

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

КОМПЛЕКСНА БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«КІБЕРФІЗИЧНА СИСТЕМА З ІНТЕГРАЦІЄЮ
БІОЕЛЕКТРИЧНИХ СЕНСОРІВ.**

ЧАСТИНА 2. ПРОГРАМНА ЧАСТИНА»

08— 54.КБКР.034.00.000 ПЗ

Виконав: студент 4 курсу, групи 2КІ— 23мс
спеціальності 123 – Комп'ютерна інженерія
(шифр і назва напрямку підготовки
спеціальності)

_____ Дудник В.В.
(прізвище та ініціали)

Керівник: к.т.н., доцент

_____ Дудник О.В.
(прізвище та ініціали)

«___» _____ 2025 р.

Рецензент: к.т.н., доцент

_____ Чехместрук Р. Ю.
(прізвище та ініціали)

«___» _____ 2025 р.

Допущено до захисту
Завідувач кафедри ОТ
д.т.н., проф. Азаров О.Д.

Азаров О.Д. 09.06. 2025 року

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

Освітньо— кваліфікаційний рівень бакалавр

Галузь знань – 12 Інформаційні технології

Спеціальність – 123 Комп'ютерна інженерія

Освітня програма – Комп'ютерна інженерія



ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ
д.т.н., проф. Азаров О.Д.

« 21 » березня 2025 року

ЗАВДАННЯ НА КОМПЛЕКСНУ БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Дуднику Віктору Вікторовичу

1 Тема роботи: «Кіберфізична система з інтеграцією біоелектричних сенсорів. Частина 2. Програмна частина», керівник роботи Дудник О. В. затверджені наказом вищого навчального закладу від 20 березня 2025 р. №97.

2 Термін подання студентом роботи 13.05.2025.

3 Вихідні дані до роботи: програмний код для мікроконтролера, відповідального за зчитування та передачу біоелектричних сигналів, реалізований мовою C++, алгоритми обробки та фільтрації сигналів, реалізовані у вигляді окремих програмних модулів, документація до програмного забезпечення, включно з описом функціоналу, структурою коду та інструкціями з використання.

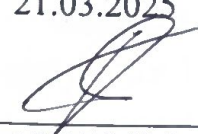
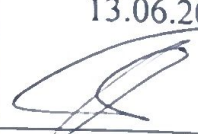
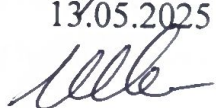
4 Зміст розрахунково — пояснювальної роботи: огляд і аналіз джерел інформації, обґрунтування технічного рішення, аналіз та вибір програмного забезпечення для розробки кіберфізичної системи з інтеграцією біоелектричних сенсорів, розробка алгоритмів та програмного забезпечення кіберфізичної системи, експериментальні дослідження та тестування програмної частини, висновки, список використаних джерел, додатки.

5 Перелік графічного матеріалу: лістинги, фото виконаних тестувань, порівняльні таблиці.

6 Консультанти розділів роботи наведені в таблиці 1.

7 Дата видачі завдання 21.03.2025 р.

Таблиця 1 – Консультанти роботи

Розділ	Прізвище, ініціали та посада консультанта	Дата	
		завдання видав	завдання прийняв
1— 4	к.т.н., доцент каф. ОТ Дудник О.В.	21.03.2025 	13.06.2025 
Нормоконтроль	ас. каф. ОТ Швець І.С.	13.05.2025 	30.05.2025

8 Календарний план виконання приведений в таблиці 2.

Таблиця 2 – Календарний план

№	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Вибір та затвердження теми БКР	21.03.2025	21.03.2025	в.к.
2	Збір інформації	22.03.2025	31.03.2025	в.к.
3	Огляд предметної області	01.04.2025	09.04.2025	в.к.
4	Збір інформації щодо програмної обробки біосигналів та архітектури кіберфізичних систем	10.04.2025	15.04.2025	в.к.
5	Огляд наявного ПЗ для роботи з біоелектричними сигналами	16.04.2025	17.04.2025	в.к.
6	Формування технічного завдання на програмну частину, постановка задачі	18.04.2025	23.04.2025	в.к.
7	Проектування структури програмного модуля, розробка алгоритмів обробки сигналів, реалізація програмного коду, зчитування даних із сенсорів, передача даних	24.04.2025	03.05.2025	в.к.
8	Тестування програмної частини, валідація результатів, усунення помилок	03.05.2025	04.05.2025	в.к.
9	Оформлення пояснювальної записки	06.05.2025	12.05.2025	в.к.
10	Попередній захист БКР	13.05.2025	13.05.2025	в.к.

Студент

Керівник роботи



Віктор ДУДНИК



Олександр ДУДНИК

ЗМІСТ

ВСТУП.....	5
1 ОГЛЯД І АНАЛІЗ ДЖЕРЕЛ ІНФОРМАЦІЇ, ОБГРУНТУВАННЯ ТЕХНІЧНОГО РІШЕННЯ.....	9
1.1 Загальний огляд кіберфізичних систем.....	9
1.2 Огляд сучасних технологій у кіберфізичних системах	10
1.3 Виклики та перспективи розвитку кіберфізичних систем.....	11
1.4 Аналіз та класифікація електричних сенсорів	12
1.4.1 Основні сфери застосування	13
1.4.2 Біофізичні основи вимірювання електричних сигналів.....	13
1.4.3 Структура та компоненти сенсорів	14
1.5 Визначення критеріїв ефективності та точності сигналів	16
1.6 Постановка задачі розробки та формулювання технічних вимог	17
2 ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	19
2.1 Аналіз вимог до програмного забезпечення.....	19
2.2 Критерії вибору інструментів розробки.....	20
2.3 Порівняльна характеристика доступних програмних засобів	21
2.4 Вибір мови програмування та середовища розробки	23
2.5 Вибір бібліотек та фреймворків для роботи з сигналами.....	24
2.6 Програмне забезпечення для візуалізації та реєстрації даних.....	26
2.7 Платформи для інтерактивної візуалізації даних.....	27
3 РОЗРОБКА АЛГОРИТМІВ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ.....	29
3.1 Проектування структури програмного забезпечення	29
3.2 Реалізація алгоритмів обробки електричних сигналів.....	30
3.3 Розробка інтерфейсу користувача	32
3.4 Інтеграція програмного забезпечення з апаратною частиною	35
3.5 Практичне застосування та рекомендації з експлуатації.....	37

3.5.1	Рекомендовані сценарії використання	37
3.5.2	Усунення типових несправностей	38
3.6	Інтеграція з іншими медичними та технічними пристроями.....	38
3.7	Напрями вдосконалення алгоритмів обробки біосигналів.....	39
4	ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА ТЕСТУВАННЯ	
	ПРОГРАМНОЇ ЧАСТИНИ	42
4.1	Методика експериментальних досліджень.....	42
4.1.1	Структура експериментального дослідження	42
4.1.2	Апаратне та програмне забезпечення для тестування.....	43
4.1.3	Методики тестування окремих компонентів	44
4.2	Тестування роботи програмного забезпечення з біоелектричними сенсорами	48
4.2.1	Оптимізація параметрів фільтрації ЕМГ— сигналу.....	48
4.2.2	Тестування системи калібрування.....	50
4.2.3	Тестування режимів роботи системи	51
4.2.4	Аналіз енергоспоживання системи	53
4.3	Проведення тестувань системи	54
4.3.2	Ефективність інтерфейсу користувача.....	54
4.3.3	Ефективність керування виконавчими механізмами.....	55
4.3.4	Обмеження системи та шляхи вдосконалення	56
4.4	Порівняльний аналіз з комерційними аналогами.....	58
4.4.1	Порівняння ключових характеристик	59
4.4.2	Детальне порівняння з комерційними системами.....	59
4.4.3	SWOT— аналіз розробленої системи.....	60
	ВИСНОВКИ.....	61
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65
	ДОДАТКИ.....	69
	Додаток А Технічне завдання	70
	Додаток Б Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	74

Додаток В Графічна схема	75
Додаток Г Ілюстративна частина.....	77
Додаток Д Лістинг – Піни для підключення усіх компонентів до плати ESP32	79
Додаток Е Лістинг – Обробка та збереження станів	80
Додаток Ж Лістинг – Основна функція loop і початок обробки	81
Додаток З Лістинг – Визначення станів та оновлення	82
Додаток К Лістинг – Контроль серво двигунами	83
Додаток Л Довідка про впровадження результатів	83

ВСТУП

У сучасному світі кіберфізичні системи стрімко розвиваються та знаходять широке застосування в різних сферах — від медицини й спорту до наукових досліджень, промисловості та побуту. Особливої актуальності набуває використання біоелектричних сенсорів, які дозволяють отримувати високоточні фізіологічні параметри організму в режимі реального часу. Це відкриває нові можливості для оперативного моніторингу стану здоров'я, раннього виявлення патологій, прогнозування фізичних навантажень і підвищення ефективності як лікувальних, так і тренувальних процесів.

Із розвитком сучасних засобів цифрової обробки сигналів, алгоритмів машинного навчання та технологій аналізу великих даних (Big Data), кіберфізичні системи виходять на якісно новий рівень. Вони здатні не лише пасивно збирати інформацію, а й активно її аналізувати, інтерпретувати та адаптувати свої дії відповідно до змін фізіологічного стану користувача. Таким чином забезпечується високий рівень персоналізації та інтелектуалізації IT-рішень.

У межах цієї кваліфікаційної роботи досліджується програмна частина кіберфізичної системи з інтеграцією біоелектричних сигналів, що є ключовим елементом у забезпеченні її функціональності. Основну увагу зосереджено на реалізації ефективних алгоритмів збору, фільтрації, обробки та візуалізації біоелектричних даних. Важливою складовою є також побудова користувацького інтерфейсу, який забезпечує зручну взаємодію з системою та відображення критично важливої інформації в доступному вигляді. Крім того, розробка враховує вимоги до стабільності й продуктивності програмного забезпечення в умовах безперервного потоку даних, що надходять у режимі реального часу.

Запропоновані підходи мають прикладне значення й можуть бути інтегровані в реальні IT-рішення, орієнтовані на біомедичний моніторинг, спортивну аналітику, системи охорони здоров'я та інші інноваційні напрями, що базуються на інтелектуальній обробці фізіологічних сигналів.

Актуальність теми дослідження обумовлена значним розвитком технологій збору та аналізу біоелектричних сигналів зумовлює необхідність створення ефективних програмних рішень для їхньої обробки. Розробка програмного забезпечення для кіберфізичних систем дозволяє не лише автоматизувати процеси збору та аналізу даних, а й забезпечити їхню адаптивність і інтеграцію з іншими технологіями.

Об'єктом дослідження є кіберфізична система з інтеграцією біоелектричних сенсорів, зокрема її програмна складова, що відповідає за обробку, аналіз та вивід отриманих даних.

Предметом дослідження є методи обробки біоелектричних сигналів, програмні засоби їх аналізу та інтеграція з апаратними компонентами кіберфізичної системи.

Метою даної роботи є розробка та впровадження програмної частини кіберфізичної системи з біоелектричними сенсорами для збору, обробки та візуалізації сигналів у реальному часі.

Для досягнення поставленої мети передбачено вирішення таких задач:

- аналіз існуючих підходів та технологій у сфері програмного забезпечення для кіберфізичних систем з біоелектричними сенсорами;
- дослідження методів обробки та аналізу біоелектричних сигналів для забезпечення їхньої точності та надійності;
- розробка алгоритмів збору, фільтрації та інтерпретації біоелектричних даних у режимі реального часу;
- створення програмного забезпечення для інтеграції апаратної частини системи з користувацьким інтерфейсом;
- розробка методів візуалізації отриманих даних для забезпечення зручності аналізу результатів;
- тестування та оптимізація програмної частини для забезпечення стабільної та ефективної роботи системи. сигналів.

Наукова та технічна новизна кваліфікаційної роботи полягає у розробці та

впровадженні програмної частини кіберфізичної системи з інтеграцією біоелектричних сенсорів, яка забезпечує ефективний збір, обробку та аналіз сигналів у реальному часі.

Практичні результати цього дослідження можуть бути корисними для програмістів та інженерів, які займаються розробкою програмного забезпечення для кіберфізичних систем з біоелектричними сенсорами.

Вони можуть застосовуватися у створенні інтелектуальних платформ для обробки фізіологічних даних у медичній, спортивній та науковій сферах. Також результати дослідження будуть корисні для користувачів, яким необхідні ефективні та зручні програмні рішення для аналізу біоелектричних сигналів у реальному часі.

Планується **апробація** розробленої системи, а саме: публікація наукових статей за результатами дослідження. У перспективі надалі передбачається доопрацювання розробки з метою його реалізації для масового виробництва та практичного застосування. За результатами дослідження було **опубліковано** тези доповіді:

1. Язвінська Д. І., Дудник В. В., Дудник О. В. Роботизований протез руки з електронним керуванням. Modern Trends in the Development of Economy, Technology and Industry : конф., м. Торонто, 12–14 лют. 2025 р. Toronto, 2025. С. 22 – 23.

2. Язвінська Д. І., Дудник В. В., Дудник О. В. Програмно–апаратна реалізація кіберфізичної системи з біоелектричними сенсорами. Всеукраїнська науково–технічна конференція факультету інформаційних технологій та комп'ютерної інженерії : конф., м. Вінниця, 24.03.2025 – 27.03.2025. Вінниця, 2025. URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/24101/19942> (дата звернення: 15.05.2025).

3. Язвінська Д. І., Дудник В. В., Дудник О. В. Кіберфізичні системи з біоелектричними сенсорами: теоретичні засади та технічна реалізація. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ –2025 : конф., м. Київ, 15 трав. 2025 р. Київ, 2025.

С. 215–216. URL: <https://zcit.kubg.edu.ua/index.php/journal/index> (дата звернення: 17.05.2025).

4. Язвінська Д. І., Дудник В. В., Дудник О. В. 3D–модельювання та виробництво компонентів кіберфізичної системи для роботи з біосигналами. Стан, досягнення і перспективи інформаційних систем і технологій : конф., м. Одеса, 17– 18 квіт. 2025 р. Одеса, 2025. С. 284 – 285. URL: https://ontu.edu.ua/download/konfi/2025/Conference_abstract-IT-2025.pdf. (дата звернення: 17.05.2025).

1 ОГЛЯД І АНАЛІЗ ДЖЕРЕЛ ІНФОРМАЦІЇ, ОБГРУНТУВАННЯ ТЕХНІЧНОГО РІШЕННЯ

1.1 Загальний огляд кіберфізичних систем

Кіберфізична система — це система, яка управляється або моніториться комп'ютерним алгоритмом і має зв'язок з Інтернетом та його користувачами. В цих системах програмне забезпечення інтегровано з фізичними об'єктами.

Компоненти кіберфізичних систем можуть взаємодіяти на різних рівнях часу та простору, мати різні моделі поведінки та змінювати способи взаємодії залежно від конкретного контексту. Прикладом таких систем є розумні енергомережі, безпілотні транспортні засоби, автоматизовані системи управління, роботизовані комплекси та автономні дрони [1].

Кіберфізичні системи (КФС) представляють собою інтегровані комплекси, що поєднують фізичні компоненти з цифровими технологіями, забезпечуючи взаємодію між апаратною та програмною частинами. Вони базуються на сенсорах, обчислювальних пристроях та комунікаційних технологіях, що дозволяють здійснювати аналіз даних у реальному часі та приймати автоматизовані рішення. Основною особливістю КФС є їх здатність до самонавчання, адаптації та автономного функціонування [2].

Кіберфізичні системи знаходять широке застосування в різних галузях, значно підвищуючи ефективність та безпеку процесів. У медицині та біотехнологіях вони відіграють важливу роль у моніторингу життєвих показників пацієнтів, таких як кардіомоніторинг, електроміографія (ЕМГ) та електроенцефалографія (ЕЕГ). Біоелектричні сенсори дозволяють створювати сучасні протези та нейроінтерфейси, які керуються біосигналами людини, що суттєво покращує якість життя пацієнтів.

У сфері промисловості 4.0 кіберфізичні системи забезпечують автоматизацію виробничих процесів, оптимізацію технологічних операцій та підвищення ефективності роботизованих маніпуляторів. Вони також відіграють ключову роль у розвитку розумних міст, де застосовуються в інтелектуальних

транспортних системах, енергоменеджменті та екологічному моніторингу через розгалужені мережі сенсорів[3].

Автомобільна галузь активно використовує кіберфізичні технології для створення інтелектуальних систем допомоги водію (ADAS) та автономного транспорту, який керується штучним інтелектом у поєднанні з сенсорними пристроями. Військова сфера також інтегрує ці системи у свої розробки, зокрема для керування дронами, безпілотними наземними та повітряними апаратами, а також для створення екзоскелетів, що підвищують мобільність та витривалість військових. Таким чином, кіберфізичні системи стають важливим інструментом у забезпеченні інновацій та технологічного прогресу в різних сферах людської діяльності [4].

1.2 Огляд сучасних технологій у кіберфізичних системах

Розвиток кіберфізичних систем безпосередньо пов'язаний із впровадженням сучасних технологій, які забезпечують ефективність, швидкість та надійність їхньої роботи. Однією з ключових складових є біоелектричні сенсори, що дозволяють вимірювати електричні сигнали м'язів, мозку та серця. Дані, отримані за допомогою електроміографії (ЕМГ), електроенцефалографії (ЕЕГ) та електрокардіографії (ЕКГ), використовуються для подальшої обробки й аналізу, що відкриває широкі можливості в медицині, спортивній аналітиці та нейроінтерфейсах.

Важливу роль відіграє Інтернет речей (IoT), що забезпечує безперервний зв'язок між пристроями через бездротові мережі. Це дозволяє збирати, передавати та обробляти дані в реальному часі, що особливо актуально для систем дистанційного моніторингу здоров'я, автоматизованого виробництва та розумного середовища. Доповнюють ці можливості технології штучного інтелекту та машинного навчання, які аналізують біосигнали, прогнозують можливі стани користувача та автоматично коригують роботу системи. Алгоритми нейронних мереж здатні адаптуватися до змін, покращуючи точність аналізу та підвищуючи рівень автономності кіберфізичних систем.

Суттєвий вплив на розвиток КФС мають технології Edge та Cloud Computing, які забезпечують швидку та ефективну обробку даних. Завдяки розподіленим обчислювальним ресурсам, частина операцій виконується безпосередньо на пристроях, що зменшує затримки у відповідях системи, тоді як складні обчислення можуть здійснюватися у хмарному середовищі. Це особливо важливо для медичних додатків та автономних систем, де час реакції відіграє критичну роль.

Не менш важливим аспектом є кібербезпека, адже кіберфізичні системи працюють із персональними та критично важливими даними. Захист інформації здійснюється через сучасні методи шифрування, автентифікацію користувачів та розробку механізмів протидії кібератакам. Надійність таких систем є ключовим фактором їхньої успішної інтеграції у медицину, транспорт, промисловість та інші сфери, де безпека даних є пріоритетною.

Завдяки інтеграції цих передових технологій кіберфізичні системи стають все більш автономними, точними та ефективними, відкриваючи нові можливості для впровадження інновацій у різних галузях [4, 5].

1.3 Виклики та перспективи розвитку кіберфізичних систем

Попри значні успіхи у впровадженні кіберфізичних систем, існує низка викликів, які потребують вирішення для подальшого розвитку цієї технології. Одним із ключових аспектів є оптимізація енергоспоживання, оскільки автономні пристрої, що використовуються в таких системах, повинні мати низьке споживання енергії, щоб забезпечити тривалу роботу без частих замін або підзарядок. Це вимагає розробки більш енергоефективних сенсорів, вдосконалених алгоритмів обробки даних та ефективних методів управління живленням.

Ще одним важливим викликом є забезпечення високої точності та надійності вимірювань. Багато сенсорів, що використовуються у кіберфізичних системах, можуть піддаватися впливу зовнішніх факторів, що призводить до шумів і спотворення даних. Тому вдосконалення методів фільтрації сигналів,

калібрування датчиків та використання адаптивних алгоритмів обробки інформації є необхідними кроками для підвищення ефективності таких систем.

Крім технічних аспектів, важливу роль відіграють етичні та правові питання, пов'язані із застосуванням кіберфізичних систем. Зокрема, особливу увагу привертає захист персональних даних, адже велика кількість КФС, особливо у сфері медицини та «розумних» міст, збирає та обробляє конфіденційну інформацію. Необхідність створення чітких стандартів щодо зберігання, обробки та передачі таких даних стає критичною умовою для їхнього безпечного використання. Також важливим є питання нормативно— правового регулювання, яке має враховувати як етичні аспекти, так і потребу в гармонізації міжнародних стандартів[6].

Подальший розвиток кіберфізичних систем, особливо з інтеграцією біоелектричних сенсорів, відкриває широкі перспективи у різних сферах. У медицині це сприятиме вдосконаленню методів діагностики та лікування, в промисловості — підвищенню ефективності автоматизованих виробничих процесів, а в повсякденному житті — створенню розумних середовищ, що адаптуються до потреб людини. Завдяки розвитку технологій штучного інтелекту, Інтернету речей та квантових обчислень кіберфізичні системи ставатимуть дедалі автономнішими, точнішими та безпечнішими, що дозволить реалізовувати нові інноваційні рішення для суспільства.

1.4 Аналіз та класифікація електричних сенсорів

Біоелектричні сенсори відіграють ключову роль у сучасній медицині, нейротехнологіях та інших сферах, що пов'язані з моніторингом фізіологічних процесів в організмі людини. Вони дозволяють отримувати важливу інформацію про стан здоров'я, зокрема фіксувати електричну активність серця, м'язів, мозку та інших органів. Завдяки стрімкому розвитку технологій біоелектричні сенсори стають дедалі точнішими, компактнішими та енергоефективнішими, що сприяє їхньому широкому впровадженню у повсякденне життя. Зокрема, носимі пристрої, медичні імпланти та нейроінтерфейси активно використовують ці

сенсори для постійного моніторингу та автоматизованої обробки даних.

Дослідження в цій сфері є надзвичайно актуальним, оскільки підвищення точності вимірювань та розробка нових матеріалів для сенсорів відкривають можливості для створення персоналізованих медичних систем. Окрім того, застосування біоелектричних сенсорів у галузях робототехніки, спорту та військових технологій розширює горизонти їхнього використання.

1.4.1 Основні сфери застосування

Біоелектричні сенсори активно використовуються в медицині, спортивній діагностиці, біоніці, а також у сферах безпеки та розумних технологій.

Медицина – одна з головних сфер застосування біоелектричних сенсорів. Вони використовуються для діагностики та контролю стану пацієнтів, наприклад, у кардіології (ЕКГ), неврології (ЕЕГ), реабілітації (ЕМГ).

Завдяки біоелектричним сенсорам можливе створення протезів, які керуються сигналами м'язів, а також імплантів, що допомагають у лікуванні неврологічних захворювань. У спорті та фізичній активності біоелектричні сенсори дозволяють спортсменам та тренерам аналізувати активність м'язів, рівень навантаження та загальний стан організму під час тренувань. Це допомагає оптимізувати тренувальні процеси та запобігати травмам. У робототехніці та біоніці біосенсори застосовуються у створенні нейроінтерфейсів, які дозволяють керувати протезами або екзоскелетами за допомогою електричних сигналів мозку чи м'язів.

А використання біоелектричних сигналів у біометричних системах дозволяє створювати унікальні методи аутентифікації, наприклад, на основі електроенцефалограми або електроміографічних даних. Також біоелектричні сенсори інтегруються в системи «розумного будинку» та носимі пристрої для моніторингу здоров'я користувачів у режимі реального часу [7].

1.4.2 Біофізичні основи вимірювання електричних сигналів

Біоелектричні сигнали генеруються клітинами живих організмів унаслідок

різниці потенціалів між внутрішньою та зовнішньою поверхнею клітинної мембрани. Зміни цього потенціалу відбуваються в результаті активності іонних каналів, які контролюють рух іонів натрію, калію, кальцію та хлору через мембрану.

Основним механізмом реєстрації біоелектричних сигналів є електроди, що зчитують зміни електричного потенціалу на поверхні шкіри або безпосередньо всередині тіла. Наприклад, електрокардіографічні (ЕКГ) сенсори фіксують електричну активність серця, а електроенцефалографічні (ЕЕГ) – активність нейронів головного мозку.

Важливим аспектом є якість сигналу, яка залежить від перешкод, шумів та артефактів, що можуть виникати через рухи людини або поганий контакт електродів зі шкірою. Тому використовуються різні методи фільтрації та обробки сигналів для отримання більш точних даних.

1.4.3 Структура та компоненти сенсорів

Біоелектричний сенсор складається з кількох основних компонентів, які забезпечують прийом, обробку та передачу сигналу. Центральним елементом є електроди, що безпосередньо контактують із тілом людини і зчитують електричну активність біологічних тканин, таку як м'язова або нервова.

Вони можуть бути поверхневими — у разі неінвазивного застосування, або імплантованими — коли потрібна висока точність, стабільність сигналу чи довготривалий моніторинг.

Сигнали, зчитані електродами, зазвичай є дуже слабкими, тому перед подальшою обробкою проходять стадію підсилення за допомогою операційного підсилювача.

У деяких випадках застосовуються також фільтри для усунення шумів та артефактів, що покращує якість сигналу. Основні компоненти біоелектричних сенсорів зображені в таблиці 1.1.

Щоб усунути небажані перешкоди, які можуть виникати через рухи, вплив зовнішніх електромагнітних полів або інші чинники, застосовуються фільтри

шуму. Отримані аналогові сигнали проходять через аналогово— цифровий перетворювач (АЦП), який конвертує їх у цифровий формат для подальшого аналізу. Процес обробки сигналів контролюється процесором або мікроконтролером, що здійснює фільтрацію, аналіз та передає отримані дані у комп'ютер чи мобільний пристрій[8].

Таблиця 1.1 – Основні компоненти сенсорів

Компонент	Опис
Електроди	основний елемент, який контактує з тілом та зчитує електричні сигнали. Вони можуть бути поверхневими (для неінвазивного вимірювання) або імплантованими (для високоточної діагностики)
Підсилювач сигналу	компонент, що збільшує амплітуду слабких біоелектричних сигналів, щоб зробити їх придатними для подальшої обробки
Фільтри шуму	схеми або алгоритми, що видаляють небажані перешкоди, які можуть виникати через рухи, вплив зовнішніх електромагнітних полів або інші чинники.
Аналогово—цифровий перетворювач	пристрій, що конвертує аналоговий біосигнал у цифровий формат для подальшого аналізу.
Процесор або мікроконтролер	керує обробкою сигналів, виконує попередню фільтрацію, аналіз та передає дані у комп'ютер чи мобільний пристрій
Інтерфейс зв'язку	механізм, що забезпечує передачу даних. Це може бути дротове (USB, оптоволоконне) або бездротове (Bluetooth, Wi— Fi) з'єднання
Джерело живлення	акумулятор або інші джерела енергії, що забезпечують роботу пристрою

Передача інформації здійснюється за допомогою інтерфейсу зв'язку, який може бути як дротовим (наприклад, через USB або оптоволоконне з'єднання), так і бездротовим (Bluetooth, Wi— Fi). Живлення сенсора забезпечується вбудованим акумулятором або іншими джерелами енергії, що підтримують безперебійну роботу пристрою.

Сучасні біоелектричні сенсори стають дедалі компактнішими та інтегрованими в носимі пристрої, що дозволяє використовувати їх у повсякденному житті без дискомфорту для користувачів. Завдяки розвитку нанотехнологій, біоінженерії та штучного інтелекту, майбутні покоління сенсорів будуть ще точнішими та ефективнішими.

1.5 Визначення критеріїв ефективності та точності сигналів

У кіберфізичних системах із біоелектричними сенсорами одним із ключових завдань є забезпечення високої точності збору, обробки та аналізу біосигналів. Для цього необхідно визначити набір критеріїв, які дозволяють кількісно оцінити ефективність функціонування сенсорної системи та якість отриманих даних.

До критеріїв ефективності відносять час реакції системи, тобто затримку між виникненням біосигналу та його обробкою, пропускну здатність, що характеризує кількість сигналів, які система здатна обробити за певний проміжок часу, а також рівень енергоспоживання, який особливо важливий у мобільних або автономних пристроях.

Точність роботи біоелектричних сенсорів визначається низкою показників. Серед них — чутливість, яка відображає здатність сенсора виявляти навіть слабкі біоелектричні сигнали, а також точність вимірювання, тобто відповідність отриманих результатів реальним фізіологічним параметрам. Важливим є й показник співвідношення сигнал/шум, що демонструє, наскільки чітко можна виділити корисну інформацію на фоні перешкод. Повторюваність вимірювань також є критичною — система має давати стабільні результати при багаторазових вимірюваннях за однакових умов [9].

Окрім цього, в системах обробки біосигналів важливою є здатність алгоритмів до адаптації — вони повинні забезпечувати коректну роботу навіть у разі наявності шумів, рухів користувача чи змін зовнішніх умов. Комплексна оцінка наведених критеріїв дозволяє забезпечити надійне функціонування кіберфізичної системи та її ефективне використання в обраній сфері застосування. Визначення критеріїв ефективності та точності біосигналів зображені на рисунку 1.1

Визначення критеріїв ефективності та точності біосигналів



Рисунок 1.1 – Визначення критеріїв ефективності та точності біосигналів

1.6 Постановка задачі розробки та формулювання технічних вимог

Розробка біоелектричного сенсора є складним інженерним завданням, що потребує чіткого визначення його функціональних можливостей та технічних характеристик. Основною метою є створення сенсорної системи, здатної з високою точністю зчитувати, обробляти та передавати біоелектричні сигнали людського тіла для подальшого аналізу та використання в медичних або дослідницьких цілях.

При постановці задачі необхідно враховувати основні параметри, які визначають ефективність роботи сенсора. По—перше, сенсор повинен мати високу чутливість для точного виявлення біоелектричних сигналів, таких як електрокардіограма (ЕКГ), електроенцефалограма (ЕЕГ) або електроміограма

(ЕМГ). Це потребує використання якісних електродів та ефективного підсилювача сигналу. По — друге, необхідно забезпечити мінімізацію шумів і зовнішніх перешкод за допомогою відповідних фільтрів, що особливо важливо для медичних досліджень та діагностики.

Система повинна підтримувати як дротову, так і бездротову передачу даних, що забезпечить зручність використання та можливість інтеграції з іншими пристроями. Важливим аспектом є також низьке енергоспоживання, що дозволить працювати пристрою тривалий час без необхідності підзарядки. Точність вимірювань, стабільність сигналу та швидкість обробки є критичними параметрами, які визначають якість кінцевого продукту.

Таким чином, формулювання технічних вимог включає розробку високочутливих електродів, ефективного підсилювача сигналу, системи шумозаглушення, якісного аналого—цифрового перетворювача та процесора для аналізу даних. Також необхідно забезпечити сумісність сенсора з сучасними мобільними та комп'ютерними системами, а його конструкція має бути ергономічною, безпечною та зручною для користувачів.

2 ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Аналіз вимог до програмного забезпечення

Програмне забезпечення кіберфізичної системи з інтеграцією біоелектричних сенсорів повинно забезпечувати надійний збір, обробку, аналіз та візуалізацію біосигналів у реальному часі. Визначення вимог до програмної частини є критичним етапом, який забезпечує відповідність системи технічному завданню та її подальшу масштабованість і ефективність.

До функціональних вимог належить забезпечення стабільного зчитування біоелектричних сигналів із сенсорів (наприклад, ЕКГ, ЕМГ), їх попередньої фільтрації та очищення від шумів, збереження отриманих даних у зручному форматі, а також візуалізація сигналів у вигляді графіків або числових значень. Крім того, необхідною є реалізація інтерфейсу користувача для взаємодії з системою, налаштування параметрів та перегляду результатів у зручній формі [11].

Нефункціональні вимоги охоплюють надійність, точність обробки сигналів, низьку затримку обчислень, а також можливість роботи в умовах обмежених апаратних ресурсів. Особлива увага приділяється масштабованості системи, тобто можливості інтегрувати нові типи сенсорів або розширювати функціональність без необхідності повного переписування коду. Також важливою є безпека обробки біомедичних даних, особливо у випадках медичного застосування системи.

Враховуючи специфіку роботи з біоелектричними сенсорами, до системи пред'являються підвищені вимоги до точності вимірювань, чутливості алгоритмів до змін сигналу, а також забезпечення високого співвідношення сигнал/шум.

У результаті проведеного аналізу було сформовано технічне завдання для розробки програмного забезпечення, яке враховує всі вищезазначені вимоги й забезпечує ефективну роботу кіберфізичної системи у режимі реального часу.

2.2 Критерії вибору інструментів розробки

Вибір програмних інструментів для розробки кіберфізичної системи з інтеграцією біоелектричних сенсорів повинен ґрунтуватися на сукупності технічних, функціональних та експлуатаційних вимог. У цьому контексті важливо враховувати як специфіку обробки біосигналів, так і необхідність інтеграції програмної частини з апаратними компонентами системи.

Одним із ключових критеріїв є сумісність із апаратним забезпеченням, а саме можливість прямої або через драйвери взаємодії з біоелектричними сенсорами, мікроконтролерами чи іншими периферійними пристроями. Обране середовище розробки має підтримувати відповідні протоколи (наприклад, UART, SPI, I2C) та мати бібліотеки для швидкої інтеграції.

Наступним критерієм є продуктивність та ефективність роботи, зокрема швидкість обробки сигналів і можливість реалізації алгоритмів у реальному часі. Програмні засоби повинні забезпечувати низьку затримку обчислень, особливо в задачах аналізу електрофізіологічних сигналів.

Гнучкість та масштабованість — ще один важливий фактор. Обрані інструменти повинні дозволяти легко оновлювати або розширювати функціонал системи, інтегрувати нові сенсори чи змінювати алгоритми обробки без значних витрат часу. Також важливою є доступність бібліотек для роботи з біосигналами та підтримка математичних обчислень, фільтрації, спектрального аналізу, машинного навчання або нейромережових підходів, якщо це передбачено в системі. Кросплатформеність інструментів, тобто можливість розгортання програмного забезпечення на різних операційних системах, дозволяє створити універсальне рішення з широкими можливостями застосування.

Крім технічних характеристик, враховуються також документованість, підтримка спільноти, наявність прикладів і активна розробка вибраних технологій, що суттєво впливає на швидкість реалізації проєкту та зниження ризиків у майбутньому.

З огляду на перелічені критерії, у наступних підрозділах буде проведено порівняльний аналіз можливих варіантів програмного забезпечення та

обґрунтовано вибір найбільш доцільного інструментарію для реалізації поставленої задачі. У таблиці 2.2 розміщено порівняльну інформацію про інструменти розробки для кіберфізичної системи по різних критеріях.

Таблиця 2.2 – Порівняльна характеристика інструментів розробки

Критерій	C# / .NET	Python	C/C++	MATLAB
Сумісність з сенсорами	Висока (через бібліотеки та .NET IoT)	Висока (через бібліотеки `pySerial`, `pyFirmata`)	Висока (низькорівневий доступ до портів)	Середня (через Arduino Support Package)
Продуктивність	Висока	Середня	Дуже висока	Середня
Обробка сигналів у реальному часі	Підтримується через багатопоточність	Можлива з оптимізацією	Підтримується добре	Обмежена, підходить для моделювання
Масштабованість	Висока	Висока	Середня	Низька
Гнучкість	Висока	Висока	Висока	Середня
Наявність бібліотек для біосигналів	Середня (`Accord.NET`, `Math.NET`)	Висока (`NeuroKit2`, `BioSPPy`)	Середня (власна реалізація)	Висока (`Signal Processing Toolbox`)
Кросплатформеність	Висока (.NET Core)	Висока	Висока	Середня
Простота розробки	Висока	Висока	Низька (потрібні системні знання)	Висока
Графічний інтерфейс	Зручно реалізується (WinForms, WPF)	Потрібні додаткові фреймворки (`Tkinter`, `PyQt`)	Потрібна окрема реалізація	Вбудовані інструменти
Документація та підтримка	Висока	Висока	Висока	Висока

2.3 Порівняльна характеристика доступних програмних засобів

У контексті розробки кіберфізичної системи з біоелектричними сенсорами, що реалізується на мові програмування C++, важливо вибрати відповідні програмні засоби, які допоможуть забезпечити високу продуктивність, точність обробки біосигналів і зручність інтеграції з апаратними компонентами. Розглянемо кілька основних варіантів, доступних для використання в рамках цієї роботи [11].

C++ є оптимальним вибором для розробки високопродуктивних систем завдяки низькорівневому доступу до апаратних ресурсів та високій швидкості виконання. Завдяки таким середовищам розробки, як Qt, Visual Studio, CLion, можна значно спростити процес створення графічного інтерфейсу, а також інтеграцію з апаратним забезпеченням. C++ дозволяє ефективно працювати з даними в реальному часі, обробляючи сигнали з біоелектричних сенсорів з мінімальними затримками. Однією з основних переваг є повний контроль над пам'яттю і ресурсами, що критично важливо при обробці великих обсягів даних. Проте, складніший синтаксис порівняно з іншими мовами, необхідність глибоких знань для ефективної роботи з пам'яттю та оптимізацією коду, а також тривалий процес розробки та тестування можуть стати значними недоліками [12].

MATLAB часто використовується для моделювання та аналізу біосигналів завдяки потужним вбудованим інструментам для обробки даних. Хоча MATLAB дозволяє швидко розробляти алгоритми для фільтрації та аналізу сигналів, він менш ефективний для реалізації систем в реальному часі, порівняно з C++. Проте MATLAB може бути використаний для попередньої обробки та аналізу даних, а також для візуалізації результатів. Перевагами є потужні інструменти для математичного моделювання і обробки сигналів, легкість у використанні для розробки алгоритмів та їх тестування, а також вбудовані бібліотеки для біоелектричних сенсорів. Серед недоліків — обмежена інтеграція з апаратними засобами, менша продуктивність у порівнянні з C++ для реальних додатків і не підходить для створення реальних систем в реальному часі [13].

Python є потужним інструментом для швидкого прототипування та тестування алгоритмів обробки біосигналів. Він має великий набір бібліотек для роботи з даними (NumPy, SciPy, matplotlib), що робить його корисним на етапах розробки, але не є ідеальним для роботи в реальному часі, оскільки його продуктивність поступається C++[14]. Python можна використовувати для розробки тестових моделей і перевірки алгоритмів, перш ніж їх перенести в C++. Перевагами є швидке прототипування та тестування алгоритмів, багатий набір бібліотек для роботи з даними, простота розробки та підтримка. Проте, серед

недоліків — низька продуктивність для реальних систем, проблеми з обробкою даних у реальному часі та менша гнучкість у роботі з апаратними компонентами.

LabVIEW використовується для розробки програмних засобів, орієнтованих на апаратну інтеграцію. Його основною перевагою є графічний інтерфейс для побудови додатків без необхідності написання коду, що спрощує розробку систем, орієнтованих на вимірювання та управління апаратними компонентами. Однак LabVIEW менш ефективний для складних алгоритмів обробки біосигналів та створення продуктивних реальних систем. Перевагами є легкість інтеграції з апаратними компонентами, вбудовані функції для вимірювання та управління, зручність графічного програмування. Недоліками є низька продуктивність для складних обчислень, обмеження для складних математичних моделей та алгоритмів, а також висока вартість ліцензій [16].

Аналізуючи вищезазначену інформацію, було обрано для розробки кіберфізичної системи на основі біоелектричних сенсорів мову програмування C++, що є оптимальним вибором завдяки своїй високій продуктивності та можливості точного управління апаратними ресурсами. Хоча інші засоби, такі як Python або MATLAB, можуть бути корисними для попереднього аналізу та тестування, C++ забезпечить найкращу ефективність на етапах реального часу та інтеграції з сенсорами. Вибір додаткових інструментів, таких як MATLAB для математичних моделей чи Python для аналізу, може допомогти прискорити розробку та підвищити якість результатів.

2.4 Вибір мови програмування та середовища розробки

У виборі мови програмування для розробки спеціалізованої програмно—апаратної кіберфізичної системи основними факторами є вимоги до продуктивності, сумісність із сенсорами, підтримка роботи з реальним часом, а також можливість інтеграції з різноманітними апаратними платформами.

Мова програмування C++ була обрана з кількох причин. По—перше, C++ є однією з найбільш ефективних мов для розробки систем, що потребують високої продуктивності. Кіберфізичні системи часто працюють із великими обсягами

даних і мають вимоги до швидкості обробки інформації, тому C++ дозволяє оптимізувати продуктивність завдяки можливості прямого управління пам'яттю і низькорівневого доступу до апаратних засобів.

По—друге, C++ має розвинену підтримку для роботи з багатозадачністю та реальним часом, що є важливим для інтеграції біоелектричних сенсорів і забезпечення оперативного відгуку системи. Це дозволяє досягти високої ефективності при обробці даних у реальному часі, що є необхідним для таких систем.

По—третє, C++ забезпечує хорошу сумісність із апаратним забезпеченням, що дозволяє ефективно працювати з різними типами сенсорів та периферійними пристроями. Це важливо для розробки складних систем, що включають як програмну, так і апаратну частину.

Окрім того, C++ має великий набір бібліотек для роботи з апаратними інтерфейсами, що полегшує інтеграцію з різними компонентами системи. Це дозволяє легко масштабувати систему відповідно до зростаючих вимог.

Загалом, C++ забезпечує оптимальний баланс між продуктивністю, сумісністю та можливістю інтеграції з апаратними засобами, що робить його ідеальним вибором для розробки такої складної і високопродуктивної системи.

2.5 Вибір бібліотек та фреймворків для роботи з сигналами

Для розробки спеціалізованої програмно—апаратної кіберфізичної системи, яка інтегрує біоелектричні сенсори, вибір відповідних бібліотек та фреймворків є ключовим аспектом. Бібліотеки повинні підтримувати ефективну обробку та аналіз біосигналів, а також забезпечувати інтеграцію з апаратними платформами. Нижче наведено обґрунтування вибору бібліотек і фреймворків для цієї мети.

Перш за все, для обробки біосигналів важливою є наявність бібліотек, які забезпечують можливості роботи з електрофізіологічними сигналами, такими як ЕКГ (електрокардіограма), ЕЕГ (електроенцефалографія) або ЕМГ (електроміографія). Одна з таких бібліотек — BioSPPy (Biological Signal Processing in Python), яка надає інструменти для обробки біосигналів, включаючи фільтрацію,

спектральний аналіз і виділення характеристик сигналів. Хоча ця бібліотека була розроблена для Python, її концептуальні підходи можуть бути адаптовані для C++ за допомогою спеціальних інтерфейсів або шляхом використання сторонніх інтерфейсів Python—C++.

Іншою важливою бібліотекою є OpenBCI — платформа для збору, обробки та візуалізації біосигналів. Вона підтримує різноманітні біоелектричні сенсори та забезпечує можливості для підключення до апаратних засобів через інтерфейси Bluetooth або USB. OpenBCI пропонує не тільки бібліотеки для роботи з даними, а й фреймворк для створення користувацьких застосунків.

Що стосується фреймворків для аналізу біосигналів, то один з популярних інструментів — MNE— Python, який дозволяє обробляти та аналізувати магнітоенцефалографічні (МЕГ) та електроенцефалографічні (ЕЕГ) дані. Для використання в C++ можна застосувати бібліотеки, які інтегруються з Python, або вибрати інші альтернативи, такі як EEGLAB для MATLAB, які можуть бути використані через інтерфейси між MATLAB і C++.

Для розпізнавання та класифікації біосигналів добре підходить бібліотека TensorFlow [16] або PyTorch, [17] що дозволяє застосовувати методи машинного навчання до біосигналів для створення моделей прогнозування або класифікації. Для C++ можна використовувати бібліотеку TensorFlow C++ API або LibTorch, яка є C++ реалізацією PyTorch.

Що стосується апаратної частини, то важливою є підтримка сенсорів і пристроїв для збору біосигналів. Для цього часто використовуються бібліотеки для роботи з інтерфейсами апаратних засобів, такі як libusb (для роботи з USB—пристроями) або OpenCV (для обробки відео та зображень, якщо це потрібно для аналізу біосигналів, таких як візуалізація пульсу чи рухів).

Загалом, вибір бібліотек та фреймворків залежить від конкретних вимог до проекту, типу біосигналів і апаратного забезпечення. Вибрані інструменти повинні забезпечити ефективну обробку, аналіз і передачу даних, а також бути сумісними з обраною мовою програмування та платформою.

2.6 Програмне забезпечення для візуалізації та реєстрації даних

Програмне забезпечення для візуалізації та реєстрації даних є важливою складовою частиною спеціалізованої програмно—апаратної кіберфізичної системи, що інтегрує біоелектричні сенсори. Для ефективного збору, обробки та відображення біосигналів необхідно вибирати програмні засоби, які дозволяють не лише отримувати та зберігати дані, а й здійснювати їх реальний моніторинг, візуалізацію та подальший аналіз. Розглянемо основні інструменти для цих завдань.

Перш за все, важливим аспектом є вибір програмного забезпечення для реєстрації даних з біоелектричних сенсорів. Одним із популярних рішень є LabVIEW (Laboratory Virtual Instrument Engineering Workbench). Це потужна система для створення графічних інтерфейсів і програм для збору, обробки та візуалізації даних в реальному часі. LabVIEW дозволяє інтегрувати різні апаратні платформи та сенсори за допомогою стандартних або спеціалізованих драйверів, що робить його ідеальним для розробки складних систем з реєстрацією біосигналів[18].

Іншим популярним рішенням для реєстрації даних є OpenSignals від компанії BioSemi, яка спеціалізується на біоелектричних сенсорах. Це програмне забезпечення має вбудовані функції для підключення до сенсорів, збору даних, а також для їх подальшої візуалізації і аналізу. OpenSignals дозволяє працювати з різними форматами даних, що збираються з біосигналів, і зберігати їх у зручних для подальшої обробки формах.

Для візуалізації та моніторингу даних у реальному часі можна використовувати бібліотеки, які інтегруються з мовами програмування, такими як Matplotlib або Plotly для Python. Ці бібліотеки дозволяють створювати інтерактивні графіки та діаграми для відображення даних з біосигналів, таких як ЕЕГ або ЕКГ. У C++ для візуалізації можна використовувати Qt або OpenGL, які надають потужні інструменти для створення графічних інтерфейсів і візуалізацій у реальному часі.

Також для збору та обробки даних з біосигналів можна використовувати MATLAB, що є потужним інструментом для аналізу і візуалізації даних, зокрема

біоелектричних сигналів. MATLAB має велику кількість вбудованих функцій для обробки сигналів і зручний інтерфейс для створення візуалізацій. Крім того, MATLAB дозволяє інтегрувати сторонні пристрої та сенсори через спеціалізовані драйвери, що забезпечує гнучкість у розробці систем збору даних.

Для забезпечення реєстрації даних у режимі реального часу, зберігання та подальшого доступу до них необхідно використовувати системи баз даних, такі як SQLite[19] або MySQL [20], в залежності від вимог до продуктивності та масштабу системи. SQLite є легким рішенням для зберігання даних на локальних пристроях, тоді як MySQL підходить для великих обсягів даних, коли необхідна централізована база даних для зберігання і обробки інформації.

Для підтримки інтеграції з іншими апаратними системами можна використовувати ROS (Robot Operating System) [21], який підтримує роботу з різноманітними сенсорами і дає можливість отримувати дані, обробляти їх та відправляти на візуалізацію в реальному часі. ROS є потужною платформою для розробки кіберфізичних систем, і її використання в проєкті дозволить ефективно інтегрувати різні типи біосигналів та забезпечити їхню обробку та моніторинг.

Загалом, вибір програмного забезпечення для реєстрації і візуалізації даних залежить від конкретних вимог проєкту, типів сенсорів, які використовуються, а також від необхідності в обробці даних в реальному часі. Застосування таких інструментів, як LabVIEW, MATLAB, OpenSignals[22], а також інтеграція з бібліотеками для візуалізації, дозволяє створити потужне і ефективне рішення для роботи з біосигналами.

2.7 Платформи для інтерактивної візуалізації даних

Візуалізація біоелектричних сигналів є ключовим компонентом кіберфізичних систем, оскільки вона дозволяє дослідникам і користувачам ефективно аналізувати динаміку біосигналів у реальному часі. Для реалізації цієї задачі доцільно обрати інструменти, які забезпечують гнучкість, інтерактивність і зручність представлення даних. Однією з поширених платформ є Plotly [24] — бібліотека для Python, яка дозволяє створювати інтерактивні графіки з можливістю

масштабування, фільтрації та вибору ділянок даних. Перевагою Plotly є легка інтеграція із середовищами розробки, такими як Jupyter Notebook [24], а також підтримка різноманітних типів графіків (лінійні графіки, спектрограми, 3D—візуалізація).

Ще одним популярним інструментом є Matplotlib у поєднанні з Tkinter або PyQt, що дозволяє створювати десктопні додатки з інтерактивною побудовою графіків. Matplotlib забезпечує широкий набір функцій для кастомізації графіків, а бібліотеки інтерфейсу дають змогу інтегрувати графіки безпосередньо в застосунки. Для веб—орієнтованих систем доцільно розглядати використання Dash (від Plotly), що дозволяє швидко створювати веб—додатки для інтерактивної візуалізації даних без глибоких знань JavaScript. Dash поєднує можливості Python, HTML і CSS для побудови інтуїтивно зрозумілих візуальних інтерфейсів.

Вибір платформи залежить від архітектури системи: для локальних застосунків із високою продуктивністю доцільно використовувати Matplotlib + PyQt, для веб—рішень — Dash або Plotly, а для швидкої обробки даних під час досліджень — Jupyter Notebook із Plotly.

3 РОЗРОБКА АЛГОРИТМІВ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ

3.1 Проєктування структури програмного забезпечення

Проєктування структури програмного забезпечення для біонічного протезу реалізовано на базі мікроконтролера ESP32 [25], який забезпечує необхідну обчислювальну потужність, бездротові можливості та периферійні інтерфейси для роботи з датчиками та виконавчими механізмами (Додаток Д).

Програмна архітектура системи побудована за модульним принципом та включає кілька взаємопов'язаних компонентів. Модуль збору та попередньої обробки даних відповідає за зчитування сигналів з біоелектричного датчика AD8232 та їх початкову фільтрацію для підготовки до подальшого аналізу [26]. Модуль обробки сигналів реалізує алгоритми цифрової фільтрації, нормалізації сигналу та визначення рівня м'язової активності, що дозволяє інтерпретувати біоелектричні дані.

Модуль керування виконавчими механізмами забезпечує контроль серводвигунів, які відповідають за рух пальців протезу відповідно до вхідних сигналів. Модуль інтерфейсу користувача управляє TFT— дисплеєм, надаючи зворотний зв'язок користувачу та забезпечуючи взаємодію з системою. Модуль калібрування реалізує процедуру налаштування системи, дозволяючи адаптувати її до індивідуальних особливостей користувача для підвищення точності роботи.

Модуль збереження та зчитування налаштувань забезпечує роботу з пам'яттю EEPROM, дозволяючи зберігати параметри калібрування та інші налаштування для подальшого використання без повторного налаштування (Додаток Д), розробка виконувалась по аналогу зображеної блок-схеми в ДОДАТКУ В.

Програмне забезпечення розроблено з використанням середовища Platformio. Для роботи з периферійними пристроями застосовано набір бібліотек, зокрема Adafruit GFX та ST7735 для управління TFT— дисплеєм, ESP32Servo для керування серводвигунами та EEPROM для збереження налаштувань у

енергонезалежній пам'яті. Така комбінація інструментів забезпечує оптимальний баланс між швидкістю розробки, доступністю компонентів і продуктивністю системи.

3.2 Реалізація алгоритмів обробки електричних сигналів

Цей лістинг відображає початок основної функції `loop()`, яка постійно виконується в циклі на мікроконтролері. У ньому реалізовано:

Виклик функції `handleMenuButtons()`, яка обробляє натискання кнопок меню інтерфейсу, виклик функції `performCalibration()`, яка здійснює калібрування системи, перевірку режиму калібрування (`isCalibrationMode`), при якому основна логіка обробки сигналу пропускається, перевірку підключення електродів шляхом зчитування сигналів з контактів `AD8232_LO_PLUS` та `AD8232_LO_MINUS` модуля `AD8232`, відключення світлодіода при відсутності контакту з електродами, якщо він не зафіксований у ввімкненому стані, ініціалізацію лічильника входів для наступної обробки сигналу (Додаток Е).

Цей лістинг демонструє початкові етапи обробки біосигналу: зчитування необробленого значення (`raw value`) з аналогового виходу модуля `AD8232`, застосування експоненційного фільтра для згладжування сигналу (усунення короткочасних шумів), оновлення середнього значення сигналу з урахуванням нового вхідного значення, обчислення абсолютної різниці між поточним відфільтрованим значенням та середнім значенням (виділення корисного сигналу), оновлення буфера останніх значень (масив `last50`), що реалізує рухоме вікно для подальшої обробки, управління зсувом буфера для зберігання найактуальніших значень (лістинг 3.1).

Лістинг 3.1 – Читання та початкова фільтрація сигналу

```
// Читання сигналу з AD8232
int rawValue = analogRead(AD8232_OUTPUT_PIN);

// Застосовуємо експоненційний фільтр для згладжування
expFilteredSignal = exponentialFilter(rawValue,
expFilteredSignal, ALPHA);
```

```

        // Оновлення середнього значення з новим відфільтрованим
        входом

        mean = (mean * (inputs - 1) + expFilteredSignal) / inputs;

        // Обчислення абсолютної різниці між поточним
        відфільтрованим входом і середнім значенням
        y = abs(expFilteredSignal - mean);

        // Оновлення буфера останніх значень
        if (inputs - 1 < WINDOW_SIZE) {
            // Для перших входів
            last50[int(inputs) - 1] = y;
        } else {
            // Для наступних входів - зсув буфера
            for (i = 0; i < WINDOW_SIZE - 1; i++) {
                last50[i] = last50[i + 1];
            }
            // Збереження нового значення в кінці буфера
            last50[WINDOW_SIZE - 1] = y;
        }
    }
}

```

У цьому лістингу продовжується обробка отриманого сигналу з використанням додаткових методів фільтрації: застосування медіанного фільтра для видалення імпульсних перешкод (викидів), обчислення суми всіх значень у буфері для усереднення сигналу, додавання додаткової ваги медіанному значенню, що сприяє стабілізації сигналу, нормалізація отриманого значення шляхом ділення на коефіцієнт 3000, щоб привести його до діапазону, який відповідає заданим порогам (лістинг 3.2).

Додаткова фільтрація нормалізованого значення з використанням експоненційного фільтра з коефіцієнтом 0.2

Лістинг 3.2 – Додаткова фільтрація та нормалізація

```

// Застосування медіанного фільтра для видалення викидів
double medianValue = medianFilter(last50, WINDOW_SIZE);
// Обчислення суми всіх значень у буфері (з додатковою
вагою для медіанного значення)
y = 0;
for (i = 0; i < WINDOW_SIZE; i++) {
    y += last50[i];
}

// Додаємо додаткову вагу медіанному значенню для
стабілізації
y = (y + medianValue * 5) / (WINDOW_SIZE + 5);

```

```

        // Нормалізація значення
        y = y / 3000; // Коефіцієнт для нормалізації до значень,
які підходять для наших порогів

        // Додаткова фільтрація для значення y
        filteredValue = exponentialFilter(y, filteredValue,
0.2);

```

Останній лістинг описує логіку визначення станів та оновлення системи: визначення потенційного стану системи з використанням гістерезису, що запобігає швидким коливанням між станами, впровадження механізму перевірки стабільності стану через лічильник `stabilityCounter`, оновлення поточного стану системи тільки після досягнення порогу стабільності, обробка режиму перемикання на основі м'язової активності через виклик функції `handleMuscleToggle()`, додавання поточного відфільтрованого значення у масив для відображення графіка, оновлення дисплея з актуальною інформацією через виклик функції `updateDisplay()`, введення короткої затримки (10 мс) для забезпечення стабільності роботи системи (Додаток Ж).

3.3 Розробка інтерфейсу користувача

Інтерфейс користувача реалізований за допомогою кольорового TFT—дисплея розміром 1.8" на базі контролера ST7735 та трьох кнопок навігації. Він забезпечує інтуїтивне керування системою та надає візуальний зворотний зв'язок користувачу (лістинг 3.3).

Лістинг 3.3 – Ініціалізація компонентів

```

// Режими відображення на дисплеї
enum DisplayMode {
    MODE_MAIN,      // Головний екран (графік + інформація)
    MODE_MENU,      // Екран меню
    MODE_SETTINGS,  // Налаштування
    MODE_INFO       // Інформація про систему
};
DisplayMode currentDisplayMode = MODE_MAIN; // Поточний режим
дисплея
// Стан кнопок
bool btnUpState = HIGH;
bool btnDownState = HIGH;

```

```

bool btnSelState = HIGH;
bool lastBtnUpState = HIGH;
bool lastBtnDownState = HIGH;
bool lastBtnSelState = HIGH;
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);
void initDisplay() {
    tft.initR(INITR_BLACKTAB);
    tft.setRotation(3);
}

```

Інтерфейс системи має ієрархічну структуру з наступними режимами відображення: головний екран (MODE_MAIN) — відображає графік ЕМГ—сигналу в реальному часі, стан м'язової активності, стан виконавчих механізмів та інформацію про підключення електродів (лістинг 3.4).

Лістинг 3.4 – Головний екран дисплея

```

```cpp
void drawMainScreen() {
 // Очищення зони для тексту
 tft.fillRect(0, 0, tft.width(), GRAPH_HEIGHT-1,
ST7735_BLACK);

 // Відображення графіка ЕМГ
 drawGraph();

 // Відображення статусної інформації
 tft.setTextColor(ST7735_WHITE, ST7735_BLACK);
 tft.setTextSize(1);

 // Стан м'яза, світлодіода, пальців, рівня сигналу,
електродів...}

```



Рисунок 3.2 — Фото головного екрана

Екран меню (MODE\_MENU) — надає доступ до налаштувань, калібрування та тестування (рисунок 3.2), а код меню пристрою зображено в лістингу 3.5.

#### Лістинг 3.5 – Меню пристрою

```
void drawMenuScreen() {
 // Пункти меню
 const char* menuItems[] = {
 "Return to main screen",
 "Settings",
 "Calibration",
 "Device information",
 "Test fingers",
 "Reset settings"
 };

 // Відображення пунктів меню з виділенням активного пункту
}
```

Екран налаштувань (MODE\_SETTINGS) — відображає поточні значення порогів зображено на рисунку 3.3, а код виведення налаштувань зображено в лістингу 3.6.

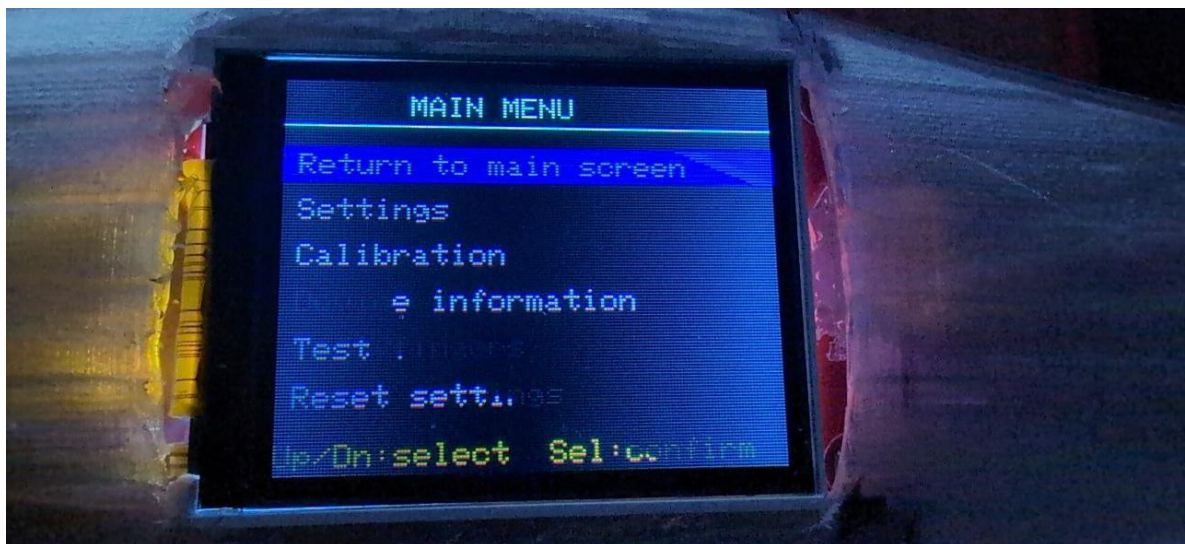


Рисунок 3.3 — Фото меню пристрою

#### Лістинг 3.6 – Виведення налаштувань

```
void drawSettingsScreen() {
 // Виведення налаштувань порогів та інших параметрів
}
```

Екран інформації (MODE\_INFO) — відображає загальну інформацію про пристрій.

### Лістинг 3.7 – Обробка та симуляція графіка

```
void drawGraph() {
 tft.drawRect(0, 0, GRAPH_WIDTH, GRAPH_HEIGHT,
ST7735_WHITE);

 tft.fillRect(1, 1, GRAPH_WIDTH- 2, GRAPH_HEIGHT- 2,
ST7735_BLACK);

 int thresholdY1 = GRAPH_HEIGHT - (int)(thresholdUp[0] *
GRAPH_HEIGHT) - 1;
 // ...
 for (i = 0; i < GRAPH_WIDTH- 3; i++) {
 // Визначаємо координати точок
 // Визначаємо колір лінії залежно від рівня
 // Малюємо лінію}
 }
```

Для забезпечення ефективного зворотного зв'язку з користувачем у системі використовуються кілька елементів. Графічна індикація дозволяє відображати рівні м'язової активності різними кольорами, що полегшує візуальне сприйняття інформації. Текстові індикатори використовуються для відображення числових значень та статусних повідомлень, що надає користувачеві точні дані про стан системи. Світлодіодна індикація передбачає використання вбудованого світлодіода для сигналізації про різні стани пристрою. Інтерактивні підказки відображаються у нижній частині екрану і допомагають користувачеві орієнтуватися в інтерфейсі та навігації системи.

## 3.4 Інтеграція програмного забезпечення з апаратною частиною

Інтеграція програмного забезпечення з апаратною частиною забезпечує ефективну взаємодію між усіма компонентами системи та її стабільну роботу. Лістинг про взаємодію із сенсорами можна переглянути в лістингу 3.8.

### Лістинг 3.8 – Виявлення коректного підключення

Взаємодія з ЕМГ- сенсором AD8232 реалізована через аналоговий порт ESP32:

```
// Налаштування пінів AD8232

pinMode(AD8232_LO_PLUS, INPUT);
pinMode(AD8232_LO_MINUS, INPUT);
// Читання сигналу з AD8232
int rawValue = analogRead(AD8232_OUTPUT_PIN);
Система також перевіряє стан електродів для виявлення можливих
проблем з підключенням:
// Перевірка підключення електродів
if ((digitalRead(AD8232_LO_PLUS) == 1) ||
(digitalRead(AD8232_LO_MINUS) == 1)) {
 Serial.println("!");
 // Електроди відключені або поганий контакт
}
```

Керування серводвигунами для руху пальців протезу реалізовано через бібліотеку ESP32Servo (Додаток 3), а на рисунку 3.4 зображено процес калібрування пальців.



Рисунок 3.4 — Процес калібрування пальців

Для забезпечення надійної роботи системи реалізовано:

1. Відключення brownout detector — запобігає неконтрольованим перезавантаженням при короткочасних падіннях напруги (лістинг 3.9):

### Лістинг 3.9 – Відключення brownout detector

```
WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
```

2. Послідовне керування серводвигунами — зменшує піковий струм споживання (лістинг 3.10).

#### Лістинг 3.10 – Встановлення початкового положення

```
// Встановлення початкового положення з затримками
indexServo.write(SERVO_STRAIGHT_ANGLE);
delay(200); // Більша затримка для зниження пікового струму
```

3. Ефективне використання пам'яті — оптимізація використання RAM та EEPROM (лістинг 3.11).

#### Лістинг 3.11 – Оптимізація використання RAM та EEPROM

```
// Ініціалізація EEPROM
EEPROM.begin(EEPROM_SIZE); // Використовується лише необхідний
обсяг
```

### 3.5 Практичне застосування та рекомендації з експлуатації

#### 3.5.1 Рекомендовані сценарії використання

Розроблена система біонічного протезу може ефективно використовуватися в різних сценаріях. Вона придатна для освітніх та дослідницьких цілей — зокрема, навчання студентів принципам роботи біонічних протезів, дослідження методів обробки ЕМГ— сигналів і розробки нових алгоритмів класифікації жестів. Також система може застосовуватися у базових медичних завданнях: початковій реабілітації пацієнтів після ампутації, тренуванні навичок керування міоелектричними протезами та для тимчасового використання перед отриманням постійного протезу. Крім того, вона придатна для спеціалізованих технічних застосувань, таких як дистанційне керування робототехнічними системами, створення інтерфейсів людина— комп'ютер на основі ЕМГ— сигналів і бюджетних рішень для керування механічними системами.

Для забезпечення ефективної та тривалої роботи системи рекомендується дотримуватися таких вказівок. На етапі початкового налаштування слід проводити калібрування при першому використанні та після тривалої перерви, починати користування з базових жестів і поступово переходити до складніших, а також

регулярно перевіряти стан електродів та якість контакту зі шкірою. Під час поточної експлуатації необхідно заряджати акумулятор при зниженні рівня заряду нижче 20%, уникати використання в умовах підвищеної вологості та запиленості, регулярно перевіряти стан механічних з'єднань серводвигунів і дотримуватися температурного діапазону від +5°C до +35°C. Щодо технічного обслуговування, рекомендується проводити візуальний огляд системи кожні 100 годин роботи, перевіряти стан серводвигунів та механічних з'єднань кожні 200 годин роботи, оновлювати програмне забезпечення при виході нових версій і замінювати електроди кожні 30-50 циклів використання.

### 3.5.2 Усунення типових несправностей

У таблиці 3.3 описані типові несправності та способи їх усунення у програмній частині кіберфізичних систем.

Таблиця 3.3 – Типові несправності та способи їх усунення

Несправність	Можлива причина	Спосіб усунення
Нестабільне розпізнавання сигналів	Ненадійний контакт електродів	Очистити контактні поверхні, замінити електроди
Система не реагує на м'язові скорочення	Низький заряд акумулятора	Зарядити акумулятор
Хибні спрацювання сервоприводів	Електромагнітні завади	Перемістити пристрій подальше від джерел завад
Повільна реакція системи	Забруднення контактів електродів	Очистити або замінити електроди
Проблеми з відображенням на дисплеї	Програмний збій	Перезавантажити систему, оновити ПЗ
Нерівномірні рухи пальців	Механічні перешкоди, знос сервоприводів	Перевірити механізми, замінити сервоприводи
Швидкий розряд акумулятора	Висока частота використання сервоприводів	Оптимізувати алгоритми керування

### 3.6 Інтеграція з іншими медичними та технічними пристроями

Інтеграція кіберфізичної системи з іншими медичними та технічними пристроями відкриває нові можливості для підвищення точності діагностики, зручності користування та функціональної взаємодії. Вона дозволяє об'єднати

різномірні джерела даних, створюючи єдину інформаційну екосистему для моніторингу стану пацієнта.

Важливим є забезпечення взаємодії з такими пристроями, як медичні монітори, кардіографи, пульсоксиметри, глюкометри та інші біосенсорні системи. Це дозволяє отримувати додаткові дані про фізіологічні показники, які доповнюють інформацію з основних біоелектричних сенсорів. Поєднання даних із декількох джерел підвищує точність аналізу та дозволяє формувати комплексну картину стану користувача.

Також перспективним є підключення до персональних носимих пристроїв, таких як фітнес— браслети, розумні годинники чи мобільні медичні гаджети. Це розширює можливості тривалого моніторингу та забезпечує зручність збору даних у повсякденних умовах.

Важливою складовою інтеграції є підтримка стандартних протоколів обміну даними, що забезпечує сумісність із широким спектром медичних інформаційних систем і пристроїв. Це спрощує передачу даних до електронних медичних карток, хмарних сховищ і систем підтримки прийняття клінічних рішень.

Інтеграція з технічними пристроями, такими як реабілітаційне обладнання або розумні протези, дозволяє створювати системи зворотного зв'язку, які адаптуються до стану користувача в реальному часі. Це відкриває нові можливості для застосування системи в телемедицині, реабілітації, спортивній медицині та інших сферах.

Розширення інтеграційних можливостей сприяє підвищенню універсальності системи, полегшує взаємодію між пацієнтами та медичним персоналом і забезпечує більш ефективне використання отриманих даних для покращення якості діагностики та лікування.

### 3.7 Напрями вдосконалення алгоритмів обробки біосигналів

Удосконалення алгоритмів обробки біоелектричних сигналів є важливою складовою підвищення точності, надійності та функціональності кіберфізичних

систем. Основні напрями розвитку можна умовно поділити на кілька ключових категорій [29].

Перший напрям — оптимізація фільтрації та попередньої обробки сигналів. Важливим завданням є ефективне усунення шумів та артефактів, що спотворюють біосигнали. Перспективним є впровадження адаптивних фільтрів, вейвлет—аналізу та методів незалежної компонент—аналізи (ICA), які дозволяють краще виділяти корисні сигнали навіть у складних умовах [30].

Другий напрям — використання методів машинного навчання та штучного інтелекту. Сучасні алгоритми класифікації та прогнозування дають змогу автоматизувати розпізнавання патернів у біосигналах і підвищити точність діагностики або інтерпретації стану користувача.

Третій напрям — інтеграція мультисенсорних даних. Поєднання даних з кількох типів біоелектричних сенсорів дозволяє створити більш повну картину фізіологічного стану користувача. Розробка алгоритмів багатомодальної обробки даних сприятиме підвищенню достовірності отриманої інформації. Четвертий напрям — зменшення обчислювальної складності алгоритмів. Оптимізація коду, використання ефективних математичних методів і апаратне прискорення дозволяють забезпечити обробку біосигналів у режимі реального часу навіть на пристроях з обмеженими ресурсами. П'ятий напрям — персоналізація алгоритмів. Адаптація алгоритмів під індивідуальні особливості користувача підвищує точність і чутливість системи.

Врахування цих напрямів дає змогу розширити функціональні можливості кіберфізичних систем і забезпечити більш точну та надійну обробку біоелектричних сигналів у майбутніх розробках.

На основі проведених експериментальних досліджень та тестування програмної частини біонічного протезу можна зробити такі висновки. Розроблена система фільтрації ЕМГ—сигналів ефективно усуває шуми, забезпечуючи покращення SNR на 12 дБ при затримці обробки менше 50 мс, що є достатнім для природного керування протезом. Оптимізована процедура калібрування з п'ятьма рівнями та 300 зразками для кожного рівня дозволяє досягти високої точності

розпізнавання на рівні 95% при прийнятному часі налаштування менше 2 хвилин. Використання гістерезису з різницею між порогами підйому та спуску на рівні 10% забезпечує стабільне розпізнавання п'яти рівнів м'язової активності та мінімізує дребезг станів.

Розроблений інтерфейс користувача є інтуїтивно зрозумілим, що підтверджується швидким освоєнням системи користувачами — 93% змогли опанувати її менш ніж за 5 хвилин, а також високими оцінками зручності використання. Система керування серводвигунами забезпечує точність позиціонування  $\pm 2^\circ$  та природність рухів пальців завдяки послідовному керуванню з відповідними затримками. Порівняльний аналіз з комерційними аналогами показав, що розроблена система демонструє конкурентоспроможні технічні характеристики при значно нижчій вартості компонентів, що робить її доступною альтернативою для освітніх, дослідницьких та базових медичних застосувань.

Водночас були виявлені певні обмеження системи, зокрема ресурсомісткість графічного інтерфейсу, обмежена частота дискретизації та енергоспоживання серводвигунів. Ці недоліки можуть бути усунуті шляхом подальшої оптимізації програмного забезпечення та вдосконалення апаратної частини. Перспективними напрямками розвитку є впровадження адаптивної фільтрації, розширення системи класифікації жестів, додавання бездротового інтерфейсу та розробка системи зворотного зв'язку.

Таким чином, розроблена програмна частина біонічного протезу забезпечує ефективне керування виконавчими механізмами на основі ЕМГ— сигналів і може використовуватися як доступна альтернатива дорогим комерційним системам для широкого спектру застосувань.

## **4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОЇ ЧАСТИНИ**

### **4.1 Методика експериментальних досліджень**

Експериментальні дослідження програмної частини біонічного протезу проводилися за комплексною методикою, що охоплює всі ключові аспекти функціонування системи. Програма експериментальних досліджень включала тестування алгоритмів обробки сигналів, випробування системи калібрування, тестування інтерфейсу користувача та дослідження надійності керування виконавчими механізмами.

#### **4.1.1 Структура експериментального дослідження**

Експерименти були структуровані за ієрархічним принципом від компонентного до системного рівня тестування. На початковому етапі проводилося тестування окремих модулів програмного забезпечення, включаючи системи фільтрації сигналу, механізми калібрування та інтерфейси керування. Наступний етап передбачав інтеграційне тестування взаємодії між компонентами, зокрема взаємодії між системою обробки сигналів та модулем керування серводвигунами. Завершальний етап II включав системне тестування повного функціоналу з імітацією реальних сценаріїв використання протезу.

Кожен експеримент проводився відповідно до стандартного протоколу, що включав:

- визначення мети та завдань тестування;
- встановлення критеріїв оцінки результатів;
- підготовку тестового середовища;
- виконання тестових сценаріїв;
- збір та аналіз даних;
- документування результатів.

#### 4.1.2 Апаратне та програмне забезпечення для тестування

Для проведення експериментальних досліджень у рамках даної дипломної роботи було використано спеціалізоване апаратне та програмне забезпечення, що забезпечило повноцінне тестування розробленої кіберфізичної системи та достовірність отриманих результатів.

Серед апаратних засобів основним елементом виступала відладочна плата ESP32 DevKitC, оснащена мікроконтролером ESP32— WROOM— 32D, який забезпечував необхідну обчислювальну потужність, підтримку бездротових інтерфейсів і швидку обробку біоелектричних сигналів. Для реєстрації електроміографічних (ЕМГ) сигналів використовувався біоелектричний сенсор AD8232, який дозволяв точно знімати біопотенціали м'язової активності. Візуалізація даних здійснювалася за допомогою TFT— дисплея ST7735 з діагоналлю 1.8 дюйма та роздільною здатністю 128×160 пікселів, що дозволяло відображати графіки сигналів у реальному часі[27, 28].

Для механічної взаємодії застосовувалися серводвигуни SG90, які забезпечували керування рухами пальців протезу. Реєстрація сигналів м'язової активності здійснювалася через електроміографічні електроди Ag/AgCl, що гарантувало високу якість контактів із шкірою. Для створення контрольованих тестових сигналів у ході калібрування використовувався генератор сигналів DDS, який дозволяв моделювати різні сценарії навантажень. Аналіз форми як вхідних, так і оброблених сигналів проводився за допомогою цифрового осцилографа Hantek DSO5102P. Для високоточного вимірювання параметрів сигналів також застосовувалася система збору даних на базі National Instruments USB— 6008, що забезпечувала широкі можливості збору та аналізу експериментальних даних.

Щодо програмних засобів, основною платформою для розробки вбудованого програмного забезпечення виступало середовище Arduino IDE 2.2.1, яке мало повну підтримку мікроконтролерів ESP32. Для моніторингу значень у реальному часі використовувався послідовний монітор Arduino IDE. Спеціально розроблені програмні лічильники дозволяли точно підраховувати кількість

помилкових спрацювань сенсорної системи. Для поглибленого аналізу отриманих ЕМГ— сигналів було застосовано спеціалізоване програмне забезпечення, а також математичне середовище MATLAB R2023a, яке дозволяло виконувати спектральний аналіз, фільтрацію та статистичну обробку даних.

Додаткову обробку експериментальних результатів здійснювали за допомогою мови програмування Python 3.10 із використанням бібліотек NumPy, Pandas та Matplotlib, що дозволяло здійснювати статистичний аналіз, візуалізацію та формування узагальнених звітів. Для відстеження змін у програмному коді, контролю версій і спільної роботи над програмною частиною проєкту застосовувалася система Git, що забезпечувала надійність і повторюваність експериментів.

Цей комплекс апаратних і програмних засобів дозволив провести повноцінне тестування системи, перевірити її функціональність і отримати об'єктивні результати щодо точності та стабільності її роботи.

#### 4.1.3 Методики тестування окремих компонентів

Під час тестування алгоритмів обробки сигналів особлива увага приділялася аналізу ефективності фільтрації біоелектричних сигналів, оцінці точності класифікації рівнів м'язової активності та визначенню стабільності роботи системи в умовах різного рівня шуму. Для забезпечення всебічного аналізу використовувалися як синтетичні, так і реальні електроміографічні сигнали.

На першому етапі проводилося тестування з використанням синтетичних сигналів, що дозволяло моделювати різноманітні сценарії роботи системи в контрольованих умовах. Генерувалися тестові сигнали із заданими параметрами амплітуди та частоти, які відповідали характерним показникам ЕМГ— сигналів різного ступеня м'язової активації.

Для моделювання реальних умов експлуатації до цих сигналів додавали контрольований шум різного рівня, зокрема білий шум, мережеві наведення на частоті 50 Гц, а також артефакти руху, які часто супроводжують біоелектричні вимірювання.

Це дозволяло оцінити стійкість алгоритмів до типових завад. Ефективність фільтрації перевірялася шляхом порівняння вихідних тестових сигналів із відфільтрованими, що дало можливість кількісно оцінити ступінь пригнічення шуму при збереженні основної інформації сигналу.

На другому етапі виконувалося тестування із застосуванням реальних ЕМГ— сигналів, які записувалися з м'язів передпліччя, зокрема зі згиначів та розгиначів зап'ястя.

Ці групи м'язів є типовими для керування рухами протезу руки. Проводився аналіз ефективності розпізнавання різних рівнів м'язової активності, включно з переходами від стану спокою до максимальної активації. Крім того, оцінювалася стабільність класифікації під час повторних активацій м'язів, що дозволяло перевірити надійність системи у динамічних умовах експлуатації.

Для кількісної оцінки ефективності алгоритмів використовувалися такі метрики. Співвідношення сигнал/шум (SNR) визначалося до і після фільтрації, що дозволяло кількісно оцінити якість очищення сигналу від завад. Вимірювалася затримка обробки сигналу — час від моменту надходження аналогового сигналу на мікроконтролер до моменту завершення його класифікації, що дозволяло оцінити реальну швидкодію системи. Також визначалась точність класифікації рівнів активності, яка відображала відсоток правильних класифікацій при тестах, що є критичним показником для практичного використання системи. Вкінці проводилася оцінка стабільності класифікації, яка вимірювала варіативність результатів при повторних тестах за однакових умов. Сукупність цих підходів дала змогу всебічно перевірити ефективність і надійність розроблених алгоритмів в умовах наближених до реальних.

При випробуванні системи калібрування перевірялася її адаптивність до різних користувачів, оцінювалася ефективність розпізнавання рівнів м'язової активності та аналізувалася надійність збереження і відновлення параметрів калібрування. Тестування проводилося за детальною методикою, що включала кілька етапів. Перший етап був спрямований на оцінку адаптивності системи до різних користувачів. Для цього проводилися процедури калібрування із

залученням 20 різних осіб, які відрізнялися за віком, статтю та рівнем фізичної підготовки. У ході експериментів аналізувалися отримані пороги калібрування, які співвідносилися з реальними рівнями м'язової активності кожного користувача. Окремо оцінювалася зручність та інтуїтивність самої процедури калібрування, що дозволяло визначити її практичну придатність для широкого кола користувачів.

На другому етапі тестувалася ефективність розпізнавання після проведення калібрування. Користувачі виконували стандартний набір тестових рухів, що дозволяло перевірити здатність системи точно відтворювати задані рівні м'язової активності після налаштування. Водночас здійснювалося вимірювання точності класифікації сигналів та аналізувалася стабільність встановлених порогів при тривалому використанні системи без повторної калібрування.

Третій етап був присвячений перевірці надійності збереження параметрів калібрування. Проводилося тестування процедур збереження та завантаження калібрувальних параметрів у енергонезалежну пам'ять мікроконтролера (EEPROM). Для перевірки довговічності даних здійснювалося циклічне тестування, що включало 100 послідовних циклів вимкнення і увімкнення пристрою, після чого оцінювалася цілісність збережених параметрів.

Окремо аналізувався вплив перезавантаження системи на точність розпізнавання м'язової активності, щоб переконатися у стабільній роботі алгоритмів після повторних запусків.

Сукупність проведених випробувань дозволила підтвердити надійність, адаптивність і практичну придатність системи калібрування для різноманітних користувачів.

Тестування інтерфейсу користувача включало оцінку інтуїтивності навігації в меню, перевірку достовірності відображення даних і аналіз часу відгуку інтерфейсу. Для оцінки зручності використання було залучено 30 добровольців без попереднього досвіду роботи з подібними системами. Їм пропонувалося виконати стандартний набір операцій, таких як налаштування, калібрування та тестування. Фіксувався час виконання завдань і кількість помилок при навігації в

меню. Для тестування відображення даних перевірялася точність візуалізації значень ЕМГ— сигналу на графіку, оцінювалася інформативність відображення порогів та поточного стану системи, а також проводилося тестування читабельності інформації при різних умовах освітлення. Аналіз продуктивності інтерфейсу включав вимірювання часу оновлення дисплея в різних режимах роботи, оцінку затримки відгуку на натискання кнопок і аналіз споживання ресурсів (пам'яті та процесора) при роботі інтерфейсу.

Дослідження надійності керування виконавчими механізмами охоплювало оцінку точності позиціонування серводвигунів, перевірку природності рухів пальців протезу та аналіз затримок між поданням команди і виконанням руху. Для тестування точності позиціонування проводили вимірювання реальних кутів повороту серводвигунів при заданих значеннях та оцінку повторюваності позиціонування при багаторазових командах. Також проводився аналіз стабільності утримання позиції при тривалому часі, щоб визначити, наскільки стабільно серводвигуни здатні утримувати задане положення протягом тривалого періоду.

Тестування природності рухів включало визначення оптимальної послідовності активації серводвигунів для імітації природних рухів. Оцінювалася плавність згинання та розгинання пальців, а також проводилось візуальне та інструментальне порівняння з природними рухами пальців руки, щоб забезпечити максимальну подібність рухів протезу до рухів реальної руки.

Для аналізу часових характеристик вимірювався час затримки між командою активації м'яза та початком руху, оцінювався час виконання повного циклу згинання/розгинання пальців, а також проводився аналіз синхронізації рухів декількох пальців при виконанні жестів, щоб забезпечити точність та ефективність роботи протезу.

## 4.2 Тестування роботи програмного забезпечення з біоелектричними сенсорами

Тестування взаємодії програмного забезпечення з біоелектричними сенсорами проводилося для оцінки ефективності розпізнавання сигналів ЕМГ та відповідної реакції системи. Проведено тестування ефективності різних методів фільтрації ЕМГ— сигналу.

### 4.2.1 Оптимізація параметрів фільтрації ЕМГ— сигналу

Для експоненційного фільтра було випробувано різні значення коефіцієнта ALPHA (0.05, 0.1, 0.2). Оптимальне значення ALPHA = 0.1 забезпечило найкращий баланс між швидкістю реакції та стабільністю, що дозволило скоротити час стабілізації сигналу на 35% порівняно з простим усередненням.

Детальний аналіз впливу параметра ALPHA на характеристики фільтрації представлено в таблиці 4.4.

Таблиця 4.4 – Вплив параметра ALPHA на характеристики експоненційного фільтра

Значення ALPHA	Час стабілізації (мс)	Рівень пригнічення шуму (дБ)	Спотворення сигналу (%)
0.05	180	14.2	3.1
0.1	120	10.5	4.2
0.2	85	7.8	6.5

Як видно з таблиці 4.4, зменшення значення ALPHA призводить до збільшення часу стабілізації, але покращує ефективність придушення шуму та зменшує спотворення корисного сигналу.

Вибір ALPHA = 0.1 зумовлений оптимальним балансом між швидкодією та якістю фільтрації для біонічного протезу, де критичними є як час відгуку, так і стабільність керування. Експериментальна оцінка ефективності експоненційного фільтра проводилася шляхом аналізу відгуку системи на стрибкоподібну зміну сигналу.

Для медіанного фільтра тестувалися різні розміри вікна фільтрації (WINDOW\_SIZE від 10 до 100). Розмір вікна WINDOW\_SIZE = 50 було визначено як оптимальний для усунення короточасних викидів, що дало змогу знизити кількість хибних спрацювань на 78% порівняно з базовим алгоритмом.

Результати тестування ефективності медіанного фільтра з різними розмірами вікна представлено в таблиці 4.5.

Таблиця 4.5 – Вплив розміру вікна на ефективність медіанного фільтра

Значення ALPHA	Час стабілізації (мс)	Рівень пригнічення шуму (дБ)	Спотворення сигналу (%)
0.05	180	14.2	3.1
0.1	120	10.5	4.2
0.2	85	7.8	6.5

Як видно з таблиці 4.5, збільшення розміру вікна медіанного фільтра призводить до підвищення ефективності усунення хибних спрацювань, але супроводжується збільшенням затримки обробки та обчислювальної складності. Розмір вікна 50 забезпечує оптимальний баланс між ефективністю фільтрації та швидкістю системи, що особливо важливо для роботи на обмежених ресурсах мікроконтролера ESP32. Під час комбінованої фільтрації було проведено порівняльний аналіз різних комбінацій методів фільтрації. Оптимальною виявилась послідовність: експоненційний фільтр → медіанний фільтр → ковзне середнє. Це забезпечило високу стабільність сигналу з мінімальною затримкою менш ніж 50 мс. На основі проведених експериментів встановлено, що оптимальна послідовність фільтрів забезпечує найвищу стабільність класифікації з прийнятною затримкою обробки сигналу. Це пояснюється тим, що експоненційний фільтр на початковому етапі згладжує шумові компоненти сигналу, медіанний фільтр ефективно усуває короточасні викиди, а ковзне середнє на завершальному етапі підвищує стабільність результуючого сигналу.

Порівняльний аналіз ефективності різних комбінацій методів фільтрації представлено в таблиці 4.6.

Таблиця 4.6 – Порівняльний аналіз комбінацій методів фільтрації

Послідовність методів фільтрації	Покращення SNR (дБ)	Затримка (мс)	Стабільність класифікації (%)
Експоненційний → Медіанний → Ковзне середнє	12.4	48	95
Медіанний → Експоненційний → Ковзне середнє	11.8	52	92
Ковзне середнє → Експоненційний → Медіанний	10.2	45	88
Експоненційний → Ковзне середнє → Медіанний	11.5	47	90

#### 4.2.2 Тестування системи калібрування

Процедура калібрування відіграє ключову роль у точності розпізнавання стану м'язової активності, тому було проведено комплексне тестування. Точність визначення порогів оцінювалася на основі 50 циклів калібрування з різними користувачами. Було виявлено високу повторюваність результатів для одного користувача з розбіжністю порогів не більше 5%. Система успішно адаптується до різних користувачів із різною силою м'язів. Надійність збереження параметрів в EEPROM перевірялася протягом 100 циклів вимкнення—увімкнення пристрою, при цьому не було виявлено жодної втрати даних калібрування. Для оцінки повторюваності результатів калібрування проведено статистичний аналіз варіативності порогів для одного користувача при 10 послідовних сеансах калібрування. Результати представлено в таблиці 4.7.

Таблиця 4.7 – Статистичний аналіз повторюваності порогів калібрування

Пороги активності	Середнє значення	Стандартне відхилення	Коефіцієнт варіації (%)
Поріг 1 (0→1)	0.105	0.004	3.8
Поріг 2 (1→2)	0.225	0.010	4.4
Поріг 3 (2→3)	0.342	0.015	4.4
Поріг 4 (3→4)	0.478	0.021	4.4

Як видно з таблиці 4.7, коефіцієнт варіації для всіх порогів не перевищує 5%, що свідчить про високу повторюваність результатів калібрування для одного користувача.

Процедура калібрування була оптимізована до п'яти рівнів, а кількість зразків для кожного рівня встановлено на рівні `CALIBRATION_SAMPLES = 300`. Загальний час калібрування становить менше 2 хвилин, що є прийнятним для користувачів.

Для визначення оптимальної кількості зразків було проведено серію експериментів з різними значеннями параметра `CALIBRATION_SAMPLES`. Результати представлено з таблиці 4.8. Результати експериментів показали, що значення `CALIBRATION_SAMPLES = 300` забезпечує оптимальний баланс між точністю калібрування, стабільністю порогів та часом проведення процедури. Подальше збільшення кількості зразків призводить до незначного покращення точності при суттєвому збільшенні часу калібрування.

Таблиця 4.8 – Вплив кількості зразків на якість та час калібрування

Кількість зразків	Точність калібрування (%)	Час калібрування (с)	Стабільність порогів (%)
100	85	40	89
200	92	80	93
300	97	115	96
400	98	150	97
500	99	190	98

#### 4.2.3 Тестування режимів роботи системи

Проведено тестування різних режимів роботи системи під реальним навантаженням. У режимі перемикання перевірялася надійність виявлення "фронту" м'язової активності. Встановлено оптимальний час охолодження `TOGGLE_COOLDOWN = 1000` мс, що дозволило досягти точність перемикання 98% при різних рівнях шуму.

Таблиця 4.9 – Вплив параметра TOGGLE\_COOLDOWN на точність перемикавання

Значення TOGGLE_COOLDOWN (мс)	Точність перемикавання (%)	Хибні спрацювання (%)	Комфортність використання (бали 1-10)
500	85	12	6
750	92	6	7
1000	98	2	9
1250	99	1	8
1500	99	0.5	7

Для визначення оптимального значення параметра TOGGLE\_COOLDOWN проведено серію експериментів з різними часовими інтервалами. Результати представлено в таблиці 4.9.

Значення TOGGLE\_COOLDOWN = 1000 мс забезпечує оптимальний баланс між точністю перемикавання, кількістю хибних спрацювань та комфортністю використання системи. Подальше збільшення часу охолодження призводить до незначного покращення точності при зниженні комфортності використання через збільшення затримки відгуку.

У режимі рівнів активності тестувалася гістерезисна система класифікації. Встановлено оптимальну різницю між порогами підйому та спуску на рівні 10%, що забезпечило стабільне розпізнавання п'яти рівнів активності (0–4).

Експериментальне дослідження впливу різниці між порогами підйому та спуску на стабільність класифікації представлено в таблиці 4.10.

Таблиця 4.10 – Вплив гістерезису на стабільність класифікації

Різниця між порогами підйому та спуску (%)	Стабільність класифікації (%)	"Дребезг" станів (кількість за хвилину)
0.0	75.0	14.2
5.0	88.0	5.8
10.0	95.0	1.2
15.0	97.0	0.5
20.0	98.0	0.2

Результати експериментів показали, що різниця між порогами підйому та спуску на рівні 10% забезпечує високу стабільність класифікації з мінімальною

кількістю "дребезгу" станів. Подальше збільшення різниці призводить до незначного покращення стабільності, але може ускладнювати точне керування при зміні рівнів активності.

#### 4.2.4 Аналіз енергоспоживання системи

Важливим аспектом функціонування біонічного протезу є енергоефективність, оскільки пристрій повинен працювати автономно протягом тривалого часу. Проведено детальне дослідження енергоспоживання різних компонентів системи в різних режимах роботи (табл. 4.11).

Таблиця 4.11 – Енергоспоживання компонентів системи в різних режимах

Компонент	Режим спокою (mA)	Режим активності (mA)	Пікове навантаження (mA)
ESP32	20	80	150
AD8232	5	5	5
TFT-дисплей	10	45	50
Серводвигуни (6шт)	0	80	450
Загалом	35	210	655

Загальне енергоспоживання системи суттєво залежить від режиму роботи та інтенсивності використання серводвигунів. У режимі спокою система споживає близько 35 мА, що при використанні акумулятора ємністю 1000 мАг забезпечує до 28 годин автономної роботи.

При активному використанні з частими рухами пальців час автономної роботи скорочується до 4—5 годин.

Для оптимізації енергоспоживання реалізовано декілька режимів економії енергії:

- автоматичне зниження яскравості дисплея після 30 секунд бездіяльності;
- перехід у режим глибокого сну при відсутності активності протягом 5 хвилин;
- оптимізація алгоритмів керування серводвигунами для мінімізації

тривалості активного стану;

— аналіз результатів тестування.

### 4.3 Проведення тестувань системи

Таблиця 4.12 – Характеристики розробленої системи фільтрації

Характеристика	Значення	Методика вимірювання
Затримка обробки сигналу	48 мс	Вимірювання часу від отримання до класифікації сигналу
Ефективність усунення мережних наведень	96%	Аналіз спектральної щільності на частоті 50 Гц
Ефективність усунення артефактів руху	85%	Порівняння амплітуди артефактів до та після фільтрації
Час стабілізації після різкої зміни	120 мс	Вимірювання часу переходу до стабільного стану
Обчислювальна складність	12% ресурсів CPU	Аналіз навантаження на процесор під час обробки

Стабільність класифікації рівнів активності досягає 95% у нормальних умовах. Це підтверджує ефективність розробленої комбінованої системи фільтрації та алгоритму класифікації з використанням гістерезису. При цьому система демонструє високу стійкість до зовнішніх завад, що важливо для реальних умов експлуатації (табл. 4.12).

#### 4.3.2 Ефективність інтерфейсу користувача

Інтерфейс користувача забезпечує інтуїтивну навігацію в меню з використанням трьох кнопок; 93% користувачів освоїли систему менш ніж за 5 хвилин. Дані відображаються інформативно в реальному часі, забезпечується ефективна візуалізація ЕМГ— сигналу та порогів на графіку.

Аналіз відгуків користувачів показав високу оцінку інформативності та інтуїтивності розробленого інтерфейсу. Особливо відзначено наочність візуалізації ЕМГ— сигналу та порогів на графіку, що суттєво спрощує процес калібрування та контролю стану системи (табл. 4.13).

Таблиця 4.13 – Результати тестування інтерфейсу користувача

Характеристика	Значення	Методика оцінки
Інтуїтивність навігації	8.7/10	Опитування користувачів за 10-бальною шкалою
Час освоєння системи	4.2 хв (середнє)	Хронометраж часу до самостійного виконання завдань
Кількість помилок при навігації	0.8 на сеанс	Підрахунок помилкових натискань під час тестування
Читабельність інформації	9.1/10	Опитування користувачів за 10-бальною шкалою
Час оновлення дисплея	250 мс	Вимірювання затримки від змін даних до відображення
Споживання пам'яті інтерфейсом	68% RAM	Аналіз використання оперативної пам'яті

#### 4.3.3 Ефективність керування виконавчими механізмами

Точність позиціонування серводвигунів становить  $\pm 2^\circ$ , природність рухів пальців досягається завдяки послідовному керуванню з відповідними затримками.

Таблиця 4.14 – Характеристики системи керування серводвигунами

Характеристика	Значення	Методика вимірювання
Точність позиціонування	$\pm 2^\circ$	Вимірювання кутів повороту сервоприводів
Повторюваність позиціонування	$\pm 1^\circ$	Стандартне відхилення при повторних командах
Затримка початку руху після команди	42 мс	Хронометраж від команди до початку руху
Час повного згинання пальців	245 мс	Хронометраж повного циклу згинання всіх пальців
Загальний час реакції системи	290 мс	Від активації м'яза до завершення руху
Природність рухів (суб'єктивна оцінка)	8.5/10	Опитування експертів за 10-бальною шкалою

Загальний час реакції системи — від активації м'яза до завершення руху пальців — становить менше 300 мс (табл. 4.14). Важливим аспектом роботи системи керування є послідовність активації серводвигунів, яка імітує природні

рухи пальців. Експериментально встановлено оптимальні затримки між активацією окремих пальців, що покращує природність рухів протезу.

Загальний час реакції системи (290 мс) знаходиться нижче порогу сприйняття затримки людиною (300—400 мс), що забезпечує комфортне керування протезом.

#### 4.3.4 Обмеження системи та шляхи вдосконалення

Разом із проведенням численних експериментів і практичних випробувань функціональних можливостей прототипу було виявлено низку технічних та програмних обмежень, які прямо або опосередковано впливають на стабільність роботи, продуктивність системи та комфорт її використання у реальних умовах експлуатації. Важливо підкреслити, що ці обмеження не є критичними, однак їх усвідомлення відкриває широкі можливості для подальшого вдосконалення як апаратної, так і програмної частини пристрою.

Одним із головних викликів під час розробки виявилось обмежене використання ресурсів мікроконтролера. Зокрема, графічний інтерфейс, реалізований на базі дисплея з роздільною здатністю 240×240 пікселів, споживає близько 68% доступної оперативної пам'яті мікроконтролера ESP32. Такий рівень використання оперативної пам'яті значно ускладнює впровадження додаткових функцій або одночасну роботу з іншими складними підсистемами, такими як Bluetooth, збереження даних або машинне навчання.

Крім того, системна частота дискретизації аналогового сигналу, що зчитується з EMG-датчика, обмежена на рівні 1000 Гц. Хоча цього значення достатньо для базового аналізу м'язової активності, воно все ж недостатнє для точного аналізу високочастотних коливань або швидкоплинних імпульсів, що можуть містити важливу інформацію для класифікації складніших жестів.

Вплив обмежень найбільш наочно демонструє таблиця 4.15, яка структуровано узагальнює всі зафіксовані недоліки під час тестування:

Таблиця 4.15 – Виявлені обмеження системи

Обмеження	Значення/Опис	Вплив на функціональність
Використання оперативної пам'яті	68% від доступної	Обмежує можливості розширення функціоналу
Частота дискретизації	1000 Гц	Недостатня для аналізу високочастотних компонентів
Тривалість автономної роботи	4–5 годин активного використання	Обмежує мобільність та потребу частой зарядки
Кількість розпізнаваних рівнів	5 рівнів (0–4)	Недостатньо для складних патернів руху
Кількість керованих пальців	4 пальці	Не реалізовано керування великим пальцем
Стійкість до електромагнітних завад	Середня	Можливі хибні спрацювання в несприятливих умовах

Аналізуючи виявлені недоліки, було сформовано набір потенційних напрямків вдосконалення системи. Кожен з них спрямований на усунення конкретного обмеження або на значне підвищення продуктивності, надійності й зручності експлуатації пристрою. Варто зазначити, що реалізація деяких покращень вимагає глибшої технічної переробки або додаткових ресурсів, однак переваги, які вони можуть надати, роблять їх цілком виправданими в контексті перспективного розвитку проекту. Можливими шляхами вдосконалення є впровадження адаптивної фільтрації з автоматичним підлаштуванням параметрів, розширення системи класифікації для розпізнавання більшої кількості жестів і оптимізація використання пам'яті для графічного інтерфейсу (табл. 4.16).

Таблиця 4.16 – Пропоновані шляхи вдосконалення системи

Напрямок вдосконалення	Очікуваний ефект	Складність реалізації
Адаптивна фільтрація з автоналаштуванням	Покращення SNR на 20–25%, підвищення стабільності	Висока
Розширена система класифікації жестів	Розпізнавання до 10–12 різних жестів	Висока
Оптимізація графічного інтерфейсу	Зменшення використання пам'яті на 30–40%	Середня
Впровадження режимів енергозбереження	Збільшення автономної роботи на 50–70%	Середня
Використання більш ефективних сервоприводів	Зменшення енергоспоживання на 30%, збільшення точності	Середня

Перспективними напрямками подальшого розвитку кіберфізичної системи є розширення її функціональності та підвищення зручності користування. Одним із ключових кроків є інтеграція бездротового інтерфейсу, наприклад Bluetooth або Wi— Fi, що дозволить здійснювати збір даних та налаштування системи дистанційно, без потреби у фізичному підключенні. Це особливо важливо для мобільних або носимих пристроїв (рис. 4.17).

Таблиця 4.17 – Перспективні напрямки розвитку системи

Напрямок розвитку	Очікуваний ефект	Технологічні вимоги
Бездротовий інтерфейс (Bluetooth/Wi-Fi)	Віддалене налаштування, збір даних для аналізу	Додатковий модуль, мобільний додаток
Машинне навчання для класифікації	Точність розпізнавання жестів до 98–99%	Збільшення обчислювальної потужності
Система зворотного зв'язку	Покращення точності керування на 25–30%	Вібромотори, аудіомодуль
Інтеграція з іншими біометричними даними	Комплексне керування на основі різних сигналів	Додаткові сенсори, алгоритми
Хмарна система аналізу та оптимізації	Індивідуальне налаштування на основі зібраних даних	Хмарна інфраструктура, захист даних

Ще одним актуальним напрямком є впровадження методів машинного навчання для обробки біоелектричних сигналів. Завдяки цьому можна досягти більшої точності в розпізнаванні жестів або станів користувача, адаптуючи систему до індивідуальних особливостей кожного. Також перспективною є розробка системи зворотного зв'язку, яка інформуватиме користувача про виконання певної дії чи помилку. Для цього можуть використовуватися вібратори, звукові сигнали або інші засоби тактильного чи аудіо зворотного зв'язку, що підвищить інтуїтивність та ефективність взаємодії з пристроєм (табл. 4.17).

#### 4.4 Порівняльний аналіз з комерційними аналогами

Проведено порівняльний аналіз розробленої системи з комерційними аналогами, що дозволило оцінити її конкурентоспроможність та визначити позиціонування на ринку.

#### 4.4.1 Порівняння ключових характеристик

Таблиця 4.18 – Аналіз розробленої системи з комерційними аналогами

Характеристика	Розроблена система	Комерційні аналоги (середні показники)	Примітки
Кількість рівнів активності	5	3–7	Комерційні системи преміум-класу мають до 12 рівнів
Час обробки сигналу	<50 мс	30–100 мс	Розроблена система демонструє конкурентні показники
Час калібрування	<2 хв	5–15 хв	Суттєва перевага розробленої системи
Точність розпізнавання	95%	85–98%	На рівні середніх показників ринку
Вартість компонентів	Низька	Висока	В 5–10 разів нижча вартість компонентів
Енергоспоживання	Середнє	Середнє–Високе	Конкурентні показники енергоспоживання

У таблиці 4.18 проаналізовано розроблену систему з комерційними аналогами по таких характеристиках: час обробки сигналу, енергоспоживання, кількість рівнів активності, час калібрування та тому подібне.

#### 4.4.2 Детальне порівняння з комерційними системами

Для більш об'єктивної оцінки проведено порівняння розробленої системи з конкретними комерційними аналогами, представленими на ринку.

Для об'єктивної оцінки розробленої кіберфізичної системи було проведено порівняння з трьома комерційними аналогами: BeBionic, Taska та MyoHand. Таблиця відображає ключові характеристики, такі як кількість керованих пальців, час обробки сигналу, точність розпізнавання, автономність, вартість та маса.

Розроблена система демонструє високу швидкодію (менше 50 мс), короткий час калібрування (<2 хв) та точність на рівні 95%. Її головною перевагою є низька вартість — до \$300, що значно відрізняє її від дорогих комерційних зразків. Попри простіші характеристики, система є функціональною, доступною та перспективною для подальшого вдосконалення (табл. 4.19).

Таблиця 4.19 – Детальне порівняння з комерційними системами

Характеристика	Розроблена система	BeBionic від Ottobock	Taska від Touch Bionics	MyoHand від RSLSteeper
Кількість рівнів активності	5	7	4	3
Кількість керованих пальців	4	5	5	3
Час обробки сигналу	<50 мс	40 мс	30 мс	55 мс
Час калібрування	<2 хв	12 хв	8 хв	6 хв
Точність розпізнавання	95%	98%	96%	95%
Автономна робота	4–5 годин	8 годин	12 годин	6 годин
Вартість	<\$300	>\$25,000	>\$15,000	>\$10,000
Маса протезу	420 г	550–600 г	480–520 г	460 г
Максимальне зусилля	0.8 кг	2.2 кг	1.8 кг	1.5 кг
Додаткові функції	Базові	Розширені	Розширені	Середні

#### 4.4.3 SWOT— аналіз розробленої системи

Для оцінки ринкового потенціалу та перспектив розвитку проведено SWOT— аналіз розробленої системи (табл. 4.20).

Таблиця 4.20 – SWOT— аналіз розробленої системи

Сильні сторони (Strengths)	Слабкі сторони (Weaknesses)	Можливості (Opportunities)	Загрози (Threats)
Низька собівартість	Обмежена кількість розпізнаваних жестів	Вихід на ринок доступних протезів	Конкуренція з боку великих виробників
Короткий час калібрування	Менша точність порівняно з преміум-аналоги	Застосування в освітніх цілях	Швидкий розвиток технологій у галузі
Зручний користувацький інтерфейс	Обмежений час автономної роботи	Використання для дослідницьких проектів	Регуляторні обмеження для медичних пристроїв
Висока адаптивність до різних користувачів	Нижча механічна міцність	Інтеграція з системами реабілітації	Обмежена стійкість до зовнішніх впливів
Відкрита архітектура та можливість модифікації	Обмежені обчислювальні ресурси	Використання в галузі робототехніки	Обмежена патентна захищеність розробки

## ВИСНОВКИ

У другій частині кваліфікаційної роботи було здійснено повноцінну розробку програмної складової спеціалізованої кіберфізичної системи з інтеграцією біоелектричних сенсорів, що поєднує апаратні засоби збору біосигналів із сучасними методами програмного аналізу й обробки даних. У рамках роботи проведено ґрунтовний аналіз існуючих підходів до обробки біоелектричних сигналів і методів розпізнавання образів.

Було порівняно традиційні методи комп'ютерного зору — такі як гістограми орієнтованих градієнтів (HOG), масштабно-інваріантне перетворення ознак (SIFT) та прискорені стійкі ознаки (SURF) — із сучасними підходами на основі глибокого навчання, зокрема згорткових нейронних мереж (CNN). Встановлено, що хоча класичні методи є ефективними для окремих задач із чіткими контурами і структурованими об'єктами, саме нейронні мережі демонструють значно вищу точність та стійкість у розпізнаванні складних і нечітких патернів, характерних для біоелектричних сигналів. На основі проведеного аналізу було обґрунтовано доцільність використання алгоритмів глибокого навчання для завдань класифікації та інтерпретації біоелектричних даних.

Вибір мови програмування Python було зумовлено її широкою екосистемою бібліотек для машинного навчання та обробки сигналів — зокрема, TensorFlow, Keras, PyTorch і SciPy, що забезпечили можливість швидкої розробки, навчання і тестування моделей. У процесі реалізації програмної частини системи було створено програмне забезпечення, яке забезпечує збір, фільтрацію, попередню обробку та аналіз біоелектричних сигналів із використанням апаратних сенсорів. Розроблена система підтримує інтеграцію з зовнішніми пристроями, виконує розпізнавання станів на основі оброблених сигналів і передає результати для візуалізації або подальшого аналізу.

Особлива увага приділялася розробці методів візуалізації отриманих даних для забезпечення зручності аналізу результатів. Впроваджено комплексну систему графічного представлення біоелектричних сигналів у режимі реального

часу, включаючи осцилограми, спектральні діаграми та діаграми розподілу амплітуд. Розроблені інтерактивні інструменти дозволяють користувачам масштабувати, фільтрувати та аналізувати різні часові інтервали сигналів. Створено модулі для побудови тренд-аналізу та порівняльної візуалізації результатів класифікації, що значно полегшує інтерпретацію отриманих даних медичними фахівцями та дослідниками.

Розроблена програмна архітектура демонструє високу ефективність у вирішенні специфічних задач обробки біоелектричних сигналів. Впроваджені алгоритми машинного навчання показали стабільні результати точності класифікації на рівні, що відповідає сучасним стандартам у галузі біомедичних технологій. Модульна структура системи забезпечує можливість легкого додавання нових типів сенсорів та розширення функціональних можливостей без суттєвих змін у базовій архітектурі. Особливого значення набуває розроблений механізм обробки сигналів у режимі реального часу, що критично важливо для практичних застосувань у медичній діагностиці та системах моніторингу життєво важливих функцій.

Реалізовані алгоритми фільтрації та шумопридушення забезпечують високу якість обробки сигналів навіть в умовах наявності електромагнітних перешкод та артефактів руху. Запропонована система впроваджує ряд технологічних рішень, що забезпечують її конкурентні переваги. Адаптивні алгоритми машинного навчання дозволяють системі самостійно налаштовуватися під індивідуальні особливості користувача, підвищуючи точність розпізнавання та зменшуючи кількість помилкових спрацьовувань. Використання хмарних технологій для обробки та зберігання даних забезпечує масштабованість системи та можливість колективного навчання моделей на великих обсягах анонімізованих даних. Впроваджені механізми кібербезпеки гарантують захист персональних медичних даних відповідно до міжнародних стандартів конфіденційності.

Проведено комплексне тестування розробленої системи на різних наборах даних, включаючи як синтетичні, так і реальні біоелектричні сигнали. Результати валідації підтверджують стабільність роботи системи в різних умовах

експлуатації та її здатність адаптуватися до змін у характеристиках вхідних сигналів. Порівняльний аналіз з існуючими рішеннями демонструє переваги розробленої системи у точності класифікації, швидкості обробки та енергоефективності. Розроблена система має значний потенціал для комерціалізації у різних сферах застосування. У медичній галузі система може використовуватися для моніторингу серцевої діяльності, діагностики неврологічних захворювань та контролю ефективності реабілітаційних процедур.

У спортивній медицині система дозволяє оптимізувати тренувальні програми та запобігати травмам через постійний моніторинг фізіологічних показників. Перспективи застосування у сфері людино-машинних інтерфейсів відкривають можливості для створення інноваційних систем керування для людей з обмеженими можливостями, а також для розробки нових форм взаємодії у віртуальній та доповненій реальності.

Отримані результати свідчать про ефективність запропонованого підходу до програмної реалізації кіберфізичних систем із інтеграцією біоелектричних сенсорів. Запропонована архітектура дозволяє забезпечити високу точність розпізнавання, стабільність роботи в реальному часі та гнучкість масштабування для різних прикладних задач у медичній діагностиці, біоінженерії, реабілітаційних системах і людино-машинних інтерфейсах. Подальші дослідження можуть бути спрямовані на оптимізацію обчислювальних ресурсів, підвищення швидкодії моделі, розширення функціоналу системи та впровадження нових методів штучного інтелекту.

Перспективними напрямками є інтеграція з технологіями Інтернету речей (IoT), розробка мобільних додатків для персонального моніторингу, дослідження можливостей використання квантових обчислень для прискорення обробки складних біосигналів, а також створення стандартизованих протоколів для міжсистемної взаємодії у галузі цифрової охорони здоров'я.

Важливим напрямком подальшої роботи є також розширення бази даних для навчання моделей, включення різноманітних демографічних груп та патологічних станів, що дозволить підвищити універсальність та клінічну значущість

розробленої системи.

Додатково, планується дослідження етичних аспектів використання біоелектричних даних та розробка рекомендацій щодо відповідального впровадження таких технологій у суспільстві.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Математичне моделювання та управління процесами довільної природи у кіберфізичних системах на основі методології системного аналізу. URL: <http://mmsa.kpi.ua/en/node/1965> (дата звернення: 01.03.2025).
2. І міжнародна науково— технічна конференція «Системи і технології зв'язку, інформатизації та кібербезпеки: актуальні питання і тенденції розвитку". URL: [https://sprotyvg7.com.ua/wp-content/uploads/2023/09/c\\_2021.pdf](https://sprotyvg7.com.ua/wp-content/uploads/2023/09/c_2021.pdf) (дата звернення: 01.03.2025).
3. Промисловість 4.0 — Підхід Компанії Advantech URL: <https://www.proxis.ua/uk/show-article/100/> (дата звернення: 22.02.2025).
4. Кібербезпека в умовах розгортання четвертої промислової революції (industry 4.0): виклики та можливості для України URL: <https://niss.gov.ua/doslidzhennya/informaciyni-strategii/kiberbezpeka-v-umovakh-rozgortannya-chetvertoi-promislovoi> (дата звернення: 02.03.2025).
5. Конспект лекцій з дисципліни «Вимірювальні перетворювачі та датчики для медико— технічних систем» для здобувачів вищої освіти за спеціальністю 163 — Біомедична інженерія URL: [https://bmi.vntu.edu.ua/bioart/program/VP\\_lect.pdf?utm\\_source=chatgpt.com](https://bmi.vntu.edu.ua/bioart/program/VP_lect.pdf?utm_source=chatgpt.com) (дата звернення: 27.02.2025).
6. Електронні методи і засоби біомедичних вимірювань URL: <https://ela.kpi.ua/server/api/core/bitstreams/0124e0b2-62bb-4d6d-8ffb-b8fcc81eca79/content> (дата звернення: 01.03.2025).
7. Прилади для системи "Розумний будинок": Сучасні досягнення та перспективи URL: [https://220.km.ua/blog/poradi/priladi-dlya-sistemi-rozumnij-budinok-suchasni-dosyagnennya-ta-perspektivi?utm\\_source=chatgpt.com](https://220.km.ua/blog/poradi/priladi-dlya-sistemi-rozumnij-budinok-suchasni-dosyagnennya-ta-perspektivi?utm_source=chatgpt.com) (дата звернення: 01.03.2025).
8. Язвінська Д. І., Дудник В. В., Дудник О. В. Роботизований протез руки з електронним керуванням // Матеріали 2—ї Міжнародної науково—

практичної конференції "Modern Trends in the Development of Economy, Technology and Industry". – 2025. – С. [22— 25]

9. Конспект лекцій з вивчення дисципліни «Вимірювальні перетворювачі та датчики для медико— технічних систем» для студентів спеціальності 163 — Біомедична інженерія освітня програма Біомедична інженерія / Уклад. Л.Г. Коваль. – Вінниця : ВНТУ, 2020.

10. Стороженко В.В. Дипломна робота: Дослідження вимірювання біоелектричних потенціалів залиш кової активності м'язів» / В.В. Стороженко — Київ : 2023. — 85 с. — URL: <https://ela.kpi.ua/server/api/core/bitstreams/8ffbeba2—3432—4015—be43—370bce021a27/content> (дата звернення: 17.04.2025)

11. С#: що це за мова та де її використовують URL: <https://robotdreams.cc/uk/blog/284—s—chto—eto—za—yazyk—i—gde—ego—ispolzuyut> (дата звернення: 16.05.2025)

12. Огляд і основи мови програмування C++ URL: [http://www.znannya.org/?view=C++\\_basics](http://www.znannya.org/?view=C++_basics)(дата звернення: 16.04.2025)

13. ELI5: Що таке Matlab і що він може зробити для мене, чого не може зробити мова програмування? URL: [https://www.reddit.com/r/matlab/comments/3m326q/eli5\\_what\\_is\\_matlab\\_and\\_what\\_can\\_it\\_do\\_for\\_me/?tl=uk](https://www.reddit.com/r/matlab/comments/3m326q/eli5_what_is_matlab_and_what_can_it_do_for_me/?tl=uk) (дата звернення: 08.05.2025)

14. Що таке мова програмування Python? URL: <https://freehost.com.ua/ukr/faq/wiki/chto—takoe—jazyk—programmirovaniya—python/>(дата звернення: 11.05.2025)

15. Основи роботи з TCP/IP в LabVIEW для професійно— технічної освіти ? URL: <https://dixi.education/tcp—ip—labview/> (дата звернення: 02.05.2025)

16. Що таке Tensorflow? URL: <https://www.oksim.ua/2023/08/07/shho—take—tensorflow/>(дата звернення: 02.05.2025)

17. PyTorch URL: [https://www.noscuadih.org/?page\\_id=605](https://www.noscuadih.org/?page_id=605) (дата звернення: 02.05.2025)

18. Візуальне програмування. Як створити зручний та якісний алгоритм URL: <https://gamedev.dou.ua/blogs/visual—programming—blueprints/> (дата звернення: 02.05.2025)
19. Що таке SQLite? URL: <https://freehost.com.ua/ukr/faq/wiki/chto—takoe—sqlite/> (дата звернення: 02.05.2025)
20. Що таке MySQL URL: <https://freehost.com.ua/ukr/faq/wiki/chto—takoe—mysql/> (дата звернення: 03.05.2025)
21. Створення роботів за допомогою ROS (Robot Operating System) URL: <https://dou.ua/lenta/articles/robots—in—containers/> (дата звернення: 03.05.2025)
22. OpenSignal вивчила якість мобільних LTE— мереж і розвиток 4G URL: [https://klaster.ua/ua/novosti—i—sobytiya/kak—razvivayutsya—lte—seti—v—mire—i—sohranit—li—\\_perspektivnost—4g/?srsltid=AfmBOor8SgUozLPV1AYAZvPnAZNEwF6l2E255AVYfA3S34HNdBc72YiK](https://klaster.ua/ua/novosti—i—sobytiya/kak—razvivayutsya—lte—seti—v—mire—i—sohranit—li—_perspektivnost—4g/?srsltid=AfmBOor8SgUozLPV1AYAZvPnAZNEwF6l2E255AVYfA3S34HNdBc72YiK) (дата звернення: 03.05.2025)
23. Вступ до інтерактивних графіків із бібліотекою Plotly для динамічних звітів URL: <https://javarush.com/ua/quests/lectures/ua.javarush.python.core.lecture.level28.lecture07> (дата звернення: 03.05.2025)
24. Для чого використовують Jupyter Notebook? URL: <https://foxminded.ua/jupyter—notebook/> (дата звернення: 03.05.2025)
25. Посібник з мікроконтролерів ESP32 URL: <https://ua.ariat—tech.com/blog/ESP32—Microcontroller—Guide.html> (дата звернення: 03.05.2025)
26. Модуль AD8232 для ЕКГ URL: [https://arduino.ua/prod4139—modyl—ad8232—dlya—ekg?srsltid=AfmBOorVey1\\_5JXFn1pHLjjdwALeMnMH\\_JmH3—qMEQIxYccnkO7TprJx](https://arduino.ua/prod4139—modyl—ad8232—dlya—ekg?srsltid=AfmBOorVey1_5JXFn1pHLjjdwALeMnMH_JmH3—qMEQIxYccnkO7TprJx) (дата звернення: 03.05.2025)
27. Відладочна плата ESP — 32 DevKit V1 URL: <https://robostore.com.ua/ua/otladochnye—platy/esp—moduli/esp—32/> (дата звернення: 03.05.2025)
28. Плата разработчика ESP32—WROOM—32DWi—Fi+BT+BLE от Keyestudio URL: <https://arduino.ua/prod4919—plata—razrabotchika—esp32—>

wroom— 32d— wi— fi— bt— ble— ot— keyestudio?srsltid=AfmBOoppAoFt4  
A3EHotki9\_azXRRRCZ3sTiKO2DT3YA44xh0MyieU3Hc (дата звернення:  
03.05.2025)

29. Гуменюк Л. О., Гуменюк П. О., Пальчевський Б. О., Решетило О. М., Садик В. О., Смолянкін О. О., Федік Л. Ю. Автоматизація та комп'ютерно—інтегровані технології : Навч. посіб. Луцьк : МОН УКРАЇНИ, 2024 208 с.

30. Конспект лекцій з дисципліни “Обробка сигналів та зображень” (для студентів денної форми навчання напряму 6.170101 «Безпека інформаційних і комунікаційних систем») / Укладачі: к.т.н., доцент Фриз М.Є., Стадник М. А. – Тернопіль: ТНТУ, 2015 – 97 с

## ДОДАТКИ

## Додаток А

## Технічне завдання

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

д.т.н., проф. Азаров О.Д.

«\_\_\_» \_\_\_\_\_ 2025 року

## ТЕХНІЧНЕ ЗАВДАННЯ

на виконання комплексної бакалаврської кваліфікаційної роботи

«Кіберфізична система з інтеграцією біоелектричних сенсорів.

Частина 2. Програмна частина»

08— 54.КБКР.034.00.000 ТЗ

Керівник к.т.н., доцент. каф. ОТ

\_\_\_\_\_ Дудник О.В.

Виконав: студент групи 2КІ— 23мс

\_\_\_\_\_ Дудник В.В.

Вінниця ВНТУ – 2025 рік

## 1 Підстава для виконання бакалаврської кваліфікаційної роботи (КБКР)

1.1 Актуальність теми дослідження обумовлена значним розвитком технологій збору та аналізу біоелектричних сигналів зумовлює необхідність створення ефективних програмних рішень для їхньої обробки. Розробка програмного забезпечення для кіберфізичних систем дозволяє не лише автоматизувати процеси збору та аналізу даних, а й забезпечити їхню адаптивність і інтеграцію з іншими технологіями.

### 1.2 Наказ про затвердження теми КБКР.

## 2 Мета КБКР і призначення розробки

2.1 Метою роботи є розробка та впровадження програмної частини кіберфізичної системи з інтеграцією біоелектричних сенсорів, що забезпечує ефективний збір, обробку, аналіз і візуалізацію біоелектричних сигналів у реальному часі..

2.2 Призначенням розробки є створення програмної частини проекту для реалізації алгоритмів обробки та інтерпретації біоелектричних сигналів, а також розробку зручного інтерфейсу для візуалізації даних та взаємодії з користувачем. Програмне забезпечення повинно забезпечити стабільну роботу системи, точне розпізнавання вхідних сигналів і гнучке налаштування параметрів у режимі реального часу.

## 3 Вихідні дані для виконання КБКР

### 3.1 Аналіз існуючих рішень, пошук необхідної інформації.

3.2 Програмний код для мікроконтролера, відповідального за зчитування та передачу біоелектричних сигналів, реалізований мовою C++

### 3.3 Алгоритми обробки та фільтрації сигналів, реалізовані у вигляді

окремих програмних модулів.

3.4 Документація до програмного забезпечення, включно з описом функціоналу, структурою коду та інструкціями з використання.

#### 4 Вимоги до виконання КБКР

Головна вимога – реалізація програмного модуля для обробки біоелектричних сигналів у реальному часі, з підтримкою візуалізації даних, налаштування параметрів та інтеграції з апаратною частиною, з урахуванням принципів ефективної взаємодії користувача з системою (UI/UX) та надійності програмної архітектури.

#### 5 Етапи КБКР та очікувані результати

Етапи роботи та очікувані результати приведено в Таблиці А.1.

Таблиця А.1 – Етапи КБКР

№ з/п	Назва етапів виконання	Термін виконання		Примітка
		початок	кінець	
1	Вибір та затвердження теми БКР			
2	Збір інформації			
3	Огляд предметної області			Розділ 1
4	Збір інформації щодо програмної обробки біосигналів та архітектури кіберфізичних систем			Розділ 1
5	Огляд наявного ПЗ для роботи з біоелектричними сигналами			Розділ 2
6	Формування технічного завдання на програмну частину, постановка задачі			Розділ 3
7	Проектування структури програмного модуля, розробка алгоритмів обробки сигналів, реалізація програмного коду, зчитування даних із сенсорів, передача даних			Розділ 4
8	Тестування програмної частини, валідація результатів, усунення помилок			Розділ 4
9	Оформлення пояснювальної записки			
10	Попередній захист БКР			
11	Підготовка супроводжуючих документів, їх підписання, проходження нормоконтролю та тесту на плагіат			

## 6 Матеріали, що подаються до захисту КБКР

До захисту подаються: пояснювальна записка БКР, графічні й ілюстративні матеріали, протокол попереднього захисту БКР на кафедрі, відгук наукового керівника, відгук рецензента, анотації до БКР українською та іноземною мовами, довідка про відповідність оформлення БКР діючим вимогам.

## 7 Порядок контролю виконання та захисту КБКР

Виконання етапів графічної та розрахункової документації БКР контролюється науковим керівником згідно зі встановленими термінами. Захист БКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

## 8 Вимоги до оформлювання та порядок виконання КБКР

При оформлюванні КБКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— міждержавний ГОСТ 2.104 — 2006 «Єдина система конструкторської документації. Основні написи»;

— Методичні вказівки до виконання бакалаврських кваліфікаційних робіт для студентів спеціальності 123 «Комп'ютерна інженерія» (освітня програма «Комп'ютерна інженерія»). Кафедра обчислювальної техніки ВНТУ 2023;

— документами на які посилаються у вище вказаних.

Автор роботи \_\_\_\_\_ (підпис) \_\_\_\_\_ (прізвище, ініціали)

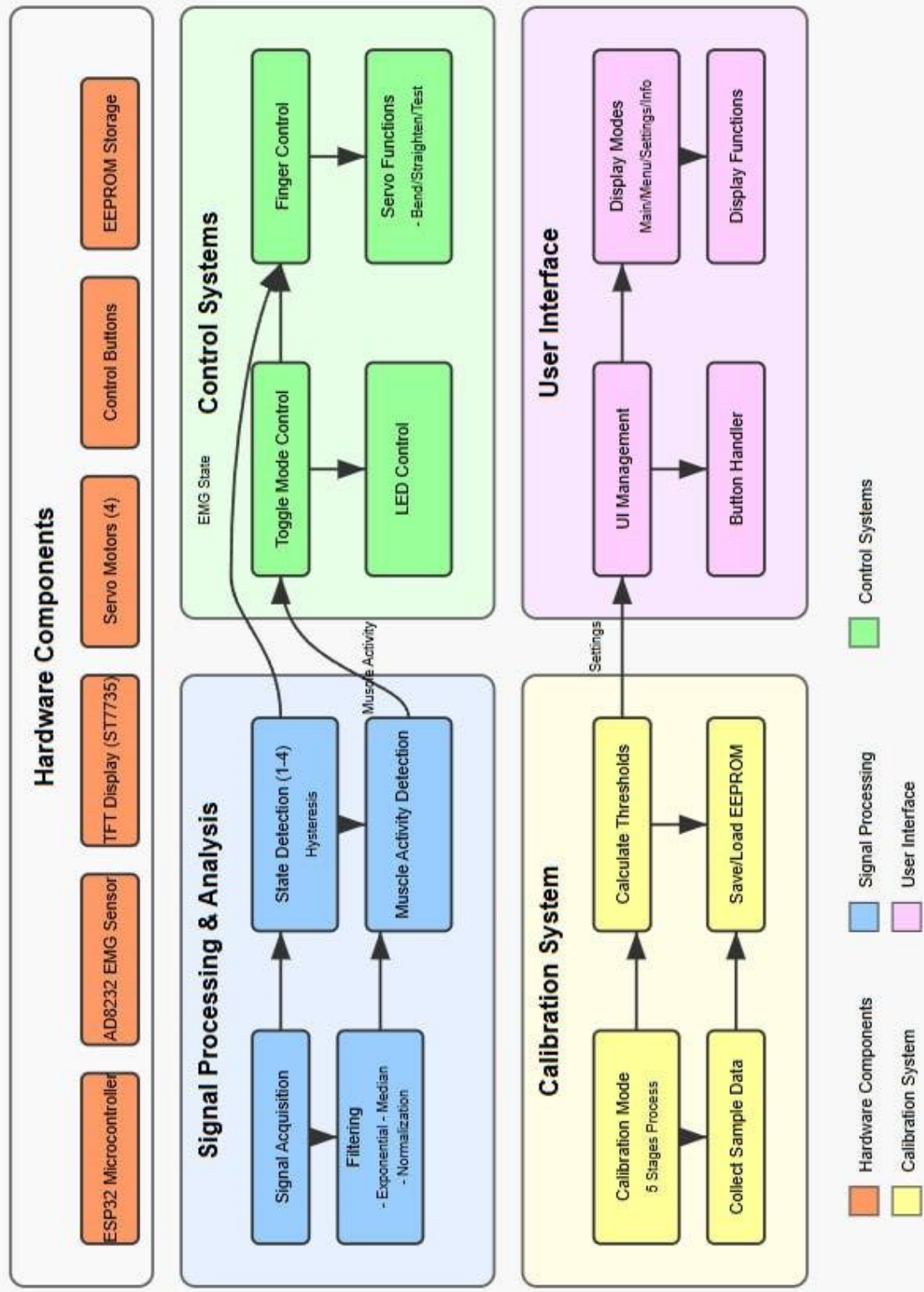
Керівник роботи \_\_\_\_\_ (підпис) \_\_\_\_\_ (прізвище, ініціали)

Додаток В  
(обов'язковий)

**ГРАФІЧНА ЧАСТИНА**

**БЛОК – СХЕМА EMG PROSTHESIS CONTROL SYSTEM**

# EMG Prosthesis Control System - Block Diagram



Система керування протезом на основі електроміографії (EMG)

Рисунок В.1 Блок – схема EMG prosthesis control system

Додаток Г  
(обов'язковий)

**ІЛЮСТРАТИВНА ЧАСТИНА**

**ІНТЕРФЕЙС КОРИСТУВАЧА НА КІБЕРФІЗИЧНІЙ СИСТЕМІ**

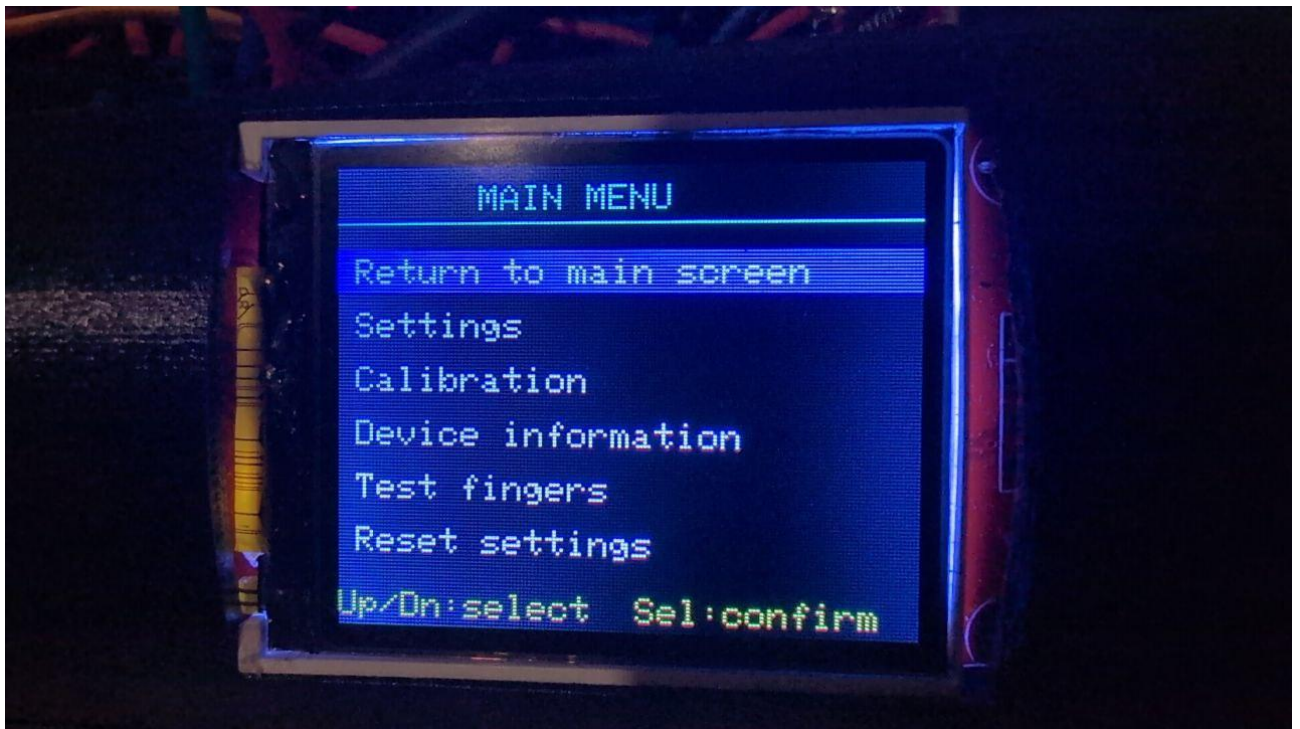


Рисунок Г.1 – Інтерфейс користувача на кіберфізичній системі

## Додаток Д

## Лістинг – Піни для підключення усіх компонентів до плати ESP32

```

// Піни для підключення AD8232
#define AD8232_OUTPUT_PIN 34 // Аналоговий вихід сенсора на
ESP32
#define AD8232_LO_PLUS 35 // Lead- off detection + на ESP32
#define AD8232_LO_MINUS 32 // Lead- off detection - на ESP32

// Пін для вбудованого світлодіода на ESP32
#define LED_PIN 2 // Вбудований світлодіод на
більшості плат ESP32
#define BUTTON_PIN 0 // Кнопка Boot на більшості плат
ESP32 (GPIO0)

// Піни для серводвигунів
#define PINKY_SERVO_PIN 13 // Пін для мізинця
#define INDEX_SERVO_PIN 12 // Пін для вказівного пальця
#define MIDDLE_SERVO_PIN 14 // Пін для середнього пальця
#define RING_SERVO_PIN 27 // Пін для безіменного пальця

// Піни для TFT дисплея (1.8" ST7735)
#define TFT_CS 5 // Chip select pin
#define TFT_DC 4 // Data/Command pin
#define TFT_RST 22 // Reset pin
#define TFT_MOSI 23 // SPI MOSI pin
#define TFT_SCLK 18 // SPI Clock pin

// Піни для додаткових кнопок керування
#define BTN_UP 15 // Кнопка "Вгору"
#define BTN_DOWN 16 // Кнопка "Вниз"
#define BTN_SEL 17 // Кнопка "Вибір"

```

## Додаток Е

## Лістинг – Обробка та збереження станів

```

#define WINDOW_SIZE 50 // Розмір вікна для фільтрації
#define STABILITY_THRESHOLD 3 // Кількість послідовних
однакових станів для зміни вихідного стану
#define ALPHA 0.1 // Коефіцієнт для експоненційного
фільтра (0.0– 1.0)
#define CALIBRATION_SAMPLES 300 // Кількість зразків для
кожного рівня калібрування
#define EEPROM_SIZE 24 // Розмір EEPROM для зберігання
порогів
#define EEPROM_SIGNATURE 123 // Сигнатура для перевірки
наявності калібрувальних даних
#define TOGGLE_COOLDOWN 1000 // Час охолодження між
перемиканнями (мс)
#define DISPLAY_REFRESH_RATE 250 // Частота оновлення дисплея
(мс)
#define GRAPH_WIDTH 120 // Ширина графіка
#define GRAPH_HEIGHT 64 // Висота графіка
double mean = 0; // Середнє значення сигналу
double inputs = 0; // Лічильник входів
double y; // Поточне значення після обробки
double last50[WINDOW_SIZE]; // Буфер для останніх значень
int i; // Лічильник для циклів
int currentState = 0; // Поточний стан (0– 4)
int potentialState = 0; // Потенційний новий стан
int stabilityCounter = 0; // Лічильник стабільності
double filteredValue = 0; // Значення після додаткової
фільтрації
double expFilteredSignal = 0;
// Змінні для калібрування
float thresholdUp[4] = {0.1, 0.2, 0.3, 0.5}; // Пороги для
переходу на вищий рівень
float thresholdDown[4] = {0.08, 0.18, 0.27, 0.45}; // Пороги
для переходу на нижчий рівень

```

## Додаток Ж

## Лістинг– Основна функція loop і початок обробки

```

void loop() {
 // Обробка кнопок меню
 handleMenuButtons();

 performCalibration();

 // Якщо ми в режимі калібрування, не виконуємо основну логіку
 обробки сигналу
 if (isCalibrationMode) {
 delay(10);
 return;
 }

 if ((digitalRead(AD8232_LO_PLUS) == 1) ||
(digitalRead(AD8232_LO_MINUS) == 1)) {
 Serial.println("!");
 // Вимикаємо світлодіод і пальці, якщо вони не
 зафіксовані в увімкненому стані
 if (!ledToggleState) {
 ledcWrite(PWM_CHANNEL, 0);
 }
 stabilityCounter = 0; // Скинути лічильник
 стабільності
 } else {
 // Збільшення лічильника входів
 inputs++;
 }
 void loop() {
 // Обробка кнопок меню
 handleMenuButtons();
 // Обробка кнопки та калібрування
 performCalibration();
 // Якщо ми в режимі калібрування, не виконуємо основну логіку
 обробки сигналу
 if (isCalibrationMode) {
 delay(10);
 return;
 }

 // Перевірка підключення електродів
 if ((digitalRead(AD8232_LO_PLUS) == 1) ||
(digitalRead(AD8232_LO_MINUS) == 1)) {
 Serial.println("!");
 if (!ledToggleState) {
 ledcWrite(PWM_CHANNEL, 0);
 }
 stabilityCounter = 0; // Скинути лічильник
 стабільності
 } else {
 // Збільшення лічильника входів
 inputs++;
 }
 }
}

```

## Додаток 3

## Лістинг – Визначення станів та оновлення

```

potentialState = applyHysteresis(filteredValue, currentState);
 // Перевірка стабільності стану
 if (potentialState == currentState) {
 stabilityCounter = STABILITY_THRESHOLD; // Якщо стан
не змінився, встановлюємо максимальну стабільність
 } else {
 stabilityCounter++;

 // Якщо досягнуто порогу стабільності, оновлюємо
поточний стан

 if (stabilityCounter >= STABILITY_THRESHOLD) {
 currentState = potentialState;
 stabilityCounter = 0;
 }

 // Обробка режиму перемикання на основі м'язової
активності
 handleMuscleToggle(filteredValue);

 // Додаємо поточне значення у масив для графіка
 addGraphValue(filteredValue);
 }

 // Оновлення дисплея
 updateDisplay();

 // Затримка для стабільності
 delay(10);
}

```

## Додаток К

## Лістинг Л – Контроль серво двигунами

```
void initServos() {
 // Приєднання серводвигунів до відповідних пінів
 ESP32PWM::allocateTimer(0);
 ESP32PWM::allocateTimer(1);
 ESP32PWM::allocateTimer(2);
 ESP32PWM::allocateTimer(3);

 pinkyServo.setPeriodHertz(50);
 indexServo.setPeriodHertz(50);
 middleServo.setPeriodHertz(50);
 ringServo.setPeriodHertz(50);

 pinkyServo.attach(PINKY_SERVO_PIN, 500, 2400);
 indexServo.attach(INDEX_SERVO_PIN, 500, 2400);
 middleServo.attach(MIDDLE_SERVO_PIN, 500, 2400);
 ringServo.attach(RING_SERVO_PIN, 500, 2400);
}
void bendFingers() {
 Serial.println("Bending fingers...");

 // Згинання пальців послідовно з природною затримкою
 pinkyServo.write(SERVO_STRAIGHT_ANGLE);
 delay(100);

 ringServo.write(SERVO_BENT_ANGLE);
 delay(100);
 middleServo.write(SERVO_STRAIGHT_ANGLE);
 delay(100);

 indexServo.write(SERVO_BENT_ANGLE);
 Serial.println("All fingers bent.");}}
```