

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної інженерії та управління _____
Кафедра _____ Автоматизації проєктування обчислювальної техніки _____
Рівень вищої освіти _____ перший (бакалаврський) _____
Спеціальність _____ 123 Комп'ютерна інженерія _____
Тип програми _____ Освітньо-професійна _____
Освітня програма _____ Комп'ютерна інженерія _____

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
« ____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувачу _____ Фенченко Артуру Валерійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Зарядний пристрій на основі мікроконтролера _____

затверджена наказом університету від " 21 " _____ 05 _____ 2025 р. № 403 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії _____ 20.06.2025 _____

3. Вихідні дані до роботи _____

Arduino Uno r3, tp4056 _____

Дисплей LCD 2004 I2C, датчик напруги Ina219 _____

Мова програмування C++ _____

4. Перелік питань, що потрібно опрацювати в роботі _____

Аналіз предметної галузі та постановка задачі проєктування. _____

Розробка програмної частини пристрою. _____

Розробка апаратної частини пристрою. _____

Тестування роботи пристрою. Програмування мікроконтролерного пристрою. _____

Висновки. _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
13 слайдів

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Видача теми проекту, узгодження і затвердження теми	02.05.2025-02.05.2025	
2	Аналіз проблемної галузі, постановка задачі, вибір інструментальних засобів	03.05.2025 -8.05.2025	
3	Розробка електричної схеми пристрою	9.05.2025 -12.05.2025	
4	Реалізація макету пристрою	13.05.2025 -20.05.2025	
5	Розробка програми для мікроконтролера	25.05.2025 -29.05.2025	
6	Тестування системи	30.05.2025 -04.06.2025	
7	Оформлення пояснювальної записки	05.06.2025 -07.06.2025	
8	Перевірка виконаного проекту керівником, допуск до захисту	08.06.2025 -10.06.2025	
9	Захист проекту	16.06.2025 -25.06.2025	

Студент _____
(підпис)

Керівник роботи _____
(підпис) проф. Литвинова Є. І.

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи бакалавра: 53 с.,
22 рис., 10 джерел.

МІКРОКОНТРОЛЕР, АКУМУЛЯТОР, ПРИСТРІЙ, ARDUINO,
СТРУМ

Метою кваліфікаційної роботи є розробка зарядного пристрою Li-Ion акумуляторів за допомогою мікроконтролеру Arduino Uno R3.

У ході виконання кваліфікаційної роботи був проведений огляд існуючих проектних рішень, розроблено структуру зарядного пристрою на основі Arduino Uno R3, здійснено вибір компонентів для реалізації задачі, розроблено програмне забезпечення зарядного пристрою, виконано його тестування.

ABSTRACT

The explanatory note of the qualification work includes: 53 pages, 22 figures, and 10 referenices.

MICROCONTROLLER, BATTERY, DEVICE, ARDUINO,
CURRENT

The purpose of the qualification work is to develop a Li-Ion battery charger using the Arduino Uno R3 microcontroller.

During the qualification work, a review of existing design solutions was conducted, the structure of the charger based on the Arduino Uno R3 was developed, components were selected to implement the task, the charger software was developed, and its testing was performed.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ	10
1.1 Загальні відомості про зарядні пристрої та акумулятори	10
1.2 Переваги акумуляторів та зарядних пристроїв	16
1.3 Мета та постановка завдання	20
2 АПАРАТНА РЕАЛІЗАЦІЯ ПРОЕКТУ	21
2.1 Структурна схема пристрою	21
2.2 Вибір мікроконтролера	21
2.3 Елементна складова пристрою	31
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЕКТУ	38
3.1 Вибір мови програмування	38
3.2 Вибір середовища розробки.....	42
3.3 Алгоритм роботи зарядного пристрою	44
3.4 Опис програмного коду проекту.....	44
4 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ЗАРЯДНОГО ПРИСТРОЮ	48
4.1 Збір апаратної системи програмного пристрою.....	48
4.2 Тестування зарядного пристрою.....	49
ВИСНОВКИ	52
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	53

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ,
ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЗП – Зарядний пристрій

BMS – Battery Management System

МК – Мікроконтролер

UART – Universal Asynchronous Receiver Transmitter
(Універсальний Асинхронний Приймач/Передавач)

I2C – Inter-Integrated Circuit протокол

ICSP – In-Circuit Serial Programming (Внутрішньо схемне
послідовне програмування)

IDE – Інтегроване середовище розробки, програмне
забезпечення для написання та компіляції програм

PWM – Широтно-імпульсна модуляція

GND – Ground (заземлення)

LCD – Рідко-кристалічний дисплей

ІОТ – Інтернет речей

СС/CV – Постійний струм / постійна напруга

ВСТУП

Сучасний світ важко уявити без електронних пристроїв, адже вони стали невід'ємною частиною повсякденного життя. Від портативних гаджетів до складних промислових систем, усі вони потребують надійного джерела живлення. Сьогодні мобільність і автономність пристроїв дуже важлива, тому виникла потреба створювати ефективні системи заряджання, які б забезпечували безпеку, зручність і оптимальну продуктивність, тобто були переносимими. Це і поштовхнуло людство на створення акумуляторів. Колись вони були зовсім великі, свинцеві акумулятори, які не можна було зручно переносити, та вони мали низьку ефективність. Але зараз технології розвинулись і продовжують розвиватись, тож ми можемо зберігати енергію в досить маленькому просторі, тому акумулятори можна зустріти будь де – наприклад в вашому смартфоні. Саме ці необхідності визначають актуальність розробки зарядних пристроїв, здатних відповідати вимогам нових технологій.

Одним із ключових напрямів розвитку електроніки є використання мікроконтролерів, тобто компактних обчислювальних пристроїв, які мають малопродуктивний процесор, невелику пам'ять і периферійні модулі в одному чіпі. Вони відкрили нові можливості для створення спеціалізованих систем, які виконують чітко визначені функції з високою ефективністю. У порівнянні з комп'ютерами, мікроконтролери мають обмежені ресурси, але вони здатні працювати в реальному часі, а компактність і низьке енергоспоживання роблять їх ідеальними для вбудованих застосувань. Сьогодні мікроконтролери є основою багатьох пристроїв — від побутової техніки до автомобільних систем і медичного обладнання.

У контексті зарядних пристроїв мікроконтролери відіграють дуже важливу роль. Вони дозволяють створювати інтелектуальні системи, які адаптуються до потреб конкретного акумулятора, тобто в реальному часі обчислюють його характеристики, забезпечують захист від перевантаження та оптимізують процес заряджання. Зокрема, для Li-Ion акумуляторів, які дуже широко застосовуються в портативній електроніці, потрібні зарядні пристрої, здатні точно контролювати параметри живлення. А мікроконтролери, поєднані з відповідними електронними компонентами, такими як датчики струму, напруги чи температури, створюють основу для таких систем.

Метою роботи є зменшення вартості зарядного пристрою для Li-Ion акумуляторів за рахунок використання мікроконтролера. У процесі роботи виконано аналіз існуючих рішень у сфері зарядних пристроїв, визначено їхні особливості та виявлено можливості для вдосконалення.

В роботі проведено оцінку функціональних можливостей існуючих зарядних систем. Після проведення аналізу існуючих рішень визначено шляхи удосконалення пристрою, підвищення його ефективності та зменшення вартості.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ

1.1 Загальні відомості про зарядні пристрої та акумулятори

Акумулятори є важливою складовою сучасних електронних пристроїв, оскільки вони забезпечують автономну роботу в умовах відсутності підключення до електромережі. Вони широко застосовуються, мобільних гаджетах, електротранспорті, аварійних системах живлення, тощо. Але акумулятори мають обмежений запас енергії, тому їх треба перезаряджати час від часу. Для цього і існують зарядні пристрої. Їхнє завдання полягає в тому, щоб подавати електроенергію з розетки для безпечного та ефективного відновлення заряду. Сучасні зарядні пристрої можуть мати різний рівень складності — від простих блоків живлення до інтелектуальних систем на основі мікроконтролерів, здатних адаптуватися до типу акумулятора і умов заряджання. Зі зростанням кількості портативних і енергозалежних пристроїв питання якості та надійності зарядних рішень стає все більш актуальним. Це визначає потребу в розробці гнучких та безпечних зарядних систем із можливістю контролю параметрів у реальному часі, що й обумовлює актуальність даної роботи.

1.1.1 Акумулятори. Акумулятори бувають різних типів залежно від використовуваної хімії, конструкції та призначення. Найбільш поширені типи включають нікель-кадмієві (Ni-Cd), нікель-металгідридні (Ni-MH), літій-іонні (Li-ion), літій-полімерні (Li-Po) та свинцево-кислотні акумулятори. Кожен із них має свої особливості та використовуються в різних випадках. Наприклад, свинцево-кислотні акумулятори часто використовуються в автомобілях і системах безперебійного живлення завдяки великій ємності та низькій вартості, проте вони важкі й

потребують обслуговування. Розуміння типів акумуляторів є важливим для розробки зарядних пристроїв, оскільки кожен тип вимагає специфічного алгоритму заряджання.

1.1.2 Нікель-кадмієві акумулятори. Нікель-кадмієві (Ni-Cd) (рис. 1.1) акумулятори — одні з найстаріших перезаряджуваних типів. Вони вирізняються високою надійністю, здатністю працювати в широкому температурному діапазоні та витримувати великі струми заряду й розряду. Проте мають значний недолік — "ефект пам'яті", тобто ємність знижується при частковому розряді перед наступною зарядкою. Також вони містять токсичний кадмій, що обмежує їх використання через екологічні вимоги.



Рисунок 1.1 – Нікель-кадмієвий акумулятор

Ці акумулятори заряджаються постійним струмом, зазвичай з автоматичним обмеженням напруги. Їх зарядні пристрої часто реалізують "розумне" завершення заряду за допомогою контролю дельта-напруги ($-\Delta V$), коли напруга на акумуляторі після досягнення повної ємності починає трохи спадати. Також можливе використання таймера або температурного датчика для захисту від перезаряду.

1.1.3 Нікель-металгідридні акумулятори. Нікель-металгідридні (Ni-MH) (рис. 1.2) акумулятори є покращеною версією Ni-Cd. Вони мають більшу енергоємність і не містять шкідливих металів, тому є більш

екологічними. Ефект пам'яті у них менш виражений, але вони більш чутливі до перезаряду і саморозряду, що вимагає точнішого контролю під час заряджання.



Рисунок 1.2 – Нікель-металгідридний акумулятор

Цей вид акумуляторів заряджаються подібно до Ni-Cd, однак вони чутливіші до перегріву та перезаряду. Зарядні пристрої для Ni-MH повинні забезпечувати точніший контроль заряду, зокрема за допомогою комбінованих методів – дельта-напруги, температурного контролю ($\Delta T/\Delta t$) та таймерів. Також можна використовувати імпульсний заряд задля зниження ризику деградації.

1.1.4 Літій-іонні акумулятори. Літій-іонні (Li-ion) акумулятори (рис 1.3) сьогодні є стандартом для портативної електроніки, таких як смартфони, ноутбуки й електроінструменти. Вони мають високу енергетичну щільність, малу вагу та не страждають на ефект пам'яті. Їх недоліком є чутливість до перевищення напруги та глибокого розряду, тому обов'язковим є використання спеціальних схем контролю (BMS) для захисту. Нижче (рис. 1.3) можна побачити такий акумулятор, який використовується в смартфонах.



Рисунок 1.3 – Літій-іонний акумулятор

Такі акумулятори потребують складнішого зарядного контролера, що реалізує двоетапний алгоритм: спочатку заряджається постійним струмом (CC) до досягнення максимальної дозволеної напруги (зазвичай 4.2 В на елемент), а далі — постійною напругою (CV) із поступовим зменшенням струму. Надзвичайно важливо забезпечити захист від перезаряду, глибокого розряду, перегріву та короткого замикання, тому зарядні пристрої завжди мають вбудовану схему захисту або працюють разом із BMS.

1.1.5 Літій-полімерні акумулятори. Літій-полімерні (Li-Po) акумулятори — різновид літій-іонних, в яких електроліт має полімерну форму. Це дозволяє створювати тонкі та гнучкі елементи різної форми. Вони легші, безпечніші у випадку механічного пошкодження та мають менший внутрішній опір. Проте вони дорожчі у виробництві та також потребують точного контролю заряду.



Рисунок 1.4 – Літій-полімерний акумулятор

Такі акумулятори заряджаються за тим самим принципом, що й Li-ion, але з ще суворішими вимогами до точності напруги та струму. Їх зарядні пристрої мають обмеження за допустимою напругою (зазвичай теж 4.2 В), однак Li-Po менш терпимі до відхилень, тому будь-яке порушення параметрів може призвести до здуття, або займання. Такі зарядні пристрої обов'язково мають балансування між елементами, якщо батарея складається з кількох елементів.

1.1.6 Свинцево-кислотні акумулятори. Свинцево-кислотні акумулятори являють собою найстаріший тип акумуляторів, проте він і сьогодні широко застосовується в автомобілях (рис 1.5), джерелах безперебійного живлення (UPS) та сонячних системах. Вони недорогі, стійкі до короткочасних перевантажень і здатні віддавати великі струми. Наприклад до такого акумулятору можна підключити інвертор на 220 вольт, та використовувати його як портативну розетку. Проте вони мають велику вагу, низьку енергетичну щільність і вимагають періодичного обслуговування (долив води у класичних моделях).

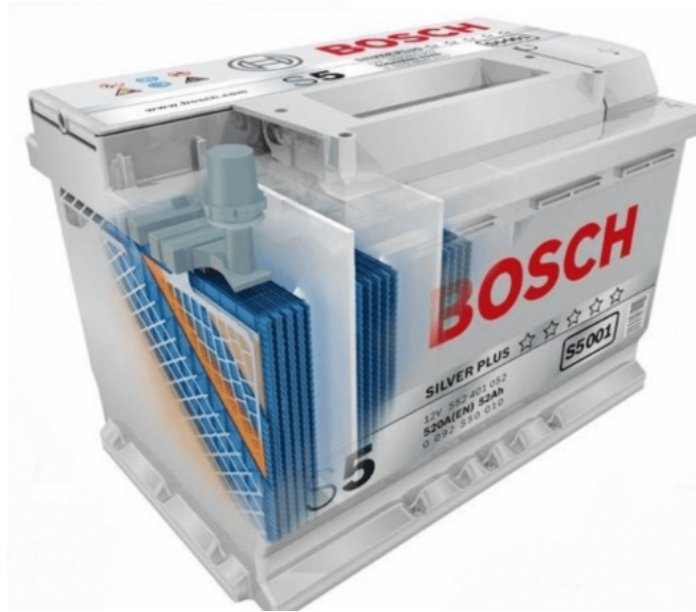


Рисунок 1.5 – Свинцево-кислотний акумулятор для автомобіля

Свинцево-кислотні акумулятори заряджаються у кілька етапів: спочатку йде заряд постійним струмом до досягнення певної напруги, далі — стабілізація напруги та зниження струму (або буферний режим). Зарядні пристрої повинні мати можливість визначати рівень заряду та перемикатись у режим підтримки (float), коли акумулятор повністю заряджений. Також важливим є захист від перенапруги та контроль температури, особливо у випадку гелевих або AGM-варіантів.

1.2 Переваги акумуляторів та зарядних пристроїв

Майже кожен з наведених акумуляторів має свої переваги та недоліки. В основному, на що треба звертати увагу – це розмір та ємність, оскільки ці характеристики і є основними для вибору під час проектування якогось пристрою. Але бувають і ситуації, коли дуже важливо, щоб пристрій мав невеликий розмір, або навпаки – великий, і тоді вибір буде невеликим.

Зарядні пристрої, в свою чергу, мають дуже багато нюансів, яких треба притримуватись при їх проектуванні. Деякі акумулятори дуже чутливі до

перенапруги, тому ЗП повинні підтримувати різноманітні захистні системи, наприклад від короткого замикання. Також вони мають КПД, який повинен бути більше ніж 80% та бути здатним контролювати напругу.

В роботі буде розроблятися зарядний пристрій для літєвих акумуляторів, тому далі будуть розглядатися вони.

1.2.1 Переваги Li-ion акумуляторів. Літєві акумулятори, мають ряд важливих переваг, завдяки яким вони широко використовуються у багатьох пристроях - портативній електроніці, електромобілях, дронах, тощо. Найбільш значимою перевагою є висока енергетична щільність, тобто здатність зберігати велику кількість енергії у відносно малому об'ємі та при низькій масі. Це дозволяє створювати легкі та компактні пристрої з довгим часом автономної роботи.

Ще однією сильною стороною літєвих акумуляторів є відсутність ефекту пам'яті. Це коли, реальна ємність акумулятора знижується при неповних циклах заряду та розряду. Тобто, коли акумулятор має на виході напругу 3.7 вольт (тобто заряджений приблизно на 50%), та якщо його почати заряджати, це йому не зашкодить, на відміну від деяких інших видів. Завдяки цьому такі батареї можна підзаряджати в будь-який момент, не чекаючи повного розряду, що є дуже комфортним плюсом. Крім того, літєві акумулятори мають низький рівень саморозряду: навіть якщо пристрій не використовується тривалий час, втрата заряду буде мінімальною, що робить їх зручними для рідко використовуваної або аварійної техніки.

До переваг також належить тривалий термін служби — зазвичай кілька сотень, а іноді й тисячі циклів заряд-розряд при правильній експлуатації. Для того щоб акумулятор прожив якомога довше, треба виконувати певні умови:

- не перезаряджати – максимальна напруга 4.2 В, якщо допустити вищу, то це може призвести до перегріву, вздуття або займання. За цим має слідкувати спеціальний мікроконтролер, який відповідає за логіку заряду;

- не допускати повного розряду – критична межа 3 В. Іноді трапляються ситуації коли пристроєм довго не користуються, і тоді може

виникнути глибокий розряд, оскільки акумулятори мають властивість потрохи втрачати енергію самі по собі, і це призведе до погіршення стану акумулятора, бо це шкодить його хімічній складовій;

- не заряджати та не розряджати великими струмами, бо, якщо акумулятор на це не розрахований, це знизить його ресурс;

- не зберігати тривалий час повністю зарядженим, або розрядженим. Оптимальний баланс 40-60%, в цьому діапазоні при зберіганні акумулятора, він не отримує майже ніякої шкоди;

- уникати перегріву, бо підвищені температури значно прискорюють деградацію акумулятора. Оптимальна робоча температура становить 20-40 °C.

- використовувати BMS. Ця система контролює при яких умовах акумулятор заряджається, розряжається, контролює його температуру та напругу кожного елемента.

Більшість літєвих акумуляторів підтримують швидку зарядку, що, насправді, дуже важливо для сучасних користувачів, бо не всі хочуть чекати довго. Також є дуже схожий тип акумуляторів - літій-полімерні варіанти особливо зручні в конструктивному плані. Вони мають схожий принцип роботи, але в другому випадку електроліт представлений у вигляді гелю або сухого полімеру, замість рідкого органічного електроліту. Завдяки цьому їх можна виготовляти у різних формах і товщинах, що відкриває великі можливості для дизайну електронних пристроїв.

Однак для безпечного використання літєвих акумуляторів необхідний чіткий контроль на процесах заряду і розряду, тому вони завжди використовуються разом із схемами захисту або системами керування батареєю (BMS). Тому сукупність переваг робить літєві акумулятори оптимальним вибором для більшості сучасних застосувань, де важливі компактність, енергоємність і надійність.

1.2.2 Недоліки Li-ion акумуляторів. Окрім великої кількості переваг, літєві акумулятори мають і багато суттєвих недоліків, які варто враховувати

при їх використанні у практичних пристроях. Одним з найголовніших мінусів є їхня висока чутливість до порушення режимів експлуатації. Тобто перезаряд, глибокий розряд, надмірні струми або перегрів можуть призвести до незворотної деградації хімічного складового акумулятора, здуття, витoku електроліту або навіть займання. Саме тому літієві батареї обов'язково потребують наявності спеціальної схеми контролю — системи керування батареєю (BMS), яка слідкує за напругою, температурою і струмом для кожного осередка та вимикає акумулятор у разі небезпеки.

Важливим недоліком є обмежений температурний діапазон роботи. Літієві елементи погано функціонують при низьких температурах, особливо нижче 0 °C — внутрішній опір зростає, напруга падає, і може зменшитися ємність. У таких умовах заряджання взагалі стає небезпечним, оскільки електрохімічні реакції порушуються, що може пошкодити структуру елемента. Тому можна часто помітити, що акумулятор смартфона дуже швидко розряджається взимку на вулиці. При високих температурах, навпаки, прискорюється старіння та підвищується ризик теплового розгону.

Ще одна проблема таких акумуляторів – це поступова втрата ємності з часом. Навіть при правильній експлуатації літієві акумулятори деградують: зі збільшенням кількості циклів заряд-розряд зменшується їхня здатність утримувати заряд. Це відбувається через утворення внутрішніх відкладень, зменшення площі електродів та інші процеси, пов'язані зі старінням. При цьому акумулятор може ззовні виглядати справним, але його реальний ресурс буде значно нижчим за початковий. Наприклад система Android дає інформацію про реальну ємність акумулятора, тому кожен користувач має можливість перевірити на скільки в нього зберігся акумулятор. Також літієві акумулятори не підлягають глибокій регенерації, як, наприклад, свинцеві.

До недоліків також належить вища вартість виробництва у порівнянні з іншими типами акумуляторів, зокрема з нікель-кадмієвими або свинцево-кислотними. Це пов'язано з використанням складних компонентів, герметичних оболонок і вимогами до точності складання. Крім того, такі

акумулятори складно утилізувати, бо вони містять хімічно активні та потенційно небезпечні речовини, тому потребують спеціалізованої переробки, тому ні в якому разі не можна їх викидати в не прилаштованих для цього місцях.

Важливим обмеженням є складність безпечного транспортування. Через можливість загоряння літєві батареї класифікуються як небезпечний вантаж, а тому їх перевезення регулюється міжнародними нормами. Це створює додаткові логістичні витрати для виробників і споживачів.

Таким чином, хоча літєві акумулятори мають дійсно корисні переваги, їх застосування вимагає ретельного контролю, дотримання технічних умов і наявності захисних систем, без яких використання таких джерел енергії може стати небезпечним або неефективним.

1.3 Мета та постановка завдання

Метою кваліфікаційної роботи є розробка ефективного зарядного пристрою для Li-ion акумуляторів, та мікроконтролерної системи, яка буде моніторити додаткову інформацію про стан акумулятора та умови його заряджання. Ця система повинна забезпечувати безпечний та нешкідливий процес заряджання акумулятора.

Основні завдання цієї роботи включають:

- розробка електричної схеми та проектування структури зарядного пристрою, включаючи вибір усіх необхідних компонентів та датчиків;
- складання апаратної частини системи, включаючи підключення зарядного модуля до акумулятора, з'єднання його з Arduino, підключення датчиків напруги та струму, та інших необхідних компонентів, згідно зі складеною схемою;
- написання програмного коду для мікроконтролера Arduino, який

буде відповідати за зчитування показників стану акумулятора та контролю заряджання акумулятора;

– тестування розробленої системи, перевірка його працездатності, виявлення та виправлення можливих помилок та недоліків, а також оптимізація роботи системи.

2 АПАРАТНА РЕАЛІЗАЦІЯ ПРОЕКТУ

2.1 Структурна схема пристрою

На схемі (рис. 2.1) відображена структура зарядного пристрою для літієвого акумулятора. Ця система буде використовувати зарядний модуль, який буде заряджати батарею, мікроконтролер Arduino для збору інформації з датчиків, дисплей, для відображення інформації, та модуль вимірювання напруги.

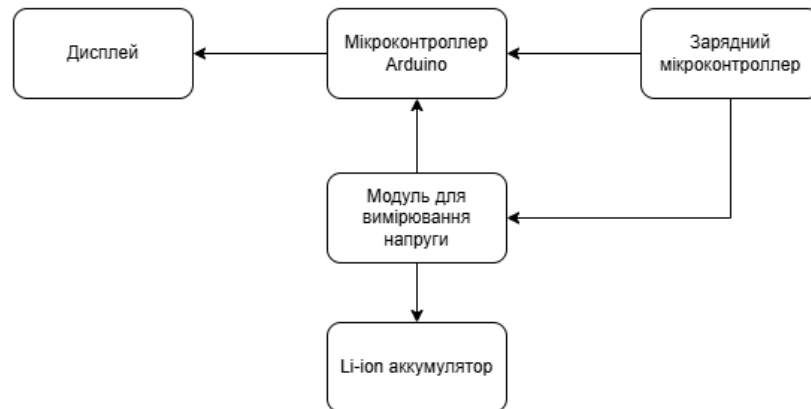


Рисунок 2.1 – Структурна схема зарядного пристрою

Зарядний модуль буде заряджати Li-ion акумулятор, подаючи стабілізовану напругу та струм, а Arduino буде зчитувати дані про напругу та струм з відповідних сенсорів і виводити інформацію на дисплей.

2.1.1 Вибір мікроконтролера

Мікроконтролер для зарядного пристрою треба обирати за кількістю необхідних входів і виходів, наявністю аналогових пінів для зчитування напруги, підтримкою протоколів зв'язку наприклад I²C або UART. Також слід

звертати увагу на енергоспоживання у разі автономної роботи, обчислювальною потужністю та об'єм пам'яті для реалізації логіки керування. Також мікроконтролер повинен бути простотим в програмуванні, та, бажано, популярним, оскільки про нього може бути багато інформації на форумах та підтримка спільноти.

2.1.2 Arduino Uno R3. Arduino Uno R3 (рис. 2.2)– це популярна плата мікроконтролера, яка широко використовується для навчання та створення різноманітних електронних проектів. Основою цієї мікросхеми є мікроконтролер ATmega328P. Саме він виконує всі обчислення і команди, які ти в нього завантажуєш. Плата має 14 цифрових входів/виходів (6 з яких можуть використовуватися як ШІМ-виходи), 6 аналогових входів, кварцовий резонатор на 16 МГц, USB порт, роз'єм живлення, роз'єм ICSP і кнопка скидання налаштувань. Для того щоб почати використовувати цей мікроконтролер, достатньо просто його підключити по USB до комп'ютера, або подати напругу за допомогою 5V блока живлення [1].

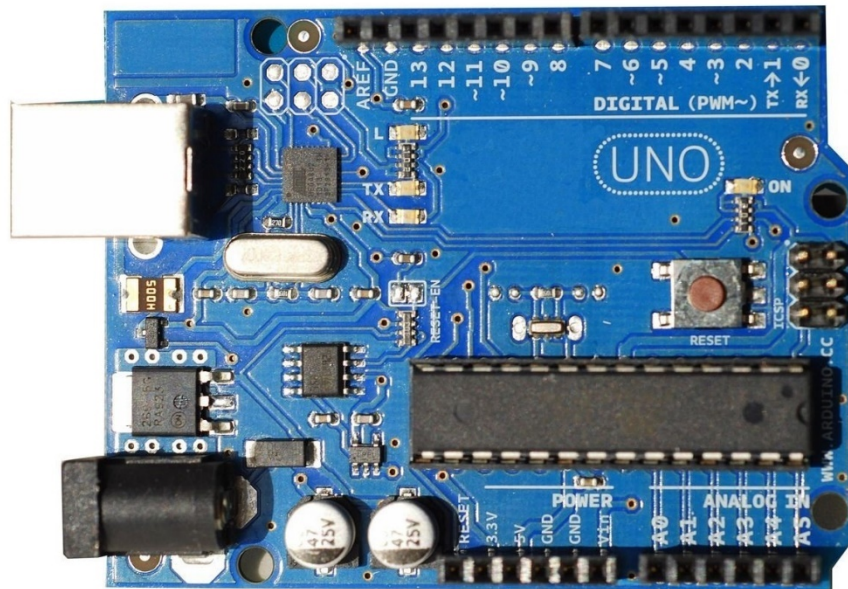


Рисунок 2.2 – Arduino Uno R3

Плата має стандартний форм-фактор, тому це дозволяє підключати до неї різноманітні додаткові модулі та "щити" (shields). Наприклад, щит для підключення до Wi-Fi, щит з дисплеєм або зі звуковим модулем. Все це дозволяє не паяти багато проводів, а просто вставити плату зверху, і все готово до роботи, це дуже практично. Arduino Uno R3 доволі таки просто використовувати, та також вона має дуже велику спільноту, тому вона є непоганим вибором для використання, особливо у новачків.

Для програмування плати використовується середовище розробки Arduino IDE, яке підтримує мову програмування, засновану на C/C++. Після написання коду його треба завантажити на плату через USB-кабель, і вона почне виконувати задані інструкції.

Arduino Uno R3 також підтримує різні протоколи зв'язку, наприклад такі як I²C, SPI та UART. Це дозволяє інтегрувати її з багатьма іншими пристроями та модулями.

Живлення Arduino Uno можна подавати через USB, або через спеціальний порт для зовнішнього джерела, який зазвичай приймає 5В. Це дозволяє використовувати пристрій навіть без комп'ютера, наприклад з батарейкою або зарядним пристроєм від іншого пристрою. Якщо потрібно, Arduino може навіть сама жити інші невеликі модулі через свої виходи. Програмний код, який був написаний завантажується на плату та зберігається у її пам'яті. Тобто якщо вимкнути живлення, а потім знову ввімкнути — нічого страшного не відбудеться, програмний код буде збережений, оскільки ардуїно використовує енергонезалежну пам'ять. Це є дуже зручним для створення постійно працюючих пристроїв [1].

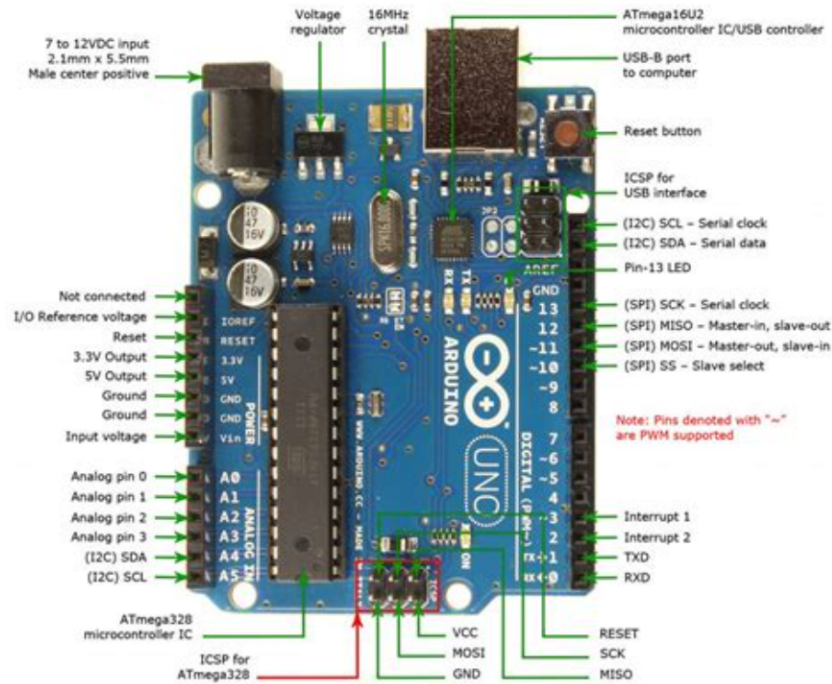


Рисунок 2.3 – Контакти Arduino Uno R3

Контакти Arduino Uno (рис. 2.3) поділяються на цифрові та аналогові, кожний з яких має свою функції. Цифрові контакти (0–13) використовуються для керування пристроями в режимі вмикання/вимикання, а деякі з них (3, 5, 6, 9, 10, 11) підтримують ШІМ для регулювання, наприклад, яскравості світла або швидкості двигуна. Аналогові входи (A0–A5) зчитують напругу з датчиків і перетворюють її в цифрові значення для обробки. Деякі піни призначені для зв'язку: 0 і 1 — UART, 10–13 — SPI, A4 і A5 — I2C. Також є контакти живлення (3.3V, 5V, GND, Vin), які відповідають за живлення плати та роботу зовнішніх компонентів. Є ще контакт AREF, який використовується для точного налаштування аналогового зчитування, та RESET для перезапуску плати.

2.1.3 Arduino Nano. Arduino Nano (рис. 2.4) – це дуже компактна мікроконтролерна плата, яка створена для зручного використання у невеликих учбових або вбудованих проектах. Вона може виконувати ті ж самі функції, що й і вищезазначена та більш відома плата Arduino Uno, але має значно менші

розміри, що і дозволяє легко вбудувати її в готові пристрої. Основою плати є мікроконтролер ATmega328P, той самий, що використовується в Arduino Uno, тому програмне забезпечення і більшість бібліотек повністю сумісні. Також ця плата існує з іншим чіпом – CH340, при використанні такого чіпу потрібно буде ще встановити спеціальний USB-UART драйвер. Завдяки цьому Nano дуже легко підключати до усіляких датчиків, світлодіодів, реле, дисплеїв та інших електронних компонентів.

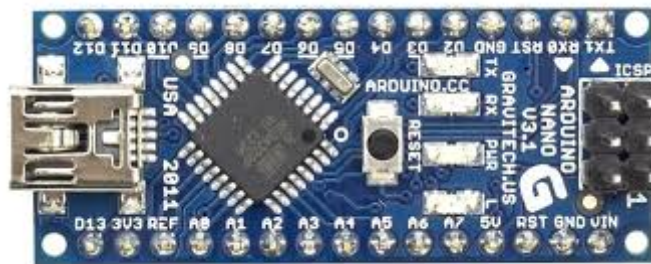


Рисунок 2.4 – Arduino Nano

Arduino Nano має непогану особливість - це розташування контактів у два ряди, що дозволяє легко вставляти її в макетну плату (Breadboard) або модулі з роз'ємами, тому це дуже зручно для експериментів та розробки девайсів. Також існують модифікації цієї моделі Ардуїно з іншим типом USB (наприклад, USB type C) або з іншими мікроконтролерами (наприклад, ATmega32u4). Але класична версія з ATmega328P залишається найпопулярнішою. Завдяки простоті у використанні, Arduino Nano підходить для новачків, які тільки знайомляться з мікроконтролерами, а також для досвідчених розробників, яким важлива компактність та ефективність.

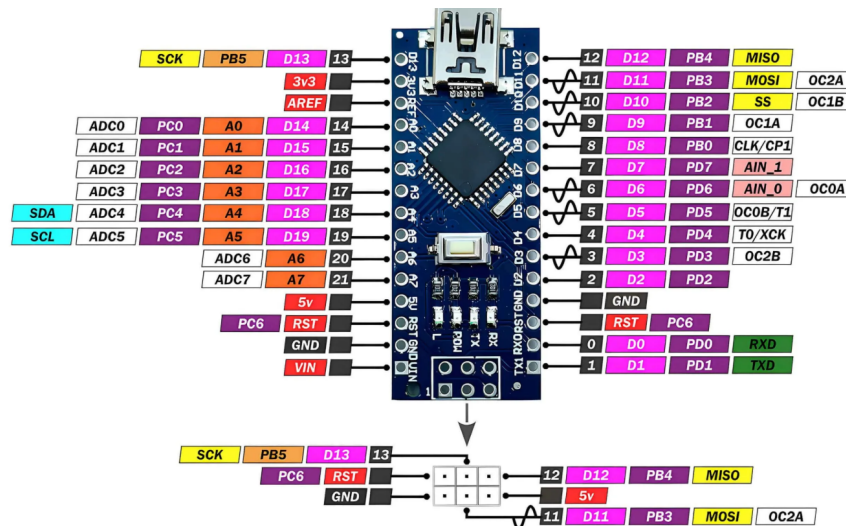


Рисунок 2.5 – Контакти Arduino Nano

На платі розміщено 22 (рис. 2.5) виводи загального призначення: 14 цифрових контактів і 8 аналогових входів. Частина цифрових пінів підтримує PWM, що дозволяє плавно керувати яскравістю світлодіодів або ще чимось, що потребує плавного регулювання. Аналогові входи дозволяють зчитувати аналогові значення, наприклад з температурних чи світлових датчиків. Ця плата має кварцовий резонатор на 16 МГц, який забезпечує стабільну частоту роботи, USB роз'єм обраного типу для програмування та живлення, а також кнопку перезапуску для скидання налаштувань.

Nano має всі основні контакти живлення, включаючи 5V, 3.3V, GND та Vin. Через Vin можна подавати напругу 7–12 В, а 5V використовується для живлення зовнішніх компонентів, таких як усілякі датчики або модулі. Крім того, Nano підтримує зв'язок через UART піни (0 та 1), SPI (D10–D13) та I2C (контакти A4 і A5). Ці піни дозволяють інтегрувати її з багатьма іншими пристроями: дисплеями, модулями пам'яті, сенсорами, Wi-Fi та Bluetooth модулями. Для кожного з пристрою, з яким вона може інтегруватися, існує своя бібліотека, яка містить функції та логіку для взаємодії.

Для того щоб запрограмувати Arduino Nano, треба використовувати середовище розробки Arduino IDE. Воно дозволяє легко створювати,

редагувати і завантажувати код на плату. Плата автоматично розпізнається системою, і програмування відбувається дуже просто, як і у випадку з іншими моделями Arduino. Ця плата, як і Uno, підтримує дуже велику кількість різних бібліотек, які дозволяють взаємодіяти з різними компонентами. Наприклад, бібліотека OneWire та DallasTemperature дозволяють взаємодіяти з датчиками температури, LiquidCrystal для взаємодії з LCD дисплеєм. Для модулів Wi-Fi використовується бібліотека ESP8266WiFi. Також використовується бібліотека SoftwareSerial для забезпечення послідовного зв'язку на будь яких контактах.

2.1.4 Raspberry Pi. Raspberry Pi (рис. 2.6) являє собою потужний одноплатний комп'ютер, створений для навчання програмуванню, електроніці та створення різноманітних цифрових проєктів. Його особливість в тому, що він працює як звичайний комп'ютер, тобто до нього можна підключити монітор, клавіатуру, та іншу периферію, а також зберігати файли на microSD-карті, яка виконує роль жорсткого диска. На відміну від вищезазначених мікроконтролерів, Raspberry Pi має власну операційну систему – Raspberry Pi Os [2], яка дозволяє запускати програми, користуватись інтернетом, тощо. Raspberry Pi має широку підтримку серед спільноти розробників, тому для неї існує безліч інструкцій, бібліотек та готових прикладів [2].

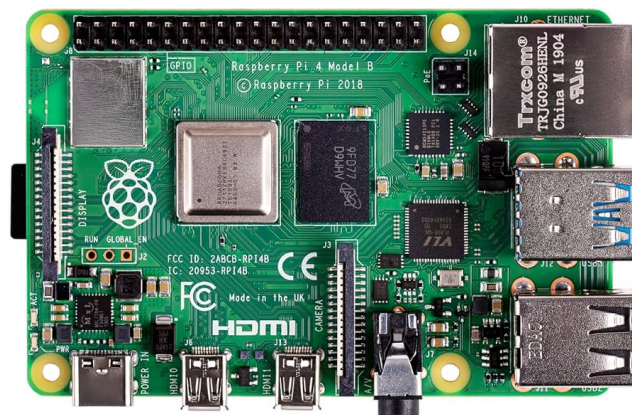


Рисунок 2.6 – Raspberry Pi

На Raspberry Pi теж можна створити зарядний пристрій, але є невеликі відмінності, які треба враховувати при виборі. Оскільки це повноцінний комп'ютер, а не мікроконтролер, він не має вбудованого аналого-цифрового перетворювача (ADC), тому він не здатний напряму зчитувати аналогову напругу чи струм, як Ардуїно наприклад. Також, GPIO виходи не призначені для керування струмом, або на зміни в електричному колі, вони працюють повільніше та менш стабільно. Однак, якщо Raspberry Pi використовувати разом з зарядним модулем TP4056, то вона буде використовуватися тільки для моніторингу даних, з чим вона цілком впорається. Також слід враховувати, що оскільки це повноцінний комп'ютер, його ціна буде відповідно вища, ніж у мікроконтролерів.

2.1.5 ESP32. ESP32 (рис. 2.7) являє собою потужний і універсальний мікроконтролер з вбудованими модулями Wi-Fi та Bluetooth, який ідеально підходить для проєктів IoT. Цей мікроконтролер був розроблений компанією Espressif та став популярним завдяки продуктивності та багатофункціональності. Попередником цього мікроконтролеру був ESP8266, який теж має гарну славу. На відміну від класичних Arduino, ESP32 має два ядра процесора, що працюють на частоті до 240 МГц, але може і поставлятися одноядерним, великий обсяг оперативної пам'яті, а також підтримку гібернації, що дозволяє значно економити енергію. Ця плата має велику кількість цифрових входів/виходів, аналогових входів (ADC), а також цифрово-аналогових перетворювачів (DAC), що дає можливість зчитувати та генерувати аналогові сигнали. Також ця плата підтримує Wi-Fi, що дозволяє підключатися до інтернету, та Bluetooth, за допомогою якого, можна створювати бездротові зв'язки з телефонами.



Рисунок 2.7 – ESP32

ESP32 активно застосовується в переносних гаджетах, розумних пристроях і проєктах, що пов'язані з Інтернетом речей. Його популярність серед ентузіастів і розробників IoT значно зросла, особливо після появи стартових комплектів, таких як ESP32 IoT Starter Kit від Mongoose OS. Завдяки широким можливостям він став одним із найпотужніших і найзручніших рішень для створення як хобі-проєктів, так і серйозних комерційних розробок у сфері інтелектуальних технологій.

В основі ESP32 лежить 32-бітний процесор Tensilica Xtensa LX6, який найчастіше працює в конфігурації з двома ядрами, хоча є й винятки, наприклад модель ESP32-S0WD, яка має лише одне ядро. Мікропроцесор може працювати на частоті до 240 мегагерц і досягати продуктивності до 600 Dhrystone MIPS. Завдяки своїй енергоефективності, ESP32 здатний проводити аналогово-цифрові вимірювання, виконувати обчислення та визначати порогові значення навіть у стані глибокого сну. Це робить його дуже привабливим для енергонезалежних і автономних IoT-проєктів.

2.1.6 STM32. Мікроконтролери серії STM32F103xx із середнім рівнем інтеграції вирізняються високою швидкістю та широким набором можливостей для використання в складних електронних системах. У їхній основі лежить 32-бітне процесорне ядро Arm Cortex-M3, яке здатне працювати з частотою до 72 мегагерц. Ці мікросхеми оснащені внутрішньою пам'яттю,

яка може містити до 128 кілобайт енергонезалежної flash-пам'яті для зберігання програми та до 20 кілобайт оперативної пам'яті. Вони забезпечують підтримку великої кількості периферійних інтерфейсів, які працюють через дві високошвидкісні внутрішні шини. Плата містить в собі два 12-бітних аналого-цифрових перетворювачі, кілька 16-бітних таймерів як для загальних завдань, так і для широтно-імпульсної модуляції, а також різноманітні засоби комунікації: USB, CAN, до трьох портів USART, два SPI та два I2C.



Рисунок 2.8 – STM32F103C8T6

Ці пристрої працюють в діапазоні живлення від 2 до 3,6 вольт, а температурні характеристики витримують в екстремальних умовах: від -40 до $+85$ °C, а в розширеній версії — до $+105$ °C, що є дуже непоганою особливістю. У набір можливостей також входять декілька режимів зниженого енергоспоживання, які роблять ці мікроконтролери хорошим вибором для задач, де критично важлива автономність або економія енергії.

Плати цієї серії можуть бути різних формфакторів — від компактних варіантів із 36 виводами до великих корпусів зі 100 контактами. Загалом серія охоплює весь спектр функціональних можливостей, потрібних у сучасній електроніці, але в деяких моделях може бути реалізована інша конфігурація периферії. STM32F103xx поєднує в собі енергоефективність, гнучкість та

компактність і тому вона часто використовуються в різноманітних медичних пристроях, системах керування двигунами, промисловій автоматизації, GPS-навігаторах, ПЛК-контролерах, ігровій периферії, охоронних системах та іншій електроніці, яка вимагає точності, надійності й багатофункціональності.

2.1.7 Обґрунтування вибору мікроконтролеру. Arduino Uno R3 ідеально підходить для створення системи моніторинга для зарядного пристрою, оскільки він має достатню кількість аналогових виходів для зчитування напруги та температури, просту структуру, підтримку великої кількості бібліотек. Також ця плата легко програмується, має невелику ціну, та сумісна з багатьма модулями. Все це робить Arduino Uno найоптимальнішим вибором для створення ефективного зарядного пристрою.

2.2 Елементна складова зарядного пристрою

Вибір елементної складової зарядного пристрою відіграє ключову роль в підготовці. Усі елементи повинні мати гарну якість, надійність та бути сумісними між собою, оскільки це безпосередньо впливає на стабільність та ефективність роботи системи.

2.2.1 Датчик напруги INA219. Датчик напруги INA219 (рис. 2.9) дозволяє точно вимірювати як напругу на шині живлення, так і струм, що протікає через нього, а також здатний обчислювати потужність в ваттах. Він працює за допомогою цифрового інтерфейсу I²C, тому його можна дуже легко підключити до Arduino Uno, або будь якого іншого мікроконтролера, який підтримує цей протокол [7]. Цей модуль розроблений компанією Texas Instruments, яка спеціалізується на виробленні напівпровідникових елементів, мікросхем, електроніки та виробач на їх основі.

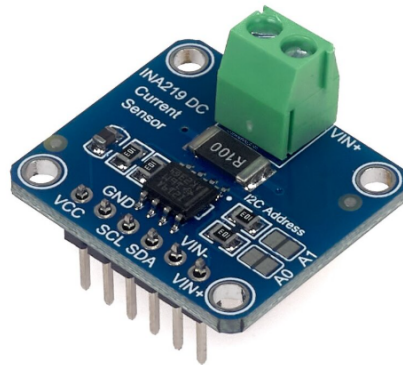


Рисунок 2.9 – Датчик напруги INA219

Датчик напруги вимірює напругу (максимальна підтримувана напруга 26В) на шунтовому резисторі (який зазвичай 0.1 або 0.001 Ом), який знаходиться між джерелом живлення та навантаженням. Після цього, знаючи опір шунта, він обчислює струм за законом Ома. Напруга на шині (на живленні пристрою) також вимірюється окремо. Після того як мікросхема виконала обчислення, вона передає значення струму, напруги та потужності в цифровому вигляді за допомогою протоколу I²C. Датчик дозволяє з високою точністю контролювати енергоспоживання в реальному часі, адже він підтримує до 9 зчитувань за секунду та має роздільну здатність 12 біт.

Для того щоб підключити датчик до Arduino, треба з'єднати чотири основні контакти:

- VCC, який подає живлення на модуль. Підключається до 5V або 3.3V роз'єму;
- GND необхідний для з'єднання землі модуля з Ардуїно. Його треба підключати відповідного роз'єму GND;
- SCL, який подає тактовий сигнал для синхронізації обміну даними. Він підключається до виводу A5;
- SDA підключається до виводу A4, він відповідає за передачу даних між пристроями за допомогою протоколу I²C [7].

Далі підключаємо струм, який буде вимірюватися. Для цього треба підключити Vin+, плюсове живлення, наприклад з модуля TP4056, та Vin-, з якого буде виходити струм, який буде живити акумулятор. Тобто струм через Vin+ протікає через вбудований в модуль шунт, де вимірюється його властивості, а виходить він на Vin-.

Отже, INA219 має суттєві переваги, з яких є одночасне вимірювання току та напруги, що дає можливість обчислювати споживану потужність без додаткових обчислень у мікроконтролері, він є енергоефективним, компакним і точним. Це і робить його популярним вибором для моніторингу живлення електроніки, ІОТ-проектів, тощо. Для зарядного пристрою він підходить ідеально.

2.2.2 Модуль TP4056. TP4056 (рис. 2.10) призначений для керування зарядом літій-іонних та літій-полімерних акумуляторів з напругою 3.7В. Самостійна реалізація логіки заряджання таких акумуляторів є складною, оскільки вони дуже чутливі до перезаряду, та до характеристик струму, який живить їх, тому і існує ця мікросхема, оскільки вона має вбудовану, свідомо справну, логіку заряджання. Тому при використанні цієї мікросхеми, можна буде не хвилюватися про безпеку. Цей модуль може постачатися як з мікросхемою захиста, так і без неї. За це відповідає позначення “with protection” [6]. У цьому проекті буде використовуватися варіант з захистом, який відповідає за захист від перезаряду, глибокого розряду та короткого замикання. Також мікросхема підтримує заряд у режимі CC/CV, що відповідає всім вимогам стандартної безпечної зарядки літійових елементів. TP4056 часто доступна з micro-USB та Type-C портом. Ця мікросхема є однією з найпопулярніших BMS систем завдяки своїй низькій вартості, багатьом захистним механізмам та надійності. Живиться плата від 4.5-5.5В блоку живлення через USB порт, або через контакт IN+[3].

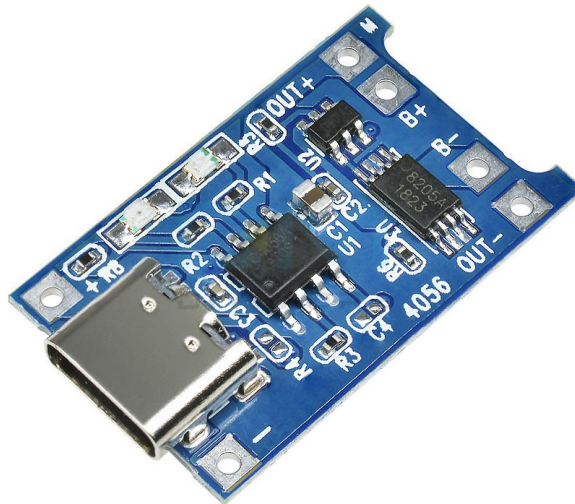


Рисунок 2.10 – Модуль tp4056

TP4056 заряджає акумулятор у двох основних фазах:

- спочатку, якщо акумулятор розряджений, зарядка починається з режиму постійного струму, тобто подається фіксований струм, зазвичай до 1А (залежно від встановленого резистора на платі);
- коли напруга на акумуляторі досягає приблизно 4.2 В, мікроконтролер переходить в режим заряджання постійної напруги CV, та поступово зменшує струм, поки він не впаде до порогового рівня, та після цього моменту зарядка завершується [6].

Тому, зазвичай, можна побачити такий ефект, що акумулятор до 80 відсотків заряджається швидко, а після, дуже повільно. Цей механізм використовується для збереження терміну придатності акумулятору, адже теоретично, його можна зарядити за кілька секунд виоким током, але це дуже сильно йому нашкодить, та він не буде працювати після цього.

Підключається модуль до Arduino дуже просто, все що треба зробити, це підключити живлення по USB, з'єднати IN- та GND на ардуїно та все. Далі треба просто підключити акумулятор. Це робиться за допомогою BAT+ та BAT- контактів, які підключаються до плюсу та мінусу акумулятора відповідно. Також на деяких версіях модуля присутні контакти OUT+ та OUT-,

з яких можна знімати напругу, коли акумулятор повністю заряджений.

Модуль TP4056 дозволяє змінювати струм заряджання. Для цього потрібно змінити номінал резистора між виводом PROG та землею. Наприклад з резистором номіналом 1.2 кОм, струм на виході буде 1А, а з номіналом 10 кОм, струм буде 130мА [3]. Це дозволяє підібрати резистор під конкретну ємність акумулятора, щоб уникнути перегріву, в разі малої ємності, або пришвидчити час заряджання, в разі великої ємності.

Отже, модуль TP4056 має ряд переваг. Серед них компактність та доступна ціна, що дозволяє використовувати його в бюджетних та невеликих проектах, простий у використанні, має версії з захистом, та підтримує повноцінний заряд літєвих акумуляторів. Серед недоліків, він не має додаткових датчиків та підтримує зарядку акумуляторів з напругою більш ніж 4.2 вольти, але в рамках цього проекту, ці недоліки ніяк не вплинуть на ефективність та надійність виробу. Тому усе це робить мікроконтролер tp4056 оптимальним вибором в рамках зарядного пристрою для Li-ion батареї, так як він відповідає за безпечне заряджання акумулятора, а усе інше бере на себе Arduino.

2.2.3 LCD дисплей. LCD дисплей (рис. 2.11) дозволяє виводити інформацію про значення напруги, струму, стан зарядки акумулятора, його температуру, у вигляді тексту. Для відображення символів використовуються рідкі кристали. Вони складаються з матриці символів, кожен з яких формується сіткою 5x8 пікселів. В середині дисплея рідкі кристали реагують на електричний струм та змінюють свою прозорість, завдяки чому підсвітка, або світло від фону, створює видимість тексту. Дисплей має зелено-жовте LED підсвічування і регулятор контрастності, підключений через змінний резистор. Дисплей використовує контролер HD44780, який керується за допомогою 8-бітного паралельного інтерфейсу. У стандартному підключенні він займає від шести до десяти пінів Arduino, але з I²C адаптером, тобто на чипі PCF8574 можна підключити дисплей лише двома проводами — SDA і SCL, що значно економить місце для інших компонентів.



Рисунок 2.11 – LCD дисплей

За допомогою LCD дисплея, користувач буде мати змогу в реальному часі контролювати процес заряджання акумулятора і вчасно реагувати на будь які відхилення. За допомогою дисплею будуть відображатися інформація про напругу на акумуляторі, сила току заряджання, стан зарядки, кількість циклів зарядки, тривалість заряджання, та інші діагностичні повідомлення. Тому усе це робить LCD дисплей дуже ефективним і економічним вибором в рамках створення зарядного пристрою.

2.3 Електрична функціональна схема пристрою

Схема (рис 3.7) відображає взаємодію усіх модулів створюваного пристрою. За процес заряджання акумулятора відповідає модуль tp4056, який підключений до модуля вимірювання напруги ina219, та через нього до акумулятора. Arduino виконує функції збирання, обробки та відображення інформації про стан та умови заряджання.

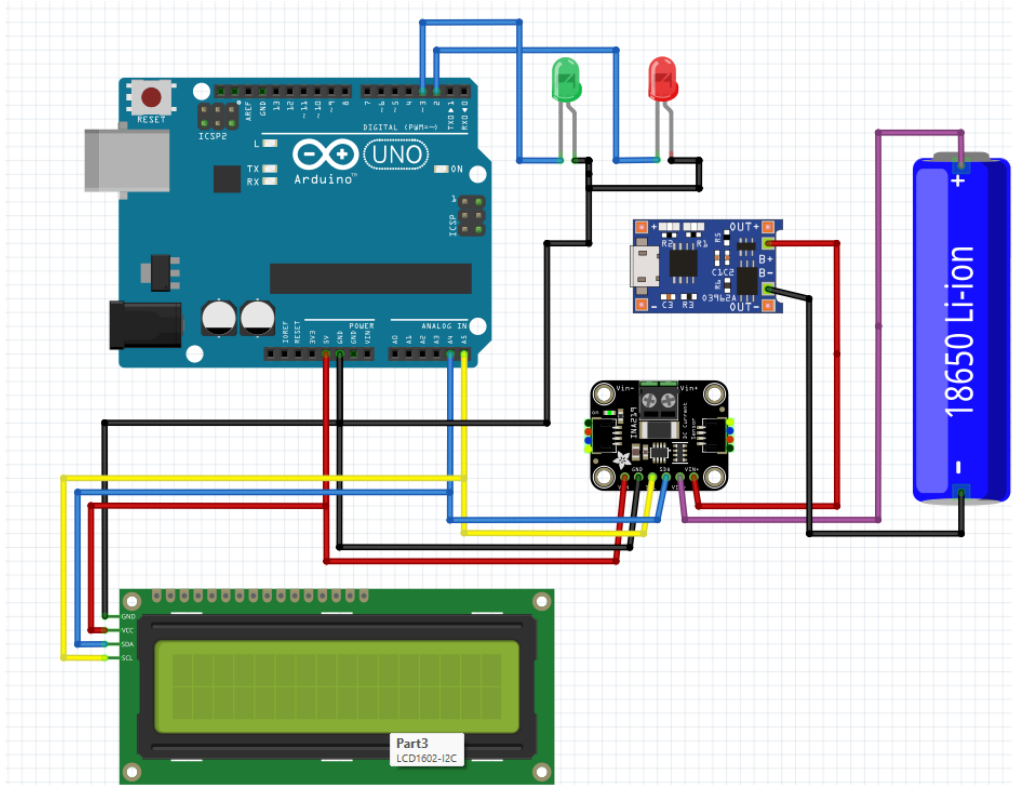


Рисунок 2.12 – Електрична функціональна схема зарядного пристрою

Як і було згадано вище, відповідати за процес зарядки буде модуль tp4056. Він має два контакти, які треба підключити: V+ та V-. Плюсовий підключається спочатку до контакту Vin+ модуля ina219, а потім до акумулятора з контакту Vin-, для того щоб відстежувалася точна напруга та сила струму під час заряджання. Мінусовий контакт підключається напряму до акумулятора. Для того щоб процес заряджання запусився, треба просто підключити USB до модуля tp4056. Ina219 також підключений до Ардуїно, за допомогою I²C шини, за допомогою якої передається вищезазначена інформація. Ардуїно, в свою чергу, збирає інформацію з модуля ina219, обробляє її, та виводить на LCD дисплей, який також підключений за допомогою I²C шини.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЕКТУ

3.1 Вибір мови програмування

Офіційно Ардуїно підтримує тільки C/C++ та Arduino Language, але спільнота створила декілька додаткових засобів для взаємодії, за допомогою інших мов програмування. Деякі з них можуть бути легше за інші та забезпечити більш швидкий процес розробки, тому дуже важливо обрати найпідходящий інструмент для програмування мікроконтролеру.

3.1.1 C/C++ та Arduino Language. Arduino Language являє собою Arduino Language являє собою спрощену версія C/C++. Оскільки ці мови програмування доволі складні, вона була створена для легкого програмування мікроконтролерів Arduino. Іншими словами, AL це розширення C++ [10], яке містить набір вбудованих функцій, що полегшують роботу з апаратними компонентами, такими як світлодіоди, датчики, двигуни та інші пристрої. Програмування відбувається у середовищі Arduino IDE, де код структурується в `setup()` функцію, для ініціалізації, та `loop()` функцію, для виконання основного циклу програми [9].

Цей підхід дозволяє уникнути багатьох складностей традиційного програмування на C/C++, оскільки більшість технічних деталей, таких як налаштування компілятора, конфігурація мікроконтролера та робота з низькорівневими регістрами, приховані від користувача. Натомість розробники зосереджуються на логіці роботи своїх проектів: керування пінами вводу/виводу, обробка сигналів з датчиків, передача даних через серійний порт, слідкування за пам'яттю, тощо. Arduino також підтримує велику кількість бібліотек, які розширюють можливості програмування – від роботи з дисплеями до керування складними комунікаційними протоколами. Також, хто завгодно може написати свою бібліотеку на мові C++, якщо має бажання [10].

Arduino Language став популярним завдяки простоті та зручності, в порівнянні з C++, що дозволяє навіть новачкам швидко створювати прототипи електронних пристроїв [9]. Програмування у цьому стилі не вимагає глибокого розуміння комп'ютерної архітектури або мови C++ у повному обсязі. Завдяки цьому ця мова програмування стає гарним вибором для використання у навчальних курсах, хобі-проектах та у деяких комерційних розробках для швидкого створення прототипів і тестування концепцій.

3.1.2 JavaScript (Espruino). JavaScript є одною з найпопулярніших мов програмування, яка широко використовується серед розробників серверних додатків, та, особливо, клієнтських веб-додатків. Тому створення інструменту для взаємодії з Ардуїно на цій мові програмування зовсім не дивує.

Espruino – це середовище виконання для JavaScript, розроблене спеціально для мікроконтролерів, включно з платами Arduino. На відміну від Arduino Language, який компілює код у машинний код мікроконтролера, Espruino виконує JavaScript-код прямо на пристрої в режимі інтерпретатора. Це дозволяє писати і запускати код без попередньої компіляції – достатньо надіслати його через серійний порт або веб-інтерфейс, і пристрій миттєво почне виконувати інструкції. З цього випливає, що код написаний на JS, буде працювати повільніше, та буде споживати більше пам'яті, що, в деяких випадках, наприклад при великому обсязі коду та складній логіці, може бути критичним.

Особливістю цієї технології є підтримка JavaScript у стандартному стилі ES5 (ECMAScript 5), хоча з деякими обмеженнями через недостатні ресурси мікроконтролерів. Це означає, що можна використовувати знайомі конструкції мови JS, такі як функції, об'єкти, масиви, замикання та навіть події, що значно спрощує написання коду для багатьох програмістів, які вже працювали з JavaScript у веб-розробці. Наприклад, керування світлодіодами чи зчитування даних із сенсорів можна робити у звичному JavaScript-стилі: `digitalWrite(LED1, 1);` або `setInterval(function() { ... }, 1000);`.

Щоб використовувати Espruino, потрібно завантажити спеціальну

прошивку Espruino на підтримувану плату. Після цього код можна завантажувати прямо в плату через Espruino Web IDE або серійний термінал. На відміну від Arduino, де код компілюється на комп'ютері, у Espruino код виконується прямо на мікроконтролері в режимі інтерпретації, що дозволяє миттєво тестувати зміни. Однак це також обмежує продуктивність і пам'ять програми, тому, як і було зазначено вище, для складних проектів або високопродуктивних завдань Espruino може бути менш ефективним порівняно з традиційним C++.

3.1.3 MicroPython. MicroPython — це спеціальна версія мови програмування Python, адаптована для роботи на мікроконтролерах. Як і в випадку з Espruino, оскільки Python теж інтерпретована мова, вона потребує інтерпретатора, який повинен бути заздалегідь завантаженим на пристрої. Програма буде виконуватися в режимі реального часу, за допомогою інтерпретатора, тому після завантаження прошивки MicroPython на плату, можна підключитися через серійний порт або WebREPL і писати та змінювати код у режимі реального часу.

Python на мікроконтролерах зберігає більшість своїх конструкцій, тому можна писати зрозумілий і лаконічний код для управління пінами, обробки даних з датчиків, роботи з протоколами, наприклад, I2C або SPI. Ця мова програмування дуже проста, в порівнянні з C++, та зрозуміла для новачків. Наприклад, щоб засвітити світлодіод, потрібно лише кілька рядків коду: `from machine import Pin; led = Pin(2, Pin.OUT); led.on()`. Це і робить MicroPython ідеальним для навчання та швидкого прототипування, особливо для тих, хто вже знайомий з Python.

Однак у використанні Python для мікроконтролерів є й свої обмеження, такі ж самі, як і для JS. Інтерпретатор займає частину пам'яті пристрою, а продуктивність Python нижча, ніж у C/C++, тому складні або високошвидкісні завдання можуть бути не такими ефективними. Ця технологія потребує потужних плат, які використовують продуктивний мікроконтролер, та мають достатній обсяг пам'яті для завантаження інтерпретатора, наприклад ESP32,

ESP8266 або Raspberry Pi Pico. Але незважаючи на це, популярність Python серед розробників робить його чудовим вибором для простих проектів, де зручність та швидкість розробки важливіші за максимальну ефективність.

3.1.4 Обґрунтування вибору мови програмування. Arduino Language є основною мовою програмування для Arduino. Вона базується на C/C++, тому має ряд переваг над іншими мовами. Найважливіше це здатність до компіляції коду. Тобто код на комп'ютері, перед завантаженням на плату, компілюється в машинний код, який розуміє Ардуїно, за допомогою компілятора. Через це, для виконання коду не потрібен інтерпретатор, який сам по собі займає певну кількість ресурсів (як пам'яті, так і процесорного часу), тому використання ресурсів буде мінімальним, а швидкодія великою. Єдиним суттєвим недоліком може бути те, що деякі початківці можуть не знати основи C/C++, тому будуть витратити більше часу на освоєння цієї мови програмування та на написання коду.

Використання цієї мови програмування дозволяє повністю контролювати апаратні можливості мікроконтролера Arduino. Це надає можливості безпосередньо керувати виводами/входами, використовувати різні датчики та пристрої, інтегрувати з різними протоколами зв'язку та багато іншого.

Arduino Language має дуже велику спільноту розробників, тому і має велику кількість різних бібліотек, які можуть бути корисними в подальшому використанні пристрою.

Отже, мова програмування Arduino Language є найоптимальнішим вибором у рамках створення зарядного пристрою, оскільки вона має ряд переваг, серед яких є висока продуктивність, велика кількість бібліотек та простота взаємодії.

3.2 Вибір середовища розробки

Для програмування коду для Ардуїно існують різні редактори коду, які

підтримують Arduino Language. При виборі, треба звертати увагу перш за все на простоту використання, та кількість функціоналу для взаємодії з платою, бо це напряду може вплинути на швидкість та ефективність розробки.

3.2.1 Arduino IDE. Arduino IDE — це середовище розробки програмного забезпечення, створене для роботи з мікроконтролерами з платформи Arduino. Воно надає зручний інтерфейс для написання, компіляції та завантаження коду на мікроконтролери. Arduino IDE підтримує мову програмування, засновану на C/C++, яка називається Arduino Language [4], і містить велику кількість бібліотек, інструментів та прикладів готового коду для спрощення розробки проєктів. Ця програма доступна для різних операційних систем, таких як Windows, macOS та Linux. Вона також дозволяє розширювати початковий функціонал через додаткові плати та бібліотеки. Завдяки своїй простоті та відкритому коду, ця програма стала ключовим інструментом для навчання, прототипування і створення інтерактивних електронних систем. Для того щоб працювати з Ардуїно, треба просто підключити його по USB-кабелю, далі програма автоматично встановить усі необхідні драйвери, та далі можна приступати до написання коду. Після того як код був написаний, щого треба спочатку скомпілювати, натиснувши на кнопку Verify, а потім завантажити на мікроконтролер, для цього є кнопка Upload. Після успішного завантаження, Ардуїно перезавантажиться та почне виконувати код [4].

3.2.2 PlatformIO. PlatformIO являє собою розширення для популярних редакторів коду, наприклад Visual Studio Code, підтримує велику кількість платформ, фреймворків та плат. Ця програма надає сучасні можливості для програмування, включаючи автоматичне завершення коду, перевірку синтаксису, підтримку системи контролю версій, інтегровану систему керування бібліотеками та багатоплатформне компілювання. Завдяки використанню конфігураційних файлів, проєкти можна легко налаштовуювати, середовище підходить як для аматорських, так і для професійних застосувань. PlatformIO також підтримує відлагодження коду, автоматичне тестування та безперервну інтеграцію. Це робить його особливо

привабливим для складних проєктів та командної роботи. Основною відмінністю від Arduino IDE є його багатofункціональність та підтримка великої кількості мікроконтролерів, а не тільки Arduino IDE. В PalfomIO є підтримка автодоповнення, підсвічування синтаксису, модулі система контролю версій. Тому ця програма є більш переважливою в випадку розробки великого проєкту, з декількома розробниками.

3.2.3 Обґрунтування вибору. ArduinoIDE є оптимальним вибором в рамках створення зарядного пристрою для літєвих батарей, оскільки цей проєкт не є великим, а отже ця програма має усе необхідне для швидкого створення, та завантаження коду на Ардуїно. ArduinoIDE не потребує складного налаштування, все що потрібно, це просто підключити плату, написати код та завантажити його на мікроконтролер, і все буде працювати. Ця програма не має великої кількості функціоналу, легко встановлюється та швидко працює, а отже робота з ним буде легкою та швидкою. В тих хто вже має досвід в програмуванні на C/C++, не виникне ніяких проблем в написанні коду.

3.3 Алгоритм роботи зарядного пристрою

На схемі (рис. 3.1) зображено алгоритм роботи зарядного пристрою на основі мікроконтролеру. Спочатку треба під'єднати акумулятор до схеми, а далі можна під'єднувати живлення. Tr4056 обов'язково треба під'єднати до мережі після Arduino, оскільки спочатку мікроконтролер повинен встановити відсоток заряду акумулятора. Це потрібно для правильного розрахунку часу заряджання та обчислення відсотку під час процесу. Після того як живлення було під'єднане, акумулятор почне заряджатися, а мікроконтролер буде збирати та оброблювати дані про процес.

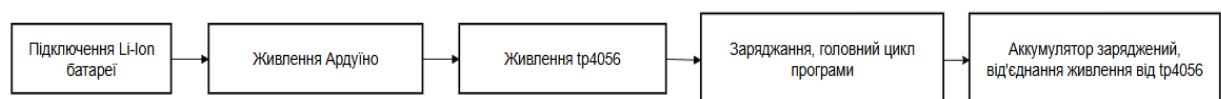


Рисунок 3.1 – Алгоритм роботи зарядного пристрою

3.4 Опис програмного коду проекту

Для початку, одним з найважливіших етапів створення будь якої програми, це підключити бібліотеки, які дозволяють працювати з тими чи іншими компонентами.

Лістинг 3.1 – Підключення бібліотек

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Adafruit_INA219.h>
```

В даному проекті використовуються три бібліотеки. Бібліотека Wire, напряду не використовується в коді, але потрібна для правильної роботи інших бібліотек, тому її обов'язково треба включити. Бібліотека LiquidCrystal реалізує функції для роботи з рідко-кристалічним дисплеєм. За допомогою цієї бібліотеки виводиться інформація на дисплей в реальному часі.

Лістинг 3.2 – Створення змінних для підключених бібліотек.

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
Adafruit_INA219 ina219;
```

Для початку треба створити глобальну змінну для об'єкту LiquidCrystal_I2C. Для цього треба передати інформацію про дисплей. Треба вказати адресу i2c шини дисплею, кількість символів в одному рядку, та кількість стовпців. В даному випадку адресом буде 0x27, кількість рядків – 16, стовпців – 2 (ліст. 3.2). Щоб почати працювати з дисплеєм в функції setup() треба визвати метод init() (ліст. 3.5).

Для бібліотеки Adafruit_INA219 також створюємо змінну (ліст. 3.2) та викликаємо метод begin() в функції setup() (ліст. 3.5). В даному випадку не треба вказувати аргументи, бібліотека буде використовувати стандартні значення.

Далі створюємо глобальні змінні для виконання логіки проекту (ліст. 3.3). Змінна `chargePercent` зберігає інформацію про те, наскільки акумулятор заряджений в відсотках, вона буде доступна в кожній функції. Флаг `isInitialized`, потрібен для того, щоб один раз встановити відсоток заряду акумулятора, як тільки він був підключений до `ina219`. Змінна `lastUpdateTime` необхідна для підрахунку відсотку заряджання під час процесу.

Лістинг 3.3 – Глобальні змінні

```
short chargePercent;
bool isInitialized;
unsigned long lastUpdateTime = 0;
```

Наступним кроком буде оголошення основних функцій проекту (ліст. 3.4). Спочатку оголошення, а потім в кінці реалізація. Функція `calculateChargingPercent` виконується тільки коли був підключений акумулятор, та не підключене живлення до `tr4056`. Ця функція підраховує відсоток заряду в акумуляторі, за допомогою вольтажу, яке він має. Також ця функція приймає ток, щоб встановити що акумулятор точно не заряджається. Функція `calculatePercentageByTime` приблизно підраховує на скільки відсотків був заряджений акумулятор, після того як живлення було подане на зарядний контроллер. Цю функцію треба викликати тільки коли функція `isCharging` повертає значення `true`. Функція `updateIna219` оновлює дані з модуля, та виводить їх на дисплей та в `Serial` монітор. Також тут викликається функція `updateLights()`, яка, на основі проценту зарядки акумулятора та току, встановлює який LED-індикатор підсвічувати – зелений, або червоний. Відповідно, якщо акумулятор має 100% заряду, то світиться зелений, якщо в процесі заряджання, то червоний, а якщо акумулятор не заряджається, то ніякий індикатор не горить.

Лістинг 3.4 – Основні функції проекту

```
short calculateChargingPercent(float voltage, float
```

```

current_mA);
void calculatePercentageByTime();
bool isCharging(float current_mA);
void updateIna219();
void updateLights(short percentage, float current_mA);
void clearSerial();

```

Як і всі програми для Ардуїно, програмний код містить функції `setup()` та `loop()`.

Лістинг 3.5 – Функція `setup()`.

```

void setup() {
  chargePercent = 0;
  isInitialized = false;

  pinMode(redLED, OUTPUT);
  pinMode(greenLED, OUTPUT);
  lcd.init();
  lcd.backlight();
  Serial.begin(9600);
  if (! ina219.begin()) {
    Serial.println("Failed to find INA219 chip");
    while (1) { delay(10); }
  }
}

```

Перша функція визивається тільки один раз, під час ініціалізації, а друга циклічна, і визивається завжди, поки є живлення на мікроконтролері [8].

В функції `setup()` (ліст. 3.5) спочатку ініціалізуємо глобальні змінні, далі встановлюємо для цифрових пінів 2 та 3 режим «Output», для того щоб вони могли виводити напругу, та керувати LED-індикаторами . Далі ініціалізуємо lcd дисплей, включаємо backlight, ініціалізуємо Serial монітор на швидкості 9600, та ініціалізуємо модуль ina219. Якщо ina219 не ініціалізувався, код не запуститься, а буде очікувати на його підключення.

Лістинг 3.6 – Функція `loop()`

```

void loop() {
  clearSerial();
  updateIna219();
}

```

```
    delay(1000);  
}
```

Функція `loop()` (ліст. 3.6) буде викликатися раз в секунду (через встановлений `delay`). Тут при кожному виклику викликається метод `clearSerial()`, який очищає послідовний монітор, та метод `updateIna219`, який оновлює дані.

Таким чином програмний код забезпечує надійний моніторинг інформації під час заряджання акумулятору, та виводить інформацію в послідовний монітор та рідко-кристалічний дисплей.

4 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ЗАРЯДНОГО ПРИСТРОЮ

4.1 Складання апаратної частини зарядного пристрою

У процесі виконання роботи було розроблено зарядний пристрій на основі мікроконтролера TP4056 та мікроконтролером Arduino Uno R3. Було детально розглянуто існуючий пристрій, проаналізована його робота, та на основі цих даних, була зібрана система, яка включає модуль TP4056, INA219, резистори на 220 Ом, LED-індикатори, Ардуїно, та з'єднувальні компоненти. Розроблений пристрій (рис. 4.1) дозволяє швидко та безпечно заряджати літієві акумулятори, а також збирати інформацію про стан батареї, наприклад відсоток заряду. Додатково були встановлені зелений та червоний світлодіоди, за допомогою яких можна відображати етап заряджання. Для того щоб понизити напругу від цифрових пінів Ардуїно, які видають 5 вольт, було використано резистор на 100 Ом для червоного, оскільки для нього потрібна напруга до 3.3 вольт, та 220 Ом для зеленого, він працює з напругою 2.5-2.7 вольти.

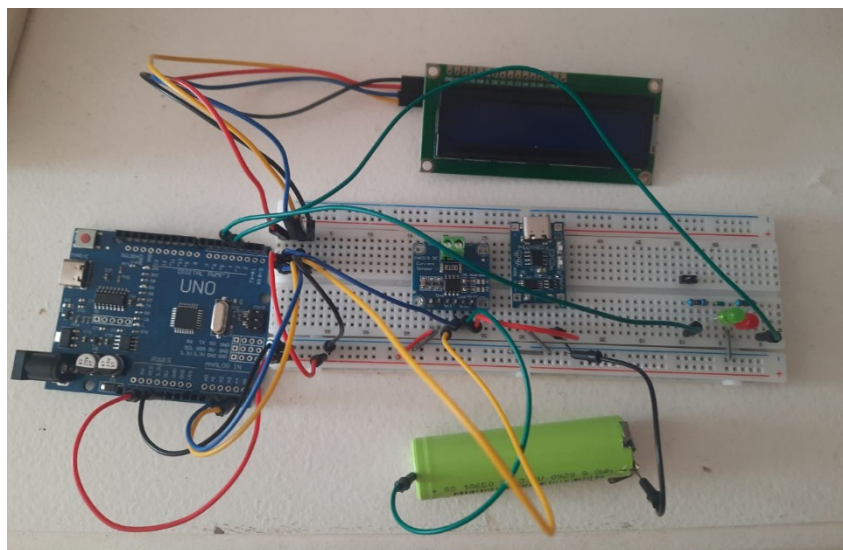


Рисунок 4.1 – Зібраний зарядний пристрій

4.2 Тестування зарядного пристрою

Для тестування був обраний 18650 літій-іонний акумулятор, який має ємність приблизно 2500 mAh. Спочатку визначаємо наскільки акумулятор заряджений за допомогою мультиметра.

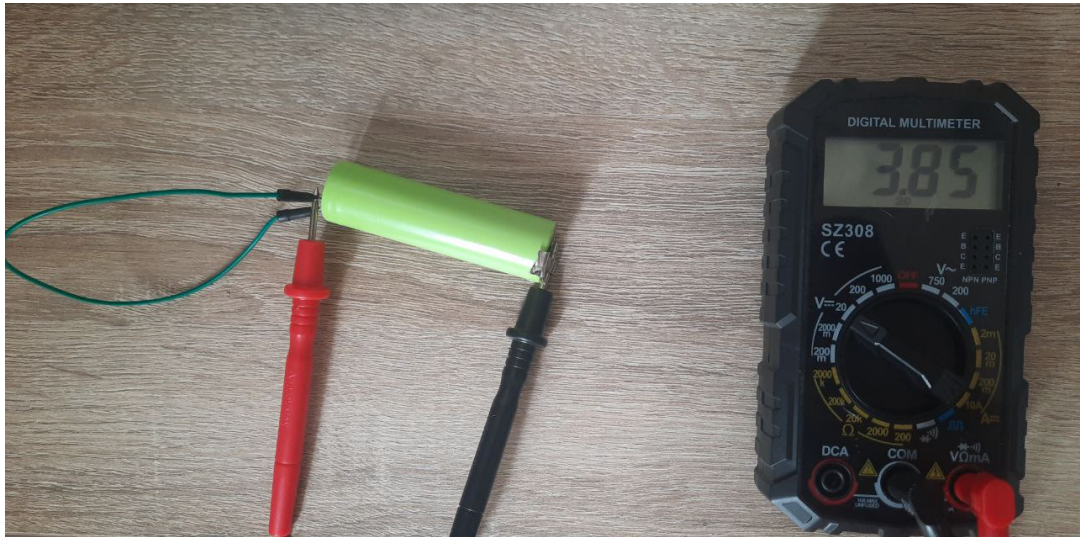


Рисунок 4.2 – Вимірювання вольтажу акумулятора

На рис. 4.2 можна побачити, що акумулятор видає 3.85 вольтів, що приблизно відповідає 59% заряду. Для того щоб зарядити, треба підключити його до схеми. Після цього чекаємо, поки дисплей не буде відображати 100% заряду, та не загориться зелений світлодіод (рис. 4.3) [5]. Було встановлено, що повне зарядження акумулятору займає приблизно дві с половиною години, тому при обчисленні відсотку зарядження враховується саме цей час. Тобто для кожної ємності акумулятора підрахування відсотку заряду буде відрізнятися.

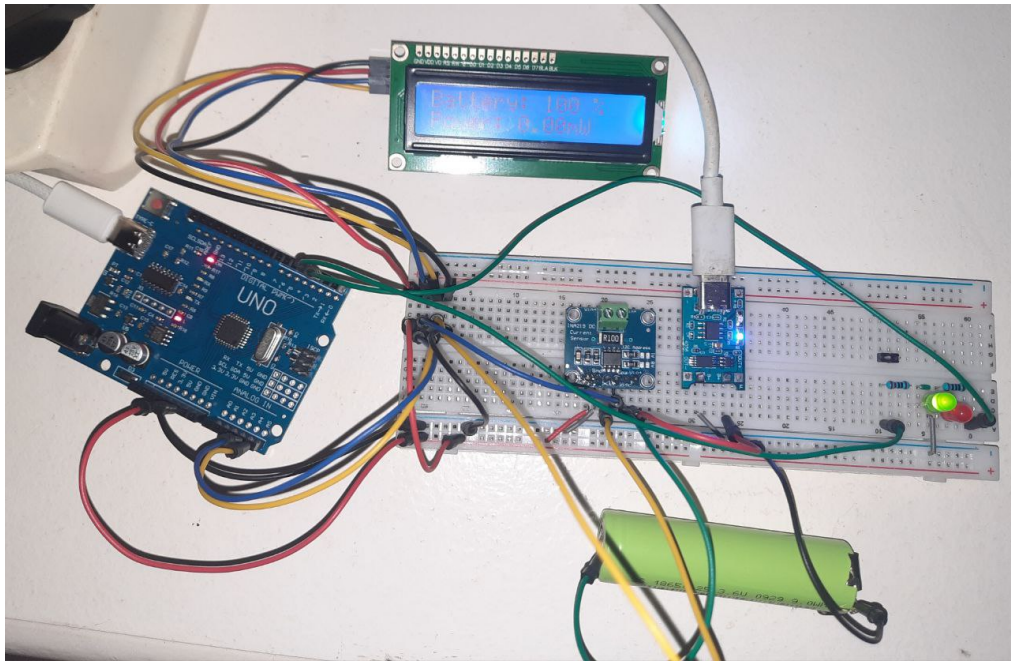


Рисунок 4.3 – Процес заряджання завершився

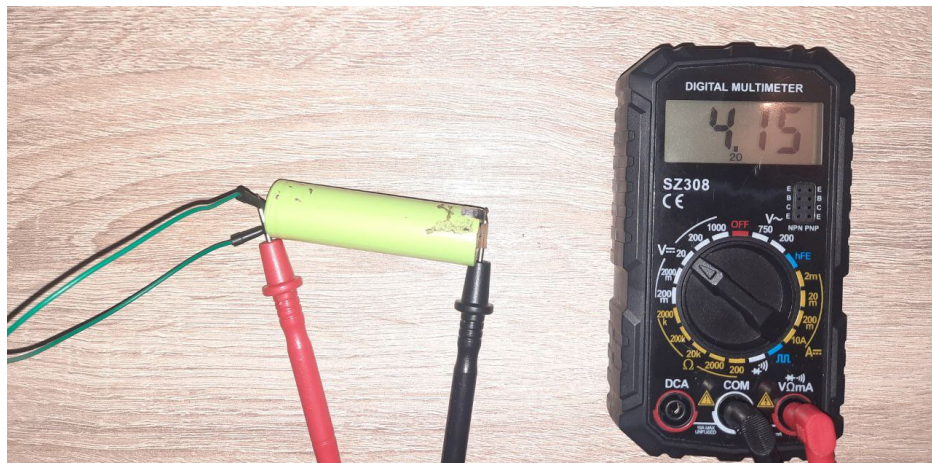


Рисунок 4.4 – Результат заряджання акумулятору

На рис. 4.4 можна побачити, що акумулятор був повністю заряджений. Горить зелений індикатор, та на дисплеї відображається 100% заряду, та 0 mW споживання, що означає про успішне повне заряджання, та про те, що живлення акумулятора було вимкнене, тому він не буде перезаряджатися. Після проведення тесту напруги мультиметром, було підтверджено результат. Тепер акумулятор повністю заряджений, з урахуванням усіх особливостей

літєвих акумуляторів. В таких умовах заряджання він майже не буде деградувати, не буде мати небезпечних властивостей, наприклад займання, та перегріватися, що свідчить про високу надійність та довготривалість системи.

ВИСНОВКИ

Актуальність кваліфікаційної роботи пояснюється суттєвими перевагами літієвих акумуляторів, що використовуються у великій кількості портативних девайсів, а зарядний пристрій є невід'ємною частиною таких засобів.

У результаті виконання кваліфікаційної роботи було спроектовано та реалізовано зарядний пристрій на основі мікроконтролеру. Отриманий результат здатний забезпечувати безпечно та ефективно заряджання акумулятору, має потенціал до вдосконалення, та може бути вбудованим в велику кількість різних систем, які потребують автономної роботи.

Було проведено огляд існуючих зарядних пристроїв, проаналізовано їх переваги та недоліки, що дозволило визначити оптимальний підхід до створення проекту. В результаті, був обраний зарядний модуль TP4056, як основний зарядний пристрій, та мікроконтролерна платформа Arduino Uno, яка відповідає за керування та обробку даних. Цей вибір обґрунтовується перш за все безпекою, оскільки літієві акумулятори можуть бути небезпечними в разі неправильного використання, низькою вартістю, ефективністю, а також широким спектром функціональності.

Розроблено програмне та апаратне забезпечення пристрою, написаний програмний код для Arduino Uno, який зчитує дані та підраховує заряд в акумуляторі. Також було розроблено та реалізовано апаратну схему, виконано її монтаж та тестування.

Створено макетний пристрій, який по функціональності не відрізняється від реального, може бути модифікований та вбудований, наприклад, у power bank. Пристрій має потенціал до вдосконалення, оскільки його розмір може бути зменшений, а функціонал доповнений, наприклад можна більш точно вимірювати відсоток заряду в акумуляторі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Документація Arduino Uno [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.arduino.cc/hardware/uno-rev3/>
2. BCM2711 - Peripherals [Електронний ресурс] – Режим доступу до ресурсу: <https://datasheets.raspberrypi.com/bcm2711/bcm2711-peripherals.pdf>
3. TP4056 [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.circuitdesigner.com/component/d3a17ee0-5b4e-5e5b-9008-cbac1cd40de8/tp4056>
4. Принципи програмування в Arduino IDE [Текст] : тез. доп. наук.-практ. конф. / ред. Рабінко А. В. – Вінниця: 2017
5. Mad Avsik Programming Digital RGB LED Strips with Arduino and the FastLED Library [Текст] / М. Avsik // Журн. Norwegian creations – 2012. – Arduino Tutorials – 16.01.2018.
6. TP4056 Lithium Ion Battery Charger Datasheet [Текст] – *NanJing Top Power ASIC Corp.*
7. INA219 Datasheet – *Texas Instruments* [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ti.com/lit/gpn/INA219>
8. Практичні приклади з Arduino, ESP та сенсорами [Електронний ресурс] – Режим доступу до ресурсу: [Random Nerd Tutorials | Learn ESP32, ESP8266, Arduino, and Raspberry Pi](#)
9. Петин В. А. Проекты с использованием контроллера Arduino (2-е издание) [Текст] / В. А. Петин. – БХВ : Петербург, 2015. – 464 с.
10. Васильев О. Програмування на C++ в прикладах і задачах [Текст] : навч. посіб. / О. Васильев. – Київ : Видавництво Ліра-К, 2017. – 382 с.