

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

РОЗРОБКА ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ
«РОБОТИЗОВАНА МЕХАНІЧНА РУКА»

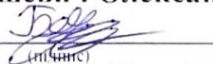
DEVELOPMENT OF SOFTWARE AND HARDWARE
COMPLEX «ROBOTIC MECHANICAL ARM»

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

освітня програма «Інженерія програмного забезпечення»
(назва освітньої програми)

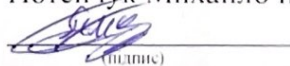
Виконав: здобувач вищої освіти
Групи ІПЗс-21

Білошевнич Олександр Андрійович


(підпис)

Керівник:
асистент

Потейчук Михайло Іванович


(підпис)

Кваліфікаційну роботу
допущено до захисту

« 8 » 06 2024 р.

к.т.н., доцент

Гарант освітньої програми:

Ліщина Наталія Миколаївна


(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

М.В. Мисирко *Н.В. Мисирко*

«2» 02 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Білошевичу Олександрю Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Розробка програмно-апаратного комплексу
«Роботизована механічна рука»»

Керівник роботи: асистент Потейчук Михайло Іванович

затверджені наказом закладу вищої освіти від «30» грудня 2023 р. № 477/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «5» червня
2024 р.

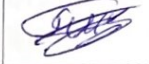
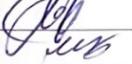
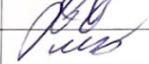
3. Вихідні дані до роботи: технічне та програмне забезпечення ЕОМ, вимоги до
розробки програмного забезпечення, ергономічні вимоги до функціонування
програмного засобу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно
розробити):

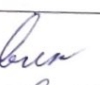
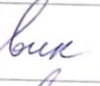
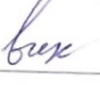
Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання,
аналіз, визначення вимог до розроблюваного програмного забезпечення
та проектування програмного забезпечення, опис функціонального наповнення
об'єкта проектування, практична реалізація об'єкта проектування.

5. Перелік графічного матеріалу: 17 рисунків

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Потейчук М. І.		
Специфікація вимог до розробленої системи	Потейчук М. І.		
Розробка об'єкта проектування	Потейчук М. І.		
Нормоконтроль	Повстяна Ю. С.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту	3,4 %		
Академічна доброчесність	Яцук А. А.		

7. Дата видачі завдання «2» лютого 2024 р.

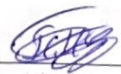
№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 07.02.2024 р.	
2	Аналіз проблеми розробки та впровадження об'єкту проектування	до 27.02.2024 р.	
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 12.03.2024 р.	
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	до 17.04.2024 р.	
5	Практична реалізація об'єкта проектування та розробка бази даних	до 18.05.2024 р.	
6	Тестування та налагодження об'єкта проектування	до 01.06.2024 р.	
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру	до 05.06.2024 р.	

Здобувач вищої освіти


(підпис)

Білошевич О. А.
(прізвище, ініціали)

Керівник кваліфікаційної роботи


(підпис)

Потейчук М. І.
(прізвище, ініціали)

Міністерство освіти і науки України
Луцький національний технічний університет

Рецензія
на кваліфікаційну роботу

Здобувач вищої освіти: Білошевич Олександр Андрійович

Тема: «Розробка програмно-апаратного комплексу «Роботизована механічна рука»»

Коротка характеристика кваліфікаційної роботи: Кваліфікаційна робота присвячена розробці програмно-апаратного комплексу «Роботизована механічна рука». Вона включає проектування механічної конструкції, розробку електронних схем, а також програмне забезпечення для керування рухами руки. Основна мета полягає в створенні високоточного та надійного пристрою, який може виконувати складні завдання в автоматизованих системах. У роботі також розглядаються питання оптимізації продуктивності та безпеки роботи комплексу.

Самостійні розробки і пропозиції автора: Автор самостійно розробив алгоритми керування роботизованою рукою, що забезпечують високу точність і плавність рухів. Також запропоновано унікальний дизайн механічних компонентів, який підвищує надійність та зменшує витрати на виробництво комплексу.

Практичне значення роботи: Практичне значення роботи полягає у створенні універсального роботизованого комплексу, здатного автоматизувати різноманітні виробничі процеси, підвищуючи ефективність та точність виконання завдань. Розроблений пристрій може використовуватись у промисловості, медицині та наукових дослідженнях, зменшуючи людську втому та ризик помилок. Крім того, запропоновані рішення сприяють зниженню витрат на обслуговування та інтеграцію роботизованих систем.

Недоліки: Недоліками роботи є обмежена функціональність програмного забезпечення в умовах складних виробничих середовищ та висока вартість виготовлення прототипу. Крім того, можлива необхідність додаткової наладки і адаптації системи під специфічні вимоги конкретних задач.

Загальний висновок: Розроблений програмно-апаратний комплекс «Роботизована механічна рука» демонструє високу ефективність та потенціал для широкого застосування в різних галузях. Незважаючи на деякі недоліки, робота пропонує перспективні рішення для автоматизації та оптимізації виробничих процесів.

Рецензент: Здолбіцька Ніна Василівна, к.т.н., доцент кафедри комп'ютерних наук
(прізвище, ім'я, по-батькові, посада)

Рецензент кваліфікаційної роботи _____


(підпис)

«07» червня 2024 р.

Міністерство освіти і науки України
Луцький національний технічний університет

**Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення**

**ВІДГУК
КЕРІВНИКА НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**
Здобувач вищої освіти Білошевич Олександр Андрійович
група ІІЗс-21

Тема кваліфікаційної роботи: «Розробка програмно-апаратного комплексу
«Роботизована механічна рука»»

Керівник Потейчук Михайло Іванович

Актуальність теми: Актуальність розробки програмно-апаратного комплексу "Роботизована механічна рука" зумовлена зростаючою потребою в автоматизації виробничих процесів, що сприяє підвищенню ефективності та якості продукції. Такі системи дозволяють виконувати складні і небезпечні для людини завдання, що підвищує безпеку та продуктивність праці.

Характеристика теоретичного рівня, наявності самостійних розробок і практичної значимості роботи: Робота демонструє теоретичний рівень проекту, який полягає у вивченні сучасних підходів до створення роботизованих механічних систем, включаючи алгоритми управління та моделювання руху. Наявність самостійних розробок проявляється в інноваційних рішеннях щодо програмного забезпечення та конструкції механічної руки, які значно підвищують її функціональність і адаптивність. Практична значимість роботи полягає у можливості впровадження комплексу в промислові процеси, що дозволить автоматизувати рутинні та небезпечні операції, підвищуючи загальну продуктивність. Окрім того, реалізація проекту може зменшити виробничі витрати і підвищити конкурентоспроможність підприємств.

Зауваження та недоліки: суттєвих недоліків в роботі не виявлено.

Загальний висновок робота виконана на високому рівні та заслуговує позитивної оцінки.

Керівник 
(підпис)

(Потейчук М. І.)
(прізвище, ім'я, по батькові)

«7» серпня 2024 р.

АНОТАЦІЯ

Білошевич О. А. Розробка програмно-апаратного комплексу «Роботизована механічна рука». Рукопис.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2024.

Метою роботи є теоретичне обґрунтування та розробка програмно-апаратного комплексу «Роботизована механічна рука». Під час розробки комплексу було вивчено функціональні вимоги та розроблено алгоритм роботи керування «Роботизованою механічною рукою». За результатами кваліфікаційної роботи було розроблено програмно-апаратний комплекс «Роботизована механічна рука» та проведено тестування даного пристрою.

Ключові слова: роботизована механічна рука, мікроконтролер, Arduino, серводвигун, ESP8266.

ABSTRACT

Biloshevich O. Development of software and hardware complex "Robotized mechanical hand". Manuscript.

Qualification work of the bachelor's program "Software engineering" in the specialty "Software engineering". Lutsk National Technical University. Lutsk, 2024.

The purpose of the work is the theoretical justification and development of the hardware and software complex "Robotized Mechanical Hand". During the development of the complex, the functional requirements were studied and an algorithm for the control of the "Robotic mechanical arm" was developed. Based on the results of the qualification work, the software and hardware complex "Robotized mechanical hand" was developed and the device was tested.

Keywords: robotic mechanical arm, microcontroller, Arduino, servo motor, ESP8266.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ РОЗРОБКИ РОБОТИЗОВАНОЇ МЕХАНІЧНОЇ РУКИ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	15
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	17
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення	17
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання ...	20
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ «РОБОТИЗОВАНА МЕХАНІЧНА РУКА»	25
3.1 Практична реалізація об'єкта проектування.....	25
3.2 Тестування та налагодження програмно-апаратного комплексу «Роботизована механічна рука»	33
3.3 Розробка бази даних для програмно-апаратного комплексу «Роботизована механічна рука».....	36
ВИСНОВКИ	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	42

ВСТУП

Актуальність теми. Сучасні промислові потреби вимагають високоточних і надійних автоматизованих систем, що здатні виконувати складні маніпуляції з високою точністю і мінімальними людськими втручаннями. Застосування роботизованих рук у промисловості дозволяє значно підвищити ефективність виробничих процесів, знижуючи витрати та підвищуючи якість продукції.

Особливу актуальність має вивчення програмного забезпечення для роботизованих механічних рук. Створення інтуїтивно зрозумілого і ефективного програмного інтерфейсу дозволяє користувачам легко налаштовувати та управляти системою, а також інтегрувати її з іншими технологічними рішеннями. Розробка спеціальних алгоритмів для рухів та взаємодії з об'єктами в просторі забезпечує плавну та надійну роботу роботизованих систем.

Метою роботи є розробка програмно-апаратного комплексу «Роботизована механічна рука».

Об'єктом дослідження є програмно-апаратний комплекс «Роботизована механічна рука», яка використовує серводвигуни та контролери для точного виконання складних маніпуляцій.

Предметом дослідження є автоматизація виробничих процесів та поліпшення ефективності роботи.

Виходячи з мети роботи, було сформульовано наступні завдання:

- провести аналіз існуючих методів та пристроїв для програмно-апаратного комплексу «Роботизована механічна рука»;
- визначити функціональні вимоги створення програмно-апаратного комплексу «Роботизована механічна рука»;
- вибрати засоби створення та алгоритм керування програмно-апаратним комплексом «Роботизована механічна рука»;
- розробити програмно-апаратний комплекс «Роботизована механічна рука»;

- провести тестування та налагодження програмно-апаратного комплексу «Роботизована механічна рука»;
- створити базу даних, яка зберігатиме команди для складних переміщень «Роботизованої механічної руки» в просторі.

РОЗДІЛ 1

АНАЛІЗ РОЗРОБКИ РОБОТИЗОВАНОЇ МЕХАНІЧНОЇ РУКИ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

В епоху, коли автоматизація та інновації знаходяться в авангарді технологічного прогресу, роботизована зброя є символічною опорою нашого суспільства, що швидко розвивається. Ці універсальні та точні механічні інновації бездоганно інтегровані в безліч галузей промисловості, залишивши свій слід у виробництві, охороні здоров'я, дослідженні космосу тощо.

Роботизована рука – це механічний пристрій або маніпулятор, призначений для виконання різноманітних завдань, часто нагадуючи структуру людської руки з суглобами та сегментами. Зазвичай він оснащений датчиками, приводами та системами керування, що дозволяє йому маніпулювати об'єктами, рухатися з точністю та виконувати широкий спектр функцій автономно або під дистанційним керуванням людини.

Захоплююча подорож роботизованою зброєю сягає своїм корінням до народних казок грецької цивілізації, де геніальні автомати були створені для виконання елементарних завдань. Однак лише в середині 20 століття концепція роботизованої зброї почала справді формуватися.

Візіонер Айзек Азімов у своїх науково-фантастичних роботах представив ідею вдосконалених роботизованих кінцівок у 1940-х роках. Це викликало надзвичайний інтерес серед інженерів і новаторів, відкривши шлях до відчутного прогресу [1].

У 1961 році сталася важлива подія: на конвеєрі General Motors було встановлено Unimate, перший у світі промисловий робот. Це стало початком ери сучасних роботизованих маніпуляторів, які можуть виконувати повторювані й трудомісткі завдання з великою точністю і послідовністю. З роками робототехніка зазнала значного прогресу, від появи шарнірних маніпуляторів у 1970-х роках до інтеграції комп'ютерних систем управління, датчиків і штучного

інтелекту. Сьогодні роботизовані маніпулятори є складними багатофункціональними інструментами, які значно вплинули на різні галузі промисловості, демонструючи людську винахідливість та технічну майстерність (рис.1.1) [2].



Рисунок 1.1 – Перший у світі промисловий робот Unimate [2]

Сучасна робототехніка спрямована на створення та використання роботів, які звільняють людей від небезпечних або важких фізичних робіт, а також простих монотонних завдань, що не потребують високої кваліфікації. Виникла вона на основі механіки та кібернетики, стимулюючи їх подальший розвиток. Для механіки це означає розвиток багатоланкових механізмів, таких як маніпулятори, а для кібернетики – розвиток інтелектуального керування, яке дозволяє роботам виконувати завдання за допомогою штучного інтелекту.

Ці зміни дозволили значно покращити важливі характеристики роботів, зокрема:

- автономність (роботи можуть працювати автоматично за заданою програмою);
- універсальність (роботи здатні переміщувати об'єкти за складними траєкторіями в просторі);
- переналаштовуваність (роботи можуть адаптуватися до виконання різних завдань).

Компанія Universal-robots, що має своє походження в Данії, спеціалізується на виробництві роботів-маніпуляторів для автоматизації процесів у промисловості. Цей напрямок їх діяльності є відповіддю на постійний розвиток великого масштабу промисловості та зростання попиту на автоматизовані технології виробництва. В асортименті продукції компанія представила лінійку з трьох промислових маніпуляційних пристроїв з розімкненим кінематичним ланцюгом, включаючи моделі UR3, UR5 та UR10 (рис. 1.2).

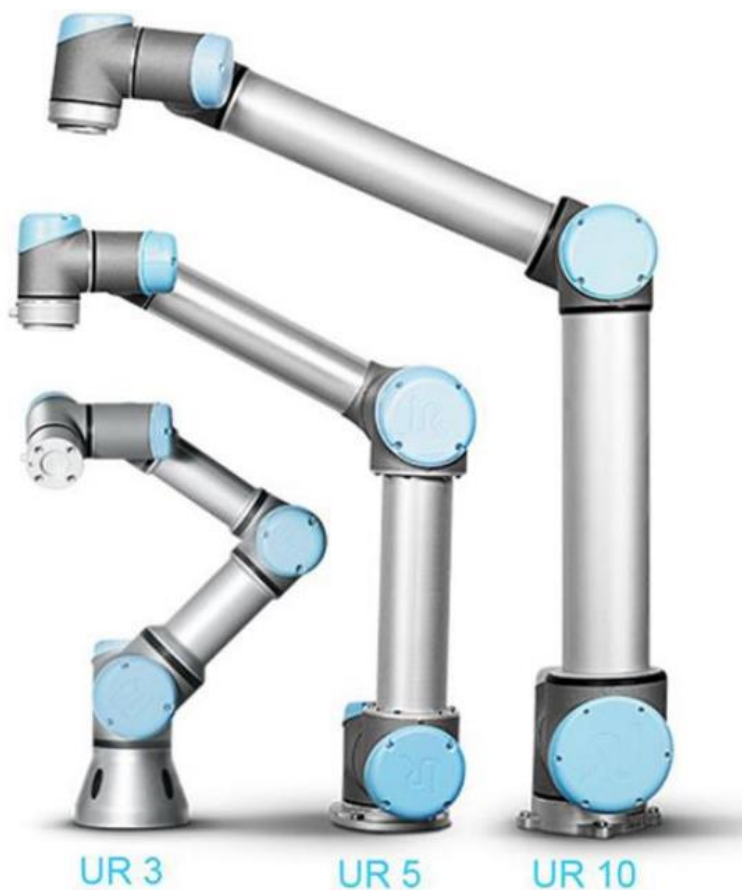


Рисунок 1.2 – Промислові маніпуляційні пристрої [3]

Кожен із цих моделей має свої унікальні особливості та пристосований для певного виду виробничих завдань у різних галузях промисловості. Роботи-маніпулятори компанії Universal-robots стали надійним помічником для багатьох підприємств, які прагнуть оптимізувати виробничі процеси та підвищити продуктивність. Вони ефективно впроваджуються в циклічні виробничі операції, забезпечуючи точність та швидкість виконання завдань. Висока технічна якість роботів-маніпуляторів UR3, UR5 та UR10 підтверджена бездоганною роботою цих пристроїв у реальних виробничих умовах. Адаптивність та гнучкість цих роботів позбавляють їх пристосовуватися до змінних потреб виробництва та легко інтегруватися в існуючі виробничі лінії. Компанія Universal-robots продовжує підтримувати високі стандарти якості своєї продукції та постійно розвивати нові технології для задоволення потреб клієнтів у сфері промислової автоматизації [3].

Розглянемо модель робота-маніпулятора від KUKA Robotics, як KR QUANTEC PA, яка має спеціальну функціональність, що дозволяє працювати в умовах низьких температур, аж до -30°C . Цей робот є ідеальним вибором для застосування в секторах, пов'язаних з харчовою їжею. промисловістю. Модель KUKA KR QUANTEC PA arctic спеціально розроблена для роботи з розкладанням палітри в морозильних камерах, не потребує додаткового захисного обладнання для роботи в низьких температурах. Крім того, робот має вбудовану електроніку, яка є стійкою до впливу морозу, готова до надійної роботи в екстремальному режимі (рис.1.3) [4].

Робот управляє передовими технологіями, спрямованими на ефективну оптимізацію виробничих процесів. Серед ключових характеристик можна виділити високу швидкість у максимальній робочій зоні, мінімальні виступаючі контури та надійність – все це необхідне для успішної автоматизації. Роботи, спроектовані спеціально для виконання складних завдань з навантаження та розвантаження.



Рисунок 1.3 – Робот-маніпулятор від KUKA Robotics [4]

Основними перевагами цієї моделі є швидкість роботи, висока продуктивність, компактні розміри та економічність. Детальніше розглянемо переваги робота-маніпулятора KR QUANTEC PA:

- забезпечення робочих циклів;
- мінімізація контурів перешкод;
- ефективне використання простору та висоти.

З точки зору тривалості робочих циклів, вона створена з найбільших серед роботів-маніпуляторів, які використовуються завданнями ринку монтажу та розвантаження. Модель KR QUANTEC PA має постійну продуктивність – навіть високу при коротких робочих циклах.

Усі роботи QUANTEC PA забезпечені маніпуляторами з пористим валом для вбудованої системи живлення та збереження місця. Ця легка та компактна конструкція дозволяє мінімізувати контури перешкод та забезпечити оптимальне прикладання захоплювачів.

Завдяки виносній конструкції та високій продуктивності KR QUANTEC PA не вимагає великої площі для монтажу. Це дає можливість розробляти новаторські концепції для роботи із завданнями розвантаження [4].

Поступово інновації в галузі робототехніки та біоніки поєднуються, що створює можливість для створення все більш пристосованих та функціональних пристроїв. Плавний перехід від роботів-маніпуляторів до біонічних рук відкриває нові перспективи в розвитку робототехніки, сприяючи покращенню якості життя та розвитку технологій.

I-LIMB Hand є відомою торговою маркою першої в світі доступної біонічної руки, яка була розроблена Девідом Гоу та його командою в центрі біоінженерії лікарні принцеси Маргарет Роуз у місті Єнбурзі. Система керування жестами дозволяє змінювати захоплення, переміщуючи I-Limb Quantum у будь-якому з чотирьох напрямків, а електричний привод великого пальця дозволяє автоматично змінювати положення.

I-Limb Quantum контролюється за допомогою міоелектричних сигналів, які набирають сигнали м'язів у залишковій частині руки для управління рухами рукою. Електроди розташовані в двох попередніх визначених місцях м'язів. Ці сигнали потім передаються до мікропроцесора, який активізує відповідну реакцію пристрою. I-Limb Quantum має можливість працювати з чотирма різними тригерами м'язів (рис.1.4.) [5].



Рисунок 1.4 – Лінійка міоелектричних роботизованих рук I-Limb Quantum [5]

Німецька компанія – Ottobock, яка спеціалізується на протезуванні. У 2017 році вона придбала Bebionic, роботизовану руку, від американської компанії Steeper, яка фокусувалася на розробці протезів, та перетворилася на одну з найбільш високотехнологічних компаній у галузі протезування.

Завдяки 14 різним варіантам стискання пальців протеза, Bebionic полегшує виконання повсякденних завдань і значно розширює спектр можливих операцій. Індивідуальні двигуни на кожні пальці можуть використовувати природні, координовані рухи захоплення (рис. 1.5) [6].



Рисунок 1.5 – Лінійка міоелектричних роботизованих рук Bebionic 2.0 [6]

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Виходячи з мети роботи, було сформульовано наступні завдання:

- провести аналіз існуючих методів та пристроїв для програмно-апаратного комплексу «Роботизована механічна рука»;
- визначити функціональні вимоги створення програмно-апаратного комплексу «Роботизована механічна рука»;

- вибрати засоби створення та алгоритм керування програмно-апаратним комплексом «Роботизована механічна рука»;
- розробити програмно-апаратний комплекс «Роботизована механічна рука»;
- провести тестування та налагодження програмно-апаратного комплексу «Роботизована механічна рука»;
- створити базу даних, яка зберігатиме команди для складних переміщень «Роботизованої механічної руки» в просторі.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Розробка програмно-апаратного комплексу «Роботизована механічна рука» значною мірою залежить від детального моделювання усіх його складових. При створенні проектів такого масштабу UML (Unified Modeling Language) виступає основною мовою для візуалізації і документування архітектури системи. UML дозволяє розробникам ефективно комунікувати між собою, а також забезпечувати зрозумілий і структурований підхід до процесу розробки.

Діаграма випадків використання (Use Case Diagram) описує взаємодії між користувачами (акторами) та системою. Наприклад, для нашого проекту актори включають користувача (оператора) та систему сенсорів. Випадки використання включають запуск і зупинку руки, вибір режиму роботи, та регулювання сили захвату. Ця діаграма допомагає визначити основні функції, які система мусить виконувати, а також зрозуміти способи, якими користувач буде взаємодіяти із системою. Вона стає прикладом структурованого підходу до визначення вимог, спрямованого на задоволення потреб користувачів. Діаграма випадків використання є важливим інструментом, що допомагає визначити межі системи та основні дії, які вона повинна виконувати. Кожен випадок використання представляє собою взаємодію системи з актором, де актор може бути як людиною, так і іншим системним компонентом або пристроєм. Діаграми випадків використання дозволяють відповісти на такі питання, як:

- що повинна робити система;
- як користувачі взаємодіють із системою;
- які основні функції повинні бути реалізовані;
- які вимоги будуть важливими для користувачів.

В нашому випадку оператор керує роботизованою рукою за допомогою інтерфейсу. Сенсори забезпечують зворотний зв'язок для коректного виконання дій, таких як захоплення об'єктів з певною силою (рис. 2.1).



Рисунок 2.1 – Структурна схема програмно-апаратного комплексу

Класова діаграма (Class Diagram) надає статичне уявлення про структуру системи, зокрема, демонструє взаємодію між класами, їхні властивості і методи. У випадку «Роботизованої механічної руки», основні класи включають клас RobotArm, який управляє сервомоторами. Клас RobotArm може мати властивості для збереження об'єктів сервомоторів та методи для керування ними. Скажімо, клас RobotArm має масив з трьох сервомоторів і методи для їхньої ініціалізації та управління, дозволяючи легко оновлювати або розширювати систему в майбутньому. Класова діаграма дозволяє виявити і описати основні компоненти системи та їхні взаємозв'язки. Кожен клас представляє собою окремий елемент системи з визначеними властивостями та методами, що описують його поведінку.

В нашому випадку клас RobotArm включає такі властивості, як серводвигуни, і методи для їхнього керування. Цей клас є центральним у нашій системі і відповідає за управління рухами руки. Крім класу RobotArm, система може також включати інші класи, такі як Sensor, Controller та UserInterface. Клас Sensor відповідає за збирання даних з сенсорів, контролер обробляє ці дані і

надсилає команди серводвигунам, а UserInterface забезпечує взаємодію оператора з системою.

Діаграма послідовностей (Sequence Diagram) ілюструє динамічні аспекти системи, допомагаючи зрозуміти взаємодію між об'єктами в часі. Вона демонструє послідовність повідомлень, що обмінюються між компонентами системи під час виконання конкретної функції. Для нашого проекту діаграма послідовностей описує процеси, пов'язані з керуванням рухами руки. Наприклад, коли користувач натискає кнопку запуску, Arduino ініціалізує серводвигуни та посилає сигнали для зміни їх позицій. Ця діаграма є важливим інструментом для визначення та аналізу динамічної поведінки системи, що забезпечує коректне функціонування під час виконання її основних функцій.

Такі діаграми допомагають виявити моменти взаємодії між різними об'єктами, що є важливими для досягнення поставлених цілей. Вони показують, як об'єкти можуть обмінюватись повідомленнями і викликати методи один одного. У випадку роботизованої руки, діаграма послідовностей може відобразити, як користувач через інтерфейс надсилає команди контролеру, який, в свою чергу, активує серводвигуни та обробляє дані з сенсорів (рис. 2.2).

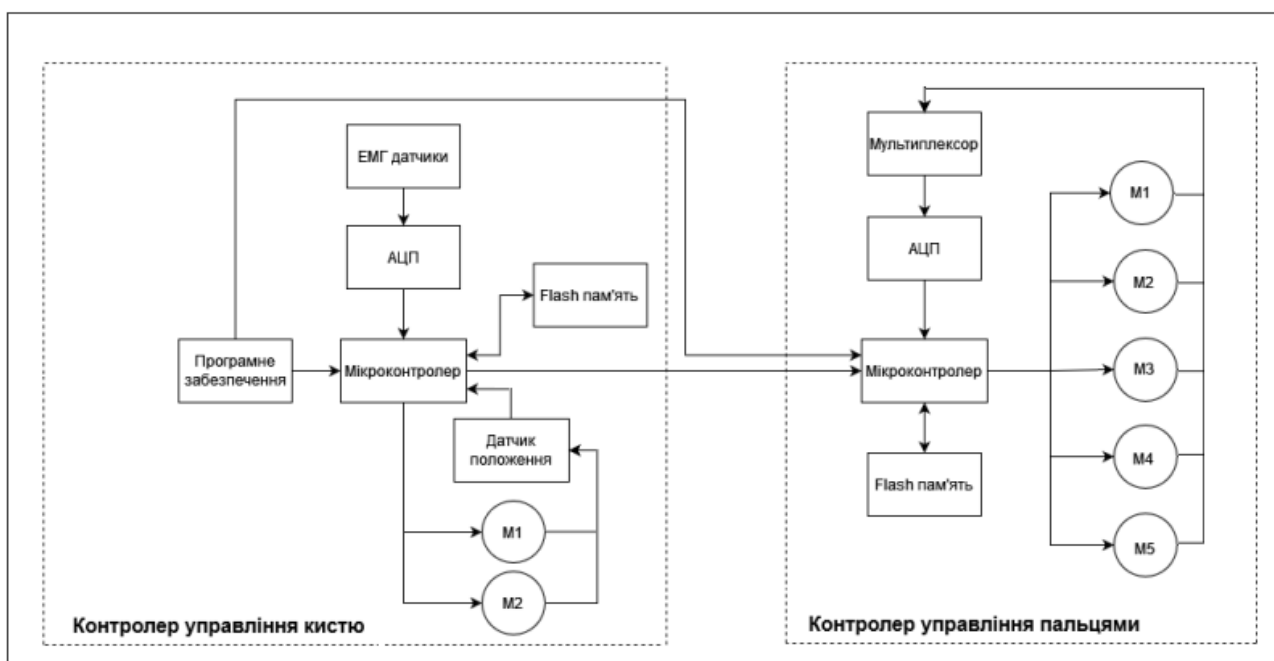


Рисунок 2.2 – Структурна схема управління роботизованою рукою

Використання у розробці структурних схем «Роботизованої механічної руки» надає численні переваги. По-перше, структурна схема надає візуальну мову, яка зрозуміла і розробникам, і зацікавленим сторонам, дозволяючи обговорювати і уточнювати вимоги до системи. Це знижує ризик неправильної інтерпретації вимог та допомагає забезпечити, що система буде розроблена відповідно до очікувань користувачів.

По-друге, структурна схема сприяє модульності та повторному використанню компонентів. Завдяки класовій діаграмі, компоненти системи можуть бути розділені на окремі модулі, кожен з яких має чітко визначені інтерфейси і функції. Це спрощує розробку, тестування та підтримку системи, дозволяючи легко оновлювати або замінювати окремі модулі без впливу на інші частини системи.

По-третє, структурна схема допомагає виявляти потенційні проблеми на ранніх етапах розробки. Діаграми дозволяють моделювати різні аспекти системи і аналізувати їх взаємодію ще до написання коду. Це допомагає ідентифікувати та вирішити проблеми, які можуть виникнути під час розробки, забезпечуючи тим самим більш стабільну і надійну систему.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Перш ніж приступити до розробки пристрою, необхідно вибрати оптимальну платформу на підходящому мікроконтролері (МК), щоб при підключенні всіх більшості модулів забезпечити коректну роботу. Це вимагає порівняння кількох МК для забезпечення сумісності та ефективної периферійності.

Під час розгляду МК для використання обраної платформи Arduino, яка є апаратною обчислювальною системою з ключовою роллю мікроконтролера. Для всебічного аналізу потрібно вивчити архітектуру кожного МК, остання сама відповідає для обробки даних та управління підключеними пристроями.

Далі слід, скільки аналогових та цифрових пінів потрібно для підключення всіх модулів, щоб уникнути конфліктів і забезпечити оптимальну роботу пристрою. Важливо врахувати обсяг оперативної пам'яті (ОЗП), потім від нього залежить швидкість проекту.

Також необхідно звернути увагу на флеш-пам'ять, після чого саме в ній зберігається програмне забезпечення та тимчасові дані, які можуть включити на подальше оновлення й функціональність пристрою.

Використання плати Arduino Uno в проектах є досить поширеним у ряді галузей, таких як освіта, розробка прототипів та творчі електронні проекти. Завдяки своїй компактній конструкції та функціональності Arduino Uno відмінно підходить для навчання основам програмування, робототехніки та електроніки. Ця плата забезпечує доступність для початківців, а також гнучкість і можливість для досвідчених користувачів для виконання різноманітних завдань.

Arduino Uno має вбудований мікроконтролер, який базується на ATmega328, із пам'яттю на 32 кілобайти флешки, 2 кілобайти оперативної пам'яті SRAM і 1 кілобайт EEPROM. Ці технічні характеристики можуть взаємодіяти з іншими периферійними пристроями та сенсорами (рис. 2.3) [7].



Рисунок 2.3 – Arduino Uno [7]

На платі Arduino Uno доступно 14 цифрових входів/виходів та 6 аналогових входів/виходів, що відкриває безліч можливостей для підключення

різноманітних датчиків, виконавчих пристроїв та інших компонентів. Ця різноманітність інтерфейсів дозволяє користувачам створювати проекти з автоматизації, вимірювання, робототехніки та інших областей.

Arduino Uno володіє відкритим програмним забезпеченням, що сприяє спільному розвитку користувачів, обміну досвідом та покращенню функціональності платформи. Це дозволяє використовувати широкий спектр бібліотек та розширити для реалізації різноманітних проектів.

Arduino Nano – це компактна і зручна для використання плата мікроконтролера з мікроконтролером ATmega328. Вона відома своєю невеликою розмірністю та широкими можливостями, що дозволяють використовувати її у різних проектах, починаючи від простих до складних. Arduino Nano має 14 цифрових входів/виводів, з яких 6 можуть використовуватися як ШИМ-виводи, та 8 аналогових входів. Вона також містить UART, SPI та I2C інтерфейси для комунікації з іншими мікроконтролерами або сенсорами. Плата живиться від 5V через USB або зовнішнє джерело живлення. Завдяки своїм невеликим розмірам і гнучкості, Arduino Nano ідеально підходить для вбудованих систем та прототипування (рис. 2.4) [8].



Рисунок 2.4 – Arduino Nano [8]

Arduino Mega – це потужна плата мікроконтролера, заснована на чіпі ATmega2560. Вона має значно більше введень/виводів, ніж стандартні моделі Arduino, зокрема 54 цифрових входи/виходи, з яких 15 підтримують ШИМ. Крім

того, вона має 16 аналогових входів, що дозволяє підключити більше сенсорів та пристроїв. Arduino Mega також оснащена чотирма апаратними UART для серійного зв'язку, що робить її ідеальним вибором для складніших комунікаційних завдань. Вона може живитися від USB або зовнішнього джерела живлення, що забезпечує гнучкість у використанні. Завдяки великій кількості пінів та розширеним можливостям введення/виведення, Arduino Mega підходить для масштабних проектів та розробки систем автоматизації (рис. 2.5) [9].

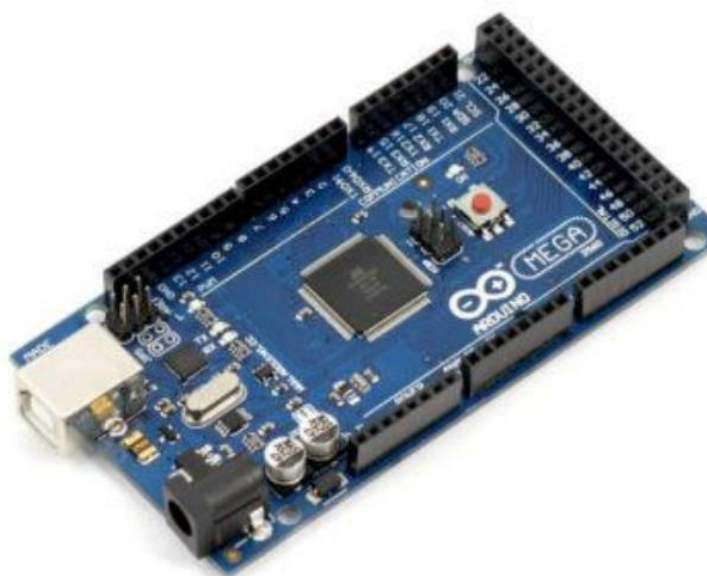


Рисунок 2.5 – Arduino Mega [9]

Wi-Fi-модуль ESP-01 з серії ESP8266 є популярним рішенням для інтеграції бездротового зв'язку у різноманітні проекти. Він має компактні розміри, що робить його зручним для використання у вбудованих системах та компактних пристроях. ESP-01 підтримує протоколи Wi-Fi 802.11 b/g/n, забезпечуючи стабільне з'єднання з бездротовими мережами. Модуль оснащений мікроконтролером Tensilica L106 32-біт, що працює з тактовою частотою до 160 МГц, що дозволяє йому обробляти складні задавання. Модуль оснащений 2 МБ флеш-пам'яті для зберігання прошивки та даних користувача. Завдяки підтримці інтерфейсів UART та GPIO, ESP-01 легко інтегрується з різними мікроконтролерами та сенсорами. Модуль також підтримує технології

шифрування, що забезпечує безпеку передаваних даних. Багата документація та численні приклади проектів роблять ESP-01 популярним вибором серед розробників (рис. 2.6) [10].

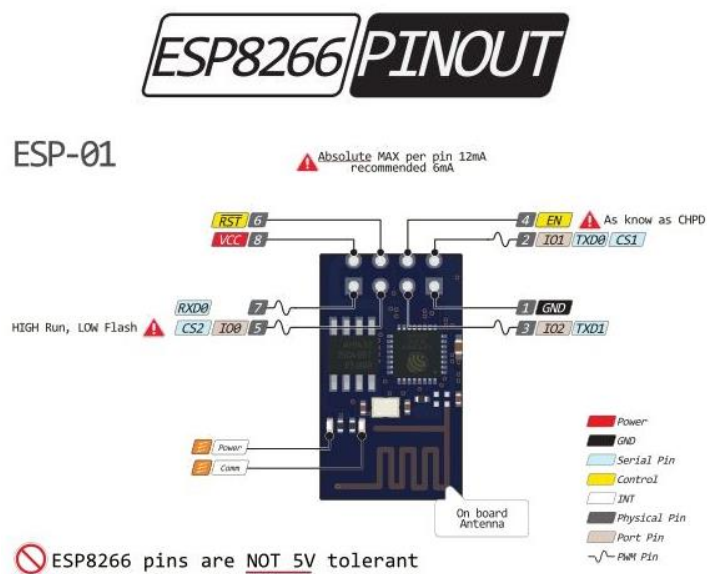


Рисунок 2.6 – Wi-Fi-модуль ESP-01 з серії ESP8266 [10]

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ «РОБОТИЗОВАНА МЕХАНІЧНА РУКА»

3.1 Практична реалізація об'єкта проектування

Основною мовою для розробки програмного забезпечення є C++, а апаратною платформою – Arduino. Використання Arduino дозволяє легко інтегрувати різні компоненти, такі як серводвигуни та сенсори, та забезпечує гнучкість у програмуванні.

На початку необхідно ініціалізувати серводвигуни та підключити їх до відповідних пінів на контролері Arduino, приклад коду наведено у лістингу 3.1.

Лістинг 3.1 – Керування серводвигунами

```
volatile int rotationCount = 0; // Лічильник обертів
volatile unsigned long lastRotationTime = 0; // Час останнього
оберту
#include <Servo.h>
// Створення об'єктів для кожного серводвигуна
Servo servo1;
Servo servo2;
Servo servo3;
void setup() {
    // Прикріплення сервомоторів до відповідних пінів
    servo1.attach(9); // Пін 9 для сервомотора 1
    servo2.attach(10); // Пін 10 для сервомотора 2
    servo3.attach(11); // Пін 11 для сервомотора 3}
void loop() {
    // Приклад рухів для кожного серводвигуна
    servo1.write(0); // Рух до 0 градусів
    delay(1000); // Затримка 1 секунда
    servo1.write(90); // Рух до 90 градусів
    delay(1000); // Затримка 1 секунда
```

```

servo1.write(180); // Рух до 180 градусів
delay(1000); // Затримка 1 секунда
servo2.write(0); // Рух до 0 градусів
delay(1000); // Затримка 1 секунда
servo2.write(90); // Рух до 90 градусів
delay(1000); // Затримка 1 секунда
servo2.write(180); // Рух до 180 градусів
delay(1000); // Затримка 1 секунда
servo3.write(0); // Рух до 0 градусів
delay(1000); // Затримка 1 секунда
servo3.write(90); // Рух до 90 градусів
delay(1000); // Затримка 1 секунда
servo3.write(180); // Рух до 180 градусів
delay(1000); // Затримка 1 секунда
}

```

Кінець лістингу 3.1

Лістинг 3.1 демонструє, як підключити три серводвигуни до Arduino та як керувати їхньою позицією. Важливо відзначити, що клас Servo, який використовується для роботи з серводвигунами, включає в себе всі необхідні функції для керування двигунами, що робить його простим у використанні [11].

Наступним кроком є інтеграція сенсорів, які забезпечують зворотний зв'язок для системи. Це дозволяє регулювати силу захоплення та інші параметри залежно від даних, отриманих із сенсорів. Приклад коду для інтеграції деяких базових сенсорів наведено у лістингу 3.2.

Лістинг 3.2 – Інтеграція сенсорів

```

#include <Servo.h>
class Sensor {
public:
    int readValue() {
        // Метод для зчитування даних з сенсора
    }
}

```

```

    // Це може бути аналогове або цифрове зчитування, залежно від
типу сенсора

    return analogRead(A0); // Наприклад, зчитування аналогового
значення з піну A0  }}
Sensor sensor;
class RobotArm {
private:
    Servo servos[3];
    Sensor sensor1;
public:
    RobotArm() {
        servos[0].attach(9);
        servos[1].attach(10);
        servos[2].attach(11);  }
    void moveServo(int index, int angle) {
        if (index >= 0 && index < 3) {
            servos[index].write(angle);  }  }
    void adjustGrip() {
        int sensorValue = sensor1.readValue();
        int gripForce = map(sensorValue, 0, 1023, 0, 180);
        moveServo(0, gripForce);  } };
RobotArm arm;
void setup() {}
void loop() {
    arm.adjustGrip();
    delay(1000);
}

```

Кінець лістингу 3.2

Лістингу 3.2 демонструє інтеграцію сенсора з системою та керування захопленням рук на основі зібраних даних. Метод `adjustGrip` зчитує значення з сенсора і на основі цього значення встановлює відповідну силу захвату, що забезпечує точнішу роботу системи [12].

Наступним кроком буде побудова електричної схеми. Почнемо з підключення Arduino до джерела живлення. Можна використовувати USB-кабель або зовнішній блок живлення, якщо серводвигуни потребують більшої потужності.

Далі підключіть живлення серводвигунів. Підключіть VCC кожного серводвигуна до 5V піну на Arduino. Це забезпечить необхідну потужність для їх роботи. Потім підключіть GND кожного серводвигуна до GND піну на Arduino, щоб забезпечити завершення живильного ланцюга. Забезпечення правильного заземлення є критичним для стабільної роботи системи, надійного захисту від електричних шумів (рис. 3.1).

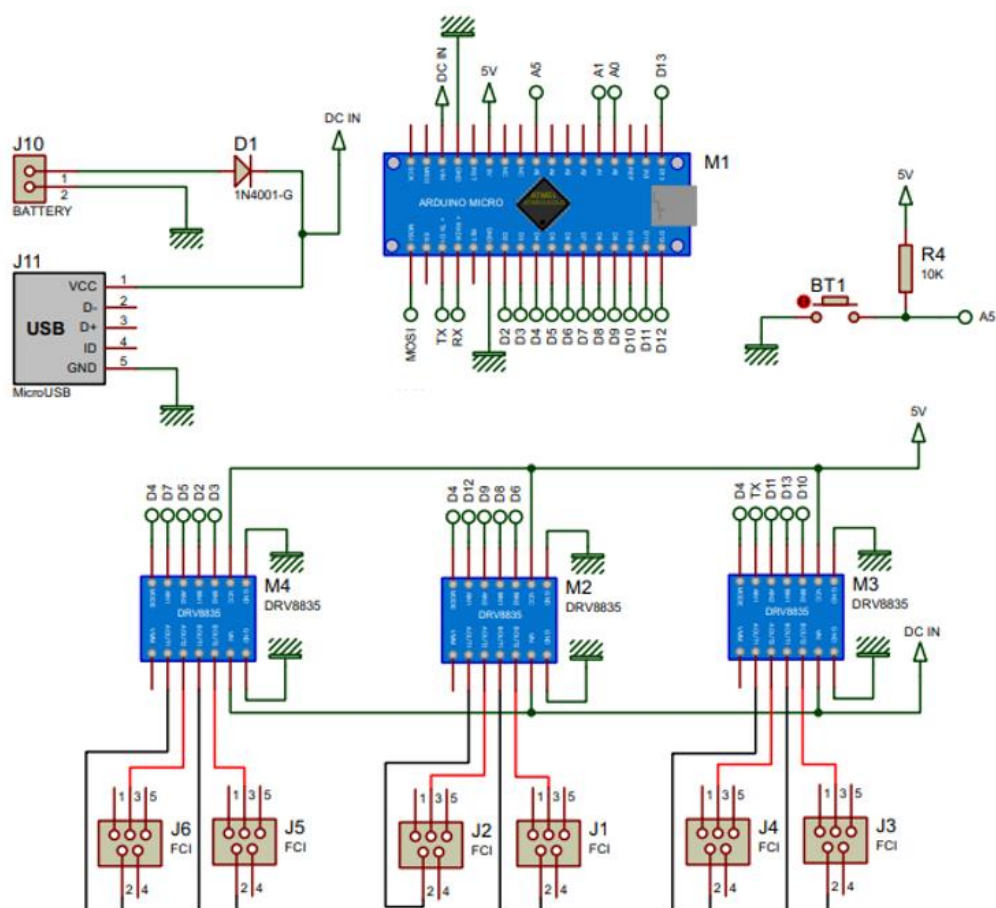


Рисунок 3.1 – Схематичне підключення потрібних компонентів

Після цього виконаємо підключенням сигнальних проводів серводвигунів. У нашому випадку їх п'ять і вони будуть підключені до аналогових пінів A1, A2,

A3, A4 і A5 на платі Arduino. Підключіть сигнальний провід першого серводвигуна до піну A1, другого серводвигуна до піну A2, третього серводвигуна до піну A3, четвертого серводвигуна до піну A4 і п'ятого серводвигуна до піну A5. Це дозволить контролеру Arduino керувати положенням кожного серводвигуна шляхом надсилання відповідних сигналів на ці піни (рис. 3.2).

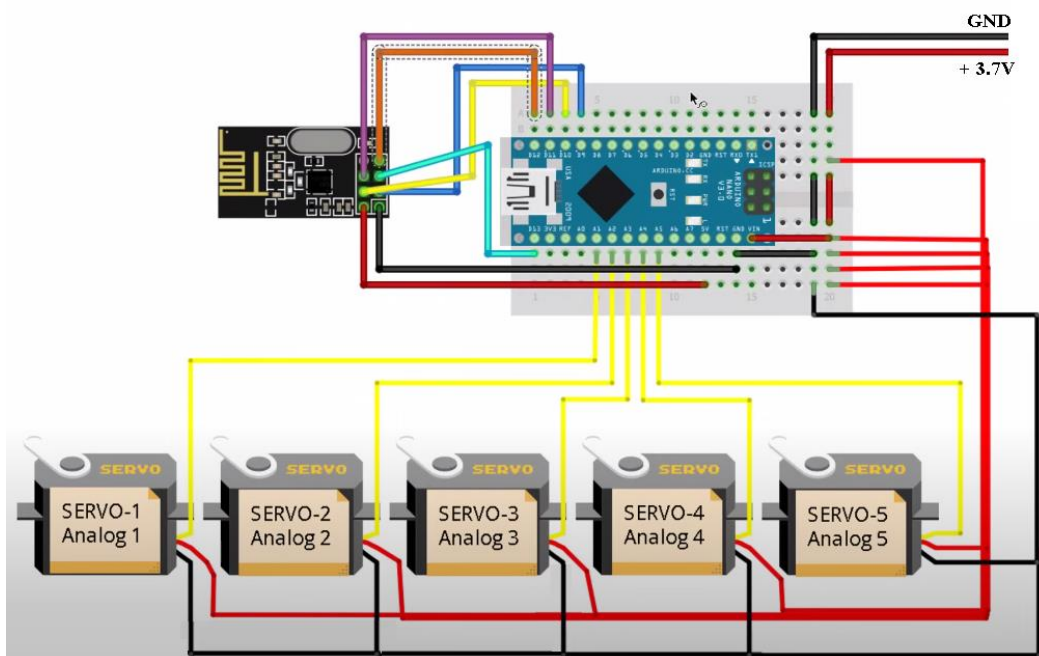


Рисунок 3.2 – Емуляція роботи серводвигунів

Наступним кроком після складання електронних компонентів є створення 3D моделі роботизованої руки. Цей етап є важливою частиною розробки, оскільки дозволяє візуалізувати конструкцію, оцінити її функціональність і забезпечити гармонійну інтеграцію всіх компонентів у кінцевий продукт. Все починається з визначення ключових вимог до моделі, таких як кількість ступенів свободи, розміри сегментів руки, типи механічних з'єднань і місця встановлення серводвигунів.

Першим етапом створення 3D моделі є розробка концептуального ескізу. На цьому етапі визначаються основні форми і розміри механічної руки. Це включає створення креслень і схематичних зображень, які допомагають уявити

загальну структуру виробу. Ці ескізи слугують основою для подальшої 3D візуалізації (рис. 3.3).

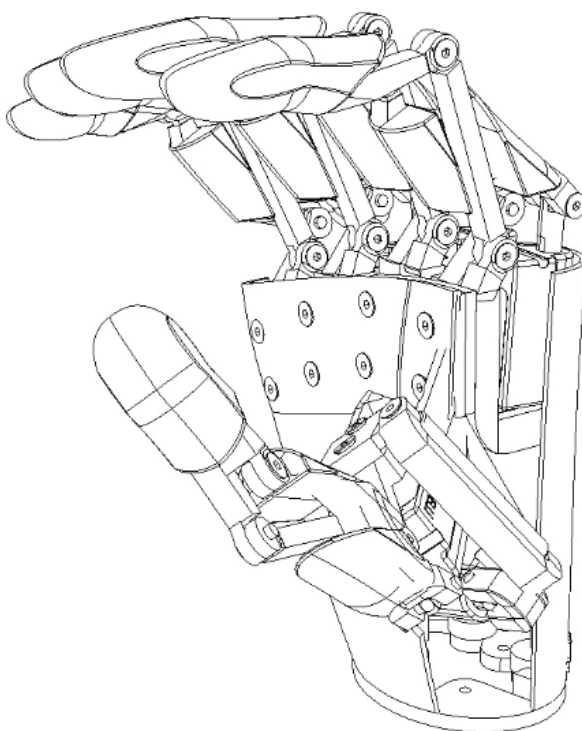


Рисунок 3.3 – Концептуальний ескіз роботизованої механічної руки

Після завершення ескізу, переходимо до створення тривимірної моделі у спеціалізованих програмах для 3D моделювання, таких як Autodesk Fusion 360, SolidWorks або Blender. Ці програми дозволяють створювати детальні тривимірні моделі з найменшими подробицями. Моделювання починається з окремих частин руки: базового сегмента, кисті та пальців. Кожна частина моделюється окремо з урахуванням габаритних розмірів і точок кріплення.

По завершенні моделювання окремих частин, їх об'єднують у збірний модуль, де забезпечують правильне положення всіх елементів і механічних з'єднань. Для цього використовуються функції з'єднання компонентів у 3D програмі, це дозволяє перевірити рухливість та взаємодію між сегментами.

Критично важливим етапом є перевірка всіх механічних з'єднань та рухомих частин на можливість зіткнень і правильність кінематичних ланцюгів.

Така перевірка дозволяє виявити потенційні проблеми та внести корективи в проект до переходу до фізичного виготовлення компонентів (рис. 3.4).

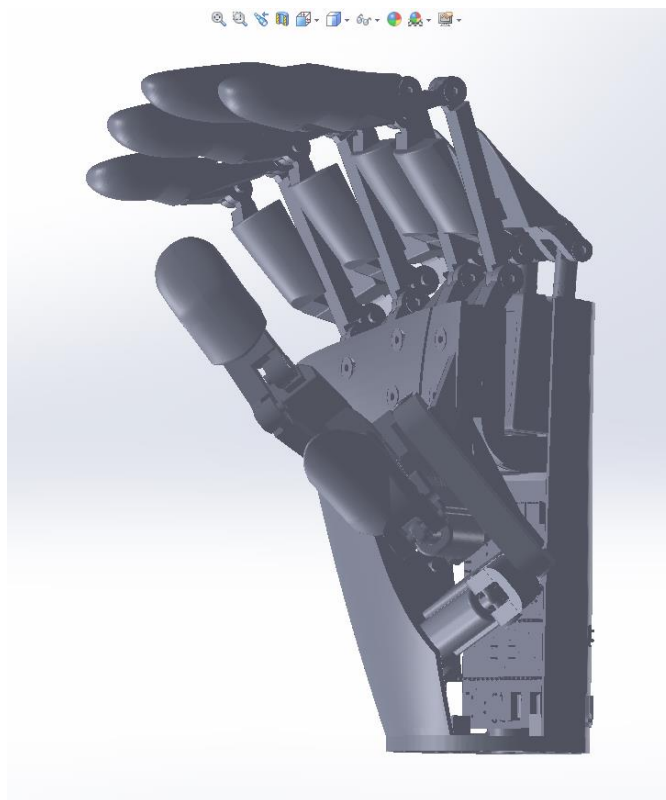


Рисунок 3.4 – 3D модель руки у solidworks

Особлива увага приділяється місцям кріплення серводвигунів, оскільки від їх розташування залежить точність і плавність рухів всієї руки. Важливо забезпечити надійне кріплення двигунів, щоб вони могли ефективно передавати рухи на сегменти без зайвого тертя або люфтів.

Далі моделі сегментів руки адаптуються для друку на 3D принтері. Це передбачає перевірку моделей на відповідність можливостям принтера, вибір матеріалів, що відповідають вимогам до міцності та гнучкості, і оптимізацію для мінімізації кількості підтримок під час друку. Це важливо для забезпечення гладкості поверхонь та точності деталей.

Матеріали для друку зазвичай вибирають серед полімерів, таких як PLA або ABS, що відрізняються високою міцністю та достатньою легкістю для

складних маніпуляцій. Після підготовки моделей, їх експортують у формати, зручні для 3D друку, такі як STL або OBJ файли.

Після друку окремих компонентів, переходять до їх складання. На цьому етапі важливо перевірити всі механічні з'єднання, щоб уникнути люфтів або неправильних рухів. Сегменти з'єднуються за допомогою гвинтів, гайок та спеціальних кріплень, щоб забезпечити надійність і міцність конструкції (рис. 3.5).

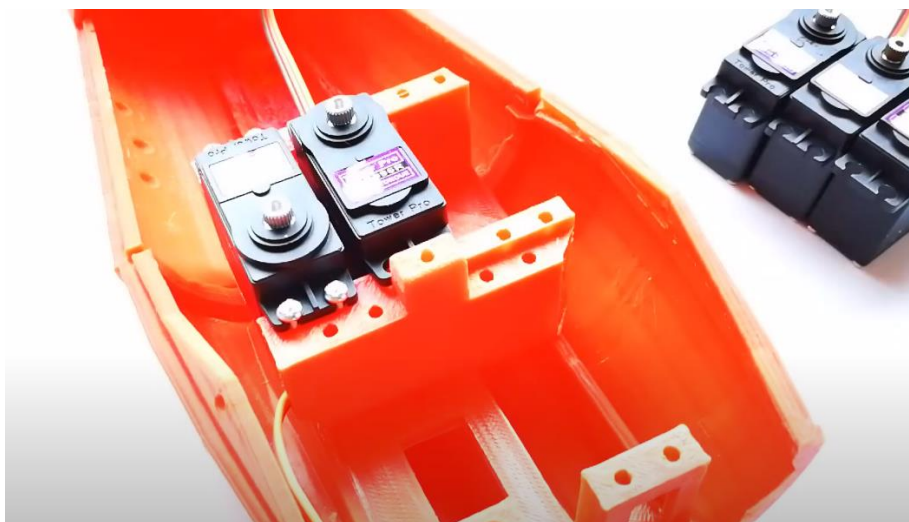


Рисунок 3.5 – Монтаж серводвигунів

Після завершення складання кожної частини, встановлюються серводвигуни на свої місця та з'єднуються згідно з проектними рішеннями, зокрема із використанням зубчастих передач або важелів для передачі рухів від двигунів до сегментів руки. Це забезпечує чіткість і плавність у виконанні рухів.

На останньому етапі проводиться тестування готової 3D моделі в зібраному вигляді. Це включає перевірку коректності усіх з'єднань, функціональності руки та надійності конструкції. У разі виявлення будь-яких дефектів або недоліків, вносяться необхідні коригування, щоб забезпечити оптимальну роботу механічної руки. Це дозволяє підготувати конструкцію до впровадження в реальних умовах експлуатації (рис. 3.6).



Рисунок 3.6 – Роботизована механічна рука

Отже, створення 3D моделі роботизованої руки є критичним етапом, що дозволяє належним чином підготувати всі компоненти до подальшої інтеграції та налагодження. Він забезпечує можливість виявлення та виправлення потенційних проблем ще до виготовлення фізичних компонентів, що суттєво підвищує якість і надійність кінцевого продукту.

3.2 Тестування та налагодження програмно-апаратного комплексу «Роботизована механічна рука»

Тестування та налагодження програмно-апаратного комплексу «Роботизована механічна рука» є важливим етапом, що забезпечує функціональність та надійність системи. Одне з ключових завдань у цьому

процесі – тестування протоколу віддаленого керування серводвигунами. Відповідність протоколу вимогам і його правильна робота є критичними для успішного виконання команди оператором на відстані.

Для початку необхідно розробити детальний план тестування. Цей план має включати всі аспекти протоколу, такі як формат повідомлень, валідація даних, передача і прийом команд, та обробка помилок. Такий підхід допоможе систематично перевірити кожен елемент протоколу і забезпечити його коректну роботу в реальних умовах.

Тестування починається з перевірки базової функціональності протоколу передачі команд. Наприклад, команда для зміни кута серводвигуна може виглядати як «SET_ANGLE 90», де числове значення вказує на кут. Спочатку необхідно протестувати, чи правильно розпізнається і обробляється кожна команда на стороні приймача.

Наступним кроком є валідація даних. Дуже важливо переконатися, що всі команди містять коректні дані. Неправильні або некоректні дані можуть призвести до неправильного функціонування системи. Наприклад, якщо команда містить кут, який перевищує допустимий для серводвигуна, система повинна відповідно реагувати, наприклад, відмовити у виконанні команди і повідомити про помилку.

Після цього тестуються сценарії, коли команди надсилаються з помилками або з відсутніми частинами. Система повинна правильно обробляти такі ситуації і не виконувати неправильні або неповні команди. Перевіряється, чи відправляється відповідь про помилку та чи зберігається стабільність системи у таких випадках.

Далі важливо перевірити синхронізацію між передавачем і приймачем. Це включає тестування затримок під час передачі команд та забезпечення їх коректної обробки у порядку надходження. Наприклад, при передачі кількох команд поспіль, таких як «SET_ANGLE 45», «SET_ANGLE 90», система має виконувати їх у правильному порядку без затримок або пропусків.

Також необхідно проводити тести на стійкість зв'язку, які передбачають симуляцію різних умов зв'язку, таких як мережеві затримки, обрив зв'язку і повторні спроби передачі команд. Система повинна підтримувати стабільну роботу у таких випадках і мати механізми для відновлення зв'язку та повторної відправки важливих команд.

Особливої уваги потребує тестування безпеки протоколу. Це включає перевірку на можливість несанкціонованого доступу, захист даних під час передачі та виявлення підозрілих дій. Важливо переконатися, що тільки авторизовані користувачі мають доступ до керування системою, а всі командні дані захищені за допомогою шифрування.

Додатково необхідно протестувати протокол у реальних умовах роботи, тобто в умовах, близьких до тих, де система буде використовуватися. Це допоможе виявити можливі проблеми, які можуть виникати у справжньому середовищі, такі як електромагнітні завади, фізичні перешкоди для сигналу або інтенсивне використання системи.

Після завершення тестування проводиться аналіз результатів. На основі цього аналізу можуть вноситися зміни або покращення до протоколу, щоб забезпечити його стабільну і коректну роботу. Всі виявлені помилки документуються та фіксуються методи їх усунення. Це допомагає створити відчуття завершеності та задокументованості процесу тестування.

Для забезпечення безперервного контролю якості важливо вбудувати механізми моніторингу в кінцеву систему. Це дозволить в реальному часі відслідковувати можливі проблеми або відхилення у роботі протоколу і оперативно реагувати на них. Постійний моніторинг допомагає підтримувати систему в робочому стані та своєчасно виявляти потенційні загрози.

Один з аспектів тестування включає перевірку взаємодії з іншими системами та пристроями. Це може включати перевірку інтеграції з інтерфейсами користувача, іншими роботизованими системами або мережевими компонентами. Взаємодія з іншими системами має бути гладкою і

безперебійною, тому важливо оцінити, як протокол поводить себе в умовах складної конфігурації мережі.

Цілісне тестування та налагодження протоколу віддаленого керування є важливим аспектом створення надійного програмно-апаратного комплексу «Роботизована механічна рука». Виконані тести забезпечують впевненість в тому, що система готова до реального використання і відповідає усім вимогам надійності та безпеки. Цей принцип надійності й безперервної перевірки забезпечує довір'я користувачів до кінцевого продукту, а розробникам дає впевненість у стійкості створеної системи.

3.3 Розробка бази даних для програмно-апаратного комплексу «Роботизована механічна рука»

Опишемо процес створення бази даних, яка зберігатиме команди для складних переміщень «Роботизованої механічної руки» в просторі.

База даних для команд повинна включати наступні таблиці:

- commands;
- servo_movements;
- coordinates.

Таблиця commands буде зберігати загальну інформацію про кожну команду, включаючи її унікальний ідентифікатор, назву, тип команди та час створення. Приклад коду для створення таблиці commands наведено у лістингу 3.3.

Лістинг 3.3 – Створення таблиці commands

```
CREATE TABLE commands
(
    id INT AUTO_INCREMENT PRIMARY KEY,
    command_name VARCHAR(50) NOT NULL,
    command_type ENUM('move', 'grasp', 'release') NOT NULL,
```

```
timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Кінець лістингу 3.3

Таблиця `servo_movements` зберігає детальну інформацію про рухи кожного серводвигуна при виконанні команди. Приклад коду для створення таблиці `servo_movements` наведено у лістингу 3.4.

Лістинг 3.4 – Створення таблиці `servo_movements`

```
CREATE TABLE servo_movements
(
    id INT AUTO_INCREMENT PRIMARY KEY,
    command_id INT,
    servo_id INT,
    start_angle INT NOT NULL,
    end_angle INT NOT NULL,
    duration INT NOT NULL,
    FOREIGN KEY (command_id) REFERENCES commands(id),
    FOREIGN KEY (servo_id) REFERENCES servos(id)
);
```

Кінець лістингу 3.4

Таблиця `coordinates` зберігає координати кінцевих точок переміщення руки. Приклад коду для створення таблиці `coordinates` наведено у лістингу 3.5.

Лістинг 3.5 – Створення таблиці `coordinates`

```
CREATE TABLE coordinates
(
    id INT AUTO_INCREMENT PRIMARY KEY,
    command_id INT,
    x_coordinate FLOAT NOT NULL,
```

```

    y_coordinate FLOAT NOT NULL,
    z_coordinate FLOAT NOT NULL,
    FOREIGN KEY (command_id) REFERENCES commands(id)
);

```

Кінець лістингу 3.5

Для подальшої роботи з базою даних необхідно заповнити таблиці початковими даними. Припустимо, ми хочемо додати команду переміщення руки до певної точки. Приклад коду заповнення таблиць даними наведено у лістингу 3.6.

Лістинг 3.6 – Заповнення таблиць даними

```

-- Додавання команди переміщення
INSERT INTO commands (command_name, command_type)
VALUES ('Move to Position', 'move');

-- Додавання координат для команди
INSERT INTO coordinates (command_id, x_coordinate, y_coordinate,
z_coordinate)
VALUES (LAST_INSERT_ID(), 10.5, 20.7, 30.2);

-- Додавання рухів серводвигунів для виконання команди
INSERT INTO servo_movements (command_id, servo_id, start_angle,
end_angle, duration)
VALUES (LAST_INSERT_ID(), 1, 0, 90, 1000),
        (LAST_INSERT_ID(), 2, 45, 135, 1500),
        (LAST_INSERT_ID(), 3, 90, 180, 2000),
        (LAST_INSERT_ID(), 4, 30, 60, 1200),
        (LAST_INSERT_ID(), 5, 10, 70, 800);

```

Кінець лістингу 3.6

Створимо GET-запит для отримання команди задля інтеграції з Arduino через API. Приклад написаного на python GET-запиту наведено у лістингу 3.7.

Лістинг 3.7 – GET-запит для отримання команди

```
@app.route('/command/<int:id>', methods=['GET'])
def get_command(id):
    cursor = db.cursor(dictionary=True)
    # Отримання основної інформації про команду
    cursor.execute("SELECT * FROM commands WHERE id = %s", (id,))
    command = cursor.fetchone()
    # Отримання координат для команди
    cursor.execute("SELECT * FROM coordinates WHERE command_id = %s", (id,))
    coordinates = cursor.fetchall()
    # Отримання рухів серводвигунів для команди
    cursor.execute("SELECT * FROM servo_movements WHERE command_id = %s", (id,))
    movements = cursor.fetchall()
    response = {
        "command": command,
        "coordinates": coordinates,
        "movements": movements
    }
    return jsonify(response)
```

Кінець лістингу 3.7

Після налаштування всіх компонентів потрібно:

- переконатися, що Arduino успішно з'єднується з Wi-Fi та може надсилати запити до серверу;
- переконатися, що сервер коректно відповідає на запити, та що дані за потребою оновлюються;
- переконатися, що серводвигуни коректно реагують на отримані команди.

Навчання та налагодження системи забезпечить стабільність та надійність роботи «Роботизованої механічної руки» в умовах реального часу, здібність виконувати складні маніпуляції з точністю та ефективністю.

ВИСНОВКИ

У сучасному світі робототехніка відіграє важливу роль у розвитку технологій та інновацій. Розробка програмно-апаратного комплексу «Роботизована механічна рука» є ключовим напрямком, який має потенціал значно підвищити продуктивність та ефективність у різних галузях, таких як промисловість, медицина та обслуговування. Така система дозволяє виконувати завдання, що вимагають високої точності та надійності, з мінімальним втручанням людини.

Проведений аналіз сучасних технологій показує, що існує широкий спектр методів і пристроїв для створення роботизованих механічних рук. Серед них виділяються електромеханічні та гідравлічні системи, які мають різні переваги та недоліки. Основними критеріями вибору є точність, надійність і вартість виробництва. Більшість сучасних роботизованих систем використовують програмовані контролери для забезпечення гнучкості та адаптивності управління.

Функціональні вимоги до створення роботизованої механічної руки включають високу точність рухів, надійність та здатність виконувати складні маніпуляції. Важливо забезпечити інтеграцію системи з різними сенсорами для отримання зворотного зв'язку. Комплекс повинен мати модульну структуру, що дозволяє легко змінювати або вдосконалювати окремі компоненти. Необхідно передбачити можливість програмування та налаштування під специфічні завдання користувача. Важливим аспектом є забезпечення безпеки як для самого пристрою, так і для оточуючих його людей.

Вибір засобів створення включає підбір відповідних електронних компонентів, матеріалів для механічних частин та програмних інструментів. Основним контролером може бути мікроконтролер або мікропроцесор, який забезпечить необхідну обчислювальну потужність. Для забезпечення високої точності рухів використовуються серводвигуни з вбудованими енкодерами. Алгоритм керування повинен враховувати кінематику та динаміку

роботизованої руки, забезпечуючи плавність і точність рухів. Використання методів машинного навчання дозволить системі адаптуватися до нових умов і оптимізувати виконання завдань. Програмне забезпечення повинно бути модульним, що дозволить легко додавати нові функції та проводити модернізацію комплексу.

Розробка програмно-апаратного комплексу включає кілька етапів, від проектування та виготовлення механічних частин до розробки електроніки та програмного забезпечення. На початковому етапі створюється прототип, який дозволяє перевірити основні технічні рішення та їх сумісність. Після успішного тестування прототипу проводиться розробка фінальної версії комплексу, з урахуванням всіх вимог і специфікацій. Важливим аспектом є інтеграція всіх компонентів та забезпечення їх коректної взаємодії.

Тестування та налагодження є критичними етапами, що забезпечують надійність та функціональність роботизованої механічної руки. На початковому етапі проводиться функціональне тестування для перевірки виконання основних завдань. Важливим аспектом є перевірка точності та повторюваності рухів, а також здатності системи адаптуватися до різних умов. Налагодження включає оптимізацію алгоритмів керування та налаштування сенсорних систем. Проведення стрес-тестів дозволяє оцінити надійність та стійкість системи до зовнішніх впливів.

База даних для зберігання команд є ключовим компонентом, що забезпечує гнучкість та ефективність роботи роботизованої механічної руки. Вона повинна підтримувати швидкий доступ до інформації та можливість збереження великої кількості різних команд. Для підвищення продуктивності роботи системи використовується індексація та оптимізація запитів до бази даних. База даних повинна мати можливість легкої інтеграції з іншими системами та програмним забезпеченням.

У підсумку, розробка програмно-апаратного комплексу «Роботизована механічна рука» є складним, але перспективним завданням, яке відкриває нові можливості для автоматизації та підвищення ефективності різних процесів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Айзек Азімов. URL: https://uk.wikipedia.org/wiki/%D0%90%D0%B9%D0%B7_%D0%B5%D0%BA_%D0%90%D0%B7%D1%96%D0%BC%D0%BE%D0%B2 (дата звернення: 5.02.2024 р.).
2. General Motors. URL: <https://habr.com/companies/smileexpo/articles/413591/> (дата звернення: 6.02.2024 р.).
3. Universal-robots. URL: <https://www.universal-robots.com/> (дата звернення: 6.02.2024 р.).
4. KUKA Robotics. URL: <https://www.kuka.com/en-se/company/press/news/2023/02/robot-express-packs> (дата звернення: 20.02.2024 р.).
5. i-Limb Bionic Hand Prosthetic. URL: <https://www.llop.com/ilimb/> (дата звернення: 23.02.2024 р.).
6. Ottobock. URL: <https://www.ottobock.com/en-us/home> (дата звернення: 25.02.2024 р.).
7. Arduino Uno. URL: <https://doc.arduino.ua/hardware/Uno> (дата звернення: 8.03.2024 р.).
8. Arduino Nano. URL: <https://arduino.ua/prod166-arduino-nano-v3-0-avr-atmega328p-s-raspayannimi-razemami> (дата звернення: 9.03.2024 р.).
9. Arduino Mega. URL: <https://doc.arduino.ua/ru/hardware/Mega2560> (дата звернення: 10.03.2024 р.).
10. Wi-Fi-модуль ESP-01 з серії ESP8266. URL: <https://arduino.ua/prod980-wifi-modul-esp8266> (дата звернення: 12.03.2024 р.).
11. Allows Arduino boards to control a variety of servo motors. URL: <https://www.arduino.cc/reference/en/libraries/servo/> (дата звернення: 10.05.2024 р.).
12. Force Sensing Resistor Applications In Advanced Robotics Solutions. URL: <https://intorobotics.com/force-sensing-resistor/> (дата звернення: 11.05.2024 р.).