

УДК 544.6,621.351/355,620.22,620 544.6 621.351/355  
УКПП  
№ держреєстрації 0119U100763  
Інв. №

Міністерство освіти і науки України  
Сумський державний університет  
40007, м. Суми, вул. Римського-Корсакова, 2, (0542) 334108

ЗАТВЕРДЖУЮ  
Проректор з наукової роботи  
д-р.фіз.-мат.наук, професор

\_\_\_\_\_ А.М. Черноус

ЗВІТ  
ПРО НАУКОВО-ДОСЛІДНУ РОБОТУ

Закономірності формування нанопористих ZnO, C, C/ZnO і ZnO/NiO для потенційного застосування у якості електродів літій-іонних акумуляторів

**ВСТАНОВЛЕННЯ ХАРАКТЕРИСТИК ФУНКЦІОНУВАННЯ ЛІТІЙ ІОННИХ АКУМУЛЯТОРІВ З ЕЛЕКТРОДАМИ НА ОСНОВІ ОТРИМАНИХ ПОРИСТИХ СИСТЕМ C, ZnO, C/ZnO І ZnO/NiO (остаточний)**

Керівник НДР,  
канд. фіз.-мат. наук., с.н.с.

Г.С. Корнющенко

2021

Рукопис закінчено 29 листопада 2021 р.

Результати роботи розглянуті науковою радою СумДУ, протокол від 25 листопада 2021 р. № 5

**СПИСОК АВТОРІВ**

|   |            |   |
|---|------------|---|
| Керівник НДР,<br>пров. наук. співроб.,<br>канд. фіз.-мат. наук              | 28.12.2021 | Г.С. Корнющенко<br>(реферат, вступ, розділи 1,<br>2, 4, висновки) |
| Відповідальний виконавець<br>старш. наук. співроб.,<br>канд. фіз.-мат. наук | 28.12.2021 | В.В. Наталіч<br>(реферат, підрозділ 4.2)                          |
| Виконавці:<br>Старш. наук. співроб.,<br>канд. техн. наук                    | 28.12.2021 | В.В. Нагорний<br>(розділ 3)                                       |
| Старш. наук. співроб.,<br>канд. техн. наук                                  | 28.12.2021 | Ю.В. Парфененко<br>(розділ 3)                                     |
| Старш. наук. співроб.,<br>канд. фіз.-мат. наук                              | 28.12.2021 | О.А. Мокренко<br>(розділ 2 )                                      |
| Аспірант  | 28.12.2021 | С.Т. Шевченко<br>(підрозділ 2.2, розділи 1, 4)                    |

## РЕФЕРАТ

Звіт про НДР: 107 с., 53 рис., 65 джерел.

ЛІТІЙ-ІОННІ АКУМУЛЯТОРИ, КОНДЕНСАЦІЯ, ЕЛЕКТРОДИ, ОКСИД ЦИНКУ, ВУГЛЕЦЬ, НАНОПОРИСТІ КОМПОЗИТИ, СТРУКТУРОУТВОРЕННЯ.

**Об'єктом дослідження** є процеси формування пористих наносистем вуглецю, окислу цинку, оксиду нікелю, та композитних матеріалів при конденсації зазначених матеріалів за умов, наближених до термодинамічної рівноваги, а також процеси, що відбуваються в літій-іонних акумуляторах при використанні в них в якості електродів отриманих пористих систем.

**Метою роботи** полягала у формуванні високопористих електродів вуглецю та окислу цинку та композиту на основі цих матеріалів з морфологією, структурою, елементним складом та товщиною, що визначають підвищення електрохімічних характеристик літій-іонних електрохімічних акумуляторів.

**Основні результати:** На основі технологічного підходу, ґрунтованого на самоорганізації малих відносних пересичень осаджуваного пару, вперше отримані пористі наносистеми Zn і Ni, які в подальшому використані в якості прекурсорів для нанесення пористого турбостратного графіту. На основі проведених комплексних досліджень отриманих пористих композитів C/Zn і C/Ni, C/ZnO і C/NiO показано, що на локальних ділянках нарощуваного турбостратного графіту створюються передумови для зародження та росту вуглецевих наностінок або волокон.

На основі поєднання магнетронного ефекту з ефектом порожнистого катоду розроблений іонний розпилувач складених з різних матеріалів мішені, що дозволило отримувати пористі системи з будь-яким елементним складом. За результатами досліджень створеного пристрою подана до розгляду заявка на патент України на винахід.

Вперше при використанні парів ацетону як робоче газове середовище, а також різних геометричних характеристик порожнистого катода отримані пористі

конденсати графіту з різними морфологічними характеристиками. При цьому показано, що просторово розподілена селективність зародження та зростання стовпчастих структур графіту визначається флуктуаціями напруженості електричного поля над ростою поверхнею. За результати проведених досліджень отриманий патент України на корисну модель.

Дослідження циклів зарядки–розрядки при використанні в якості анодів пористих наносистем на основі C/Zn, C/Ni, C/ZnO, C/NiO, ZnO/NiO та Zn/ZnO/NiO і таких розчинників, як LiPF<sub>6</sub> і LiBF<sub>4</sub> показали, що залежно від структурно–морфологічних характеристик, елементного складу анодів та кількості циклів питома ємність акумуляторів змінюється в межах від 800 до 120 мА год/грам. При цьому вперше показано, що найбільшу стабільність до кількості циклів проявили аноди на основі наносистем C/ZnO при питомій ємності 250-320 мА год/грам.

## ЗМІСТ

|   |    |
|---|----|
| ВСТУП.....  | 6  |
| 1 ВИКОРИСТАННЯ ОКИСЛІВ МЕТАЛІВ ТА КОМПОЗИТНИХ МАТЕРІАЛІВ НА ЇХ ОСНОВІ В ЯКОСТІ ЕЛЕКТРОДІВ ЛІТІЙ ІОННИХ АКУМУЛЯТОРІВ ..  | 8  |
| 2 АВТОМАТИЗОВАНИЙ КОМПЛЕКС ОТРИМАННЯ ЗАРЯДНО-РОЗРЯДНИХ ХАРАКТЕРИСТИК .....  | 17 |
| 2.1 Принцип вимірювання електричних параметрів .....  | 17 |
| 2.2 Апаратна реалізація гальваностату.....  | 20 |
| 2.2.1 Джерело постійної регульованої напруги .....  | 22 |
| 2.2.2 Розробка мікропроцесорного вимірювального блоку.....  | 26 |
| 2.3 Автоматизація процесу вимірювання.....  | 30 |
| 3 ІНФОРМАЦІЙНА СИСТЕМА МОНІТОРИНГУ ХАРАКТЕРИСТИК ЛІТІЙ-ІОННИХ БАТАРЕЙ .....   | 32 |
| 3.1 Адміністрування інформаційної системи .....   | 50 |
| 4 ДОСЛІДЖЕННЯ ЕЛЕКТРОХІМІЧНИХ ВЛАСТИВОСТЕЙ ЕЛЕКТРОДІВ ...   | 55 |
| 4.1 Дослідження електрохімічних властивостей електродів.....  | 55 |
| 4.2 Структурно-морфологічні характеристики та елементний склад анодів на основі Zn/ZnO/NiO, Zn/ZnO, ZnO/NiO Zn/C і Ni/C та результати досліджень циклів зарядки-розрядки і ВАХ створених на їх основі електрохімічних комірок ..... | 58 |
| 4.2.1 Аноди на основі металевих оксидів.....  | 58 |
| 4.2.2. Аноди на основі турбостратного графіту.....  | 63 |
| ВИСНОВКИ.....   | 69 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....   | 70 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....   | 71 |
| Додаток А – Програмне забезпечення мікроконтролера .....  | 78 |

## ВСТУП

У зв'язку зі зростаючим попитом на чисту енергію та передові системи зберігання енергії, електродні матеріали з високою питомою ємністю та тривалим терміном служби у вигляді високоефективних літій-іонних акумуляторів (LIB), є актуальною темою досліджень та широко вивчаються [1]. LIB вважаються найбільш перспективною технологією зберігання енергії для портативної електроніки, електромобілів і гібридних електромобілів, а також в сучасних портативних засобах пересування, таких як моноколесо, електросамокат та інших схожих електричних транспортних засобів через високу щільність енергії, тривалий термін служби, відсутність ефекту пам'яті та безпечність для навколишнього середовища.

На сьогодні в комерційно доступних ЛІА найбільш часто використовують в якості анодів графіт. Відомо, що вуглецеві матеріали мають невисокі значення теоретичної та практично отриманої питомої ємності, порядку 372 мА·год./грам. Це значення набагато менше ємності анодів на основі окислів перехідних металів, наприклад, для оксиду цинку значення теоретичної питомої ємності досягає 978 мА·год/грам. З іншого боку, оксиди металів мають високий опір, що призводить до незадовільних характеристик роботи акумулятора при довготривалих та чисельних циклах заряджання-розряджання. Основна гіпотеза проекту полягала у виготовленні композитних матеріалів на основі вуглецю та оксидів металів, що дозволить поєднати добрі провідникові властивості вуглецевих матеріалів з підвищеними ємнісними характеристиками матеріалів на основі оксидів металів.

Отже, проект базується на основній гіпотезі про те, що використовуючи дослідження літій-іонних акумуляторів на базі комерційно доступних катодів і отриманих анодів з різними структурно-морфологічними характеристиками і елементним складом, можна визначити оптимальну конфігурацію електродів, яка відповідає за підвищені характеристики акумуляторів.

Основна прикладна проблема, на вирішення якої було спрямовано наукову роботу полягала в розробці технології, що дозволить сформувати аноди на основі пористих наносистем ZnO, C та композитів у вигляді C/Zn, C/Ni, C/ZnO, C/NiO, ZnO/NiO та Zn/ZnO/NiO з підвищеними пористістю, електро-хімічними властивостями та зменшеними розмірами структурних елементів до нанорівня, використовуючи які можна підвищити ефективність роботи літій-іонних систем накопичування та зберігання енергії.

## **1 ВИКОРИСТАННЯ ОКИСЛІВ МЕТАЛІВ ТА КОМПОЗИТНИХ МАТЕРІАЛІВ НА ЇХ ОСНОВІ В ЯКОСТІ ЕЛЕКТРОДІВ ЛІТІЙ ІОННИХ АКУМУЛЯТОРІВ**

Катод, анод, електроліт, сепаратор і струмоприймач є основними компонентами літій-іонної батареї. Літєві елементи працюють на окислювально-відновній хімічній реакції іонів літію, в процесі якої потенціал елемента створюється різницею потенціалів між іонами Li в двох матеріалах, які можуть оборотно включати Li. Іони літію переміщуються між електродами в процесі заряду і розряду. Коли акумулятор заряджений, іони літію витягуються з матеріалу катода і вбудовуються в матеріал анода після проходження через електроліт і сепаратор. У той же час електрони переходять від катода до анода через зовнішній контур для підтримки загальної нейтральності. Коли акумулятор розряджається, іони літію та електрони течуть у протилежних напрямках. Вилучення/введення іонів літію між матеріалами електродів відповідає перенесенню електронів через периферійні ланцюги, що реалізує взаємне перетворення між хімічною енергією та електричною енергією. Перенесення іонів літію всередині акумулятора супроводжується потоком заряду в зовнішньому контурі, тому ефективність перенесення іонів літію в електроліті впливає на ємність акумулятора. Ефективність батареї зазвичай вимірюється фізичними величинами, такими як питома ємність, продуктивність швидкості, кулонівська ефективність і термін служби.

Початковим матеріалом негативного електрода був металевий літій, який є найлегшим елементом періодичної таблиці. Літєві електроди та полярні апротонні електролітні розчинники створювали щільну поверхневу плівку, що унеможливило досягнення достатньої пасивації [2]. Коли батарея заряджалася та розряджалася, в ній утворювалися літєві дендрити, що спричиняло проблеми з безпекою та призводило до величезного споживання літію.

Як правило, електроди визначають більшу частину електрохімічних характеристик LIB, і, таким чином, вибір матеріалів анода та катода стає



ключовим фактором під час оптимізації акумуляторних систем. В комерційних ЛІВ, що перезаряджаються, використовуються графітові аноди і катод з оксиду металу такого як літій-кобальт оксид ( $\text{LiCoO}_2$ ). І хоча графіт є найбільш широко використовуваним анодним матеріалом у ЛІВ, він має деякі недоліки. Як зазначалося, металевий літій і вуглець мають подібні електричні потенціали, що викликає осадження літійових дендритів, що призводить до короткого замикання, а також низька теоретична ємність  $372 \text{ mA} \cdot \text{год} \text{ g}^{-1}$  графіту обмежує застосування у великомасштабних накопичувачах енергії.

Багато зусиль було присвячено розробці анодних матеріалів для літій-іонних акумуляторів високої ємності з хорошою стабільністю, щоб замінити графіт. Для підтримки термодинамічної стабільності батареї енергетичний зазор електроліту повинен бути більшим за потенціальну енергію між катодом і анодом [3]. Іншими словами, літійові батареї мають певні вимоги до потенціалу їх електродних матеріалів. Потенційна енергія анода повинна бути меншою за LUMO електроліту, а потенціальна енергія катода повинна бути більшою за НОМО електроліту. Такі вимоги можуть зіграти захисну роль на електроліті і продовжити термін служби акумулятора.

Щоб отримати високоефективні анодні матеріали, ключовим підходом є створення більшої кількості хімічних зв'язків з літієм за допомогою хімічних реакцій між активними матеріалами та  $\text{Li}^+$ , в яких при електрохімічному циклі бере участь багатоелектронний перенос і можна очікувати високу питому ємність [4]. До цього часу багато досліджень досліджували нові анодні матеріали для літій-іонних батарей, включаючи кремній, оксиди олова, оксиди перехідних металів тощо. Згідно з попередніми дослідженнями [5], було доведено, що оксиди перехідних металів оборотно адаптують  $\text{Li}^+$ , за допомогою реакцій конверсії та навіть легування, що призводить до великої питомої ємності [6].

Оксидні електроди перехідних металів мають високу теоретичну питому ємність ( $500\text{--}800 \text{ mA} \cdot \text{год} \text{ g}^{-1}$ ), регульовану робочу напругу та чудову продуктивність циклу [7]. Звичайні перехідні метали для даних оксидів частіше за все включають  $\text{Co}$ ,  $\text{Fe}$  та  $\text{Mn}$ .

Серед цих оксидів перехідних металів особливий інтерес представляє MnO через цю природну кількість, низьку вартість та високу теоретичну ємність ( $756 \text{ mA} \cdot \text{год} \text{ g}^{-1}$ ) [8]. На жаль, практичній роботі чистого матеріалу MnO перешкоджає його низька електропровідність і вібрація структури під час циклічного розряду заряду, що призводить до поганої оборотності та сильного згасання ємності. У цьому відношенні були прийняті різні підходи, включаючи вуглецеве покриття та наноструктуралізацію, щоб покращити продуктивність зберігання літію. Наприклад, Liu та ін. повідомили про виготовлення кубічного нанокompозиту MnO/C як анодного матеріалу для LIB, який забезпечував питому ємність  $470 \text{ mA} \cdot \text{год} \text{ g}^{-1}$  при  $75 \text{ mA} \text{ g}^{-1}$ , використовуючи переваги меншого розміру частинок та ефективного вуглецевого покриття [9]. Більше того, Sheng та ін. запропонували леговане азотом вуглецеве покриття для наночастинок MnO, що забезпечує високу оборотну ємність  $904 \text{ mA} \cdot \text{год} \text{ g}^{-1}$  при  $0,5 \text{ A} \text{ g}^{-1}$  після 500 циклів в якості анодів LIB [10]. Незважаючи на досягнутий великий прогрес, нові, швидкі та прості методи синтезу наноструктурованих композитів також можуть сприяти подальшому розвитку цих матеріалів у практичному застосуванні.

Електроди на основі феруму мають доволі високу теоретичну оборотну ємність ( $926 \text{ mA} \cdot \text{год} / \text{г}$ ) при розряді до 0 В порівняно з металевим Li, оскільки він має можливість приєднувати 8 іонів літію. Однак, через агломерацію,  $\text{Fe}_3\text{O}_4$  матеріали мають погані циклічні властивості та величезні зміни об'єму активних матеріалів під час літіювання/делітіювання.

Оксиди на основі кобальту включають CoO та  $\text{Co}_3\text{O}_4$ . Обидва мають високу питому ємність. Розмір і форму оксиду на основі Co можна регулювати шляхом зміни методу та умов синтезу. Наприклад, Cao та ін. [11] безпосередньо вирощували нанокластери CoO на мідній фользі. Матеріал досяг оборотної ємності  $1516,2 \text{ mA} \cdot \text{год} \text{ g}^{-1}$ . Wang та ін. [12] підготували порожнисті наночастинки  $\text{Co}_3\text{O}_4/\text{C}$  за допомогою процесу твердотілого синтезу без шаблонів, і він забезпечив стабільну оборотну ємність близько  $880 \text{ mA} \cdot \text{год} \text{ g}^{-1}$  при щільності струму  $50 \text{ mA} \cdot \text{год} \text{ g}^{-1}$ , що може заощадити вартість приготування порожнистих частинок.

У порівнянні з промисловими графітовими анодами, дані оксидні анодні матеріали можуть уникати літієвих дендритів і мають кращу безпеку та питому ємність. Також, у порівнянні з легованими анодними матеріалами, вартість виробництва нижча. Однак їх поточне комерційне застосування є незрілим, і є ще багато проблем, які потрібно вирішити, включаючи притаманну погану провідність, розсіпання електродів, викликане великим розширенням об'єму, затримку напруги між зарядом і розрядом і безперервне розкладання електроліту.

Проблема гістерезису напруги впливає з того факту, що перебудова матеріалу анода під час процесу літіювання/делітіювання частково розсіює внутрішнє тепло, що призводить до зниження ефективності щільності енергії циклу заряду/розряду. Однією зі стратегій вирішення проблеми поганої провідності ТМО, яка може ефективно покращити провідність всього матеріалу є комбінація ТМО і провідних вуглецевих матеріалів [13].

У порівнянні з більшістю оксидів перехідних металів наведених вище, таких як  $\text{Co}_3\text{O}_4$  і  $\text{Fe}_3\text{O}_4$  [14],  $\text{ZnO}$  та  $\text{NiO}$  мають переваги більш високої теоретичної ємності, нижчу вартість та більшу екологічність, чудові електрохімічні властивості [15], але їм приділено відносно мало уваги. Так, теоретична ємність з'єднань сплавів  $\text{Li}_2\text{O}$  і  $\text{LiZn}$  складає  $978 \text{ мАгод г}^{-1}$ , що майже у три рази вище, ніж у графіту, а теоретична ємність  $\text{NiO}$  може досягати  $718 \text{ мАгод г}^{-1}$  [16].

Оксид нікелю ( $\text{NiO}$ ) є власним напівпровідником р-типу з широкою забороненою зоною та хорошою термічною та хімічною стабільністю. Теоретична питома ємність  $\text{NiO}$  очевидно перевищує графітову, має велике практичне значення розробка нового покоління літій-іонних акумуляторів великої ємності. Однак електрохімічні характеристики матеріалів  $\text{NiO}$  значно відрізняються через відмінності між розміром частинок і морфологією.  $\text{NiO}$  мікронного розміру має низьку необоротну ємність і майже не може бути перероблений. Відповідно до механізму реакції перетворення, розряд  $\text{NiO}$  зобов'язаний генерувати велику кількість оксиду літію низької щільності, що

змушує електрод швидко розширюватися. При накопиченні напруги відбувається згасання ємності, оскільки активна речовина остаточно випадає з колектора.

Регулювання та контроль структури (форми, розміру чи компонента) матеріалів NiO, для покращення їх електрохімічних характеристик, є важливим викликом для вчених, оскільки властивості матеріалів зазвичай залежать від структури. Наприклад, великі зусилля було зосереджено на контролі об'ємних змін за допомогою використання малих частинок NiO з різною морфологією, включаючи наностінку [17], наноліст [18] та пористу тверду речовину [19]. Було доведено, що ці матеріали демонструють кращі електрохімічні характеристики, що свідчить про те, що модифікація структури може бути ефективним рішенням для підвищення електрохімічних характеристик. Зокрема, порожнисті нанотрубки NiO представляють великий інтерес для оборотного зберігання іонів літію через їх коротку дифузію Li<sup>+</sup> та хорошу переносимість зміни об'єму під час циклювання. Проте повідомлення про великомасштабний синтез порожнистих нанотрубчастих анодів NiO є досить рідкісними.

Також, для вирішення цієї проблеми було використано метод електроспінінгу для отримання нанорозмірних матеріалів анода літій-іонних акумуляторів [20], завдяки яким NiO покращує збереження ємності. У роботі [21] наностержень NiO, синтезований при 600 °C, показав найвищу початкову ємність (1256 мАгод г<sup>-1</sup>), але наностержень, виготовлений при 500 °C, мав найкращу циклічну продуктивність. Хоча nano-NiO був менш ефективним у першому циклі заряду-розряду, ефективність зарядки та розряду поступово збільшувалася в міру збільшення циклу. Після 50 циклів пористі 3D NiO наностержня все ще мали 99% збереження ємності при 400 мАгод г<sup>-1</sup>.

Останнім часом шаблонний метод став простим і швидким виготовленням порожнистих наноструктур NiO. Механізм формування полягає в тому, що шаблон належного розміру та морфології виступає «гостем», а наночастинки ростуть на ньому як «господар», і таким чином виходить матеріал з регульованою морфологією та розміром. Після чого, шаблони можна легко видалити шляхом високотемпературного прожарювання або координаційного травлення [22].

Безсумнівно, метод із використанням біошаблону є ефективним підходом до отримання бажаних порожнистих структур [23]. Проте, до теперішнього часу анодні матеріали NiO, синтезовані звичайним шаблонним методом, все ще мають більший розмір частинок, нестабільність петлі та інші недоліки.

ZnO є одним із найпривабливіших функціональних напівпровідникових матеріалів, який можна використовувати як ефективну механічну опору та шлях електропровідності. Крім того, ZnO має високу теоретичну ємність  $978 \text{ мАгод г}^{-1}$  як анодний матеріал [24]. Однак різка зміна обсягу під час розряду-заряду легко призводить до сильного подрібнення [25]. В результаті електрод демонструє серйозне зниження ємності під час циклування навіть при низькій густині струму [26]. На жаль, різке розширення та стиснення структури під час реакцій перетворення та легування, а також незадовільна електропровідність роблять ZnO та NiO невідповідними для використання як високоефективні анодні матеріали для LIB [27].

Для вирішення цих проблем було запропоновано деякі стратегії підвищення структурної стабільності ZnO. Наприклад, пористі або наноструктуровані матеріали можуть пристосуватися до зміни обсягу, а потім покращити циклічні характеристики ZnO [28]. Нанесення буферних шарів на поверхню ZnO є ще одним привабливим способом. Більше того, композитні матеріали також є одним з ефективних способів покращення електрохімічних характеристик, оскільки композитна структура може буферизувати деформацію та покращувати електричну провідність.

За останні роки у застосуванні нових наноматеріалів у LIB відбулося багато значних змін. Для покращення циклічності анодних матеріалів на основі оксидів перехідних металів розроблено різноманітні оксиди та їх гібридні композити з багатовимірними мережами, що характеризуються високою площею поверхні, високою провідністю та високопористими наноструктурами [29]. Наприклад, групою Huang та ін. були підготовлені пористі нанопласти ZnO, вирощені на мідних підкладках, які показали хороші електрохімічні характеристики ( $\sim 400 \text{ мАгод г}^{-1}$  при  $500 \text{ мА г}^{-1}$ ) [28]. Choi та ін. синтезував ZnO/мезопористі

вуглецеві наноккомпозити методом преципітації, що забезпечило високу оборотну ємність  $610 \text{ мАгод г}^{-1}$  при  $100 \text{ мА г}^{-1}$ , але незадовільну продуктивність: лише  $180 \text{ мАгод г}^{-1}$  при  $1000 \text{ мА г}^{-1}$  [30]. Хіе та ін. розробив метод для отримання мікросфер ZnO-C з розчину з унікальними структурами жовткової оболонки, які продемонстрували чудову циклічну стабільність [31]. Та, подібним чином Ту та ін. спочатку підготували композит з аморфної квантової точки ZnO (менше 3 нм)/мезопористого вуглецевої гранули, який забезпечував високу оборотну ємність майже  $930 \text{ мАгод г}^{-1}$  при  $100 \text{ мА г}^{-1}$  [32].

Хоча оборотна здатність анодів на основі ZnO була певною мірою покращена після значних зусиль, швидкісні характеристики все ще незадовільні.

Розробка гібридних гетероструктур може повною мірою використовувати переваги кожного компонента наноматеріалів, і в той же час синергетичний ефект і взаємодія між різними компонентами також можуть до певної міри покращити стабільність циклу батареї та провідність. Завдяки поєднанню унікальних властивостей окремих компонентів такий електрод продемонстрував покращену продуктивність. У системах гібридних матеріалів неактивна або менш активна композиція зазвичай використовується як підтримуюча або провідна матриця для буферування зміни об'єму або полегшення реакцій. Крім того, можуть бути деякі взаємовигідні реакції між композиціями в помірних умовах. Багато з цих гібридних матеріалів, наприклад,  $\text{SnO}_2\text{-RuO}_2$ ,  $\text{SnO}_2\text{-CoO}$ ,  $\text{MnO-Ni}$  [33],  $\text{ZnO-CoO}$ ,  $\text{TiO}_2\text{-MoO}_3$ ,  $\text{ZnO-MnO}_2$  [34],  $\alpha\text{-Fe}_2\text{O}_3@\text{SnO}_2$  та  $\text{SnO}_2\text{-ZnO}$  повідомлялося як високоефективні анодні матеріали для LIB. Проте стабільність інтерфейсу при встановленні гібридного електрода все ж викликає занепокоєння [35].

Останнім часом двокомпонентні активні оксиди перехідних металів розглядаються як перспективні кандидати для високоефективних LIB [27] в порівнянні з однокомпонентними оксидами, таких як  $\text{NiCo}_2\text{O}_4$ ,  $\text{ZnCo}_2\text{O}_4$  [36],  $\text{MnCo}_2\text{O}_4$  та  $\text{CoMn}_2\text{O}_4$ . Наприклад, Wei та ін. [37] синтезували наночастинки  $\text{NiCo}_2\text{O}_4$  за допомогою золь-гель-методу з епоксидом і досягли значно збільшеної ємності. NuLi та ін. [38] підготували наноккомпозит  $\text{NiCo}_2\text{O}_4/\text{C}$  гідротермальним методом, отримали високу оборотну ємність  $958,4 \text{ мА} \cdot \text{год г}^{-1}$  при  $40 \text{ мА} \cdot \text{г}^{-1}$ .

Shen та ін. [39] повідомили про нову архітектуру, що складається з вуглецевого текстилю, рівномірно покритого мезопористими нанодротами  $\text{NiCo}_2\text{O}_4$  (NWA), виготовленими за допомогою простого гідротермального методу за допомогою поверхнево-активних речовин у поєднанні з короткою обробкою після відпалу, яка може забезпечити високу ємність  $854 \text{ mA} \cdot \text{год} \text{ g}^{-1}$  за 100 циклів при  $0,5 \text{ A} \text{ g}^{-1}$ . Багатопористі квазіпорожністі мікросфери  $\text{MnCo}_2\text{O}_4$  та  $\text{CoMn}_2\text{O}_4$  були отримані топотактичним хімічним перетворенням із відповідних попередників [40], оборотні ємності яких становили 755 і 706  $\text{mA} \cdot \text{год} \text{ g}^{-1}$  при щільності струму  $200 \text{ mA} \text{ g}^{-1}$  після 25 циклів для квазіпорожністих мікросфер  $\text{MnCo}_2\text{O}_4$  і  $\text{CoMn}_2\text{O}_4$  відповідно.

Повідомляється, що  $\text{ZnO}/\text{SnO}_2$  [41],  $\text{SnO}_2/\text{NiO}$  [42],  $\text{SnO}_2/\text{TiO}_2$  [43] та  $\text{Co}_3\text{O}_4/\text{Fe}_2\text{O}_3$  [44] демонструють покращені характеристики зберігання літію порівняно з однокомпонентними оксидами через синергетичні ефекти. Для двокомпонентно-активних оксидів перехідних металів вставка/вилучення  $\text{Li}^+$  з двох оксидів відбувається поетапно під час циклу, що може забезпечити значну окислювально-відновну хімічну кінетику та підвищену електропровідність, що призводить до підвищених електрохімічних характеристик.

Одними з найбільш широко досліджуваними сучасними структурами є порожністі наночастинки зі структурою «жовткової» оболонки, сконструйованою за допомогою нанорозмірних субодиниць. Група Ванга повідомила про порожністу сферу  $\text{CuO}/\text{NiO}$  з «жовткової» оболонки на основі  $\text{Cu-Ni}$  біметалічних органічних каркасів, і була отримана висока оборотна питома ємність близько  $1000 \text{ mA} \cdot \text{год} \text{ g}^{-1}$  з високою циклічною стабільністю. Крім того,  $\text{ZnO}/\text{ZnCo}_2\text{O}_4/\text{C}$  зі структурою ядро/оболонка, отриманою з  $\text{Zn-Co}$  метало-органічних каркасів, демонстрував оборотну ємність  $669 \text{ mA} \cdot \text{год} \text{ g}^{-1}$  після 250 циклів при  $0,5 \text{ A} \text{ g}^{-1}$ , і структура ядро/оболонка добре підтримувалася навіть після 250 циклів. У роботі [45] мікросфери  $\text{ZnO}/\text{NiO}$  зі структурою жовткової оболонки були синтезовані за допомогою простого сольвотермічного методу з подальшим процесом прожарювання на повітрі. Отриманий гібрид характеризувався унікальною оболонкою і жовтком мікросфер, що складався з

наностержнів, високою електрохімічною активністю та раціональною гібридизацією ZnO та NiO. В результаті можна отримати високу питому ємність 1008,6 мАгод г<sup>-1</sup> при 0,1 Аг<sup>-1</sup> після 200 циклів і 592,4 мАгод г<sup>-1</sup> при 0,5 Аг<sup>-1</sup> після 1000 циклів, а також відмінну швидкість. Виходячи з цих обнадійливих досліджень, згаданих вище, використання метало-органічних каркасів як попередників для синтезу двокомпонентних активних оксидів металів зі структурою жовткової оболонки, що відрізняються високою ємністю та правильною морфологією, є можливим і перспективним.

Сумуючи вищенаведене, можна сказати, що дослідники вкладали багато енергії в дослідження відповідних анодних матеріалів, але, частіше за все, оптимізація одного аспекту продуктивності часто призводить до зниження інших властивостей [13].

Загальною метою дослідження є підготовка анодних матеріалів із гарною комплексною продуктивністю. Аноди на основі оксидів є перспективними кандидатами на заміну графітового анода, і в порівнянні з анодами легованого типу вони демонструють менші зміни об'єму, більш тривалі циклічні характеристики, кращу швидкість і більш високу надійність.



## **2 АВТОМАТИЗОВАНИЙ КОМПЛЕКС ОТРИМАННЯ ЗАРЯДНО-РОЗРЯДНИХ ХАРАКТЕРИСТИК**

### **2.1 Принцип вимірювання електричних параметрів**

Вимірювання електрофізичних параметрів різноманітних електронних пристроїв (плівкових та напівпровідникових структур, датчиків, комірок гальванічних елементів тощо) дозволяє виявити та дослідити такі явища, як корозію матеріалів, дію інгібіторів, процеси осадження та розчинення матеріалів, електродні процеси в хімічних джерелах струму, вольтамперометричну та кулонометричну складову речовини, корозійну стійкість матеріалів, вплив на корозію різних добавок та ін.

В роботі особлива увага приділена розробці вторинних хімічних джерел струму багаторазової дії, який може бути знову заряджений після розряду (акумуляторів).

Акумулятори використовуються для циклічного накопичення енергії (заряд-розряд) та автономного електроживлення різних електротехнічних пристроїв та обладнання, а також для забезпечення резервних джерел енергії у медицині, виробництві, транспорті та інших сферах. Для заряду акумулятора постійний електричний струм пропускається в напрямку зворотному напрямку струму при розряді.

Водночас проектування нових акумуляторних елементів потребує дослідження їх зарядно – розрядних характеристик, що дозволяє встановити закономірності протікання як прямих, так і зворотних електрофізичних процесів.

З цією метою використовуються потенціостати-гальваностати.

Потенціостат - це електронний прилад, призначений для автоматичного контролю потенціалу електрода і підтримка заздалегідь заданої величини цього електрода.

Гальваностат (амперстат) - це прилад, який підтримує постійний струм у певній комірці, причому незалежно від різниці потенціалу електродів. Обидва

пристрої знаходять широке застосування для вимірювання електрохімічного аналізу та інших показників.

Якість зазначених приладів повністю залежить від точних характеристик та можливостей апаратури, у яких їх застосовують.

Щоб реалізувати режим гальваностатичного виміру достатньо перетворити потенціостат на гальваностат. Знімаємо сигнал зворотного зв'язку (негативний зв'язок) з перетворювача струм-напруга, а сигнал-відгук (тобто напруга) подаємо на пристрій, який обробляє сигнал.

Лабораторні потенціостати-гальваностати застосовують у таких галузях дослідження:

- процеси корозії та електродної електрокаталізації;
- електрохімічне розчинення та осадження;
- випробування невеликих хімічних джерел струму та їх батареї;
- сенсорні системи (твердотільні та рідкі);
- електроаналітика;
- електрохімічний синтез.

Водночас вартість існуючих потенціостати-гальваностати (наприклад, VersaSTAT 3/3F/4, Autolab PGSTAT302N або аналоги) складає неменше 11000 €. Виробники не забезпечують споживачів їх продукції повноцінними принциповими та функціональними схемами, що є предметом комерційної таємниці та надають лише технічну та консультативну підтримку користувачів.

Таким чином розроблення лабораторних приладів для проведення гальваностатичних вимірювань є актуальною задачею.

Дослідження електричних параметрів джерела струму можна дослідити використавши одну зі схем, наведених на рисунку 2.1.

Перший метод (рисунок 2.1, а) використовує для проведення дослідження джерело постійної напруги, потенціали клем якого не змінюються незалежно від прикладеного навантаження. При таких вимірюваннях необхідно зважати на те, що при значному розряджанні та низькому внутрішньому опорі елемента в колі виникають значні струми, які можуть призвести до його пошкодження в

результаті протікання критичних струмів. Останній чинник бажано враховувати при проведенні експериментальних досліджень та вжити заходів щодо обмеження значення струму в колі.

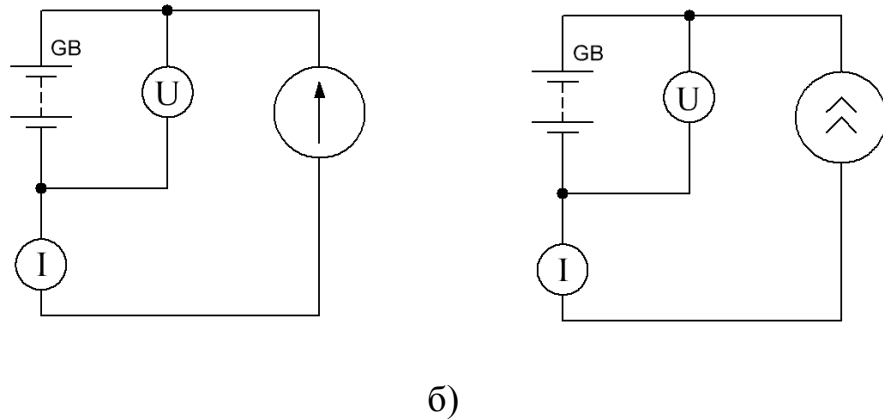


Рисунок 2.1 – Схеми дослідження електричних параметрів (а – з використанням потенціостату, б – з використанням гальваностату)

Другий метод (рисунок 2.1, б), на відміну від першого, містить джерело постійного струму. В цьому випадку струм у колі завжди буде мати постійне значення незалежно від навантаження. При проведенні вимірювань необхідно врахувати той факт, що при високому внутрішньому опорі елемента напруга на елементі може прийняти значення, яке визначається законом Ома:

$$U = I \cdot R$$

і може збільшитися до такого значення, що відбудеться електричний пробій внутрішніх структур елемента та його подальше руйнування. З метою запобігання цьому необхідно обмежити напругу на досліджуваному елементі.

Струм в колі та напруга на елементі вимірюються амперметром (A) та вольтметром (U) відповідно. Полярність напруги та напрямок протікання постійного струму в колі може бути змінено в залежності від умов проведення експерименту.

Як зазначено вище, кожен метод має свої переваги та недоліки та доповнюючи один одного може надати більш об'єктивні результати.

Найбільш широкого використання для тримання зарядно-розрядних характеристик вторинних хімічних джерел струму багаторазової дії набув другий метод, оскільки використання джерела струму дозволяє не використовуючи складні розрахунки обчислити ємність акумулятора шляхом виміру часу заряду (розряду).

## 2.2 Апаратна реалізація гальваностату

Враховуючи вказане розділі 2.1 актуальною є розробка автоматизованого комплексу для вимірювання електрофізичних параметрів, який дозволить сумістити в одному пристрої наступні елементи:

- джерело постійного регульованого струму 0,01 мА до 50 мА (з можливістю зміни напрямку протікання струму в колі) з дискретністю зміни значення струму 0,01 мА;
- електронний вольтметр з діапазоном вимірювання 0 – 3.2 В (роздільна здатність  $\pm(1\%+3$  молодших розряди), теоретична роздільна здатність не гірша  $\pm 1$  мВ в діапазоні 0-3.0 В) ;
- цифрову систему керування джерелом постійного струму та вимірювання значення напруги;
- можливість локального та віддаленого керування вимірювальним комплексом.

На основі існуючої елементної бази та схемотехнічних реалізацій гальваностатів визначено наступну елементну базу:

- платформою для реалізації вимірювального комплексу є мікроконтролер STM8S003F3P6 виробника STelectronics, який має наступні апаратні характеристики: покращене STM8 ядро з гарвардською архітектурою з частотою ядра до 16МГц з 3 ступеневим конвеєром, таймери лічильники (8 біт та 16 біт) з функцією широтно-імпульсної модуляції (ШИМ), 5 каналний 10 бітний аналого-цифровий перетворювач (АЦП), апаратний універсальний асинхронний

прийомо-передавач (УАПП), послідовний периферійний інтерфейс (SPI) та ін., напруга живлення від 2,95 до 5,5 В.

– аналого цифровий перетворювач AD7705 виробника Analog Devices, який має наступні характеристики: 2 повністю диференціальних входи, 16 бітна роздільна здатність АЦП, 0,003% нелінійність, програмований коефіцієнт посилення від 1 до 128, 3-провідний послідовний інтерфейс SPI-, QSPI™-, MICROWIRE™- і DSP-сумісний, можливість буферизації аналогового входу, напруга живлення від 2,7 В до 3,3 В або від 4,75 В до 5,25 В;

– операційний підсилювач MCP6002: полоса пропускання 1 МГц, Rail-to-Rail Вхід/Вихід, напруга живлення: від 1,8 В до 6,0 В, типовий струм живлення 100 мкА.

Задача мікропроцесорного блоку полягає в курування джерелом струму, зміна напрямку протікання струму у досліджуваному об'єкті, вимірювання напруги на затискачах елемента, взаємодія з електронною обчислювальною машиною (ЕОМ) по послідовній шині.

Структурна схема гальваностату зображена нижче.

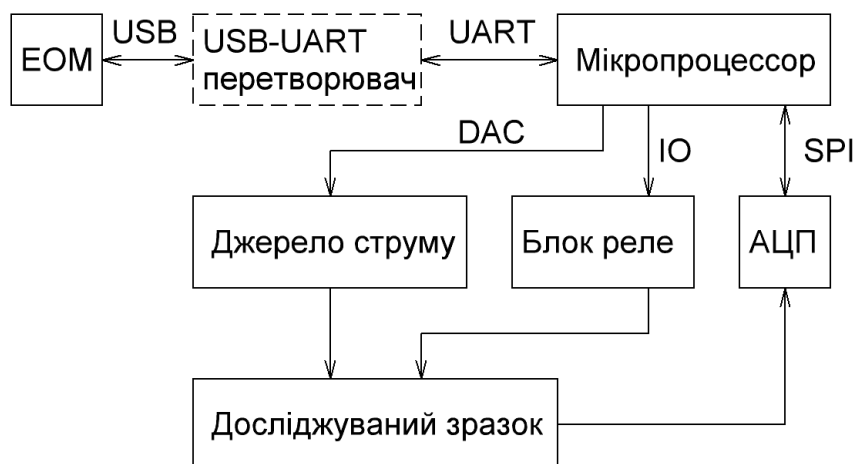


Рисунок 2.2 – Структурна схема гальваностату (USB – універсальна послідовна шина USB, DAC – цифро-аналоговий перетворювач, ІО – порти вводу-виводу, UART - універсальний асинхронний прийомо-передавач)

На рисунку 2.2 пунктирною лінією зображено USB-UART перетворювач, який може бути використаний на ЕОМ, які не мають універсальний асинхронний прийомо-передавача та оснащені лише USB портами. Як видно з структурної схеми всі функції керування здійснюються виключно мікроконтроллером, який окрім того здійснює прийом команд керування від програмного забезпечення ЕОМ та передає результати у зворотному напрямку.

### 2.2.1 Джерело постійної регульованої напруги

Отримання стабільного струму, який подається на досліджуваний зразок, являється важливим чинником, що впливає в кінцевому рахунку на результат вимірювання ємності акумулятора.

Реалізувати джерело струму можливо двома шляхами:

1. Використати джерело напруги послідовно з опором, який на декілька порядків більше внутрішнього опору акумулятора. Оскільки опір досліджуваного зразка заздалегідь невідомий, то використання такого підходу є недоцільним з точки зору підбору опору для кожного експерименту та буде приводити до збільшення похибки вимірювань протягом часу їх проведення. Таким чином такий підхід є недоцільним, хоча є найпростішим.

2. Використати джерело струму, побудоване на активних елементах. Такий підхід дозволяє за рахунок зворотного зв'язку забезпечити стабільність вихідного струму незалежно від навантаження, реалізувати можливість встановлення необхідного значення струму керуючими сигналами, а не зміною опору в колі.

Джерело струму, яке керується напругою призначене для забезпечення навантаження струмом  $I_2$ , який не залежить від вихідної напруги  $U_2$  та регулюється тільки напругою  $U_1$ . Рівняння

$$I_1 = 0 * U_1 + 0 * U_2$$

$$I_2 = S * U_1 + 0 * U_2$$

На практиці задовольняється приблизно. Представимо реальне джерело струму еквівалентною схемою (рисунок 2.3) для якої справедливі наступні рівняння

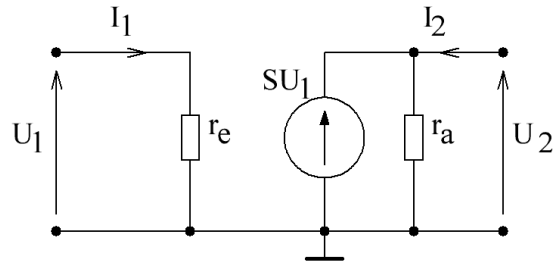


Рисунок 2.3 – Еквівалентна схема джерела струму, яке керується напругою

$$\begin{cases} I_1 = \frac{1}{r_e} * U_1 + 0 * U_2 \\ I_2 = S * U_1 + 1/r_a * U_2 \end{cases} \quad (2.1)$$

При  $r_e \rightarrow \infty$  та  $r_a \rightarrow \infty$  отримаємо ідеальне джерело струму. Параметр  $S$  називають крутизною або провідністю схеми.

Як слідує з виразу (2.1) струм  $I_2$  прямопропорційно залежить від напруги  $U_1$  та провідності схеми, тим самим дозволяє реалізувати керування вихідним струмом шляхом зміни напруги на вході.

Оскільки мікропроцесор містить у своєму складі 16 розрядний таймер з апаратною функцією широтно-імпульсного модулятора, що дозволяє реалізувати досить просте та зручне керування вихідним струмом.

Досліджуваний зразок, як показано на рисунку 2.1, приєднується до вихідних клем джерела уніполярного струму. Зміна напрямку протікання струму реалізуються шляхом зміни контактних клем блоком реле (рисунок 2.2).

Перевагами уніполярного джерела струму є можливість віддавати в коло значні струми при високих напругах, але водночас такі схеми мають недоліки, а саме однополярні струми, навантаження, як правило повинне бути ізольоване. Схемна реалізація джерел струму можлива з використанням як біполярних, так і польових транзисторів, але в першому випадку неможливо забезпечити ізоляцію

навантаження від кола керування, в той час як використання біполярних транзисторів з ізольованими затвором вирішує задачу розробки схеми з необхідними вимогами.

Принципова схема джерела струму, яке керується напругою наведено на рисунках 2.4, 2.5.

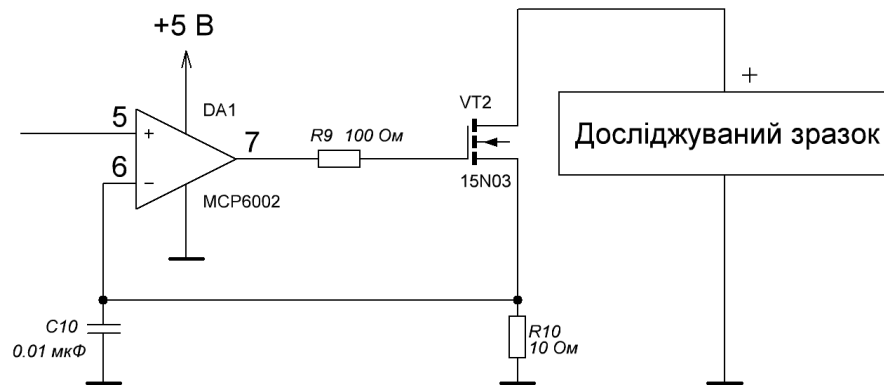


Рисунок 2.4 – Принципова схема джерела струму, яке керується напругою, використовується при отриманні розрядних характеристик

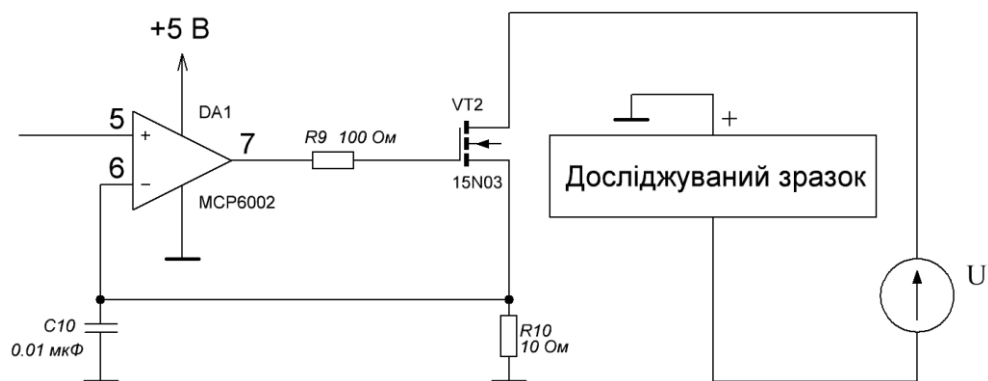


Рисунок 2.5 – Принципова схема джерела струму, яке керується напругою, використовується при отриманні розрядних характеристик

З рисунку 2.4 видно, що джерело струму використовується в якості навантаження для забезпечення стабільного керованого струму розряду, в той час



рисунок 1.5 показує використання того ж самого джерела струму для заряджання. В цьому випадку значення напруги джерела  $U$ , внутрішній опір досліджуваного зразка не мають критичного значення, оскільки заданий струм в колі підтримується за рахунок активних елементів DA1 та VT2 в автоматичному режимі в залежності від напруги на вході 5 операційного підсилювача та від'ємного зворотного зв'язку, утвореного елементами DA1, C10, R9, R10, VT2.

Використання в схемі операційного підсилювача DA1 вирішує одночасно декілька задач: узгоджує вихід мікроконтролера з входом транзистора, забезпечує керування польовим транзистором VT2 та необхідну провідність схеми, утворює коло зворотного зв'язку за рахунок подачі на інвертуючий вхід 6 з давача струму R10. Перелічене вище дозволяє реалізувати керування струмом пропорційно напрузі на вході 5 підсилювача.

Реалізувати схему постійної регульованої напруги здійснюється шляхом цифро-аналогового перетворювача (див. рисунок 2.2). Використовуючи наявні можливості мікропроцесора цифро-аналогове виконано з використанням програмно-апаратної ШИМ з роздільною здатністю 5000 одиниць, що дасть змогу формувати вихідну напругу в діапазоні 0-0.5 В, а вихідний струм джерела струму в діапазоні 0 – 50.00 мА, з дискретністю зміни 0.01 мА.

Формування керуючої напруги здійснюється за допомогою ШИМ частота якого складає 1кГц. Подібний метод дозволяє здійснити керування середнім значенням вихідної напруги шляхом зміни тривалості вихідного сигналу на цифровому виході мікропроцесора.

Отримання постійної керуючої напруги на вході операційного підсилювача DA1 забезпечується інтегруючими елементами R4, C2, C4, R3, C6, C7, які утворюються фільтр низьких частот (ФНЧ) та дільника напруги R5, R6.

Оскільки живлення ( $U_{ж}$ ) мікроконтролерної системи складає +3.3 В, а скважність  $D$  (коефіцієнт заповнення) ШИМ сигналу лежить в діапазоні 0-100%, то напруга на неінвертуючому вході 5 операційного підсилювача DA1 складає

$$U_5 = U_{ж} \cdot D$$

Таким чином напруга на вході дільника напруги R5, R6 складає 0-3.3 В. Щоб сформувати необхідний сигнал на вході 5 DA1 в межах 0-0,5 В використовується дільник напруги з коефіцієнтом 0.15, значення керуючої напруги визначається виразом

$$U_5 = U_{\text{ж}} \cdot D \cdot \frac{R_5}{R_5 + R_6} \quad (2.2)$$

Струм у колі стоку польового транзистора VT2 враховуючи вираз (2.2) складатиме

$$I_{VT2} = \frac{U_5}{R_{10}} = U_{\text{ж}} \cdot D \cdot \frac{R_5}{(R_5 + R_6) \cdot R_{10}} \quad (2.3)$$

Роздільна здатність ШИМ, як зазначалося раніше, складає 5000, тобто на 1 одиницю молодшого розряду припадає  $0.01 \cdot 10^{-3} \text{ А}$  в діапазоні 0-50 мА.

Для зміни напрямку струму в колі (зміна режиму розряджання на заряджання) використовується блок реле (рисунок 1.2), який комутує контакти реле К1 згідно схем (рисунок 1.4, рисунок 1.5) відповідно.

Таким чином, використовуючи описаний вище підхід реалізовано кероване джерело постійного струму 0 до 50 мА (з можливістю інверсії напрямку протікання струму) та кроком зміни 0,01 мА з цифровим задаванням автоматизованою системою збору та обробки даних.

### 2.2.2 Розробка мікропроцесорного вимірювального блоку

Принципова схема мікропроцесорного вимірювального блоку гальваностату зображено на рисунку 2.6 відповідає функціональній схемі (рисунок 2.2).

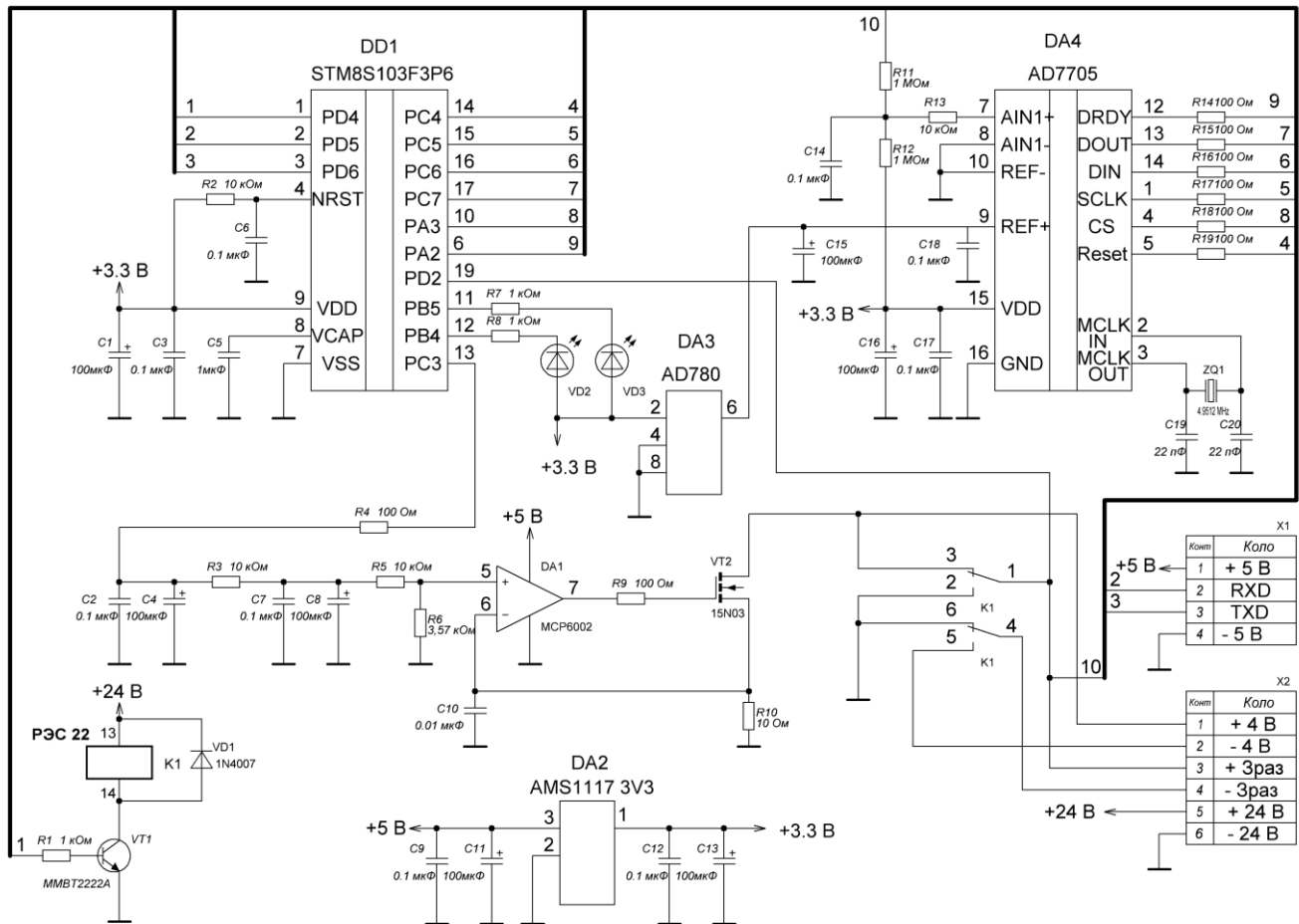


Рисунок 2.6 – Принципова схема гальваностату

Оснoву обчислювального блоку становить мікроконтроллер DD1, який обмінюється з аналого-цифровим перетворювачем DA4 по шині SPI: лінія PC4 використовується для початкового скидання; PC5 – лінія тактового сигналу SCLK (англ. Serial Clock), служить для передачі тактового сигналу до відомих (slave) пристроїв; PC6 – сигнал MOSI (англ. Master Out Slave In), служить для передачі даних від ведучого (master) до відомих пристроїв; PC7 – сигнал MISO (англ. Master In Slave Out), служить для передачі даних від відомих пристроїв до ведучого; PA3 – сигнал вибору кристалу (англ. Chip Select, Slave Select), служить для задіяння необхідного пристрою, активний рівень – низький; PA2 – сигнал, який подається від відомого до ведучого пристрою та сигналізує готовність вихідних даних, активний рівень – низький.

Порти PD5, PD6 налаштовані на вихід та вхід відповідно і використовуються для передачі даних по послідовній універсальній асинхронній лінії (UART) по стандарту RS-232 з TTL рівнями, які приєднуються до клем 2, 3 роз'єму X1 до якого підключається перетворювач USB-UART персональної обчислювальної машини.

Порт PD4 призначений для керування блоком реле K1 через транзистор VT1, яке змінює напрямок протікання струму через досліджуваний зразок шляхом комутації клем 1-3, 4-6 – в режимі розряду, 1-2, 4-5 - в режимі заряду. Індикація поточного режиму здійснюється світлодіодним індикатором під'єднаним до порту PB4, який представляє собою каскад з відкритим стоком. Аналогічний порт PB5 призначений для візуальної інтерпретації ходу налаштування апаратної частини мікроконтролера DD1.

Порт PC3 налаштований на вихід та являється лінією ШИМ сигналу 3 каналу внутрішнього таймера TIM1 мікроконтролера. Сигнал з цієї лінії подається на вхід фільтра низьких частот R4, C2, C4, R3, C6, C7 та дільник напруги R5, R6 та далі на неінвертуючий вхід 5 операційного підсилювача DA1. На інверсний вхід 4 операційного підсилювача DA1 подається сигнал з датчика струму R10. Оскільки різниця потенціалів між входами 4 та 5 операційного підсилювача близька до 0 В, то за рахунок зворотного зв'язку струм в колі стоку транзистора VT2 Таким чином, враховуючи значний коефіцієнт підсилення операційного підсилювача (близько 112 дБ) та малі вхідні струми ( $\pm 1.0$  пА) відбувається регулювання вихідного струму відповідно до рівняння (3).

Живлення мікроконтролера DD1 та АЦП DA4 здійснюється від стабілізатора напруги DA2, який перетворює постійну напругу +5 В в +3.3 В. Напруга + 5В використовується для живлення операційного підсилювача DA1, живлення реле K1 здійснюється від окремого джерела живлення + 24 В. Стабілізатор зразкової напруги DA3 використовується виключно для забезпечення опорної напруги +2.5 В АЦП DA4.

Для створення зарядного струму (див. рисунок 1.5, 1.6) використовується окреме незалежне джерело напруги + (4...5) В.

Контроль напруги при циклах розряду-заряду здійснюється відповідно мікроконтролером DD1 (аналоговий вхід PD2) та вхід AIN+ (вхід 7) аналого-цифрового перетворювача DA4.

Блок-схема алгоритму роботи основної програми мікроконтролеру DD1 наведено на рисунку 2.7. Під час роботи основної програми мікроконтролера відбуваються її переривання в результаті спрацювання апаратних переривань для обслуговування таких пристроїв як таймер реального часу, який спрацьовує кожну секунду для підрахунку вимірної ємності та переривання, яке виникає в результаті отримання у вхідний буфер приймача будь-якої інформації, яка надходить по послідовній шині UART.

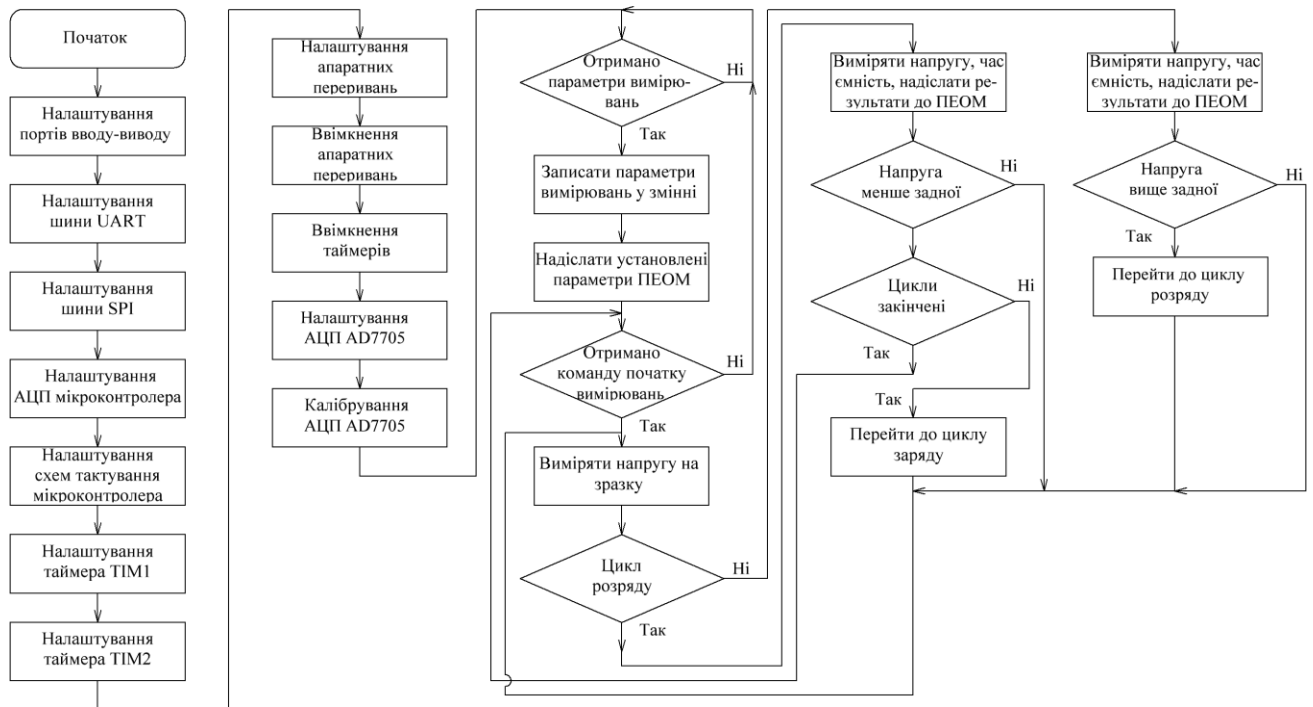


Рисунок 2.7 – Блок-схема роботи мікроконтролеру

Використання апаратних переривань дозволяє реалізувати систему виміру в реальному часі, тобто в цьому випадку незалежно від зовнішніх чинників та процесу виконання основної програми пріоритет буде залишатися за найбільш

важливими процесами: прийом керуючих команд та вимірювання часових інтервалів, які визначають повну ємність.

### **2.3 Автоматизація процесу вимірювання**

Автоматизація отримання результатів дослідження електрофізичних характеристик вторинних хімічних джерел струму багаторазової дії дозволяє мінімізації вплив на процес вимірювання оперативного персоналу.

Задача дослідження полягає у визначенні таких параметрів:

- струм заряду-розряду;
- напруга, при якій має розпочатися процес заряду після розрядження;
- напруга, при якій має розпочатися процес розряду після зарядження;
- кількість циклів заряд-розряд.

Керуючі параметри передаються до вимірювального комплексу від ПЕОМ, після чого перший надсилає у відповідь отримані параметри. Якщо параметри не співпадають в результаті можливих завад або інших чинників, то програмне забезпечення (ПЗ) має повторно передати дані або, в разі неможливості, проінформувати про виникнення помилки.

У разі успішної передачі параметрів для вимірювання оператор за допомогою ПЗ ЕОМ може здійснити початок процесу вимірювання параметрів досліджуваного об'єкту. В цьому випадку вимірювальний комплекс здійснює щосекундне вимірювання таких параметрів як напруга на комірці, підрахунок поточної ємності, формування вихідного пакету даних, який передається до ПЗ ПЕОМ.

Оскільки в процесі вимірювання отримання будь-яких команд від ПЗ ПЕОМ не є необхідним, а лише вимагається вчасне приймання сформованого інформаційного пакету, то основною вимогою і необхідною умовою для керуючої програми є використання апаратних переривань від послідовного прийомо-передавача у окремому потоці. Саме реалізація подібного ПЗ дозволить уникнути втрати даних.

Розширити функціонал спроектованого гальваностату можливо за допомогою використання бездротового інтерфейсу стандарту IEEE802.11 (Wi-Fi), який замінить послідовну шину для передачі даних до ПЕОМ за рахунок реалізації інтерфейсу UART – Wi-Fi. Відмова від фізичних ліній передач робить систему більш гнучкою та універсальною у використанні, але водночас може знизити завадостійкість.

### 3 ІНФОРМАЦІЙНА СИСТЕМА МОНІТОРИНГУ ХАРАКТЕРИСТИК ЛІТІЙ-ІОННИХ БАТАРЕЙ

Інформаційна система складається з мобільного додатку, веб-додатку, загальної бази даних та експериментальної установки. Архітектура інформаційної системи представлена на рисунку 3.1.

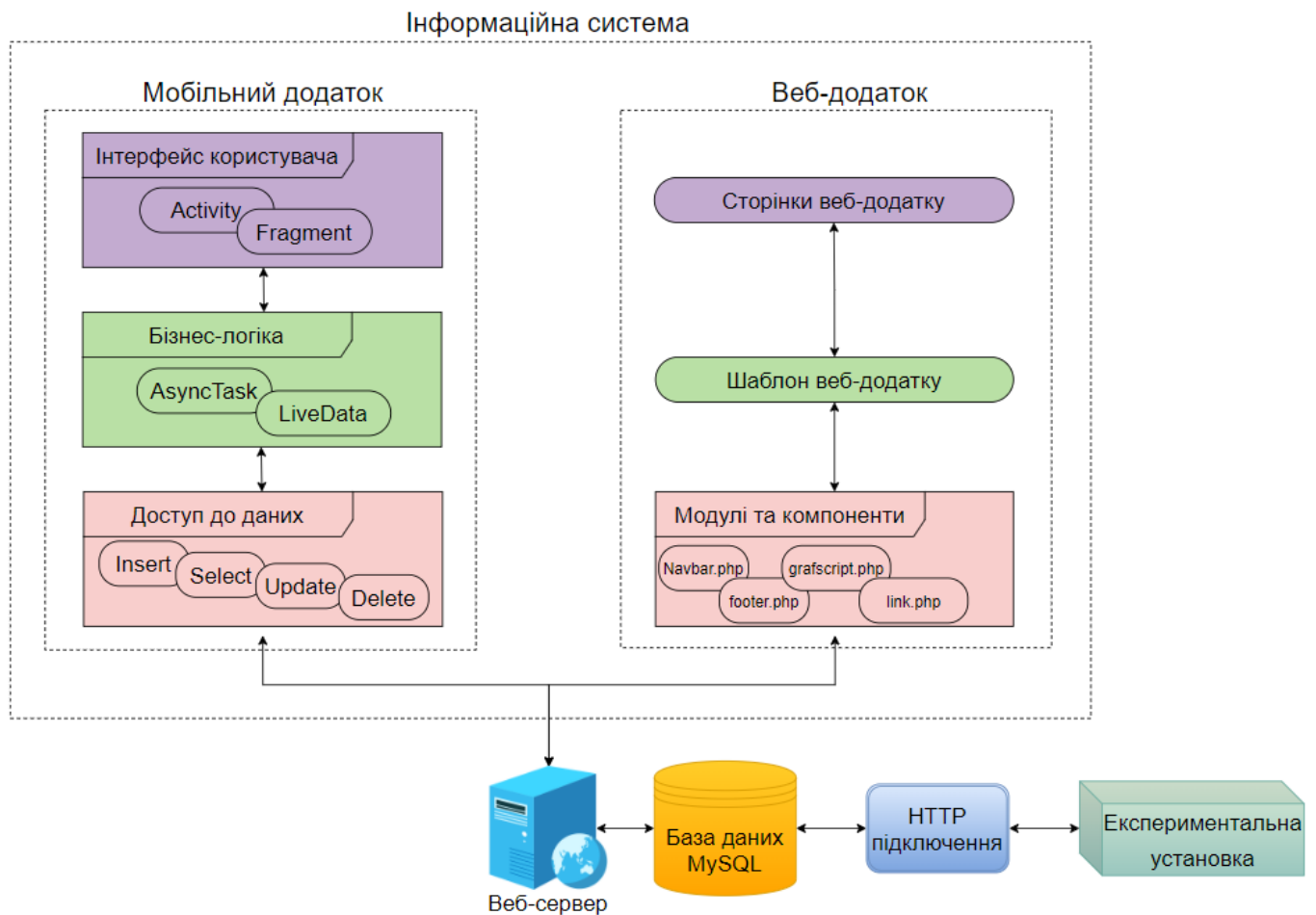


Рисунок 3.1 – Архітектура інформаційної системи

Під час проведення експерименту вольт-амперні характеристики батареї зчитуються експериментальною установкою, після чого з інтервалом у декілька секунд, експериментальна установка передає їх через підключення до бази даних. У свою чергу, мобільний та веб-додаток, за допомогою модулів та компонентів звертається до бази даних через веб-сервер, щоб отримати значення вольт-амперних характеристик у вигляді потоку значень напруги та струму. Після



отримання вольт-амперних характеристик з бази даних, мобільний та веб-додаток представляє отримані значення в графічній залежності напруги від струму.

Програмна реалізація мобільного додатку відбувалася у середовищі Android Studio з використанням мови програмування Java. Структура розробленого мобільного додатку, яка складається з файлів та папок представлена на рисунку 3.1.

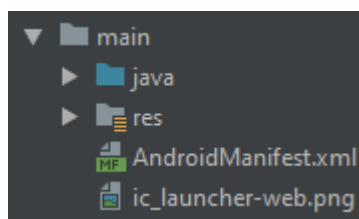


Рисунок 3.2 – Структура файлів мобільного додатку

Згідно зображеної на рисунку 3.2 структури файлів було виконано опис основних папок мобільного додатку, які представлено в таблиці 3.1.

Таблиця 3.1 – Опис структури мобільного додатку.

| № | Папка  | Короткий опис вмісту папки                               |
|---|--------|--|
| 1 | «java» | Містить усі файли класів, активності та конструкторів.   |
| 2 | «res»  | Містить усі файли ресурсів, розмітки активності та меню. |

Опис сторінок мобільного додатку представлено в таблиці 3.2.

Таблиця 3.2 – Опис сторінок мобільного додатку.

| № | Назва сторінки       | Короткий опис сторінки  |
|---|----------------------|---|
| 1 | Сторінка авторизації | Дана сторінка містить: <ul style="list-style-type: none"> <li>– поле для введення логіну;</li> <li>– поле для введення пароля;</li> <li>– кнопку увійти.</li> </ul> |

|   |   |  |
|---|---|--|
| 2 | Меню  | <p>Дана сторінка містить:</p> <ul style="list-style-type: none"> <li>– посилання до сторінки історії замірів;</li> <li>– посилання до сторінки поточного заміру;</li> <li>– посилання до сторінки архіву замірів;</li> <li>– посилання до сторінки налаштувань;</li> <li>– кнопку виходу з системи.</li> </ul> |
| 3 | Сторінка списку історії замірів                         | <p>Дана сторінка містить:</p> <ul style="list-style-type: none"> <li>– список історії замірів.</li> </ul>  |
| 4 | Сторінка детального перегляду обраного заміру з історії | <p>Дана сторінка містить:</p> <ul style="list-style-type: none"> <li>– поле з інформацією про замір з історії;</li> <li>– графічна залежність напруги від струму;</li> <li>– кнопка завантаження залежності;</li> <li>– кнопка видалення заміру з історії.</li> </ul>  |
| 5 | Сторінка поточного заміру за умови його проведення      | <p>Дана сторінка містить:</p> <ul style="list-style-type: none"> <li>– поле з інформацією про поточний замір;</li> <li>– графічна залежність напруги від струму;</li> <li>– кнопка зупинки експерименту.</li> </ul>  |
| 6 | Сторінка поточного заміру за умови його відсутності     | <p>Дана сторінка містить:</p> <ul style="list-style-type: none"> <li>– поле з інформацією про відсутність експерименту;</li> <li>– кнопка з відображенням форми для запуску експерименту.</li> </ul>   |
| 7 | Сторінка списку архівів замірів                         | <p>Дана сторінка містить:</p> <ul style="list-style-type: none"> <li>– список архівів замірів;</li> <li>– кнопку створення архіву.</li> </ul>  |
| 8 | Сторінка детального перегляду обраного архіву           | <p>Дана сторінка містить:</p> <ul style="list-style-type: none"> <li>– поле з інформацією про архів;</li> <li>– кнопка завантаження архіву;</li> <li>– кнопка видалення архіву.</li> </ul>   |

|          |   |   |
|----------|---|---|
| <b>9</b> | Сторінка<br>налаштувань<br>графічної залежності | Дана сторінка містить: <ul style="list-style-type: none"> <li>– поле для налаштування розміру тексту;</li> <li>– поле для налаштування кольору тексту;</li> <li>– поле для налаштування розміру крапки;</li> <li>– поле для налаштування кольору крапки.</li> </ul> |
|----------|---|---|

У свою чергу програмна реалізація веб-додатку відбувалася з використанням таких технологій як: HTML, CSS, PHP, JavaScript та Sublime. Структура розробленого веб-додатку, яка складається з файлів та папок представлена на рисунку 3.3.

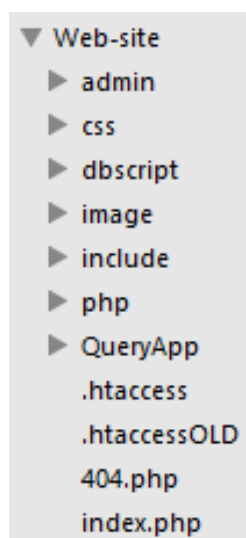


Рисунок 3.3 – Структура файлів веб-додатку

Згідно зображеної на рисунку 3.3 структури файлів було виконано опис основних папок веб-додатку, які представлено в таблиці 3.3.

Таблиця 3.3 – Опис структури веб-додатку.

| №        | Папка   | Короткий опис вмісту папки                                |
|----------|---------|---|
| <b>1</b> | «admin» | Містить файл сторінки адміністрування та її налаштування. |
| <b>2</b> | «css»   | Містить файли опису стилю сторінок.                       |

|   |            |  |
|---|------------|--|
| 3 | «dbscript» | Містить файли підключення до бази даних.               |
| 4 | «image»    | Містить файли використовуваних зображень.              |
| 5 | «include»  | Містить файли підключення меню та футеру.              |
| 6 | «php»      | Містить файли скриптів для веб-додатку.                |
| 7 | «QueryApp» | Містить файли скриптів з базою для мобільного додатку. |

Опис сторінок веб-додатку представлено в таблиці 3.4.

Таблиця 3.4 – Опис сторінок веб-додатку.

| № | Назва сторінки                              | Короткий опис сторінки  |
|---|---|---|
| 1 | Головна сторінка                            | Дана сторінка містить: <ul style="list-style-type: none"> <li>– логотип веб-додатку у header;</li> <li>– зображення по центру;</li> <li>– підпис розробника у footer;</li> <li>– кнопку переходу до сторінки авторизації.</li> </ul>  |
| 2 | Сторінка авторизації                        | Дана сторінка містить: <ul style="list-style-type: none"> <li>– поле для введення логіну;</li> <li>– поле для введення пароля;</li> <li>– кнопку увійти.</li> </ul>   |
| 3 | Головна сторінка після успішної авторизації | Дана сторінка містить: <ul style="list-style-type: none"> <li>– кнопки переходу до сторінки історії замірів, поточного заміру та кабінету.</li> </ul>   |
| 4 | Сторінка кабінету                           | Дана сторінка містить: <ul style="list-style-type: none"> <li>– форму для додавання результатів експерименту;</li> <li>– форму для додавання користувачів зі встановленням їм прав доступу;</li> <li>– форму для створення архіву даних по поточним результатам;</li> </ul> |

|   |                           |  |
|---|---------------------------|--|
|   |                           | <ul style="list-style-type: none"> <li>– таблицю з цифровими даними результатів експерименту та полем пошуку;</li> <li>– таблицю з переліком користувачів;</li> <li>– таблицю з пошуком експерименту;</li> <li>– таблицю з замірами;</li> <li>– таблицю з поточним заміром;</li> <li>– таблицю з історією замірів;</li> <li>– таблицю з переліком архівів;</li> <li>– форму для редагування та видалення результатів за порядковим номером;</li> <li>– форму для редагування та видалення користувачів за ідентифікатором;</li> <li>– форму для редагування назви заміру за порядковим номером;</li> <li>– форму для видалення заміру з історії;</li> <li>– форму для завантаження архівного заміру;</li> <li>– форму для видалення архівного заміру.</li> </ul> |
| 5 | Сторінка історії замірів  | <p>Дана сторінка містить:</p> <ul style="list-style-type: none"> <li>– кнопку для відображення форми вводу номеру заміру для подальшого перегляду;</li> <li>– візуалізацію обраного заміру у графічній залежності.</li> </ul>  |
| 6 | Сторінка поточного заміру | <p>Дана сторінка містить:</p> <ul style="list-style-type: none"> <li>– кнопку для відображення форми керування поточним заміром;</li> <li>– візуалізація поточного заміру у графічній залежності.</li> </ul>   |

Після запуску мобільного додатку дослідник потрапляє до вікна авторизації (рис. 3.4), тому що доступ до інформаційної системи закритий.

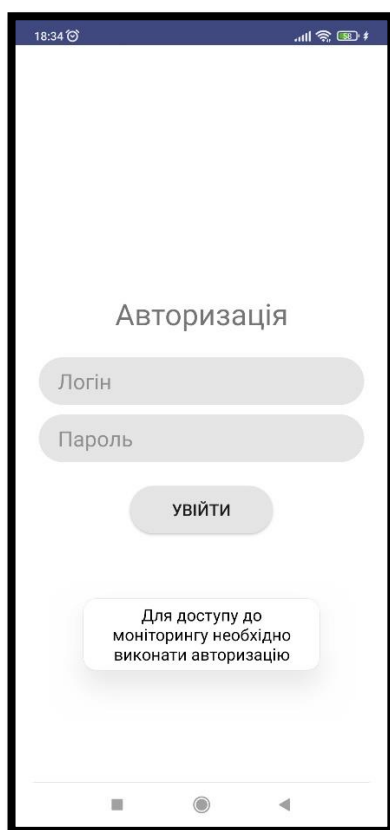


Рисунок 3.4 – Вікно авторизації мобільного додатку

У результаті успішної авторизації у мобільному додатку відкривається вікно з переглядом списку історії замірів (рис. 3.5).

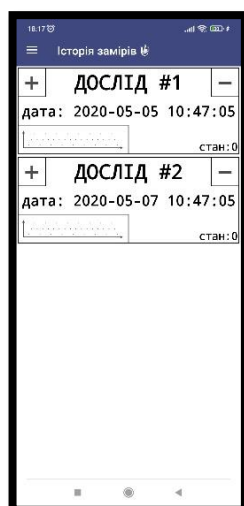


Рисунок 3.5 – Вікно зі списком історії замірів

Для детального перегляду заміру потрібно натиснути на нього у списку, після чого відкриється вікно з детальним переглядом, яке включає в себе опис початкових налаштувань обраного експерименту та візуалізацію вольт-амперних характеристик у графічній залежності струму від напруги (рис. 3.6).

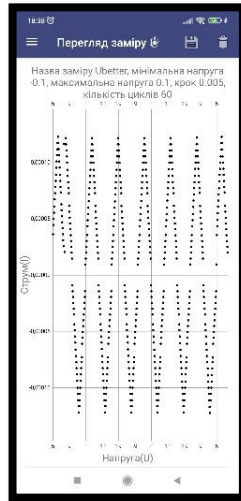


Рисунок 3.6 – Детальний перегляд обраного історичного заміру

Під час детального перегляду історичних замірів дослідник може зберегти графічну залежність у вигляді зображення (рис. 3.7), а також видалити обраний замір (рис. 3.8).

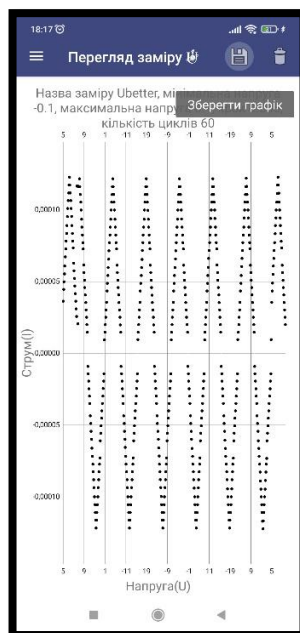


Рисунок 3.7 – Збереження графічної залежності



Рисунок 3.8 – Видалення обраного заміру

Для перегляду поточного заміру передбачено два сценарії. Перший сценарій відображається якщо на момент переходу до сторінки поточного заміру ніяких експериментів не було розпочато (рис. 3.9). У цьому випадку дослідник може розпочати експеримент заповнивши форму початкових налаштувань (рис. 3.10).



Рисунок 3.9 – Вікно поточного заміру для першого сценарію



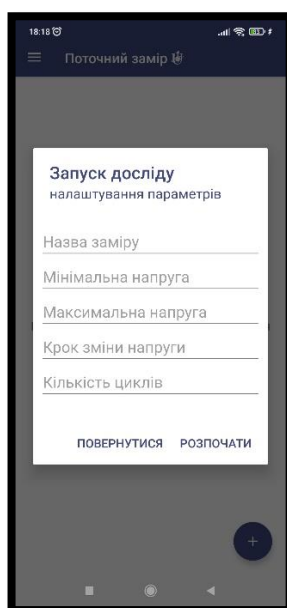


Рисунок 3.10 – Форма початкових налаштувань

Другий сценарій відображається якщо на момент переходу до сторінки поточного заміру відбувається проведення експерименту (рис. 3.11). У цьому випадку досліднику представляється опис початкових налаштувань обраного експерименту та візуалізацію вольт-амперних характеристик у графічній залежності струму від напруги у реальному часі, а також можливість зупинити експеримент (рис. 3.12).

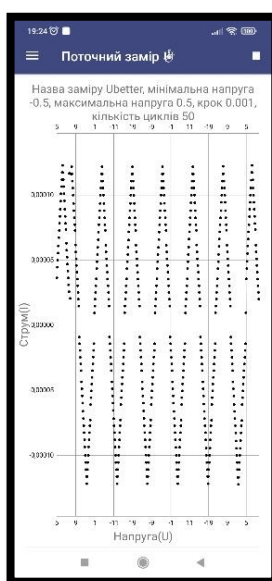


Рисунок 3.11 – Вікно поточного заміру для другого сценарію



Рисунок 3.12 – Зупинка поточного заміру

Для перегляду списку архівів потрібно перейти до сторінки архівних замірів, після чого буде відображено список архівів (рис. 3.13). Також, на сторінці архівних замірів є можливість створити новий архів, по поточним результатам, заповнивши відповідну форму (рис. 3.14).

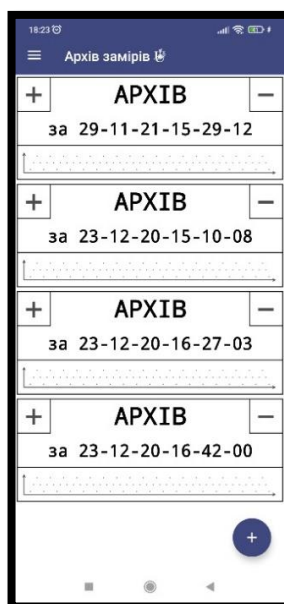


Рисунок 3.13 – Вікно зі списком архівних замірів

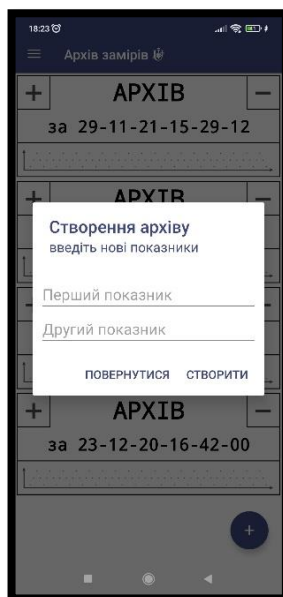


Рисунок 3.14 – Форма створення нового архіву

Для детального перегляду архівного заміру потрібно натиснути на нього у списку, після чого відкриється вікно з детальним переглядом, під час якого дослідник може завантажити обраний архів для перегляду (рис. 3.15), а також видалити обраний архів (рис. 3.16).

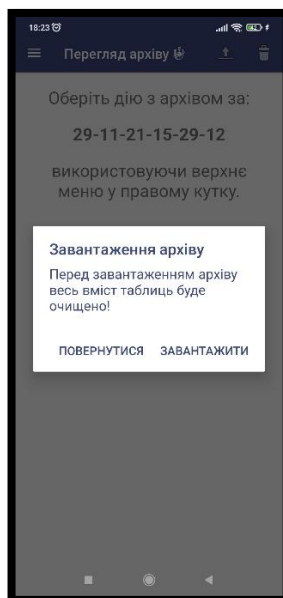


Рисунок 3.15 – Завантаження обраного архіву



Рисунок 3.16 – Видалення обраного архіву

Для налаштування візуалізації графічної залежності потрібно перейти до сторінки налаштувань та змінити потрібні параметри (рис. 3.17).

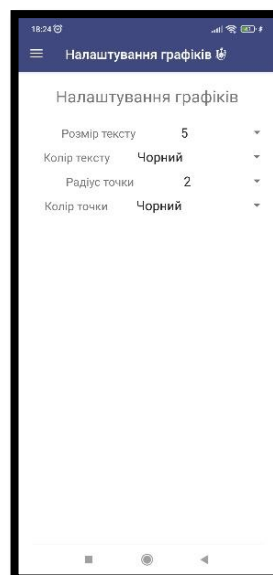


Рисунок 3.17 – Вікно налаштувань графічної залежності

За необхідності дослідник може вийти з поточного облікового запису натиснувши на кнопку виходу (рис. 3.18).

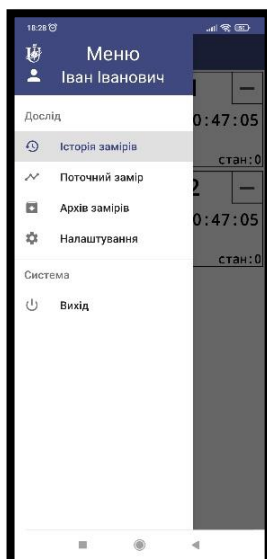


Рисунок 3.18 – Вихід з поточного облікового запису

На відміну від мобільного додатку, після запуску веб-додатку дослідник потрапляє спочатку на головну сторінку (рис. 3.19), після чого вже буде можливість перейти до сторінки авторизації (рис. 3.20).



Рисунок 3.19 – Головна сторінка веб-додатку

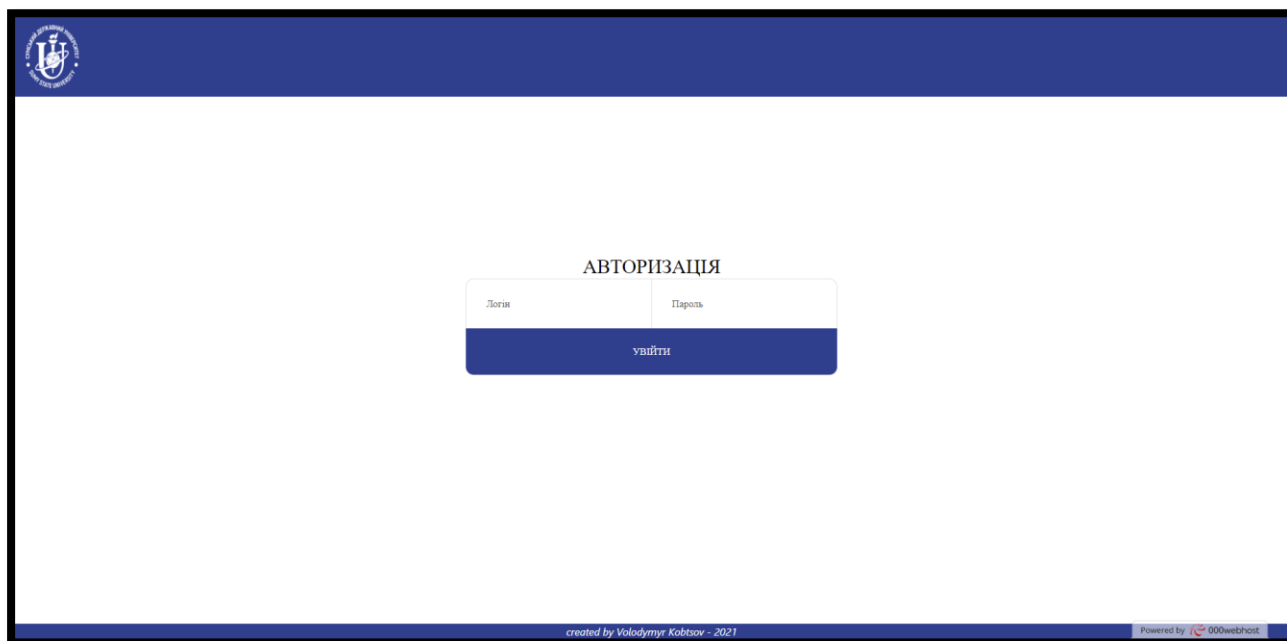


Рисунок 3.20 – Сторінка авторизації веб-додатку

У результаті успішної авторизації у веб-додатку відкривається сторінка з посиланнями на історію замірів, поточний замір та кабінет (рис. 3.17).



Рисунок 3.17 – Сторінка з посиланнями

На сторінці історії замірів, дослідник по унікальному ідентифікатору заміру, який можна побачити у кабінеті, може переглянути обраний історичний замір (3.18).

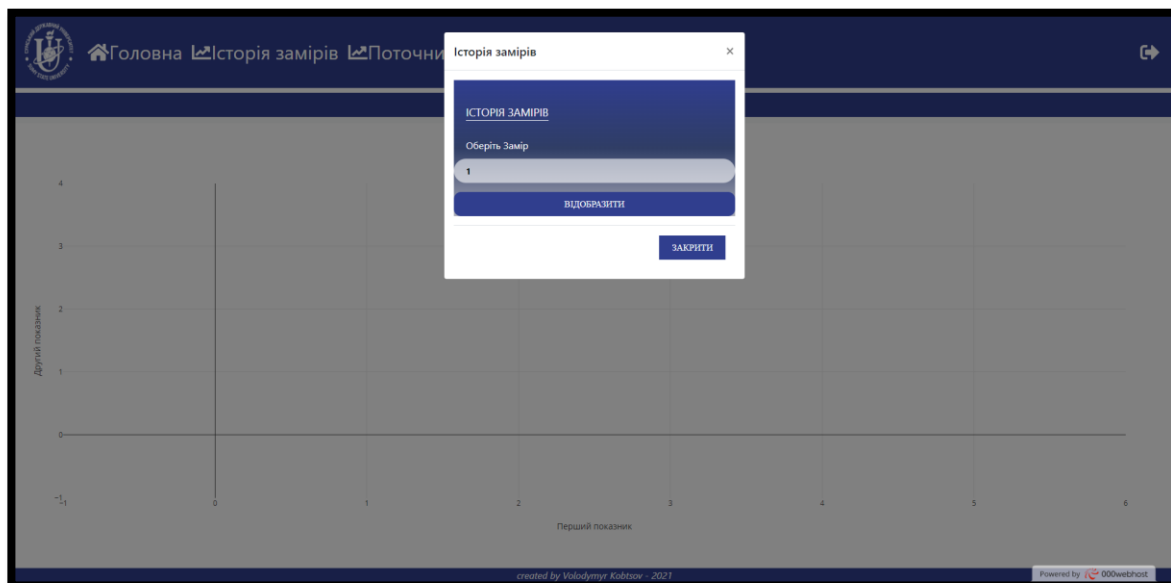


Рисунок 3.18 – Сторінка з історією замірів

Для перегляду сторінки з поточним заміром передбачено два сценарії. Перший сценарій відображається якщо на момент переходу до сторінки поточного заміру ніяких експериментів не було розпочато (рис. 3.19). У цьому випадку дослідник може розпочати експеримент заповнивши форму початкових налаштувань (рис. 3.20).



Рисунок 3.19 – Сторінка поточного заміру для першого сценарію

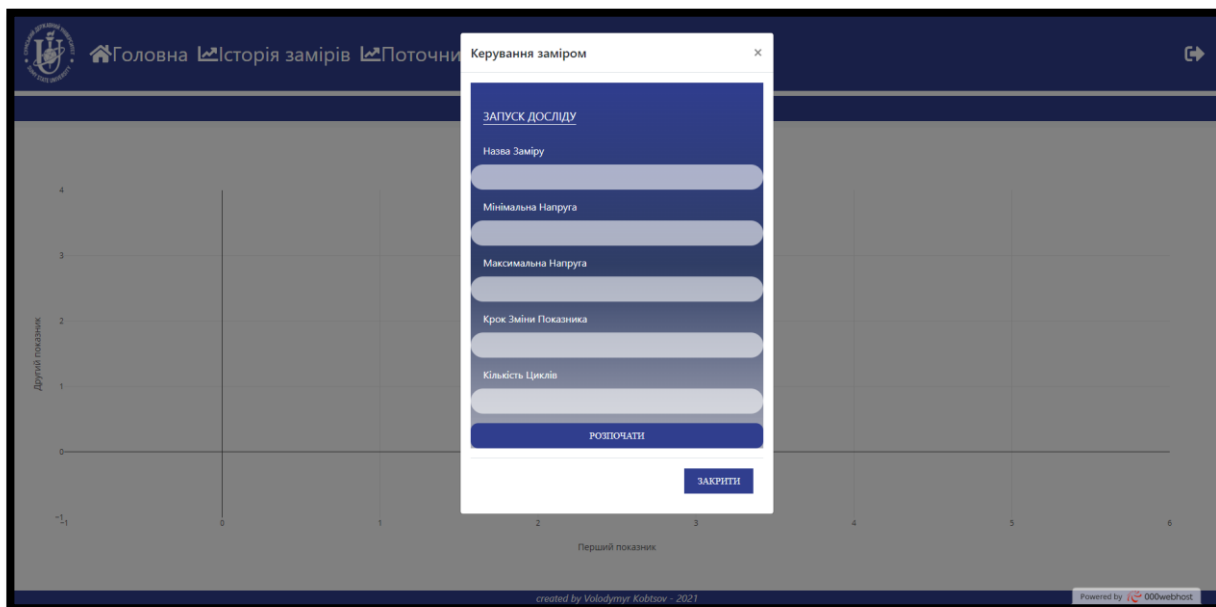


Рисунок 3.20 – Форма початкових налаштувань

Другий сценарій відображається якщо на момент переходу до сторінки поточного заміру відбувається проведення експерименту (рис. 3.21). У цьому випадку досліднику представляється візуалізація вольт-амперних характеристик у графічній залежності струму від напруги, а також можливість зупинити експеримент (рис. 3.22).

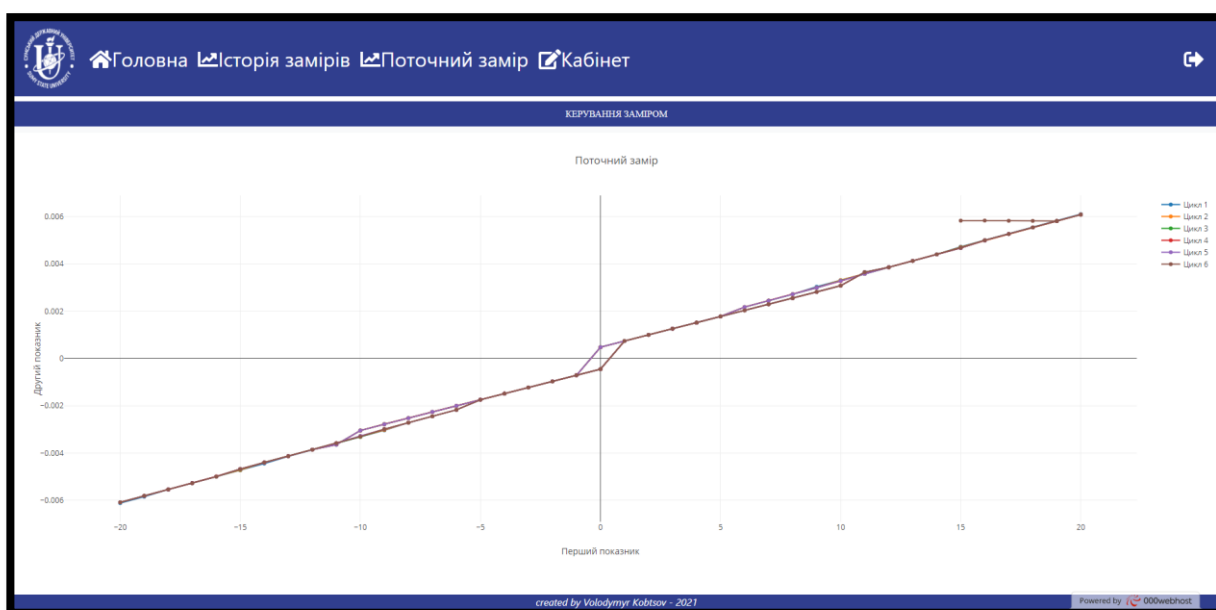


Рисунок 3.21 – Вікно поточного заміру для другого сценарію





Рисунок 3.22 – Зупинка поточного заміру

У будь який момент часу дослідник може виконувати налаштування графічної залежності у верхній панелі налаштувань. Панель містить завантаження графічної залежності у вигляді зображення, зміну масштабу, переміщення та відновлення налаштувань за замовчуванням (рис. 3.23). Лістинг програмного коду мобільного та веб-додатку наведено в Додатку В.

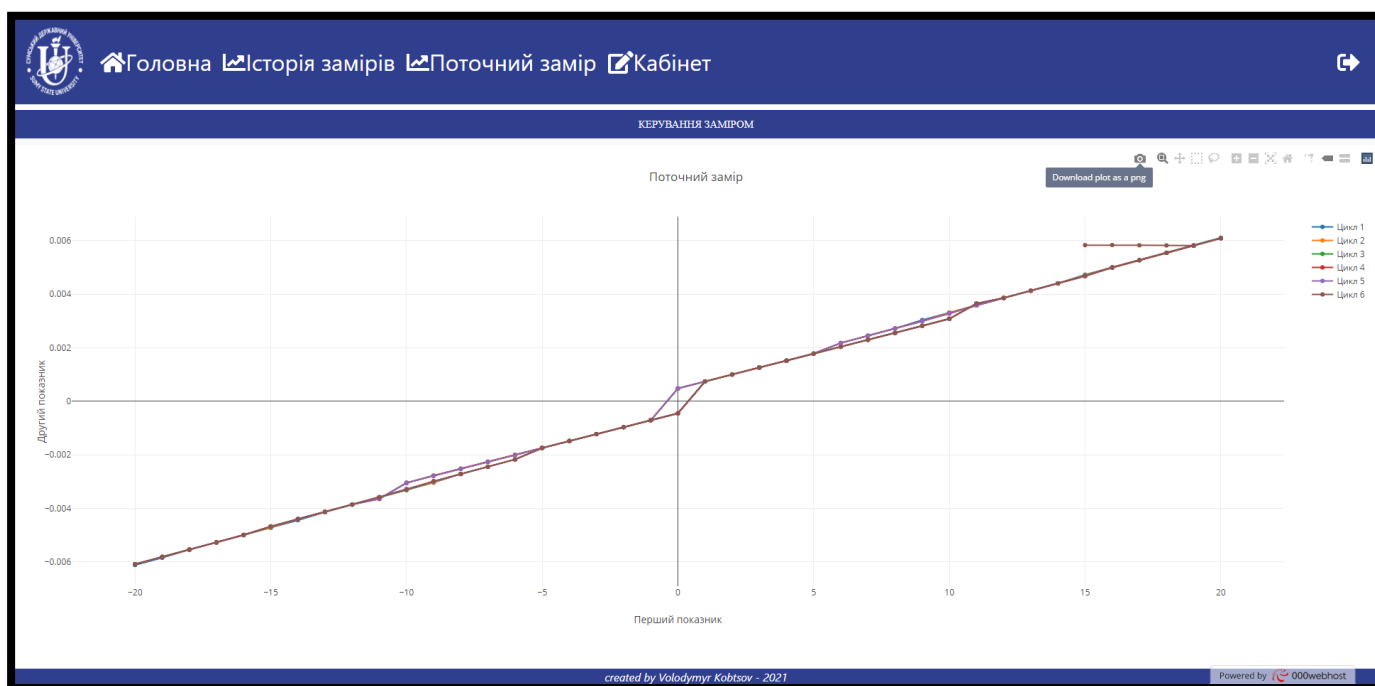


Рисунок 3.23 – Панель налаштування графічної залежності

### 3.1 Адміністрування інформаційної системи

Адміністрування інформаційної системи відбувається на сторінці особистого кабінету. Адміністратор може виконувати:

Додавання результатів експерименту (рис. 3.24).

The screenshot shows a web interface for adding experiment results. The top navigation bar includes links for 'Головна', 'Історія замірів', 'Поточний замір', and 'Кабінет'. The main content area is divided into three columns. The left column contains a form with the following fields: 'Номер Заміру', 'Номер Циклу', 'Час', 'Напруга', and 'Струм'. Below these fields is a 'ДОДАТИ РЕЗУЛЬТАТ' button. The middle column contains a header with 'РЕЗУЛЬТАТИ', 'КОРИСТУВАЧІ', 'ПОШУК', 'ЗАМІР', 'ПОТОЧНИЙ ЗАМІР', and 'ІСТОРІЯ ЗАМІРІВ', and a sub-header 'АРХІВ'. The right column contains a header with 'РЕЗУЛЬТАТИ', 'КОРИСТУВАЧІ', 'ЗАМІР', 'ІСТОРІЯ ЗАМІРІВ', 'ЗАВАНТАЖИТИ АРХІВ', and 'ВИДАЛИТИ АРХІВ'. The footer of the page includes the text 'created by Volodymyr Kobtsov - 2021' and 'Powered by 000webhost'.

Рисунок 3.24 – Додавання експерименту

Додавання користувачів зі встановленням їм прав доступу (рис. 3.25).

The screenshot shows a web interface for adding a user. The top navigation bar is identical to the previous screenshot. The main content area is divided into three columns. The left column contains a form with the following fields: 'Ім'я', 'Прізвище', 'Логін', 'Пароль', and 'Ранг (1 -Адміністратор, 2-Користувач)'. Below these fields is a 'ДОДАТИ КОРИСТУВАЧА' button. The middle column contains a header with 'РЕЗУЛЬТАТИ', 'КОРИСТУВАЧІ', 'ПОШУК', 'ЗАМІР', 'ПОТОЧНИЙ ЗАМІР', and 'ІСТОРІЯ ЗАМІРІВ', and a sub-header 'АРХІВ'. The right column contains a header with 'РЕЗУЛЬТАТИ', 'КОРИСТУВАЧІ', 'ЗАМІР', 'ІСТОРІЯ ЗАМІРІВ', 'ЗАВАНТАЖИТИ АРХІВ', and 'ВИДАЛИТИ АРХІВ'. The footer of the page includes the text 'created by Volodymyr Kobtsov - 2021' and 'Powered by 000webhost'.

Рисунок 3.25 – Додавання користувачів

Створення архіву даних по поточним результатам (рис. 3.26).

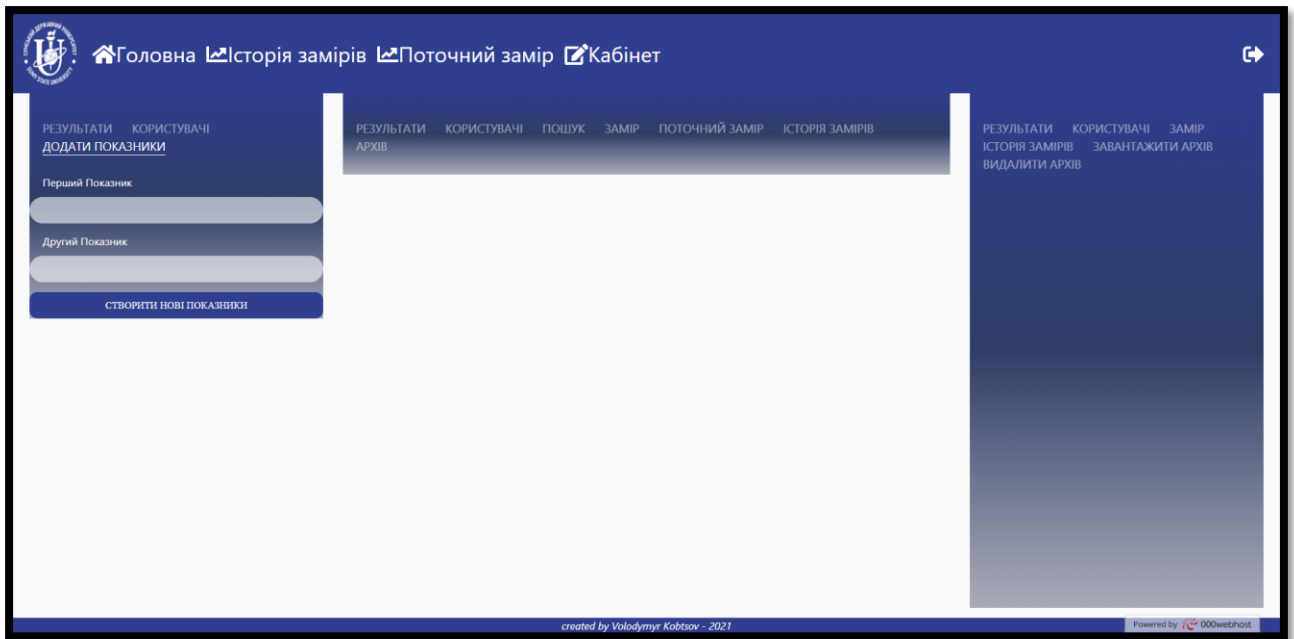


Рисунок 3.26 – Створення архіву

Редагувати та видаляти результати за порядковим номером (рис. 3.27).

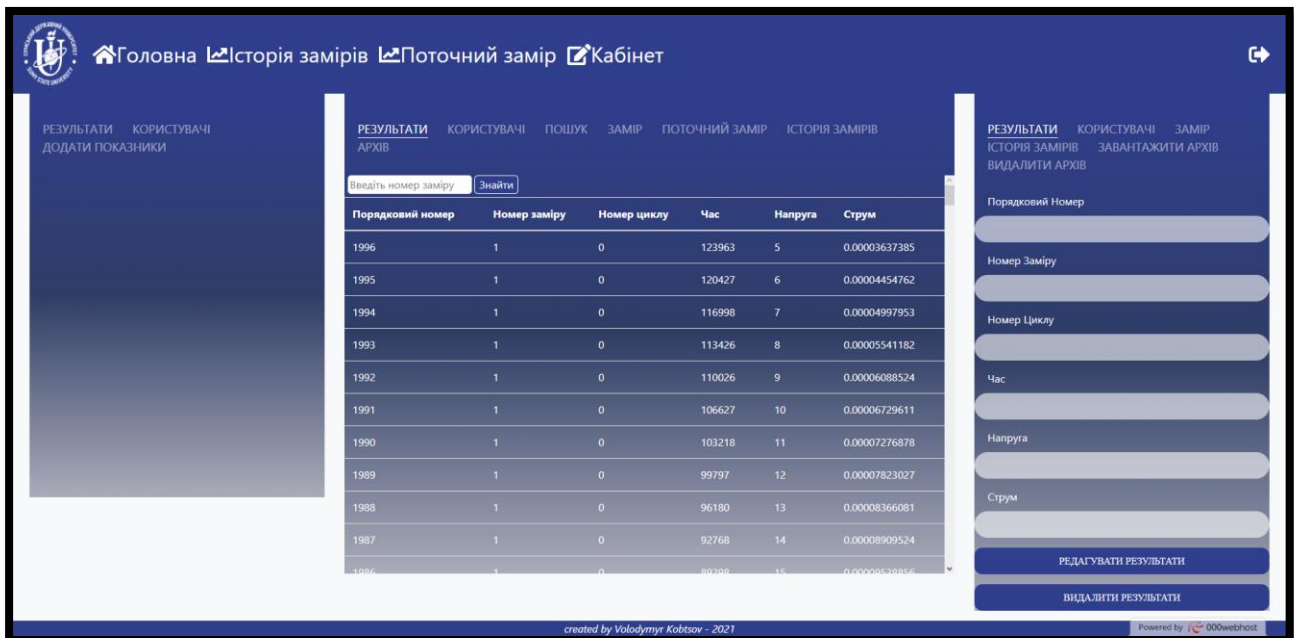


Рисунок 3.27 – Редагування та видалення результатів

Редагувати та видаляти користувачів за ідентифікатором (рис. 3.28).

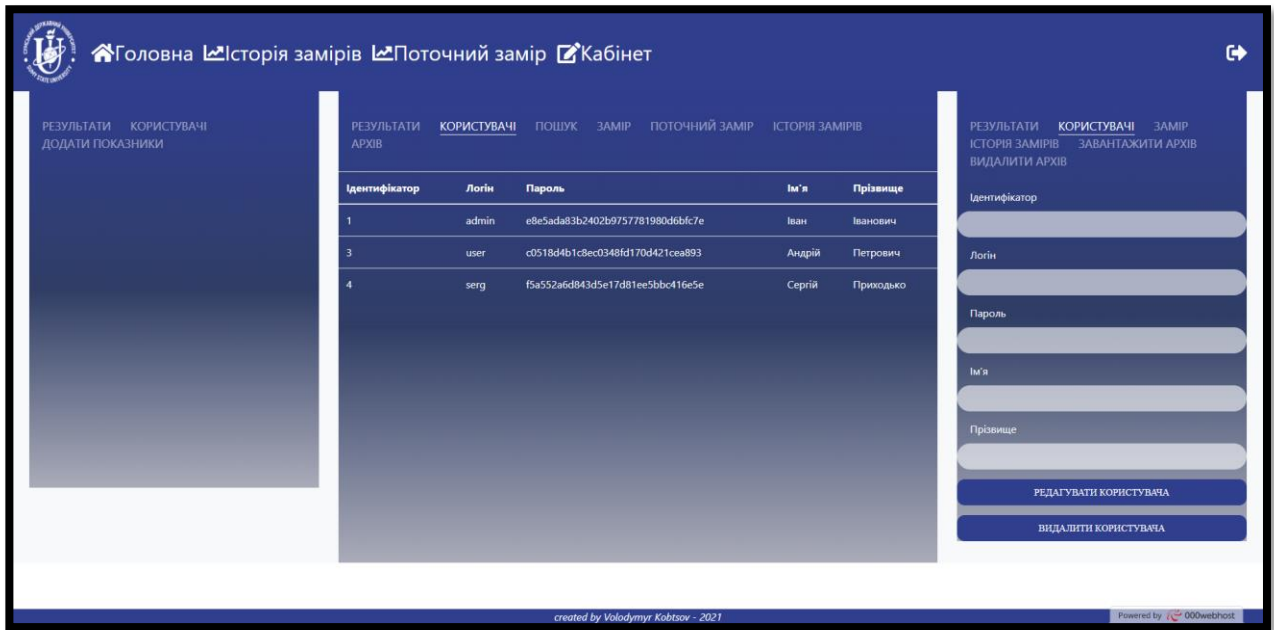


Рисунок 3.28 – Редагування та видалення користувачів

Редагувати назву заміру за порядковим номером (рис. 3.30).



Рисунок 3.29 – Зміна назви заміру

Видаляти замір з історії (рис. 3.30).

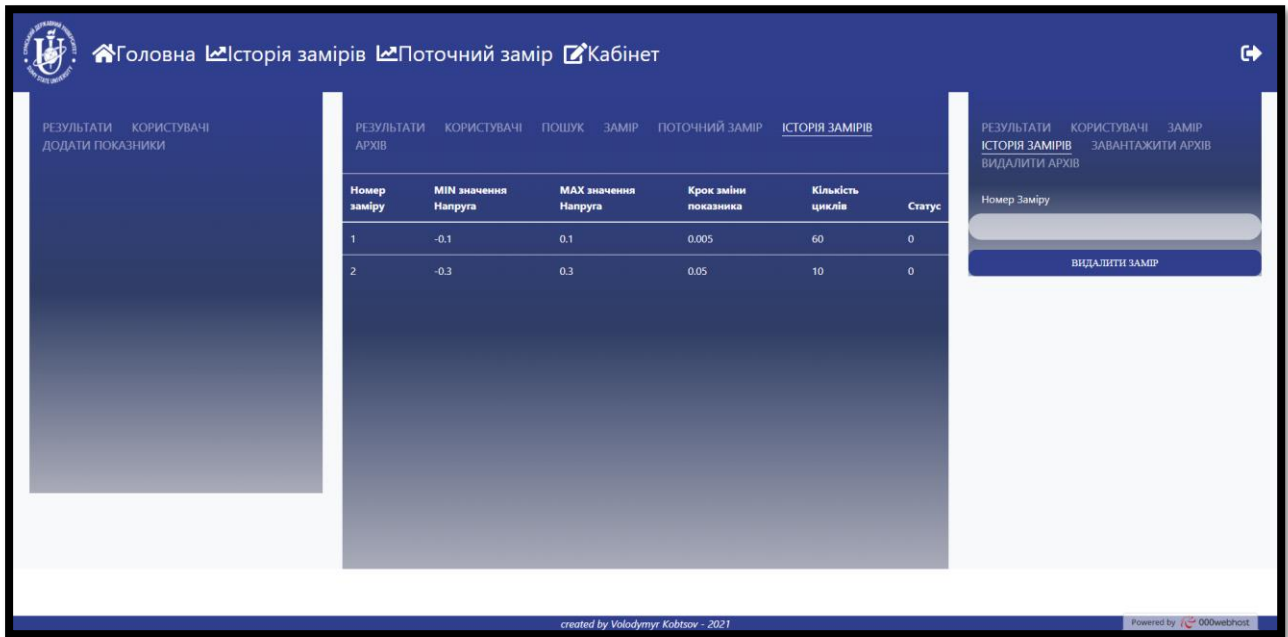


Рисунок 3.30 – Видалення заміру

Завантажувати архівні заміри (рис. 3.31).



Рисунок 3.31 – Завантаження архіву

Видаляти архівні заміри (рис. 3.32).



Рисунок 3.32 – Видалення архіву

## 4 ДОСЛІДЖЕННЯ ЕЛЕКТРОХІМІЧНИХ ВЛАСТИВОСТЕЙ ЕЛЕКТРОДІВ

### 4.1 Дослідження електрохімічних властивостей електродів

На основі отриманих пористих наноструктур вуглецю, окислів металів та композитів на їх основі були виготовлені аноди з метою дослідження електрохімічних характеристик літій-іонної батареї укомплектованої розробленими нами електродами. Для електрохімічних вимірів ми використовували трьох-електродну тестову комірку, де у якості катоду та електроду відліку (проти-електроду) застосовувалася пластинка високочистого 99.9% літію, а анод був виготовлений на основі синтезованих нами пористих наноструктур. Слід зазначити, що перед нанесенням активного матеріалу електроду, на підкладки із скла попередньо наносився шар хрому, який відповідав за високу адгезію контактів. Потім одночасно з поступовим зниженням швидкості нанесення плівки хрому за допомогою магнетронного розпилення зверху наносилась стійка до хімічних взаємодій плівка золота.

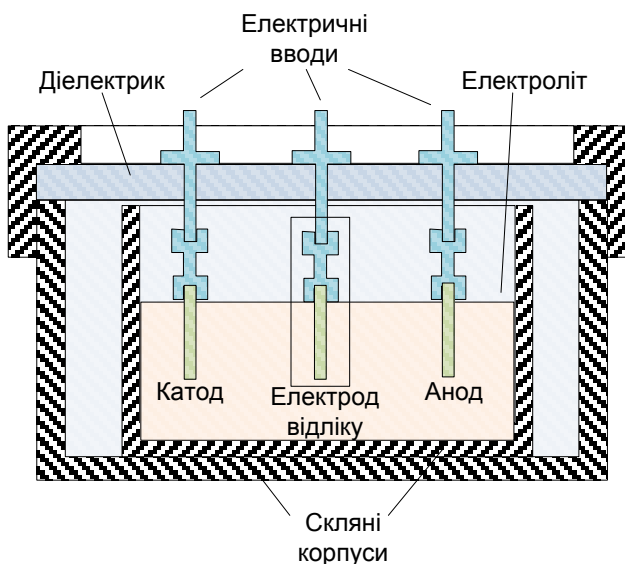


Рисунок 4.1– Схематичне зображення трьохелектродної електрохімічної комірки

Щоб визначити масу активного матеріалу, яка на наступному етапі буде необхідна для розрахунку значення питомої ємності, підкладку з нанесеним

контактним шаром Au/Cr зважували до осадження матеріалу аноду та після нанесення активного шару. У якості електроліту, що містить іони літію, виготовлявся розчин літієвої солі  $\text{LiPF}_6$  або  $\text{LiBaF}_4$  концентрацією 1М. При цьому у якості розчинників слугувала суміш етиленкарбонату та диметилкарбонату у співвідношенні 50 на 50 %. Всі виміри проводились при кімнатній температурі. Схематична будова електрохімічної комірки представлена на рисунку 4.1.

Добре відомо, що літій являється дуже активним хімічним елементом. Так при його взаємодії з повітрям на поверхні швидко утворюється оксид та нітрид літію. Разом з тим у літій-іонних акумуляторах повинен використовуватися тільки літій без домішок. Також слід зазначити, що взаємодія електроліту на основі солей  $\text{LiPF}_6$  та  $\text{LiBaF}_4$  з повітрям може привести до утворення токсичного фторо-водню HF. З цих причин для збирання електрохімічних комірок та подальшого дослідження функціональних характеристик виготовлених анодів було використано інертне середовище у вигляді аргону при атмосферному тиску. Реалізація зазначених вище умов підготовки та подальших електро-хімічних досліджень проводилася з залученням вакуумної рукавичної шафи. При цьому зазначена шафа спочатку відкачувалась до необхідного вакууму, а потім проводився напуск чистого аргону до атмосферного тиску.

Для визначення електрохімічних характеристик матеріалу електроду, використовують два основні виміри: циклічні зарядження-розрядження вольт-амперні характеристики. Слід зазначити, що для кожного матеріалу аноду та електроліту існує певний діапазон напруги, у якому електрохімічна комірка працює ефективно. Так надмірне зарядження або розрядження призводить до деградації матеріалу аноду та розкладання електроліту з утворенням небажаних сполук, які можуть бути вогнебезпечними та вибухонебезпечними. Для електродів окислів металів та вуглецю, діапазон робочих напруг при кімнатній температурі відповідає значенням від 0.01 В до 3 В. Літій-іонні акумулятори заряджалися у режимі постійний струм-постійна напруга (ПС-ПН). Так спочатку до електрохімічної комірки прикладали постійний струм, при цьому напруга на комірці зростає до тих пір поки не буде досягнуте граничне її значення 3 В. Потім



на комірці утримували постійне значення напруги, при цьому значення струму зменшується. Постійну напругу підтримували до тих пір поки струм не зменшиться до значень, близьких до нуля. Потім полярність струму змінювали на протилежну, і до комірки прикладали постійний зворотній струм, до тих пір поки значення напруги не зменшиться до 0.01 В. В подальшому утримували постійне значення напруги 0.01 В до моменту наближення значення струму до нуля. Таким чином ми отримували перший цикл заряджання-розряджання. Помноживши час першого циклу на значення струму можна отримати значення ємності в одиницях міліампер за годину (мА·год). Так як значення ємності залежить від маси речовини, то при виконанні проекту, для характеристики ємнісних властивостей електрода були встановлені питомі значення ємності в одиницях міліампер за годину на один грам активної речовини аноду (мА·год/г). Як правило значення ємності першого циклу максимальне і при циклічному заряджання-розряджання значення ємності поступово зменшуються. Далі цикли розрядки-зарядки повторюють до тих пір, поки значення ємності на комірці встановиться приблизно на одному рівні.

Методика дослідження циклічних вольт-амперних характеристик полягала у наступному. Значення напруги на електрохімічній комірці змінювалось від 0.01 до 3 В із кроком 0.1 мВ/с. При цьому реєструвалось значення відповідного струму. На першому циклі у зразку відбуваються електрохімічні процеси, які є незворотними, далі цикли зміни напруги повторюються. За декілька наступних циклів електрохімічні характеристики електрода стабілізуються, і вольт-амперні характеристики наступних циклів співпадають між собою, що свідчить про оборнені реакції інтерколяції-деінтерколяції літію у матеріал аноду.

Електрохімічні реакції на поверхні аноду при використанні у якості активного матеріалу оксиду цинку, оксиду нікелю та вуглецю мають наступний вигляд:



На основі кривих заряджання-розряджання було визначене значення ємності батареї для кожного циклу та побудовані графіки залежності ємності батареї від номеру циклу. Також проведені дослідження циклічної вольтаметрії в діапазоні напруги від 0,01 В до 3 В з кроком зміни напруги 0.1 мВ/с.

## 4.2 Структурно-морфологічні характеристики та елементний склад анодів на основі Zn/ZnO/NiO, Zn/ZnO, ZnO/NiO Zn/C і Ni/C та результати досліджень циклів зарядки-розрядки і ВАХ створених на їх основі електрохімічних комірок

### 4.2.1 Аноди на основі металевих оксидів

Використовуючи трьох-електродну електрохімічну комірку, що представлена на рис. 4.1 були проведені дослідження циклів зарядки та розрядки літій-іонних акумуляторів та циклічних ВАХ за умови використання в якості анодів наносистем Zn/ZnO/NiO, Zn/ZnO і ZnO/NiO.

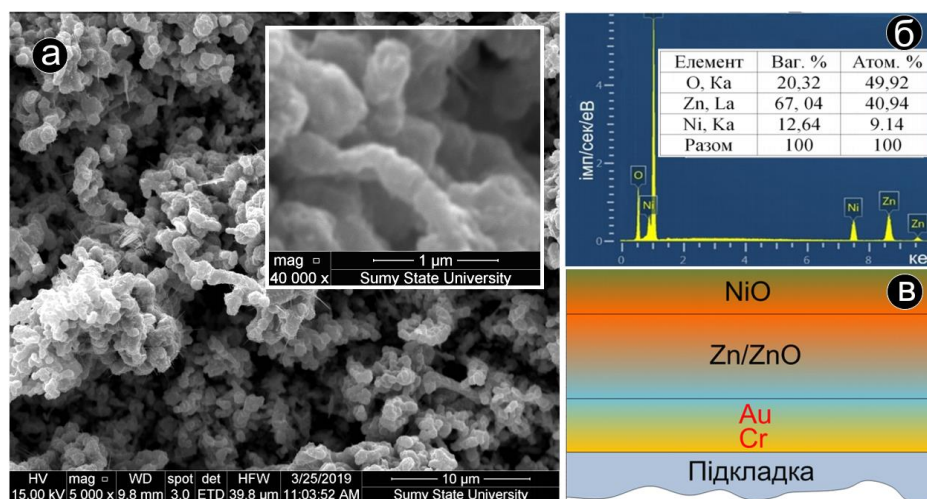


Рисунок 4.2 – РЕМ-зображення морфології поверхневого шару аноду на основі Zn/ZnO/NiO (а), результати елементного аналізу трьохшарової системи за допомогою EDX-спектру (б) та схематичне зображення пошарової будови аноду (в)

При цьому була віддана перевага пористим системам, які мають підвищену механічну міцність, тобто складаються з пов'язаних між собою об'ємних структурних фрагментів.

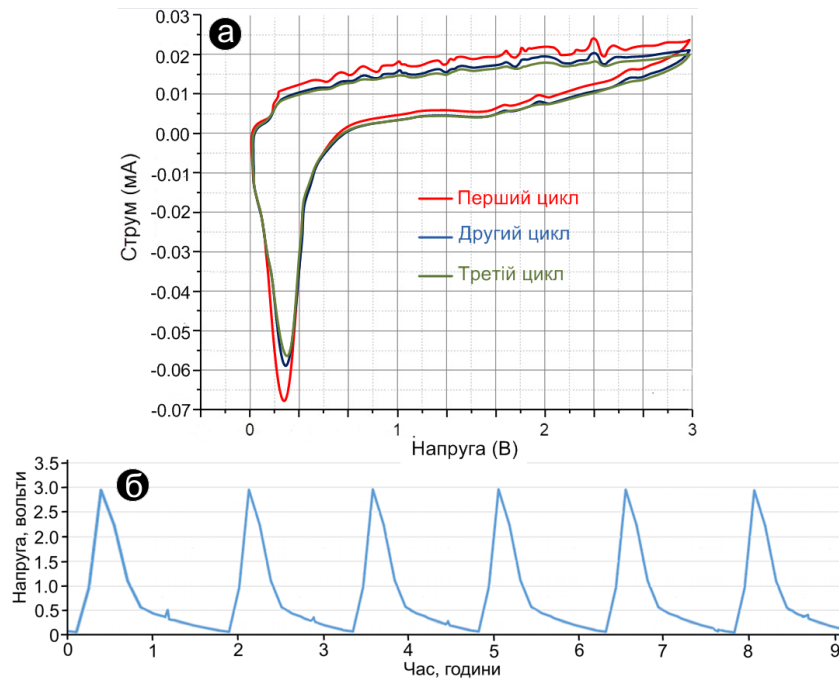


Рисунок 4.3 – Циклічні ВАХ (а) та цикли розрядки-зарядки (б) при використанні анодів на основі Zn/ZnO/NiO (струм зарядки  $I=2$  мА, маса активної речовини аноду  $\Delta m = 0.0125$  г)

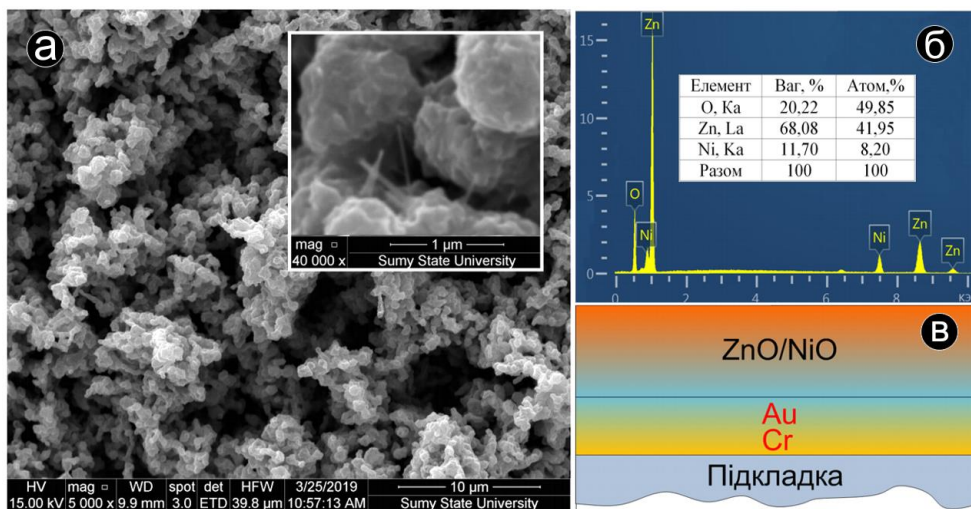


Рисунок 4.4 – РЕМ-зображення морфології поверхневого шару аноду на основі ZnO/NiO (а), результати елементного аналізу двохшарової системи за допомогою EDX-спектру (б) та схематичне зображення пошарової будови аноду (в)

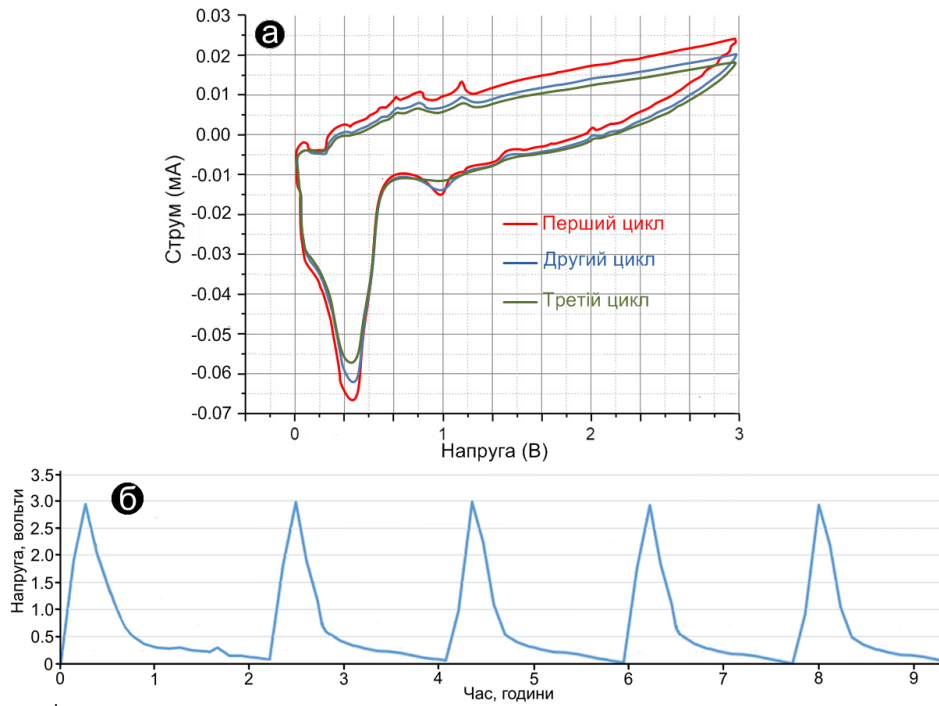


Рисунок 4.5 – Циклічні ВАХ (а) та цикли розрядки-зарядки (б) при використанні анодів на основі ZnO/NiO (струм зарядки  $I=2$  мА, маса активної речовини аноду  $\Delta m = 0.0243$  г)

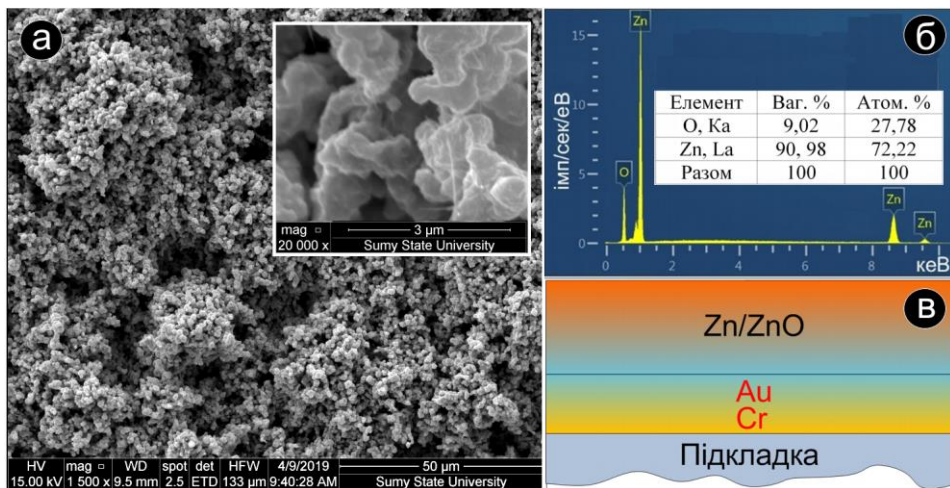


Рисунок 4.6 – РЕМ-зображення морфології поверхневого шару аноду на основі Zn/ZnO (а), результати елементного аналізу двошарової системи за допомогою EDX-спектру (б) та схематичне зображення пошарової будови аноду (в)

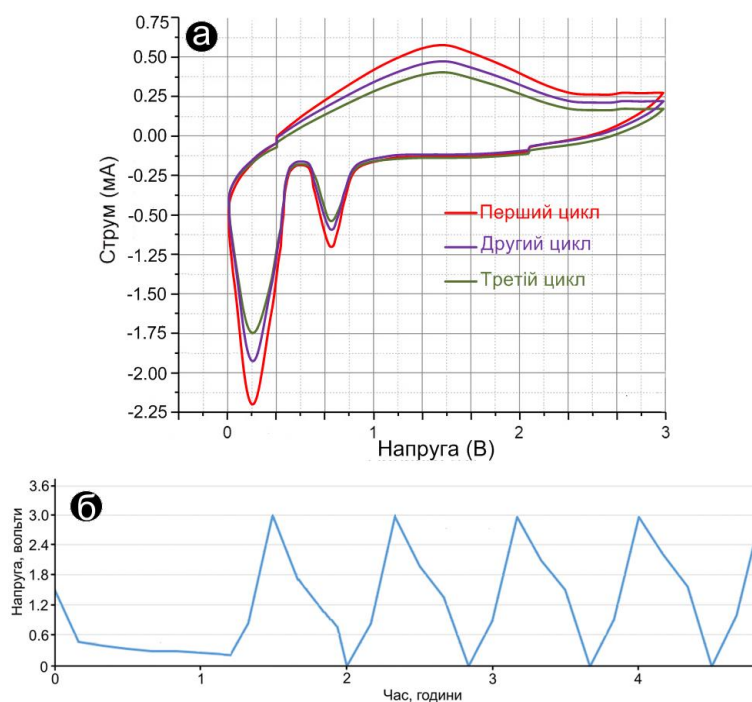


Рисунок 4.7 – Циклічні ВАХ (а) та цикли розрядки-зарядки (б) при використанні анодів на основі Zn/ZnO (струм зарядки  $I=2$  мА, маса активної речовини аноду  $\Delta m=0.016$  г)

На рис. 4.2, 4.4 і 4.6 представлені РЕМ-зображення морфології поверхневого шару аноду на основі Zn/ZnO/NiO, ZnO/NiO та Zn/ZnO (а), результати елементного аналізу (б) та схематичне зображення пошарової будови анодів (в). Раніше нами встановлено, що структури на основі мереж з нанониток ZnO мають низькі механічні характеристики, що знижує стабільність циклів зарядки-розрядки. Маючи це на увазі з усіх отриманих при виконанні проекту структурно-морфологічних характеристик оксидів були відібрані системи, що мають поєднані між собою високу пористість і підвищену механічну стійкість. Зазначені оптимальні структурно-морфологічні характеристики були отримані на основі поєднаних між собою об'ємних мікро- та нанокристалів (див. позицію а на рис. 4.2, 4.4 і 4.6). При цьому елементний склад активної частини анодів відповідав необхідній стехіометрії (див. позицію б на рис. 4.2, 4.4 і 4.6).

В усіх випадках циклічні ВАХ мали значний гістерезис, що можливо при впливі на процес їх формування не тільки зовнішньої різниці потенціалів (див. позицію а на рис. 4.3, 4.5 і 4.7). Так струм електрохімічної комірки при переході

від підвищення зовнішньої різниці потенціалів до її зниження значною мірою підтримується деінтеркаляцією літію з аноду, що в підвищеному режимі знижує результуючий струм і тим самим підтверджує попередній процес зарядки. Наявність від'ємних струмів при малих значеннях зовнішньої різниці потенціалів на етапі її зниження також підтверджує наявність зарядки акумуляторної системи. При цьому звертає на себе увагу наявність максимумів та мінімумів, що вказує на різні характери процесів ітеркаляції-деінтеркаляції літію, які в свою чергу визначаються пошаровою будовою активної частини анодів (див. позицію в на рис. 4.2, 4.4 і 4.6).

На рис. 4.8 наведене фото працюючої електрохімічної комірки, що також підтверджує працездатність отриманих анодів на основі Zn/ZnO/NiO.

Ємнісні характеристики анодів на основі Zn/ZnO/NiO, ZnO/NiO та Zn/ZnO, що розраховані з використанням циклів зарядки-розрядки представлені на рис 4.14. Порівняльний аналіз залежності ємності від кількості циклів зарядки-розрядки вказує на те, що аноди на основі Zn/ZnO/NiO мають найбільш високі ємнісні характеристики (до 240 мА год./грам). Разом з тим після 50 циклів зарядки-розрядки найбільш стабільні і разом з тим найбільш низькі ємнісні характеристики мають аноди на основі Zn/ZnO, що представляють собою напівокислені наносистеми Zn.



Рисунок 4.8 – Фото працюючої електрохімічної комірки при використанні аноду на основі Zn/ZnO/NiO

#### 4.2.2. Аноди на основі турбостратного графіту

Подальші дослідження були проведені з використанням анодів на основі вуглецевих матеріалів. Коротко проведемо аналіз їх можливих структурно-морфологічних характеристик та перспектив використання в ЛІА. Нові архітектури вуглецевих матеріалів за підтримкою нанотехнологій мають такі переваги при використанні в літій іонних акумуляторах, як збільшення електрохімічно активних ділянок і особливо посилення здатності буферувати напругу [46]. Нановуглецеві матеріали зі специфічною морфологією та нанорозміром мають високу електричну провідність та теплопровідність, відмінну механічну гнучкість та гарну хімічну стійкість. Інженерія нановуглецевих матеріалів для анодної та катодної архітектури являє собою ефективний принцип для підвищення продуктивності ЛІА, включаючи більшу потужність, швидкість руху та стабільну циклічність [47, 48].

Нановуглецеві матеріали – це наносистеми з різноманітними морфологічними характеристиками, мезопористі вуглецеві матеріали, а також нові сімейства вуглецю - фулерени, вуглецеві нанотрубки (ВНТ) та графен. Крім того, до цього класу матеріалів належать нанокомпозити на основі нановуглецевих матеріалів та металів, оксидів металів (ОМ), сульфідів металів та полімерів [49, 50]. Через низьку розмірність нановуглецеві матеріали, як правило, мають специфічні морфологічні властивості. Нановимірність із специфічною морфологією надає їм унікальних фізичних та хімічних властивостей: високу електричну провідність та теплопровідність, відмінну механічну гнучкість та хімічну стійкість. З цих причин новітні вуглецеві наноматеріали широко вивчаються з перспективою використання у якості електродів ЛІА.

Відомо, що вуглецеві аноди ЛІА мають найнижчий потенціал 0,1 В по відношенню до літію. Цей факт обумовлює його широке застосування у якості аноду літій-іонних акумуляторів з високою напругою. Для покращення характеристик аноду із вуглецю, використовують різноманітні його форми: графен, нанотрубки та нанопористі структури [51-58].

Для отримання нановуглецевих матеріалів, розроблено багато нових технологій їх синтезу. До останніх слід віднести такі методи, як хімічне осадження із газової фази (CVD) [59], синтез із газової фази без підкладки [60], відновлення оксиду графіту (GO) [61], плазмово-дуговий метод [62], сольвотермальний метод [63], термічне розщеплення гранул SiC [64] тощо.

З метою отримання вуглецевих наноматеріалів з керованими структурно-морфологічними характеристиками, значний інтерес представляють дослідження селективних процесів при конденсації відносно високоенергетичних атомів (іонів) вуглецю [65]. При цьому в області взаємодії високоенергетичної частинки з ростоюю поверхнею, яка захоплює досить обмежену кількість атомів, виникають температура та інші термодинамічні параметри, які відрізняються від подібних параметрів всієї термодинамічної системи пар-конденсат. Слід також зазначити, що при підвищенні енергії конденсованих частинок посилюється індивідуальність характеру їх взаємодії з ростоюю поверхнею. У зв'язку з цим теоретичні уявлення про прояв фазової селективності у вуглецевих конденсатах базуються на феноменологічному підході, а використовувані при цьому ймовірності процесів визначаються експериментально.

В якості вуглецевої складової для анодів ЛІА нами була синтезована та використана структура пористого турбостратного графіту (ТГ). Його будова базується на міцно з'єднаних між собою кластерах з декількох графенових шарів. Важлива особливість будови кластерів полягає в тому, що паралельні графенові шари розміщені один відносно одного не регулярно, як в графіті. При цьому графенові шари в кластері можуть бути розвернуті на деякий кут навколо перпендикулярної до графенових шарів осі. Двовимірність в упаковці кластерів і визначає їх турбостратність. В силу того, що відстань між графеновими шарами в ТГ більша ніж у графіті на  $0,06 \text{ \AA}$  нтеркаляція іонів літію в ТГ буде визначати менші розбухання, що може запобігати подальше руйнування анодів ЛІА. При цьому в кластерах ТГ атоми вуглецю на краях графенів реалізують міцні хімічні зв'язки з сусідніми графеновими площинами, що є передумовою механічної стійкості ТГ. Модельні уявлення про структуру структуру ТГ представлені на



рис. 4.8. Виходячи з того, що синтез ТГ відносно простий при виконанні проекту він був вибраний в якості активної частини анодів.

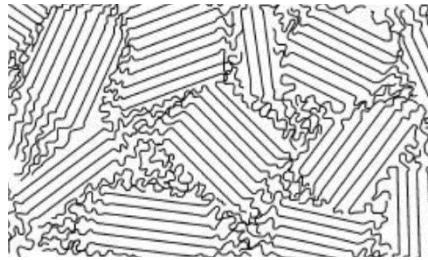


Рисунок 4.9 – Модельні уявлення про будову ТГ

При цьому для підвищення пористості шари ТГ формувалися на пористих шарах Zn або Ni за методикою, яка викладена в розд. 4.1, морфологія поверхні та елементний склад яких представлена на рис. 4.10 і 4.12 (позиції а, б і в). Як у попередніх випадках циклічні ВАХ мають добре виражений гістерезис, що визначається дією процесів інтеркаляції-деінтеркаляції літію в ТГ.

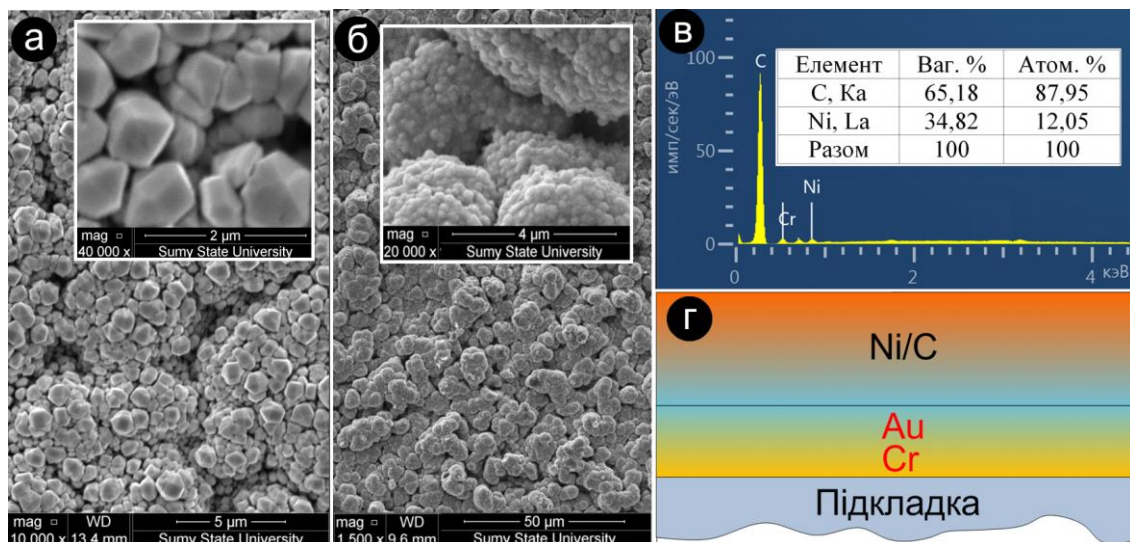


Рисунок 4.10 – РЕМ-зображення морфології поверхневого шару аноду на основі Ni (а) та нанесеного на нього шару C (б), результати елементного аналізу двошарової системи за допомогою EDX–спектру (в) та схематичне зображення пошарової будови аноду (г)

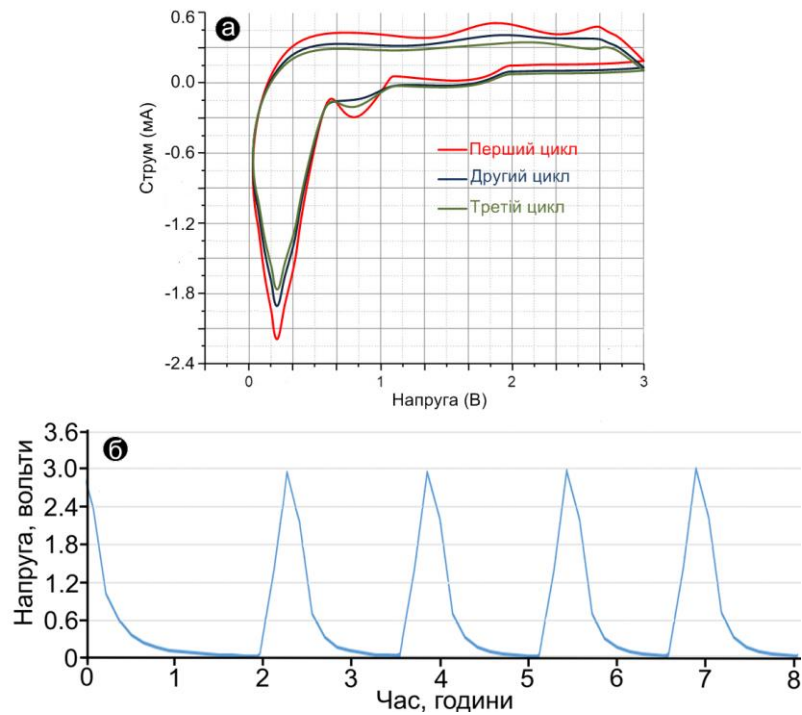


Рисунок 4.11– Циклічні ВАХ (а) та цикли розрядки-зарядки (б) при використанні анодів на основі Ni/C (струм зарядки  $I=2$  мА, маса активної речовини аноду  $\Delta m = 0.0149$  г)

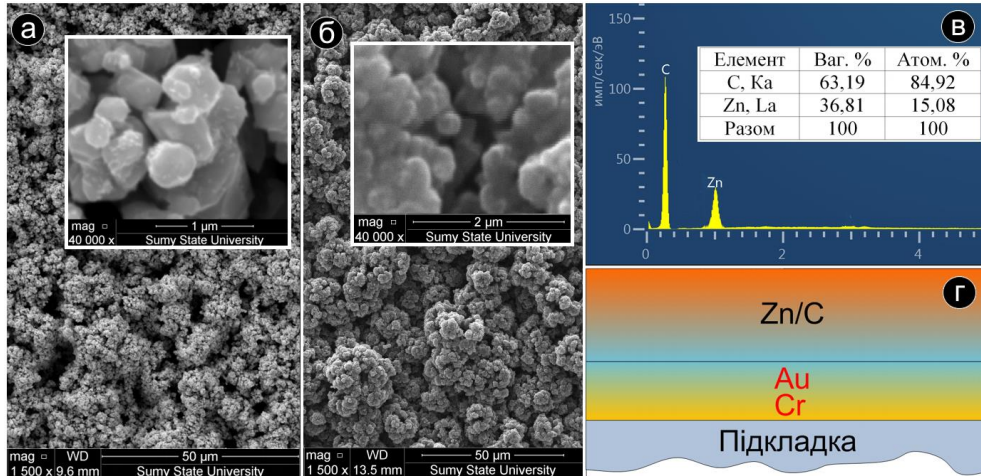


Рисунок 4.12 – РЕМ-зображення морфології поверхневого шару аноду на основі Zn (а) та нанесеного на нього шару C (б), результати елементного аналізу двошарової системи за допомогою EDX-спектру (в) та схематичне зображення пошарової будови аноду (г)

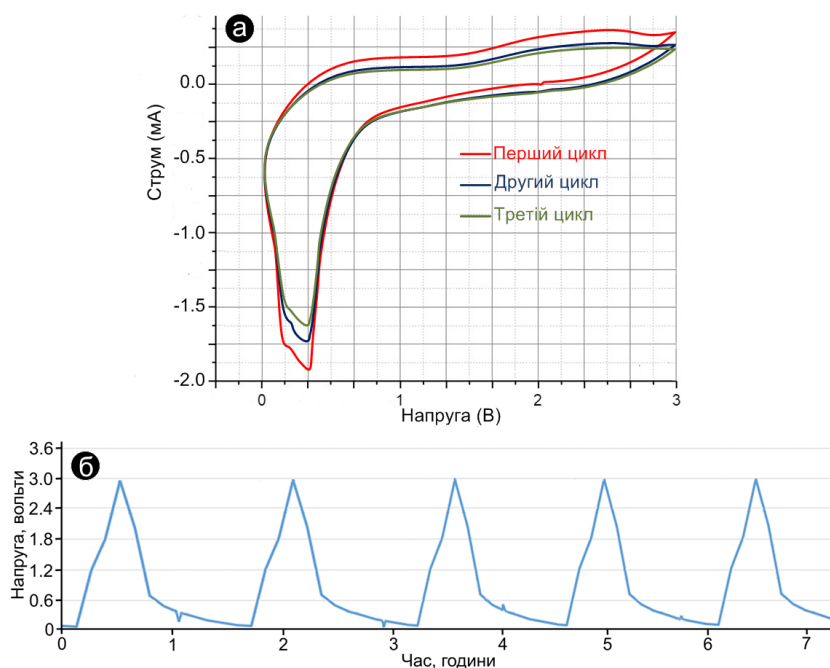


Рисунок 4.13 – Циклічні ВАХ (а) та цикли розрядки-зарядки (б) при використанні анодів на основі Zn/C (струм зарядки  $I=2$  мА, маса активної речовини аноду  $\Delta m = 0.0126$  г)

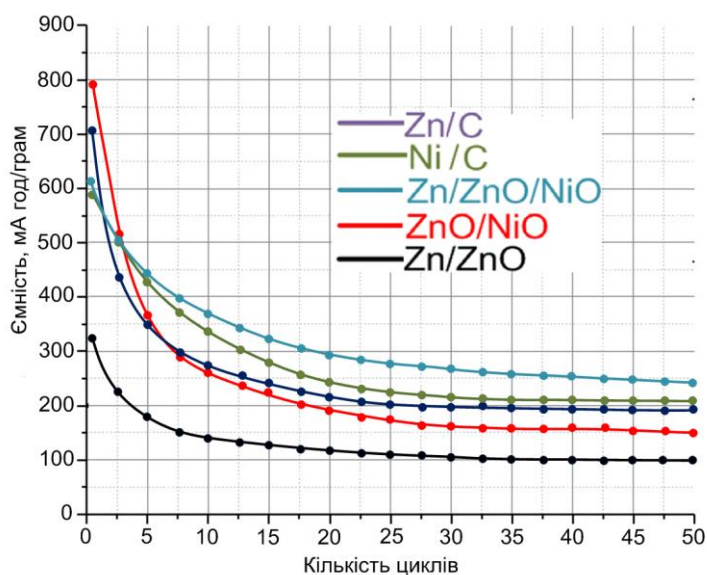


Рисунок 4.14 – Залежності ефективності модельних зразків ЛІА від кількості циклів розрядки-зарядки при використанні анодів на основі пористих систем Zn/C, Ni/C, Zn/ZnO/NiO, ZnO/NiO та Zn/ZnO

Розрахунки залежних від кількості циклів зарядка-розрядки ємнісних характеристик для анодів на основі ТГ представлені на рис. 4.14. З наведених

акспериментальних даних можна зробити висновок про те, що ЛПА на основі пористого ТГ мають приблизно однакову питому ємність ( $\sim 190 \div 210$  мА год/грам) за наявністю підвищеної стабільності (див. рис.4.14).

## ВИСНОВКИ

1. У роботі спроектовано та розроблено гальваностат для автоматизованого отримання зарядно-розрядних характеристик вторинних джерел струму. Автоматизований вимірювальний комплекс дозволяє здійснювати дослідження параметрів акумуляторів струмом в діапазоні 0-50 мА та дискретністю зміни 0,01 мА, роздільна здатність вимірювання складає 0.1 мВ в діапазоні 0-3.2 В. Похибка встановлення вихідного струму у всьому діапазоні не перевищує 0.1 %. Керування вимірювальним комплексом здійснюється за допомогою програмного забезпечення для ПЕОМ.

2. Із аналізу структурно-мофологічних характеристик, пошарової будови, елементного складу та електрохімічних властивостей анодів на основі Zn/ZnO/NiO, Zn/ZnO, ZnO/NiO, Zn/C і Ni/C можна зробити наступні висновки – після 50-ти циклів зарядки-розрядки найбільшу питому ємність (~240 мА·год/грам) і найменшу стабільність мають системи, що основані на використанні трьохшарових композитів Zn/ZnO/NiO.

3. При використанні в якості електродів систем на основі оксидів металів і переході від Zn/ZnO/NiO до ZnO/NiO і далі до Zn/ZnO по завершенні 50-ти циклів зарядки-розрядки відбувається постове зниження питомої ємності від 240 до 100 мА·год/грам з одночасним незначним підвищенням стабільності роботи анодів.

4. Аноди на основі пористих Zn/C і Ni/C мають приблизно однакову питому ємність (~190÷210 мА год/грам) за наявності підвищеної стабільності їх роботи, що визначає більш високою стійкістю ТГ до процесів інтеркаляції-деінтеркаляції літію.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. A.S. Kornyushchenko, Yu. O. Kosminska, S. T. Shevchenko, G. Wilde, V.I. Perekrestov. Structural, morphological and sensor properties of the fractal-percolation nanosystem ZnO/NiO // *Journal of Electronic Materials*. – 2021 – № 50 – P. 2268-2276.

2. A.S. Kornyushchenko, A.H. Jayatissa. Advanced nanocomposites as cathode materials for Li-ion batteries“ in Book “Applications of Nanocomposites”, USA:Taylor and Francis. – 2021. Прийнято до друку

3. Г.С. Корнющенко, Ю.О. Космінська, С.Т. Шевченко, В.В. Наталіч, В.І. Перекрестов. Етапи нуклеації квазірівноважних конденсатів іонно розпилених атомів Cr, Zn, Cu, Si, Ag та Al // *Журнал нано- та електронної фізики*. – 2021. – № 13(2) – P. 02034–1-6.

4. Г.С. Корнющенко, С.Т. Шевченко, В.В. Наталіч, В.І.Перекрестов. Отримання та структурно-морфологічні характеристики поруватих наносистем Zn/ZnO та Zn/ZnO/NiO // *Металофізика та новітні технології*. – 2021. – № 43(5) – P. 613–628.

5. Г.С. Корнющенко, С.Т. Шевченко, В.В. Наталіч, В.І.Перекрестов. Фізико-технологічні передумови формування та структурно-морфологічні характеристики композитів C/Zn і C/Ni з вуглецевою турбостратною складовою // *Наносистеми, наноматеріали, нанотехнології*. – 2021. – № 19(2). – С. 375–389.

6. V.I. Perekrestov, A.S. Kornyushchenko, Y.O. Kosminska, S.T. Shevchenko. Systems of complete self-organization of the nanotechnological process. Proceedings of the 11th International Conference «Nanomaterials: Applications and Properties». – Odessa: Sumy State University, 2021.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Synthesis of SnO<sub>2</sub> nanorods and hollow spheres and their electrochemical properties as anode materials for lithium ion batteries / Tan L [et al.] // Mater. Technol. – 2012. – Vol. 27, № 2. – p.191–195.
2. Nainville I., Lemarchand A., Badiali J.-P. Passivation of a lithium anode: A simulation model // Electrochim. Acta. - 1996. – p. 2855–2863.
3. Mahmood N., Tang T., Hou Y. Nanostructured Anode Materials for Lithium Ion Batteries: Progress, Challenge and Perspective // Adv. Energy Mater. – 2016. – Vol. 6, № 17. – P. 1600374.
4. Sn-doped TiO<sub>2</sub> nanotubes as superior anode materials for sodium ion batteries / Yan D [et al.]// Chem. Commun. – 2015. – Vol. 51, № 39. – p. 8261–8264.
5. Controllable synthesis of Cu-doped CoO hierarchical structure for high performance lithium-ion battery / Chen C [et al.] // J. Power Sources. – 2016. – Vol. 314. – p. 66–75.
6. Hao Q., Wang J., Xu C. Facile preparation of Mn<sub>3</sub>O<sub>4</sub> octahedra and their long-term cycle life as an anode material for Li-ion batteries // J. Mater. Chem. A. – 2014. – Vol. 2, № 1. – p. 87–93.
7. Hierarchically Nanostructured Transition Metal Oxides for Lithium-Ion Batteries / Zheng M. [et al.] // Adv. Sci. – 2018. – Vol. 5, № 3. – P. 1700592.
8. Green and Facile Fabrication of Hollow Porous MnO/C Microspheres from Microalgae for Lithium-Ion Batteries / Xia Y. [et al.] // ACS Nano. – 2013. Vol. 7, № 8. – p. 7083–7092.
9. Facile synthesis of MnO/C anode materials for lithium-ion batteries / Liu Y. [et al.] // Electrochim. Acta. – 2011. – Vol. 56, № 18. – p. 6448–6452.
10. Nitrogen-doped carbon-coated MnO nanoparticles anchored on interconnected graphene ribbons for high-performance lithium-ion batteries / Sheng L. [et al.] // J. Power Sources. – 2018. – Vol. 397. – p. 325–333.

11. Ultra-High Capacity Lithium-Ion Batteries with Hierarchical CoO Nanowire Clusters as Binder Free Electrodes / Cao K. [et al.] // *Adv. Funct. Mater.* – 2015. – Vol. 25, № 7. – p. 1082–1089.
12. Template-Free Synthesis of Hollow-Structured Co<sub>3</sub>O<sub>4</sub> Nanoparticles as High-Performance Anodes for Lithium-Ion Batteries / Wang D. [et al.] // *ACS Nano.* – 2015. – Vol. 9, № 2. – p. 1775–1781.
13. Recent progress of advanced anode materials of lithium-ion batteries / Cheng H. [et al.] // *J. Energy Chem.* – 2021. – Vol. 57. – p. 451–468.
14. The lithium-ion battery: State of the art and future perspectives / Zubi G. [et al.] // *Renew. Sustain. Energy Rev.* – 2018. – Vol. 89. – p. 292–308.
15. Electrostatic Assembly of Sandwich-like Ag-C@ZnO-C@Ag-C Hybrid Hollow Microspheres with Excellent High-Rate Lithium Storage Properties / Xie Q. [et al.] // *ACS Nano.* – 2016. – Vol. 10, № 1. – p. 1283–1291.
16. Hydrothermal Synthesis of Nickel Oxide Nanosheets for Lithium-Ion Batteries and Supercapacitors with Excellent Performance / Mondal A.K. [et al.] // *Chem. – An Asian J.* – 2013. – Vol. 8, № 11. – p. 2828–2832.
17. Co-doped NiO nanoflake arrays toward superior anode materials for lithium ion batteries / Mai Y.J. [et al.] // *J. Power Sources.* – 2011. – Vol. 196, № 15. – p. 6388–6393.
18. Hierarchically ordered porous nickel oxide array film with enhanced electrochemical properties for lithium ion batteries / Yuan Y.F. [et al.] // *Electrochem. commun.* – 2010. – Vol. 12, № 7. – p. 890–893.
19. Synthesis of Ordered Mesoporous NiO with Crystalline Walls and a Bimodal Pore Size Distribution / Jiao F. [et al.] // *J. Am. Chem. Soc.* – 2008. – Vol. 130, № 15. – p. 5262–5266.
20. Electrospun NiO nanofibers as high performance anode material for Li-ion batteries / Aravindan V. [et al.] // *J. Power Sources.* – 2013. – Vol. 227. – p. 284–290.



21. Electrospinning synthesis of 3D porous NiO nanorods as anode material for lithium-ion batteries / Wei Kong X. [et al.] // *Mater. Sci.* – 2016. – Vol. 34, № 2. – p. 227–232.
22. Gold coating for a high performance Li<sub>4</sub>Ti<sub>5</sub>O<sub>12</sub> nanorod aggregates anode in lithium-ion batteries / Wang W. [et al.] // *J. Power Sources.* – 2014. – Vol. 245. – p. 624–629.
23. Morphology-controlled synthesis of cage-bell Pd@CeO<sub>2</sub> structured nanoparticle aggregates as catalysts for the low-temperature oxidation of CO / Guo H. [et al.] // *J. Mater. Chem. A.* – 2013. – Vol. 1, № 25. – P. 7494.
24. Evaluation of ZnO nanorod arrays with dandelion-like morphology as negative electrodes for lithium-ion batteries / Wang H. [et al.] // *Electrochim. Acta.* – 2009. – Vol. 54, № 10. – p. 2851–2855.
25. Belliard F., Irvine J.T.S. Electrochemical performance of ball-milled ZnO–SnO<sub>2</sub> systems as anodes in lithium-ion battery // *J. Power Sources.* – 2001. – Vol. 97–98. – p. 219–222.
26. Electrochemical Performances of Ni-Coated ZnO as an Anode Material for Lithium-Ion Batteries / Zhang C.Q. [et al.] // *J. Electrochem. Soc.* – 2007. – Vol. 154, № 2. – P. A65.
27. Zhao D., Hao Q., Xu C. Nanoporous TiO<sub>2</sub>/Co<sub>3</sub>O<sub>4</sub> Composite as an Anode Material for Lithium-Ion Batteries // *Electrochim. Acta.* – 2016. – Vol. 211. – p. 83–91.
28. Porous ZnO nanosheets grown on copper substrates as anodes for lithium ion batteries / Huang X.H. [et al.] // *Electrochim. Acta.* – 2011. – Vol. 56, № 14. – p. 4960–4965.
29. In situ synthesis of SnO<sub>2</sub> nanosheet/graphene composite as anode materials for lithium-ion batteries / Liu H. [et al.] // *J. Mater. Sci. Mater. Electron.* – 2013. – Vol. 24, № 10. – p. 3640–3645.
30. Choi J.W., Aurbach D. Promise and reality of post-lithium-ion batteries with high energy densities // *Nat. Rev. Mater.* – 2016. – Vol. 1, № 4. – P. 16013.

31. Yolk-shell ZnO-C microspheres with enhanced electrochemical performance as anode material for lithium ion batteries / Xie Q. [et al.]// *Electrochim. Acta.* – 2014. – Vol. 125. – P. 659–665.
32. Amorphous ZnO Quantum Dot/Mesoporous Carbon Bubble Composites for a High-Performance Lithium-Ion Battery Anode / Tu Z. [et al.]// *ACS Appl. Mater. Interfaces.* – 2017. – Vol. 9, № 1. – p. 439–446.
33. Interconnected porous MnO nanoflakes for high-performance lithium ion battery anodes / Li X. [et al.] // *J. Mater. Chem.* – 2012. – Vol. 22, № 18. – p. 9189.
34. Synthesis and electrochemical performances of ZnO/MnO<sub>2</sub> sea urchin-like sleeve array as anode materials for lithium-ion batteries / Fang J. [et al.]// *Electrochim. Acta.* – 2013. – Vol. 112. – p. 364–370.
35. Combination-based nanomaterial designs in single and double dimensions for improved electrodes in lithium ion-batteries and faradaic supercapacitors / Pham T.N. [et al.]// *J. Energy Chem.* – 2019. – Vol. 38. – p. 119–146.
36. High-Performance Lithium-Ion Battery Anode by Direct Growth of Hierarchical ZnCo<sub>2</sub>O<sub>4</sub> Nanostructures on Current Collectors / Qu B. [et al.]// *ACS Appl. Mater. Interfaces.* – 2014. – Vol. 6, № 1. – p. 731–736.
37. A Cost-Effective Supercapacitor Material of Ultrahigh Specific Capacitances: Spinel Nickel Cobaltite Aerogels from an Epoxide-Driven Sol-Gel Process / Wei T.-Y. [et al.] // *Adv. Mater.* – 2010. – Vol. 22, № 3. – p. 347–351.
38. NiCo<sub>2</sub>O<sub>4</sub>/C Nanocomposite as a Highly Reversible Anode Material for Lithium-Ion Batteries / NuLi Y. [et al.] // *Electrochem. Solid-State Lett.* – 2008. – Vol. 11, № 5. – P. A64.
39. Mesoporous NiCo<sub>2</sub>O<sub>4</sub> Nanowire Arrays Grown on Carbon Textiles as Binder-Free Flexible Electrodes for Energy Storage / Shen L. [et al.]// *Adv. Funct. Mater.* – 2014. – Vol. 24, № 18. – p. 2630–2637.
40. A facile route to synthesize multiporous MnCo<sub>2</sub>O<sub>4</sub> and CoMn<sub>2</sub>O<sub>4</sub> spinel quasi-hollow spheres with improved lithium storage properties / Li J. [et al.] // *Nanoscale.* – 2013. – Vol. 5, № 5. – P. 2045.

41. Electrospun ZnO-SnO<sub>2</sub> composite nanofibers with enhanced electrochemical performance as lithium-ion anodes / Luo L. [et al.] // *Ceram. Int.* – 2016. – Vol. 42, № 9. – p. 10826–10832.
42. A High-Capacity and Long-Cycle-Life Lithium-Ion Battery Anode Architecture: Silver Nanoparticle-Decorated SnO<sub>2</sub>/NiO Nanotubes / Kim C. [et al.] // *ACS Nano.* – 2016. – Vol. 10, № 12. – p. 11317–11326.
43. Graphene nanoribbons wrapping double nanoshells of SnO<sub>2</sub>@TiO<sub>2</sub> for high lithium storage / Li X. [et al.] // *J. Power Sources.* – 2016. – Vol. 336. – p. 298–306.
44. Understanding Structure–Function Relationship in Hybrid Co<sub>3</sub>O<sub>4</sub>–Fe<sub>2</sub>O<sub>3</sub>/C Lithium-Ion Battery Electrodes / Sultana I. [et al.] // *ACS Appl. Mater. Interfaces.* – 2015. – Vol. 7, № 37. – p. 20736–20744.
45. Metal-organic frameworks derived yolk-shell ZnO/NiO microspheres as high-performance anode materials for lithium-ion batteries / Li J. [et al.] // *Chem. Eng. J.* – 2018. – Vol. 335. – p. 579–589.
46. Abu-Lebdeh, Y. *Nanotechnology for Lithium-ion Batteries* / Y. Abu-Lebdeh, I. Davidson // Springer, New York, 2013. – 282 p.
47. Nanostructured carbon for energy storage and conversion / S. L. Candelaria, Y. Shao, W. Zhou, X. Li, J. Xiao, J. G. Zhang, Y. Wang, J. Liu, J. Li, G. Cao // *Nano Energy.* – 2012. – V. 1. – p. 195-220.
48. Xin, S. Nanocarbon networks for advanced rechargeable lithium batteries / Xin S., Guo Y. G., Wan L.J. // *Acc. Chem. Res.* – 2012. – V. 45. – p. 1759-1769.
49. Su, D.S. Nanostructured carbon and carbon nanocomposites for electrochemical energy storage applications / D.S. Su, R. Schlogl // *ChemSusChem.* – 2010. – V.3. – p. 136-168.
50. Su, D.S. The use of natural materials in nanocarbon synthesis / D.S. Su // *ChemSusChem.* – 2009. – V. 2. – p. 1009-1020.
51. Scrosati, B. Lithium batteries: status, prospects and future / B. Scrosati, J. Garche // *J Power Sources.* – 2010. – V.195. – p.2419–2430

52. Tollefson, J. Car industry: charging up the future / J. Tollefson // *Nat News.* – 2008. – V. 456. – p.436-440
53. Geim, A.K. Graphene: status and prospects / A.K. Geim // *Science.* – 2009. – V.324. – p.1530–1534
54. Thackeray, M.M. Electrical energy storage for transportation—approaching the limits of, and going beyond, lithium-ion batteries / M.M. Thackeray, C. Wolverton, E.D. Isaacs // *Energy Environ Sci.* – 2015. – V. 5. – p. 7854–7863
55. Tarascon, J-M. Issues and challenges facing rechargeable lithium batteries / J-M. Tarascon, M. Armand // *Nature.* – 2001. – V.414. – p.359–367
56. Gogotsi, Y. Materials science: energy storage wrapped up / Gogotsi Y // *Nature.* – 2004. – V.509. – p. 568–570.
57. Sun, H. Multifunctional, ultra-flyweight, synergistically assembled carbon aerogels / H. Sun, Z. Xu, V. Gao // *Adv Mater.* – 2013. – V.25. – p. 2554-2560.
58. Baughman, R.H. Carbon nanotubes-the route toward applications / R. Baughman, A. Zakhido, W.A. De Heer // *Science.* – 2002. – V.297. –p.787-792
59. Largescale pattern growth of graphene films for stretchable transparent electrodes / K. S. Kim, Y. Zhao, H. Jang, S. Y. Lee, J. M. Kim, K.S. Kim, J.H. Ahn, P. Kim, J. Y. Choi, B. H. Hong, // *Nature.* – 2009. – V. 457. – p. 706-710.
60. Substrate-free gas-phase synthesis of graphene sheets / A. Dato, V. Radmilovic, Z. Lee, V. Phillips, M. Frenklach // *Nano Lett.* – 2008. – V. 8. – p. 2012-1016.
61. Synthesis of graphene-based nanosheets via chemical reduction of exfoliated graphite oxide / S. Stankovich, D. A. Dikin, R. D. Piner, K. A. Kohlhaas, A. Kleinhammes, Y. Jia, Y. Wu, S. T. Nguyen, R. S. Ruoff // *Carbon.* – 2007. – V. 45. – p. 1558-1565.
62. Subrahmanyam, K.S. Simple method of preparing graphene flakes by an arc-discharge method / K. S. Subrahmanyam, L. S. Panchakarla, A. Govindaraj, C. R. N. Rao // *J. Phys. Chem. C .* – 2009. – V.113. – p. 4257-4259.

63. Toward N-doped graphene via solvothermal synthesis / D. Deng, X. Pan, L. Yu, Y. Cui, Y. Jiang, J. Qi, W. X. Li, Q. Fu, X. Ma, Q. Xue [et al.] // Chem. Mater. – 2011. – V. 23. – p.1188-1193.
64. Freestanding graphene by thermal splitting of silicon carbide granules / D. Deng, X. Pan, H. Zhang, Q. Fu, D. Tan, X. Bao // Adv. Mater. – 2010. – V.22. – p. 2168-2171.
65. Kosminska, Yu. O. Perekrestov, V. I. Regularities of self-organization of technological conditions during plasma-arc synthesis of carbon nanotubes // Diam. Rel. Mater. – 2018. – V.85 – p. 37-48.

## ДОДАТОК А – ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МІКРОКОНТРОЛERA

```

#include "stm8s_conf.h"
#include "stm8s_flash.h"
#include "com.h"
#include "PWM.h"
#include <math.h>
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <string.h>
#include <float.h>

#define f_master 800000
#define baud_rate 9600
#define internal_ADC

#define intADC_cal 32700 // calibration data from metrix mtx 3293
#define extADC_cal 295 // calibration data from metrix mtx 3293

uint16_t ext_ADC_Zero_cal;
uint8_t u8_ZeroCal = 0;

uint16_t brr;
uint8_t u8_SPI_data=0;
uint8_t RX_SPI_CH1 [4], *pRX_SPI_CH1;
uint8_t u8_SPI;
volatile uint8_t i;
uint8_t u8_Charge = 0, u8_NumCycle = 0, u8_CurrentCycle = 0, u8_Measure = 0;
uint32_t u32_Capacity = 0, u32_Time = 0, u32_ADC, u32_ADC_Ext; // measured capacity
uint16_t u16_LowVoltage, u16_HighVoltage, u16_CurrentVoltage; //
uint16_t u16_Current; // уставка тока
uint8_t u8_RXnumber; //счетчик принятых байт
volatile uint16_t iPWM; // ШИМ задача тока
volatile uint16_t iADC_data, iADC_AIN; // данные АЦП канала напряжения
volatile uint8_t *ptrData;

uint8_t u8_ReadADC_Flag, u8_DataParsingFlag;
uint8_t u8_i, u8_Temp;
uint8_t u8_CounterTime; // счетчик времени

char szText[100]; //объявлена беззнаковая строковая 8ми битная переменная szText
char szRX_data[54]; //объявлена беззнаковая строковая 8ми битная переменная
char szTX_data[54]; //объявлена беззнаковая строковая 8ми битная переменная szTX_data за которой зарезервировано 54 знака
char szADC[10]; //объявлена беззнаковая строковая 8ми битная переменная szText за которой зарезервировано 20 знаков

void DelayTim2Sec(uint8_t u8_Count);
void CallNumber(char * cNumber, uint8_t CallFlag);
void ClearRXbuf();
void RelayON();
void RelayOFF();
void UartDebugString(uint8_t *cString);
void WaitADC_DRDY();
void ADCReadRegister(uint8_t amount, uint8_t reglength);
void ADCRegisterOperation(uint8_t adress, uint8_t operation, uint8_t reglength);
void ReadADC_data();
void IntADC_read();
void ExtADC_read();
void IntADC_volt_calc();
void ExtADC_volt_calc();
void OutResult();
void ExtADC_Zero_Calibration();

// Отправка байта
void uart_tx_byte(unsigned char data){
while(!((UART1 -> SR) & UART1_SR_TXE));
//while(((UART1 -> SR) & UART1_SR_TXE) != 0x80) {};
UART1->DR = data;

```

```

}

// Отправка массива данных
void uart_tx_data(unsigned char * data, unsigned char len){
    while(len--){
        uart_tx_byte(*data++);
    }
}

// Отправка байта
void SPI_tx_byte(unsigned char data){
    while(!(SPI->SR) & SPI_SR_TXE);
        //while(((UART1->SR) & UART1_SR_TXE) != 0x80) {};
    SPI->DR = data;
}

int main( void )
{
    /** Настройка портов **/

    GPIOA->DDR |= GPIO_PIN_3;
    GPIOA->CR1 |= GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3; // включаем режим Input with pull-up for PA1, PA2; push-pull for PA3 - SS
    GPIOA->CR2 |= GPIO_PIN_1|GPIO_PIN_2; // включаем для них режим External interrupt enabled
    EXTI->CR1 = 0b00000010; // Falling edge only for port A

    /** Pin for LED **/
    GPIOB->DDR |= GPIO_PIN_4|GPIO_PIN_5; //Настраиваем пин PB5 на выход
    GPIOB->CR1 |= GPIO_PIN_4|GPIO_PIN_5; //И включаем для них режимPush-pull
    GPIOB->ODR &= ~GPIO_PIN_5; //Зажигаем светодиод на PB5 (Red)
    GPIOB->ODR &= ~GPIO_PIN_4; //Зажигаем светодиод на PB5 (Green)

    /** Настраиваем порт C **/
    GPIOC->DDR |= GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6; /** Настраиваем пины PC на выход включаем для них режим with pull-
    up PC3 PWM TIM1_CH3 **/
    GPIOC->CR1 |= GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7;

    /** For UART TX & Relay **/
    GPIOD->DDR |= GPIO_PIN_4|GPIO_PIN_5; //Настраиваем пин PD4, PD5 на выход
    GPIOD->CR1 |= GPIO_PIN_4|GPIO_PIN_5; //И включаем для них режим Push-pull

    GPIOB->ODR |= GPIO_PIN_5; //OFF RED светодиод на PB5 (open Drain)
    GPIOB->ODR |= GPIO_PIN_4; //OFF GREEN светодиод на PB4 (open Drain)

    /** Настройка последовательного порта **/ //9600
    // Значение регистра BRR
    brr = f_master/ baud_rate;
    UART1->BRR2 = brr & 0x000F;
    UART1->BRR2 |= brr >> 12;
    UART1->BRR1 = (brr >> 4) & 0x00FF;
    UART1->CR2 |= UART1_CR2_REN|UART1_CR2_TEN|UART1_CR2_RIEN; /** enable reciever and transmitter Receiver interrupt enable**/
    /** Настройка шины SPI **/ //9600
    SPI->CR1 = 0b00000000;
    SPI->CR1 |= SPI_CR1_CPOL; //Enable SPI
    SPI->CR1 |= SPI_CR1_CPHA;
    SPI->CR2 = 0b00000000; /** 1: Software slave management enabled **/
    SPI->ICR = 0b01000000; /** RX buffer not empty interrupt enable **/
    SPI->CR1 |= SPI_CR1_MSTR;
    SPI->CR1 |= SPI_CR1_SPE; //Enable SPI

    /** Настройка АЦП на PD2 - AIN2 **/

    ADC1->CSR = 0x03; //AIN3 on PD2
    ADC1->CR1 = 0b01110010; //continuous conversion ON
    ADC1->TDRH = 0b01111100; //ADC Schmitt trigger disable
    ADC1->CR2 |= ADC1_CR2_ALIGN;
    ADC1->CR1 |= ADC1_CR1_ADON;
    ADC1->CR1 |= ADC1_CR1_ADON;

```

```

/** Инициализация тактирования */
// Тактирование МК от внутреннего высокочастотного кварца(HSI) 16 MHz
// Частота CPU и периферии 16 MHz / 2 = 8 MHz
CLK->CKDIVR &= (uint8_t)(~CLK_CKDIVR_HSIDIV);
CLK->CKDIVR |= (uint8_t)CLK_PRESCALER_HSIDIV2;

sprintf(szText,"CV Controller\r\n");
UartDebugString(szText);

/* Настройка таймера TIM1: */
#define TIM1_ARR ((uint16_t)(1 * 5001 - 1)) // до которго значения считает таймер for PWM 4970
/* Настройка делителя таймера */
TIM1->PSCRH = (uint8_t)(0x00); //Настройка делителя
TIM1->PSCRL = (uint8_t)(0x01); //Настройка делителя 1 - 3.86kHz, 4 - 1.5kHz 20 - 250Hz, 25 - 202.6Hz, 35 - 142.6
/* Настройка значения по которому будет выполняться сброс счетчика таймера */
TIM1->ARRH = (uint8_t)((TIM1_ARR>>8) & 0xFF);
TIM1->ARRL = (uint8_t)((TIM1_ARR & 0xFF));
TIM1->SR1 = (uint8_t)(~TIM1_FLAG_UPDATE); // очистка флага прерывания TIM1 UPDATE
TIM1->IER |= (uint8_t)TIM1_IT_UPDATE; //включение прерывания TIM1 UPDATE

/** Настройка таймера TIM2:
- TIM2CLK = 8 MHz, Делитель выберем равный 128, тогда счетчи будет считать с
частотой 8 MHz / 128 = 62500 Hz
Одна секунда пройдет когда счетчик насчитает TIM2_ARR: */

#define TIM2_ARR ((uint16_t)(1 * 62500 - 1))
TIM2->PSCR = (uint8_t)(TIM2_PRESCALER_128); //Настройка делителя переполнение счетчика 2 раза в секунду при 64

/** Настройка значения по которому будет выполняться сброс счетчика таймера */
TIM2->ARRH = (uint8_t)((TIM2_ARR>>8) & 0xFF);
TIM2->ARRL = (uint8_t)((TIM2_ARR & 0xFF));
TIM2->SR1 = (uint8_t)(~TIM2_FLAG_UPDATE); // очистка флага прерывания TIM2 UPDATE
TIM2->IER |= (uint8_t)TIM2_IT_UPDATE; //включение прерывания TIM2 UPDATE

/** PWM */
/** For channel 3 TIM1 on PC3 */
TIM1->BKR = 0b10001000; //p.213
TIM1->CR1 = 0b01000000; //p.186
TIM1->CCER2 = 0b00000111; //для канала 3 на PC3 p.206
TIM1->CCMR3 = 0b11111000; //для канала 3 на PC3 p.201

enableInterrupts(); //глобально разрешаем прерывания

TIM1->CR1 |= TIM1_CR1_CEN; //запуск счета таймера 1 для ШИМа на порту PC3
TIM2->CR1 |= TIM2_CR1_CEN; //запуск счета таймера 2 для отчета времени 2 раза в секунду

/** ADC AD7705 reset set to 1 */

GPIOC->ODR &= ~GPIO_PIN_4; //RST to LOW level (enable ADC)
GPIOB->ODR &= ~GPIO_PIN_5; //ON RED светодиод на PB5 (open Drain)
GPIOB->ODR |= GPIO_PIN_4; //OFF GREEN светодиод на PB5 (open Drain)

DelayTim2Sec(1);

GPIOC->ODR |= GPIO_PIN_4; //RST to HIGH level (enable ADC)
GPIOB->ODR &= ~GPIO_PIN_4; //ON GREEN светодиод на PB5 (open Drain)

DelayTim2Sec(1);

GPIOB->ODR |= GPIO_PIN_5; //OFF RED светодиод на PB5 (open Drain)

SPI_tx_byte(0x20); // CH1
SPI_tx_byte(0x0C);
SPI_tx_byte(0x10); // CH1
SPI_tx_byte(0x44); // gain 1 unipolar
uart_tx_byte(u8_SPI_data);

WaitADC_DRDY(); /** wait for /DRDY to go low */

```



```

iPWM = 0; // Set 0 mA current
PWMSet(iPWM);
RelayOFF(); // OFF Relay discharge
/** External ADC zero calibration **/
u32_ADC_Ext = 0;
while(1)
{
    if(strstr(szRX_data,"\r\n") != NULL)
    {
        if(strstr(szRX_data,"ZEROCAL") != NULL)
        {
            ExtADC_Zero_Calibration();

            sprintf(szText,"ADC_cal = %u\r\n", ext_ADC_Zero_cal);
            uart_tx_data(szText, strlen(szText));
            ClearRXbuf();
            sprintf(szText,"ZEROCAL\r\nOK\r\n");
            uart_tx_data(szText, strlen(szText));
        }

        if(strstr(szRX_data,"AT") != NULL)
        {
            ClearRXbuf();
            sprintf(szText,"\r\nAT\r\nOK\r\n");
            uart_tx_data(szText, strlen(szText));
        }

        if(strstr(szRX_data,"PWM=") != NULL)
        {
            iPWM = (szRX_data[4] - 48) * 1000;
            iPWM = iPWM + (szRX_data[5] - 48) * 100;
            iPWM = iPWM + (szRX_data[6] - 48) * 10;
            iPWM = iPWM + szRX_data[7] - 48;
            ClearRXbuf();
            PWMSet(iPWM);
            sprintf(szText,"PWM = %i\r\nOK\r\n",iPWM);
            uart_tx_data(szText, strlen(szText));
        }

        if(strstr(szRX_data,"DISCHARGE") != NULL)
        {
            ClearRXbuf();
            RelayOFF();
            sprintf(szText,"DISCHARGE OK\r\n");
            uart_tx_data(szText, strlen(szText));
        }

        if(strstr(szRX_data,"CHARGE") != NULL)
        {
            ClearRXbuf();
            RelayON();
            sprintf(szText,"CHARGE OK\r\n");
            uart_tx_data(szText, strlen(szText));
        }

        if(strstr(szRX_data,"MEASURE_START") != NULL)
        {
            u8_Measure = 1;
            u32_Time = 0;
            u32_Capacity = 0;
            u8_CurrentCycle = 0;
            u8_Charge = 0;
            ClearRXbuf();
            sprintf(szText,"\r\nOK\r\n");
            uart_tx_data(szText, strlen(szText));
        }
    }
}

```

```

if(strstr(szRX_data,"MEASURE_STOP") != NULL)
{
    u8_Measure = 0;
    iPWM = 0;
    u8_Charge = 0;
    RelayOFF();

    PWMSet(iPWM);
    ClearRXbuf();
    sprintf(szText,"\r\nOK\r\n");
    uart_tx_data(szText, strlen(szText));
}

if(strstr(szRX_data,"MEAS_ADC_INT_1") != NULL)
{
    u32_ADC = 0;
    IntADC_read();
    /** Calculate voltage **/
    IntADC_volt_calc();
    memset(szText,0,strlen(szText)); //обнулили строку szText
    sprintf(szText,"V=%05u\r\n",u16_CurrentVoltage);
    uart_tx_data(szText, strlen(szText));
    ClearRXbuf();
}

if(strstr(szRX_data,"MEAS_ADC_EXT_1") != NULL)
{
    u32_ADC_Ext = 0;
    /** Read data from ADC **/
    ExtADC_read();
    /** Calculate voltage **/
    ExtADC_volt_calc();
    memset(szText,0,strlen(szText)); //обнулили строку szText
    sprintf(szText,"V=%05u\r\n", u16_CurrentVoltage);
    uart_tx_data(szText, strlen(szText));
    ClearRXbuf();
}

if(strstr(szRX_data,"SETUP") != NULL) //SETUP XXXX XXXXXX XXXXXX XXX
{
    u16_Current = (szRX_data[6] - 48) * 1000;
    u16_Current = u16_Current + (szRX_data[7] - 48) * 100;
    u16_Current = u16_Current + (szRX_data[8] - 48) * 10;
    u16_Current = u16_Current + szRX_data[9] - 48;

    u16_LowVoltage = (szRX_data[11] - 48) * 10000;
    u16_LowVoltage = u16_LowVoltage + (szRX_data[12] - 48) * 1000;
    u16_LowVoltage = u16_LowVoltage + (szRX_data[13] - 48) * 100;
    u16_LowVoltage = u16_LowVoltage + (szRX_data[14] - 48) * 10;
    u16_LowVoltage = u16_LowVoltage + szRX_data[15] - 48;

    u16_HighVoltage = (szRX_data[17] - 48) * 10000;
    u16_HighVoltage = u16_HighVoltage + (szRX_data[18] - 48) * 1000;
    u16_HighVoltage = u16_HighVoltage + (szRX_data[19] - 48) * 100;
    u16_HighVoltage = u16_HighVoltage + (szRX_data[20] - 48) * 10;
    u16_HighVoltage = u16_HighVoltage + szRX_data[21] - 48;

    u8_NumCycle = (szRX_data[23] - 48) * 100;
    u8_NumCycle = u8_NumCycle + (szRX_data[24] - 48) * 10;
    u8_NumCycle = u8_NumCycle + szRX_data[25] - 48;

    sprintf(szText,"Cur= %04u\tLV= %05u\tHV= %05u\tCycles= %03u\r\n",u16_Current,u16_LowVoltage,u16_HighVoltage,u8_NumCycle);
    uart_tx_data(szText, strlen(szText));
    ClearRXbuf();
    sprintf(szText,"OK\r\n");
    uart_tx_data(szText, strlen(szText));
}
}

```

```

if(u8_ReadADC_Flag == 1)
{
    iPWM = u16_Current; // set current
    PWMSet(iPWM);      // set current PWM

    if(u8_CurrentCycle == 0) // разрядить до 0
    {
        u32_Time++;
        u32_ADC = 0;
        IntADC_read();
        /** Calculate capacity **/
        IntADC_volt_calc();
        u32_Capacity = u32_Capacity + (uint32_t)u16_Current;
        /**Out result every 1 second**/
        OutResult();

        if(((intADC_cal / 1023) * iADC_data) <= u16_LowVoltage) // end of discharge
        {
            u8_CurrentCycle++; // next cycle
            u8_Charge = 0;      // start charge
            u32_Time = 0;      // reset time
            u32_Capacity = 0; // reset capacity
            PWMSet(0);
            TIM2->CR1 &= ~TIM2_CR1_CEN; //останов счета таймера 2 для отсчета времени 2 раза в секунду
            while(!(ADC1->CSR & ADC1_CSR_EOC)); // wait end of ADC conversion
            iADC_data = ADC1->DRL;
            iADC_data = iADC_data + (ADC1->DRH << 8);
            for(brr = 0; brr < 10000; brr++)
            {
                nop();
            }
            brr = 0;
            iADC_data = 0;
            IntADC_read();
            iADC_data = iADC_AIN;
            /** Calculate capacity **/
            IntADC_volt_calc();
            u32_Time = 0;
            u32_Capacity = 0;
            /**Out result every 1 second**/
            OutResult();
            TIM2->CR1 |= TIM2_CR1_CEN; //запуск счета таймера 2 для отсчета времени 2 раза в секунду
            RelayON(); // start charge u8_Charge = 1;
        }
    }
}
else // other cycles charge/discharge
{
    u32_Time++;
    u32_ADC_Ext = 0;
    /** Read data from ADC **/
    if(u8_Charge == 1) // AD7705 for charge
    {
        PWMSet(iPWM);
        ExtADC_read();
        /** Calculate capacity **/
        ExtADC_volt_calc();
        u32_Capacity = u32_Capacity + (uint32_t)u16_Current;
        /**Out result every 1 second**/
        OutResult();

        if(u16_CurrentVoltage >= u16_HighVoltage) //u16_CurrentVoltage
        {
            TIM2->CR1 &= ~TIM2_CR1_CEN; //останов счета таймера 2 для отсчета времени 2 раза в секунду
            PWMSet(0); // set current to 0 mA
            RelayOFF(); // start discharge

            while(!(ADC1->CSR & ADC1_CSR_EOC)); // wait end of ADC conversion
        }
    }
}

```

```

iADC_data = ADC1->DRL;
iADC_data = iADC_data + (ADC1->DRH << 8);
for(brr = 0; brr < 10000; brr++)
{
    nop();
}
u32_ADC = 0;
brr = 0;
iADC_data = 0;
IntADC_read();
/** Calculate capacity **/
IntADC_volt_calc();
u32_Time = 0;
u32_Capacity = 0;
/**Out result every 1 second**/
OutResult();
TIM2->CR1 |= TIM2_CR1_CEN; //запуск счета таймера 2 для отсчета времени 2 раза в секунду
}
}
else // internal ADC for discharge
{
    PWMSet(iPWM);
    u32_ADC = 0;
    IntADC_read();
    /** Calculate capacity **/
    IntADC_volt_calc();
    u32_Capacity = u32_Capacity + (uint32_t)u16_Current;
    /**Out result every 1 second**/
    OutResult();

    if(((intADC_cal / 1023) * iADC_data) <= u16_LowVoltage) // end of discharge
    {
        u8_Charge = 1; // if voltage is lower than u16_LowVoltage
        u8_CurrentCycle++;
        u32_Time = 0;
        u32_Capacity = 0;
        PWMSet(0);
        RelayON(); // start charge
    }
}
}
u8_ReadADC_Flag = 0;
/**Out result every 1 second**/
if(u8_CurrentCycle > u8_NumCycle)
{
    iPWM = 0; // I = 0
    PWMSet(iPWM);
    u8_Measure = 0;
    u8_CurrentCycle = 0;
    u32_Time = 0;
    u32_Capacity = 0;
    RelayOFF(); // start discharge
    ClearRXbuf();
    sprintf(szText, "STOP\r\nOK\r\n");
    uart_tx_data(szText, strlen(szText));
}
}
}

/** Объявляем обработчик прерывания по вектору 10 SPI **/
INTERRUPT_HANDLER(SPI_IRQHandler, 10)
{
    if((SPI->SR & SPI_SR_RXNE) != 0) // SPI data register not empty
    {
        u8_SPI_data = SPI->DR;
    }
}

```

```

SPI->SR &= (~SPI_SR_RXNE); // очистка флага прерывания SPI_SR_RXNE
}

/** Объявляем обработчик прерывания по вектору 11 TIM1 */
INTERRUPT_HANDLER(TIM1_UPD_OVF_BRK_IRQHandler, 11)
{
    TIM1->SR1 = (uint8_t)(~TIM1_IT_UPDATE); // очистка флага прерывания TIM2 UPDATE
}

/** Объявляем обработчик прерывания по вектору 13 TIM2 */
INTERRUPT_HANDLER(TIM2_UPD_OVF_BRK_IRQHandler, 13) // T ~ 0.5s
{
    u8_CounterTime++;
    if(u8_ZeroCal < 10)
    {
        u8_ZeroCal++;
        if(u8_ZeroCal == 5)
        {
            sprintf(szRX_data,"ZEROCAL\r\n");
        }
    }
    if(u8_Measure == 1)
    {
        u8_ReadADC_Flag = 1;
    }
    TIM2->SR1 = (uint8_t)(~TIM2_IT_UPDATE); // очистка флага прерывания TIM2 UPDATE
}

INTERRUPT_HANDLER(UART1_RX_IRQHandler, 18)
{
    szRX_data[u8_RXnumber] = UART1->DR; // переменную из регистр DR познаково отправить в szRX_data
    u8_RXnumber++; //после каждого знака сдвигать строку на 1 знак

    if(u8_RXnumber == 54) //при достижении ячейки 54
    {
        memset(szRX_data,0,strlen(szRX_data)); //обнулили строку szRX_data
        u8_RXnumber = 0; //вернуться в нулевую ячейку
        u8_DataParsingFlag = 0;
    }
    UART1->SR &= (~UART1_CR2_RIEN); // очистка флага прерывания EOC
}

void DelayTim2Sec(uint8_t u8_Count)
{
    u8_Temp = u8_CounterTime;
    while((int8_t)(u8_CounterTime-u8_Temp) < u8_Count*2)
    {
    }
}

void ClearRXbuf()
{
    u8_RXnumber = 0;
    memset(szRX_data,0,strlen(szRX_data)); //обнулили строку szRX_data
}

void RelayON()
{
    GPIOD->ODR |= GPIO_PIN_4; // Reley ON
    u8_Charge = 1;
    GPIOB->ODR &= ~GPIO_PIN_4; //ON GREEN светодиод на PB4 (open Drain)
}

void RelayOFF()
{
    GPIOD->ODR &= ~GPIO_PIN_4; // Reley OFF
    u8_Charge = 0;
    GPIOB->ODR |= GPIO_PIN_4; //OFF GREEN светодиод на PB4 (open Drain)
}

```

```

}

void UartDebugString(uint8_t *cString)
{
    uart_tx_data(cString, strlen(cString));
}

void WaitADC_DRDY()
{
    /** wait for /DRDY to go low **/
    while(((GPIOA->IDR) & 0b00000100) == 0b00000100) // wait LOW on DRDY pin of AD7705
    {
    }
    delay_nop(10);
}

void IntADC_read()
{
    ADC1->CSR &= (~ADC1_CSR_EOC); // очистка флага прерывания EOC - end of conversion
    for(i = 0; i < 4; i++)
    {
        while(!(ADC1->CSR & ADC1_CSR_EOC)); // wait end of ADC conversion
        u32_ADC = u32_ADC + ADC1->DRL;
        u32_ADC = u32_ADC + (ADC1->DRH << 8);
        ADC1->CSR &= (~ADC1_CSR_EOC); // очистка флага прерывания EOC - end of conversion
    }
    u32_ADC = u32_ADC >> 2; //no oversampling
    iADC_AIN = (uint16_t)u32_ADC;
    iADC_data = iADC_AIN;
}

void ExtADC_read()
{
    memset(RX_SPI_CH1,0,strlen(RX_SPI_CH1));
    pRX_SPI_CH1 = (uint8_t*)&RX_SPI_CH1; //get string adress
    SPI_tx_byte(0x38); // ch2
    WaitADC_DRDY(); /** wait for /DRDY to go low **/
    SPI_tx_byte(0x00);
    nop(); /** for TEST **/
    for(i = 0; i < 2; i++)
    {
        SPI_tx_byte(0x00);
        *pRX_SPI_CH1 = (uint8_t)u8_SPI_data; // read SPI data register
        pRX_SPI_CH1++;
    }
    iADC_data = RX_SPI_CH1[0] << 8;
    iADC_data = iADC_data + RX_SPI_CH1[1];
}

void IntADC_volt_calc()
{
    {
        u32_ADC = (uint32_t)iADC_data;
        u32_ADC = u32_ADC * intADC_cal;
        u32_ADC = u32_ADC / 1023;
        u16_CurrentVoltage = (uint16_t)u32_ADC;
    }
}

void ExtADC_volt_calc()
{
    {
        u32_ADC_Ext = (uint32_t)(ext_ADC_Zero_cal - iADC_data);
        u32_ADC_Ext = u32_ADC_Ext * 250;
        u32_ADC_Ext = u32_ADC_Ext / extADC_cal;
        u16_CurrentVoltage = (uint16_t)u32_ADC_Ext;
    }

    void OutResult()
    {
        /** Out result **/
    }
}

```

```
sprintf(szText,"Ch=%01i          I=%04i          V=%05u          Time=%010li          Cy=%03i\n",u8_Charge,u16_Current,u16_CurrentVoltage,u32_Time,u8_CurrentCycle,u32_Capacity);
uart_tx_data(szText, strlen(szText));
}

void ExtADC_Zero_Calibration()
{
  iPWM = 0; // Set 0 mA current
  PWMSet(iPWM);
  RelayOFF(); // OFF Relay discharge

  /** External ADC zero calibration **/
  u32_ADC_Ext = 0;
  ExtADC_read();
  ext_ADC_Zero_cal = iADC_data;
}

void delay_nop(uint8_t time)
{
  for(i; i < time; i++)
  {
    nop();
  }
}
```

## Додаток 2 – Програмне забезпечення ПК

```

#include <stdio.h>
#include <conio.h>
#include <iomanip>
#include <iostream>
#include <locale.h>
#include <vector>
using namespace std;
#include <fstream>
#include <direct.h>
#include <time.h>
#include <unistd.h>
#include <io.h>
#include <windows.h>
#include <windowsx.h>
#ifdef __MINGW_H
#define _WIN32_IE 0x0400
#endif
#include <commctrl.h>
#include <Dbt.h> //для слезження
#include <Winuser.h>
#include <stdlib.h>
#include <winbase.h>
#include <cmath>
#include <assert.h>
#define _WIN32_IE 0x0400 /* min */ // для INITCOMMONCONTROLSEX
#include <commctrl.h>
#include "KWnd.h"
#include "commands.h"
#include "resource.h"
#ifdef __MINGW_H
#define _WIN32_IE 0x0400
#endif
#if (_WIN32_IE >= 0x0300)
BOOL WINAPI InitCommonControlsEx(LPINITCOMMONCONTROLSEX);
#endif
#define _WIN32_WINDOWS 0x0410
#define TIME_SLEEP 1000 //время приостановки текущего потока в мс
#define DEBUG_WINDOW
static std::vector<double> CurrentVector;
static std::vector<double>::iterator itCurrent;
/* Для синхронизации потоков*/
bool event = FALSE;
int COM_speed = 2;
int COM2_speed = 2;
static int mx, my;
// для работы с данными, принятыми и отправленными из устройства
static int iDU, iDU_sum, iIter_U, iIter = 0;
static float iU_min, iU_max;
static int iDataSet;
static DWORD dwBytesWritten; // тут будет количество собственно переданных байт
static char data[11] = ""; // строка для передачи
static char data_conv[11] = ""; // строка для передачи
static char sReceivedChar[32] = "";
static DWORD dwSize = sizeof(data); // размер этой строки
static unsigned char cConnectionON = 0, cPause = 0, cStart = 0;
static unsigned char cConnectionON_Com1 = 0, cZoom = 0, cScaleType = 0;
static unsigned char cVoltage = 20; //диапазон измерения напряжения: 2, 10, 20
static float fKcoef_x = 0; //диапазон измерения напряжения: 2, 10, 20
static char path[256] = "C:\\log.txt";
static char szTime_interval[256] = "";
static int iTime_period = TIMER_PERIOD2;
static unsigned int u16_LowVoltage, u16_HighVoltage, u16_CurrentVoltage;
static int iCurrent_set; //ток для установки, считывается с hWndEditCurrent
static int iCheckData = 1;
static int iRawData = 0;

```



```

static uint16_t u16_Cycles;
static float fLowVoltage, fHighVoltage, fCurrent;
static char cSetupOkFlag = 0; // флаг успешных уставок
static unsigned int uiMeasureVoltage;
static float fMeasureVoltage;
static char text_array[BUFSIZE];
/*****/
static long int ADC_old_DMM, ADC_new_DMM;
static long int DAC_old, DAC_new;
static unsigned int cRevers, cInv = 0;
static float iKa = 1E+6;
static DWORD EventMask;
int retcode;
FILE* fout;
std::ofstream out_file;
std::ofstream LOG_file;
enum ShapeSize { MAX, MIN};
typedef struct {
    int id_shape;
    BOOL fRed;
    BOOL fGreen;
    BOOL fBlue;
    int id_bright;
} ShapeData;
BOOL CALLBACK AboutDlgProc(HWND, UINT, WPARAM, LPARAM);
BOOL CALLBACK PathDlgProc(HWND, UINT, WPARAM, LPARAM);
BOOL CALLBACK TimeDlgProc(HWND, UINT, WPARAM, LPARAM);
BOOL CALLBACK PathDlgProcZoom(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
void AddMenus(HWND); //создание меню
void AddMenus1(HWND); //создание меню
void DrawDiagram(HWND,HDC); //рисование в окне
//Для работы с потоком
/*****/
HANDLE hEvent;
BOOL fSuccess;
HANDLE hThread;
DWORD bytesRead, bWritten;
DWORD dwParam, dwThreadId;
static BOOL bShow = TRUE;
struct LogData
{
    LogData(int _LogTime = 0, float _LogADC = 0): LogTime(_LogTime), LogADC(_LogADC) {}
    int LogTime;
    float LogADC;
};
// Поток для приема данных из COM - порта
/*****/
DWORD WINAPI ThreadProc(LPVOID)
{
    //;
    COMSTAT comstat;
    int counter_memory = 0;
    DWORD btr, temp;
    char szLogFile[] = "";
    time_t tmpTime = time(NULL);
    tm * dateTime = localtime(&tmpTime);
    sprintf(szLogFile, "C:\\Log_%d_%.2d_%.2d_%.2d_%.2d.log", dateTime->tm_year+1900, dateTime->tm_mon+1, dateTime->tm_mday,
        dateTime->tm_hour, dateTime->tm_min, dateTime->tm_sec);
    out_file.open(szLogFile, std::ios::app);
#ifdef DEBUG_WINDOW
    int wait_bytes = 0;
    while (1)
    {
        WaitCommEvent(hSerialCom2, &EventMask, &OL); // поток переходит в состояние блокировки
        retcode = WaitForSingleObject(OL.hEvent, INFINITE);
    }
#endif
}

```

```

switch (retcode)
{
    case WAIT_OBJECT_0:
    {
        if(GetOverlappedResult(hSerialCom2, &OL, &temp, true)
if( (EventMask & EV_RXCHAR) != 0 )
{
    ClearCommError(hSerialCom2, &temp, &comstat);
    btr = comstat.cbInQue;
    if (btr > 0)
    {
        ReadFile(hSerialCom2, bufrd, btr, &temp, &OL);
        counter+=btr;
        memcpy(&bufSort[counter-btr], bufrd, sizeof(bufrd));
        if (counter > 4)
        {
            if(strstr(bufSort,"OK\r\n"))
            {
                counter = 0;
                if(strstr(bufSort,"AT"))
                {
                    sprintf(szText,"AT OK");
                    LOG(szText);
                }
                if(strstr(bufSort,"Cycles"))
                {
                    sprintf(szText,"%s\n%s","Параметры успешно записаны",bufSort);
                    LOG(szText);
                }
                if(strstr(bufSort,"PWM ="))
                {
                    sprintf(szText,"%s\n%s","Ток установлен",bufSort);
                    LOG(szText);
                }
                memset(bufSort, 0, BUFSIZE); // обнуляем обработанный пакет данных
            }
            if(strstr(bufSort,"Cap="))
            {
                if(strstr(bufSort,"\r\n")) // end of string
                {
                    counter = 0;
                    time_t tmpTime = time(NULL);
                    tm * dateTime = localtime(&tmpTime);
                    out_file<<setw(4)<<dateTime->tm_year+1900<<":"<<setw(2)<<setfill('0')<<dateTime->tm_mon+1<<":"<<setw(2)<<setfill('0')<<dateTime->tm_mday<<"t"
                    <<setw(2)<<setfill('0')<<dateTime->tm_hour<<":"<<setw(2)<<setfill('0')<<dateTime->tm_min<<":"<<setw(2)<<setfill('0')<<dateTime->tm_sec<<"t";
                    out_file<<bufSort;
                    out_file.flush(); //Записать файл немедленно
                    /** Определить положение в пакете заряд-разряд, напряжения, времени, емкости **/
                    pdest = strstr(bufSort,"V=");
                    result = (int)(pdest - bufSort + 1);
                    uiMeasureVoltage = bufSort[result]*10000 + bufSort[result+1]*1000 + bufSort[result+2]*1000 + bufSort[result+3]*100 +
bufSort[result+4]*10 + bufSort[result+5];
                    memset(bufSort, 0, BUFSIZE); // обнуляем обработанный пакет данных
                }
            }
        }
    }
}
}
}
} break;

case WAIT_TIMEOUT:
    break;
case WAIT_FAILED:
    break;
}
}

```

```

}
/*****/
HINSTANCE g_hinst;
COLORREF g_color;
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)
{
    HACCEL hAccel;
    setlocale(LC_ALL, "Russian");
    MSG msg;
    KWnd mainWnd("CV Controller control application (alex-sdu@yandex.ru)",
hInstance,nCmdShow,WndProc,MAKEINTRESOURCE(Main),
50,100,440,295);//Создание главного окна с заданными размерами
    g_hinst = hInstance;
    hAccel = LoadAccelerators(g_hinst, MAKEINTRESOURCE(IDA_ACCEL));
    while (GetMessage(&msg, NULL, 0, 0))
    {
        if(!TranslateAccelerator(mainWnd.GetHWnd(), hAccel, &msg))
        {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }
    return msg.wParam;
}
// Процедура основного окна
LRESULT CALLBACK WndProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    static HDC hDC;
    static HDC hMemDcFrame;
    RECT Rect;
    RECT rVCC;
    RECT rect;
    HDC hdca;
    static BOOL bTracking = FALSE;
    PAINTSTRUCT ps;
    static LPCSTR portname = szPortName; // имя порта
    static DWORD accessdirection = GENERIC_READ | GENERIC_WRITE; // режим работы
    static int cTxtLen;
    static PSTR pszMem;
    static LPCSTR portnameCom2 = szPortNameCom2; // имя порта
    static DWORD accessdirectionCom2 = GENERIC_READ | GENERIC_WRITE; // режим работы
    static int cTxtLenCom2;
    static PSTR pszMemCom2;
    static DWORD iSize;
/*****/
    HWND hFlag1;
    HWND hFlag2;
    HWND hFlag3;
    static HWND hStartStop;
    static HWND hGroupBorder_Com1;
    static HWND hGroupBorder_Com2;
    static HWND hGroupBorder_Scale;
    static HWND hGroupBorder_Voltage;
    static HWND hGroupBorder_CVC;
    static HWND hCleanVCC;
    HBRUSH hBrush, holdBrush;
    HPEN hPen, holdPen;
/*****/
    static HWND hWndCombo, hWndStatic, hWndComboSpeed, hWndStaticSpeed, hWndMsg, hWndConnect, hWndDisconnect, hWndZoom;
    HWND hWndStatic_U_min, hWndStatic_U_max;
    static HWND hWndStatic_dU;
    const TCHAR * items[] = { TEXT("COM1"), TEXT("COM2"),
        TEXT("COM3"), TEXT("COM4") }; //порт
    const TCHAR * baud[] = { TEXT("4800"), TEXT("7200"),
        TEXT("9600"), TEXT("14400"), TEXT("19200") }; //скорость

    LRESULT sel_port = 0;
    LRESULT sel_speed = 0;
    LPTSTR sel_comport;

```

```

static HMENU hMenu;
static ShapeData shapeData;
/*****/
//высплывающее меню
    HMENU hMenu_popup;
    POINT point_popup;
HBITMAP hbmpPie; // handle to pie chart bitmap
HBITMAP hbmpLine; // handle to line chart bitmap
HBITMAP hbmpBar; // handle to bar chart bitmap
    HMENU hmenuMain; // handle to main menu
    HMENU hmenuChart; // handle to Chart menu
    HMENU hSubmenu;
/*****/
//Строка ввода
static HWND hWndEdit0; static HWND hWndEdit1;
static HWND hWndEdit2, hWndEditCurrent; //поля редактирования для порта 1
static HWND hWndEditCom2[4]; //поля редактирования для установки напряжения разряда, напряжения заряда, количества циклов
static HWND hWndEditCom2Line, hWndSpinControl[2];
static HWND hWndButton;
static int len;
static TCHAR text[30];
/*****/
int userReply;
switch (uMsg)
{
    char key_port[5], PortNumber[5];
    char speed_port[5], PortSpeed[5];
    char szLine [11];
    int key, new_key, key_Com2;
    TCHAR * pVarBlock, * pVarBeg, * pVarEnd, * pVarName ;
    int iLength ;
    int i;
    TCHAR szBuffer[MAX_PATH + 1] ;
    case WM_INITDIALOG:

/*
* TODO: Add code to initialize the dialog.
*/
        return TRUE;
    case WM_COMMAND:
//Обработка пунктов меню
/*****/
        switch(LOWORD(wParam))
        {
        case IDM_FILE_CREATE:
            {
            // Изменение пути сохранения файла
                DialogBox(g_hinst,MAKEINTRESOURCE(IDD_FILE_PATH),hWnd,PathDlgProc);
            }
            break;
        case IDM_TIME_EDIT:
            {
            // Изменение пути сохранения файла
                DialogBox(g_hinst,MAKEINTRESOURCE(IDD_TIME_EDIT),hWnd,TimeDlgProc);
            }
            break;
        case IDM_CHARGE:
            {
                if (cConnectionON != 0)
            {
                if (!cStart) //если кнопка СТАРТ не была нажата
                {
                    // Разряд
                    wsprintf(text_array,"CHARGE\r\n");
                    DWORD dwBytesRead = 0;
                    WriteFile(hSerialCom2, text_array, strlen(text_array), &dwBytesRead, &OL);
                }
            }
            else
            {

```

```

        CheckDlgButton(hWnd, ID_WRITE_COM_2, BST_UNCHECKED);
        wsprintf(szText, "Устройство в режиме измерения\nВыполнение команды невозможно");
        MessageBox(hWnd, szText, "Error", MB_OK | MB_APPLMODAL | MB_ICONERROR );
    }
}
else
{
    MessageBox(NULL, "Нет соединения. Подключите коммуникационный порт.", "Connection error",
MB_OK|MB_ICONEXCLAMATION);
}
} break;

        case IDM_DISCHARGE:
    {
        if (cConnectionON != 0)
        {
            if (!cStart)
            {
                wsprintf(text_array, "DISCHARGE\r\n");
                DWORD dwBytesRead = 0;
                WriteFile(hSerialCom2, text_array, strlen(text_array), &dwBytesRead, &OL);
            }
            else
            {
                CheckDlgButton(hWnd, ID_WRITE_COM_2, BST_UNCHECKED);
                wsprintf(szText, "Устройство в режиме измерения\nВыполнение команды невозможно");
                MessageBox(hWnd, szText, "Error", MB_OK | MB_APPLMODAL | MB_ICONERROR );
            }
        }
        else
        {
            MessageBox(NULL, "Нет соединения. Подключите коммуникационный порт.", "Connection error",
MB_OK|MB_ICONEXCLAMATION);
        }
    } break;

        case IDM_FILE_EXIT:
    {
        SendMessage(hWnd, WM_CLOSE, 0, 0);
    } break;
        case IDM_AT_COMMAND:
    {
        // Щелчек правой опросить прибор
        if (cConnectionON == 0)
        {
            CheckDlgButton(hWnd, ID_WRITE_COM_2, BST_UNCHECKED);
            wsprintf(szText, "Устройство не подключено\nВыполните проверку оборудования");
            MessageBox(hWnd, szText, "Connection error", MB_OK | MB_APPLMODAL |
MB_ICONERROR );
        }
        break;
    }
    else
    {
        if (!cStart) //если кнопка СТАРТ не была нажата
        {
            memset(text_array, 0, sizeof(text_array)); // обнулить буфер для приема данных
            wsprintf(text_array, "AT\r\n");
            DWORD dwBytesRead = 0;
            WriteFile(hSerialCom2, text_array, strlen(text_array), &dwBytesRead, &OL);
        }
        else
        {
            CheckDlgButton(hWnd, ID_WRITE_COM_2, BST_UNCHECKED);
            wsprintf(szText, "Устройство в режиме измерения\nВыполнение команды невозможно");
            MessageBox(hWnd, szText, "Error", MB_OK | MB_APPLMODAL | MB_ICONERROR );
        }
    }
}
} break;
case IDM_SETUP_COMMAND:
    {

```

```

long int liSet;

        // Щелчек правой кнопкой мыши записать уставки в прибор
        if (cConnectionON == 0)

{
    CheckDlgButton(hWnd, ID_WRITE_COM_2, BST_UNCHECKED);
    wsprintf(szText, "Устройство не подключено\nВыполните проверку оборудования");
    MessageBox(hWnd, szText, "Connection error", MB_OK | MB_APPLMODAL | MB_ICONERROR );
    break;
}
else
{
    if (!cStart)          //если кнопка СТАРТ не была нажата
    {
        // Считать значения из полей редактирования
        /** Read current **/
        GetWindowText(hWndEditCurrent, szText, 16);
        fCurrent = atof(szText);          // преобразовать строку в float
        /** Read LowVoltage float **/
        GetWindowText(hWndEditCom2[0], szText, 16); //считать напряжение разряда из формы
        fLowVoltage = atof(szText);      // преобразовать строку в float
        /** Read Cycles **/
        GetWindowText(hWndEditCom2[1], szText, 16); //считать количество циклов из формы
        liSet = strtol (szText, NULL, 0); //преобразовать строку в long int
        /** Read HighVoltage **/
        GetWindowText(hWndEditCom2[2], szText, 16); //считать напряжение заряда из формы
        fHighVoltage = atof(szText);     // преобразовать строку в float
        cSetupOkFlag = 1;
        if((fCurrent < 0)||(fCurrent > 500))
        {
            wsprintf(szText, "Диапазон тока 0..50 мА, шаг 0.01 мА");
            MessageBox(hWnd, szText, "Setup error", MB_OK | MB_APPLMODAL | MB_ICONERROR );
            cSetupOkFlag = 0;
        }
        if(fLowVoltage < 0)
        {
            wsprintf(szText, "Напряжение разряда не может быть меньше 0 В");
            MessageBox(hWnd, szText, "Setup error", MB_OK | MB_APPLMODAL | MB_ICONERROR );
            cSetupOkFlag = 0;
        }
        if(fLowVoltage > 3.3)
        {
            wsprintf(szText, "Напряжение разряда не может быть больше 3.3 В");
            MessageBox(hWnd, szText, "Setup error", MB_OK | MB_APPLMODAL | MB_ICONERROR );
            cSetupOkFlag = 0;
        }
        if(fHighVoltage < 0)
        {
            wsprintf(szText, "Напряжение заряда отрицательно, проверьте параметры");
            MessageBox(hWnd, szText, "Setup error", MB_OK | MB_APPLMODAL | MB_ICONERROR );
            cSetupOkFlag = 0;
        }
        if(fHighVoltage > 3.3)
        {
            wsprintf(szText, "Напряжение заряда не может быть больше 3.3 В, проверьте параметры");
            MessageBox(hWnd, szText, "Setup error", MB_OK | MB_APPLMODAL | MB_ICONERROR );
            cSetupOkFlag = 0;
        }
        if(u16_Cycles < 0)
        {
            wsprintf(szText, "Количество циклов 0..255");
            MessageBox(hWnd, szText, "Setup error", MB_OK | MB_APPLMODAL | MB_ICONERROR );
            cSetupOkFlag = 0;
        }
        if(fLowVoltage > fHighVoltage)
        {
            wsprintf(szText, "Напряжение заряда должно быть больше напряжения разряда, проверьте параметры");
            MessageBox(hWnd, szText, "Setup error", MB_OK | MB_APPLMODAL | MB_ICONERROR );
            cSetupOkFlag = 0;
        }
    }
}

```

```

}
if(cSetupOkFlag == 1)
{
    fCurrent = fCurrent * 10;
    iCurrent_set = (int)fCurrent;        // преобразовать строку в int
    /* fLowVoltage = fLowVoltage * 1023 / 3.3; */
    fLowVoltage = fLowVoltage * 10000; // data for low voltage
    u16_LowVoltage = (unsigned int)(fLowVoltage); //преобразовать float в int
    u16_Cycles = (int)liSet;           //преобразовать long int в int
    fHighVoltage = 38364 - fHighVoltage * 10000 * 1.1625; // data for high voltage
    u16_HighVoltage = (unsigned int)(fHighVoltage); //преобразовать float в int
    wsprintf(text_array, "SETUP %04i %05u %03i\r\n", iCurrent_set, u16_LowVoltage, u16_HighVoltage, u16_Cycles);
    DWORD dwBytesRead = 0;
    WriteFile(hSerialCom2, text_array, strlen(text_array), &dwBytesRead, &OL);
}
else
{
    wsprintf(szText, "Проверьте значения уставок");
    MessageBox(hWnd, szText, "Error", MB_OK | MB_APPLMODAL | MB_ICONERROR );
}
}
else
{
    CheckDlgButton(hWnd, ID_WRITE_COM_2, BST_UNCHECKED);
    wsprintf(szText, "Устройство в режиме измерения\nВыполнение команды невозможно");
    MessageBox(hWnd, szText, "Error", MB_OK | MB_APPLMODAL | MB_ICONERROR );
}
}
} break;
case IDM_SET_CURRENT:
{
    if (cConnectionON == 0)
    {
        CheckDlgButton(hWnd, ID_WRITE_COM_2, BST_UNCHECKED);
        sprintf(szText, "Старт");
        SendMessage(hStartStop, WM_SETTEXT, 0, LPARAM (szText));
        wsprintf(szText, "Устройство не подключено\nВыполните проверку оборудования");
        MessageBox(hWnd, szText, "Connection error", MB_OK | MB_APPLMODAL |
MB_ICONERROR );
    }
    else
    {
        GetWindowText(hWndEditCurrent, szText, 16);
        fCurrent = atof(szText); // преобразовать строку в float
        if((fCurrent < 0)|| (fCurrent > 500))
        {
            wsprintf(szText, "Диапазон тока 0...500 мА, шаг 0.1 мА");
            MessageBox(hWnd, szText, "Set current error", MB_OK | MB_APPLMODAL |
MB_ICONERROR );
        }
        else
        {
            if (lcStart) //если кнопка СТАРТ не была нажата
            {
                fCurrent = fCurrent * 10;
                iCurrent_set = (int)fCurrent; // преобразовать строку в int
                //Установка тока
                memset(text_array, 0, sizeof(text_array)); // обнулить буфер для приема данных
                wsprintf(text_array, "PWM=%04i\r\n", iCurrent_set);
                DWORD dwBytesRead = 0;
                WriteFile(hSerialCom2, text_array, strlen(text_array), &dwBytesRead, &OL);
            }
            else
            {
                CheckDlgButton(hWnd, ID_WRITE_COM_2, BST_UNCHECKED);
                wsprintf(szText, "Устройство в режиме измерения\nВыполнение команды невозможно");
                MessageBox(hWnd, szText, "Error", MB_OK | MB_APPLMODAL | MB_ICONERROR );
            }
        }
    }
}
}
}

```

```

    }
        } break;
    case IDM_SHOW_SHAPE:
        bShow = TRUE;

    CheckMenuRadioItem(GetSubMenu(hMenu,1),IDM_SHOW_SHAPE,IDM_HIDE_SHAPE,LOWORD(wParam),MF_BYCOMMAND);
    DrawMenuBar(hWnd);

    break;
    case IDM_HIDE_SHAPE:
        bShow = FALSE;

    CheckMenuRadioItem(GetSubMenu(hMenu,1),IDM_SHOW_SHAPE,IDM_HIDE_SHAPE,LOWORD(wParam),MF_BYCOMMAND);
    DrawMenuBar(hWnd);

    break;
    case IDM_ABOUT:
        //вызов диалога
        DialogBox(g_hinst,MAKEINTRESOURCE(IDD_ABOUT),hWnd,AboutDlgProc);

    break;
}

if(wParam==ID_WRITE_COM_2) // Кнопка СТАРТ
{
    if (!cConnectionON)
    {
        CheckDlgButton(hWnd, ID_WRITE_COM_2, BST_UNCHECKED);
        wsprintf(szText, "Устройство не подключено\nВыполните проверку оборудования");
        MessageBox(hWnd, szText, "Connection error", MB_OK | MB_APPLMODAL | MB_ICONERROR );
        break;
    }
    else
    {
        if (!cStart) //если кнопка СТАРТ не была нажата
        {
            memset(sReceivedChar, 0, sizeof(sReceivedChar)); // обнулить буфер для приема
            // данных

            memset(szText, 0, sizeof(szText)); // обнулить буфер для приема данных
            iDU_sum = 0;
            SetWindowText((HWND)lParam, "Стоп"); //изменить значение кнопки
            memset(text_array, 0, sizeof(text_array)); // обнулить буфер для приема данных
            wsprintf(text_array, "MEASURE_START\r\n");
            DWORD dwBytesRead = 0;
            WriteFile(hSerialCom2, text_array, strlen(text_array), &dwBytesRead, &OL);
            cStart = 1; //установить флаг начала измерения
            Sleep(TIME_SLEEP); //подождать 500мс
            SetTimer(hWnd, TIMER_ID2, iTime_period, NULL); //запуск таймера для обновления
            // напряжения цикла в форме

        }
        else
        {
            SetWindowText((HWND)lParam, "Старт");
            memset(text_array, 0, sizeof(text_array)); // обнулить буфер для приема данных
            wsprintf(text_array, "MEASURE_STOP\r\n");
            DWORD dwBytesRead = 0;
            WriteFile(hSerialCom2, text_array, strlen(text_array), &dwBytesRead, &OL);
            cStart = 0;
            cPause = 0;
            CheckDlgButton(hWnd, ID_READ_COM_2, BST_UNCHECKED);
            KillTimer (hWnd, TIMER_ID2); //остановка таймера для обновления показаний

        }
    }
}

if(wParam==ID_READ_COM_2) //кнопка пауза
{
    // Pause кнопка
    if (cStart == 0)
    {
        CheckDlgButton(hWnd, ID_READ_COM_2, BST_UNCHECKED);
    }
}

```



```

else
{
    if (cPause == 0)
    {
        KillTimer (hWnd, TIMER_ID2); //остановка таймера для обновления показаний
        cPause = 1;
    }
    else
    {
        if (cStart == 1)
        {
            SetTimer (hWnd, TIMER_ID2, iTime_period, NULL); //запуск таймера для
записи показаний в устройство
            SetTimer (hWnd, TIMER_ID, TIMER_PERIOD, NULL); //запуск таймера для
записи показаний в устройство
            cPause = 0;
        }
        else
        {
            KillTimer (hWnd, TIMER_ID2); //остановка таймера для обновления показаний
            cPause = 1;
        }
    }
}

if (wParam == ID_CLEAN)
{
    //LOG("Test");
    curves.clear();
    curve.clear();
    curves2.clear();
    curve2.clear();
    InvalidateRect(hWnd, 0, true);
}
if (wParam == ID_SETKY) //установка коэффициента масштабирования по оси Y (Ток)
{
    GetWindowText(hWndEditCom2[3], szText, 1024); // считать данны из диалогового окна для устакновки минимального
напряжения
    if (!cStart)
    {
    }
    else
    {
        if ( abs (ADC_new_DMM / iKa) * atof(szText) > 190.)
        {
            sprintf(szText, "%f", abs (ADC_new_DMM / iKa));
            iKa = 190. / abs(ADC_new_DMM) / 10.;
            sprintf(szText, "%.3e", iKa);
            SendMessage(hWndEditCom2[3], WM_SETTEXT, 0, LPARAM(szText));
        }
        else
        {
            iKa = atof(szText);
        }
    }
}
if (wParam == ID_CHECK_DATA)
{
    if (!iCheckData) iCheckData = 1;
    else iCheckData = 0;
}
if (wParam == ID_RAW_DATA)
{
    if (!iRawData) iRawData = 1;
    else iRawData = 0;
}
}
/*****/

```

```

if(wParam==ID_CONNECT_COM_2)
{
    if (cConnectionON) //если подключено, то отключить
    {
        SetWindowText((HWND)lParam, "Connect");
        cConnectionON = 0;
        cStart = 0;
        cPause = 0;
        CheckDlgButton(hWnd, ID_WRITE_COM_2, BST_UNCHECKED); // отжать кнопку
        CheckDlgButton(hWnd, ID_READ_COM_2, BST_UNCHECKED); // отжать кнопку
        SetWindowText(hWndMsg, "Disconnected"); //вывод строки
        memset(text_array, 0, sizeof(text_array)); // обнулить буфер для приема данных
        wsprintf(text_array, "MEASURE_STOP\r\n");
        DWORD dwBytesRead = 0;
        WriteFile(hSerialCom2, text_array, strlen(text_array), &dwBytesRead, &OL);
        TerminateThread(hThread, 0); // остановить поток
        CloseHandle(OL.hEvent);
        closeSerialPort(hSerialCom2);
        KillTimer (hWnd, TIMER_ID2); //остановка таймера для обновления показаний
        //KillTimer (hWnd, TIMER_ID2); //остановка таймера для обновления показаний
        sprintf(szText, "Старт");
        SendMessage(hStartStop, WM_SETTEXT, 0, LPARAM (szText));
    }
    else
    {
        SetWindowText((HWND)lParam, "Disconnect"); //изменить надпись
        cConnectionON = 1; //установить

        TerminateThread(hThread, 0); //** остановить поток **/
        CloseHandle(OL.hEvent);
        closeSerialPort(hSerialCom2); //закрывать порт
        // Открыть порт
        hSerialCom2 = CreateFile(portnameCom2,
                                accessdirection,
                                0,
                                NULL, //0,
                                OPEN_EXISTING,
                                FILE_FLAG_OVERLAPPED, //для асинхронной работы с портом
                                //0, //для синхронной работы с портом
                                NULL); //0);

        if (hSerialCom2 == INVALID_HANDLE_VALUE)
        {
            cConnectionON = 0;
            CheckDlgButton(hWnd, ID_CONNECT_COM_2, BST_UNCHECKED);
            SetWindowText((HWND)lParam, "Connect");
            MessageBox(NULL, "Невозможно открыть порт. Проверьте его наличие в системе.",
"Connection error", MB_OK|MB_ICONEXCLAMATION);
        }
        DCB dcbSerialParamsCom2 = {0};
        dcbSerialParamsCom2.DCBlength = sizeof(dcbSerialParamsCom2);
        if (!GetCommState(hSerialCom2, &dcbSerialParamsCom2))
        {
            MessageBox(NULL, "Не удалось получить состояния порта", "Connection error",
            MB_OK|MB_ICONEXCLAMATION);
        }
        switch (COM2_speed)
        {
            case 0: dcbSerialParamsCom2.BaudRate = 4800; break;
            case 1: dcbSerialParamsCom2.BaudRate = 7200; break;
            case 2: dcbSerialParamsCom2.BaudRate = 9600; break;
            case 3: dcbSerialParamsCom2.BaudRate = 14400; break;
            case 4: dcbSerialParamsCom2.BaudRate = 19200; break;
            default: dcbSerialParamsCom2.BaudRate = 19200; break;
        }

        dcbSerialParamsCom2.ByteSize = 8;
        dcbSerialParamsCom2.StopBits = ONESTOPBIT;
        dcbSerialParamsCom2.Parity = NOPARITY;
        dcbSerialParamsCom2.fNull = FALSE; //разрешить приём нулевых байтов
    }
}

```

```

if(!SetCommState(hSerialCom2, &dcbSerialParamsCom2))
{
//analyse error
    MessageBox(NULL, "Не удалось установить состояния порта", "Connection
error", MB_OK|MB_ICONEXCLAMATION);
}
COMMTIMEOUTS timeoutsCom2={0};
timeoutsCom2.ReadIntervalTimeout=0xFFFFFFFF//50;
timeoutsCom2.ReadTotalTimeoutConstant=0//50;
timeoutsCom2.ReadTotalTimeoutMultiplier=0//10;
timeoutsCom2.WriteTotalTimeoutConstant=0//50;
timeoutsCom2.WriteTotalTimeoutMultiplier=1500//10;
if(!SetCommTimeouts(hSerialCom2, &timeoutsCom2))
{
//handle error
    MessageBox(NULL, "Не удалось установить тайминги порта", "Connection
error", MB_OK|MB_ICONEXCLAMATION);
    break;
}
SetWindowText(hWndMsg, "Connected"); //вывод строки
memset(text_array, 0, sizeof(text_array)); // обнулить буфер для приема данных
wsprintf(text_array, "MEASURE_STOP\r\n");
DWORD dwBytesRead = 0;
WriteFile(hSerialCom2, text_array, strlen(text_array), &dwBytesRead, &OL);
//** Запустить поток обработки данных COM порта**/
// Создаем объект события для использования
// функцией WaitCommEvent.
OL.hEvent= CreateEvent(
    NULL, // атрибуты безопасности по умолчанию // default security attributes
    true, // автосброс события // manual-reset event
    true, // состояние не сигнальное // not signaled
    NULL // без имени
); // создание события
// Инициализация остатка структуры OVERLAPPED нулями.
OL.Internal = 0;
OL.InternalHigh = 0;
OL.Offset = 0;
OL.OffsetHigh = 0;
assert(OL.hEvent);
SetCommMask(hSerialCom2, EV_RXCHAR); //маска прерывания
hThread = CreateThread(NULL, 0, ThreadProc, NULL, 0, NULL);
SetThreadPriority(hThread, 1);
}
if(wParam==ID_DISCONNECT)
{
}
if(wParam == ID_DISCONNECT_COM_2)
{
    out_file.close();
    memset(text_array, 0, sizeof(text_array)); // обнулить буфер для приема данных
wsprintf(text_array, "MEASURE_STOP\r\n");
DWORD dwBytesRead = 0;
WriteFile(hSerialCom2, text_array, strlen(text_array), &dwBytesRead, &OL);
    cConnectionON = 0;
    cStart = 0;
    cPause = 0;
    CheckDlgButton(hWnd, ID_WRITE_COM_2, BST_UNCHECKED); // отжать кнопку
    CheckDlgButton(hWnd, ID_READ_COM_2, BST_UNCHECKED); // отжать кнопку
    KillTimer(hWnd, TIMER_ID2); //остановка таймера для обновления показаний
    SetWindowText(hWndMsg, "Disconnected"); //вывод строки
// KillTimer(hWnd, ID_TIMER_U);
}
// Обработка кнопки выбора порта
if (wParam == ID_PORT)
{
    SendMessage(hWndCombo, CB_SHOWDROPDOWN, (WPARAM) TRUE, 0);
    SetWindowText(hWndMsg, szText); //вывод строки
}

```

```

}
if (wParam == ID_PORT_COM_2)
{
    SendMessage(hWndComboCom2, CB_SHOWDROPDOWN, (WPARAM) TRUE, 0);
    SetWindowText(hWndMsg, szText); //вывод строки
}
// Обработка кнопки выбора скорости порта
if (wParam == ID_SPEED)
{
    SendMessage(hWndComboSpeed, CB_SHOWDROPDOWN, (WPARAM) TRUE, 0);
    SetWindowText(hWndMsg, szText); //вывод строки
}
if (wParam == ID_SPEED_COM_2)
{
    SendMessage(hWndComboSpeedCom2, CB_SHOWDROPDOWN, (WPARAM) TRUE, 0);
    SetWindowText(hWndMsg, szText); //вывод строки
}
// Обработка кнопки выбора параметра порта
if ((HIWORD(wParam)) == CBN_SELCHANGE)
{
    if (LOWORD(wParam) == ID_PORT)
    {
        sel_port = SendMessage(hWndCombo, CB_GETCURSEL, 0, 0);
        SetWindowText(hWndStatic, items[sel_port]);
        key = SendMessage(hWndCombo, CB_GETCURSEL, 0, 0);
        strcpy(szPortName, items[sel_port]);
        strcpy(szLine, szPortName); //copy
        strcat(szLine, " "); //add
        strcat(szLine, szPortSpeed); //add
        SetWindowText(hWndMsg, szLine); //out
    };
    if (LOWORD(wParam) == ID_SPEED)
    {
        sel_speed = SendMessage(hWndComboSpeed, CB_GETCURSEL, 0, 0);
        SetWindowText(hWndStaticSpeed, baud[sel_speed]);
        strcpy(szPortSpeed, baud[sel_speed]);
        COM_speed = sel_speed;
        strcpy(szLine, szPortName); //copy
        strcat(szLine, " "); //add
        strcat(szLine, szPortSpeed); //add
        SetWindowText(hWndMsg, szLine); //out
    };
    if (LOWORD(wParam) == ID_PORT_COM_2)
    {
        sel_portCom2 = SendMessage(hWndComboCom2, CB_GETCURSEL, 0, 0);
        SetWindowText(hWndStatic, items[sel_portCom2]);
        key_Com2 = SendMessage(hWndComboCom2, CB_GETCURSEL, 0, 0);
        strcpy(szPortNameCom2, items[sel_portCom2]);
        strcpy(szLine, szPortNameCom2); //copy
        strcat(szLine, " "); //add
        strcat(szLine, szPortSpeed); //add
        SetWindowText(hWndMsg, szLine); //out
    };
    if (LOWORD(wParam) == ID_SPEED_COM_2)
    {
        sel_speedCom2 = SendMessage(hWndComboSpeedCom2, CB_GETCURSEL, 0, 0);
        SetWindowText(hWndStaticSpeed, baud[sel_speedCom2]);
        strcpy(szPortSpeed, baud[sel_speedCom2]);
        COM2_speed = sel_speedCom2;
        strcpy(szLine, szPortNameCom2); //copy
        strcat(szLine, " "); //add
        strcat(szLine, szPortSpeed); //add
        SetWindowText(hWndMsg, szLine); //out
    };
    SetFocus(hWnd);
}
if (wParam == ID_ZOOM)
{

```

```

    DialogBox(NULL,MAKEINTRESOURCE(IDD_ZOOM),NULL,PathDlgProcZoom);
    }
    break;
/*****/
    case WM_RBUTTONDOWN:
    {
        point_popup.x = LOWORD(IParam);
        point_popup.y = HIWORD(IParam);
        hMenu_popup = CreatePopupMenu();
        ClientToScreen(hWnd, &point_popup);
        switch(cVoltage)
        {
            case 20:
            {
                AppendMenu(hMenu_popup, MF_STRING, IDM_AT_COMMAND, TEXT("&Определить подключение
прибора"));
                AppendMenu(hMenu_popup, MF_STRING, IDM_SETUP_COMMAND, TEXT("&Отправить уставки"));
                AppendMenu(hMenu_popup, MF_SEPARATOR, 0, NULL);
            } break;
        }
        AppendMenu(hMenu_popup, MF_STRING, IDM_SET_CURRENT, TEXT("&Установить ток"));
        TrackPopupMenu(hMenu_popup, TPM_RIGHTBUTTON, point_popup.x, point_popup.y, 0, hWnd, NULL);
        DestroyMenu(hMenu_popup);
    }
    break;
/*****/
    case WM_CLOSE:
    {
        out_file.close();
        memset(text_array, 0, sizeof(text_array)); // обнулить буфер для приема данных
        wsprintf(text_array, "MEASURE_STOP\r\n");
        DWORD dwBytesRead = 0;
        WriteFile(hSerialCom2, text_array, strlen(text_array), &dwBytesRead, &OL);
        TerminateThread(hThread,0);
        CloseHandle(OL.hEvent);
        closeSerialPort(hSerialCom2);
        DestroyWindow(hWnd);
    }
    break;
/*****/
    case WM_DESTROY:
    {
        PostQuitMessage(0);
    }
    break;
/*****/
    case WM_PAINT:
    {
        GetClientRect(hWnd, &Rect);
        hDC = BeginPaint(hWnd, &ps);
        DrawDiagram(hWnd, hDC);
        hPen = CreatePen(PS_SOLID,1,RGB(255,255,0));
        SelectObject(hDC,hPen);
        for(long int iI = 0; iI < curves.size(); ++iI)
        {
            it = curves[iI].begin();
            MoveToEx(hDC,it->x, it->y, NULL);
            for(it+1;it != curves[iI].end(); ++it)
                LineTo(hDC, it->x, it->y);
        }
        for(long int iI = 0; iI < curves2.size(); ++iI)
        {
            it2 = curves2[iI].begin();
            MoveToEx(hDC,it2->x, it2->y, NULL);
            for(it2+1;it2 != curves2[iI].end(); ++it2)
                LineTo(hDC, it2->x, it2->y);
        }
        EndPaint(hWnd, &ps);
    }
}

```

```

    }
    break;
/*****/
//Обработка сообщения создания главного окна
case WM_CREATE:
{
    hGroupBorder_Com1 = CreateWindow(TEXT("button"), TEXT("Настройка параметров приема пакета"), WS_CHILD |
WS_VISIBLE | BS_GROUPBOX, 10, 0, 420, 65, hWnd, NULL, g_hinst, NULL);
    hGroupBorder_Com2 = CreateWindow(TEXT("button"), TEXT("Настройка COM порта"), WS_CHILD | WS_VISIBLE |
BS_GROUPBOX, 10, 65, 420, 55, hWnd, NULL, g_hinst, NULL);
    hGroupBorder_CVC = CreateWindow(TEXT("button"), TEXT("Снятие х-ки"), WS_CHILD | WS_VISIBLE |
BS_GROUPBOX, 280, 125, 140, 100, hWnd, NULL, g_hinst, NULL);
    // Окно для вывода параметров (вверху)
    // Для вывода номера порта
    hWndStatic = CreateWindow(TEXT("static"), TEXT(""), WS_CHILD | WS_VISIBLE, 450, 75, 50, 20, hWnd, NULL, g_hinst,
NULL);

    // Для вывода скорости порта
    hWndStaticSpeed = CreateWindow(TEXT("static"), TEXT(""), WS_CHILD | WS_VISIBLE, 500, 75, 50, 20, hWnd, NULL,
g_hinst, NULL);

    // "Склеивание строки"
    strcpy(szLine, szPortName); //copy
    strcat(szLine, " "); //add
    strcat(szLine, szPortSpeed); //add
    // Вывод строки на экран
    SetWindowText(hWndMsg, szLine); //out - Connect Disconnect
    // Окно для вывода параметров (внизу)
    hWndMsg = CreateWindow(TEXT("static"), szLine, WS_CHILD | WS_VISIBLE, 450, 100, 300, 20, hWnd, NULL, g_hinst,
NULL);

    hChannel[0] = CreateWindow(TEXT("static"), "Цикл", WS_CHILD | WS_VISIBLE, 250, 235, 280, 20, hWnd, NULL, g_hinst,
NULL);

    hChannel_data[0] = CreateWindow(TEXT("static"), "0", WS_CHILD | WS_VISIBLE | ES_RIGHT, 340, 235, 90, 20, hWnd,
NULL, g_hinst, NULL);
    hChannel[1] = CreateWindow(TEXT("static"), "Напряжение, В", WS_CHILD | WS_VISIBLE, 20, 235, 150, 20, hWnd, NULL,
g_hinst, NULL);
    hChannel_data[2] = CreateWindow(TEXT("static"), "0.000", WS_CHILD | WS_VISIBLE, 160, 235, 90, 20, hWnd, NULL, g_hinst, NULL);
    hWndConnectCom2 = CreateWindow(TEXT("button"), TEXT("Connect"), WS_CHILD | WS_VISIBLE | BS_PUSHLIKE |
BS_AUTOCHECKBOX | WS_TABSTOP, 300, 85, 100, 20, hWnd, (HMENU)ID_CONNECT_COM_2, g_hinst, NULL);
    hStartStop
=
CreateWindow(TEXT("button"), TEXT("Старт"), WS_CHILD | WS_VISIBLE | BS_PUSHLIKE | BS_AUTOCHECKBOX |
WS_TABSTOP, 300, 155, 100, 20, hWnd, (HMENU)ID_WRITE_COM_2, g_hinst, NULL);
    hWndComboCom2 = CreateWindow(TEXT("combobox"), NULL,
WS_CHILD | WS_VISIBLE | CBS_DROPDOWN |
WS_TABSTOP,
20, 85, 120, 210, hWnd, (HMENU)ID_PORT_COM_2,
g_hinst, NULL);
    hWndComboSpeedCom2 = CreateWindow(TEXT("combobox"), NULL,
WS_CHILD | WS_VISIBLE | CBS_DROPDOWN |
WS_TABSTOP,
150, 85, 120, 210, hWnd, (HMENU)ID_SPEED_COM_2,
g_hinst, NULL);

    for ( i = 0; i < 4; i++)
    {
        SendMessage(hWndComboCom2, CB_ADDSTRING, 0, (LPARAM) itemsCom2[i]);
    };
    for ( i = 0; i < 5; i++)
    {
        SendMessage(hWndComboSpeedCom2, CB_ADDSTRING, 0, (LPARAM) baudCom2[i]);
    };
    CreateWindow(TEXT("button"), TEXT("Контроль ответа устройства"), WS_VISIBLE | WS_CHILD | BS_AUTOCHECKBOX, 20, 20, 350, 20,
hWnd, (HMENU)ID_CHECK_DATA, g_hinst, NULL);
    CreateWindow(TEXT("button"), TEXT("Записывать в файл принятый пакет данных"), WS_VISIBLE | WS_CHILD | BS_AUTOCHECKBOX,
20, 40, 350, 20, hWnd, (HMENU)ID_RAW_DATA, g_hinst, NULL);
    CheckDlgButton(hWnd, ID_CHECK_DATA, BST_CHECKED);
    hWndBtr = CreateWindow(TEXT("static"), TEXT("Задать ток, мА"), WS_CHILD | WS_VISIBLE, 20, 125, 200, 20, hWnd,
NULL, g_hinst, NULL);
    hWndStatic_U_min = CreateWindow(TEXT("static"), TEXT("Напряжение разряда, В"), WS_CHILD | WS_VISIBLE, 20, 155,
230, 20, hWnd, NULL, g_hinst, NULL);
}
}

```

```

        hWndStatic_U_max = CreateWindow(TEXT("static"), TEXT("Напряжение заряда, В"), WS_CHILD | WS_VISIBLE, 20, 180,
230, 20, hWnd, NULL, g_hinst, NULL);
        hWndStatic_dU = CreateWindow(TEXT("static"), TEXT("Количество циклов"), WS_CHILD | WS_VISIBLE, 20, 205, 180,
20, hWnd, NULL, g_hinst, NULL);
        hWndEditCurrent = CreateWindow(TEXT("Edit"), TEXT("0"), WS_CHILD | WS_VISIBLE | WS_BORDER | WS_TABSTOP |
ES_RIGHT, 200, 125, 62, 20, hWnd, (HMENU) ID_EDIT3, NULL, NULL);
        hWndEditCom2[0] = CreateWindow(TEXT("Edit"), TEXT("1"), WS_CHILD | WS_VISIBLE | WS_BORDER | WS_TABSTOP
| ES_RIGHT, 200, 155, 62, 20, hWnd, (HMENU) ID_EDIT_0, NULL, NULL);
        hWndEditCom2[1] = CreateWindow(TEXT("Edit"), TEXT("5"), WS_CHILD | WS_VISIBLE | WS_BORDER | WS_TABSTOP
| ES_RIGHT | ES_NUMBER, 200, 205, 62, 20, hWnd, (HMENU) ID_EDIT_1, NULL, NULL);
        hWndEditCom2[2] = CreateWindow(TEXT("Edit"), TEXT("2"), WS_CHILD | WS_VISIBLE | WS_BORDER | WS_TABSTOP
| ES_RIGHT, 200, 180, 62, 20, hWnd, (HMENU) ID_EDIT_2, NULL, NULL);
/*****/
        AddMenus(hWnd);
/*****/

        hMenu = GetMenu(hWnd);
        SetMenuDefaultItem(GetSubMenu(hMenu,0), IDM_OPEN, FALSE);
        hmenuMain = GetMenu(hWnd);
        hmenuChart = CreatePopupMenu();
        AppendMenu(hmenuMain, MF_STRING | MF_POPUP, (UINT) hmenuChart, "&Help");
        AppendMenu(hmenuChart, MF_STRING, IDM_ABOUT, "A&bout\tCtrl+A");
        TBADDBITMAP tbab;
        INITCOMMONCONTROLSEX InitCtrlEx;
        InitCtrlEx.dwSize = sizeof(INITCOMMONCONTROLSEX);
        InitCtrlEx.dwICC = ICC_BAR_CLASSES;
        InitCommonControlsEx(&InitCtrlEx);
        hImageList = ImageList_Create(16, 16, ILC_COLOR32 | ILC_MASK, 3, 0);
        SetWindowText(hWndComboCom2, items[0]);
        SetWindowText(hWndComboSpeed, baud[2]);
        SetWindowText(hWndComboSpeedCom2, baud[2]);

hMemDcFrame = CreateCompatibleDC(hDC);
hBmpFrame = CreateCompatibleBitmap(hDC, 500, 500);
SelectObject(hMemDcFrame, hBmpFrame);
SetGraphicsMode(hMemDcFrame, GM_ADVANCED);
        UpdateWindow(hWnd);
    }
    break;
    case WM_GETMINMAXINFO: //Получили сообщение от Винды
{
    MINMAXINFO *pInfo = (MINMAXINFO *)lParam;
    POINT Min = { 448, 315 };
    #ifdef DEBUG_WINDOW
    POINT Max = {448, 730 };
    #else
    POINT Max = {448, 315 };
    #endif
    pInfo->ptMinTrackSize = Min; // Установили минимальный размер
    pInfo->ptMaxTrackSize = Max; // Установили максимальный размер
    return 0;
} break;
    case WM_SIZE:
    {
        GetClientRect(hWnd, &rect);
    } break;
    case WM_NOTIFY:
    {
        switch (((LPNMHDR) lParam)->code)
        {
            case TTN_GETDISPINFO:
            {
                LPTOOLTIPTEXT lpttt = (LPTOOLTIPTEXT)lParam;
                // Set the instance of the module that contains the resource.
                lpttt->hinst = g_hinst;
                UINT_PTR idButton = lpttt->hdr.idFrom;
                switch (idButton)
                {
                    case ID_BTN_1:
                        lpttt->lpszText = MAKEINTRESOURCE(1901);

```

```

        break;
    case ID_BTN_2:
        lpptt->lpszText = MAKEINTRESOURCE(1902);
        break;
    case ID_BTN_3:
        lpptt->lpszText = MAKEINTRESOURCE(1903);
        break;
    }
    break;
}
}
return TRUE;
    break;
    case WM_TIMER:
        switch(wParam)
        {
            case TIMER_ID2: //таймер для выполнения и измерений
            {
                break;
            case TIMER_ID:
            {
                hDC = GetDC(hWnd);
                if ( abs(ADC_new_DMM * iKa) > 190.)
                {
                    iKa = 190. / abs(ADC_new_DMM) / 10.;
                }
                curve.push_back(Point(210+(int)(fKoeff_x*iIter/iIter_U) + (int)(iU_min/cVoltage*190), 460 -
ADC_old_DMM * iKa));
                curves.push_back(curve);
                if (iIter == 0)//если номер итерации равен 0
                {
                    int DAC_zero;
                    DAC_zero = 210 + (int)(iU_min * 190. / cVoltage); //первое установлено значение ЦАП
                    DrawDiagram_ADC(hWnd, hDC, hMemDcFrame, cRevers, 460-ADC_old_DMM * iKa, 460-
ADC_new_DMM * iKa, DAC_zero, DAC_zero);
                }
                else
                {
                    int DAC_iter_prev, DAC_iter_current;
                    DAC_iter_prev = 210 + (int)(fKoeff_x*(iIter-1)/iIter_U) + (int)(iU_min * 190. / cVoltage);
                    DAC_iter_current = 210 + (int)(fKoeff_x*iIter/iIter_U) + (int)(iU_min * 190. / cVoltage);
                    DrawDiagram_ADC(hWnd, hDC, hMemDcFrame, cRevers, 460-ADC_old_DMM * iKa, 460-
ADC_new_DMM * iKa, DAC_iter_prev, DAC_iter_current);
                    ADC_old_DMM = ADC_new_DMM ;
                    DAC_old = DAC_new;
                }
                RECT rect;
                GetClientRect(hWnd,&rect);
                ReleaseDC(hWnd, hDC);
            }
            break;
            case ID_TIMER_U:
            {
                GetWindowText(hWndEditCurrent, szText, 16);
                DWORD dwWrite = 0;
                WriteFile(hSerialCom2, szText, strlen(szText), &dwWrite, NULL);
            }
            break;
        }
        break;
    default:
        return DefWindowProc(hWnd, uMsg, wParam, lParam);
    };
}
return 0;
};
BOOL CALLBACK PathDlgProc (HWND hDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)

```



```

{
    switch (uMsg)
    {
        HINSTANCE hInst;
        static HWND hDlgStatic;
        static HWND hEdit;
        case WM_INITDIALOG:
        {
            hEdit = GetDlgItem(hDlg, IDC_EDIT_PATH);
            SetDlgItemText(hDlg, IDC_EDIT_PATH, path);
            return TRUE;
        }
        case WM_COMMAND:
            switch (LOWORD(wParam))
            {
                case IDOK:
                {
                    GetDlgItemText (hDlg, IDC_EDIT_PATH, path, 256);
                    EndDialog(hDlg, 0);
                } break;
                case IDCANCEL:
                    EndDialog(hDlg, 0);
                    return TRUE;
            } break;
    }
    return FALSE;
}
BOOL CALLBACK TimeDlgProc (HWND hDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    switch (uMsg)
    {
        HINSTANCE hInst;
        static HWND hDlgStatic;
        case WM_INITDIALOG:
        {
            wsprintf(szTime_interval, "%i", iTime_period); //напечатать текст в переменную для вывода в диалоговом окне
            SetDlgItemText(hDlg, IDC_TIME_EDIT, szTime_interval); //вывести текст в диалоговое окно
            return TRUE;
        }
        case WM_COMMAND:
            switch (LOWORD(wParam))
            {
                case IDOK:
                {
                    //LOG(Time_interval);
                    GetDlgItemText (hDlg, IDC_TIME_EDIT, szTime_interval, 256); //считать текст из диалогового окна
                    iTime_period = atoi(szTime_interval); //преобразовать текст в число для
запуска таймера
                    EndDialog(hDlg, 0);
                } break;
                case IDCANCEL:
                    EndDialog(hDlg, 0);
                    return TRUE;
            } break;
    }
    return FALSE;
}
BOOL CALLBACK PathDlgProcZoom (HWND hDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    switch (uMsg)
    {
        HINSTANCE hInst;
        static HWND hDlgStatic;
        static HWND hEdit;
        RECT rect;
        HBRUSH hBrush;
        HPEN hPen, hPen1;
        GetClientRect(hDlg, &rect);
    }
}

```

```

hBrush = (HBRUSH) GetStockObject(WHITE_BRUSH);
hPen = CreatePen(PS_SOLID, 1, RGB(0,128,256));
case WM_MOVING:
{
    InvalidateRect(hDlg, 0, true);
} break;
    case WM_INITDIALOG:
    {
        return TRUE;
    }
    case WM_COMMAND:
        switch (LOWORD(wParam))
        {
            case IDOK:
            {
                EndDialog(hDlg, 0);
            } break;
            case IDCANCEL:
                EndDialog(hDlg, 0);
                return TRUE;
        } break;
case WM_PAINT:
{
    HDC hDc = GetDC(hDlg);
    PAINTSTRUCT ps;
    hDc = BeginPaint(hDlg, &ps);
    Rectangle(hDc,10,10,880,640);
    SelectObject(hDc,hBrush);
    int iLogNum = 10;
    if (cScaleType)
    {
        for (int i = 0; i < iLogNum; i++)
        {
            if (i != 0)
            {
                sprintf(szText,"10^(%i) A", -iLogNum + i);
                TextOutA(hDc,800,30 + int ((iLogNum - 1 - i) * 610/(iLogNum - 1)),szText,9);
            }
            MoveToEx(hDc,20,20 + int (((iLogNum - 1) - i) * 610/(iLogNum - 1)), NULL);
            LineTo(hDc,870,20 + int (((iLogNum - 1) - i) * 610/(iLogNum - 1)));
        }
        MoveToEx(hDc,20,630, NULL);
        LineTo(hDc,870,630);
        MoveToEx(hDc,20,630, NULL);
        LineTo(hDc,20,20);
    }
    else
    {
        MoveToEx(hDc,20,315, NULL);
        LineTo(hDc,870,315);
        MoveToEx(hDc,20,630, NULL);
        LineTo(hDc,20,20);
    }
    if (CurrentVector.size() > 0)
    {
        if (!cScaleType)
        {
            MoveToEx(hDc,20,315 - (int)(abs(CurrentVector[0]) * 1E+6 * iKa * 305. / 190.), NULL);
        }
        else
        {
            MoveToEx(hDc,20,20 - (int)(log10(abs(CurrentVector[0])) * 610. / (iLogNum - 1)), NULL);
        }
    }
    hPen1=CreatePen(PS_SOLID, 1, RGB(0,120,0));
    SelectObject(hDc, hPen1);
    if (!cScaleType)
    {

```

```

for (int i = 0; i < CurrentVector.size(); ++i)
{
    if (abs(CurrentVector[i]) > 1E-6)
    {
        LineTo(hDc, 20 + (i * 850 / iIter_U), 315 - (int)(CurrentVector[i] * 1E+6 * iKa *
305. / 190.));
    }
    else
    {
        LineTo(hDc, 20 + (i * 850 / iIter_U), 315 - (int)(CurrentVector[i] * 1E+9 * iKa *
305. / 190.));
    }
}
else
{
    for (int i = 1; i < CurrentVector.size(); ++i)
    {
        LineTo(hDc, 20 + ((i - 1) * 850 / iIter_U), 20 - (int)(log10(abs(CurrentVector[i])) * 610. / (iLogNum - 1)));
    }
}
EndPoint(hDlg, &ps);
}
}
return FALSE;
}
}

```