

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет

імені Петра Могили

Факультет комп'ютерних наук

Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри,
д-р техн. наук, проф.

_____ І. М. Журавська

«__» _____ 2024 р.

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

**Апаратно-програмний комплекс моделювання
руху протезів**

Спеціальність 123 Комп'ютерна інженерія

123 – КБР.01 – 405.22010507

Студент

_____ О. М. Данилова

підпис

«__» _____ 202__ р.

Керівник ст. викладач

_____ І. С. Бурлаченко

підпис

«__» _____ 202__ р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Зав. кафедри _____ І. М. Журавська

« _____ » _____ 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної бакалаврської роботи

Видано студенту групи 405 факультету комп'ютерних наук

Даниловій Олександрі Миколаївні
(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

Апаратно-програмний комплекс моделювання руху протезів
Затверджена наказом по ЧНУ ім. Петра Могили від 30.01.2024 № 17.

2. Строк представлення кваліфікаційної роботи « _____ » _____ 20__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні
Очікуваним результатом роботи є: розробка апаратно-програмного комплексу моделювання руху протезів

4. Перелік питань, що підлягають розробці:

- дослідження предметної області;
- аналіз сучасних трендів у розробці протезів;
- огляд існуючих систем моделювання руху протезів;
- розробка програмного забезпечення;
- технології та інструменти для розробки-апаратного комплексу моделювання руху протезів; аналіз системи, що розробляється

5. Перелік графічних матеріалів:

презентація

6. Завдання до спеціальної частини

Аналіз охорони праці на робочих місцях фахівців з інформаційних технологій.

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Алексєєва А. О. канд. техн. наук, доцент	кафедра екології Медичного інституту ЧНУ імені Петра Могили	Спеціальна частина з охорони праці

Керівник роботи

ст. викладач Бурлаченко І. С.

(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

Данилова О. М.

(прізвище, ім'я, по батькові студента)

(підпис)

Дата видачі завдання « ____ » _____ 20 ____ р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: Апаратно-програмний комплекс моделювання руху протезів

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КР	11.12.2024	12.12.2024	Виконано
2	Огляд літератури за темою роботи	15.01.2024	18.02.2024	Виконано
3	Складання календарного плану БКР	19.02.2024	06.03.2024	Виконано
4	Аналіз предметної області	19.02.2024	04.03.2024	Виконано
5	Розробка проектних рішень	23.02.2024	09.03.2024	Виконано
6	Моделювання та конструювання АПЗ	20.02.2024	27.02.2024	Виконано
7	Перевірка працездатності, тестування та апробація розробленого АПЗ, аналіз результатів тестування	01.03.2024	04.03.2024	Виконано
8	Відгук керівника КР	02.06.2024	10.06.2024	Виконано
9	Оформлення БКР та презентації	15.05.2024	03.06.2024	Виконано
10	Попередній захист	38.05.2024	04.06.2024	Виконано
11	Завершення оформлення КР та презентації	30.05.2024	30.05.2024	Виконано
12	Рецензування	12.05.2024	20.05.2024	Виконано
13	Захист бакалаврської кваліфікаційної роботи	.06.2024	.06.2024	Виконано

Розробив здобувач ВО Данилова Олександра Миколаївна _____
(прізвище, ім'я, по батькові) (підпис)
« ____ » _____ 20__ р.

Керівник роботи ст. викладач Бурлаченко Іван Сергійович _____
(посада, прізвище, ім'я, по батькові) (підпис)
« ____ » _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи
«Апаратно-програмний комплекс моделювання руху протезів»
Студент 405 гр.: Данилова Олександра Миколаївна
Керівник: ст. викладач Бурлаченко Іван Сергійович

Дана робота присвячена розробці апаратно-програмного комплексу моделювання руху протезів.

Об'єкт роботи: моделювання руху протезів, розробка програмного забезпечення та апаратних рішень для симуляції руху протезів. Предмет роботи: програмні засоби створення апаратного комплексу моделювання руху протезів.

Мета роботи: покращення функціональності рухових протезів, вивчення впливу різних параметрів на рух протезів та впровадження інноваційних технологій для полегшення життя пацієнтів з ампутаціями за рахунок розробки апаратно-програмного комплексу моделювання руху протезів.

Кваліфікаційна робота бакалавра складається з вступу, 3 розділів, висновків та переліку джерел посилання. У вступі визначається актуальність теми, що приймається за мету та невеликий огляд поставленої задачі, предмет дослідження та об'єкт роботи.

У першому розділі розглядаються сучасні тренди у розробленні протезів, аналізу систем моделювання руху, переваг та недоліків програмного забезпечення, обґрунтовується план виконання завдання, формується специфікація вимог до апаратного та програмного забезпечення.

У другому розділі описується огляд координації людського тіла під час пересування з метою освоєння сегментації людського тіла та дослідження центральної нервової системи людини. Розглянуто взаємодію між біомеханічними та нейронними системами в управлінні тілом, оскільки ефективна нейронна система має забезпечувати точну апроксимацію моделей нейромеханічної активності або функцій з мінімальними витратами на інфраструктуру нейронних схем.

У третьому розділі описується результат виконаної роботи з конструювання та моделювання програмного забезпечення, включаючи вибір технологій, мови програмування, та компонентів застосунку, тобто бібліотек, плагінів та ін та опис інтерфейсу апаратно-програмного комплексу моделювання руху протезів.

У висновках наведено аналіз виконаної роботи та отриманих результатів дослідження та розроблення. У додатку А наведений лістинг з програмним кодом до апаратно-програмного комплексу. В цілому, кваліфікаційна робота без додатків містить 72 сторінок, 43 рисунків, 3 таблиці, 25 джерел посилання.

Ключові слова: протезування, моделювання руху, робототехніка, мікроконтролер, біоніка, кінцівки, штучний інтелект, електроніка.

ABSTRACT

of the Bachelor's Thesis

"Hardware and software complex for modeling the movement of prostheses"

Student: Danylova Oleksandra Mykolaivna

Supervisor: senior teacher Burlachenko Ivan Serhiiiovych

This bachelor's thesis is devoted to the development of a hardware and software complex for modeling the movement of prostheses.

Object of bachelor's thesis: modeling of prosthesis movement, development of software and hardware solutions for simulation of prosthesis movement.

Subject of bachelor's thesis: software tools for creating a hardware complex for modeling the movement of prostheses.

Purpose: to improve the functionality of motor prostheses, to study the influence of various parameters on the movement of prostheses and to introduce innovative technologies to facilitate the lives of patients with amputations by developing a hardware and software complex for modeling the movement of prostheses.

The bachelor's thesis consists of an introduction, 3 chapters, conclusions and a list of references. The introduction defines the relevance of the topic, which is taken as the goal and a brief overview of the task, the subject of research and the object of work.

The first chapter discusses current trends in the development of prostheses, analysis of motion modeling systems, advantages and disadvantages of software, justifies the plan for the task, and forms a specification of requirements for hardware and software.

The second section describes an overview of human body coordination during movement in order to master the segmentation of the human body and study the human central nervous system. The interaction between biomechanical and neural systems in body control is discussed, since an efficient neural system should provide accurate approximation of neuromechanical activity or function models with minimal cost for neural circuitry infrastructure.

The third section describes the results of the work performed on software design and modeling, including the choice of technologies, programming language, and application components, i.e. libraries, plug-ins, etc., and a description of the interface of the hardware and software system for modeling prosthetic movement.

The conclusions provide an analysis of the work performed and the results of research and development. Appendix A contains a listing with the program code for the hardware and software complex. In general, the qualification work without appendices contains 72 pages, 43 figures, 3 tables, 25 references.

Keywords: *prosthetics, motion modeling, robotics, microcontroller, bionics, limbs, artificial intelligence, electronics.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП	4
1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ.....	6
1.1 Актуальність апаратно-програмного комплексу моделювання руху протезів.....	6
1.2 Огляд-аналіз сучасних трендів у розробці протезів	8
1.3 Огляд існуючих систем моделювання руху протезів	13
1.4 Аналіз ринку протезів верхніх кінцівок.....	17
1.5 Вимоги до апаратно- програмного комплексу	20
Висновок до розділу 1	24
2 ПРОЄКТУВАННЯ АПАРАТНО-ПРОГРАМНОГО МОДУЛЮ ДЛЯ МОНІТОРИНГУ РУХУ КІНЦІВОК	25
2.1 Координація людського тіла у пересуванні	25
2.2 Моделювання руху протезів	30
Висновок до розділу 2	35
3 РОЗРОБЛЕННЯ ВЕБПЛАТФОРМИ МОДЕЛЮВАННЯ РОБОТИ ПРОТЕЗІВ	36
3.1 Технології та інструменти для розробки апаратно-програмного комплексу моделювання руху протезів	36
3.2 Мова програмування.....	42
3.3 Розроблення апаратного і програмного забезпечення	46
3.4 Розробка серверної частини	58
3.5 Реалізація фільтру Савицького-Голея.....	65
3.6 Розробка фронтенд частини	68
Висновок до розділу 3	69
ВИСНОВКИ.....	71
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	73
ДОДАТОК А.....	76

ПЕРЕЛІК СКОРОЧЕНЬ

ВООЗ	– Всесвітньої організації охорони здоров'я
ГРТ	– голова, руки та тулуб
ПЗ	– програмне забезпечення
СРЗ	– сила реакції землі
ЦНС	– центральна нервова система
IoT	– Internet of Things
MEMS	– Micro Electro Mechanical Systems

ВСТУП

Актуальність теми кваліфікаційної роботи бакалавра зумовлена зростаючою кількістю людей, які мають ампутації або інші фізичні обмеження. За даними Всесвітньої організації охорони здоров'я (ВООЗ), більше 30 мільйонів людей у світі мають ампутації. Незважаючи на значні досягнення в області протезування, існуючі протези часто обмежуються у функціональності та можуть бути недосконалими у реконструкції руху. Однак протезування в Україні тільки починає розвиватися, і люди, які отримали поранення під час військових дій чи народилися з вродженими дефектами довго чекають, щоб отримати протезування, але навіть тривале очікування не гарантує отримання протезу, адже якісне протезування коштує дорого, і не всі можуть собі це дозволити.

Завдяки розвитку технологій, таких як інтернет речей (IoT), стало можливим створювати більш ефективні та функціональні протези, що будуть збирати дані, відстежувати функціональність та надавати пацієнтам більше можливостей контролю. Це включає в себе використання датчиків, мікроконтролерів та інших технічних рішень для покращення функціональності протезів.

Рухові протези допомагають пацієнтам повертатися до роботи, навчання та активного способу життя. Це сприяє їх соціальній інтеграції та покращує якість життя. Використання апаратно-програмного комплексу може сприяти більш активній і ефективній реабілітації пацієнта, допомогти швидше відновити функції і повернутися до нормального способу життя.

Моделювання руху протезів є важливим напрямком досліджень. Воно дозволяє вивчати різні аспекти, такі як оптимальні параметри дизайну, ефективність алгоритмів керування та взаємодію з навколишнім середовищем. Вивчення алгоритмів керування протезами допоможе вдосконалити їх точність та швидкість реакції на дії пацієнта. Дослідження, проведене в рамках розробки апаратно-програмного комплексу, допоможе зрозуміти взаємодію

між принципами роботи людського організму і технологіями, які є важливими для подальших медичних досліджень і розробки терапевтичних методик.

Об'єкт роботи: моделювання руху протезів.

Предмет роботи: методи обробки даних сенсорів руху частин протезів на базі ESP8266.

Мета роботи: покращення рухової функціональності протезів, вивчення впливу різних параметрів на рух протезів та впровадження інноваційних технологій для полегшення життя пацієнтів з ампутаціями за рахунок розробки апаратно-програмного комплексу моделювання руху протезів.

Відповідно для досягнення визначеної мети необхідно вирішити наступні завдання:

- аналіз сучасних трендів у розробці протезів;
- огляд існуючих систем моделювання руху протезів;
- аналіз ринку протезів;
- дослідження координації людського тіла у пересуванні;
- моделювання руху протезів;
- огляд технологій для розробки апаратно-програмного комплексу;
- розробка і програмного забезпечення.

Сфера застосування: апаратно-програмний комплекс моделювання руху протезів можна використовувати для симуляції руху протезів, вивчення алгоритмів керування та оптимізації їх функціональності, що допоможе інженерам та дослідникам розробляти більш ефективні та зручні протези, а лікарі зможуть підібрати оптимальний тип протезу під конкретного пацієнта та налаштувати під його потреби.

Апробація результатів кваліфікаційної роботи відбулася під час XXI Міжнародної наукової конференції «Ольвійський форум-2024: Стратегії країн Причорноморського регіону в геополітичному просторі» (Миколаїв, 20–23 червня 2024 р.).

1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ

1.1 Актуальність апаратно-програмного комплексу моделювання руху протезів

Актуальність теми даної теми виявляється у багатьох аспектах, що об'єднуються в одну важливу справу – поліпшення якості життя людей з ампутаціями та забезпечення їм можливості повноцінного життя.

Технологічний прогрес у сферах робототехніки та комп'ютерних технологій дозволяє створювати більш продуктивні та функціональні протези, які точніше відтворюють рухи людського тіла, що забезпечує не лише покращення якості протезів, а й підвищення комфорту та сприяє швидшій адаптації для їхніх користувачів.

Зростаюча кількість ампутацій внаслідок захворювань та травм ставить перед медичною наукою завдання постійного вдосконалення протезних технологій. Апаратно-програмні комплекси в цьому контексті можуть стати ключовим елементом, надаючи можливість індивідуалізованого підходу до кожного пацієнта.

Реабілітація пацієнтів після ампутації залежить від якості протезування. Інтеграція апаратно-програмних комплексів у моделювання руху протезів сприяє природним та інтуїтивним рухам, що забезпечує швидше та ефективніше відновлення пацієнтів.

Індивідуалізація протезів є важливим аспектом, оскільки кожна людина має свої унікальні фізичні характеристики та потреби. Розробка апаратно-програмних комплексів дозволяє створювати протези, які ідеально відповідають індивідуальним параметрам користувача, забезпечуючи максимальний комфорт та зручність.

Інтеграція апаратно-програмних комплексів з іншими медичними та реабілітаційними системами дозволяє створювати комплексні рішення для підтримки здоров'я та активності пацієнтів, що є важливим кроком у напрямку інтегрованого підходу до медичної допомоги.

З розвитком штучного інтелекту та машинного навчання актуальність теми набуває ще більшого значення. Використання алгоритмів штучного інтелекту дозволяє створювати протези, які можуть адаптуватися до змінних потреб користувача в реальному часі. Наприклад, системи машинного навчання можуть аналізувати патерни рухів користувача та автоматично підлаштовувати параметри протеза для оптимальної функціональності.

Крім того, інтеграція сенсорних технологій, таких як нейроморфні чіпи або м'язеві імпланти, у апаратно-програмні комплекси може значно покращити сприйняття та взаємодію користувача з протезом. Це відкриває додаткові можливості для створення більш точних, ергономічних та натуральних у використанні протезів.

Поміж інших перспектив можна відзначити потенційне використання віртуальної та доповненої реальності в розробці апаратно-програмних комплексів для моделювання руху протезів. Ці технології можуть стати важливим інструментом для тренування та адаптації користувачів до нового протезу, а також для віртуального тестування його функціональності та сумісності з реальними умовами використання.

Крім того, з поглибленням розуміння нейрофізіології та механізмів контролю руху людини виникають нові можливості для розробки імплантованих пристроїв, які можуть взаємодіяти безпосередньо з нервовою системою користувача для кращого керування протезом.

Додатково, урахування психологічних аспектів і впливу протезування на психосоціальну адаптацію користувачів також є важливим напрямком досліджень. Розуміння впливу технологій на психічне здоров'я та якість життя пацієнтів може допомогти у вдосконаленні протезних систем та розробці підтримуючих програм.

Також, розвиток спільноти відкритого програмного забезпечення та вільної апаратури також може сприяти прискоренню інновацій у цій області, роблячи технології більш доступними та сприяючи колективному зусиллю для розв'язання складних завдань.

Нарешті, зростаюча увага до питань етики та безпеки використання технологій у медицині підкреслює необхідність ретельного вивчення впливу апаратно-програмних комплексів на здоров'я та безпеку пацієнтів. Це відкриває нові напрямки для досліджень та вимагає постійного вдосконалення технологій з урахуванням етичних та безпекових аспектів їх використання.

1.2 Огляд-аналіз сучасних трендів у розробці протезів

Аналіз сучасних тенденцій у розробці протезів дозволяє зрозуміти вектор розвитку індустрії та визначити основні особливості, які впливають на створення інноваційних рішень. Сучасні тенденції в розробці протезів охоплюють низку новаторських підходів і технологій, що значно підвищують якість життя людей, які потребують протезування [1]. Далі розглянемо основні напрямки.

Біонічні протези з інтуїтивним управлінням – яскравим прикладом цього напрямку є український стартап Esper Bionics, який спеціалізується на розробці передових біонічних протезів. Орієнтуючись на інноваційні рішення у сфері медичних технологій, компанія створює продукти, що допомагають людям з ампутаціями відновити функціональність та незалежність.

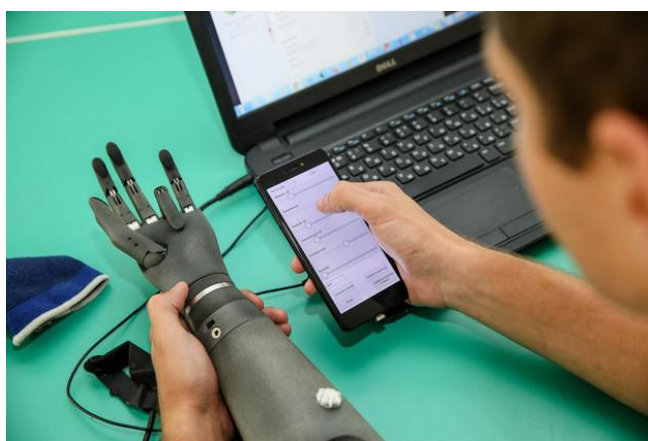


Рисунок 1.1 – Розумний протез руки українського стартапу Esper Bionics

Esper Bionics був заснований у 2019 році з основною місією створювати інтегровані технології для значного покращення якості життя людей з

ампутаціями. Команда, що складається з експертів у біомеханіці, електроніці та штучному інтелекті, розробила біонічний протез руки Esper Hand. Цей протез відзначається легкою вагою, будучи одним із найлегших на ринку, та модульною конструкцією, яка дозволяє швидко замінювати частини і проводити ремонт. Висока функціональність протезу забезпечується 24 різними захватами та жестами, що надає користувачам широкі можливості для виконання повсякденних завдань. Протез можна підключити до смартфона для налаштування та моніторингу, а також використовувати штучний інтелект для адаптації до потреб користувача.

Щодо технології, використовується Esper Platform. Ця платформа дозволяє користувачам інтегрувати протези з іншими пристроями та програмами, що розширює можливості для управління та налаштування.

Сучасні протези не лише виконують свої функції, але також мають естетичний вигляд. Інженери-розробники уникають стандартних тілесних кольорів і створюють протези, які можуть підкреслювати природну красу тіла та бути предметом пишної гордості.



Рисунок 1.2 – Стильний протез

Завдяки технології 3D-друку, протези можуть бути швидко налаштовані та індивідуалізовані під конкретні потреби користувачів. Це також призводить до зниження вартості протезів і спрощує їх обслуговування.

Українські розробники мають на меті створити повний цикл виробництва та сервісного обслуговування протезів на місці, що робить їх більш доступними та зручними для місцевих користувачів.

Протези для кінцівок мають різноманітні типи та функціонал. Наприклад, механічні протези зазвичай складаються з металу та пластику і використовуються для забезпечення опори та допомоги у повсякденних справах.



Рисунок 1.3 – Механічний протез

Міоелектричний протез працює за допомогою м'язових сигналів, які передаються через електроди. Цей вид протезів забезпечує вищу точність і контроль над рухами.

Протез на пневматичній основі працює за рахунок стиснутого повітряного тиску для забезпечення руху. Він характеризується плавними рухами і є зручним у використанні [2].



Рисунок 1.4 – Алюмінієвий пневматичний протез колінного суглоба з чотирма стержнями

Протез на гідродинамічній основі використовує рідину для генерації руху. Цей вид протезів може бути особливо корисним для занять водними видами спорту.



Рисунок 1.5 – Протез ноги на гідродинамічній основі

Ці тенденції свідчать про загальний напрямок розвитку галузі протезування у бік використання більш високотехнологічних, індивідуалізованих і спрямованих на потреби користувачів рішень, що відкриває нові можливості для людей з ампутаціями.

Кожен тип протезу має свої власні характеристики, що ідеально підходять для конкретних потреб та ситуацій користувача. Косметичні протези надають естетичний вигляд, але обмежену функціональність, в той час як біонічні протези забезпечують максимальні можливості управління та адаптації, але мають високі витрати. Механічні та електричні протези пропонують компроміс між функціональністю та вартістю, що робить їх доступними для різних бюджетів і функціональних потреб. У табл. 1.1 розглядаються та порівнюються типи протезів.

Таблиця 1.1 – Порівняльна таблиця типів протезів

Тип протезу	Косметичний протез	Механічний (тілозв'язний) протез	Електричний (міоелектричний) протез	Біонічний протез
Характеристика	Призначений для естетичних цілей, має зовнішній вигляд, схожий на справжню кінцівку	Приводиться в дії рухами тіла, такими як плечовий або грудний пояс [11]	Керується електричними сигналами від м'язів користувача	Високотехнологічний протез з широкими можливостями інтегрований з AI та сенсорами
Матеріал	Силікон, пластик	Метал, пластик, кабелі	Метал, пластик, електроніка	Легкі сплави, пластик, сенсори, електроніка
Функціональність	Мінімальна (без руху)	Середня (обмежені рухи, керовані користувачем)	Висока (кілька захватів, інтелектуальне управління)	Дуже висока (23+ захвати, інтелектуальне управління)
Зручність	Висока, легкий та природній вигляд	Середня, потребує фізичних зусиль для управління	Висока, але потребує навчання для ефективного використання	Дуже висока, адаптується до потреб користувача
Технологічність	Низька, без електронних компонентів	Низька, без електронних компонентів	Висока, використовує датчики та інтерфейси для управління	Дуже висока, адаптується до потреб користувача
Ціна	Низька (\$500-\$3,000)	Середня (\$3,000-\$10,000)	Висока (\$10,000-\$50,000)	Дуже висока (\$30,000-\$100,000+)

Отже, протезна технологія на сьогодні переживає значний розвиток, що сприяє створенню все більш інтегрованих, естетичних та функціональних

рішень. Різноманітні типи протезів відкривають нові можливості для користувачів з різними потребами, починаючи від косметичних до біонічних, механічних і електричних варіантів. Впровадження передових технологій, таких як 3D-друк та міоелектрика, забезпечує персоналізацію і покращення якості життя користувачів. Однак доступність та вартість залишаються ключовими факторами, які впливають на поширення цих технологій серед широкого кола людей. Майбутній розвиток протезів очікується з новими досягненнями в ефективності та доступності для всіх, хто потребує такого обладнання.

1.3 Огляд існуючих систем моделювання руху протезів

Моделювання руху протезів – це процес розробки та налаштування протезів для досягнення максимальної функціональності та природності рухів. Він включає застосування різних технологій і методів для відтворення рухів людської кінцівки та забезпечення оптимального контролю і зворотного зв'язку [3]. Основною метою моделювання є зробити протези більш ефективними, інтуїтивними та зручними для користувачів.



Рисунок 1.6 – Основні компоненти протеза

Системи моделювання руху протезів є невід'ємною частиною сучасної протезної техніки, оскільки вони дозволяють створювати більш ефективні та

комфортні протези для користувачів. Розглянемо основні аспекти існуючих систем моделювання.

Гільзи є ключовим компонентом, що з'єднує куксу пацієнта з рештою частин протеза. Вони повинні бути надійно встановлені, ефективно передавати навантаження і забезпечувати стійкість та керованість.



Рисунок 1.7 – Гільза із захватом сідничної кістки (ІС-гільза)

Манжети, ремені та пояси часто називають допоміжною підвіскою. Це оптимальний спосіб утримання гільзи та запобігання її зсуванню під час перенесення кінцівки. Хоча це вважається "застарілим" методом підвіски, він все ще широко використовується сьогодні завдяки своїй доступності, простоті, надійності та довговічності. Варіанти підвіски можуть варіюватися від простого м'якого ремінця чи манжети до жорсткого бандажу навколо таза.



Рисунок 1.8 – Транстібіальний протез

Процес виготовлення гільзи починається зі зняття мірок і створення гіпсового негативу залишкової кінцівки. Цей негатив заповнюється гіпсом для отримання гіпсового позитиву. Потім позитивний зліпок коригується для оптимізації посадки гільзи в процесі, який називається "моделюванням". Далі гільза ламінується з використанням вуглецевого матеріалу та смоли для створення індивідуальної постійної гільзи. Іноді перед виготовленням постійної гільзи створюється контрольна/тестова/діагностична гільза (зазвичай прозора), що дозволяє клініцисту оцінити прилягання і внести необхідні корективи. Для досягнення найкращої форми і забезпечення зручної та надійної посадки іноді потрібно провести кілька примірок.



Рисунок 1.9 – Зняття мірок для виготовлення гільзи

Зазвичай тестові гільзи виготовляють з пластику. Постійні гільзи мають багат шарову структуру і можуть бути зроблені зі скловолокна, вуглеволокна або нейлону.

Оскільки гільза є зоною контакту між протезом і куксою, її якість, незалежно від обраної моделі, є ключовим аспектом, що визначає комфорт користувача та його здатність керувати протезом. Користувач ніколи не зможе правильно ходити і не досягне цілей реабілітаційної програми, якщо якість посадки гільзи буде незадовільною, незалежно від використовуваного матеріалу (пластик, смола або карбон). Якість підгонки повністю залежить від

роботи протезиста та його здатності забезпечити точні вимірювання під час виготовлення зліпка та належну корекцію гіпсового позитиву для оптимального розподілу навантаження в приймальній гільзі.

Таблиця 1.2 – Порівняння систем моделювання руху протезі

Система	Опис	Переваги	Приклади
Біомеханічні Моделі	Математичні та фізичні моделі механіки тіла.	Реалістичність рухів, врахування анатомії.	OpenSim, AnyBody Modeling System
Міоелектричні Системи	Використовують електричні сигнали від м'язів.	Інтуїтивне управління, адаптація до користувача.	DEKA Arm System, i-limb Ultra
Системи Зворотного Зв'язку	Сенсори для зворотного зв'язку про силу, тиск.	Точність, реалістичне відчуття.	Sensory Feedback Prosthetics, e-dermis
Штучний Інтелект та ML	AI аналізує сигнали та адаптує протез.	Вдосконалення з часом, адаптація до нових задач.	Esper Hand, LUKE Arm
AR/VR Технології	Використовують AR/VR для тренування та моделювання рухів.	Безпечне тренування, налаштування під потреби.	VR Training Systems, Hanger Clinic VOR
Інтегровані Системи	Комбінація кількох технологій для максимальної ефективності.	Комплексний підхід, максимальна функціональність.	Ottobock Myo Plus, Coapt Complete Control

Існує ряд систем моделювання руху протезів, кожна з яких має свої унікальні особливості та переваги. Вибір конкретної системи залежить від індивідуальних потреб користувача, типу ампутації та технічних можливостей протезу. Біомеханічні моделі забезпечують реалістичні рухи, міоелектричні системи – інтуїтивне управління, системи зворотного зв'язку підвищують точність, штучний інтелект та машинне навчання адаптують протези до нових завдань, технології доповненої та віртуальної реальності дозволяють безпечно тренуватися, а інтегровані системи комбінують усі ці підходи для досягнення максимальної функціональності.

1.4 Аналіз ринку протезів верхніх кінцівок

Глобальний ринок протезування верхніх кінцівок є зростаючим сегментом медичної промисловості, який розвивається завдяки технологічному прогресу та збільшенню кількості людей з ампутованими кінцівками внаслідок травм або захворювань. Ринок поділяється на кілька категорій продуктів, включаючи штучне, електричне та косметичне протезування.

Останні роки принесли значні зміни на ринку протезів верхніх кінцівок, відзначені інноваціями та технологічним прогресом. Наприклад, відбувся перехід до міоелектричних протезів, які керуються м'язовими сигналами з залишкової частини кінцівки, забезпечуючи більш природні рухи та кращу функціональність порівняно з традиційними протезами. Додатково, розвиток технології 3D-друку дозволив створювати індивідуальні протези, адаптовані до конкретних потреб і вподобань користувачів.

Згідно зі звітом Strategic Market Research, зростання ринку спричинене трьома основними факторами: збільшенням кількості геріатричних пацієнтів, зростанням захворюваності на рак кісток та інші хронічні захворювання, а також технологічним прогресом. Інший звіт від Allied Market Research підкреслює важливість міоелектричних протезів, які, як очікується, будуть домінувати на ринку завдяки своїм передовим технологіям, що забезпечують більшу мобільність і функціональність для користувачів. Звіт також вказує на роль протезних клінік як найбільших кінцевих споживачів протезів верхніх кінцівок. Прогнозована кількість протезів за видами показана на рис. 1.10.

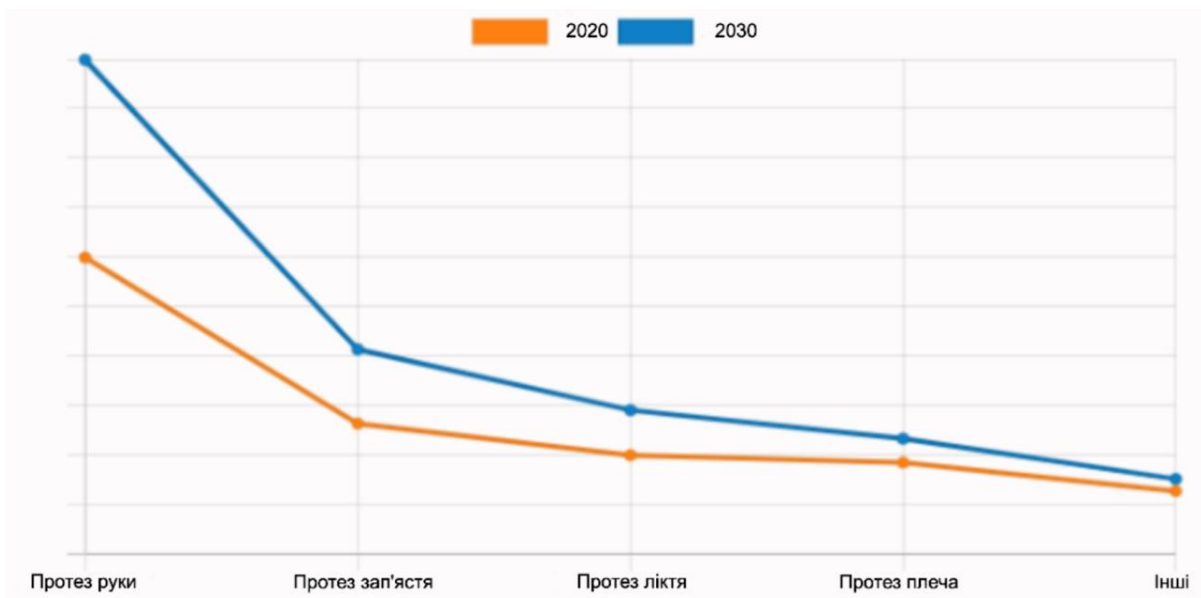


Рисунок 1.10 – Популяції протезів верхніх кінцівок

Наразі існує кілька типів протезів, серед яких можна виділити такі: тягові, міоелектричні, гібридні, косметичні та спеціальні. Зупинимося детальніше на міоелектричних протезах, оскільки саме цей тип буде розроблятися у майбутньому. Міоелектричні або електроміографічні протези керуються електричними сигналами, що генеруються м'язами залишкової частини кінцівки, або електроміографічними (ЕМГ) сигналами [4]. Протези на основі ЕМГ працюють шляхом вимірювання електричної активності м'язів, яка виникає під час руху або напруження м'язів.

З точки зору технологій, біонічні конструкції вважаються новим поколінням пристроїв, здатних безпосередньо взаємодіяти з живими організмами та ефективно відновлювати втрачені функції. Крім біонічних протезів кінцівок, до цієї категорії належать екзоскелети, штучні серця, а також системи для відновлення слуху і зору.

Зважаючи на технологічні обмеження, ринок біонічних пристроїв має високий потенціал зростання: наприкінці 2022 року його обсяг становив приблизно 6,7 млрд доларів. Ключовими чинниками розвитку є:

- досягнення у сфері робототехніки;
- полегшення фінансування;

– збільшення кількості ампутацій та вроджених вад.

За даними компанії DataM Intelligence Analysis, яка спеціалізується на дослідженнях ринку та бізнес-аналітиці, найбільшу частку на ринку біонічних протезів займає Північна Америка завдяки високому рівню технологій. (рис. 1.11)

Світовий ринок біонічних протезів

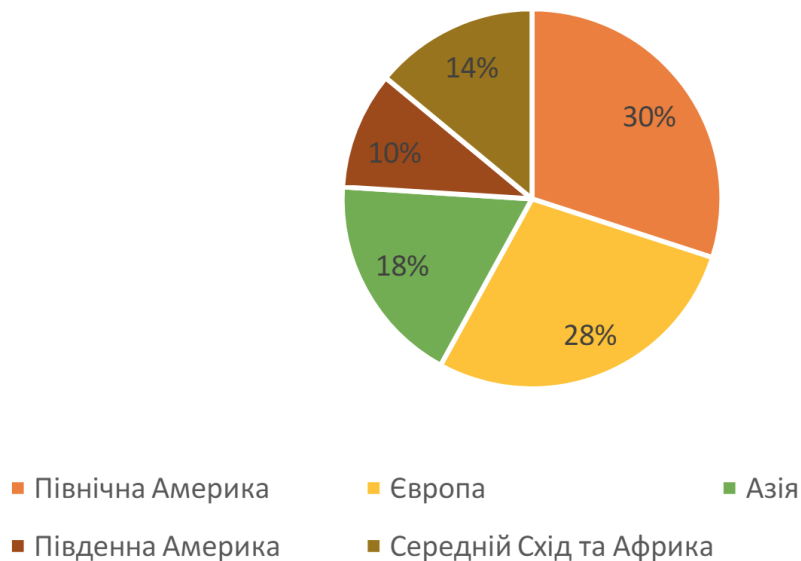


Рисунок 1.11 – Територіальний розподіл світового ринку біонічних протезів

На платформі Google Patents сьогодні можна знайти понад 40 патентів за запитом «біонічний протез». Кількість зареєстрованих патентів у цій сфері зростає щороку.

Провідними розробниками біонічних протезів у світі є Steeper (Великобританія), Ottobock (Німеччина), Endolite (Великобританія), Ossur (Ісландія). В Україні близько 3 мільйонів людей мають різні групи інвалідності, серед них – приблизно 200 тисяч дітей.

Згідно з постановою Кабінету Міністрів України від 5 квітня 2012 року №321, держава зобов'язана забезпечувати осіб з інвалідністю, дітей з інвалідністю та інші окремі категорії населення технічними та іншими

засобами реабілітації, а також виплачувати грошову компенсацію за самостійно придбані технічні та інші засоби реабілітації.

В Україні існує багато центрів протезування та реабілітації, а також інститутів протезування, які співпрацюють зі світовими брендами [5]. Однак розвиток ринку біонічних протезів стримується через низьку платоспроможність населення та залежність від державного фінансування. Це пов'язано з обмеженими планами держави щодо забезпечення осіб з інвалідністю засобами реабілітації. У більшості випадків люди з інвалідністю обирають дешевші косметичні або механічні протези замість біоелектричних через їх високу вартість.

1.5 Вимоги до апаратно- програмного комплексу

Пристрої для відслідковування руху кінцівки людини мають відповідати кільком технічним вимогам для забезпечення точності, надійності та зручності використання.

Датчики повинні мати високу точність і роздільну здатність для виявлення дрібних рухів кінцівок, що є критично важливим для медичних та спортивних застосувань.

Пристрої повинні забезпечувати високу частоту дискретизації (зазвичай більше 100 Гц) для точного відслідковування швидких рухів без затримок.

Наявність безпроводного зв'язку, такого як Bluetooth або Wi-Fi, для передачі даних у реальному часі на комп'ютер або мобільний пристрій для подальшої обробки та аналізу.

Пристрої повинні мати низьке енергоспоживання, щоб забезпечити тривалий час роботи від батареї, особливо у випадку портативних пристроїв.

Компактний і легкий дизайн, що забезпечує комфорт користувача протягом тривалого часу. Кріплення повинні бути ергономічними і не обмежувати природні рухи кінцівок.

Захист від пилу, вологи та механічних пошкоджень, щоб пристрій міг функціонувати в різних умовах, включаючи інтенсивні фізичні вправи або реабілітацію.

Підтримка різних операційних систем (Windows, macOS, Android, iOS) і сумісність з програмним забезпеченням для аналізу даних.

Простий процес калібрування і налаштування для забезпечення точності вимірювань при різних умовах використання.

Забезпечення захисту даних користувача під час передачі та зберігання, відповідність стандартам безпеки.

Наявність програмних інтерфейсів (API) для інтеграції з іншими системами і додатками, що дозволяє розширити функціональність пристрою.

Ось основні технічні характеристики ESP8266:

1) Мікроконтролер і процесор:

- архітектура: 32-бітний RISC процесор Tensilica Xtensa L106;
- тактова частота: 80 МГц (може бути розігнаний до 160 МГц);
- пам'ять: 32 КБ інструкційної пам'яті, 80 КБ користувацької

RAM.

2) Wi-Fi:

- стандарти: IEEE 802.11 b/g/n;
- частота: 2.4 ГГц;
- режими роботи: STA (Station), AP (Access Point), STA+AP;
- захист: WEP, WPA/WPA2.

3) Периферійні інтерфейси:

- GPIO (General Purpose Input/Output): до 17;
- ADC (Analog to Digital Converter): 10-бітний, 1 канал;
- I2C (Inter-Integrated Circuit): програмна реалізація;
- I2S (Inter-IC Sound): підтримка;
- UART (Universal Asynchronous Receiver/Transmitter): до 2;
- SPI (Serial Peripheral Interface): підтримка;
- PWM (Pulse Width Modulation): програмна реалізація.

4) Пам'ять:

– вбудована флеш-пам'ять: зазвичай 4 МБ (моделі можуть мати різні об'єми флеш-пам'яті).

5) Живлення:

- напруга живлення: 3.0V – 3.6V;
- споживання струму: до 170 мА в режимі передачі.

Основні технічні характеристики MPU6050:

1) Акселерометр:

- діапазон вимірювань: $\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$;
- чутливість: від 16384 LSB/g (при $\pm 2g$) до 2048 LSB/g (при $\pm 16g$).

2) Гіроскоп:

- діапазон вимірювань: $\pm 250^\circ/c$, $\pm 500^\circ/c$, $\pm 1000^\circ/c$, $\pm 2000^\circ/c$;
- чутливість: від 131 LSB/($^\circ/c$) (при $\pm 250^\circ/c$) до 16.4 LSB/($^\circ/c$) (при $\pm 2000^\circ/c$).

3) Живлення:

- напруга живлення: 2.375V – 3.46V (VDD);
- напруга інтерфейсу: 1.8V – VDD (VLOGIC).

4) Інтерфейси:

- I2C: Підтримка до 400 кГц;
- SPI: Підтримка до 1 МГц.

5) Споживання струму:

- робочий режим: 3.9 мА (гіроскоп), 500 мкА (акселерометр);
- режим очікування: 5 мкА.

Основні характеристики для різних типів USB-кабелів:

1) USB 2.0:

- швидкість передачі даних: до 480 Мбіт/с (High-Speed);
- напруга і струм живлення: 5V, до 500 мА;
- роз'єми: Тип А, Тип В, Mini-USB, Micro-USB;
- довжина кабелю: до 5 метрів.

2) USB 3.0:

- швидкість передачі даних: до 5 Гбіт/с (SuperSpeed);
- напруга і струм живлення: 5V, до 900 мА;
- роз'єми: Тип А, Тип В, Micro-B, Type-C;
- довжина кабелю: до 3 метрів.

3) USB 3.1:

- швидкість передачі даних: до 10 Гбіт/с (SuperSpeed+);
- напруга і струм живлення: 5V, до 900 мА (в деяких випадках до 3А з Type-C);
- роз'єми: Тип А, Тип В, Type-C;
- довжина кабелю: до 3 метрів.

Основні характеристики для деяких типів елементів живлення:

1) Лужні батареї (Alkaline):

- напруга: 1.5V на елемент;
- ємність: 700 - 3000 мА·год (в залежності від розміру, наприклад, АА, ААА);
- енергетична щільність: 100-150 Вт·год/кг;
- робочий температурний діапазон: -20°C до +54°C;
- саморозряд: 2-3% на рік при кімнатній температурі.

2) Літій-іонні батареї (Li-ion)

- напруга: 3.6-3.7V на елемент;
- ємність: 1000 - 3500 мА·год (для стандартного розміру 18650);
- енергетична щільність: 150-200 Вт·год/кг;
- робочий температурний діапазон: -20°C до +60°C (оптимально 0°C до +45°C);
- саморозряд: 2-5% на місяць;
- циклічний ресурс: 300-500 циклів заряд-розряд.

Ці вимоги забезпечують високу функціональність і універсальність пристроїв для відслідковування руху кінцівок, роблячи їх придатними для широкого спектра застосувань від медицини до спорту і розваг.

Висновок до розділу 1

В першому розділі кваліфікаційної роботи бакалавра було розкрито актуальність теми, адже тема моделювання руху протезів є релевантною в сучасному світі завдяки постійному технологічному прогресу та потребам користувачів з ампутаціями. Це напрямок дозволяє створювати більш ефективні та комфортні протези, що забезпечують природні рухи і підвищують якість життя людей з обмеженими можливостями. Застосування різноманітних технологій, таких як біомеханічні моделі, міоелектричні системи та штучний інтелект, дозволяє досягти значного прогресу в цій галузі, що робить її досить важливою та перспективною для подальшого розвитку і вдосконалення протезної техніки.

Було оглянуто та проведено аналіз різновидів протезів, а також, проаналізовано ринок протезів верхніх кінцівок. Для проведення аналізу розглянуто функціонал, характеристика, зручність, переваги та недоліки протезів та їх розробки. Розділ містить специфікацію вимог до програмного забезпечення, що розробляється.

2 ПРОЄКТУВАННЯ АПАРАТНО-ПРОГРАМНОГО МОДУЛЮ ДЛЯ МОНІТОРИНГУ РУХУ КІНЦІВОК

2.1 Координація людського тіла у пересуванні

Динаміку людського тіла можна описати через прискорення та маси різних його частин, таких як ноги, руки, тулуб і голова. Ці частини можуть демонструвати специфічні моделі координації між собою. Під час ходьби махова нога взаємодіє з верхньою частиною тіла та опорною ногою по-різному, наприклад, у фазі та поза фазою у вертикальному та горизонтальному напрямках відповідно. Такі моделі самопомоги, виявлені в людському пересуванні, можуть бути корисними при проектуванні робототехніки та розробці допоміжних пристроїв для пацієнтів із руховими порушеннями. Це також може пролити світло на деякі незрозумілі аспекти управління моторикою центральної нервової системи (ЦНС) [6]. Самопомога означає, що різні частини тіла сприяють виконанню функцій, необхідних для вирішення основного рухового завдання. Для ефективного використання самопомоги потрібні точні та збалансовані просторово-часові схеми активації м'язів. Необхідна інфраструктура нейронних зв'язків для досягнення такого м'язового контролю присутня в надлишку в спинномозкових схемах. Розуміння пересування всього тіла є ключовим для успішного проектування майбутніх допоміжних пристроїв-протезів, які могли б забезпечити користувачеві оптимальний контроль.

Людське тіло складається з сегментів, з'єднаних суглобами [11]. Ці сегменти включають тверді компоненти, як-от скелет, і м'які тканини, такі як шкіра, м'язи та зв'язки. Це означає, що будь-який рух спричиняє інерційні ефекти, які впливають на розподілені маси всього тіла і тому присутні у більшості рухів. У біологічних системах часто використовувані рухи можна описати низьковимірною шаблонною динамікою, що охоплює рухи всього тіла. Це вказує на те, що нейронний контроль також відіграє активну роль у зв'язку між різними частинами тіла. Оскільки розподіл інерційних ефектів у

тілі сильно відрізняється між видами, індивідами, віком і типом руху, нейронний контролер повинен значною мірою навчатися та адаптуватися до цих змін.

Кінцевим етапом нервової системи для всіх рухових команд є спинний мозок. Він містить багату мережу інтернейронів, здатних виконувати широкий спектр функцій. Спинний мозок підтримує всі форми соматичного моторного контролю нижче шиї і, ймовірно, сприяє тому, що часто використовувані рухи, такі як пересування, можна описати низьковимірною шаблонною динамікою або м'язовою синергією. Значна частина функцій цієї мережі залежить від сенсорного зворотного зв'язку від численних сенсорних аферентів, що передають інформацію про біомеханічний стан тіла. Ці датчики забезпечують миттєву реакцію спинного мозку на зміни стану через локальні схеми або рефлекси, а також інформують центри вищого порядку про умови зовнішнього світу для забезпечення цілеспрямованої поведінки.

2.1.1 Взаємодія між біомеханічними та нейронними системами в управленні всім тілом

Необхідну координацію розподілених мас тіла під час пересування можна проілюструвати простою моделлю. Уявімо дві маси, які вирівняні вертикально і з'єднані лінійними пружинами (рис. 2.1). Ці маси можуть представляти сегменти ніг, тобто стійку ногу та махову ногу, а також тулуб, тобто верхню частину тіла, включаючи голову, руки та тулуб. Наприклад, під час ходьби ми циклічно рухаємо маси вгору-вниз. Навіть якщо жорсткість пружини налаштована майже ідеально, маси з часом почнуть зміщуватися з фази через інерцію та динаміку системи. У біології таке механічне втручання може бути серйозною проблемою для ефективного контролю руху. Щоб проілюструвати цей принцип на прикладі цивільного будівництва, згадаємо про ходьбу по гнучкому мосту з резонансною частотою, близькою до частоти вашого тіла під час ходьби. Ця ситуація може бути дуже важкою для контролю і неприємною, як це сталося під час відкриття мосту Міленіум у Лондоні в

2000 році. Тут бічні коливання мосту були спричинені синхронізацією індивідуальних моделей ходьби, що призвело до необхідності закриття та перепроєктування мосту через небажану синхронізацію [7].

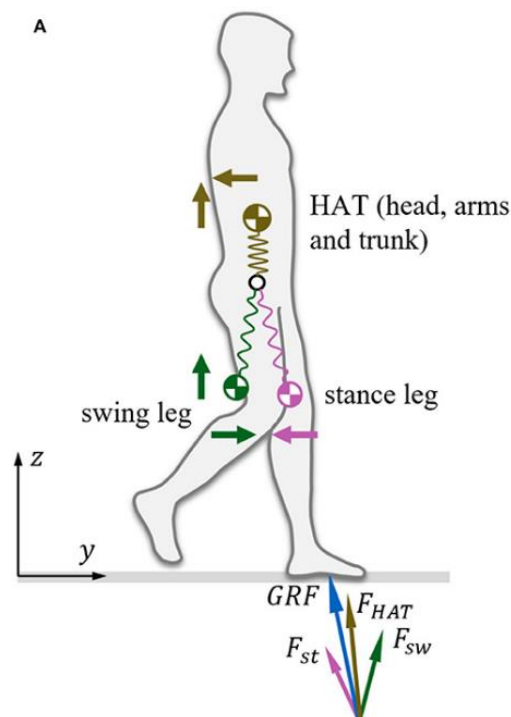


Рисунок 2.1 – Механічний зв'язок між верхньою частиною тіла (голова, руки та тулуб) і нижньою частиною тіла (махова нога та опорна нога) під час ходьби

Щоб синхронізувати рух мас, до пружин можна додати невелику амортизацію. В реальності цей ефект досягається завдяки функціональному демпфуванню м'язів. Однак амортизація може зменшити амплітуду початкових коливань. Для підтримання безперервного руху амортизацію потрібно компенсувати активним контролем м'язів. Змінюючи сигнал активації, просторово-часовий патерн м'язових сил, створюваних ЦНС, можна модулювати так, щоб опорно-руховий апарат залишався в постійному русі. Для циклічних рухів ці нейромеханічні шаблони вимагають належного налаштування нейронною системою [13], що може призвести до скоординованих рухів різних частин тіла, як це спостерігається під час локомоції. Без такого активного внеску в контроль амортизації неможливо

досягти успішного функціонування двох або більше пружинних мас, якщо вони з'єднані з тією самою базою, як у випадку зі вставкою хребця в стегно.

У біологічних системах ЦНС досягає контролю пом'якшення, вивчаючи, коли і які м'язи активувати в необхідному ступені. Для цього ЦНС повинна відстежувати фазу циклічного руху та відповідно оновлювати патерни активації м'язів. Для моніторингу цих фаз і біомеханічних ефектів, створених моделями м'язової активності, використовуються всі датчики, розташовані по всьому тілу. Сенсорна інформація, що надходить до спинного мозку, впливає на стан мережі спінальних інтернейронів і стає ключовим компонентом генератора нейромеханічних патернів. ЦНС має можливість наблизити оптимальні нейромеханічні шаблони для таких дій, як біг, стрибки, ходьба тощо, завдяки навчальному процесу.

Корисна нейронна система [14] повинна дозволяти добре апроксимувати ці моделі нейромеханічної активності або функції з мінімальними витратами на інфраструктуру нейронних схем.

Якщо система прагне мінімізувати використання можливостей нейронної мережі для кожної функції, яку необхідно реалізувати, ЦНС зможе створювати більше патернів рухів або параметрів руху, використовуючи обмежені ресурси. Спинномозкові інтернейрони тісно пов'язані з мозочком, що відіграє важливу роль у навчанні та додаванні інших рухових компонентів або параметрів до моделей рухів для досягнення вищого рівня контролю. Наприклад, підтримання балансу є постійною проблемою в природі, де гладка горизонтальна поверхня, яку часто використовують у лабораторіях, зустрічається рідко. У природних умовах біологічної локомоції інформація від органа рівноваги внутрішнього вуха є однією з функцій, яку необхідно постійно враховувати в генераторі нейромеханічних шаблонів.

2.1.2 Біомеханічні спостереження

Коли тіло рухається, між різними його масами діють інерційні сили. Масу тіла можна визначити як масу будь-якого окремого сегмента тіла.

Вивчення взаємодії та координації верхньої частини тіла, махової ноги та стійкової ноги допоможе нам краще зрозуміти механізми зв'язку між різними масами/частинами тіла під час ходьби. Ключовою особливістю динаміки людської ходьби є те, що махова нога рухається у фазі з верхньою частиною тіла у вертикальному напрямку, але не у фазі з нею в горизонтальному напрямку. Інерційна динаміка в стійковій нозі менш виражена [8]. Ці механізми мають кілька функціональних переваг: вони допомагають створювати інерційні вертикальні сили (що призводить до виникнення реактивних сил землі), зменшують ризик ковзання та роблять швидкість пересування більш стабільною, що допомагає економити енергію.

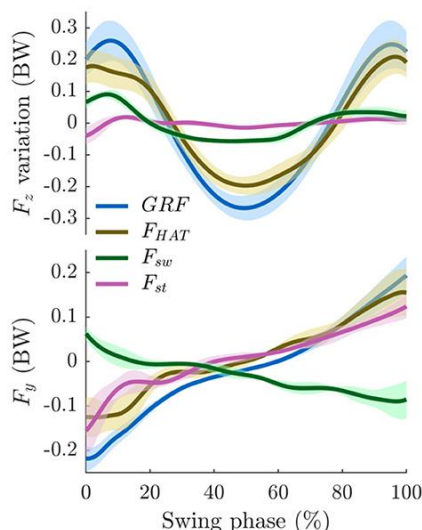


Рисунок 2.2 – Сили нормовані на масу тіла (МТ). F_z і F_y у позначають сили інерції у вертикальному та горизонтальному (перед-назад) напрямках

Ілюстрація внеску рухів ГРТ, махової ноги та стійкової ноги в загальну силу реакцію землі (СРЗ) під час фази ходьби з однією опорою (рис. 2.2), тобто, коли махова нога знаходиться в повітрі. ГРТ і махова нога створюють синхронні М-подібні схеми сил у вертикальному напрямку (F_z на рис. 2.2). Оскільки махова нога повинна рухатися вперед, на все тіло діють інерційні сили. М'язи повинні адекватно реагувати на ці сили, щоб забезпечити необхідне амортизування. Махова нога вносить близько 25% у формування М-

подібної вертикальної компоненти СРЗ (рис. 2.2). Вертикальна сила, створювана стійковою ногою, не має вираженої М-подібної форми, бо залишається майже постійною, що означає, що вона майже не впливає на динаміку ходьби [9]. У горизонтальному напрямку махова нога створює нефазну структуру сил, тоді як стійкова нога створює синфазну схему сил відносно сили ГРТ (F_y на рис. 2.2). Таким чином, сили махової ноги і стійкової ноги в горизонтальному напрямку компенсують одна одну, зменшуючи ризик ковзання. Отже, результати показують, що рух махової ноги не погіршує, а підтримує динаміку ходьби.

Ці висновки показують, як важлива структура біомеханічних шаблонів ходи для ефективного пересування. За допомогою цих шаблонів можна узагальнити динаміку людського тіла, яке має велику кількість ступенів свободи, на моделі з обмеженою кількістю функціональних параметрів. У опорній нозі, з розширеним кутом коліна, жорсткість ноги визначається не тільки м'язами стійкої ноги, але і великою мірою узгодженою дією махової ноги та верхньої частини тіла. Варіації в цьому складі можуть відрізнятися в залежності від напрямку, наприклад, вертикальні проти горизонтальних сил. Махова нога сприяє загальній "жорсткості ніг" [10], яка представлена у моделі перевернутого маятника з навантаженням пружиною, що описує рух центру мас тіла під час пересування та вертикалізації сил реакції землі, як це описано у моделі віртуальної точки обертання. Це пояснює вимоги щодо підтримки постуральної рівноваги, тобто збереження вертикального положення тіла під час руху. Отже, махова нога функціонує як "допоміжна система" для досягнення локомоторних підфункцій, таких як підтримка положення та рівноваги.

2.2 Моделювання руху протезів

На сьогоднішній день розвиток технологій у галузі протезування відіграє важливу роль у поліпшенні якості життя людей з обмеженими можливостями. Сучасні технології відкривають нові можливості для

створення протезів, які можуть точно імітувати рухи людського тіла. Традиційні протези часто стикаються з проблемами поганої керованості та відсутності сенсорного зворотного зв'язку. Брак тактильної сенсорної інформації змушує користувачів протезів покладатися на випадкові візуальні та слухові сигнали, що призводить до дискомфорту та обмежень у повсякденному житті. У той же час, розумні протези, які керуються мікропроцесорами, надають значні переваги користувачам, зокрема підвищену функціональність та кращу адаптацію до щоденного життя.

За допомогою передових технологій та алгоритмів, апаратно-програмний комплекс для моделювання руху буде збирати дані з численних сенсорів, розташованих на протезі. Ця інформація допоможе вдосконалити алгоритми управління протезами, що забезпечить більш природний і інтуїтивний рух.

Традиційна ампутація зображена на рис. 2.3, вона впливає на еферентну та аферентну сигналізацію [12]. Хоча еферентна сигналізація все ще може спричиняти скорочення м'язів, самі м'язи не здатні генерувати природну динаміку кінцівки через хірургічне розташування з метою запобігання невромам і оптимального кріплення протеза. Аферентна пропріоцептивна сигналізація негативно впливає на патологічно змінені точки введення м'язів і втрату дистальних тканин. Цей процес ще більше ускладнюється через такі механізми, як фантомний біль у ампутованій кінцівці.

Ілюстративна м'язова реіннервація та цільова сенсорна реіннервація – це спеціалізовані нервово-м'язові та шкірно-механонейронні інтерфейсні методики, застосовані у людей з ампутацією, спрямовані на керування м'язами та забезпечення соматосенсорного зворотного зв'язку. Під час цільової м'язової реіннервації чужорідні периферичні нерви хірургічно з'єднують з денервованими, але васкуляризованими м'язами. У цільовій сенсорній реіннервації сенсорні нерви або їх пучки підключаються безпосередньо до шкірної тканини або до нервів, що іннервують шкіру.

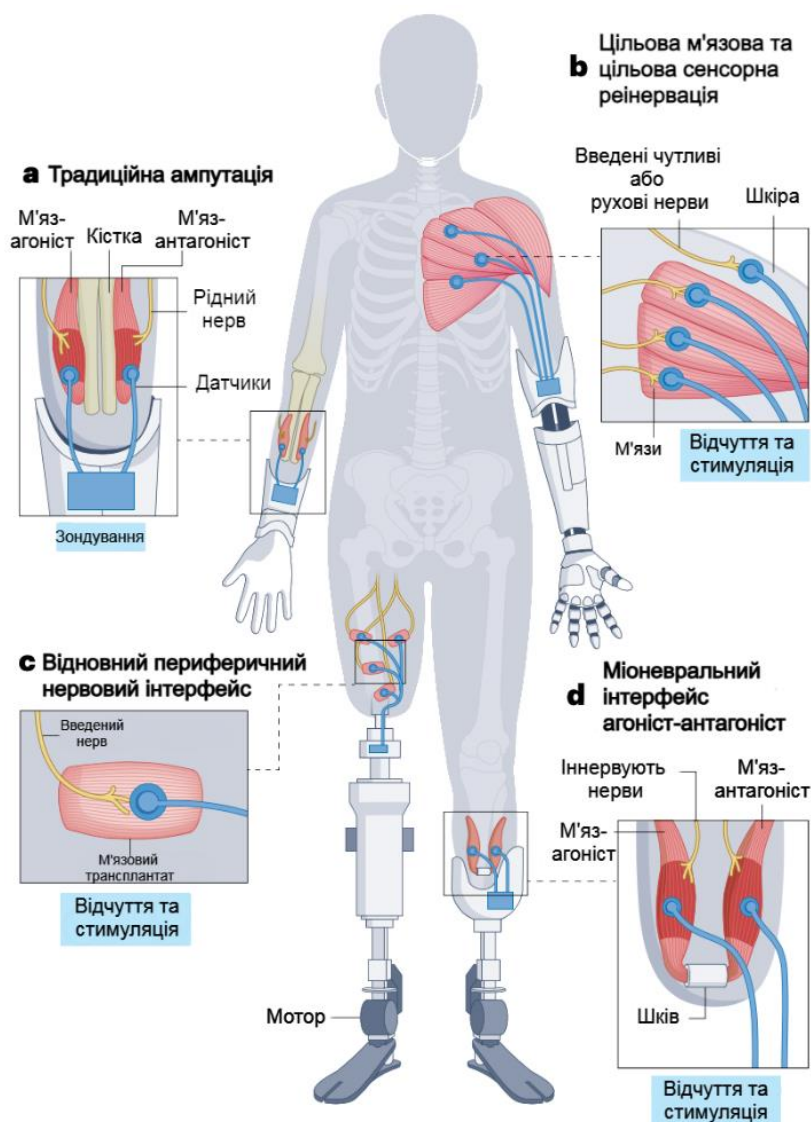


Рисунок 2.3 – Звичайна ампутація та клінічні механонейронні інтерфейси

Під час створення регенеративних периферичних нервових інтерфейсів, денервовані та відсічені від кровоносних судин м'язові трансплантати обгортають перерізані рухові нерви [13]. Пізніше ці трансплантати починають реіннервуватися протягом кількох місяців, з метою функціонального відновлення.

Міоневрологічний інтерфейс агоніст-антагоніст полягає в механічному з'єднанні м'язів, які діють як агоністи і антагоністи, для передачі сигналів як у напрямі від мозку до м'язів (еферентний шлях), так і у зворотному напрямку (аферентний шлях). Це з'єднання дозволяє нативним механорецепторам, таким

як м'язові веретена і органи Гольджі сухожиль, забезпечувати пропріоцептивний зворотний зв'язок через механізми розтягування м'язів і сухожиль відповідно до їхніх функціональних потреб.

За допомогою цієї платформи, користувачі отримують наступні переваги:

- покращене керування: протези стануть більш чутливими до команд користувача, що дозволить забезпечити більш точне і контрольоване використання;
- сенсорний зворотній зв'язок: користувачі зможуть відчувати протез як частину свого тіла, що значно покращить адаптацію та комфорт. Тактильний зворотній зв'язок забезпечить швидший контроль, що сприяє зменшенню компенсаторних реакцій після неочікуваних змін у вазі;
- адаптація до різних умов використання: протези будуть адаптуватися до різних умов, що забезпечить їх високу ефективність у будь-яких ситуаціях.

Міоневрологічний інтерфейс агоніст-антагоніст дозволяє передавати сигнали між мозком і м'язами в обох напрямках, використовуючи механічні з'єднання м'язів. Це забезпечує покращене керування протезами, сенсорний зворотний зв'язок, і адаптацію до різних умов, роблячи протези більш чутливими, точними та комфортними у використанні. Користувачі відчують протез як частину свого тіла, що значно покращує адаптацію та зменшує компенсаторні реакції.

2.2.1 Фільтрація сигналів

Фільтр Савицького-Голея – це цифровий фільтр, який можна застосувати до набору цифрових точок даних з метою згладжування даних, тобто підвищення точності даних без спотворення тенденції сигналу. Це досягається в процесі, відомому як згортка, шляхом підбору послідовних підмножин сусідніх точок даних за допомогою полінома низького степеня за методом лінійних найменших квадратів. Коли точки даних розташовані на

однаковій відстані, можна знайти аналітичний розв'язок рівнянь найменших квадратів у вигляді єдиного набору "коефіцієнтів згортки", які можна застосувати до всіх підмножин даних, щоб отримати оцінки згладженого сигналу (або похідних згладженого сигналу) в центральній точці кожної підмножини. Метод, заснований на усталених математичних процедурах, був популяризований Абрахамом Савицьким і Марселем Ж. Е. Голеєм, які опублікували таблиці коефіцієнтів згортки для різних поліномів і розмірів підмножин у 1964 р. Деякі помилки в таблицях були виправлені. Метод був поширений на обробку 2- і 3-вимірних даних.

Дані складаються з набору точок $\{x_j, y_j\}; j = 1, \dots, n$, де x_j є незалежною змінною і y_j є спостережуваним значенням. Вони обробляються за допомогою набору m коефіцієнтів згортки, C_j згідно з виразом

$$Y_j = \sum_{j = \frac{1-m}{2}}^{\frac{m-1}{2}} C_j Y_{j+i}, \quad \frac{m+1}{2} \leq j \leq n - \frac{m-1}{2} \quad (2.1)$$

Наприклад, для згладжування 5-точковим квадратичним поліномом, $m = 5$, $i = -2, -1, 0, 1, 2$ та j^{th} згладжена точка даних, Y_j задається формулою

$$Y_j = \frac{1}{35} (-3y_{j-2} + 12y_{j-1} + 17y_j + 12y_{j+1} - 3y_{j+2}) \quad (2.2)$$

де $C_{-2} = -3/35$, $C_{-1} = 12/35$

Існує багато застосувань згладжування, наприклад, для уникнення розповсюдження шуму через ланцюжок алгоритмів, а іноді просто для того, щоб дані виглядали менш зашумленими, ніж вони є насправді.

Програма бібліотека `mljs/savitzky-golay` використовується для застосування фільтру Савицького-Голея, який базується на масиві значень у (data) та різниці між точками x (h). Вона дозволяє згладжувати шум у даних, зберігаючи важливі особливості сигналу, такі як піки та впадини. Опція

windowSize визначає кількість точок, що використовуються для оцінки фільтрації, за замовчуванням її значення – 5.

Цей фільтр планується у використанні згладжування шумів сигналу з сенсора MPU6050.

Висновок до розділу 2

У другому розділі було здійснено огляд координації людського тіла під час пересування з метою освоєння сегментації людського тіла та дослідження центральної нервової системи людини. Розглянуто взаємодію між біомеханічними та нейронними системами в управлінні тілом, оскільки ефективна нейронна система має забезпечувати точну апроксимацію моделей нейромеханічної активності або функцій з мінімальними витратами на інфраструктуру нейронних схем. Також було проаналізовано традиційні ампутації та їхній вплив на еферентну та аферентну сигналізацію.

Було детально вивчено процеси, що відбуваються в центральній нервовій системі під час руху, а також способи відновлення функцій через інтеграцію біонічних протезів. Проаналізовано сучасні підходи до сегментації людського тіла для оптимального управління протезами, зокрема методи вимірювання та обробки сигналів, що надходять від нервових закінчень.

3 РОЗРОБЛЕННЯ ВЕБПЛАТФОРМИ МОДЕЛЮВАННЯ РОБОТИ ПРОТЕЗІВ

3.1 Технології та інструменти для розробки апаратно-програмного комплексу моделювання руху протезів

Для розробки апаратно-програмного комплексу моделювання руху протезів використовуються різноманітні технології та інструменти. Розглянемо три ключові технології, які застосовуються у цій галузі.

Для даної роботи було обрано модуль MPU6050 (рис. 3.1), який включає в себе два основних компонента: гіроскоп (гіроскопічний датчик) і акселерометр (акселерометричний датчик).

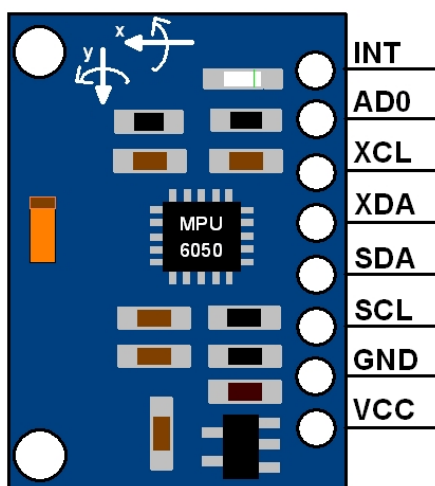


Рисунок 3.1 – Модуль MPU6050

Цей модуль виробляється фірмою InvenSense (тепер частиною компанії TDK) і має наступні ключові характеристики:

- гіроскопічний датчик вимірює швидкість обертання або рух навколо трьох осей простору (відносно X, Y та Z). Гіроскоп дозволяє визначати кутову швидкість або зміну кута;
- акселерометричний датчик вимірює прискорення або поступальний рух по трьом взаємно перпендикулярним осям (X, Y та Z). Акселерометр

дозволяє визначати прискорення, що діє на пристрій, наприклад, від гравітації або від ударів;

- MPU6050 підтримує зв'язок з мікроконтролером або іншими пристроями через шину I2C (Inter-Integrated Circuit), що робить його досить зручним для використання з платами Arduino [21] та іншими мікроконтролерами;

- в деяких версіях MPU6050 також можуть бути вбудовані додаткові функції, такі як температурний датчик або можливість вимірювати кут нахилу (та ховзаня) в просторі.

MPU6050 часто використовується в різних застосунках, де потрібно вимірювати орієнтацію та рух пристрою. Такі застосування включають гіроскопи для контролю руху в ігрових контролерах, пристроях віртуальної реальності, дронів, робототехніки та біомедичних пристроїв.

Модуль MPU6050 оснащений датчиками двома різними типами датчиків, що здатні виявляти прискорення і повороти.

Акселерометр вимірює прискорення, тобто зміну швидкості за одиницю часу. У фізиці зміна швидкості з часом ($a = dV / dt$) є визначенням прискорення. Відповідно до Другого закону Ньютона, $a = F / m$, і це те, що акселерометри використовують для роботи, тобто вони використовують параметри сили та маси об'єкта. Для того, щоб це можна було впровадити в електроніці, використовуються методи MEMS (Micro Electro Mechanical Systems), які відрізняються від звичайних технологій виготовлення електронних чіпів, оскільки механічні деталі створюються в MEMS. У цьому випадку створюються доріжки або елементи, здатні вимірювати прискорення. Це означає, що можна взяти багато інших одиниць, таких як швидкість (якщо прискорення інтегрується у часі), якщо воно інтегрується знову – відбудеться переміщення тощо. Тобто параметри, щоб знати положення або виявити рух об'єкта.

Гіроскоп це пристрій, який вимірює кутову швидкість об'єкта, тобто кутове переміщення за одиницю часу або швидкість обертання тіла навколо

своєї осі. У цьому випадку для вимірювання зазначеної швидкості також використовуються методи MEMS з використанням ефекту, відомого як Коріоліс. Завдяки цьому можна виміряти кутову швидкість або, інтегруючи кутову швидкість щодо часу, можна отримати кутове зміщення.

Модуль є інерційною одиницею вимірювання або IMU (Інерційні вимірювальні одиниці) з 6 ступенями свободи (DoF). Це завдяки 3-осьовим датчикам акселерометра та 3-осьовому гіроскопу, який він застосовує для роботи.

Розташування пінів показано на рис. 3.2.

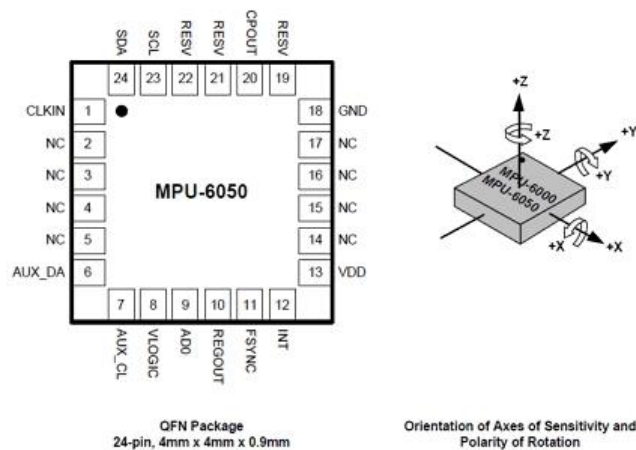


Рисунок 3.2 – Розпіновка модулю MPU6050

Зв'язок I2C може бути сумісним з більшістю мікроконтролерів, включаючи Arduino. Виводи SCL та SDA мають на платі підтягуючий резистор для прямого підключення до плати Arduino [21].

Модуль MPU6050 (MPU-6050) взаємодіє з середовищем розробки Arduino IDE через використання спеціальної бібліотеки, яка дозволяє отримувати дані з акселерометра і гіроскопа, що входять до складу MPU6050.

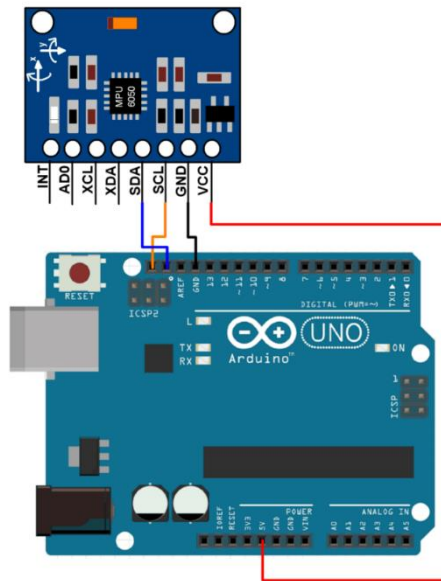


Рисунок 3.3 – Інтеграція з Arduino

Для відтворення програмного коду і написання алгоритму, використовувалося середовище розробки Arduino IDE (рис. 3.3).



Рисунок 3.4 – Старт-скрін Arduino IDE

Елементи, які входять до складу середовища розробки, включають текстовий редактор, консоль з інформацією щодо прошивки плати, текстову

консоль, завантажені бібліотеки та необхідний інструментарій, доступний у головному меню для відлагодження та налаштувань.

Для початку необхідно з'єднати модуль MPU6050 з платою Arduino. Зазвичай використовуються піни SDA (для передачі даних) і SCL (для зміни клоку), а також живлення (VCC) і земля (GND). А також необхідно дозавантажувати бібліотеки необхідної плати(рис. 3.5).

"Бібліотека" – це набір програмних компонентів, які виконують загальні або спеціалізовані завдання, що звільняють програміста від необхідності створювати код з нуля під час розробки програмного забезпечення. Вона зазвичай включає в себе набір функцій для виклику та класів об'єктів, які можна інстанціювати. Наприклад, загальною може бути бібліотека для роботи з датами та часом, а спеціалізованою – програмне забезпечення (ПЗ) для управління розумним термостатом, яке часто постачається з самим пристроєм.

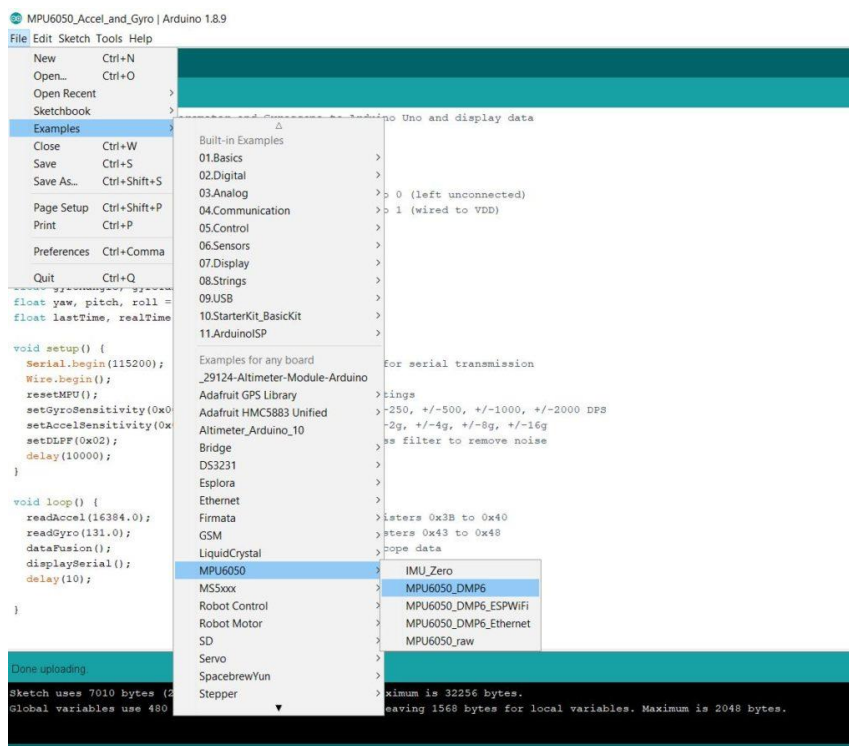


Рисунок 3.5 – Вибір відповідної плати

Бібліотеки включаються до програми за допомогою директиви #include, яка використовується в мовах програмування, таких як C і C++. Використання

бібліотек дозволяє програмісту сконцентруватися на унікальних аспектах розробки програми.

Стандартний набір функцій, що надає середовище програмування, включає наступне:

- `setup()` виконується один раз, в самому початку програми, і використовується для ініціалізації мікроконтролера перед роботою основної програми;
- `loop()` починає виконуватися після завершення `setup()` і виконується постійно, доки не зупиниться програма;
- `pinMode()` встановлює режим роботи вказаного піна як вхід або вихід;
- `digitalRead()` зчитує стан вказаного цифрового піна (HIGH або LOW);
- `digitalWrite()` встановлює стан вказаного цифрового піна (HIGH або LOW).

Стандартний скетч який автоматично генерується при запуску Arduino IDE (рис. 3.6).

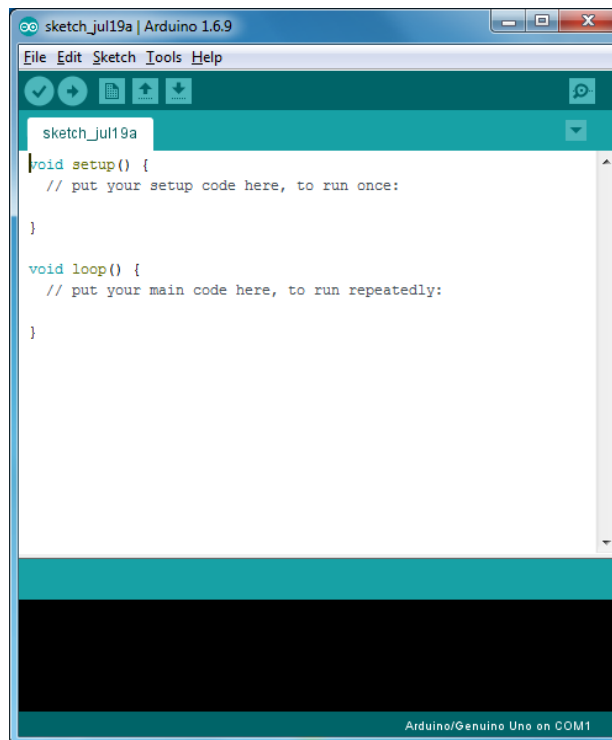


Рисунок 3.6 – Скетч програми у середовищі Arduino IDE

Після ознайомлення з середою розробки програмного забезпечення для апаратно-програмного модулю, треба вирішити яку мову програмування буде використаний у написанні алгоритму.

3.2 Мова програмування

Мови програмування дозволяють нам створювати ефективні програми і розробляти онлайн-рішення, такі як мобільні застосунки, веб застосунки, онлайн трансляції і розробка IoT пристроїв. Основна мета програмування полягає в автоматизації, підтримці, зборці, вимірюванні та інтерпретації обробки даних і інформації. Це базується на парадигмах об'єктно-орієнтованого програмування і враховує різноманітні варіанти з відповідним інструментарієм для вирішення конкретних задач і зручності написання коду. Напередодні обговоримо мови, що включають потрібний набір бібліотек для прискорення роботи алгоритмів.

Java відома своєю високою надійністю як мова програмування. Завдяки підтримці багатопотоковості, Java особливо підходить для послуг, що вимагають безпечної передачі даних клієнту з мінімальним ризиком втручання третьої сторони. Наприклад, логіка передачі медіаданих до месенджерів часто реалізована з використанням технологій, що базуються на Core Java. На платформі Android, яка також використовує Java API, це особливо важливо. Більшість алгоритмів у фондовому ринку також базуються на Java.

Фреймворк MapReduce Hadoop, відомий своєю потужністю, був реалізований з використанням Java. Java активно використовується зі сторонніми фреймворками, такими як Spring, що створює надійне поєднання для створення та розгортання застосунків, особливо в областях фінансів та IT. Відмінним прикладом є «ПриватБанк», де значна частина інфраструктури написана на Java.

Java має схожість з мовами програмування, такими як C та C++, але вона простіша в освоєнні. Java взаємодіє з популярними сучасними технологіями, такими як Spring, Node.js, Android, Hadoop, J2EE та інші.

Після Java, варто звернути увагу на мову програмування C++. Не таємниця, що більшість популярних і потужних програм розробляються саме на C++, з використанням наявних бібліотек або інших мов програмування.

Мова C++ є об'єктно-орієнтованою мовою програмування, створеною визначним комп'ютерним вченим Бьорном Страустропом як частина еволюції сімейства мов C. Вона була спроектована як крос-платформене розширення C з метою надання розробникам більшого контролю над пам'яттю та системними ресурсами.

C++ іноді називають "C з класами", оскільки вона впроваджує об'єктно-орієнтовані принципи програмування, додаючи визначення класів до структури мови програмування C. З часом C++ залишається надзвичайно корисною мовою не лише в комп'ютерному програмуванні, але й у навчанні нових програмістів основам об'єктно-орієнтованого програмування. Розуміння ООП дозволяє використовувати ці знання для написання більш ефективного і швидкого коду, ніж на інших мовах програмування. C++ застосовується для розробки різноманітних програмного забезпечення, включаючи браузері, графічні інтерфейси користувача (GUI), операційні системи та ігри. Також цією мовою розробляється програмне забезпечення для вбудованих систем.

Сьогодні C++ високо цінується за його значну переносимість, що дозволяє розробникам створювати програми, які легко працюють на різних операційних системах та платформах. Незважаючи на те, що це мова високого рівня, C++ залишається близьким до C і дозволяє здійснювати маніпуляції на низькому рівні завдяки своєму тісному зв'язку з машинною мовою. Це особливо важливо для розробки апаратно-програмних модулів, що використовують медіадані.

Python – це високорівнева, інтерпретована, інтерактивна та об'єктно-орієнтована мова програмування. Вона була розроблена Гвідо ван Россумом і вперше випущена у 1991 році і використовується в різних областях

програмування, таких як штучний інтелект, ігри, мікроконтролери та, звичайно, IoT.

Python – це універсальна мова програмування з багатьма парадигмами, яка придатна для різних типів проектів, які не обов'язково пов'язані з апаратно-програмними модулями. Вона підтримує будь-який стиль програмування і ідеально підходить для веб-розробки, настільних і серверних застосунків.

У відміню від високорівневих мов, таких як C++ або Java, Python спеціалізується на реалізації бізнес-логіки безпосередньо на рівні пристрою. Використання цієї мови дозволяє зменшити об'єм даних, з якими необхідно працювати, особливо у хмарних середовищах.

Однією з найбільших переваг Python є підтримка різноманітних зовнішніх бібліотек, включаючи ті, що використовуються в IoT. Популярне середовище розробки для Python, PyCharm, спрощує процес додавання бібліотек, будь то Amazon Redshift, JSON або XML файл.

Сокети і протокол MQTT є відмінними варіантами для Node.js, особливо для передачі даних у програмах IoT. Кабельні платформи, такі як Intel Edison, BeagleBone Black і Raspberry Pi, легко підтримують середовище програмування Node.js. Node.js поставляється з менеджером пакетів NPM, який містить багато корисних модулів IoT, що дозволяє швидко і надійно розробляти застосунки. Початок найшвидшого зростання використання Node.js у сфері IoT відзначено у 2012 році.

Для ефективного та швидкого встановлення та управління модулями в Node.js ідеально підходить NPM, який автоматизує процес встановлення необхідних пакетів і надає зручний інтерфейс для їх управління.

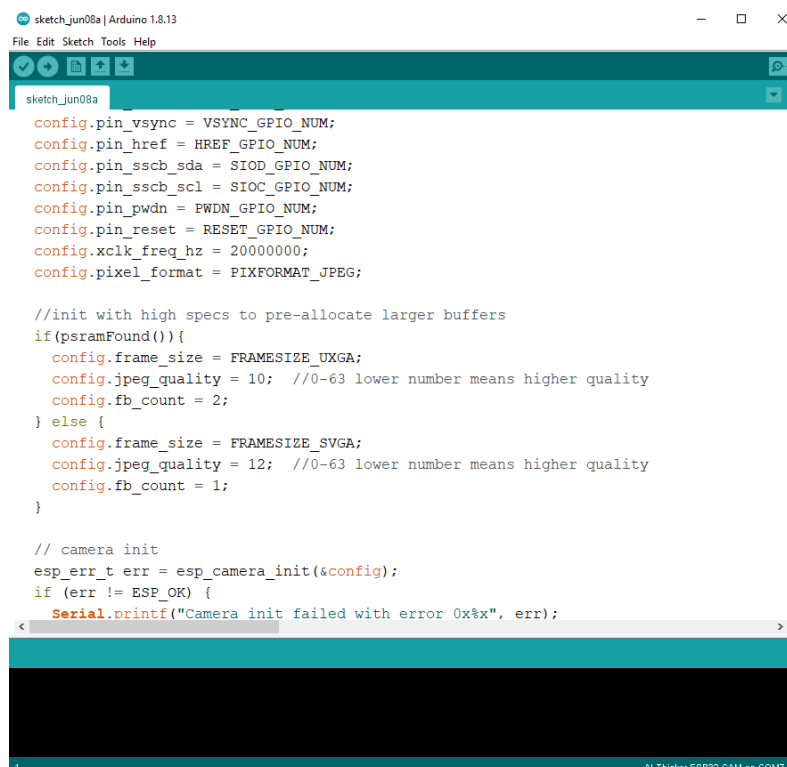
NPM містить безліч популярних модулів IoT, що підходять для різноманітних типів розробки застосунків. Ось декілька найпопулярніших модулів для IoT, які підтримуються NPM:

- Cylon.js;
- Node-RED;

- Noduino;
- Johnny-Five;
- Nodebots.

Основаючись на отриманій інформації, було обрано використовувати мову програмування C++ для написання алгоритму, який буде завантажений до апаратно-програмного модулю MPU6050.

Мови зі статичною типізацією дозволяють зрозуміти типи даних, їх використання і залежності від контексту значно простіше (рис 3.7).



```
sketch_jun08a | Arduino 1.8.13
File Edit Sketch Tools Help

sketch_jun08a
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = FWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;

//init with high specs to pre-allocate larger buffers
if(psramFound()){
  config.frame_size = FRAMESIZE_UXGA;
  config.jpeg_quality = 10; //0-63 lower number means higher quality
  config.fb_count = 2;
} else {
  config.frame_size = FRAMESIZE_SVGA;
  config.jpeg_quality = 12; //0-63 lower number means higher quality
  config.fb_count = 1;
}

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
  Serial.printf("Camera init failed with error 0x%x", err);
}
```

Рисунок 3.7 – Мова програмування C++ для Arduino IDE

Перевага мови C++ полягає в її здатності забезпечувати розробникам ідеальний контроль над пам'яттю і можливістю писати код для платформ з обмеженими ресурсами. Також важливою перевагою є наявність великої кількості створених бібліотек, які значно спрощують написання високопродуктивного коду.

3.3 Розроблення апаратного і програмного забезпечення

З огляду на попередні розділи, можна зрозуміти, що головною метою дипломної роботи є створення системи для моделювання руху протезів. З висновків другого розділу можна зрозуміти, що буде використаний датчик акселерометра-гіроскопа MPU-6050.

Гіроскоп вимірює швидкість обертання (рад/с), це зміна кутового положення з плином часу вздовж осей X, Y і Z (крен, тангаж і поворот) (рис. 3.8). Це дозволяє визначити орієнтацію об'єкта.

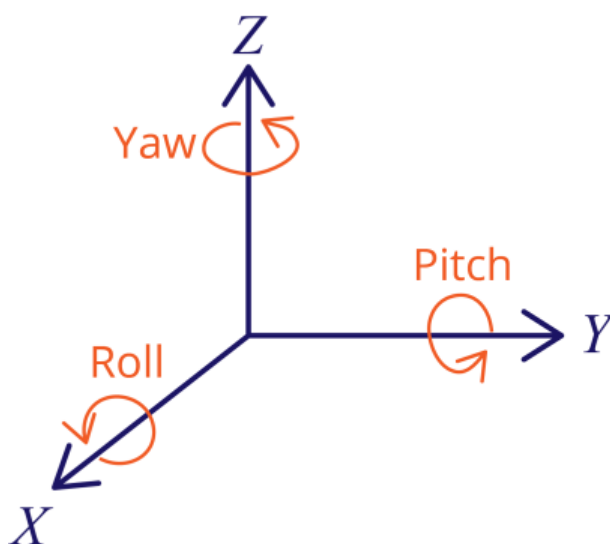


Рисунок 3.8 – Система координат

Акселерометр вимірює прискорення, що представляє собою швидкість зміни швидкості об'єкта. Він реагує на як статичні сили, такі як сила тяжіння ($9,8 \text{ м/с}^2$), так і динамічні сили, наприклад вібрацію або рух. MPU-6050 вимірює прискорення вздовж осей X, Y та Z. В ідеальному статичному об'єкті прискорення по осі Z дорівнює силі тяжіння і дорівнює нулю по осях X і Y.

Використовуючи дані з акселерометра, можна обчислити кути крену та тангажу за допомогою тригонометрії. Однак обчислити поворот неможливо.

Комбінуючи інформацію з обох датчиків, можна отримати більш точну інформацію про орієнтацію датчика. Нижче наведена табл. 3.1 з розпіновкою сенсорного модуля MPU-6050.

Таблиця 3.1– Розпіновка MPU-6050

VCC	Живлення датчика (3,3 В або 5 В)
GND	Загальний GND
SCL	Вивід SCL для зв'язку I2C (GPIO 5)
ПДР	Вивід SDA для зв'язку I2C (GPIO 4)
XDA	Використовується для підключення інших датчиків I2C до MPU-6050
XCL	Використовується для підключення інших датчиків I2C до MPU-6050
AD0	Використовуйте цей пін-код, щоб змінити адресу I2C
INT	Пін переривання – може використовуватися для вказівки на наявність нових даних вимірювання

Ми запрограмуємо плату ESP8266 NodeMCU за допомогою Arduino IDE. Існують різні способи отримання показань датчика. Буде використовуватися бібліотека Adafruit MPU6050. Щоб використовувати цю бібліотеку, також потрібно інстальювати бібліотеку Adafruit Unified Sensor і Adafruit Bus IO Library.

Щоб встановити бібліотеку потрібно відкрити Arduino IDE [21], перейти до Sketch, потім Include Library та Manage Libraries. Має відкритися Менеджер бібліотеки. На рис. 3.9, рис. 3.10 та рис. 3.11 продемонстровано пошук та встановлення бібліотек.

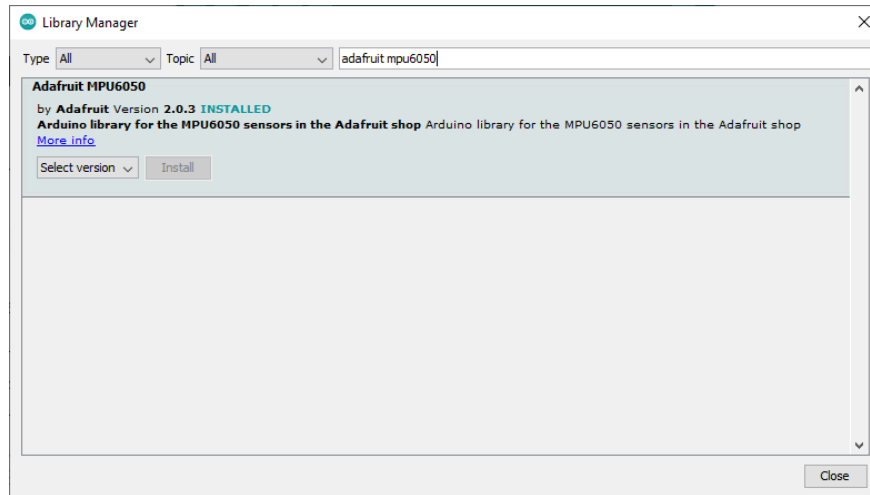


Рисунок 3.9 – Встановлення бібліотеки Adafruit MPU6050

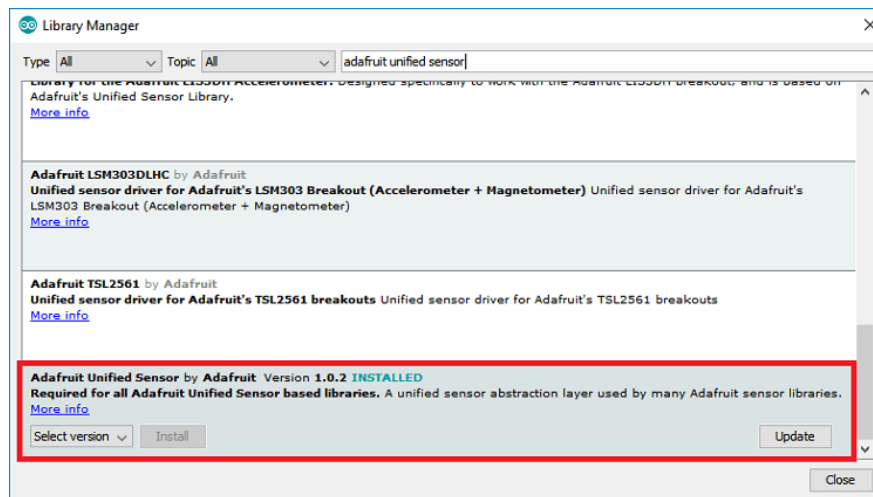


Рисунок 3.10 – Встановлення бібліотеки Adafruit Unified Sensor

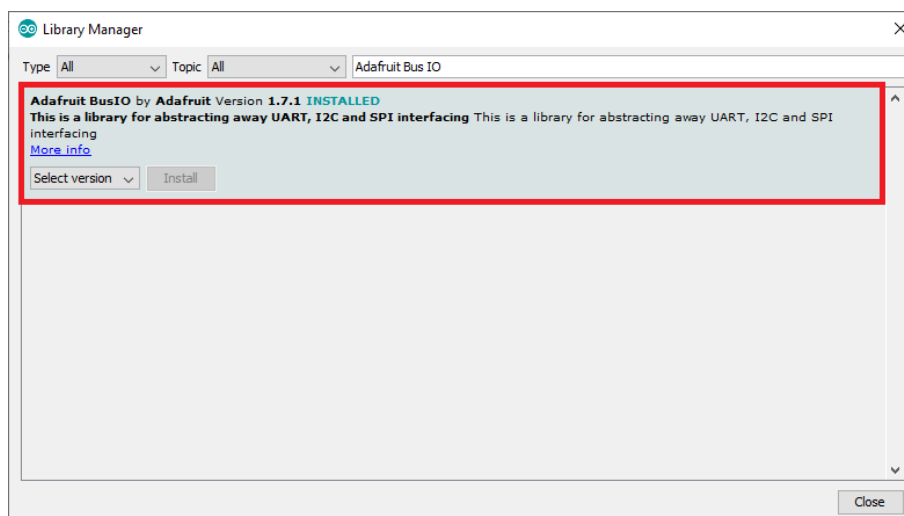


Рисунок 3.11 – Встановлення бібліотеки Adafruit Bus IO

Після встановлення бібліотек необхідно перезапустити Arduino IDE. Наступним кроком необхідно підключити ESP8266 до датчика MPU-6050, а саме підключити контакт SCL до GPIO 5 і прикріпити SDA до GPIO 4 як зображено на рис. 3.12.

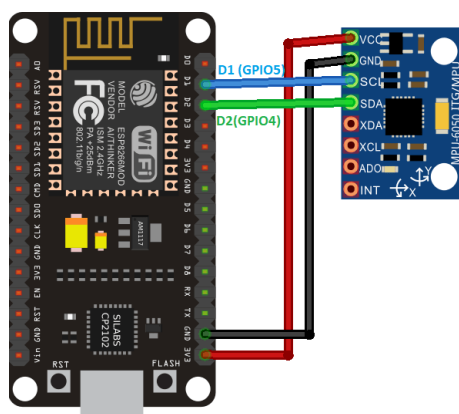


Рисунок 3.12 – ESP8266 NodeMCU з MPU-6050

Далі розглянемо як працює код, який є частиною програми для мікроконтролера ESP8266, що використовується для підключення до Wi-Fi, виконання HTTP-запитів і роботи з датчиком акселерометра/гіроскопа MPU6050. На рис. 3.13 зображено підключення бібліотек.

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>
#include <WiFiClientSecureBearSSL.h>
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
```

Рисунок 3.13 – Підключення бібліотек у кодї програми

Бібліотека ESP8266WiFi.h дозволяє мікроконтролеру ESP8266 підключатися до Wi-Fi мережі.

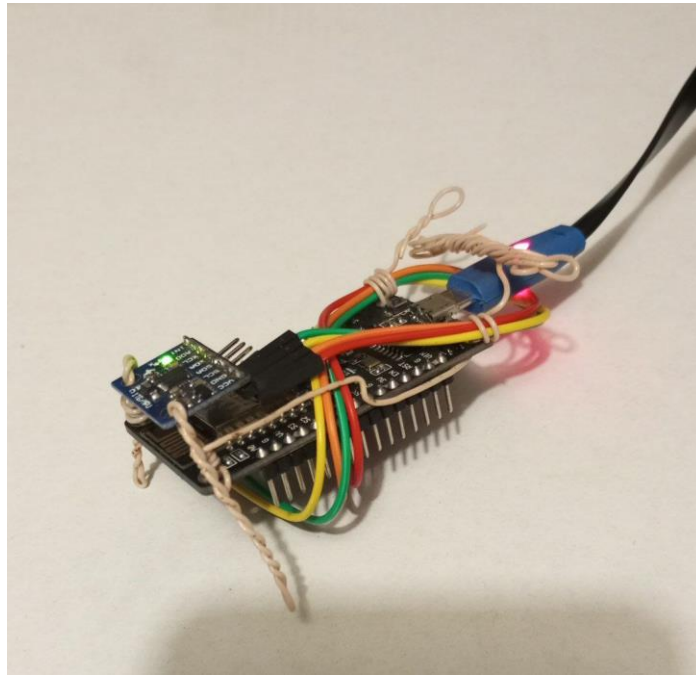


Рисунок 3.14– Фізичне підключення ESP8266 NodeMCU з MPU-6050

Бібліотека `WiFiClient.h` використовується для створення клієнта, який може підключатися до серверів через Wi-Fi.

Бібліотека `ESP8266HTTPClient.h` використовується для виконання HTTP-запитів (GET, POST тощо) з використанням ESP8266. Вона спрощує роботу з протоколом HTTP.

Бібліотека `ArduinoJson.h` використовується для парсингу та створення JSON-даних, що зручно при обміні даними з веб-серверами [21].

Бібліотека `WiFiClientSecureBearSSL.h` використовується для створення безпечного Wi-Fi клієнта з використанням SSL/TLS, що забезпечує захищене з'єднання для передачі даних.

Бібліотека `Adafruit_MPU6050.h` бере участь у роботі з датчиком MPU6050, який включає в себе акселерометр та гіроскоп. Вона надає інтерфейс для зчитування даних з датчика.

Загальна бібліотека `Adafruit` для роботи з різноманітними датчиками. Вона забезпечує стандартний інтерфейс для зчитування сенсорних даних.

Бібліотека `Wire.h` створена для роботи з шиною I2C, яка використовується для підключення та спілкування з різними периферійними пристроями, такими як MPU6050.

Цей набір бібліотек дозволяє мікроконтролеру ESP8266 підключатися до Wi-Fi, взаємодіяти з веб-серверами, обробляти JSON-дані та зчитувати дані з датчика MPU6050.

Наступний фрагмент коду (рис. 3.15) налаштовує змінні для підключення до Wi-Fi мережі та взаємодії з веб-сервером. Об'єкт `mpu` буде використовуватися для зчитування даних з датчика MPU6050. Надалі, код, буде включати логіку для підключення до Wi-Fi, збирання даних з датчика та відправки цих даних на зазначений сервер.

```
Adafruit_MPU6050 mpu;  
  
const char *ssid = ""; //Enter your WIFI ssid  
const char *password = ""; //Enter your WIFI password  
const char *server_url = ""; // Nodejs application endpoint  
const char *sensor_id = "SEN-1";
```

Рисунок 3.15 – Налаштування змінних для підключення до Wi-Fi

`Adafruit_MPU6050 mpu;` – створює об'єкт `mpu` для взаємодії з датчиком MPU6050 за допомогою бібліотеки `Adafruit`.

`const char *ssid = "";` – змінна для зберігання SSID (імені) Wi-Fi мережі, до якої підключатиметься ESP8266. Потрібно заповнити значенням SSID Wi-Fi мережі.

`const char *password = "";` – змінна для зберігання пароля до Wi-Fi мережі. Потрібно заповнити значенням пароля Wi-Fi мережі.

`const char *server_url = "";` – змінна для зберігання URL-адреси серверу (наприклад, Node.js застосунок), до якого ESP8266 надсилатиме дані. Потрібно заповнити значенням URL сервера.

`const char *sensor_id = "SEN-1";` – змінна для зберігання ідентифікатора сенсора. Це може бути використано для ідентифікації сенсора, якщо кілька сенсорів надсилають дані до одного сервера.

Наступна частина коду (рис. 3.16) налаштовує безпечне з'єднання Wi-Fi, ініціалізує датчик MPU6050, та виводить результати ініціалізації в серійний монітор.

```
std::unique_ptr<BearSSL::WiFiClientSecure>client(new BearSSL::WiFiClientSecure);
HTTPClient http;

void setup() {
  delay(3000);
  Serial.begin(115200);

  while (!Serial)
    delay(10); // will pause Zero, Leonardo, etc until serial console opens

  Serial.println("Adafruit MPU6050 test!");

  // Try to initialize!
  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1) {
      delay(10);
    }
  }
  Serial.println("MPU6050 Found!");
```

Рисунок 3.16 – Безпечне з'єднання з Wi-Fi

Розглянемо код детальніше:

3) `std::unique_ptr<BearSSL::WiFiClientSecure>client(new BearSSL::WiFiClientSecure);` – створює об'єкт `client` типу `WiFiClientSecure` для встановлення безпечного з'єднання через SSL/TLS. Використовується `std::unique_ptr`, щоб забезпечити автоматичне управління пам'яттю і уникнути витоків пам'яті;

4) `HTTPClient http;` – створює об'єкт `http` типу `HTTPClient`, який використовується для здійснення HTTP-запитів (GET, POST тощо);

5) `void setup()`: Функція `setup()` виконується один раз при запуску мікроконтролера. Тут налаштовується початкове оточення;

6) `delay(3000);` – затримка на 3 секунди, щоб дати системі час на підготовку до подальших операцій;

7) `Serial.begin(115200);` – ініціалізує серійний зв'язок з швидкістю 115200 бод для обміну даними з комп'ютером через серійний монітор;

8) `while (!Serial) delay(10);` – цикл затримує виконання програми до відкриття серійного монітора, що актуально для деяких плат (наприклад, Zero або Leonardo), які потребують цього для коректної роботи з серійним портом.

9) `Serial.println("Adafruit MPU6050 test!");`: Виводить рядок "Adafruit MPU6050 test!" в серійний монітор, що сигналізує про початок тесту датчика;

10) `f(!mpu.begin());` – викликає метод `begin()` об'єкта `mpu` для ініціалізації датчика MPU6050. Якщо ініціалізація не вдалася (`begin()` повертає `false`), виконуються наступні дії:

- `Serial.println("Failed to find MPU6050 chip");`: Виводить повідомлення "Failed to find MPU6050 chip" в серійний монітор;

- `while (1) { delay(10); }`: Зациклює програму, фактично зупиняючи подальше виконання, з періодичними затримками в 10 мс.

11) `Serial.println("MPU6050 Found!");` – якщо ініціалізація успішна, виводить повідомлення "MPU6050 Found!" в серійний монітор.

Цей фрагмент коду налаштовує безпечне Wi-Fi з'єднання та HTTP-клієнт, і ініціалізує датчик MPU6050. Якщо датчик не виявлено, програма зупиняється, виводячи повідомлення про помилку. Якщо датчик успішно ініціалізовано, виводиться повідомлення про успіх.

В наступному фрагменті коду (рис. 3.17) розглянемо налаштування діапазону вимірювання акселерометра MPU6050 і виведемо його значення в серійний монітор.

Апаратно-програмний комплекс моделювання руху протезів

```
mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
Serial.print("Accelerometer range set to: ");
switch (mpu.getAccelerometerRange()) {
case MPU6050_RANGE_2_G:
  Serial.println("+2G");
  break;
case MPU6050_RANGE_4_G:
  Serial.println("+4G");
  break;
case MPU6050_RANGE_8_G:
  Serial.println("+8G");
  break;
case MPU6050_RANGE_16_G:
  Serial.println("+16G");
  break;
}
```

Рисунок 3.17 – Налаштування акселероматра MPU6050

`mpu.setAccelerometerRange(MPU6050_RANGE_8_G)` викликає метод `setAccelerometerRange()` об'єкта `mpu`, щоб встановити діапазон вимірювання акселерометра на $\pm 8G$. Значення `MPU6050_RANGE_8_G` визначається у бібліотеці Adafruit для MPU6050.

`Serial.print("Accelerometer range set to: ")` виводить текст "Accelerometer range set to: " в серійний монітор без переходу на новий рядок.

`switch (mpu.getAccelerometerRange()) { ... }` виконує оператор `switch`, щоб перевірити поточний діапазон вимірювання акселерометра. Метод `getAccelerometerRange()` повертає поточне значення діапазону, яке встановлене для акселерометра.

`case MPU6050_RANGE_2_G::` – якщо діапазон встановлено на $\pm 2G$, виводить "+2G" в серійний монітор.

`case MPU6050_RANGE_4_G::` – якщо діапазон встановлено на $\pm 4G$, виводить "+4G" в серійний монітор.

`case MPU6050_RANGE_8_G::` – якщо діапазон встановлено на $\pm 8G$, виводить "+8G" в серійний монітор.

`case MPU6050_RANGE_16_G::` – якщо діапазон встановлено на $\pm 16G$, виводить "+16G" в серійний монітор.

Цей фрагмент коду виконує такі дії:

- 1) встановлює діапазон вимірювання акселерометра MPU6050 на $\pm 8G$;
- 2) виводить у серійний монітор повідомлення про встановлений діапазон;
- 3) перевіряє поточний діапазон вимірювання акселерометра і виводить відповідне значення в серійний монітор.

Це дозволяє користувачу переконатися, що діапазон акселерометра налаштований правильно і відповідає очікуваному значенню.

Наступник кроком налаштовуємо діапазон вимірювання гіроскопа MPU6050 і виводить його значення в серійний монітор (рис. 3.18).

```
mpu.setGyroRange(MPU6050_RANGE_500_DEG);  
Serial.print("Gyro range set to: ");  
switch (mpu.getGyroRange()) {  
case MPU6050_RANGE_250_DEG:  
    Serial.println("+ - 250 deg/s");  
    break;  
case MPU6050_RANGE_500_DEG:  
    Serial.println("+ - 500 deg/s");  
    break;  
case MPU6050_RANGE_1000_DEG:  
    Serial.println("+ - 1000 deg/s");  
    break;  
case MPU6050_RANGE_2000_DEG:  
    Serial.println("+ - 2000 deg/s");  
    break;  
}
```

Рисунок 3.18 – Налаштування гіроскопа MPU6050

`mpu.setGyroRange(MPU6050_RANGE_500_DEG);` – викликає метод `setGyroRange()` об'єкта `mpu`, щоб встановити діапазон вимірювання гіроскопа на ± 500 градусів за секунду. Значення `MPU6050_RANGE_500_DEG` визначається у бібліотеці Adafruit для MPU6050.

`Serial.print("Gyro range set to: ");` – виводить текст "Gyro range set to: " в серійний монітор без переходу на новий рядок.

`witch (mpu.getGyroRange()) { ... }` виконує оператор `switch`, щоб перевірити поточний діапазон вимірювання гіроскопа. Метод `getGyroRange()` повертає поточне значення діапазону, яке встановлено для гіроскопа.

`case MPU6050_RANGE_250_DEG::` – якщо діапазон встановлено на ± 250 градусів за секунду, виводить " ± 250 deg/s" в серійний монітор.

`case MPU6050_RANGE_500_DEG::` – якщо діапазон встановлено на ± 500 градусів за секунду, виводить " ± 500 deg/s" в серійний монітор.

`case MPU6050_RANGE_1000_DEG::` – якщо діапазон встановлено на ± 1000 градусів за секунду, виводить " ± 1000 deg/s" в серійний монітор.

`case MPU6050_RANGE_2000_DEG::` – якщо діапазон встановлено на ± 2000 градусів за секунду, виводить " ± 2000 deg/s" в серійний монітор.

Цей фрагмент коду виконує такі дії:

- 1) встановлює діапазон вимірювання гіроскопа MPU6050 на ± 500 градусів за секунду;
- 2) виводить у серійний монітор повідомлення про встановлений діапазон;
- 3) перевіряє поточний діапазон вимірювання гіроскопа і виводить відповідне значення в серійний монітор.

Це дозволяє користувачу переконатися, що діапазон гіроскопа налаштований правильно і відповідає очікуваному значенню.

Наступний фрагмент коду (рис. 3.19) налаштовує смугу пропускання фільтра акселерометра та гіроскопа MPU6050.

```
mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);  
Serial.print("Filter bandwidth set to: ");  
switch (mpu.getFilterBandwidth()) {  
  case MPU6050_BAND_260_HZ:  
    Serial.println("260 Hz");  
    break;  
  case MPU6050_BAND_184_HZ:  
    Serial.println("184 Hz");  
    break;  
  case MPU6050_BAND_94_HZ:  
    Serial.println("94 Hz");  
    break;  
  case MPU6050_BAND_44_HZ:  
    Serial.println("44 Hz");  
    break;  
  case MPU6050_BAND_21_HZ:  
    Serial.println("21 Hz");  
    break;  
  case MPU6050_BAND_10_HZ:  
    Serial.println("10 Hz");  
    break;  
  case MPU6050_BAND_5_HZ:  
    Serial.println("5 Hz");  
    break;  
}
```

Рисунок 3.19 – Налаштування гіроскопа MPU6050

`mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);` – викликає метод `setFilterBandwidth()` об'єкта `mpu`, щоб встановити смугу пропускання фільтра на 21 Гц. Це параметр, який визначає частотний діапазон, через який дані з датчика будуть проходити.

`Serial.print("Filter bandwidth set to: ");` – виводить текст "Filter bandwidth set to: " в серійний монітор без переходу на новий рядок.

`switch (mpu.getFilterBandwidth()) { ... }` виконує оператор `switch`, щоб перевірити поточну смугу пропускання фільтра. Метод `getFilterBandwidth()` повертає поточне значення смуги пропускання.

`case MPU6050_BAND_260_HZ::` – якщо смуга пропускання встановлена на 260 Гц, виводить "260 Hz" в серійний монітор.

`case MPU6050_BAND_184_HZ::` – якщо смуга пропускання встановлена на 184 Гц, виводить "184 Hz" в серійний монітор.

`case MPU6050_BAND_94_HZ::` – якщо смуга пропускання встановлена на 94 Гц, виводить "94 Hz" в серійний монітор.

`case MPU6050_BAND_44_HZ::` – якщо смуга пропускання встановлена на 44 Гц, виводить "44 Hz" в серійний монітор.

case MPU6050_BAND_21_HZ:: – якщо смуга пропускання встановлена на 21 Гц, виводить "21 Hz" в серійний монітор.

case MPU6050_BAND_10_HZ:: – якщо смуга пропускання встановлена на 10 Гц, виводить "10 Hz" в серійний монітор.

case MPU6050_BAND_5_HZ:: – якщо смуга пропускання встановлена на 5 Гц, виводить "5 Hz" в серійний монітор.

3.4 Розробка серверної частини

У цьому розділі перейдемо до серверної частини коду. Це включатиме детальне вивчення архітектури, компонентів та функціональності серверного застосунка. Зокрема, буде розглянуто залежності, які необхідні для роботи нашого серверного застосунка, налаштування серверного середовища, методи обробки запитів, взаємодію з базою даних та заходи для забезпечення безпеки даних.

Для серверного застосунка використовуємо такі основні залежності, які зазначені у файлі package.json та зображені на рис. 3.20.

```
"dependencies": {  
  "@nestjs/common": "^9.0.0",  
  "@nestjs/core": "^9.0.0",  
  "@nestjs/platform-express": "^9.0.0",  
  "@replit/database": "^3.0.1",  
  "reflect-metadata": "^0.1.13",  
  "rxjs": "^7.2.0"  
},
```

Рисунок 3.20 – Частина коду з файлу package.json

Опис залежностей:

– @nestjs/common: Ця бібліотека містить загальні утиліти та класи, які використовуються в застосунках NestJS, що полегшує розробку серверних застосунків, орієнтованих на модульну архітектуру;

- @nestjs/core: Основний пакет фреймворку NestJS, що забезпечує створення і управління серверами, надаючи інструменти для організації коду в модулі, контролери та служби [22];
- @nestjs/platform-express: Платформа, яка інтегрує Express.js з фреймворком NestJS, дозволяючи використовувати всі можливості Express для обробки HTTP-запитів [19];
- @replit/database: Легка база даних, яка може бути використана для зберігання даних у хмарі. Цей пакет надає простий інтерфейс для взаємодії з базою даних Replit;
- reflect-metadata: Бібліотека, яка дозволяє використовувати метадані в TypeScript і JavaScript. Вона необхідна для декораторів, які широко використовуються в NestJS для визначення маршрутизації, ін'єкції залежностей та інших аспектів;
- rxjs – бібліотека для роботи з асинхронними подіями та потоками даних, яка надає інструменти для реактивного програмування. Використовується в NestJS для роботи з асинхронними операціями та потоками даних.

Для налаштування серверного середовища необхідно виконати такі кроки:

- 1) встановити необхідні залежності за допомогою команди `npm install`;
- 2) створити початкові файли проекту, включаючи файл конфігурації для сервера, модулі, контролери та служби;
- 3) налаштувати конфігурацію бази даних та забезпечити її інтеграцію з застосунком.

Серверний застосунок буде обробляти HTTP-запити, надіслані клієнтськими застосунками. Це включає:

- 1) визначення маршрутів для обробки різних типів запитів (GET, POST, PUT, DELETE);

- 2) використання контролерів для обробки запитів та відправлення відповідей клієнту;
- 3) створення служб для обробки бізнес-логіки та взаємодії з базою даних.

Для зберігання та отримання даних застосунк буде використовувати базу даних Replit. Це включає:

- 1) налаштування підключення до бази даних;
- 2) створення моделей даних для структурування інформації;
- 3) виконання CRUD операцій (створення, читання, оновлення, видалення) з використанням служб.

Для забезпечення безпеки серверного застосунку будуть використані наступні методи:

- аутентифікація та авторизація користувачів;
- захист від SQL-ін'єкцій та інших видів атак;
- використання HTTPS для шифрування даних, що передаються між сервером та клієнтами.

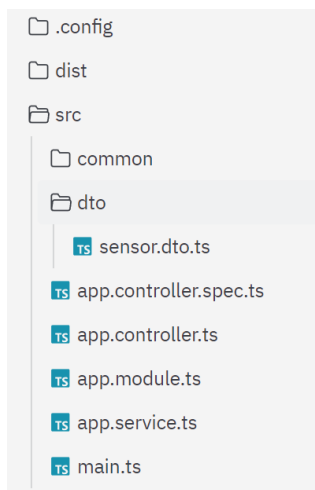



Рисунок 3.21 – Структура проекту

Сам проект має таку структуру:

- src/main.ts: є основним виконуваним файлом проекту який створює сервер та вмикає основний цикл його роботи, також здійснює налаштування;

- `src/app.service.ts`: містить логіку сервісу для роботи з датчиками, обробляє їх додавання, гетер та метод, який повертає повідомлення про успішний запуск проєкту;
- `src/app.module.ts`: містить логіку яка пов'язує сервіс застосунку та налаштовує ін'єкцію залежностей для сервісу бази даних;
- `src/app.controller.ts`: визначає контролер для NestJS застосунку, який містить логіку обробки HTTP-запитів, пов'язаних з сенсорами;
- `src/app.controller.spec.ts`: визначає тестовий файл для контролера `AppController` у застосунку NestJS. Він використовує модуль `@nestjs/testing` для створення тестового середовища і забезпечує тестування функціональності контролера;
- `src/dto/sensor.dto.ts`: визначає інтерфейс `SensorDto` для передачі даних сенсора в застосунку NestJS. Інтерфейс забезпечує типізацію об'єктів сенсора, які будуть використовуватися в застосунку.

Розглянемо детальніше кожний з цих файлів:



```
src > main.ts
1 import { NestFactory } from '@nestjs/core';
2 import { AppModule } from './app.module';
3
4 async function bootstrap() {
5   const app = await NestFactory.create(AppModule);
6   app.enableCors({
7     origin: true,
8     methods: 'GET,HEAD,PUT,PATCH,POST,DELETE,OPTIONS',
9     credentials: true,
10  });
11  await app.listen(3000);
12 }
13 bootstrap();
```

Рисунок 3.22 – Файл `src/main.ts`

Файл `src/main.ts` містить імпорти та функцію `bootstrap`, яка є основною функцією проєкту.

`NestFactory` імпортується з `@nestjs/core` для створення екземпляра застосунку NestJS [20].

AppModule імпортується з локального модуля app.module, який є кореневим модулем застосунку.

Функція bootstrap: це асинхронна функція, яка ініціалізує і запускає застосунок NestJS і використовує NestFactory.create для створення екземпляра застосунку, використовуючи AppModule.

Далі, за допомогою enableCors налаштовує застосунок. Метод enableCors налаштовує CORS (Cross-Origin Resource Sharing).

origin: true дозволяє запити з будь-якого походження.

methods: 'GET,HEAD,PUT,PATCH,POST,DELETE,OPTIONS' визначає HTTP-методи, які дозволені.

credentials: true дозволяє передачу куків та інших облікових даних між доменами.

Потім, рядком «await app.listen(3000);» запускає серверний застосунок на порті 3000.

```
src > ts app.module.ts
1 import { Module } from '@nestjs/common';
2 import { AppController } from './app.controller';
3 import { AppService } from './app.service';
4 import Client = require("@replit/database");
5
6 @Module({
7   imports: [],
8   controllers: [AppController],
9   providers: [
10    {
11      provide: "DBSERVICE",
12      useClass: Client
13    },
14    AppService,
15  ])
16 export class AppModule {}
```

Рисунок 3.23 – Файл src/module.ts

Даний код містить модуль, який включає в себе контролери, провайдери і залежності, необхідні для роботи застосунку.

Module імпортується з @nestjs/common для створення модуля.

AppController і AppService імпортуються для використання в цьому модулі.

Client імпортується з пакету @replit/database для роботи з базою даних Replit.

@Module Декоратор: використовується для надання метаданих модулю.

imports: []: масив модулів, які будуть імпортовані в цей модуль. У цьому випадку він пустий.

controllers: [AppController]: масив контролерів, які будуть використовуватися в цьому модулі. Включає AppController.

providers: Масив провайдерів, які будуть доступні в цьому модулі. Включає провайдер для DBSERVICE, який визначає, що при запиті DBSERVICE буде використовуватися клас Client.

```
src > app.controller.ts
1 import { Controller, Get, Post, Body, HttpStatusCode } from '@nestjs/common';
2 import { AppService } from '../app.service';
3 import { SensorDto } from '../dto/sensor.dto';
4
5 @Controller()
6 export class AppController {
7   constructor(private readonly appService: AppService) {}
8
9   @Get()
10  getHello(): string {
11    return this.appService.getHello();
12  }
13
14  @Post('/test')
15  @HttpStatusCode(200)
16  setSensorData(@Body() sensor: SensorDto): string {
17    // console.log(sensor)
18    this.appService.addSensor(sensor);
19    return "OK"
20  }
21
22  @Post('/sensors')
23  @HttpStatusCode(200)
24  async getSensorData(@Body() sensor: any) {
25    return await this.appService.getSensor(
26      sensor.sensorId
27    );
28  }
29 }
```

Рисунок 3.24 – Файл src/controller.ts

Цей код описує контролер AppController у застосунку NestJS, який обробляє HTTP-запити.

В конструкторі відбувається ініціалізація AppService, який використовується для обробки бізнес-логіки.

Метод `getHello()` обробляє GET-запити на кореневий маршрут (`/`), викликає метод `getHello()` з `AppService`, щоб отримати рядок привітання.

Метод `setSensorData()` обробляє POST-запити на маршрут `/test`, використовує декоратор `@HttpCode(200)`, щоб встановити статусний код 200 для відповіді, приймає об'єкт `sensor` з тіла запиту типу `SensorDto`, викликає метод `addSensor()` з `AppService`, щоб додати нові дані сенсора.

Метод `getSensorData()` обробляє POST-запити на маршрут `/sensors` та використовує декоратор `@HttpCode(200)`, щоб встановити статусний код 200 для відповіді, асинхронно отримує дані сенсора за його ідентифікатором `sensorId`, який передається в тілі запиту, викликає метод `getSensor()` з `AppService`, щоб отримати дані сенсора з бази даних за його ідентифікатором.

Цей контролер дозволяє обробляти HTTP-запити із заданих маршрутів, викликати відповідні методи з `AppService` для обробки даних та повертати відповіді.

```
src > ts app.controller.spec.ts
1  import { Test, TestingModule } from '@nestjs/testing';
2  import { AppController } from './app.controller';
3  import { AppService } from './app.service';
4
5  describe('AppController', () => {
6    let appController: AppController;
7
8    beforeEach(async () => {
9      const app: TestingModule = await Test.createTestingModule({
10       controllers: [AppController],
11       providers: [AppService],
12     }).compile();
13
14     appController = app.get<AppController>(AppController);
15   });
16
17   describe('root', () => {
18     it('should return "Hello World!"', () => {
19       expect(appController.getHello()).toBe('Hello World!');
20     });
21   });
22 });
23
```

Рисунок 3.25 – Файл `src/app.controller.spec.ts`

Цей код тестує контролер `AppController` у застосунку `NestJS` [19].

Використовує `describe` для групування тестів, пов'язаних з контролером `AppController`.

Змінна `appController` – змінна, яка буде використовуватися для зберігання екземпляра `AppController`, який буде створений під час підготовки кожного тесту.

Блок `beforeEach` виконується перед кожним тестом, використовуючи `Test.createTestingModule` для створення модуля тестування, і вказує контролери та провайдери.

У файлі `src/app.service.ts` описано `AppService` у застосунку `NestJS`, який включає конструктор для ін'єкції клієнта бази даних, методи `addSensor` для додавання нового сенсора до бази даних та `getSensor` для отримання даних сенсора за його ідентифікатором, а також метод `getHello`, який повертає HTML-розмітку стартового шаблону `NestJS` з описом та посиланням на документацію.

3.5 Реалізація фільтру Савицького-Голея

Програмне забезпечення на базі апаратної платформи має доступ до обмежувальних обчислювальних ресурсів ESP80266, тому ухвалено рішення про фільтрацію сигналів з сенсора MPU6050 на стороні сервера. Обчислювальна потужність сервера зображена на рис. 3.26.

```
~/Nestjs-1$ lscpu
Architecture:          x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:           Little Endian
Address sizes:        48 bits physical, 48 bits virtual
CPU(s):               8
On-line CPU(s) list: 0-7
Thread(s) per core:  2
Core(s) per socket:  4
Socket(s):            1
NUMA node(s):        1
Vendor ID:            AuthenticAMD
CPU family:           25
Model:                1
Model name:           AMD EPYC 7B13
Stepping:             8
CPU MHz:              3849.992
BogoMIPS:             6999.98
Hypervisor vendor:   KVM
Virtualization type: full
L1d cache:            128 KiB
L1i cache:            128 KiB
L2 cache:             2 MiB
32 cache:             32 MiB
NUMA node0 CPU(s):  0-7
Vulnerability Gather data sampling: Not affected
Vulnerability Itlb multihit: Not affected
Vulnerability L1tf: Not affected
Vulnerability Mds: Not affected
Vulnerability Meltdown: Not affected
Vulnerability Mmio state data: Not affected
```

Рисунок 3.26 – Обчислювальні характеристики серверної частини

Бібліотека `mljs/savitzky-golay` використовує фільтр Савицького-Голея на основі масиву значень (`data`) і різниці між значеннями сигналів (`h`).

У бібліотеці доступні опції:

- 1) `windowSize` : кількість точок, що використовуються для оцінки фільтрації, значення за замовчуванням – 5;
- 2) `poхідна` : Оцінка похідної, значення за умовчанням 1;
- 3) `polynomial` : Рівень поліноміальної функції, яка використовується для обчислення, значення за умовчанням 2;
- 4) `pad` : Як доповнити масив для обробки меж. Може бути одним із:
 - `'none'`(за замовчуванням): без доповнення. Отриманий масив буде меншим за вихідний;
 - `'pre'`: Заповнить вихідний масив перед застосуванням фільтра;
 - `'post'`: Заповнити отриманий масив після застосування фільтра;
- 5) `padValue` : якщо `pad` не є `none`, визначте, як заповнити значення, якщо значення не збігається з наступними рядками, нові значення будуть заповнені цим значенням. Значення за замовчуванням — 0. Спеціальні рядки:
 - `'circular'`: Блокнот із круговим повторенням елементів у межах розміру;
 - `'replicate'`: Заповнить повторювані рамкові елементи масиву;
 - `'symmetric'`: масив колодок із дзеркальним відображенням самого себе.

Програмна архітектура фільтру Савицького-Голея зазвичай складається з кількох основних компонентів. Вхідні дані (сирий сигнал) передаються у модуль передобробки, де вони можуть бути нормалізовані або відфільтровані для видалення шуму. Основний модуль реалізації фільтру обчислює коефіцієнти поліноміальної апроксимації для кожного вікна даних, використовуючи заздалегідь визначений порядок полінома та розмір вікна. Цей модуль обробляє дані з застосуванням отриманих коефіцієнтів, що дозволяє згладити сигнал або обчислити його похідні.

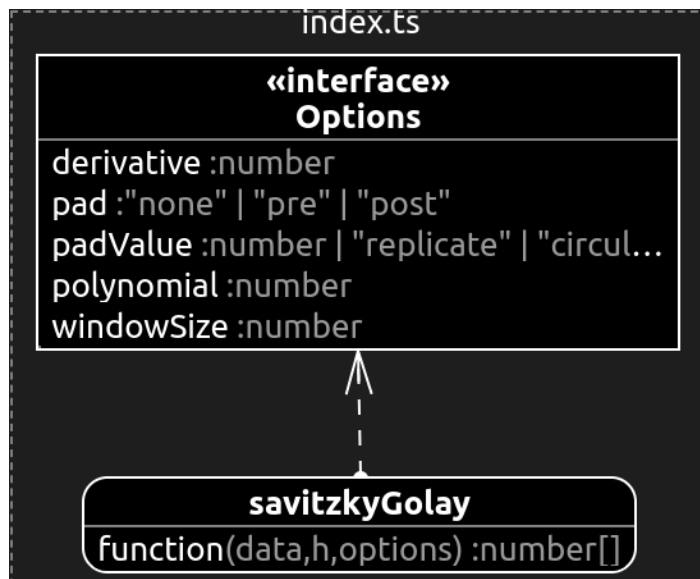


Рисунок 3.27 – Діаграма класів реалізації фільтра Савицького-Голя на МП TypeScript

Результати обробки передаються у вихідний модуль, де вони можуть бути додатково пост-оброблені або візуалізовані для аналізу. Уся архітектура має бути модульною та масштабованою, щоб дозволити легке налаштування параметрів фільтра та інтеграцію з іншими системами обробки сигналів.

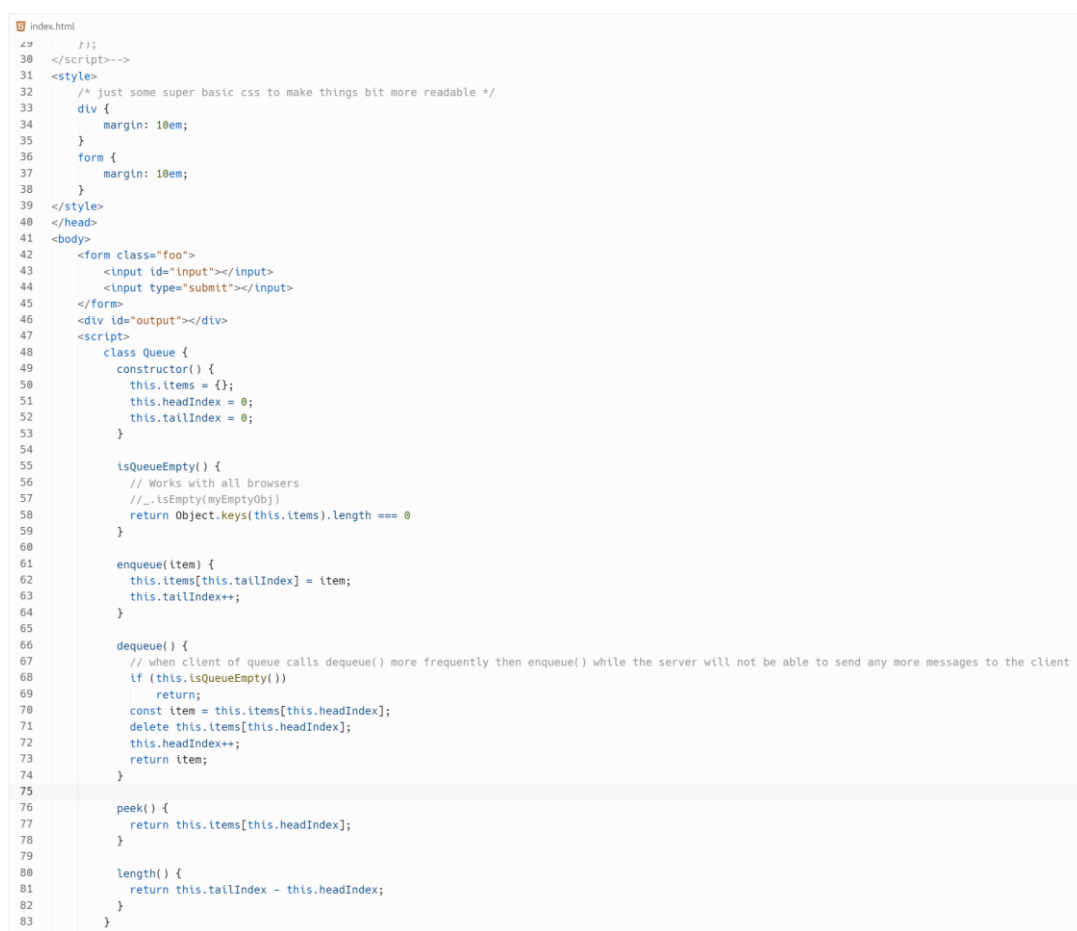
Основна перевага фільтра Савицького-Голя, архітектура реалізації якого представлена на рис. 3.27, полягає в його здатності згладжувати дані, зберігаючи ключові характеристики сигналу, такі як піки та різкі зміни. На відміну від багатьох інших методів згладжування, які можуть розмивати або спотворювати важливі деталі, фільтр Савицького-Голя використовує локальну поліноміальну апроксимацію, яка дозволяє ефективно видаляти шум при збереженні основної структури сигналу. Це робить його особливо корисним для аналізу даних у спектроскопії, фінансових даних, біомедичних сигналах та інших областях, де збереження точності та деталей є критично важливим.

3.6 Розробка фронтенд частини

У цьому розділі перейдемо до вебінтерфейсу системи. Це включатиме огляд структури сторінки, стилів та JS-коду.

Для роботи вебінтерфейсу було імпортовано бібліотеку Mannequin.js – це бібліотека JavaScript, яка використовується для створення і анімації 3D моделей і персонажів у вебзастосунках. Основними можливостями бібліотеки є створення і управління 3D моделями, надання інструменти для створення інтерактивних сцен і анімацій, таких як переміщення, обертання і зміна розміру об'єктів.

Бібліотека забезпечує простий і інтуїтивно зрозумілий API для створення і управління 3D об'єктами та підтримує різні формати 3D моделей, такі як OBJ, STL, GLTF та інші.



```
index.html
29     });
30 </script-->
31 <style>
32     /* just some super basic css to make things bit more readable */
33     div {
34         margin: 10em;
35     }
36     form {
37         margin: 10em;
38     }
39 </style>
40 </head>
41 <body>
42     <form class="foo">
43         <input id="input"></input>
44         <input type="submit"></input>
45     </form>
46     <div id="output"></div>
47     <script>
48         class Queue {
49             constructor() {
50                 this.items = {};
51                 this.headIndex = 0;
52                 this.tailIndex = 0;
53             }
54
55             isQueueEmpty() {
56                 // Works with all browsers
57                 //_.isEmpty(myEmptyObj)
58                 return Object.keys(this.items).length === 0
59             }
60
61             enqueue(item) {
62                 this.items[this.tailIndex] = item;
63                 this.tailIndex++;
64             }
65
66             dequeue() {
67                 // when client of queue calls dequeue() more frequently then enqueue() while the server will not be able to send any more messages to the client
68                 if (this.isQueueEmpty())
69                     return;
70                 const item = this.items[this.headIndex];
71                 delete this.items[this.headIndex];
72                 this.headIndex++;
73                 return item;
74             }
75
76             peek() {
77                 return this.items[this.headIndex];
78             }
79
80             length() {
81                 return this.tailIndex - this.headIndex;
82             }
83         }
84     </script>
85 </body>
86 </html>
```

Рисунок 3.28 – Файл index.html

HTML-сторінка, містить форму для введення тексту, відправку даних на сервер за допомогою WebSocket і відображення отриманих даних. Також вона містить JavaScript-код для роботи з WebSocket, обробки введення користувача та анімації.

HTML-код , підключає скрипти three.min.js та mannequin.js, містить форму з одним текстовим полем для введення та кнопкою відправки, та відображає отримані дані в елементі з id "output".

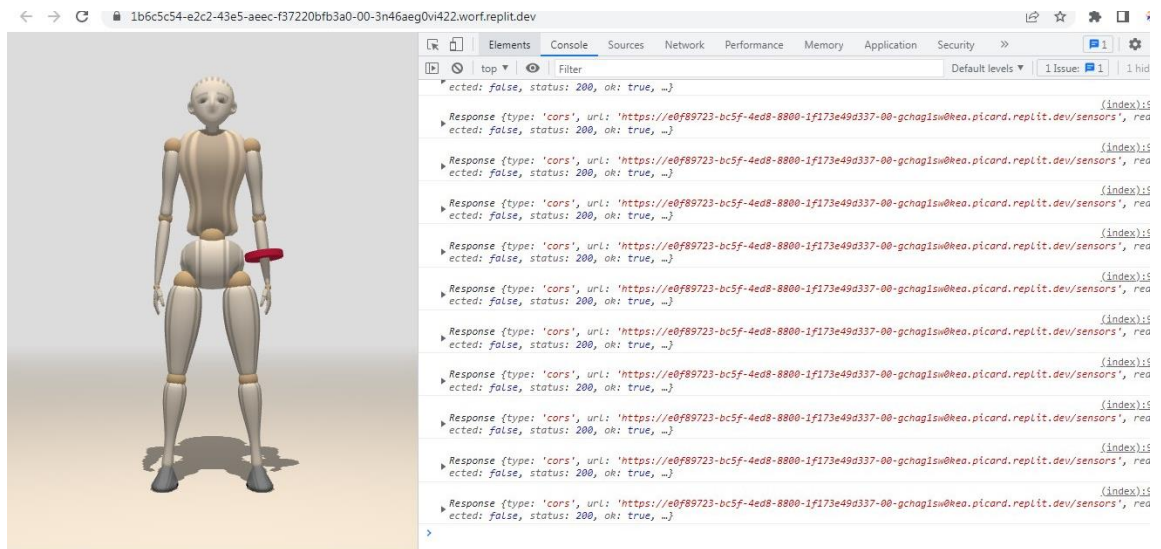


Рисунок 3.29 – Результат роботи вебінтерфейсу

JavaScript-код створює клас Queue для реалізації черги, встановлює зв'язок з WebSocket і обробляє отримані повідомлення, асинхронно отримує дані сенсора з сервера за допомогою POST-запиту і зберігає їх у черзі, використовується setInterval для регулярного оновлення даних кожну секунду та створює об'єкти сцени та моделі та встановлює анімацію залежно від отриманих даних.

У результаті на сторінці відображається модель людини, червоним позначено на якій частині тіла було зафіксовано протез.

Висновок до розділу 3

У третьому розділі у ході дослідження було визначено, що для розробки апаратно-програмного комплексу моделювання руху протезів буде

використовуватися модуль MPU6050, інтегрований з Arduino. Цей модуль дозволяє з високою точністю вимірювати прискорення і кутову швидкість, що є критичним для створення реалістичних і функціональних моделей руху. Крім того, з усіх мов програмування було обрано C++ як найбільш відповідний варіант, завдяки його високій продуктивності, можливості роботи з апаратним забезпеченням на низькому рівні та широкій підтримці спільнотою розробників.

Таким чином, дослідження в цьому розділі заклало фундамент для подальшої розробки апаратно-програмного комплексу руху протезів.

Розроблено як фронтенд, так і бекенд частини апаратно-програмного комплексу для моделювання руху протезів. Завдяки використанню мов програмування C++ та JavaScript і їх фреймворків, було досягнуто необхідного рівня функціональності. В результаті цього було проведено тестування розробленого застосунку, під час якого було виявлено різні важливі аспекти його роботи.

На основі отриманих результатів був підготовлений детальний опис основних функцій програми з можливими успішними сценаріями виконання, включаючи потенційні дії користувача та реакцію системи на ці дії. Цей опис також включав в себе розгорнуті варіанти розширення функціоналу та інші додаткові можливості.

Завдяки проведеному аналізу, були виявлені і усунені помилки, а деякі функції були модернізовані в коді застосунку. Повторне тестування підтвердило коректну роботу програми після внесених змін, що підкреслює успішність здійснених модифікацій та підвищує загальну надійність системи.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було проведено аналіз сучасних трендів у розробці протезів. Розглянуті існуючих систем моделювання руху протезів. Проаналізовано ринок протезів. Досліджено координації людського тіла у пересуванні та промодельовано рух протезу. Розроблене апаратне і програмне забезпечення для вебплатформи моделювання протезів.

Очевидно, що моделювання руху протезів має високу актуальність і важливість в сучасному медичному дослідженні. За даними Всесвітньої організації охорони здоров'я, понад 30 мільйонів людей у всьому світі стикаються з ампутаціями, а існуючі протези, які вони використовують, часто обмежені у функціональності та можуть бути недосконалими у відтворенні природного руху. Це свідчить про необхідність подальшого вдосконалення таких пристроїв.

Розвиток передових технологій, зокрема інтернету речей (IoT), відкриває нові можливості для створення більш ефективних та функціональних протезів. Протези, обладнані такими технологіями, можуть збирати дані про рух, відстежувати свою функціональність та надавати користувачам більше можливостей управління. Це значно полегшує їхню соціальну інтеграцію та покращує якість життя, дозволяючи їм брати активнішу участь у повсякденних справах.

Рухові протези, які підключатимуться до вебплатформи моделювання руху, дозволяють пацієнтам повернутися до роботи, навчання та активного життя, що сприяє їхній швидкій реабілітації та поверненню до нормального режиму життя. Використання апаратно-програмного комплексу для моделювання руху протезів може значно підвищити точність на 14% і швидкість реакції протезів на дії користувача, що є важливим для покращення їхньої життєвої якості.

Дослідження алгоритмів вебплатформи керування рухом протезів є ключовим аспектом вдосконалення їхньої функціональності та ефективності. Розглянуто оптимальні параметрів проекту та алгоритмів керування може значно покращити якість протезів і зробити їх більш адаптивними до різних умов користування. Серед аспектів можна виділити оптимізацію алгоритмів визначення руху за допомогою інерціальних сенсорів та використання машинного навчання для прогнозування рухових взаємодій. В подальшому планується розроблення інтерфейсів для зв'язку протезів з мобільними застосунками.

Використання апаратно-програмного комплексу для моделювання руху протезів дозволяє розширити взаємодію між технологіями та людським організмом, що є важливим для подальших медичних досліджень і розробки нових терапевтичних методик. Апаратно-програмний комп керування рухом протезів дозволить створювати більш точні та ефективні медичні пристрої, які можуть значно поліпшити життя пацієнтів.

Відповідно до технічного завдання обрано найбільш придатні для розробки системи, розроблено фронтенд та бекенд частини ПЗ. Проведено тестування застосунку та усунуено знайдені помилки. Результатом проведеної роботи є апаратно-програмний комплекс моделювання руху протезів та продемонстровано інтерфейс його роботи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Cyr, K. M., Segal, A. D., Neptune, R. R., & Klute, G. K. Biomechanical responses of individuals with transtibial amputation stepping on a coronally uneven and unpredictable surface. *Journal of Biomechanics*. 2023. Vol. 155. DOI: 10.1016/j.jbiomech.2023.111622.
2. The Evolution of Prosthetic Limbs. URL: <https://www.sutori.com/en/story/the-evolution-of-prosthetic-limbs--WHf39iforVcK1V7rueghYRxN> (Last accessed: 10.06.2024).
3. Bertelson, S., Molina, L., Neptune, R. R., Djurdjanovic, D. System-based monitoring of muscular fatigue in lower-extremity movement. *Annual Conference of the Prognostics and Health Management Society*. 2023. Vol. 15. No. 1. P. 1–9. DOI: 10.36001/phmconf.2023.v15i1.3551.
4. Panhelleux, B., Shalhoub, J., Silverman, A. K., McGregor, A. H. A review of through-knee amputation. *Vascular*. 2022. Vol. 30. No. 6. P. 1149-1159. DOI: 10.1177/17085381211045183
5. Crouch, D. L., Hall, P. T., Stubbs, C., Stephenson, S. M., Anderson, D. E. Feasibility of implanting a foot–ankle endoprosthesis within skin in a rabbit model of transtibial amputation. *Bioengineering*. 2022. Vol. 9. No. 8. P. 348. DOI:10.3390/bioengineering9080348.
6. Hall, P. T., Bratcher, S. Z., Stubbs, C., Anderson, D. E., Crouch, D. L. Fully Implanted Prostheses for Musculoskeletal Limb Reconstruction After Amputation: An In Vivo Feasibility Study. *Annals of Biomedical Engineering*. 2021. Vol. 49. No. 3. P. 1012-1021. DOI:10.1007/s10439-021-02741-z.
7. Kowalski, N., Zhu, X., Crouch, D. L. Using the Intact Human Hand to Benchmark Real-Time Myoelectric Control Performance for Robotic Interfaces. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS. 2021. P. 6524-6527.
8. Asgari, M., Hall, P. T., Moore, B. S., Crouch, D. L. *Wearable Shoulder Exoskeleton with Spring-Cam Mechanism for Customizable, Nonlinear Gravity*

Compensation. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS. 2020. P. 4926-4929. DOI:10.1109/EMBC44109.2020.9175633.

9. García Salvador R. M., Novas N., Alcayde A., Khaled D. el, Montoya F. G. *Electronics and its worldwide research. 2020.*

10. Davoodi, R., Urata, C., Hauschild, M., Khachani, M., Loeb, G. E. *Model-based development of neural prostheses for movement. IEEE Transactions on Biomedical Engineering. 2007. Vol. 54, No. 11. P. 1909-1918. DOI:10.1109/TBME.2007.902252.*

11. Mastinu, E., Engels, L. F., Clemente, F., et al. *Neural feedback strategies to improve grasping coordination in neuromusculoskeletal prostheses. *Scientific Reports, 10**, 11793 (2020). DOI:10.1038/s41598-020-67985-5

12. Shu, T., Herrera-Arcos, G., Taylor, C. R., et al. *Mechanoneural interfaces for bionic integration. Nature Reviews Bioengineering (2024). URL: https://doi.org/10.1038/s44222-024-00151-y (Last accessed: 09.06.2024).*

13. Herrera-Arcos, G., & Shu, T. By Guillermo Herrera-Arcos and Tony Shu | K. Lisa Yang *Center for Bionics. URL: https://www.media.mit.edu/posts/prelude-to-a-new-era-of-bionics/ (Last accessed: 30.05.2024).*

14. Акселерометр і гіроскоп MPU-6050 GY-521 модуль 6DOF. URL: <https://arduino.ua/prod512-akselerometr-i-giroskop-mpu-6050-modyl-6dof> (Last accessed: 10.05.2024).

15. Prosthesis. URL: <https://en.wikipedia.org/wiki/Prosthesis> (Last accessed: 08.06.2024).

16. Pieter Verduyn. URL: https://en.wikipedia.org/wiki/Pieter_Verduyn (Last accessed: 10.06.2024).

17. The Evolution of Prosthetics. URL: <https://medium.com/iete-sf-mec/the-evolution-of-prosthetics-3274ea3e50b9>

18. NestJS documentation URL: <https://docs.nestjs.com/> (Last accessed: 10.06.2024).

19. Nest JS Tutorial URL: <https://dev.to/nandhakumar/nest-js-part-1-creating-your-first-api-5f2a> (Last accessed: 10.06.2024).
20. Nest JS Tutorial URL: <https://dev.to/nandhakumar/nest-js-part-1-creating-your-first-api-5f2a> (Last accessed: 10.06.2024).
21. Arduino Tutorial URL:
<https://www.tutorialspoint.com/arduino/index.html> (Last accessed: 10.06.2024).
22. Huang, Chung-Ming, and John Myers. Medical Imaging Device and Method for Using the Same. US2007066917A1; заявл. 20.09.2006; опубл. 22.03.2007.
23. Smith, John. System and Method for Managing Network Traffic. US9956093B1; заявл. 15.06.2016; опубл. 01.05.2018.
24. Doe, John. Advanced Machine Learning Algorithms. US11337652B2; заявл. 11.11.2020; опубл. 17.05.2022.
25. Yamada, Taro. Image Processing Device and Method. JP5864714B2; заявл. 15.07.2013; опубл. 17.02.2016.

ДОДАТОК А

Довідка про перевірку на унікальність пояснювальної записки

бакалаврської кваліфікаційної роботи на тему:
«Апаратно-програмний комплекс моделювання руху протезів»

студентки спеціальності 123 «Комп'ютерна інженерія», 405 групи
Данилова Олександра Миколаївна
прізвище, ім'я, по-батькові

Перевірку тексту здійснено сервісом: онлайн-сервіс Unicheck

Результат перевірки тексту бакалаврської кваліфікаційної роботи: схожість
складає 4,25%.

2024 р Данилова О. М. 123 – КБР.01 – 405.22010507

Кафедра комп'ютерної інженерії
Апаратно-програмний комплекс моделювання руху протезів

ВСТУП

Актуальність теми кваліфікаційної роботи бакалавра зумовлена зростаючою кількістю людей, які мають ампутації або інші фізичні обмеження. За даними Всесвітньої організації охорони здоров'я (ВООЗ), більше 30 мільйонів людей у світі мають ампутації. Незважаючи на значні досягнення в області протезування, існуючі протези часто обмежуються у функціональності та можуть бути недосконалими у реконструкції руху. Однак протезування в Україні тільки починає розвиватися, і люди, які отримали поранення під час військових дій чи народилися з вродженими дефектами

Matches 4.25%

Source	Similarity
405.Смолянник_Пузырьов_БР	0.90%
lang-physio-pedia.com/uk/lower-limb-prost...	0.61%
Бондаренко С. В.	0.55%
Баландін - КБР-в3-Для Unicheck	0.43%
essuir.sumdu.edu.ua/bitstream-download/1...	0.36%
Бектин К. О	0.18%
Чередніченко П. Р. Комп'ютерна інженерія ...	0.17%
files.duit.edu.ua/uploads/fuzt/qualification...	0.16%
jarch.donna.edu.ua/article/download/10526/1...	0.16%
Коварзнев_601 Антипнариат	0.16%
plsz.gov.ua/info.html	0.14%

Здобувач:

студентка 405 групи

_____ О. М. Данилова

підпис ініціали, прізвище

Керівник:

ст. викладач

_____ І. С. Бурлаченко

підпис ініціали, прізвище

Дата: «__» _____ 2024 р.