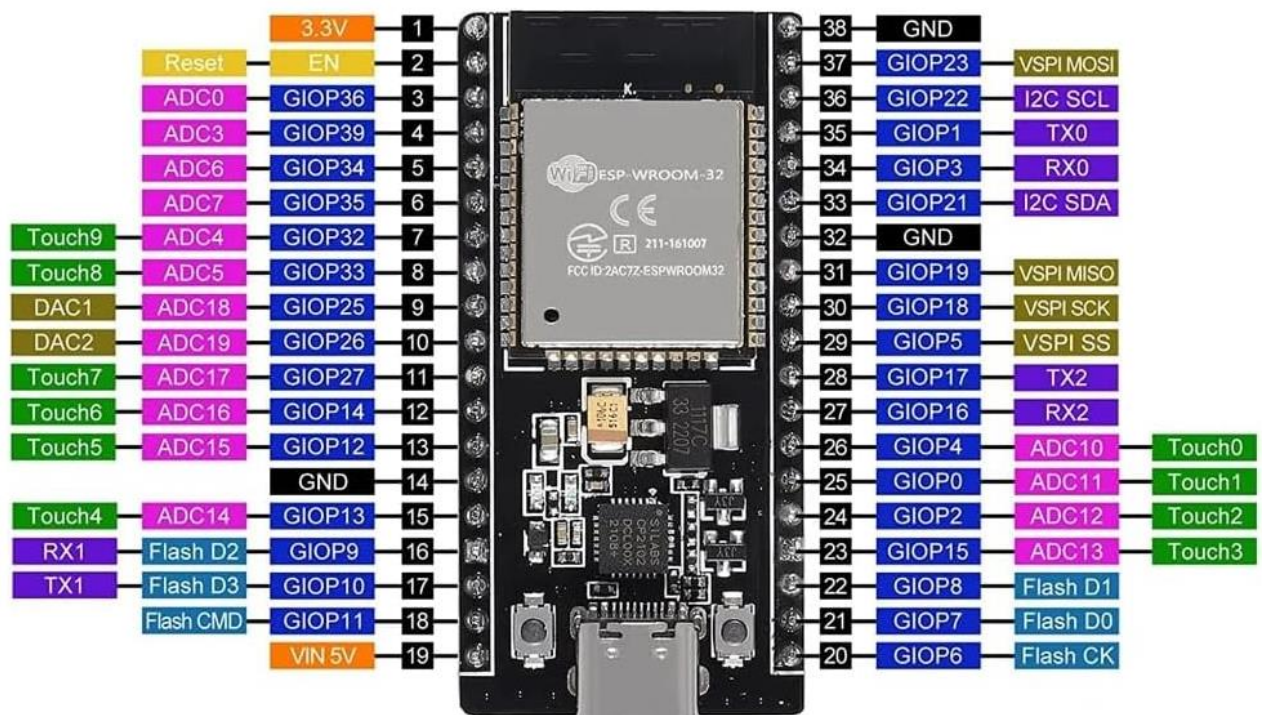


Рисунок 2.1 – Налагоджувальна плата ESP32 Dev Module [6]

Як бачимо з рисунку 2.1, плата має 38 контактів, серед яких є контакти живлення (VIN, 3V3 та GND), контакти зв'язку, контакти загального

призначення (GPIO) та спеціальні функціональні контакти [2]. Основні функції контактів наведено в таблиці 2.2.

Таблиця 2.2 – Основні функції пінів ESP32 Dev Module

Назва контакту	Призначення/Роль	Примітки
3V3	вихід 3,3 В	напруга живлення ESP32
GND	заземлення	
EN	Reset (Enable)	перезапуск ESP32
IO0	GPIO / Flash Mode	під час завантаження прошивки
IO2	GPIO / LED / Boot-пін	може використовуватись для LED
IO34–39	вхідні GPIO (тільки INPUT), ADC1	не мають можливості виводу
IO32–33	GPIO / ADC1	можна використовувати як АЦП
IO25–27	GPIO / DAC / ADC1	25 і 26 – також ЦАП виходи
IO4–5, 12–15	GPIO / HSPI / ADC2	використовуються для SPI
IO16–17	GPIO / UART / I2C	можуть працювати як UART
IO21, 22	стандартні GPIO / I2C SDA/SCL (типово)	часто використовуються як I2C
IO23, 19, 18	GPIO / SPI	часто використовуються як SPI
TX0 / RX0	UART0 (Serial)	для зв'язку з комп'ютером
VIN	вхід живлення 5 В	через USB або зовнішнє джерело

GPIO 6-11 використовуються для внутрішнього зв'язку з флеш-пам'яттю, тому не варто використовувати ці пини в проекті, також пини GPIO 0, GPIO 2, GPIO 12 варто обережно використовувати (краще не використовувати, оскільки вони відповідають за завантаження прошивки), також варто враховувати що контакти GPIO 34-39 працюють лише на вхід. Також деякі пини, окрім базового вводу/виводу, забезпечують спеціальні функції та надають додаткові можливості (таблиця 2.3).

Таблиця 2.3 – Функціональні пини ESP32 Dev Module

Контакти	Функція
GPIO 32-39	ADC1 – аналого-цифровий перетворювач
GPIO 0, 2, 4, 12-15	ADC2 – аналого-цифровий перетворювач
GPIO 25, 26	DAC – цифро-аналоговий перетворювач
GPIO 1 (TX), 3 (RX)	UART0 – послідовний інтерфейс зв'язку для обміну даними
GPIO 9, 10	UART1 – послідовний інтерфейс зв'язку для обміну даними
GPIO 16, 17	UART2 – послідовний інтерфейс зв'язку для обміну даними
GPIO 21 (SDA), 22 (SCL)	I2C (як правило) – послідовна шина зв'язку ( )
GPIO 18 (SCK), 19 (MISO), 23 (MOSI), 5 (CS)	SPI (як правило) – послідовний периферійний інтерфейс

## 2.2 Аналогово-цифровий перетворювач

Один із важливих модулів мікроконтролера ESP-WROOM-32 є аналого-цифровий перетворювач (АЦП). Оскільки ми вибрали плату на базі ESP32, то вона має два основних канали АЦП, які забезпечують перетворення аналогових у цифрові сигнали. ESP32 має 18 вхідних каналів АЦП, кожен з яких має 12-бітну роздільну здатність, що є значною модернізацією порівняно з одним 10-бітним каналом АЦП ESP8266. Ці GPIO служать входами АЦП зі своїми відповідними каналами.

Як бачимо з таблиці 2.3, піни GPIO 32-39 (АЦП1) та GPIO 0, 2, 4, 12-15 (АЦП2) використовуються в ролі АЦП. Проте нам варто пам'ятати про важливість Wi-Fi-підключення в даному проєкті, тому АЦП2 використовувати не варто. Отже, вимірювання будуть побудовані на основі АЦП1.

Вхідні канали АЦП ESP32 мають 12-бітну роздільну здатність, що дозволяє аналогове зчитування від 0 до 4095. Тут 0 відповідає 0 В, а 4095 – 3,3 В. За потреби є можливість налаштування роздільної здатності у інших діапазонах напруг АЦП. Варто враховувати, що вбудований АЦП на ESP32 не має лінійної поведінки при зчитуванні вхідного сигналу (рисунок 2.2).



Рисунок 2.2 – Нелінійність АЦП на краях діапазону напруги [2]

Як бачимо з рисунку 2.2, є невеликі перепади напруги в діапазонах 0-0,1 В та між 3,2-3,3 В, тобто відбувається «загинання» характеристики АЦП на краях діапазону напруги. Цю нелінійність потрібно враховувати при використанні контактів АЦП.

Схема підключення АЦП1 до комірок батареї наведена на рисунку 2.3.

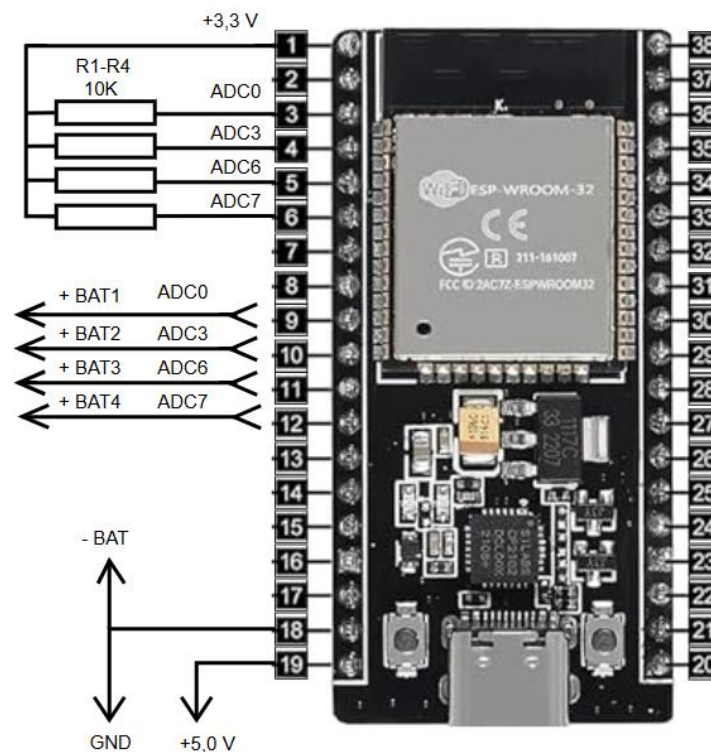


Рисунок 2.3 – Схема підключення АЦП до комірок батареї

Як було зазначено вище, канали АЦП2 (GPIO 4, 2, 15, 13, 12, 14, 27, 25, 26) використовуються модулем Wi-Fi або Bluetooth, тому зчитування з АЦП2 може бути нестабільним, або взагалі не працювати під час роботи радіомодуля. З метою уникнення конфлікту з Wi-Fi використано канали АЦП1.

Відповідність контактів:

- ADC0 відповідає GPIO 36;
- ADC3 відповідає GPIO 39;
- ADC6 відповідає GPIO 34;
- ADC7 відповідає GPIO 35.

ESP32 АЦП має максимальний вхідний діапазон близько 3,3 В (залежить від  $V_{ref}$ ). За замовчуванням, функція `analogRead()` для ESP32 повертає значення від 0 до 4095 (12-бітний АЦП).

Поки не будемо використовувати атенюацію, щоб змінити діапазон вхідної напруги. Використаємо стандарт по замовчуванню:

- ADC\_0db: ~ 0-1,1 В;
- ADC\_2\_5db: ~ 0-1,5 В;
- ADC\_6db: ~ 0-2,2 В;
- ADC\_11db (за замовчуванням): ~ 0-3,3 В.

### 2.3 Монтаж пристрою

Сконструюємо пристрій для моніторингу одночасно чотирьох акумуляторів типу АА або ААА. Наша система буде вимірювати напругу кожного акумулятора окремо. Оскільки максимальна напруга зарядженого NiMH/NiCd акумулятора типу АА/ААА становить близько 1,4-1,5 В, а АЦП на ESP32 розрахований на вхідну напругу до 3,3 В, нам знадобляться дільники напруги для кожного елемента (рисунок 2.4). Це дозволить безпечно подати вимірювану напругу на АЦП-входи ESP32, не перевищуючи їх допустимий рівень.

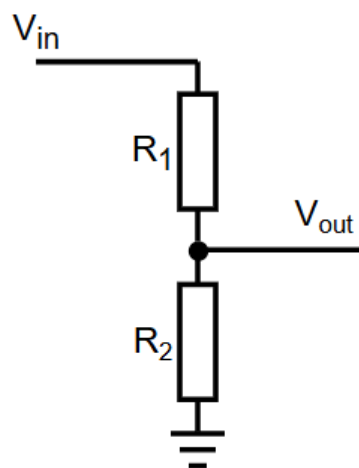


Рисунок 2.4 – Дільник напруги на резисторах

Вихідна напруга розраховується за формулою (2.1):

$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in}, \quad (2.1)$$

де  $V_{in}$  – вхідна напруга;

$V_{out}$  – вихідна напруга;

$R_1$  та  $R_2$  – опори резисторів.

Отже, дільник напруги складається з двох резисторів  $R_1$  та  $R_2$ , додатній вивід акумулятора підключається до  $R_1$ , а від’ємний – до  $R_2$  та загальної землі (GND). Точка з’єднання  $R_1$  та  $R_2$  підключається до відповідного АЦП-контакту ESP32.

Нехай ми будемо моніторити 4 NiMH/NiCd акумулятори типу AA або AAA, кожен з них має номінальну напругу 1,2 В і повністю заряджений до 1,45 В. Максимальна сумарна напруга послідовно з’єднаних 4 таких акумуляторів буде становити:  $4 \times 1,45 \text{ В} = 5,8 \text{ В}$ . Тому для оцінки загальної напруги потрібен дільник.

Основні компоненти проекту:

- модуль ESP-WROOM-32 – ядро системи;
- дільники напруги для кожного акумулятора (наприклад, по два резистори);
- тримач для 4-х акумуляторів, з виводами для кожного окремого елемента;
- з’єднувальні дроти;
- діоди для індикації;
- джерело живлення для ESP32: Micro USB кабель та 5 В блок живлення або стабілізований блок живлення на 3,3 В.

Принципова схема пристрою зображена на рисунку 2.5, а послідовно кроки монтажу всіх елементів показано на рисунках 2.6-2.7.





Рисунок 2.7 – Монтаж пристрою

Як видно з рисунків 2.6-2.7, для пристрою було використано корпус від неробочого блоку зарядки, в якому вже є готові місця для розміщення акумуляторів з відповідними контактами. На задній стінці корпусу було розміщено мікроконтролер, який «втопили» в стінку щоб уникнути пошкодження контактів, всі з'єднання розмістили всередині корпусу.

## РОЗДІЛ 3

### РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ МОНІТОРИНГУ

#### 3.1 Початкові налаштування середовища програмування

Для програмування пристрою ми обрали середовище Arduino IDE [11]. Оскільки IDE призначене для програмування плат фірми Arduino, то нам потрібно його адаптувати до нашого проекту – здійснити налаштування, які є специфічними для обраного апаратного модуля ESP32 Dev Module. Тому нам потрібно додати посилання на менеджер плат ESP32 та встановити відповідний пакет. Для цього потрібно зайти в налаштування (меню «Файл») та в полі «Додаткові URL-адреси менеджера плат» додати посилання: [https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json) (рисунок 3.1). Там вже було посилання на ESP8266, тому наше нове посилання додали через кому і з нового рядка.

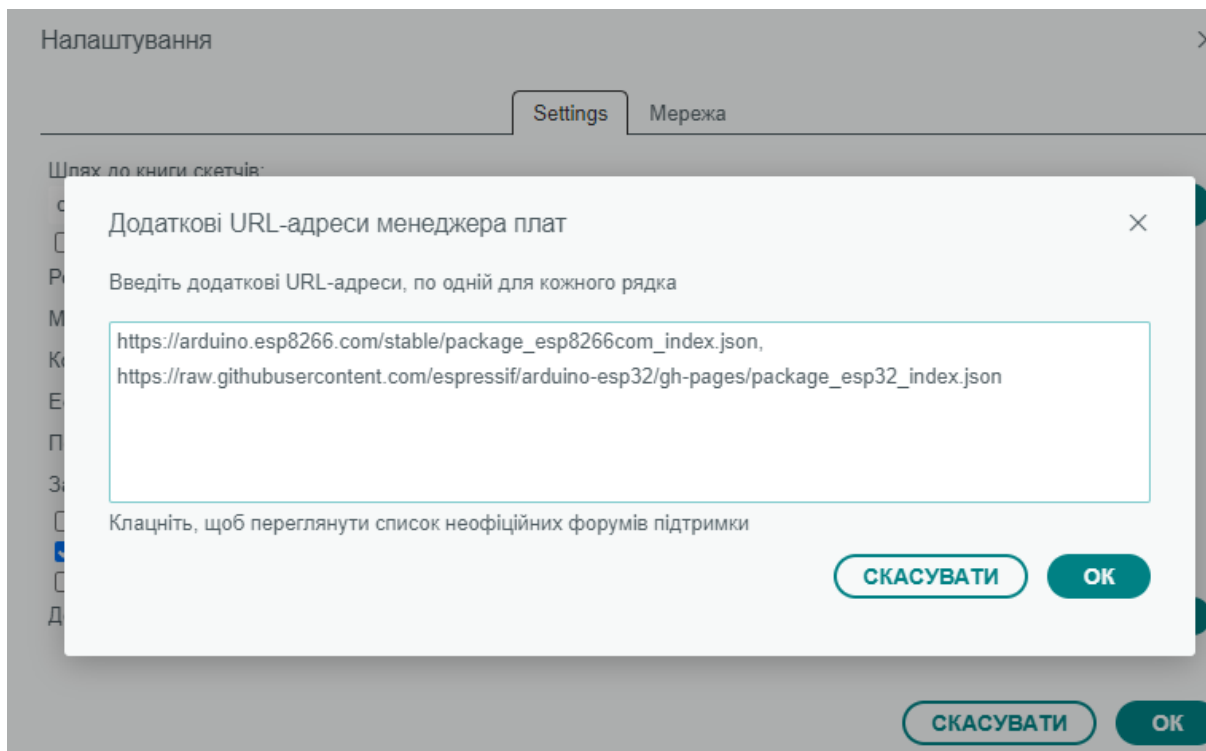


Рисунок 3.1 – Налаштування посилань на конфігуратор ESP32

Далі потрібно відкрити менеджер плат (рисунок 3.2), в поле пошуку ввести esp32, обрати плату від фірми espressif (рисунок 3.3) та встановити програмне забезпечення для її підтримки в середовищі розробки (рисунок 3.4).

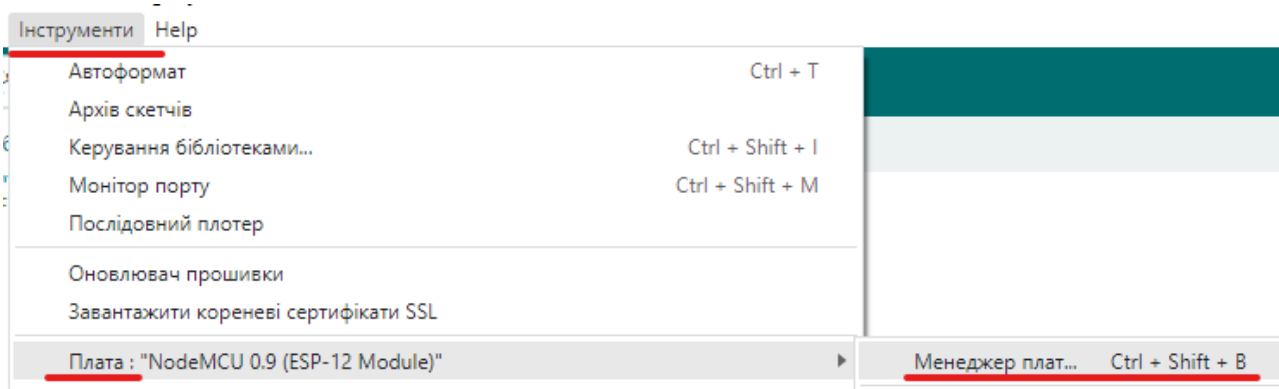


Рисунок 3.2 – Менеджер плат через меню «Інструменти»

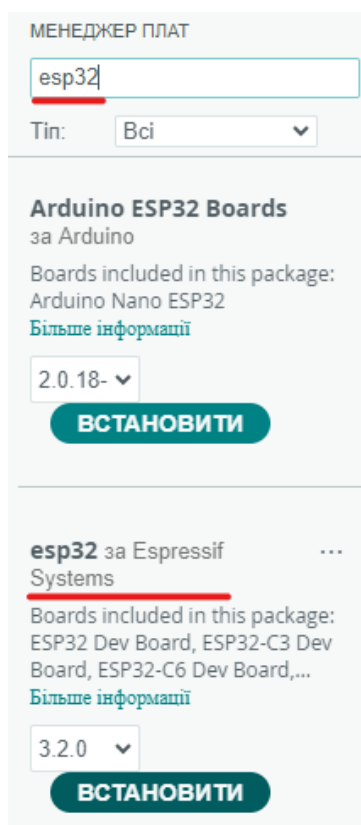


Рисунок 3.3 – Пошук потрібної плати для встановлення



Рисунок 3.4 – Процес встановлення ПЗ для підтримки плати esp32

Після завершення процесу установки в менеджері плат з'явився новий розділ – esp32, в якому ми й обираємо потрібний нам модуль розробки – ESP32 Dev Module (рисунк 3.5).

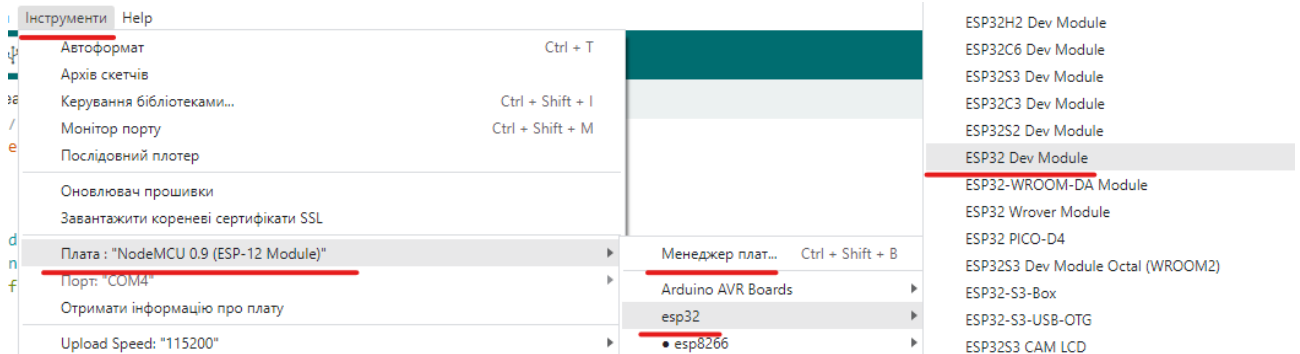


Рисунок 3.5 – Вибір налагоджувальної плати

Тепер середовище Arduino IDE повністю готове для роботи з платами на базі ESP32, зокрема з модулем ESP-WROOM-32. Після фізичного під'єднання плати до ПК, нам потрібно обрати потрібний COM-порт у меню інструменти.

Щоб зчитувати аналогові значення з чотирьох каналів АЦП плати ESP32 Dev Module та виводити чи обробляти їх, нам потрібно використовувати функції `analogRead()` та `analogSetPinAttenuation()`, які будуть доступні тільки після інсталяції відповідних бібліотек (рисунки 3.6-3.7).

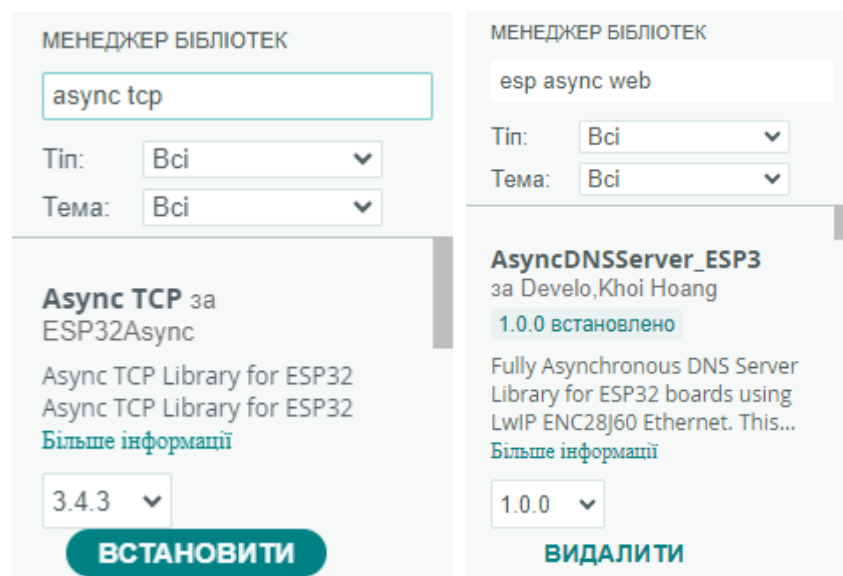


Рисунок 3.6 – Встановлення бібліотек

```
1 // імпортуємо бібліотеки
2 #include <ESPAsyncWebServer.h>
3 #include <AsyncTCP.h>
4 #include <WiFi.h>
```

Рисунок 3.7 – Підключення бібліотек

### 3.2 Налаштування та калібрування АЦП для діапазону напруги

Зробимо калібрування для діапазону 0,5 В-2,0 В, оскільки більшість елементів типу AA та AAA мають саме такий діапазон напруги. Для цього ми можемо використати лінійну апроксимацію (метод двох точок). Це означає, що ми виміряємо відомі напруги у двох точках нашого діапазону (наприклад, 0,5 В і 2,0 В), запишемо відповідні значення АЦП, а потім використаємо їх для розрахунку коефіцієнтів масштабування та зміщення.

Калібрування АЦП на ESP32 є дуже важливим етапом, адже цей процес забезпечує підвищення точності вимірювань напруги. Незважаючи на те, що ESP32 має вбудовану функцію калібрування, додаткове «ручне» лінійне калібрування за кількома точками може значно покращити результати.

На практиці співвідношення між реальною вхідною напругою та «сирим» цифровим значенням, яке видає АЦП, може бути не ідеальним. Отримання калібрувальних точок дозволяє побудувати точнішу залежність між вхідною напругою та «сирим» значенням АЦП, що значно покращує точність вимірювань.

Для отримання калібрувальних точок нам потрібно:

– під'єднати модуль ESP32 і завантажити на нього прошивку (додаток Г), яка дозволить зчитувати «сирі» значення з потрібних ADC-пінів та виводити їх у послідовний монітор. Цей код має просто циклічно зчитувати значення з вибраних АЦП-пінів і друкувати їх;

- мати мультиметр для точного вимірювання напруги, яку ви подаєте на вхід ADC ESP32;

- мати стабільне джерело напруги – лабораторний блок живлення або прецизійний дільник напруги (з використанням відомих резисторів) від стабільного джерела живлення. Чим стабільніша напруга, тим точнішими будуть калібрувальні точки.

Покроковий опис процесу отримання калібрувальних точок (процес збору точок є ітеративним і вимагає точності):

- беремо один з АЦП1 пінів (наприклад, GPIO 36, який відповідає ADC1\_CHANNEL\_0) і приєднуємо до нього дріт (плюс контакт) від нашого стабільного джерела напруги, а його (мінус) – до GND ESP32;

- щоб знайти першу калібрувальну точку (низька напруга), налаштуємо наше стабільне джерело напруги на 0,5 В, одночасно вимірюємо точну напругу на піні GPIO 36 за допомогою мультиметра; записуємо це значення, наприклад,  $Actual\_Voltage\_0\_5V = 0,501\text{ В}$ ; також дивимося в Serial Monitor нашого IDE та записуємо «сире» значення АЦП, яке відображається для GPIO 36. Це буде  $Raw\_Value\_0\_5V = [значення\_з\_консолі]$ . Записуємо пару: (0,501 В; [значення\_з\_консолі]);

- щоб знайти другу калібрувальну точку (висока напруга), налаштуємо наше стабільне джерело напруги на 2,0 В, знову одночасно вимірюємо напругу на піні GPIO 36 за допомогою мультиметра. Записуємо це значення, наприклад,  $Actual\_Voltage\_2\_0V = 2,003\text{ В}$ , зчитуємо та записуємо відповідне «сире» значення АЦП з консолі. Це буде  $Raw\_Value\_2\_0V = [значення\_з\_консолі]$ . Записуємо пару: (2,003 В; [значення\_з\_консолі]);

- повторюємо дії для знаходження двох точок для кожного каналу, щоб отримати максимальну точність для кожного акумулятора окремо.

Маючи дві пари точок ( $Actual\_Voltage\_0\_5V$ ,  $Raw\_Value\_0\_5V$ ) та ( $Actual\_Voltage\_2\_0V$ ,  $Raw\_Value\_2\_0V$ ), ми можемо побудувати лінійну функцію, яка перетворюватиме будь-яке «сире» значення АЦП на відповідну напругу.

Лінійна інтерполяція напруги на основі цих значень здійснюється за формулою (3.1):

$$U = A \cdot Raw\_Value + B, \quad (3.1)$$

де  $A = \frac{Actual\_Voltage\_2\_0V - Actual\_Voltage\_0\_5V}{Raw\_Value\_2\_0V - Raw\_Value\_0\_5V}$  – нахил лінії;

$B = Actual\_Voltage\_2\_0V - A \cdot Raw\_Value\_0\_5V$  – зміщення.

На основі формули (3.1) матимемо частину коду в прошивці (лістинг 3.1).

Лістинг 3.1 – Розрахунок `calibration_scale` та `calibration_offset` для кожного каналу

---

```

calibration_scale = (RealVoltage_2_0V - RealVoltage_0_5V) /
(rawValue_2_0V - rawValue_0_5V)
RealVoltage_2_0V= 2.0 (В)
RealVoltage_0_5V = 0.5 (В)
rawValue_2_0V = значення АЦП, отримане при 2.0 В
rawValue_0_5V = значення АЦП, отримане при 0.5 В
calibration_offset = RealVoltage_0_5V - (calibration_scale *
rawValue_0_5V)

```

---

кінець лістингу 3.1

Для ADC1\_CH0\_PIN ми отримали:

– Raw\_Value\_0\_5V\_CH0 = 620;

– Raw\_Value\_2\_0V\_CH0 = 2480.

Тоді

$$calibration\_scale\_CH0 = \frac{2,0 - 0,5}{2480 - 620} = \frac{1,5}{1860} = 0,00080645,$$

$$calibration\_offset\_CH0 = 0,5 - (0,00080645 \cdot 620) = -0,00002 \approx 0.$$

Потрібно провести аналогічні розрахунки для кожного каналу АЦП. Отримані значення слід вставити у код програми прошивки мікроконтролера.

### 3.3 Створення прошивки мікроконтролера

Прошивка мікроконтролера здійснювалася поетапно: спочатку створено скретч для моніторингу заряду одного акумулятора (додаток В), а після його тестування, код було переписано для моніторингу чотирьох елементів (додаток Г).

Веб-сервер для відображення характеристик (односторінковий) розміщено на самому мікроконтролері. Знаючи його IP-адресу ми можемо за допомогою браузера під'єднатися та отримати значення напруг комірок батареї (рисунок 3.8).

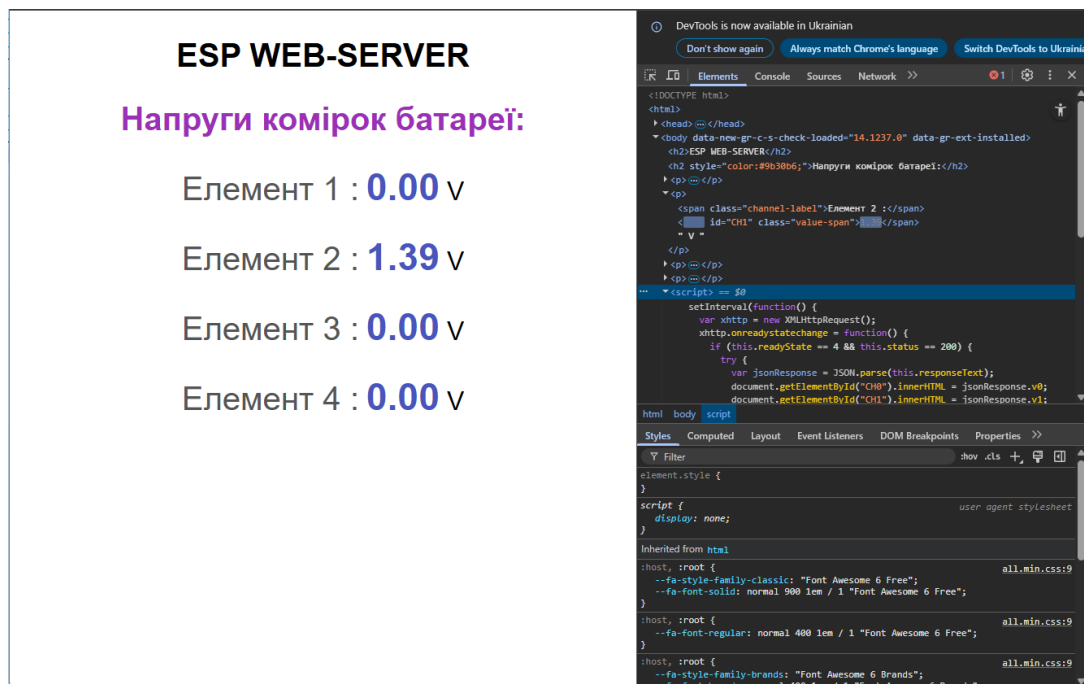


Рисунок 3.8 – Загальний вигляд програми для 4-х каналів вимірювання

Програма зчитує напругу з чотирьох аналогових входів (АЦП1) ESP32, перетворює сирі значення АЦП у вольти та виводить результати у послідовний монітор. Вибрані входні тільки ADC1 GPIO (36-39), що не конфліктують з Wi-Fi.

Програму (додаток В) можна побілити на такі логічні кроки:

- підключення бібліотеки WiFi.h;
- оголошення пінів АЦП1;

- оголошення опорної напруги (VREF);
- налаштування Wi-Fi;
- налаштування атенюації (ADC\_11db дозволяє вимірювати напругу в діапазоні до  $\approx 3,3$  В, що підходить для батарей);
- читання АЦП та обчислення напруги;
- вивід у послідовний порт;
- затримка (1 секунда паузи між вимірюваннями).

Програма, наведена в додатку Г, – це розширена версія попередньої системи зчитування напруги на основі мікроконтролера ESP32, вона орієнтована на багатоканальне вимірювання, калібрування, усереднення та веб-візуалізацію даних. Основні завдання цієї програми:

- використання 4 аналогових каналів (ADC1\_CH0-3) на ESP32: GPIO 36, 39, 35, 34;
- калібрування кожного каналу: для точності використано індивідуальні scale і offset коефіцієнти, розраховані експериментально (в діапазоні 0,5-2,0 В);
- веб-сервер з інтерфейсом містить сторінку з поточними напругами 4 елементів батареї та автоматичне оновлення JSON-даних кожну секунду;
- сумісність із Wi-Fi забезпечує передачу даних по мережі, IP-адреса виводиться у консоль;
- використано асинхронний веб-сервер (ESPAsyncWebServer), що не блокує loop().

Порівняння двох прошивок, описаних вище та наведених в додатках, показано в таблиці 3.1.

Таблиця 3.1 – Порівняння варіантів прошивки

Параметр/Функціональність	Додаток В	Додаток Г
Кількість каналів	1 або 2	4 (ADC1_CH0-CH3)
Калібрування	спільне або грубе	індивідуальне для кожного каналу, для 0,5-2,0 В
Wi-Fi та веб-інтерфейс	відсутній	повноцінна підтримка Wi-Fi + веб-інтерфейс із автооновленням
HTML/JS сторінка	відсутня	стилізований інтерфейс з AJAX та обробкою помилок

## Продовження таблиці 3.1

Параметр/Функціональність	Додаток В	Додаток Г
Асинхронний веб-сервер	ні	так (на основі ESPAsyncWebServer)
Зчитування з АЦП	поодинокі	з потенційним усередненням
Інформація у консоль	може бути обмежена	докладний вивід з каліброваними значеннями
Гнучкість і масштабованість	низька	висока (можна розширити кількість каналів або логіку обробки)
JSON-API для даних	відсутній або примітивний	повноцінна JSON-відповідь для всіх каналів

На основі таблиці 3.1 можемо зробити висновок, що покращена версія прошивки дає такі переваги:

- висока точність завдяки індивідуальному калібруванню;
- зручність моніторингу через веб-інтерфейс без потреби підключатися через UART;
- масштабованість – легко додати нові канали або розширити функціональність;
- асинхронний сервер не блокує виконання інших завдань в loop().

### 3.4 Тестування система багатоканального моніторингу

Пристрій у готовому вигляді зображено на рисунку 3.9.

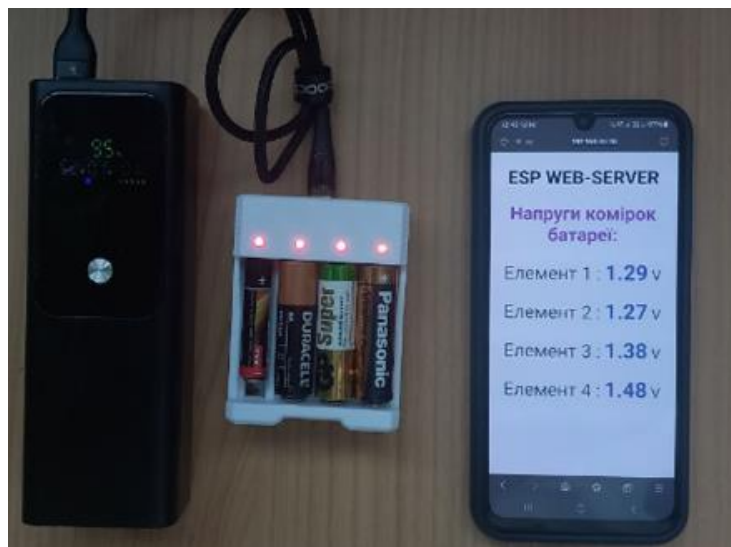


Рисунок 3.9 – Загальний вигляд проєкту

Як бачимо, він працює як з акумуляторами типу АА, так і ААА. Всередині мікроконтролера прошивка містить односторінковий веб-сервер, до якого під'єднано смартфон по мережі Wi-Fi.

Щоб перевірити достовірність отриманих даних було використано універсальний тестер заряду [27]. Він призначений для вимірювання характеристик лише одного елемента, тому ми по черзі вимірювали напругу для кожного акумулятора (рисунок 3.10) та порівнювали її з напругою, яку видав наш багатоканальний пристрій.

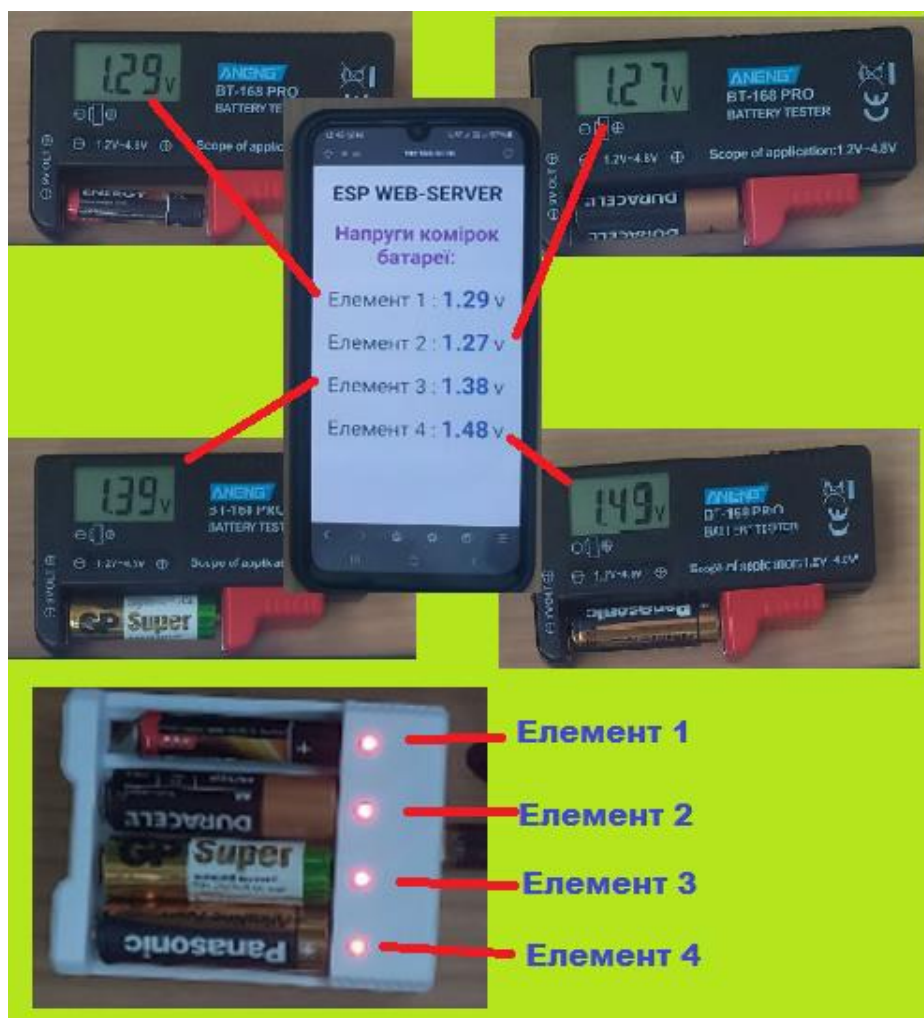


Рисунок 3.10 – Тестування

Як бачимо з рисунку 3.10, точність вимірювання нашого пристрою становить 0,1 В, що є досить непоганим показником для побутового використання.

## ВИСНОВКИ

Під час виконання кваліфікаційної роботи було розроблено систему багатоканального моніторингу акумуляторів, яка дає можливість тестувати одразу до чотирьох елементів типу AA або AAA.

Проаналізовано існуючі підходи до моніторингу заряду акумуляторних батарей, а саме: Монітор BMV-700, Renogy 500A Battery Monitor, паралельний диференціальний тестер акумуляторів з інтегрованою камерою, кулонометр.

Також здійснено огляд сучасних публікацій в напрямку теми кваліфікаційної роботи.

Обґрунтовано вибір мікроконтролера (ESP-WROOM-32) та периферії до нього.

Розроблено електричну принципову схему, обґрунтовано вибір складових, опираючись на закони електротехніки та потрібний функціонал пристрою.

У результаті виконання роботи було реалізовано дві версії програмного забезпечення для мікроконтролера ESP32, призначені для зчитування, калібрування та виведення аналогових сигналів із декількох каналів АЦП – базову та розширену.

Додано веб-сервер, який через інтерфейс HTTP дозволяє переглядати значення напруг у режимі реального часу через браузер. Веб-інтерфейс має простий, інформативний дизайн із відображенням чотирьох каналів у вигляді окремих елементів.

Щоб перевірити достовірність отриманих даних від розробленого пристрою було використано універсальний тестер заряду. В результаті тестування було виявлено відхилення порядку 0,1 В, що є досить непоганим показником для побутового використання.

Використання мікроконтролера ESP-WROOM-32 дає можливість подальшого масштабування проекту в майбутньому.

Розроблений пристрій може використовуватися в побуті для тестування акумуляторів або як лабораторний навчальний стенд.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 500A Battery Monitor With Shunt. *Renogy United States*. URL: <https://surl.li/wsaqnw> (дата звернення: 21.04.2025).
2. A Comprehensive to ESP32 Pinout Diagram. *Xecor*. URL: <https://www.xecor.com/blog/esp32-pinout-diagram> (дата звернення: 01.04.2025).
3. Adaptive Extended Kalman Filtering for Battery State of Charge Estimation on STM32. *arXiv.org*. URL: <https://arxiv.org/abs/2504.05936> (дата звернення: 11.05.2025).
4. Alamin K. S. S., Chen Y., Macii E., Poncino M. and Vinco S. A Machine Learning-based Digital Twin for Electric Vehicle Battery Modeling. *2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, Barcelona, Spain, 2022, pp. 1-6, doi: 10.1109/COINS54846.2022.9854960.
5. Battery Research. *Arbin Instruments*. URL: <https://www.arbin.com/battery-research.html> (дата звернення: 25.04.2025).
6. Hailege 3pcs ESP-WROOM-32 ESP32 CP2012 USB C WiFi + Bluetooth Dual Core 38 Pin Type-C Interface ESP32-DevKitC-32 Development Board Module STA/AP/STA+AP. *Amazon*. URL: <https://www.amazon.de/Hailege-ESP-WROOM-32-Bluetooth-ESP32-DevKitC-32-Development/dp/B0CNYM28CK> (дата звернення: 01.04.2025).
7. Anupama R Itagi, et al. Cell Balancing Paradigms: Advanced Types, Algorithms, and Optimization Frameworks. *arXiv*. 2024. Pp. 1-33. URL: <https://doi.org/10.48550/arXiv.2411.05478> (дата звернення: 25.03.2025).
8. Lewis Urry – A Powerful Man. *News and jobs from the chemical, biochemical and process engineering sectors – The Chemical Engineer*. URL: <https://www.thechemicalengineer.com/features/cewctw-lewis-urry-a-powerful-man/> (дата звернення: 12.03.2025).
9. Nikhil K., Sharma S., Khubalkar P., Daigavane and Vaidya P. «IoT-Enabled Battery Monitoring System for Enhanced Electric Vehicle Performance», *2023 3rd*

*International Conference on Intelligent Technologies (CONIT)*, Hubli, India, 2023, pp. 1-6, doi: 10.1109/CONIT59222.2023.10205665.

10. Pico-series Microcontrollers. *Raspberry Pi*. URL: <https://www.raspberrypi.com/documentation/microcontrollers/pico-series.html> (дата звернення: 21.03.2025).

11. Software. *Arduino – Home*. URL: <https://www.arduino.cc/en/software/> (дата звернення: 15.04.2025).

12. STM32F103 – PDF Documentation. *st.com*. URL: <https://www.st.com/en/microcontrollers-microprocessors/stm32f103/documentation.html> (дата звернення: 15.03.2025).

13. The ultimate guide to battery technology. *Interesting Engineering*. URL: <https://interestingengineering.com/innovation/the-ultimate-guide-to-battery> (дата звернення: 03.04.2025).

14. UNO R3. *Arduino Documentation*. URL: <https://docs.arduino.cc/hardware/uno-rev3/> (дата звернення: 18.03.2025).

15. Who Really Invented the Rechargeable Lithium-Ion Battery? *IEEE spectrum*. URL: <https://spectrum.ieee.org/lithium-ion-battery-2662487214> (дата звернення: 22.04.2025).

16. Wireless SoCs, Software, Cloud and AIoT Solutions. *Espressif Systems*. URL: <https://www.espressif.com/> (дата звернення: 25.03.2025).

17. N. Zdolbitska, M. Delyavskyy, N. Lishchyna, V. Lishchyna, S. Lavrenchuk, V. Sulim. DIY Smart Auxiliary Power Supply for Emergency. *Advances in Artificial Systems for Logistics Engineering III*. 2023. Pp. 382–392. URL: [https://doi.org/10.1007/978-3-031-36115-9\\_35](https://doi.org/10.1007/978-3-031-36115-9_35) (дата звернення: 21.03.2025).

18. Громова В. Історія першої батарейки: винахід, етапи розвитку. *ProIT*. URL: <https://proit.com.ua/news/yak-bula-vynajdena-persha-batarejka/> (дата звернення: 27.03.2025).

19. Індикатор ємності, рівня заряду акумулятора кулонометр. *Doctor Smarts*. URL: <https://doctorsmarts.com.ua/ua/p2043346160-indikator-emkosti-urovnya.html> (дата звернення: 27.04.2025).

20. Кваліфікаційна робота Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти освітньої програми «Комп'ютерна інженерія» галузі знань 12 Інформаційні технології спеціальності 123 Комп'ютерна інженерія денної та заочної форм навчання/ уклад. С.В. Лавренчук, Н.В. Багнюк. Луцьк: ЛНТУ, 2025. 56 с.

21. Коротка історія акумулятора. *DILER – інтернет-магазин*. URL: <https://diler.com.ua/korotka-istoriia-akumuliatora/> (дата звернення: 20.03.2025).

22. Міжнародна науково-практична конференція молодих вчених та студентів «Програмне та апаратне забезпечення в інформаційних технологіях». URL: <https://conference.net.ua/> (дата звернення: 19.05.2025).

23. Місюра Д., Лавренчук С., Кайдик О., Здолбіцька Н. Система моніторингу заряду акумуляторної батареї на основі ESP32 та BMS-контролера. *Програмне та апаратне забезпечення в інформаційних технологіях*: матеріали Міжнар. науково- практ. конф., м. Луцьк, 6 травня 2025 р. м. Луцьк, 2025. С. 123-126.

24. Монітор BMV-700. *Victron Energy*. URL: <https://www.victronenergy.ua/battery-monitors/bmv-700> (дата звернення: 21.04.2025).

25. Таблиця розмірів акумуляторів. *Sungold Solar*. URL: <https://www.sungoldsolar.com/uk/battery-sizes-chart/>(дата звернення: 18.04.2025).

26. Типи акумуляторних батарей: повний огляд. *Компанія DEPS*. URL: <https://deps.ua/ua/knowegable-base/reference-information/10591.html> (дата звернення: 11.03.2025).

27. Універсальний тестер заряду батарей BT-168 Pro AAA AA з цифровим дисплеєм. *Autobaza.com.ua*. URL:<https://autobaza.kiev.ua/ua/p2425596054-universalnyj-tester-zaryada.html> (дата звернення: 02.05.2025).

# ДОДАТКИ

## Додаток А

### Сертифікат про участь в конференції



# СЕРТИФІКАТ

ЗАСВІДЧУЄ, ЩО

*Дмитро МІСЮРА*

БРАВ(ЛА) УЧАСТЬ У

Міжнародній науково-практичній конференції

молодих вчених та студентів

«ПРОГРАМНЕ ТА АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ В ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЯХ»

Голова оргкомітету,  
ректор Луцького національного  
технічного університету



6 травня 2025 р.  
м. Луцьк

Ірина ВАХОВИЧ

## Додаток А

Тези в збірнику конференції

Міністерство освіти і науки України  
 Луцький національний технічний університет (м. Луцьк)  
 Наукове товариство студентів, аспірантів, докторантів та  
 молодих вчених ЛНТУ (м. Луцьк)  
 Люблінська Політехніка (Польща, Люблін)  
 Університету «Дюнаре де Жос» (Румунія, Галац)  
 Університет Коменського (Словаччина, Братислава)  
 Університет Оснабрюк (Німеччина, Оснабрюк)  
 Університет Трансмонтани і Верхнього Дору (Португалія, Віла-Реал)  
 Чеський університет природничих наук (Чехія, Прага)  
 Національний університет «Чернігівська Політехніка» (м. Чернігів)  
 Тернопільський національний технічний університет  
 імені Івана Пулюя (м. Тернопіль)  
 Чернівецький національний університет імені Юрія Федьковича (м. Чернівці)

6 травня 2025 року, м. Луцьк

### ПРОГРАМНЕ ТА АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ В ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЯХ

Тези доповідей Міжнародної науково-практичної конференції  
 молодих вчених та студентів

### SOFTWARE AND HARDWARE IN INFORMATION TECHNOLOGIES

Abstracts of the International Scientific Conference  
 for Young Scientists and Students

**Випуск 1**

Луцький національний технічний університет

Луцьк – 2025

<b>Yevhenii MARTSENIUK</b> UNIVERSAL CENTRALIZED SECRET DATA MANAGEMENT FOR AUTOMATED PUBLIC CLOUD PROVISIONING	<b>117</b>
<b>Вадим МІРКЕВИЧ, Людмила КОНКЕВИЧ</b> СИСТЕМИ ЛАЗЕРНОГО ПОЗИЦІОНУВАННЯ	<b>120</b>
<b>Дмитро МІСЮРА, Світлана ЛАВРЕНЧУК, Олег КАЙДИК, Ніна ЗДОЛБИЦЬКА</b> СИСТЕМА МОНІТОРИНГУ ЗАРЯДУ АКУМУЛЯТОРНОЇ БАТАРЕЇ НА ОСНОВІ ESP32 ТА BMS-КОНТРОЛЕРА	<b>123</b>
<b>Андрій МОРОЗ, Вадим СВІТАЧ, Сергій ГРИНЮК, Олег КАЙДИК</b> ДО ПИТАННЯ СИСТЕМИ ДОМАШНЬОЇ АВТОМАТИЗАЦІЇ ІЗ ГОЛОСОВИМ КЕРУВАННЯМ НА ПЛАТФОРМІ ARDUINO	<b>126</b>
<b>Данило МОТРУНЧИК, Микола САДОВИЙ, Олег КАЙДИК, Тарас ТЕРЛЕЦЬКИЙ, Анатолій ПРИСТУПА</b> ПРО ВІДДАЛЕНУ РЕЄСТРАЦІЮ СУБ'ЄКТІВ ДОСТУПУ В СКУД	<b>128</b>
<b>Ілля МОТРУНЧИК, Сергій ГРИНЮК</b> ІНТЕЛЕКТУАЛЬНИЙ СОНЯЧНИЙ ТРЕКЕР НА БАЗІ ARDUINO	<b>130</b>
<b>Наталія НАКОНЕЧНА</b> МЕТОДИ ЗАХИСТУ ВІД ФІНАНСОВИХ МАХІНАЦІЙ В КІБЕРПРОСТОРИ	<b>133</b>
<b>Сергій НАУМЕНКО, Інна РОЗЛОМІЙ</b> МОДЕЛЬ АДАПТИВНОЇ ПОБУДОВИ ДОВІРЧИХ ІОТ-МЕРЕЖ ІЗ ДИНАМІЧНОЮ РОТАЦІЄЮ ВУЗЛІВ	<b>136</b>
<b>Артем НИКИТЮК, Лілія ПОЛЩУК</b> ІНТЕГРАЦІЯ SMART-СИСТЕМ З ХМАРНИМИ ПЛАТФОРМАМИ	<b>139</b>
<b>Роман НИЧИПОРУК, Катерина БОРТНИК</b> МЕРЕЖЕВА АРХІТЕКТУРА СИСТЕМИ ВІДЕОСПОСТЕРЕЖЕННЯ ЗА ЖИТЛОВИМ КОМПЛЕКСОМ	<b>142</b>
<b>Арсен ПАЙЦУН, Олександр БУРБАН, Олексій КОСТЮЧКО</b> СИСТЕМА УПРАВЛІННЯ РОБОТИЗОВАНИМ МЕХАНІЗМОМ	<b>144</b>

**Дмитро МІСЮРА,**  
здобувач вищої освіти,  
**Світлана ЛАВРЕНЧУК,**  
канд. техн. наук, доцент,  
**Олег КАЙДИК,**  
канд. техн. наук, доцент,  
**Ніна ЗДОЛЬЩЬКА,**  
канд. техн. наук, доцент

Луцький національний технічний університет

## СИСТЕМА МОНІТОРИНГУ ЗАРЯДУ АКУМУЛЯТОРНОЇ БАТАРЕЇ НА ОСНОВІ ESP32 ТА BMS-КОНТРОЛЕРА

На ефективність роботи пристроїв, що працюють від автономних джерел енергії, значною мірою впливають характеристики акумуляторів, що використовуються для їх живлення. Напруга, струм, температура та параметри оцінки стану заряду (SoC) акумуляторної батареї мають вирішальне значення для функціонування всієї системи.

Вбудовані BMS зустрічаються повсюди [1] (електротранспорт, портативні гаджети, системи накопичення енергії, електроінструмент), їх лише відрізняє різна складність (табл. 1).

Таблиця 1 – Функціонал BMS

Сфера застосування BMS	Особливості
Електротранспорт	<ul style="list-style-type: none"> <li>– складні BMS з CAN-шиною;</li> <li>– активне балансування комірок;</li> <li>– температурний контроль;</li> <li>– алгоритми прогнозування зносу.</li> </ul>
Портативні гаджети	<ul style="list-style-type: none"> <li>– захист від перезаряду та глибокого розряду;</li> <li>– контроль температури батареї;</li> <li>– оптимізація енергоспоживання.</li> </ul>
Енергосистеми та накопичувачі енергії	<ul style="list-style-type: none"> <li>– управління зарядом від сонячної енергії;</li> <li>– підключення до IoT;</li> <li>– захист від перегріву та перевантаження.</li> </ul>
Інструменти та електроніка	<ul style="list-style-type: none"> <li>– автоматичне вимкнення при перегріві;</li> <li>– контроль струму при навантаженнях.</li> </ul>

Як бачимо з таблиці 1, система керування батареєю (BMS) має важливе значення для підтримки працездатності, безпеки та

продуктивності батареї. Зазвичай моніторингу включає такі параметри, як температура, напруга та струм.

В [2] розглядають різні схеми балансування комірок, особлива увага акцентована на алгоритмах балансування комірок і методах оптимізації, окреслено також важливі параметри, залучені до процесу балансування комірок.

У статті [3] запропоновано цифровий двійник батареї, що точно відображає її динаміку зміни стану заряду (SOC) та працездатності (SOH) в режимі реального часу. Модель SOH здійснює оцінку втрати ємності, алгоритм незмінний в часі, а модель SOC враховує вплив старіння і динамічно перенавчається. Ефективність підходу продемонстровано на відкритому наборі даних, що підтверджує його точність і придатність для використання в бортових системах.

Для досягнення більшої вихідної потужності елементи з'єднують послідовно, важливим тут є моніторинг стану кожної з комірок окремо.

Для створення власної бюджетної системи керування акумуляторами використано елементну базу: мікроконтролер ESP32/ESP8266, датчики напруги та температури, захист від перезаряду і короткого замикання (модуль 4S 40A), дисплей для виведення даних. Модуль 4S 40A поєднує одночасно функції балансування комірок, захисту від перенапруги, захист від короткого замикання та захисту від зниженої напруги (рис. 1).

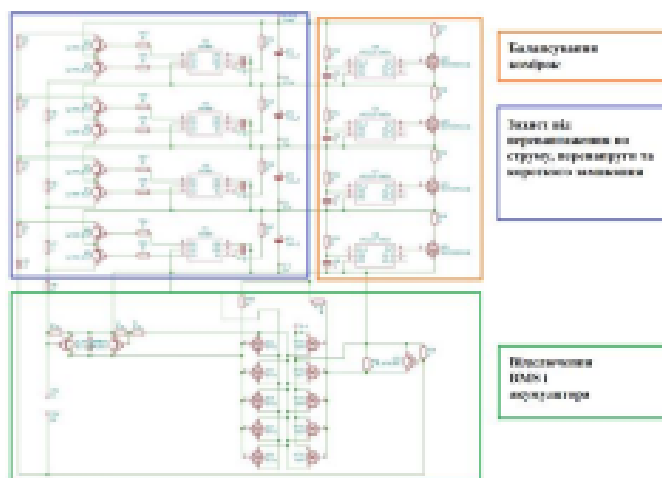


Рисунок 1 – Принципова схема контролера заряду 4S 40A

*Міжнародна науково-практична конференція молодих науковців та студентів  
(м. Луцьк, 6 травня 2023 р.)*

Схема підключення комірок наведена на рисунку 2.

BMS-контролер функціонує як 4 окремі модулі для кожної з комірок, які потім інтегруються за допомогою транзисторів та пасивних компонентів, утворюючи єдину систему, здатну подавати струм до 40 А і забезпечувати захист для кожної комірки.

Створено алгоритми: вимірювання та зберігання температури комірок й напруги, обчислення SOC і SOH, оновлення даних на дисплеї.

Для віддаленого моніторингу є можливість під'єднатися до смартфона або ноутбуку чи комп'ютера через Wi-Fi.

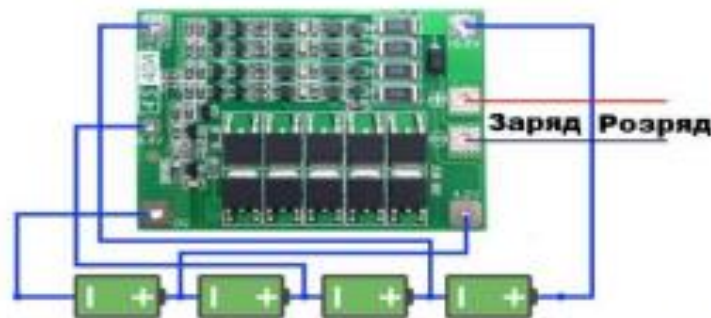


Рисунок 2 – Підключення BMS-контролера до акумулятора

Для тестування роботи системи використано акумуляторний блок – з'єднані послідовно 4 акумулятори типу 18650.

Розроблена система моніторингу стану батареї дозволяє не лише здійснювати моніторинг та ефективний контроль за станом акумуляторів, але й підвищити їх безпеку при використанні та продовжити термін служби. Систему легко можна адаптувати для інших пристроїв, що використовують літійові акумулятори. Подальший розвиток даної системи може бути в напрямку аналізу даних та застосування алгоритмів машинного навчання для прогнозування термінів служби або ж надання рекомендацій щодо оптимальної експлуатації.

#### Перелік використаних джерел

1. N. Zdobitska, M. Delyavskyy, N. Lishchyna, V. Lishchyna, S. Lavrenchuk, V. Sulim. DIY Smart Auxiliary Power Supply for Emergency. *Advances in Artificial Systems for Logistics Engineering III*. 2023. Pp. 382-392. DOI: [https://doi.org/10.1007/978-3-031-36115-9\\_35](https://doi.org/10.1007/978-3-031-36115-9_35)

2. Anupama R Itagi, et al. Cell Balancing Paradigms: Advanced Types, Algorithms, and Optimization Frameworks. *arXiv*. 2024. Pp. 1-33. DOI: <https://doi.org/10.48550/arXiv.2411.05478>

3. K. S. S. Alamin, Y. Chen, E. Macii, M. Poncino and S. Vinco. A Machine Learning-based Digital Twin for Electric Vehicle Battery Modeling. *International Conference on Omni-layer Intelligent Systems*. 2022. Pp. 1-6. DOI: <https://doi.org/10.1109/COINS54846.2022.9854960>

**Андрій МОРОЗ,**

здобувач вищої освіти,

**Вадим СВІТАЧ,**

здобувач вищої освіти,

**Сергій ГРИНЮК,**

канд. техн. наук, доцент,

**Олег КАЙДИК,**

канд. техн. наук, доцент

Луцький національний технічний університет

## **ДО ПИТАННЯ СИСТЕМИ ДОМАШНЬОЇ АВТОМАТИЗАЦІЇ ІЗ ГОЛОСОВИМ КЕРУВАННЯМ НА ПЛАТФОРМІ ARDUINO**

Сучасні тенденції розвитку Інтернету речей (IoT) спрямовані на підвищення комфорту та безпеки користувачів. Одним із популярних напрямків є створення «розумних» будинків, які дозволяють автоматизувати управління освітленням, кліматичними системами, побутовими приладами. Зокрема, зручним способом взаємодії з такими системами є голосове керування, яке не потребує додаткових фізичних дій з боку користувача.

В даній роботі представлено рішення для реалізації системи домашньої автоматизації з голосовим керуванням на основі мікроконтролера Arduino Uno та модуля Bluetooth HC-05. Система дозволяє керувати підключеними пристроями за допомогою голосових команд, що передаються зі смартфона [1].

Система (рис. 1) складається з мікроконтролера Arduino Uno, до якого підключено модуль Bluetooth HC-05 та реле для управління електричними приладами. Також використовується мікрофонний модуль або мобільний додаток з підтримкою голосових команд.

## Додаток В

### Код програми. Одноканальний режим

```
// Код для ESP32 Dev Module (ESP32-WROOM-32)
/* Вивід на консоль значень напруги з чотирьох АЦП каналів ADC1_0, ADC1_1, ADC1_2,
ADC1_3*/
// Сумісно з Wi-Fi, оскільки використовуються канали ADC1
#include <WiFi.h> // Додано для демонстрації можливості використання Wi-Fi
// Визначаємо GPIO піни для кожного АЦП каналу ADC1, який використовуватимемо
const int ADC1_CH0_PIN = 36; // GPIO 36 (ADC1_CH0)
const int ADC1_CH1_PIN = 37; // GPIO 37 (ADC1_CH1)
const int ADC1_CH2_PIN = 38; // GPIO 38 (ADC1_CH2)
const int ADC1_CH3_PIN = 39; // GPIO 39 (ADC1_CH3)

// Значення VREF для ESP32 (зазвичай близько 3.3V, але може відрізнятись)
// Для більшої точності VREF можна виміряти
const float VREF = 3.3; // Типова опорна напруга для ESP32

// Налаштування Wi-Fi (для демонстрації сумісності)
const char* ssid = "my_SSID"; // тут назва нашої Wi-Fi мережі
const char* password = "my_PASSWORD"; // тут пароль нашої Wi-Fi мережі

void setup() {
  Serial.begin(115200); // Ініціалізація послідовного порту для виводу на консоль
  delay(100); // Невелика затримка для стабілізації Serial

  Serial.println("ESP32 ADC1 Multiple Channel Read Example with Wi-Fi
Compatibility");

  // --- Налаштування Wi-Fi ---
  Serial.print("Connecting to WiFi: ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  int wifi_retries = 0;
  while (WiFi.status() != WL_CONNECTED && wifi_retries < 20) {
    delay(500);
    Serial.print(".");
    wifi_retries++;
  }

  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("\nWiFi connected!");
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());
  } else {
    Serial.println("\nFailed to connect to WiFi. Please check SSID/Password.");
  }
  Serial.println("-----");

  // --- Налаштування АЦП ---
  // Налаштування атенюації для кожного піна ADC1
  // ADC_11db дає діапазон 0-3.3V, що підходить для більшості застосувань
  analogSetPinAttenuation(ADC1_CH0_PIN, ADC_11db);
  analogSetPinAttenuation(ADC1_CH1_PIN, ADC_11db);
  analogSetPinAttenuation(ADC1_CH2_PIN, ADC_11db);
  analogSetPinAttenuation(ADC1_CH3_PIN, ADC_11db);

  Serial.println("ADC1 channels initialized:");
  Serial.print(" ADC1_CH0 (GPIO "); Serial.print(ADC1_CH0_PIN);
  Serial.println(")");
}
```

```

    Serial.print(" ADC1_CH1 (GPIO "); Serial.print(ADC1_CH1_PIN);
Serial.println(")");
    Serial.print(" ADC1_CH2 (GPIO "); Serial.print(ADC1_CH2_PIN);
Serial.println(")");
    Serial.print(" ADC1_CH3 (GPIO "); Serial.print(ADC1_CH3_PIN);
Serial.println(")");
    Serial.println("-----");
}

void loop() {
    // Зчитування аналогових значень з кожного каналу ADC1
    int rawValue0 = analogRead(ADC1_CH0_PIN);
    int rawValue1 = analogRead(ADC1_CH1_PIN);
    int rawValue2 = analogRead(ADC1_CH2_PIN);
    int rawValue3 = analogRead(ADC1_CH3_PIN);

    // Перетворення сирих значень АЦП в напругу (Вольти)
    // Максимальне сире значення для 12-бітного АЦП = 2^12 - 1 = 4095
    float voltage0 = (float)rawValue0 / 4095.0 * VREF;
    float voltage1 = (float)rawValue1 / 4095.0 * VREF;
    float voltage2 = (float)rawValue2 / 4095.0 * VREF;
    float voltage3 = (float)rawValue3 / 4095.0 * VREF;

    // Вивід значень на консоль
    Serial.print("ADC1_CH0 (GPIO "); Serial.print(ADC1_CH0_PIN); Serial.print("):
Raw = "); Serial.print(rawValue0); Serial.print(", Voltage = ");
Serial.print(voltage0, 3); Serial.println(" V");
    Serial.print("ADC1_CH1 (GPIO "); Serial.print(ADC1_CH1_PIN); Serial.print("):
Raw = "); Serial.print(rawValue1); Serial.print(", Voltage = ");
Serial.print(voltage1, 3); Serial.println(" V");
    Serial.print("ADC1_CH2 (GPIO "); Serial.print(ADC1_CH2_PIN); Serial.print("):
Raw = "); Serial.print(rawValue2); Serial.print(", Voltage = ");
Serial.print(voltage2, 3); Serial.println(" V");
    Serial.print("ADC1_CH3 (GPIO "); Serial.print(ADC1_CH3_PIN); Serial.print("):
Raw = "); Serial.print(rawValue3); Serial.print(", Voltage = ");
Serial.print(voltage3, 3); Serial.println(" V");
    Serial.println("-----");

    delay(1000); // Затримка 1 секунда між зчитуваннями
}

```

## Додаток Г

### Код програми. Багатоканальний режим

```
//Багатоканальний режим: 4 канали із роздільними коефіцієнтами та зміщеннями,
//додатковим усередненням по трьох точках. Дані виводяться веб-сервером по http
// ESP32 Dev Module
// Код для ESP32 Dev Module (ESP32-WROOM-32)
// Вивід на консоль каліброваних значень напруги
//з чотирьох АЦП каналів ADC1_0, ADC1_1, ADC1_2, ADC1_3
// Сумісно з Wi-Fi, використовуються канали ADC1
// Враховано зовнішні підтягуючі резистори 10K до 3.3V
// Додано коефіцієнти калібрування для діапазону 0.5V - 2.0V

#include <ESPAsyncWebServer.h>
#include <AsyncTCP.h>
#include <WiFi.h>

// Визначаємо GPIO піни для кожного АЦП каналу ADC1
const int ADC1_CH0_PIN = 36; // GPIO 36 (ADC1_CH0)
const int ADC1_CH1_PIN = 39; // GPIO 39 (ADC1_CH3)
const int ADC1_CH2_PIN = 35; // GPIO 35 (ADC1_CH5)
const int ADC1_CH3_PIN = 34; // GPIO 34 (ADC1_CH6)

// --- Коефіцієнти калібрування для ДІАПАЗОНУ 0.5V - 2.0V ---
// ці значення отримано дослідним шляхом, за допомогою мультиметра

// Для ADC1_CH0 (GPIO 36)
// Якщо (rawValue_0_5V_CH0 = 620, rawValue_2_0V_CH0 = 2480)
// scale = (2.0 - 0.5) / (2480 - 620) = 1.5 / 1860 = 0.00080645
// offset = 0.5 - (0.00080645 * 620) = 0.5 - 0.5000 = -0.00002
float calibration_scale_CH0 = 0.00078520;//0.00078055; // // 1
float calibration_offset_CH0 = -0.000025;
float calibratedVoltage0 = 0.0 ;

// Для ADC1_CH1 (GPIO 39)
float calibration_scale_CH1 = 0.00078530;//0.00078080; // // 2
float calibration_offset_CH1 = -0.000025;
float calibratedVoltage1 = 0.0 ;

// Для ADC1_CH2 (GPIO 35)
float calibration_scale_CH2 = 0.00078553;//0.00078103; // // 3
float calibration_offset_CH2 = -0.000025;
float calibratedVoltage2 = 0.0 ;

// Для ADC1_CH3 (GPIO 34)
float calibration_scale_CH3 = 0.00078550;//0.00078100; // // 4
float calibration_offset_CH3 = -0.000025;
float calibratedVoltage3 = 0.0 ;

// Налаштування Wi-Fi (вказуємо параметри підключення до нашої мережі)
const char* ssid = "wifissid"; // назва нашої Wi-Fi мережі
const char* password = "pass"; // пароль нашої Wi-Fi мережі

// означаємо 80-й порт по замовчуванню для об'єкта ASYNC-сервера -> AsyncWebServer
AsyncWebServer server(80);
// Як правило, потрібно використовувати "unsigned long" для змінних,
//які зберігають час
// Значення швидко стане занадто великим для збереження типу int та виникне
//переповнення
unsigned long prevMillis = 0; // зберігаємо час для останнього оновлення

// зчитуємо покази кожних X...X miliseconds
```

```

const long time_interval = 600;

// ----- HTML-код web-сторінки -----
const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
  <meta charset="UTF-8"> <meta name="viewport" content="width=device-width,
initial-scale=1">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css" integrity="sha512-
iecdLmaskl7CVkqkXNQ/ZH/XLlvWZOJyj7Yy7tcenmpDlypASozpmT/E0iPtmFIB46ZmdtAc9eNBvH0H/Z
piBw==" crossorigin="anonymous" referrerpolicy="no-referrer" />
  <style>
    html { font-family: Arial; display: inline-block; margin: 0px auto; text-
align: center; }
    h2 { font-size: 2.5rem; }
    p { font-size: 2.0rem; /* Зменшив для кращого розміщення */}
    .value-span { font-size: 2.9rem; color:#4855bc; font-weight: bold; } /* Клас
для спанів */
    .channel-label { font-size: 2.5rem; color: #555; }
  </style>
</head>
<body>
  <h2>ESP WEB-SERVER</h2>
  <h2 style="color:#9b30b6;">Напруги комірок батареї:</h2>
  <p>
    <span class="channel-label">Елемент 1 :</span> <span id="CH0" class="value-
span">%CH0%</span> V
  </p>
  <p>
    <span class="channel-label">Елемент 2 :</span> <span id="CH1" class="value-
span">%CH1%</span> V
  </p>
  <p>
    <span class="channel-label">Елемент 3 :</span> <span id="CH2" class="value-
span">%CH2%</span> V
  </p>
  <p>
    <span class="channel-label">Елемент 4 :</span> <span id="CH3" class="value-
span">%CH3%</span> V
  </p>
</body>
<script>
  setInterval(function() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
      if (this.readyState == 4 && this.status == 200) {
        try {
          var jsonResponse = JSON.parse(this.responseText);
          document.getElementById("CH0").innerHTML = jsonResponse.v0;
          document.getElementById("CH1").innerHTML = jsonResponse.v1;
          document.getElementById("CH2").innerHTML = jsonResponse.v2;
          document.getElementById("CH3").innerHTML = jsonResponse.v3;
        } catch (e) {
          console.error("Помилка розбору JSON: ", e);
          // Можна вивести повідомлення про помилку для користувача, якщо потрібно
          document.getElementById("CH0").innerHTML = "Пом.";
          document.getElementById("CH1").innerHTML = "Пом.";
          document.getElementById("CH2").innerHTML = "Пом.";
          document.getElementById("CH3").innerHTML = "Пом.";
        }
      } else if (this.readyState == 4) {
        // Обробка помилок запиту (наприклад, status 404, 500)
        console.error("Помилка HTTP запиту: " + this.status);
        document.getElementById("CH0").innerHTML = "Пом. HTTP";
      }
    };
    xhttp.open("GET", "http://192.168.1.100/status", true);
    xhttp.send();
  }, 600);
</script>
)rawliteral";

```

```

        document.getElementById("CH1").innerHTML = "Пом. HTTP";
        document.getElementById("CH2").innerHTML = "Пом. HTTP";
        document.getElementById("CH3").innerHTML = "Пом. HTTP";
    }
};
xhttp.open("GET", "/allvoltages", true); // Запит на новий ендпоінт
xhttp.send();
}, 1000); // Оновлювати кожні 1000 мілісекунд (1 секунда)
</script>
</html>rawliteral";

// Функція-процесор для заміни плейсхолдерів в HTML
String processor(const String& var){
    if(var == "CH0"){ return String(calibratedVoltage0, 2); }
    if(var == "CH1"){ return String(calibratedVoltage1, 2); }
    if(var == "CH2"){ return String(calibratedVoltage2, 2); }
    if(var == "CH3"){ return String(calibratedVoltage3, 2); }
    return String(); // Повертаємо порожній рядок, якщо плейсхолдер не знайдено
}

void setup() {
    Serial.begin(115200); // Ініціалізація послідовного порту для виводу на консоль
    delay(100); // Невелика затримка для стабілізації Serial

    Serial.println("ESP32 ADC1 Multi-Channel Read with Calibration (0.5V-2.0V) and
Averaging");
    Serial.println("External 10K pull-ups to 3.3V considered. Wi-Fi compatible.");

    // --- Налаштування Wi-Fi ---
    Serial.print("Connecting to WiFi: ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);

    int wifi_retries = 0;
    while (WiFi.status() != WL_CONNECTED && wifi_retries < 20) {
        delay(500);
        Serial.print("."); // візуалізація циклу ... очікування
        wifi_retries++;
    }

    if (WiFi.status() == WL_CONNECTED) {
        Serial.println("\nWiFi connected!");
        Serial.print("IP address: ");
        Serial.println(WiFi.localIP());
    } else {
        Serial.println("\nFailed to connect to WiFi. Please check SSID/Password.");
    }
    Serial.println("-----");

    // --- Налаштування АЦП ---
    // Налаштування атенюації для кожного піна ADC1. ADC_11db дає діапазон 0-3.3V.
    analogSetPinAttenuation(ADC1_CH0_PIN, ADC_11db);
    analogSetPinAttenuation(ADC1_CH1_PIN, ADC_11db);
    analogSetPinAttenuation(ADC1_CH2_PIN, ADC_11db);
    analogSetPinAttenuation(ADC1_CH3_PIN, ADC_11db);

    Serial.println("ADC1 channels initialized:");
    Serial.print("  ADC1_CH0 (GPIO "); Serial.print(ADC1_CH0_PIN);
Serial.println(")");
    Serial.print("  ADC1_CH1 (GPIO "); Serial.print(ADC1_CH1_PIN);
Serial.println(")");
    Serial.print("  ADC1_CH2 (GPIO "); Serial.print(ADC1_CH2_PIN);
Serial.println(")");
}

```

```

Serial.print(" ADC1_CH3 (GPIO "); Serial.print(ADC1_CH3_PIN);
Serial.println(")");
Serial.println("-----");
Serial.println("Calibration Parameters (update these after performing
calibration!):");
Serial.print("CH0: Scale = "); Serial.print(calibration_scale_CH0, 7);
Serial.print(", Offset = "); Serial.println(calibration_offset_CH0, 5);
Serial.print("CH1: Scale = "); Serial.print(calibration_scale_CH1, 7);
Serial.print(", Offset = "); Serial.println(calibration_offset_CH1, 5);
Serial.print("CH2: Scale = "); Serial.print(calibration_scale_CH2, 7);
Serial.print(", Offset = "); Serial.println(calibration_offset_CH2, 5);
Serial.print("CH3: Scale = "); Serial.print(calibration_scale_CH3, 7);
Serial.print(", Offset = "); Serial.println(calibration_offset_CH3, 5);
Serial.println("-----");

// старт навігації по веб сторінці Route for root/web page
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/html", index_html, processor);
});

// НОВИЙ ОБРОБНИК: для отримання всіх значень напруги у форматі JSON
server.on("/allvoltages", HTTP_GET, [](AsyncWebServerRequest *request){
    String jsonResponse = "{";
    jsonResponse += "\"v0\":\\"" + String(calibratedVoltage0, 2) + "\", ";
    jsonResponse += "\"v1\":\\"" + String(calibratedVoltage1, 2) + "\", ";
    jsonResponse += "\"v2\":\\"" + String(calibratedVoltage2, 2) + "\", ";
    jsonResponse += "\"v3\":\\"" + String(calibratedVoltage3, 2) + "\"";
    jsonResponse += "}";
    request->send(200, "application/json", jsonResponse);
});

server.begin();
Serial.println("HTTP server started");

}

// Функція для зчитування значень АЦП
int readAndAverageADC(int adcPin) {
    return analogRead(adcPin);
}

void loop() {
    unsigned long currMillis = millis();
    if (currMillis - prevMillis >= time_interval) {
        // зберегти час останнього оновлення для давача
        prevMillis = currMillis;

        // Зчитування аналогових значень з кожного каналу ADC1 з усередненням
        int rawValue0 = readAndAverageADC(ADC1_CH0_PIN);
        int rawValue1 = readAndAverageADC(ADC1_CH1_PIN);
        int rawValue2 = readAndAverageADC(ADC1_CH2_PIN);
        int rawValue3 = readAndAverageADC(ADC1_CH3_PIN);

        // Застосування калібрувальних коефіцієнтів
        // Важливо: Калібрування застосовується безпосередньо до сирого значення АЦП,
        // оскільки коефіцієнти вже розраховані для перетворення RawValue -> Voltage.
        calibratedVoltage0 = (float)rawValue0 * calibration_scale_CH0 +
        calibration_offset_CH0;
        calibratedVoltage1 = (float)rawValue1 * calibration_scale_CH1 +
        calibration_offset_CH1;
        calibratedVoltage2 = (float)rawValue2 * calibration_scale_CH2 +
        calibration_offset_CH2;
        calibratedVoltage3 = (float)rawValue3 * calibration_scale_CH3 +
        calibration_offset_CH3;
    }
}

```

```
// Вивід значень на консоль
Serial.print("CH0 (GPIO "); Serial.print(ADC1_CH0_PIN); Serial.print("): Raw Avg
= "); Serial.print(rawValue0); Serial.print(", Calibrated V = ");
Serial.print(calibratedVoltage0, 3); Serial.println(" V");
Serial.print("CH1 (GPIO "); Serial.print(ADC1_CH1_PIN); Serial.print("): Raw Avg
= "); Serial.print(rawValue1); Serial.print(", Calibrated V = ");
Serial.print(calibratedVoltage1, 3); Serial.println(" V");
Serial.print("CH2 (GPIO "); Serial.print(ADC1_CH2_PIN); Serial.print("): Raw Avg
= "); Serial.print(rawValue2); Serial.print(", Calibrated V = ");
Serial.print(calibratedVoltage2, 3); Serial.println(" V");
Serial.print("CH3 (GPIO "); Serial.print(ADC1_CH3_PIN); Serial.print("): Raw Avg
= "); Serial.print(rawValue3); Serial.print(", Calibrated V = ");
Serial.print(calibratedVoltage3, 3); Serial.println(" V");
Serial.println("-----");

}

}
```