

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та кібербезпеки

(повне найменування кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**РОБОТИЗОВАНА РУХОМА РУКА НА ПЛАТФОРМИ ARDUINO
UNO**

ROBOTIC MOBILE HAND ON ARDUINO UNO PLATFORM

спеціальність 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія
(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІс-21
Степасюк Михайло Сергійович

(підпис)

Керівник:
к.т.н., старший викладач
Гринюк Сергій Васильович

(підпис)

Кваліфікаційну роботу
допущено до захисту
« _____ » червня 2023 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2023 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та кібербезпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

проф. Н.Черняшук

« _____ » _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Степасюку Михайлу Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Роботизована рухома рука на платформі Arduino UNO

Керівник роботи Гринюк Сергій Васильович, к.т.н., старший викладач

затверджені наказом закладу вищої освіти від «28» грудня 2022 року № 982/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 01.06.2023р.

3. Вихідні дані до роботи Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області та різні інтернет-ресурси технічного спрямування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

Вступ

Аналітична частина

Теоретична частина

Розробка роботизованої рухомої руки

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Плата Arduino UNO R3

Інтегроване середовище розробки Arduino(IDE)

Модуль Bluetooth HC-05

Макетна схема підключення елементів робота

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Загальні відомості</i>	<i>Гринюк С.В.</i>		
<i>Детальний аналіз засобів розробки</i>	<i>Гринюк С.В.</i>		
<i>Розробка роботизованої рухомої руки</i>	<i>Гринюк С.В.</i>		
<i>Висновки</i>	<i>Гринюк С.В.</i>		

7. Дата видачі завдання 01.11.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Обґрунтування теми кваліфікаційної роботи</i>	10.11.2022 р.	Виконано
2.	<i>Огляд літературних джерел</i>	22.11.2022 р.	Виконано
3.	<i>Дослідження предметної області</i>	12.01.2023 р.	Виконано
4.	<i>Вибір апаратної платформи та середовища розробки</i>	14.02.2023 р.	Виконано
5.	<i>Розробка програмно-апаратних засобів керування роботом</i>	26.03.2023 р.	Виконано
6.	<i>Оформлення матеріалів роботи</i>	03.04.2023 р.	Виконано
7.	<i>Нормоконтроль</i>	17.05.2023 р.	Виконано
8.	<i>Інструментальна перевірка на академічний плагіат</i>	02.06.2023 р.	Виконано
9.	<i>Представлення кваліфікаційної роботи бакалавра до захисту</i>	16.06.2023 р.	Виконано

Здобувач вищої освіти

(підпис)

(Степасюк М.С.)

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

(Гринюк С.В.)

(прізвище, ініціали)

АНОТАЦІЯ

Степасюк М.С. Роботизована рухома рука на платформі Arduino UNO.
Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2023.

Кваліфікаційна робота бакалавра складається з вступу, трьох розділів, висновків, списку використаних джерел, додатків.

В першому розділі проведено дослідження, проаналізовані аналогічні готові рішення та вимоги до систем керування роботами-маніпуляторами.

В другому розділі описується розроблюваний пристрій та здійснюється вибір апаратної платформи та середовища програмування.

В третьому розділі описується підбір елементної бази робота-маніпулятора, розробка плати розширення та проектування вузлів роботизованої рухомої руки.

Область дослідження – прототип роботизованої рухомої руки.

Предмет дослідження – послідовний процес розробки та програмування роботизованої рухомої руки на базі мікроконтролерної платформи Arduino UNO

Областю застосування розроблюваного пристрою є навчальні заклади та дослідні лабораторії.

Ключові слова: робототехніка, роботизована рука, мікроконтролерна техніка, Arduino.

ANNOTATION

Stepasyuk M.S. A robotic mobile arm on the Arduino UNO platform. Manuscript. Bachelor's qualifying thesis of the OP "Computer Engineering" specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2023.

The bachelor's qualification work consists of an introduction, three sections, conclusions, a list of used sources, and appendices.

In the first chapter, research was conducted, similar ready-made solutions and requirements for manipulator robot control systems were analyzed.

The second section describes the device under development and selects the hardware platform and programming environment.

The third chapter describes the selection of the element base of the manipulator robot, the development of the expansion board and the design of the nodes of the robotic mobile arm.

The field of research is a prototype of a robotic mobile arm.

The subject of the research is the sequential process of development and programming of a robotic mobile arm based on the Arduino UNO microcontroller platform

The field of application of the developed device is educational institutions and research laboratories.

Keywords: robotics, robotic hand, microcontroller technology, Arduino.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1 АНАЛІТИЧНА ЧАСТИНА	6
1.1 Огляд та аналіз області використання роботів-маніпуляторів	6
1.2 Аналіз аналогічних продуктів	12
1.3 Вимоги до системи керування роботом-маніпулятором.....	15
РОЗДІЛ 2 ТЕОРЕТИЧНА ЧАСТИНА	19
2.1 Загальний опис розроблюваного пристрою	19
2.2 Обґрунтування вибору апаратної мікроконтролерної платформи	20
2.3 Середовище програмування Arduino IDE	24
РОЗДІЛ 3 РОЗРОБКА РОБОТИЗОВАНОЇ РУХОМОЇ РУКИ	27
3.1 Визначення загальної конструкції пристрою	27
3.2 Підбір елементної бази	28
3.2.1 Мікроконтролерна платформа.....	28
3.2.2 Сервоприводи та крокові двигуни.....	31
3.2.3 Модуль комунікації.....	35
3.3 Схема підключення елементів роботизованої руки	36
3.4 Проектування плати розширення Arduino UNO.....	39
3.5 Проектування вузлів роботизованої руки.....	42
ВИСНОВКИ	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТКИ	53

ВСТУП

Актуальність дослідження Роботизована рухома рука (робот-маніпулятор) – це механічна система, яка частково або повністю імітує структуру та рухи людської руки з метою виконання різноманітних завдань, таких як збирання, переміщення та розміщення об'єктів. Вона складається з кількох сегментів, з'єднаних між собою рухомими з'єднаннями, що надають руці широкий діапазон рухів. В залежності від завдань, які має виконувати роботизована рухома рука, вона може мати різну кількість і типи сегментів, актуаторів та датчиків.

Роботизовані рухомі руки знаходять застосування в різних сферах і залежно від сфери використання роботи-маніпулятори можуть бути використані для автоматизації виробничих ліній, проведення хірургічних операцій, різних досліджень в сфері науки, обслуговування клієнтів готелів та ресторанів, демінування територій та інших військових завдань.

Роботизовані маніпулятори можуть виконувати точні рухи в повторюваних циклах без зміни параметрів, вони можуть виконувати рухи значно швидше, ніж людина, можуть виконувати роботи у небезпечних умовах, наприклад, в радіаційно забруднених зонах, під водою або в просторах, де не можуть працювати люди, можуть бути автоматизовані для виконання задач без участі людей, що в свою чергу може знизити ризик виникнення помилок.

Мета дослідження полягає у створенні роботизованої рухомої руки на платформі Arduino UNO.

Об'єктом дослідження є прототип роботизованої рухомої руки.

Предметом дослідження є послідовний процес розробки та програмування роботизованої рухомої руки на базі мікроконтролерної платформи Arduino UNO.

Для реалізації поставленої мети потрібно вирішити наступні завдання:

1. Виконати огляд та аналіз предметної області використання роботів-маніпуляторів.
2. Сформулювати вимоги до системи керування роботом-маніпулятором.
3. Виконати підбір елементної бази робота-маніпулятора.
4. Сформулювати технічне завдання на розробку пристрою, розробити схему

пристрою.

5. Здійснити розробку апаратного та програмного рішення відповідно до технічного завдання та схеми.

6. Розробити інструкцію користувача з експлуатації пристрою.

Новизна даної роботи полягає у застосуванні новітніх підходів до розробки апаратного та програмного забезпечення із застосуванням сучасних технологій та засобів.

РОЗДІЛ 1

АНАЛІТИЧНА ЧАСТИНА

1.1 Огляд та аналіз області використання роботів-маніпуляторів

Сьогодні в багатьох галузях різних сферах, таких як виробництво, медицина, наука та інженерія використовуються роботи-маніпулятори, що можуть виконувати повторювані та рутинні завдання, а також працювати в умовах, що небезпечні для людей.

Робот-маніпулятор – це автоматизований пристрій, який здатен виконувати маніпуляційні операції з різноманітними об'єктами [1]. Зазвичай робот-маніпулятор складається з механічної руки з гнучкими зв'язками, спеціального затисного пристрою, що забезпечує захоплення та утримання розноманітних об'єктів та комп'ютерної системи керування.

В промисловості роботи-маніпулятори використовуються для автоматизації виробничих процесів, таких як збірка, пакування, монтаж, фарбування та інші. Робот-маніпулятор може виконувати складні операції, що забезпечує збільшення продуктивності і якості продукту, а також зменшення витрат на працю [1]. Вони здатні здійснювати рутинні та монотонні дії з високою точністю та швидкістю, що знижує ризик виникнення помилок та забезпечує ефективність виробництва. Такі роботи-маніпулятори використовуються в автомобілебудуванні, електронній та металургійній промисловості, у виробництві харчових продуктів, фармацевтичній та текстильній галузях та багатьох інших.

Крім того, роботи-маніпулятори знаходять своє застосування в медицині, де вони можуть виконувати різноманітні операції з високою точністю та мінімальними травмами для пацієнта [1]. Також, роботи-маніпулятори можуть бути використані для виконання робіт в умовах, які небезпечні для життя людей, наприклад, в радіаційному середовищі, на глибині моря або в космосі.

Існує кілька видів роботів-маніпуляторів залежно від їх конструкції та сфери застосування:

– промислові роботи-маніпулятори (рис. 1.1). Ці роботи зазвичай мають багатоосеву конструкцію та використовуються в промисловості для автоматизації виробничих процесів [2]. Вони можуть мати різні форми та розміри, але зазвичай мають здатність до швидких та точних рухів;



Рисунок 1.1 – Промисловий робот-маніпулятор

– сервісні роботи-маніпулятори (рис. 1.2). Ці роботи зазвичай мають меншу конструкцію та використовуються для обслуговування різних об'єктів, таких як автомобілі, літаки, медичні прилади тощо [2]. Вони можуть мати різні види захоплюючих пристроїв, що дозволяє їм виконувати різноманітні завдання;

– роботи-маніпулятори для досліджень. Ці роботи зазвичай мають більш складну конструкцію та використовуються для наукових досліджень в областях, таких як астрономія, біологія, медицина та інші [1]. Вони можуть мати різні захоплюючі пристрої та сенсори, що дозволяє їм збирати різноманітні дані та виконувати складні дії;

– медичні роботи-маніпулятори: ці роботи призначені для використання в медичній галузі, наприклад, для проведення хірургічних операцій [2]. Вони можуть бути оснащені різноманітними інструментами та датчиками, що дозволяє забезпечити високу точність та контроль за операцією;



Рисунок 1.2 – Сервісний робот-маніпулятор

– роботи-асистенти: ці роботи-маніпулятори призначені для підтримки людей у виконанні різних завдань. Наприклад, вони можуть допомагати людям з обмеженими можливостями в повсякденному житті або допомагати працівникам у виконанні рутинних робіт [2];

– роботи-маніпулятори для освіти: ці роботи призначені для використання в навчальних закладах [3]. Вони можуть використовуватися для навчання студентів різноманітним навичкам, таким як програмування, механіка, електроніка та інше;

– службові роботи-маніпулятори: ці роботи призначені для використання в різних службах, наприклад, у поліції для демінування, у пожежній службі для боротьби з пожежами [3];

– мікророботи: роботи-маніпулятори, що зазвичай мають розмір до 1 сантиметра і використовуються для проведення мікрооперацій, таких як монтаж електронних компонентів на плати або виробництво мікročіпів. Ці роботи зазвичай мають високу точність та здатність працювати в умовах обмеженого простору, де недоступна ручна праця [3]. Деякі мікророботи можуть бути безпроводними та здатними до автономної роботи, використовуючи вбудовану електроніку та сенсори;

– роботи-садівники – це роботи-маніпулятори, що використовуються для сівби, поливу та збирання різних рослин на великих фермах. Вони можуть бути програмовані для виконання різних завдань, таких як розсіювання насіння, обрізання гілок та збирання врожаю [1];

– роботи-кухарі – це роботи-маніпулятори, які розроблені для автоматизації процесу приготування їжі. Вони можуть виконувати різноманітні завдання, такі як нарізання овочів, змішування складових, виконання етапів приготування страв та інші, що допомагає зменшити витрати на оплату праці та зменшити час приготування страв.

Перші роботи-маніпулятори з'явилися у 1960-х роках в США. Вони були оснащені двопальцевим захоплювачем з п'ятьма ступенями свободи, який працював на пневмо- та гідроприводі. Ці роботи могли переміщувати деталі масою до 12 кг з точністю до 1,25 мм та запам'ятовувати координати свого маршруту. Такі промислові роботи переважно складалася з механічних компонентів (75%) та електроніки (25%) [1]. Переналагодження роботів для нових завдань вимагало часу та простою устаткування, а для їх перепрофілювання потрібна була повна зміна програми керування.

У другому поколінні роботів-маніпуляторів передбачалось більш точне керування, а самі роботи потребували середовища, яке було відповідно підготовлене для їхньої роботи. Цей етап характеризувався розробкою великої кількості датчиків, що забезпечували так звану «чутливість» роботів до зовнішнього середовища. Завдяки цьому роботи отримували інформацію про навколишнє середовище і здійснювали оптимальний вибір дій, узгоджених з навколишнім середовищем. Наприклад, робот міг самостійно оминати перешкоди, беручи деталь за допомогою маніпулятора [1]. Це стало можливим завдяки мікропроцесорній обробці інформації, отриманої від датчиків. Операції, виконані роботом, були адаптивними, тобто вони підлаштовувалися до конкретних умов виконання завдання з метою поліпшення їх якості.

Метод адаптивного керування, зазвичай, реалізується програмно та ґрунтується на багатоваріантному програмному забезпеченні. При цьому, сам

робот приймає рішення щодо вибору типу роботи програми на підставі інформації про оточуюче середовище, яку отримує від датчиків.

Роботи другого покоління характеризувалися попередньо встановленими режимами роботи, які були активовані автоматично при отриманні певних показників з зовнішнього середовища.

Третє покоління роботів-маніпуляторів має здатність до автоматичного створення програми дій відповідно до поставленого завдання та умов зовнішнього середовища. Вони не потребують заздалегідь заданої послідовності технологічних дій для різних варіантів зовнішнього середовища [1]. Ці роботи можуть самостійно вибирати оптимальний алгоритм роботи та оперативно його реалізовувати. Електроніка таких промислових роботів коштує в десятки разів більше, ніж механічна частина.

Сучасний робот-маніпулятор здатен здійснювати захоплення об'єктів за допомогою сенсорів та оцінювати ефективність цієї дії. Крім того, такий робот може самостійно регулювати силу захоплення, використовуючи зворотний зв'язок по зусиллю, залежно від крихкості матеріалу об'єкта [2]. Це робить його інтелектуальним, а система керування – «мозком» такого приладу. Регулювання може бути здійснене за допомогою методів штучного інтелекту, що дає найбільші перспективи для цих машин. Інтелект таких роботів визначається пакетами прикладних програм, програмованими логічними контролерами та інструментами моделювання.

Виконавчий механізм робота-маніпулятора – це складний просторовий механізм, що складається з кількох зв'язаних ланок. У загальному випадку він може мати різні типи кінематичних пар – поступальні, обертальні, циліндричні, сферичні та інші [2]. Кінематична пара – це з'єднання двох елементів механізму, які дозволяють виконувати певний рух.

Кількість ланок та типів кінематичних пар в виконавчому механізмі залежить від поставленої задачі. Для відтворення просторового руху захоплювача в загальному випадку робот-маніпулятор повинен мати шість ступенів свободи, тобто можливість виконувати рухи вздовж шести різних координат. Якщо ж

потрібно відтворювати просторову траєкторію лише однієї точки захвату, то необхідне число ступенів свободи може бути менше.

Надлишкові ступені свободи – це додаткові ступені свободи, які необхідні для реалізації певних рухів, але не є обов'язковими для виконання поставленої задачі. Вони дають змогу оптимізувати кінематичні, динамічні, енергетичні та інші критерії якості процесу маніпулювання [4]. Збільшення числа ступенів маневреності робота-маніпулятора розширює його можливості при виконанні складних рухів, збільшує робочий простір, зменшує мертві зони та розширює варіантність вибору траєкторій рухів в обмежених умовах.

До основних характеристик роботів-маніпуляторів відносяться кількість ступенів свободи, максимальна вантажопідйомність та робоча область.

Робоча область маніпулятора описує область, в якій виконавчий пристрій може рухатись під час функціонування. Розмір та конфігурація робочого простору залежать від кількості ступенів свободи маніпулятора, розташування та типу кінематичних пар, а також розмірів ланок [5]. Рух руки маніпулятора може бути описаний у прямокутній, циліндричній або сферичній системах координат. У прямокутній системі координат можна забезпечити переміщення руки маніпулятора лише за допомогою поступальних пар, тоді як маніпулятор з оберतालними рухами ланок може переміщувати об'єкт маніпулювання в об'ємно-сферичному робочому просторі. Маніпулятори з двома поступальними та одним обертальним рухом ланок мають широкі можливості та здатні маніпулювати об'єктом в об'ємно-циліндричному робочому просторі. Також поширені маніпулятори, що мають структуру з двома обертальними та одним поступальним рухом ланок, які дають можливість маніпулювати у значно більшій об'ємно-сферичній зоні.

Не всі частини робочого простору є однаково зручними для виконання заданих рухів захвату. Щоб уникнути проблем, рух захоплювача поділяють на чотири класи залежно від умов:

– рух захоплювача з вільним об'єктом маніпулювання у вільному робочому просторі;

- рух захоплювача з вільним об'єктом у невідомому робочому просторі з перешкодами у вигляді нерухомих об'єктів;
- рух захоплювача у відомому робочому просторі з об'єктом маніпулювання, на який накладено обмеження;
- рух захоплювача у невідомому робочому просторі з невідомим об'єктом маніпулювання.

Зона обслуговування маніпулятора визначає простір, в якому робот може виконувати свої функції відповідно до призначення та встановлених характеристик.

1.2 Аналіз аналогічних продуктів

На ринку в наш час представлена велика кількість вже готових рішень, що пропонують широкий спектр можливостей, залежно від виконуваних завдань. Можна виділити також цілу лінійку роботів-маніпуляторів, орієнтованих на освітній сегмент.

Робот-маніпулятор ReactorX 150 від фірми Trossen Robotics (рис. 1.3) має п'ять ступенів свободи, максимальну вантажопідйомність 100 грамів та робочою областю 900 мм.



Рисунок 1.3 – Робот-маніпулятор ReactorX 150

Маніпулятор може керуватися наперед визначеною програмою або за допомогою джойстика в режимі реального часу. Також присутня можливість використання синхронного керування одночасно двома маніпуляторами.

Робот використовує високоточні серводвигуни з плавним ходом, що забезпечує максимально точні рухи, однак це значно підвищує його вартість.

Робот-маніпулятор 4 DOF (рис. 1.4) на базі Arduino UNO має 4 ступені свободи та максимальну робочу область 600 мм.

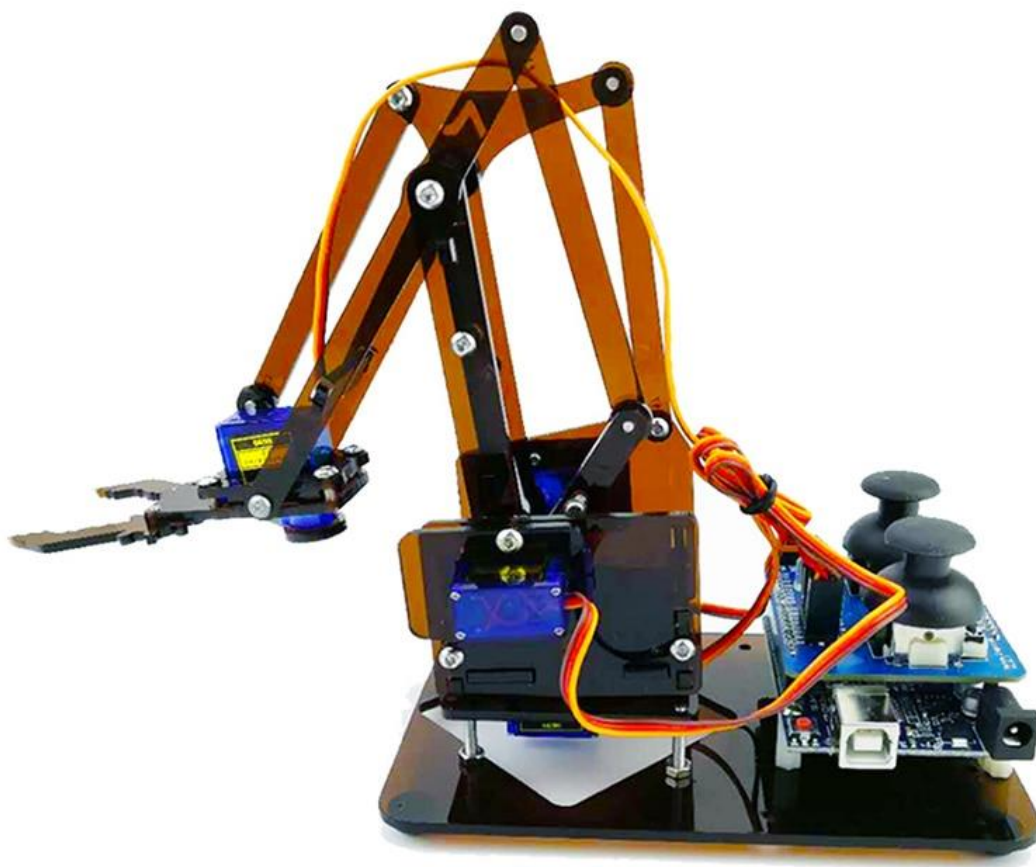


Рисунок 1.4 – Робот-маніпулятор 4 DOF

Даний робот є одним з найдешевших рішень на ринку, однак має ряд недоліків: керування лише за допомогою джойстиків, дешеві серводвигуни, що робить його рухи доволі різкими та неточними, невелика вантажопідйомність. Однак використання в якості плати керування Arduino UNO дає значні можливості по розширенню функціональності робота та дозволяє використовувати маніпулятор для вивчення програмування.

Робот-маніпулятор Lego EV3 із серії Lego Mindstorms (рис. 1.5) має п'ять ступенів свободи та, на відміну від попередніх аналогів, може бути мобільним.

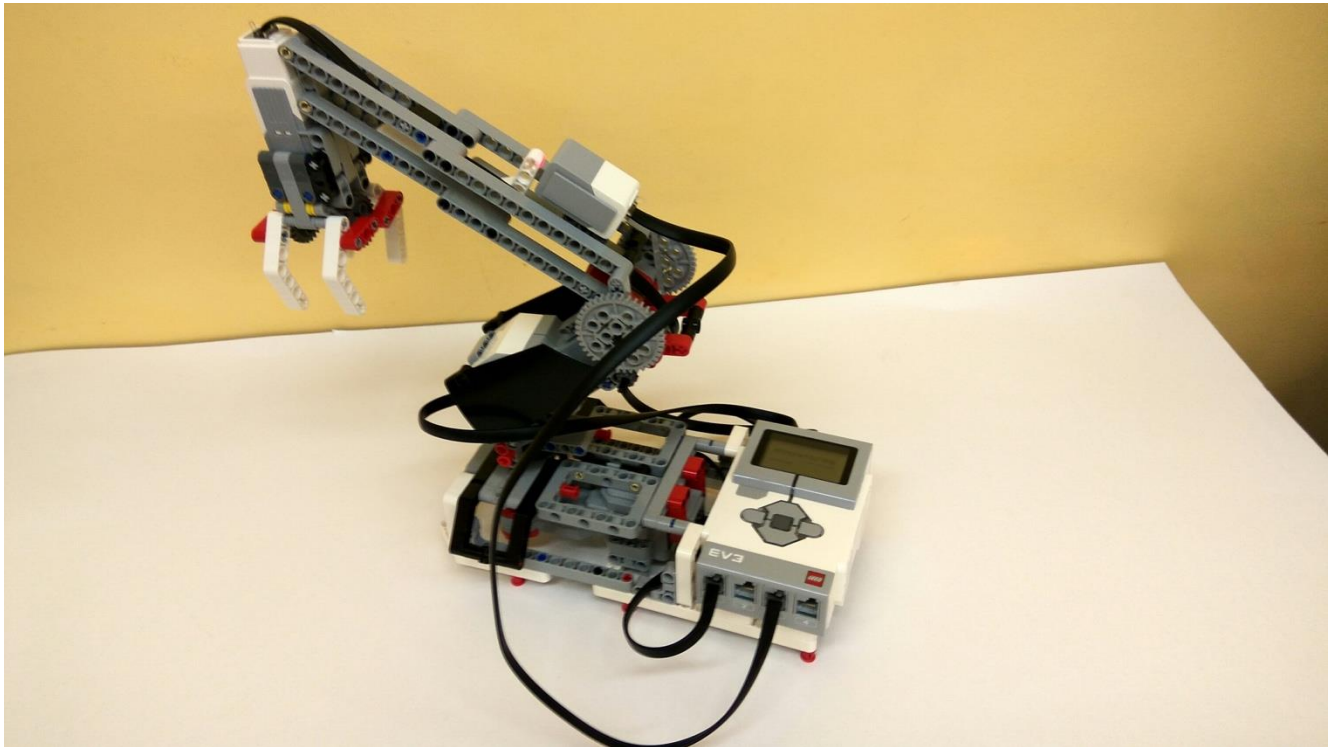


Рисунок 1.5 – Робот-маніпулятор Lego EV3

Завдяки тому, що робот-маніпулятор входить в лінійку Lego Mindstorms – існують широкі можливості по модифікації даного робота, використовуючи інші компоненти. Керування здійснюється за наперед заданою програмою. До недоліків можна віднести доволі різкі рухи маніпулятора, певну ненадійність загальної конструкції (за рахунок використання великої кількості блоків, що можуть роз'єднатися від вібрації) та доволі висока вартість.

Робот-маніпулятор Dobot Magician Basic (рис. 1.6) має чотири ступені свободи має в комплекті набір додаткових аксесуарів, що можуть розширювати функціонал, в тому числі тип захоплюючого механізму.

Керування даним маніпулятором може здійснюватися за допомогою наперед створеної програми (в тому числі, створеної методом повторення рухів за користувачем), мобільного додатку через Wi-Fi або Bluetooth, а також з використанням комп'ютерної мишки.

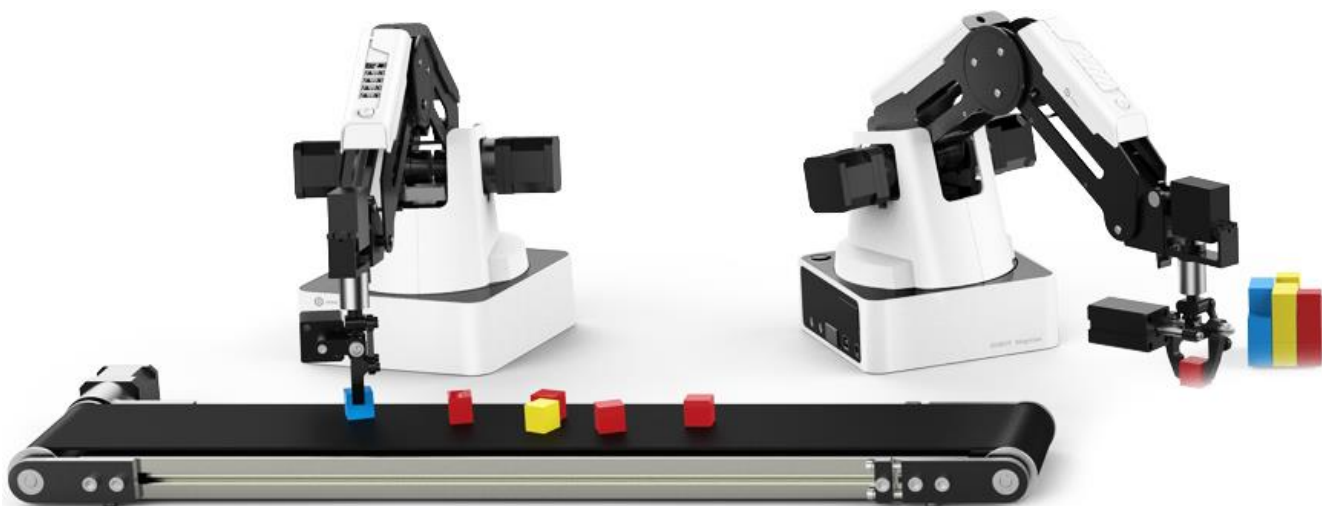


Рисунок 1.6 – Робот-маніпулятор Dobot Magician Basic

Отже, при розробці проекту роботизованої рухомої руки варто врахувати всі основні переваги та недоліки аналогічних рішень, забезпечити не менше п'яти ступенів свободи, використання якісних серводивигунів, а також додати можливості різних типів керування маніпуляторів.

1.3 Вимоги до системи керування роботом-маніпулятором

Система керування роботом-маніпулятором повинна задовольняти ряд вимог для ефективної та безпечної роботи [2]. Основні вимоги до системи керування роботом-маніпулятором включають наступне:

- надійність: система керування повинна бути надійною та стійкою до впливу зовнішніх факторів.

- швидкість реакції: система керування повинна забезпечувати швидку реакцію на зміни в середовищі та відповідну швидкість виконання рухів робота.

- точність та повторюваність: система керування повинна забезпечувати точність та повторюваність рухів робота, що є важливим для виконання завдань з високою точністю.

- гнучкість: система керування повинна бути гнучкою та легко налаштовуватись для виконання різноманітних завдань.

– безпека: система керування повинна забезпечувати безпеку взаємодії робота з оточуючим середовищем та людьми.

– простота використання: система керування повинна бути простою у використанні та мати зрозумілий інтерфейс для користувачів.

– сумісність: система керування повинна бути сумісною з різними типами роботів-маніпуляторів та іншими компонентами робототехнічної системи.

– наявність документації та підтримки: система керування повинна мати документацію та підтримку з боку виробника, щоб забезпечити користувачам можливість вирішення технічних питань та отримання необхідної допомоги.

Система керування роботом-маніпулятором має бути досить гнучкою, щоб забезпечувати точне позиціонування та керування рухом кожного з сегментів робота. Вона повинна мати можливість працювати з різними типами сенсорів, таких як енкодери, гіроскопи, акселерометри, камери, лазерні датчики та інші, для збору інформації про стан динаміки робота та його оточення [6].

Окрім того, система керування повинна мати можливість планувати траєкторії руху робота на основі заданих параметрів, таких як координати, швидкість, прискорення та обмеження на моменти великої сили [6]. Вона також повинна забезпечувати керування роботом у режимі реального часу та забезпечувати безпеку під час роботи з ним.

Для програмування системи керування зазвичай використовують мови програмування, такі як C++, Python, MATLAB, а також спеціалізовані бібліотеки та фреймворки, такі як ROS, OpenCV, TensorFlow, PyTorch тощо [7]. Важливо також забезпечити відповідну апаратну базу для системи керування, яка забезпечує швидку та ефективну обробку даних та виконання розрахунків.

При використанні керування зі смартфона або комп'ютера важливим є інтерфейс користувача: система керування має мати зручний та інтуїтивно зрозумілий інтерфейс. Це може включати графічний інтерфейс користувача, який дозволяє встановлювати та налаштовувати параметри робота, програмувати його, а також переглядати результати виконання завдань.

Можна скористатися однією з існуючих на даний час універсальних систем керування: ROS, MATLAB Robotics System Toolbox, LabVIEW Robotics Module, OpenCV.

ROS (Robot Operating System) – це відкрите програмне забезпечення для розробки програмного забезпечення для роботів [6]. ROS забезпечує бібліотеки та інструменти для створення, взаємодії, управління, візуалізації та тестування роботів. ROS може бути використаний для розробки різноманітних видів роботів, таких як маніпулятори, мобільні роботи та багато інших.

ROS складається з двох основних частин: різноманітні бібліотеки, що забезпечують функціональність, такі як взаємодія з апаратним забезпеченням, роботи з датчиками та елементами керування, інструменти, що забезпечують розробникам засоби для створення роботів, такі як система керування пакетами та інші інструменти.

ROS використовується в широкому спектрі дослідницьких проєктів та промислових застосуваннях, включаючи медицину, автомобільну промисловість, науку та інженерію. Він дозволяє розробникам швидко та ефективно розробляти програмне забезпечення для роботів, що можуть працювати в різних середовищах та виконувати різноманітні завдання.

MATLAB Robotics System Toolbox – це набір інструментів, що дозволяє розробникам та інженерам розробляти алгоритми для робототехнічних застосувань. Вона надає функції для моделювання та симуляції роботів, проектування та візуалізації кінематики та динаміки руху, планування траєкторій, керування рухом робота та багато іншого [7].

MATLAB Robotics System Toolbox забезпечує інтеграцію з популярними робототехнічними платформами та фреймворками, такими як ROS, URDF, Simulink, Gazebo та інші. Вона підтримує понад 100 видів роботів від провідних виробників, включаючи ABB, KUKA, Fanuc, Universal Robots та інші.

Завдяки своїм можливостям, MATLAB Robotics System Toolbox є потужним інструментом для розробки та тестування роботів у багатьох галузях, включаючи виробництво, медицину, науку та освіту.

LabVIEW Robotics Module – це розширення до програмної платформи LabVIEW, яке надає можливості для розробки і керування роботами-маніпуляторами [7]. Цей модуль містить набір інструментів для робототехніки, включаючи інтерфейси для підключення до різних типів роботів, програмування рухів, симуляцію руху та збір даних з різних датчиків.

З використанням LabVIEW Robotics Module можна розробляти програмне забезпечення для різних видів роботів, включаючи маніпулятори, мобільні роботи та інші. Цей модуль надає можливість розробляти програми в зручному візуальному інтерфейсі, що спрощує процес програмування для користувачів без досвіду в робототехніці.

Додатково, LabVIEW Robotics Module надає можливість підключатися до інших платформ керування роботами, таких як ROS (Robot Operating System) та National Instruments Robotics Starter Kit. Це робить цей модуль більш універсальним та зручним для використання в різних проектах з робототехніки.

OpenCV – це відкрите програмне забезпечення для комп'ютерного зору [6]. Це дозволяє розробникам створювати програми для визначення об'єктів на зображеннях та відео, що може бути корисним для систем керування роботами-маніпуляторами. OpenCV може бути використано для розпізнавання обличчя, рухів та інших визначених об'єктів.

РОЗДІЛ 2

ТЕОРЕТИЧНА ЧАСТИНА

2.1 Загальний опис розроблюваного пристрою

Основне призначення роботизованої руки – це здійснення різноманітних маніпуляцій з об'єктами. Роботизовані руки можуть здійснювати різноманітні операції, такі як підйом, переміщення, поворот, захоплення та відкладання об'єктів. Завдяки точній та повторюваній роботі роботизованих рук, вони можуть забезпечити значну ефективність роботи, а також допомагають знизити ризики пошкодження для людей [2].

Основні характеристики розроблюваної роботизованої руки:

- кількість ступенів свободи – 5-6;
- вантажопідйомність – не менше 50 г;
- робоча область – 600-900 мм;
- використання кількох різних типів керування (джойстик, смартфон);
- надійні сервоприводи з плавними і точними рухами;
- якісні матеріали конструкції;
- використання платформи Arduino.

Важливо реалізувати проект, забезпечивши при цьому оптимальне співвідношення між якістю та надійністю комплектуючих, функціоналом та вартістю. Також потрібно забезпечити мобільність робота-маніпулятора, надавши йому можливість додатково розширити зону впливу робота [2].

Даний продукт розробляється для широкого загалу, яким зможе користуватися навіть непідготовлений користувач. Для цього, в першу чергу, потрібно забезпечити зручну та інтуїтивно зрозумілу систему керування зі зручним інтерфейсом користувача – апаратним або програмним.

Також важливо зробити проект ремонтпридатним та легконалаштовуваним, щоб дати можливість користувачу освоїти також інженерні навички (за потреби).

2.2 Обґрунтування вибору апаратної мікроконтролерної платформи

Вибір апаратної мікроконтролерної платформи є одним з важливих етапів у проектуванні будь-якої автоматизованої системи керування.

Мікроконтролер – це спеціалізована мікропроцесорна система у формі мікросхеми, яка містить мікропроцесор, блоки пам'яті для збереження програмного коду та даних, порти введення-виведення та блоки зі спеціальними функціями, такими як лічильники, компаратори, перетворювачі та інші [8].

Мікроконтролери використовуються для керування електронними пристроями та можуть бути розглянуті як самостійні однокристальні комп'ютери, що здатні виконувати прості завдання. Використання мікроконтролерів дозволяє значно зменшити розміри, енергоспоживання та вартість пристроїв, що побудовані на їх базі.

Мікроконтролери застосовуються в різноманітних сучасних пристроях, включаючи телефони, пральні машини, а також відповідають за роботу двигунів і систем гальмування у сучасних автомобілях [9]. З їх допомогою розробляються системи контролю та збору інформації. Мікроконтролери є основною складовою процесорів, що випускаються у всьому світі, завдяки їхнім компактним розмірам, енергоефективності та доступній ціні.

Основні варіанти базових платформ для проектування можуть включати:

- мікроконтролери Atmel AVR;
- платформу Arduino;
- платформу STM328;
- платформу Raspberry Pi.

AVR – сімейство восьмибітних мікроконтролерів фірми Atmel.

Архітектура AVR базується на гарвардській архітектурі, що передбачає розміщення програм і даних в різних адресних просторах [4]. Процесори AVR володіють системою команд, що наближена до ідеології RISC, і містять 32 регістри загального призначення ширини 8 біт. Максимальна тактова частота AVR становить 20 мегагерц, а короткі команди можуть виконуватися за один такт.

Набільш поширене сімейство AVR – ATmega (рис. 2.1):

- флеш-пам'ять до 256 Кб; SRAM до 16 Кб; EEPROM до 4 Кб;
- число портів вводу-виводу 23-86;
- апаратний множник;
- розширена система команд і периферійних пристроїв.

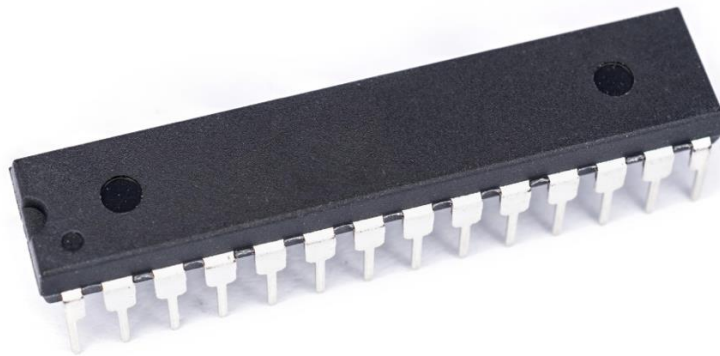


Рисунок 2.1 – Мікроконтролер ATmega8

Arduino є набором апаратно-програмних засобів для створення простих систем автоматизації та робототехніки, який спрямований на непрофесійних користувачів [10].

Програмне забезпечення для написання програм, компіляції та програмування апаратури є складовою програмної частини, яка надається у вигляді безкоштовного інтегрованого середовища розробки (IDE). Апаратна частина складається з набору змонтованих друкованих плат, доступних для придбання як від офіційного виробника, так і від сторонніх виробників [10]. Архітектура системи є повністю відкритою, що дозволяє вільно копіювати або доповнювати лінійку продуктів Arduino (рис. 2.2).

Плати Arduino можна використовувати для створення автономних систем автоматизації, а також для підключення до комп'ютерних програм через різноманітні провідні та безпроводні інтерфейси.

В класичній лінійці пристроїв Arduino в основному застосовуються мікроконтролери Atmel AVR: ATmega2560, ATmega32U4, ATmega328.

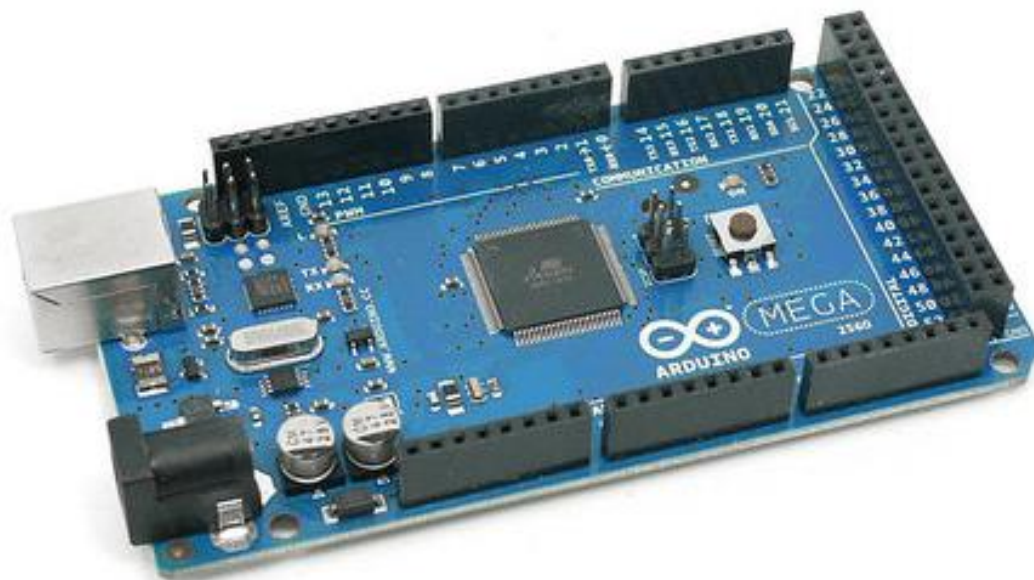


Рисунок 2.2 – Плата Arduino Mega 2560 R3

STM32 – це серія 32-бітних мікроконтролерів виробництва компанії STMicroelectronics (рис. 2.3). В рамках кожної серії використовується одне і те ж 32-бітне ядро ARM, таке як Cortex-M7F, Cortex-M4F, Cortex-M3, Cortex-M0+ або Cortex-M0 [8]. Кожен мікроконтролер містить ядро процесора, статичну RAM-пам'ять, флеш-пам'ять, інтерфейс відналагодження та інші периферійні інтерфейси.

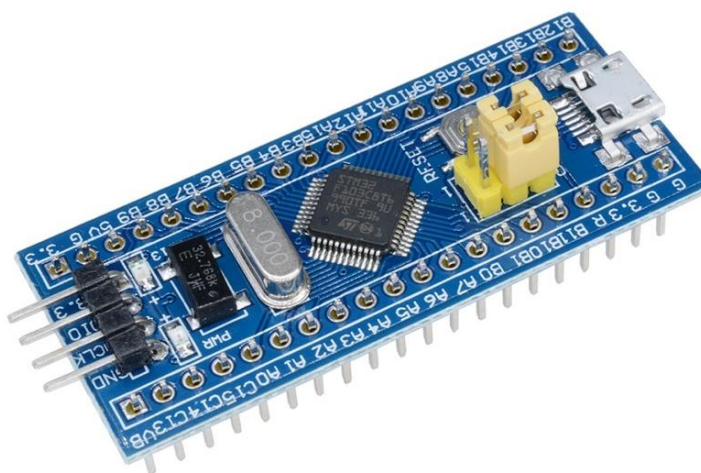


Рисунок 2.3 – Плата STM32F103C8T6

Raspberry Pi є одноплатним комп'ютером, розміром приблизно з банківську карту (рис. 2.4). Початково створений як бюджетна система для навчання інформатиці, Raspberry Pi отримав широке застосування в подальшому [11].



Рисунок 2.4 – Одноплатний комп'ютер Raspberry Pi 3 Model B

Мікрокомп'ютер Raspberry Pi має компактний розмір, збирається на чотирьохшаровій друкованій платі, яка має розміри банківської карти (за винятком моделей Zero і Zero W). У стандартний комплект поставки входить лише плата, корпус, блок живлення та флеш-карту необхідно замовляти окремо.

Використання мікроконтролерів у «чистому» вигляді ускладнюється за раунок наступних причин [4]:

- складність підключення периферії;
- складність налаштування та програмування;
- використання спеціального програматора;
- відсутність готових модулів.

Даних проблем позбавлені апаратні платформи.

Використання платформи Raspberry Pi є недоцільним вибором через її високу вартість та складність програмування, особливо через те, що ця платформа має операційну систему [11].

Порівняльний аналіз характеристик найбільш використовуваних апаратних платформ Arduino UNO та STM32F103C8T6 наведено в таблиці 2.1.

З таблиці видно, що хоча платформа Arduino UNO має меншу продуктивність (щоправда, цілком достатню для нашого проекту) і трохи вищу вартість плати, але вона має ряд переваг, що робить її оптимальним вибором для побудови автоматизованої системи керування:

- можливість завантаження коду через USB напряму із програмного середовища;
- наявність вбудованої енергонезалежної EEPROM-пам'яті для зберігання даних;
- розвинута екосистема, що включає велику кількість бібліотек, модулів та плат розширення;
- достатня кількість документації на українській мові;
- простота програмування;
- наявність системи відналадження в самому середовищі програмування.

Таблиця 2.1 – Порівняльний аналіз платформ Arduino UNO та STM32

Характеристика	STM32F103C8T6	Arduino UNO
Тактова частота, МГц	72	16
Пам'ять програм, Кб	64	32
Живлення, В	3.3	5
ОЗП, Кб	20	2
Підключення по USB	+	+
Прошивка через USB	–	+
Вбудована EEPROM-пам'ять	–	+
Вартість, грн	244	324







2.3 Середовище програмування Arduino IDE

Для програмування під платформу Arduino використовується стандартне середовище розробки зі своєю мовою програмування, яку можна легко освоїти навіть тим, хто тільки починає займатися програмуванням [20].

Мова програмування Arduino базується на основі стандартної мови C++ та використовує компілятор AVR-GCC. Хоча вона має кілька особливостей, які полегшують написання програм для початківців.

Код, написаний для середовища Arduino має назву скетч і може бути написаний у звичайному текстовому редакторі. Однак середовище Arduino IDE дозволяє додатково отримати пояснення та повідомлення про можливі помилки, що можуть виникнути під час написання коду [12]. У вікні виведення тексту (консолі) можна бачити повідомлення від Arduino, включаючи повні звіти про помилки та іншу корисну інформацію. Кнопки на панелі інструментів дозволяють

перевірити та записати програму в мікроконтролер, створити, відкрити та зберегти скетч, а також відкрити монітор послідовного порту (для обміну даними між платою Arduino та комп'ютером):

-  – перевірити скетч на помилки;
-  – скомпілювати код за вивантажити його на мікроконтролер;
-  – створити новий файл (скетч);
-  – відкрити вже існуючий файл (скетч);
-  – зберегти файл (скетч);
-  – відкрити монітор послідовного порту.

Найпростіша Arduino-програма складається з двох функцій:

- `setup()` – функція, що викликається одноразово при старті мікроконтролера.
- `loop()` – функція, що викликається після `setup()` в нескінченному циклі увесь час роботи мікроконтролера.

Для налаштування середовища розробки Arduino IDE для конкретної платформи Arduino необхідно вказати назву моделі Arduino та номер COM-порту, який їй присвоєно [12].

Під час завантаження скетчу використовується спеціальна програма, відома як завантажувач (Bootloader) Arduino. Це невеликий код, який завантажується в мікроконтролер, що знаходиться на платі Arduino. Завантажувач дозволяє завантажувати програмний код без необхідності використовувати додаткове апаратне забезпечення (програмактор).

Завантажувач активується протягом короткого часу під час перезавантаження платформи або при завантаженні будь-якого скетчу в мікроконтролер. Його роботу можна розпізнати за миганням світлодіода, який знаходиться на 13-му піні плати Arduino [12].

При вивченні середовища Arduino IDE з часом може знадобитися використання додаткових бібліотек для розширення можливостей.

Бібліотека – це набір функцій, створених для спрощення роботи з різними компонентами, такими як датчики, дисплеї, модулі та інші. Наприклад, бібліотека `LiquidCrystal`, яка входить до стандартного набору, дозволяє досить легко

працювати з символьними LCD-дисплеями [10]. Крім цього, існує велика кількість додаткових бібліотек, які можна завантажити з Інтернету. Проте перед тим, як почати використовувати додаткові бібліотеки, необхідно їх спочатку встановити.

Зазвичай бібліотеки доступні у вигляді ZIP-архіву або просто папки з назвою, яка збігається з назвою бібліотеки. У папці міститься файл з розширенням .cpp, файл з розширенням .h, а також keywords.txt - текстовий файл, папка examples з прикладами та інші файли, необхідні для роботи бібліотеки. Починаючи з версії Arduino IDE 1.0.5, сторонні бібліотеки можна встановити прямо в середовищі розробки без необхідності розпакування архіву.

РОЗДІЛ 3

РОЗРОБКА РОБОТИЗОВАНОЇ РУХОМОЇ РУКИ

3.1 Визначення загальної конструкції пристрою

Перше, що важливо зробити при розробці робота-маніпулятора – визначитися із загальною конструкцією пристрою, а саме [13][2]:

- кількість ланок рухомої руки;
- кількість ступенів вільності механізму;
- типи кінематичних пар на вузлах основних ланок;
- типи механізмів для реалізації кінематичних пар.

На рисунку 3.1 показана загальна кінематична схема проектованого пристрою.

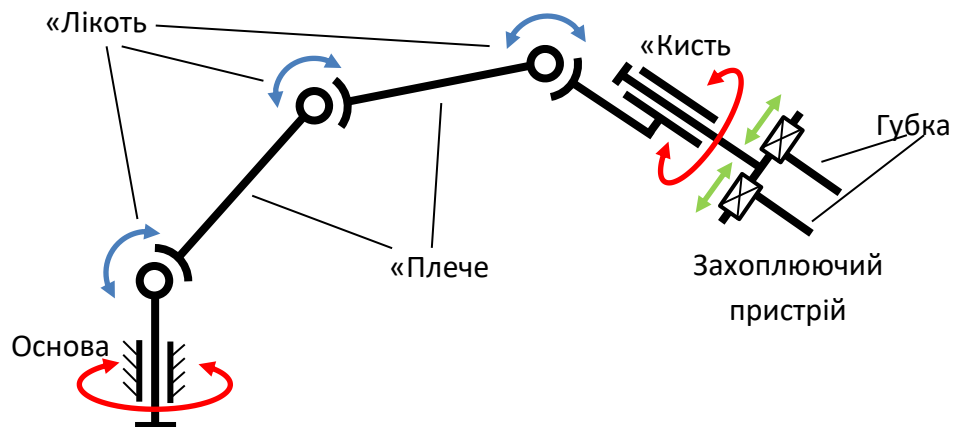


Рисунок 3.1 – Кінематична схема роботизованої руки

Конструкція маніпулятора передбачає 5 основних ланок та пристрій захоплення. В загальному дана конструкція (рис. 3.1) передбачає 5 ступенів вільності. Переважна більшість кінематичних пар проектованої роботизованої руки будуть обертальними (основа механізму та «кисть» руки повинні мати можливість здійснювати повний оберт, а інші – на 180 – 270 градусів). Кінематичні пари губок пристрою захоплення матимуть поступальний рух [2].

Для реалізації повного обертового руху будуть використовуватися крокові двигуни – вони забезпечать точні та плавні рухи без обмежень по кількості обертів. Інші обертові рухи можна реалізувати з використанням сервоприводів, що є

більш дешевими та простими в керуванні, однак яких цілком достатньо для реалізації поставлених цілей.

Поступальні рухи губок захоплюючого пристрою маніпулятора будуть реалізовані за допомогою направляючої та двох простих серводвигунів. Це дасть можливість більш надійніше фіксувати утримуваний об'єкт та регулювати силу стиснення, на відміну від більш поширеного варіанту реалізації типу «ножиці» [2].

3.2 Підбір елементної бази

3.2.1 Мікроконтролерна платформа

В наш час Arduino пропонує два великих сімейства мікроконтролерних плат: Arduino MKR та Arduino Classic.

Arduino MKR – це сімейство програмованих плат на базі процесорів ARM Cortex M0 SAMD21 з низьким енергоспоживанням та забезпечені радіомодулем для безпроводного зв'язку. Часто використовуються для реалізації систем Інтернету речей та проектів, що потребують підвищеної продуктивності [12].

Arduino Classic – сімейство, що базується переважно на мікроконтролерах серії ATmega фірми Atmel (Microchip). Вони є доступним та зручним рішенням при побудові як простих проектів (взаємодія кількох датчиків та виконавчих механізмів) так і доволі складних систем (робототехніка, 3D-принтери та ЧПК-стандки, дрони тощо).

Для реалізації проекту було обрано плату Arduino UNO R3 (рис. 3.2) як найбільш поширеного та доступного представника сімейства [12].



Рисунок 3.2 – Плата Arduino UNO R3

Arduino UNO побудована на базі мікроконтролера ATmega328 (16 МГц), має 14 цифрових пінів (можуть працювати як входи так і виходи) та 6 аналогових входів (рис. 3.3), що вистачає для реалізації більшості проектів [12]. Крім цього, деякі піни Arduino можуть використовуватися для інших цілей:

– Послідовний інтерфейс: піни 0 (RX) та 1 (TX). Використовуються для отримання (RX) і передачі (TX) даних по послідовному інтерфейсу. Ці піни з'єднані з відповідними виходами мікросхеми ATmega8U2, яка виконує роль перетворювача USB-UART.

– Зовнішні переривання (2 та 3 піни).

– ШІМ (широтно-імпульсна модуляція): піни 3, 5, 6, 9, 10 і 11. За допомогою них є можливість виводити 8-бітні аналогові значення у формі ШІМ-сигналу.

– Інтерфейс I2C: вихід A4 (SDA) і вихід A5 (SCL).

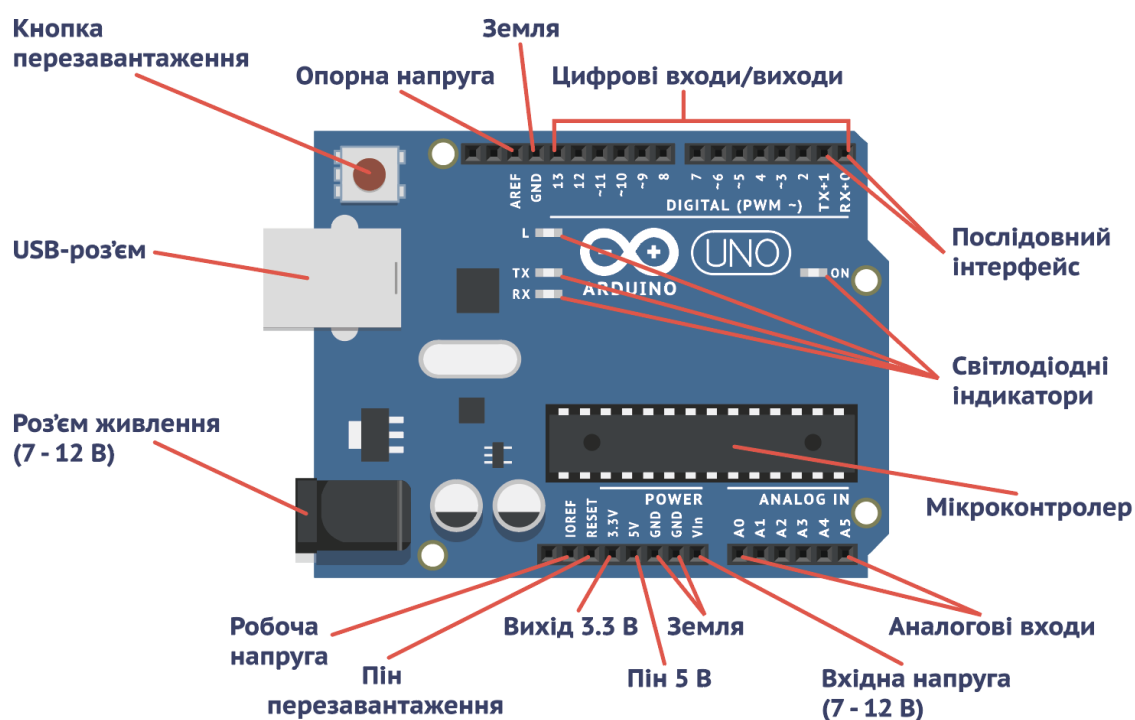


Рисунок 3.3 – Схема розміщення виходів Arduino UNO R3

Живлення плати може здійснюватися як за допомогою USB-порта (5В), так і через універсальний роз'єм живлення DC 5,5×2,1 (7 – 12 В).

Об'єм вбудованої в мікроконтролер флеш-пам'яті (для зберігання програми) становить 32 КБ (0,5 КБ використовуються завантажувачем). ATmega328 також

має 2 КБ пам'яті SRAM (оперативна пам'ять) і 1 КБ EEPROM (пам'ять для зберігання даних після вимкнення живлення контролера).

Ще однією з вагомих переваг використання Arduino UNO є можливість використання додаткових плат розширення, наприклад, для більш зручного підключення потрібної кількості сервоприводів чи крокових двигунів.

Крім основної плати потрібно також реалізувати пульт керування. Для його реалізації буде використано плату Arduino Pro Mini (рис. 3.4).

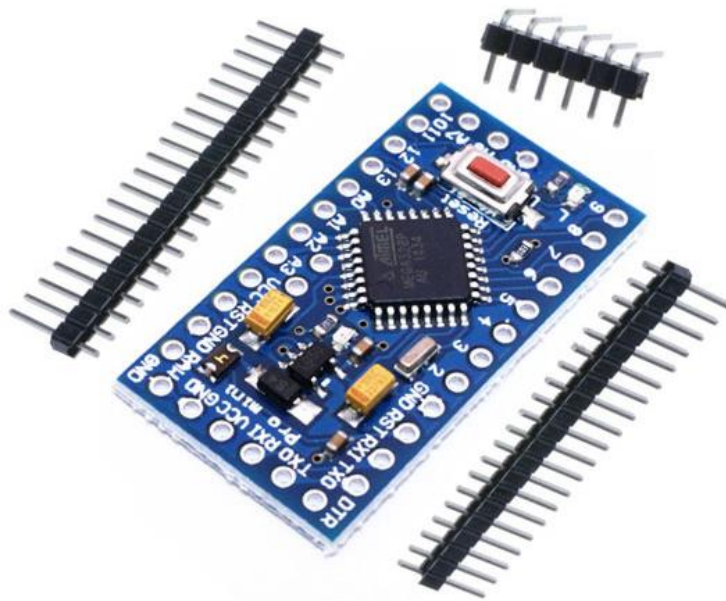


Рисунок 3.4 – Плата Arduino Pro Mini

Дана плата побудована на тому ж мікроконтролері (ATmega328), однак на відміну від Arduino UNO має більш компактні розміри. Разом з цим, тут відсутній стабілізатор напруги (живлення можливе лише від 5 В) та перетворювач інтерфейсів (для програмування плати потрібен USB-TTL перетворювач або програматор) [12].

Обмін інформацією між основною платою Arduino UNO та платою пульта керування Arduino Pro Mini буде здійснюватися за допомогою інтерфейсу I2C.

3.2.2 Сервоприводи та крокові двигуни

Для реалізації основних кінетичних пар роботизованої руки необхідно підібрати відповідні виконавчі механізми. Як вже зазначалося вище, будуть використовуватися два крокові двигуни та сервоприводи [11].

В якості робочого механізму для поворотної основи було обрано кроковий двигун NEMA17 (рис. 3.5).



Рисунок 3.5 – Кроковий двигун NEMA17

NEMA17 – це компактний малопотужний синхронний безщітковий кроковий електродвигун, який зазвичай використовується для побудови ЧПК-верстатів та 3D-принтерів [9].

Даний кроковий двигун має кут повороту $1,8^\circ$ на один крок, що дозволяє реалізувати висоточне позиціонування, а його потужності цілком достатньо для повороту всієї конструкції робота-маніпулятора.

Основні технічні характеристики NEMA17:

- кут повороту за 1 крок: $1,8^\circ$;
- діаметр валу: 5 мм;
- довжина валу: 20 мм;
- робочий струм: 1,33 А;
- робоча напруга: 12 В;
- крутний момент: 2,2 кг/см.

Кроковий двигун 28BYJ-48 (рис. 3.6) буде використовуватися для реалізації механізму кінетичної пари «кисті» роботизованої руки.

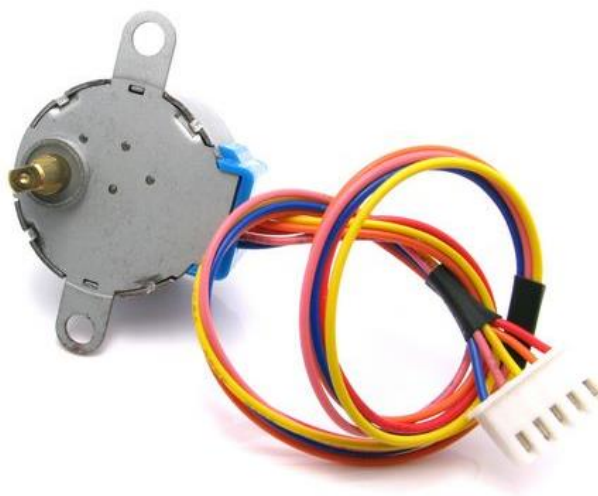


Рисунок 3.6 – Кроковий двигун 28BYJ-48

Даний кроковий двигун є найбільш поширеним серед аматорів, що працюють з робототехнікою, оскільки є недорогим та простим у використанні рішенням [9]. Він має дещо простіші характеристики ніж NEMA17, однак цього повністю вистачає для реалізації механізму «кисті» роботизованої руки.

Основні характеристики 28BYJ-48:

- кут повороту за 1 крок: $5,625^\circ$;
- діаметр валу: 5 мм;
- довжина валу: 7,5 мм;
- робочий струм: 0,32 А;
- робоча напруга: 5 В;
- крутний момент: 0,6 кг/см.

Враховуючи те, що крокові двигуни мають кілька робочих обмоток (зазвичай, дві), керування ними відбувається дещо складніше, ніж звичайними електродвигунами. У зв'язку з цим, для керування кроковими двигунами потрібно використовувати спеціальний драйвер [11].

Найкращим рішенням для крокового двигуна буде використання драйвера на базі чіпа A4988 (рис. 3.7).



Рисунок 3.7 – Драйвер A4988 для крокового двигуна

Даний драйвер забезпечує керування кроковим двигуном за допомогою всього двох сигнальних контактів та часто використовується в парі з NEMA17 для побудови 3D-принтерів та інших ЧПК-станків. Він дозволяє керувати двигунами з робочою напругою 8 – 35 В та струмом до 2 А.

Оскільки кроковий двигун 28BYJ-48 має робочу напругу живлення 5 В, для нього буде використовуватися інший драйвер – ULN2003 (рис. 3.8).

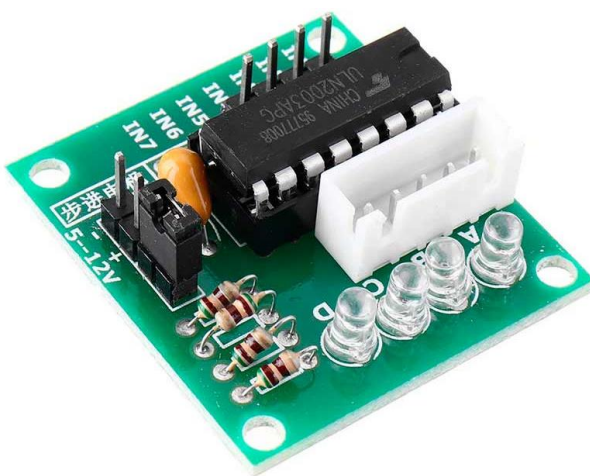


Рисунок 3.8 – Драйвер ULN2003 для крокового двигуна

Драйвер ULN2003 має відповідний інтерфейс підключення для крокового двигуна 28BYJ-48 та керується за допомогою мікроконтролера за допомогою 4-х контактів. Також даний драйвер сумісний зі стандартною бібліотекою Arduino Stepper [11].

Для реалізації обертальних рухів «ліктів» рухомої руки немає необхідності робити повні оберти, тому для цього буде достатньо серводвигунів з кутом повороту валу на 180 – 270 градусів.

В даному проекті будуть використовуватися сервоприводи з цифровим керуванням MG995 (рис. 3.9).



Рисунок 3.9 – Сервопривод з цифровим керуванням MG995

Даний сервопривід має достатню потужність для утримання та повороту всієї конструкції роботизованої руки, високу точність та плавність рухів [13]. Використання металічного приводу робить його більш надійним та довговічним.

Основні технічні характеристики:

- Кут повороту: 270°;
- Робоча швидкість: 0,17 с/60°;
- Крутний момент: 8,5 – 10 кг/см;
- Робоча напруга: 4,8 – 7,2 В;
- Підшипник ковзання;
- Металічний привід.

На відміну від крокових двигунів, сервоприводи не потребують драйверів та працюють напряму від мікроконтролера. Однак сервоприводу потрібно забезпечити стабільне джерело живлення [11].

Для реалізації механізму переміщення губок пристрою захоплення немає необхідності використовувати потіжні сервоприводи, тому в якості робочого механізму було обрано прості та малопотіжні сервоприводи SG90 (рис. 3.10).



Рисунок 3.10 – Сервопривод з цифровим керуванням SG90

Даний двигун має низьке енергоспоживання та доволі компактні розміри.

Основні технічні характеристики:

- кут повороту: 180°;
- робоча швидкість: 0,12 с/60°;
- крутний момент: 2 кг/см;
- робоча напруга: 3,5 – 5 В;
- робочий струм: 50 – 80 мА.

Як і у випадку з серводвигуном MG995 – керується з мікроконтролера, без використання додаткових драйверів.

3.2.3 Модуль комунікації

Для забезпечення прийому даних зі смартфона (віддалене керування) потрібен модуль безпроводного зв'язку, наприклад Bluetooth. Такий модуль повинен бути адаптований для Arduino, щоб мати можливість безперешкодного встановлення зв'язку між цими пристроями, а також модуль повинен мати режим роботи як ведений пристрій [10].

Серед існуючих Bluetooth-модулів було обрано модуль HC-06 (рис. 3.11) з підтримкою версії Bluetooth 2.0, чого цілком достатньо для реалізації пристрою.

Модуль HC-06 доволі поширений, має низьку вартість та легко підключається і налаштовується.

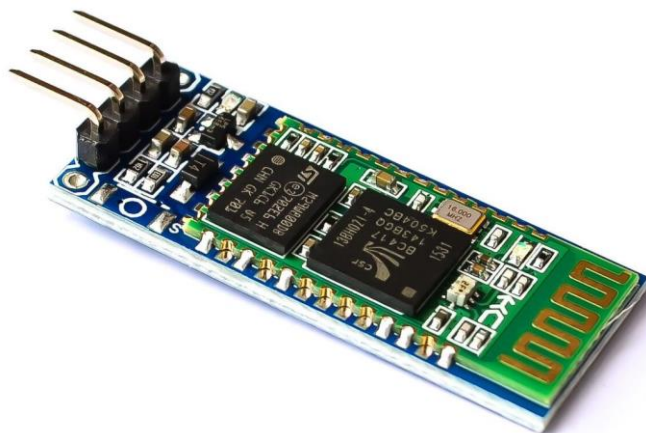


Рисунок 3.11 – Bluetooth-модуль HC-06

3.3 Схема підключення елементів роботизованої руки

Для правильної роботи роботизовано рухомої руки необхідно забезпечити належне підключення всіх електронних компонентів розроблюваного пристрою, починаючи від серводвигунів і завершуючи системою керування та комунікації.

Типова схема підключення крокового двигуна NEMA17 до мікроконтролера за допомогою драйвера A4988 [11] показано на рисунку 3.12

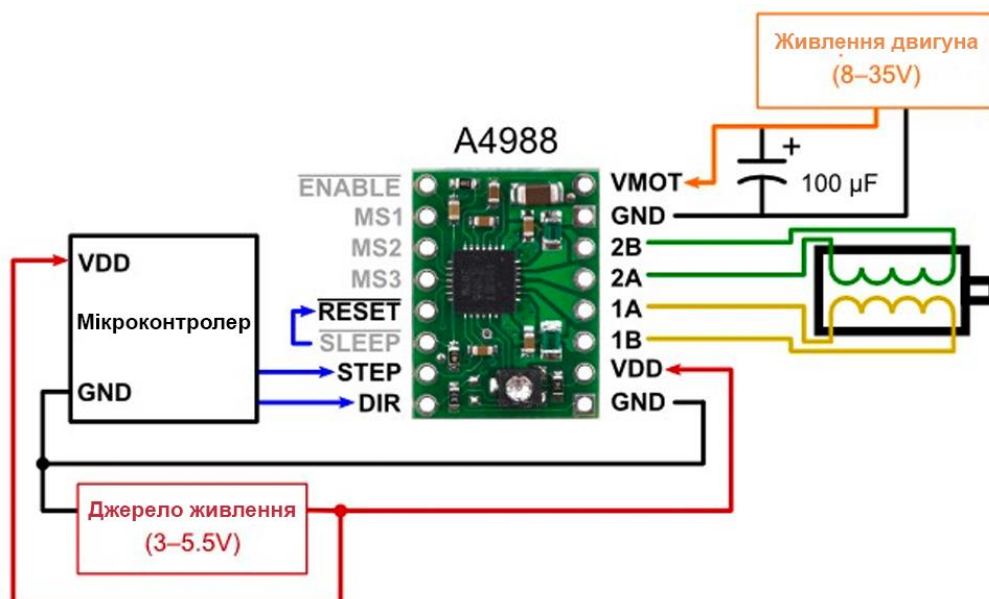


Рисунок 3.12 – Підключення крокового двигуна до мікроконтролера

Взаємодія драйвера з мікроконтролером відбувається через виходи STEP та DIR, яка визначають кут повороту (в кроках) та напрямок обертання відповідно. Підключаються до будь-яких цифрових виходів Arduino. Додатково необхідно з'єднати контакти живлення макроконтролера з пінами VDD та GND драйвера відповідно.

Виходи 1A, 1B, 2A, 2B під'єднуються до відповідних контактів обмоток крокового двигуна. Живлення на кроковий двигун подається через виходи VMOT та GND драйвера [11].

Також потрібно з'єднати виходи SLEEP та RESET драйвера, а на контакти живлення двигуна під'єднати електролітичний конденсатор ємністю 47 – 100 мкФ для захисту мікроконтролера від випадкових стрибків напруги.

Схему підключення крокового двигуна 28BYJ-48 показано на рисунку 3.13.

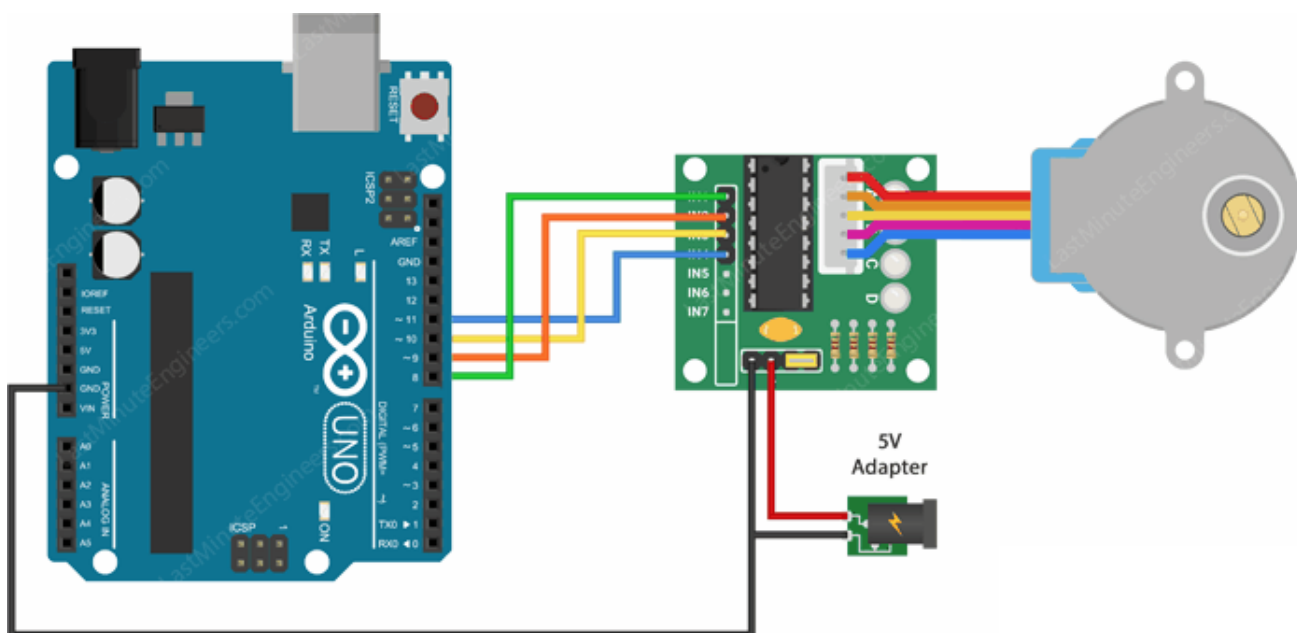


Рисунок 3.13 – Схема підключення крокового двигуна 28BYJ-48

Для підключення даного крокового двигуна використовується драйвер ULN2003, що вже містить готовий роз'єм підключення, тому залишається під'єднати лише драйвер до Arduino: контакти IN1 – IN4 з'єднуються з будь-якими цифровими пінами.

Підключення всіх сервоприводів здійснюється за схожою схемою (рис. 3.14), на відповідні контакти потрібно подати живлення (з окремого блока живлення), а також під'єднати сигнальний контакт до цифрового піна Arduino з підтримкою ШІМ (широтно-імпульсної модуляції) [12].

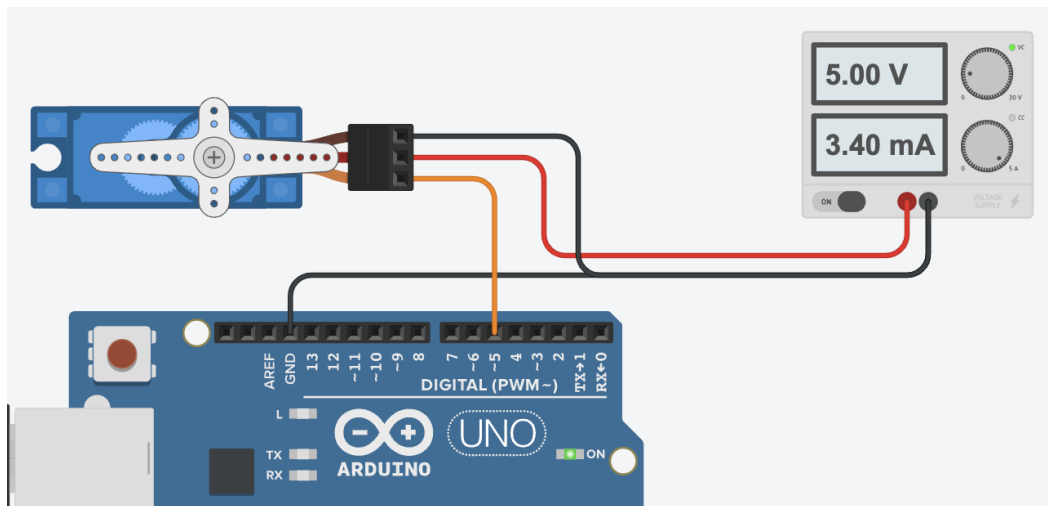


Рисунок 3.14 – Схема підключення сервопривода

Для комунікації зі смартфоном потрібно під'єднати Bluetooth-модуль до мікроконтролера. Типова схема підключення модуля HC-06 до плати Arduino показано на рисунку 3.15.

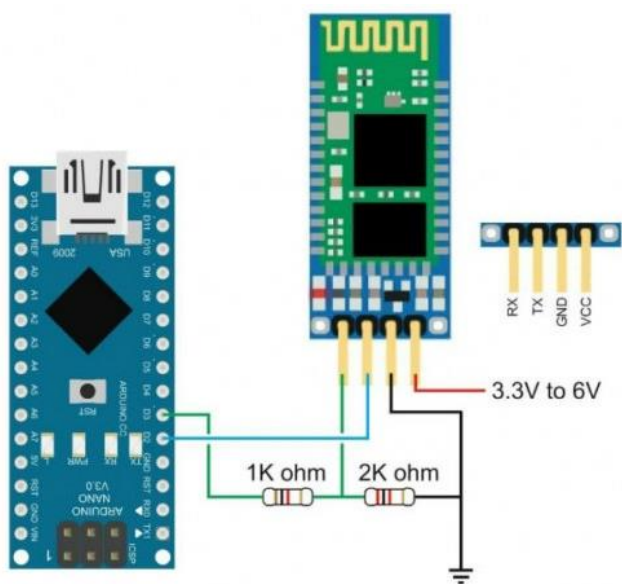


Рисунок 3.15 – Підключення Bluetooth-модуля

Дана плата розширення містить також стабілізатори напруги, що перетворюють вхідну напругу живлення 12 В (для NEMA17) в напругу 5 В, необхідну для живлення плати Arduino та інших двигунів [10].

На основі даної схеми було спроектовано друковану плату для даного рішення (рис. 3.17, 3.18) та виготовлено її (рис. 3.19).

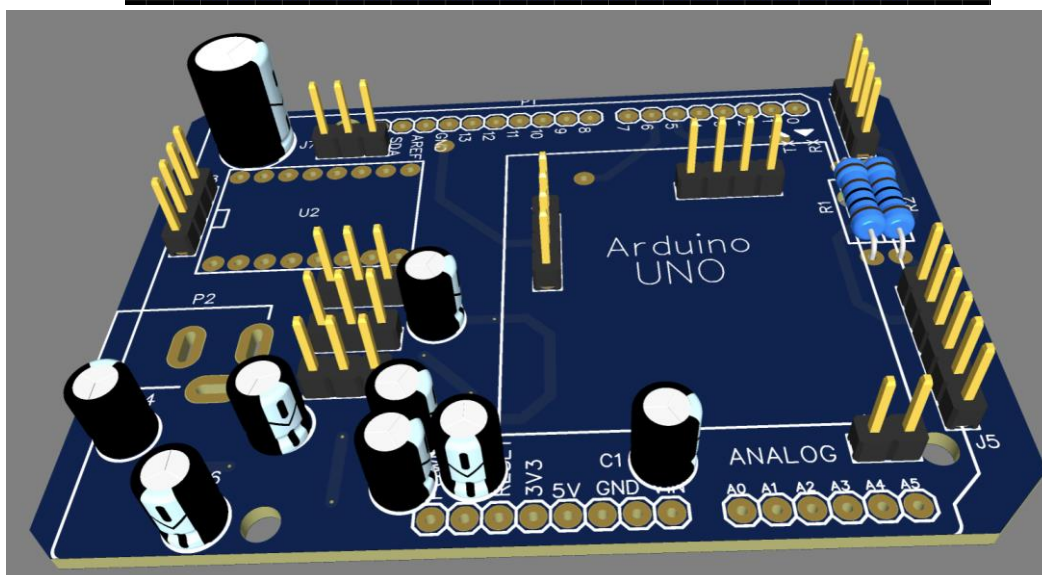
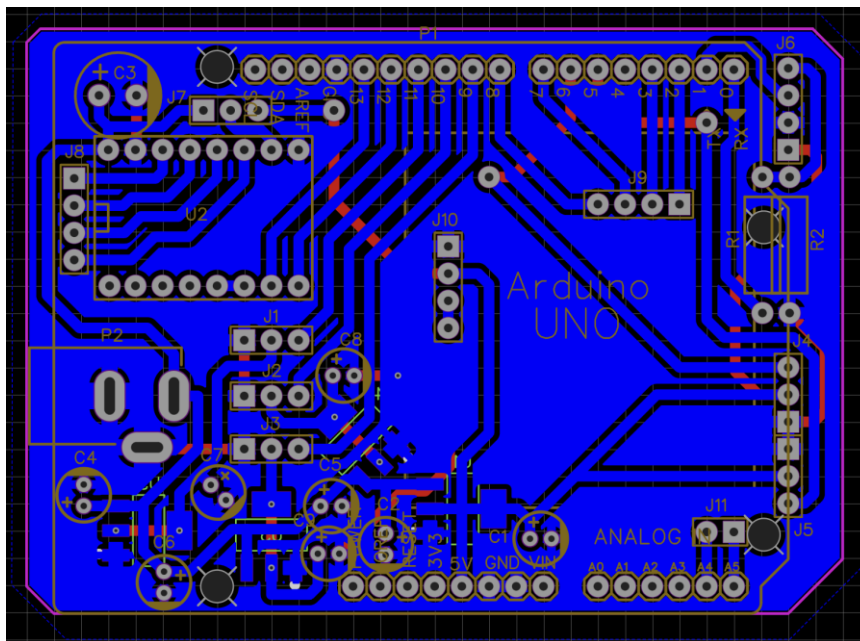


Рисунок 3.17– Спроектвана плата розширення

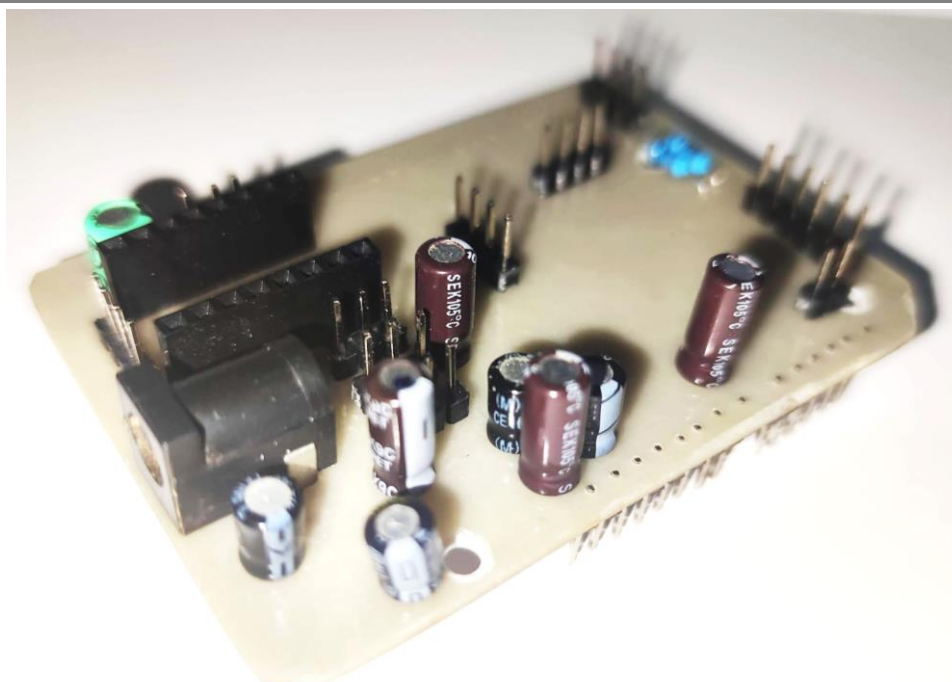
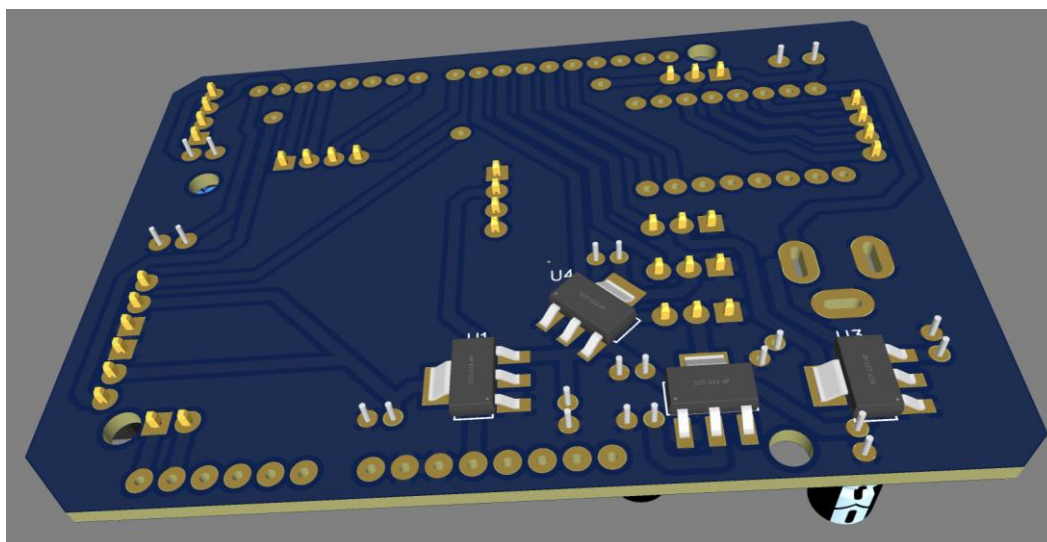


Рисунок 3.18 – 3D-модель плати розширення

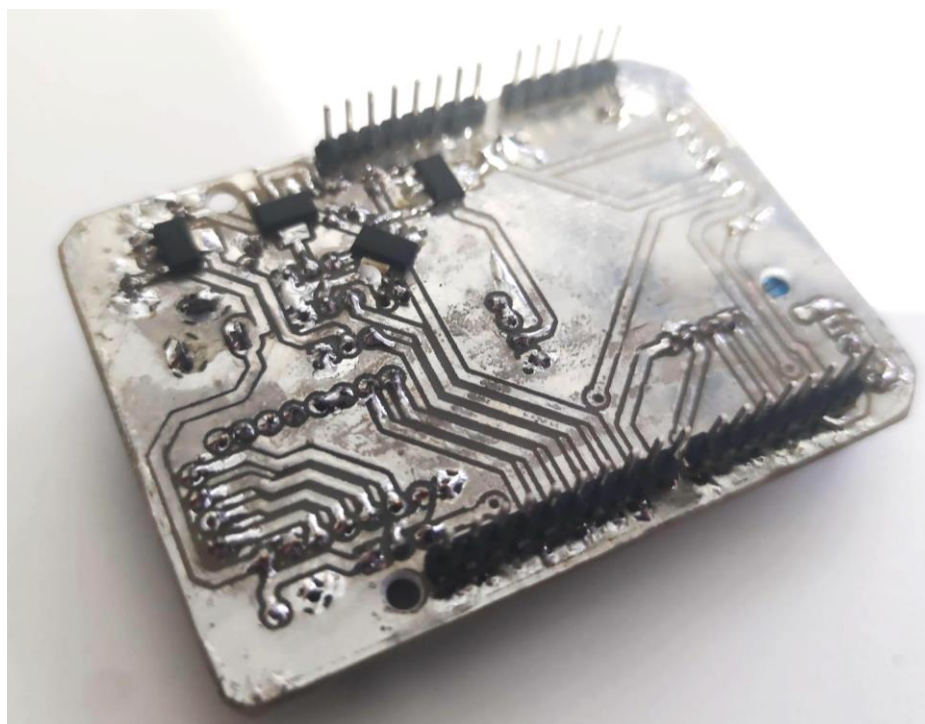


Рисунок 3.19 – Виготовлена плата розширення

3.5 Проектування вузлів роботизованої руки

Важливим етапом розробки проектного робота-маніпулятора є проектування його вузлів. Після здійснення підбору компонентів та всіх необхідних складових потрібно визначитися з матеріалами виготовлення основних деталей конструкції, типи їх з'єднання та інші деталі, що можуть бути використанні при побудові рухомої руки [2].

Для виготовлення основних деталей було вирішено використати 3D-друк, який є в наш час доволі доступним, дозволяє створювати доволі складні деталі, які можна максимально адаптувати до потреб.

В якості матеріалу для друку було обрано PETG пластик. Даний вид пластику має ряд переваг:

- низька деформаційна усадка та високий ступінь адгезії між шарами;
- висока міцність, зносостійкість та ударостійкість;
- стійкий до впливу зовнішнього середовища та ультрафіолету;
- є цілком безпечним при контакті з харчовими продуктами;
- відсутність неприємного запаху при друку;

– відносно невисока вартість.

В основі маніпулятора буде розміщено кроковий двигун, а також вона має тримати всю конструкцію пристрою, тому необхідно забезпечити високу стійкість даної деталі. Не зайвим буде також додати елементи кріплення та фіксації крокового двигуна (рис. 3.20).

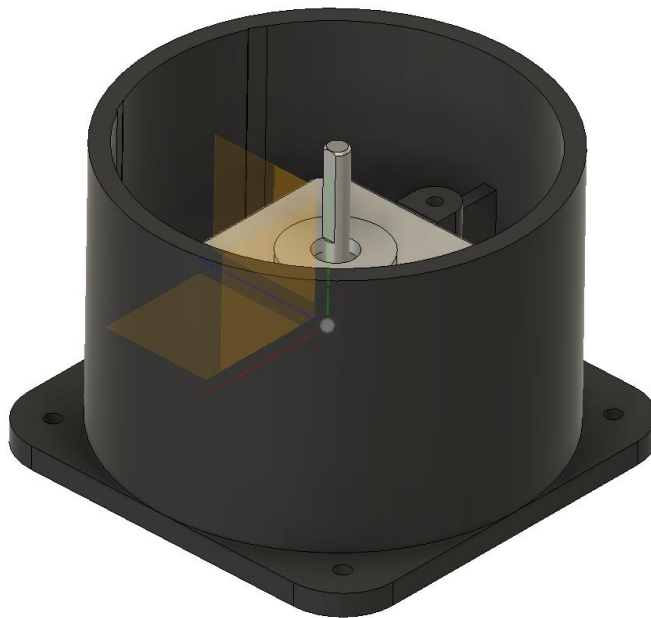


Рисунок 3.20 – Основа робізованої руки

Сам кроковий двигун кріпиться за допомогою спеціальної пластини, яка утримує його у фіксованому положенні (рис. 3.21). Пластина кріпиться гвинтами до двигуна та кріплень основи.

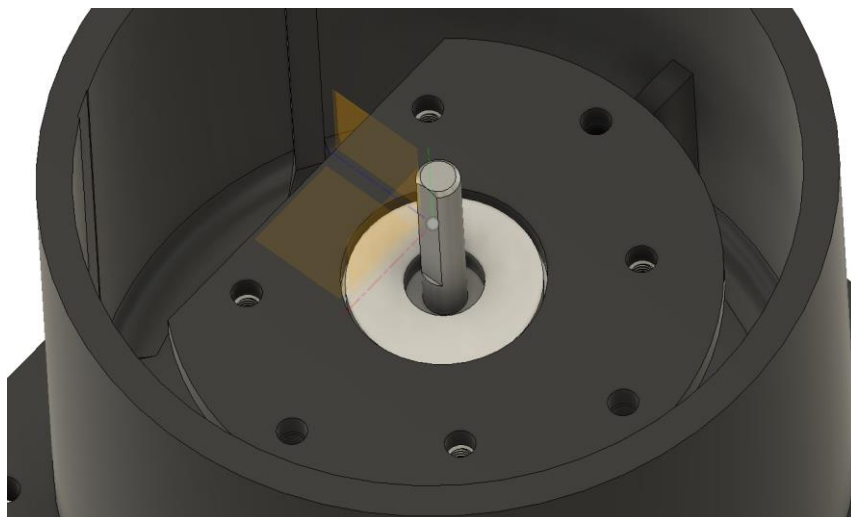


Рисунок 3.21 – Пластина фіксації крокового двигуна

Поворотний механізм, який буде обертатися кроковим двигуном, виконаний у вигляді кришки з кріпленням для сервопривода (рис. 3.22), яка щільно фіксується на валу та вільно обертається навколо осі за рахунок однієї з переваг даного виду пластику – гладкої поверхні.

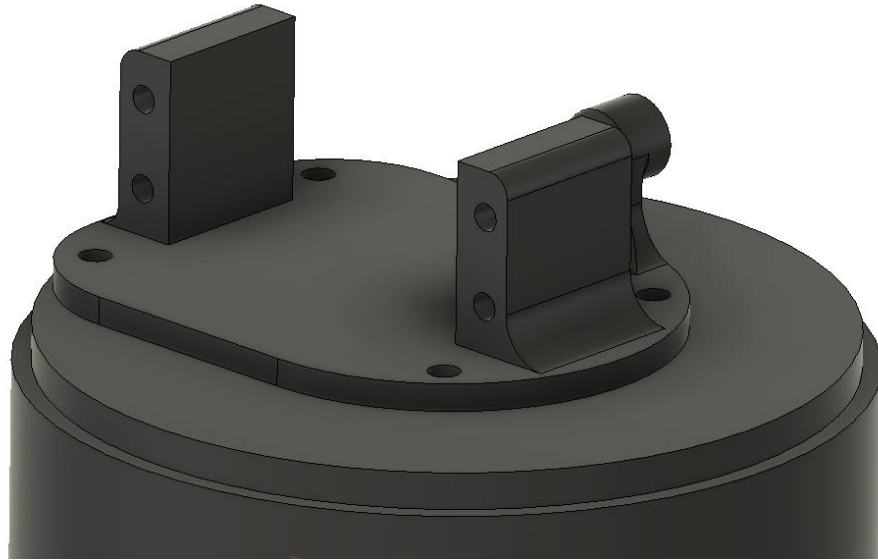


Рисунок 3.22 – Кришка поворотного механізму з кріпленням для сервопривода

На кришку встановлюється серводвигун MG995 з кріпленням під плече роботизованої руки (рис. 3.23).

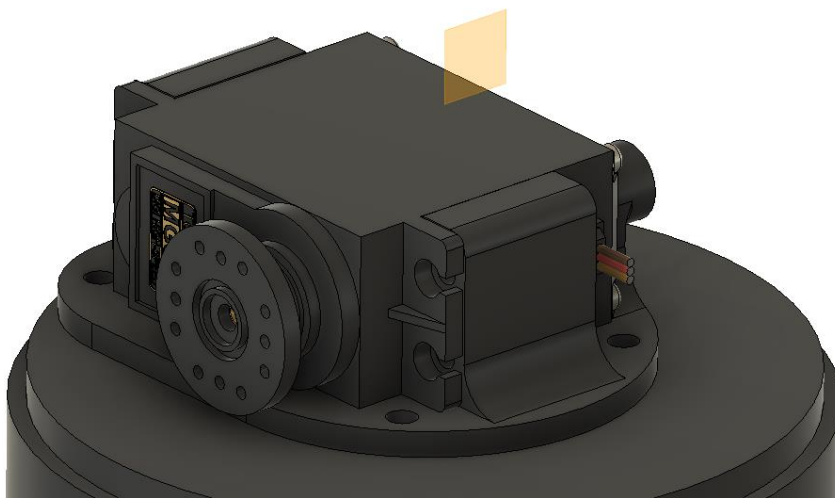


Рисунок 3.23 – Фіксація серводвигуна

До шків серводвигуна кріпиться елемент плеча, що буде використовуватися для здійснення поворотних рухів (рис. 3.24).

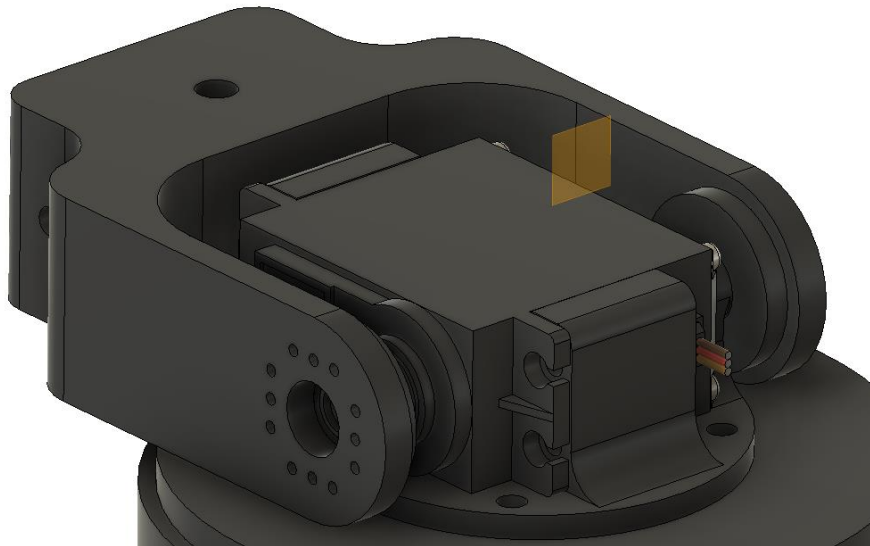


Рисунок 3.24 – Поворотний елемент плеча

Кріплення другого серводвигуна з'єднується з поворотним елементом за допомогою алюмінієвого квадратного профілю 15 x 15 мм (рис. 3.25), який фіксується з обох сторін гвинтами, в деталях також зроблено спеціальні отвори, за допомогою яких профіль використовується для прокладання всередині нього кабелів від двигунів, що робить конструкцію більш надійною та розбавить ризику обриву проводів під час роботи.

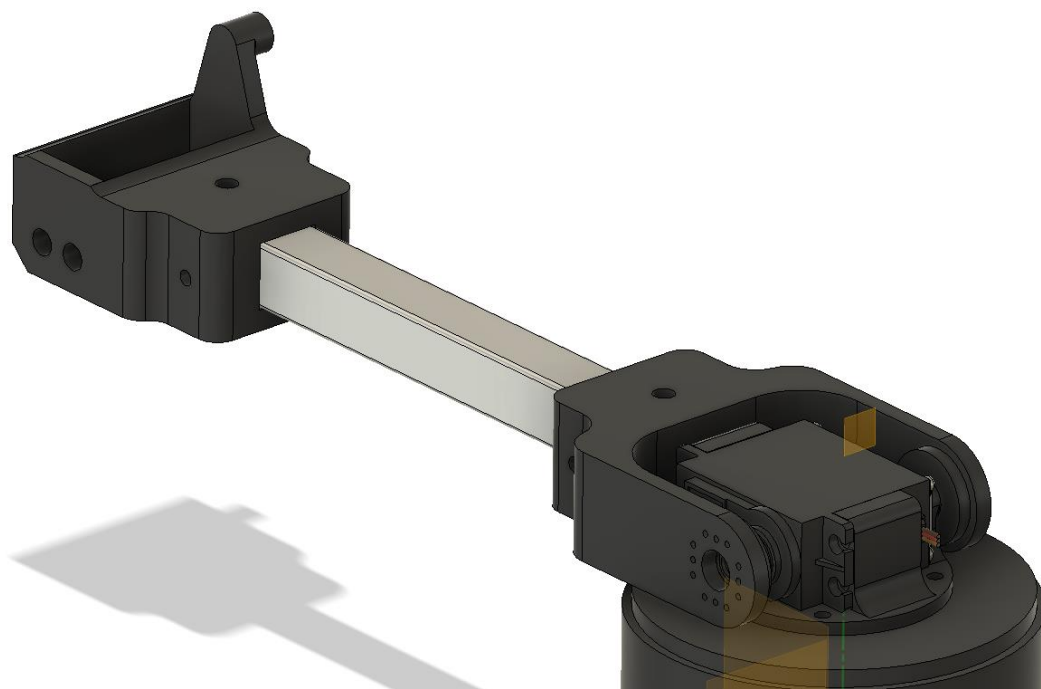


Рисунок 3.25 – Перше плече робота-маніпулятора

Наступне поворотне з'єднання також реалізоване за допомогою серводвигуна та має подібну конструкцію рухомого елемента другого плеча (рис. 3.26).

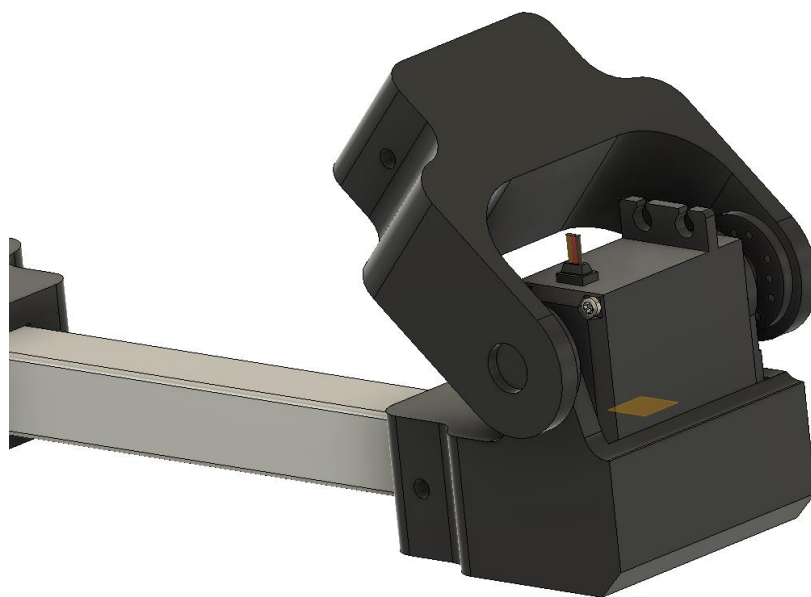


Рисунок 3.26 – Рухомий елемент другого плеча

Аналогічно до першого формується друге плече робота-маніпулятора (рис. 3.27), з використанням такого ж алюмінієвого профілю та використанням його в якості кабелепроводу.

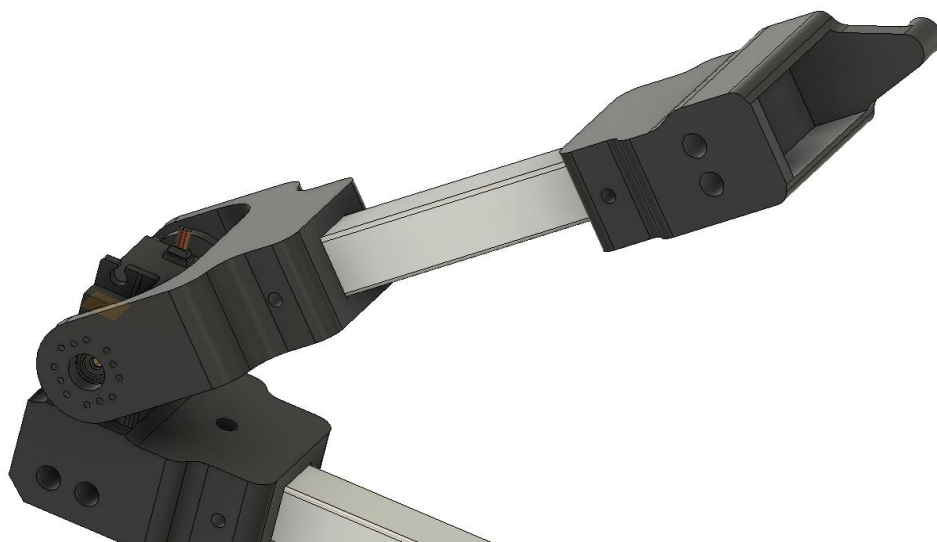


Рисунок 3.27 – Друге плече роботизованої руки

Далі формується «кисть руки» на базі сервопривода та крокового двигуна. До третього серводвигуна кріпиться на рухоме з'єднання елемент утримання крокового двигуна 28BYJ-48 (рис. 3.28).



Рисунок 3.28 – Кріплення крокового двигуна «кисті» роботизованої руки

На шків крокового двигуна, за допомогою гвинтів, кріпиться поворотний механізм «кисті» з направляючою для губок пристрою захоплення (рис. 3.29). Також даний елемент містить посадочні місця для двох сервоприводів, що будуть керувати переміщенням цих губок по направляючій.



Рисунок 3.29 – Поворотний механізм «кисті» руки

На направляючій розміщується пара губок пристрою захоплення, які мають вільний хід вздовж неї (рис. 3.30). За допомогою серводвигунів SG90, що працюють синхронно, та спеціальних дугоподібних направляючих губки захоплюючого пристрою приводяться в рух, отримуючи можливість зафіксувати переміщуваний об'єкт.

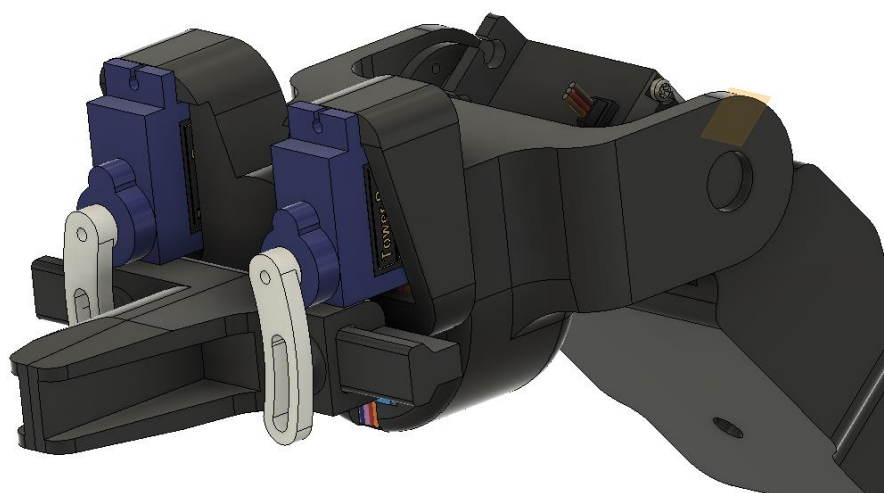


Рисунок 3.30 – Пристрій для захоплення об'єкта

Загальний вигляд роботизованої руки наведено на рис. 3.31

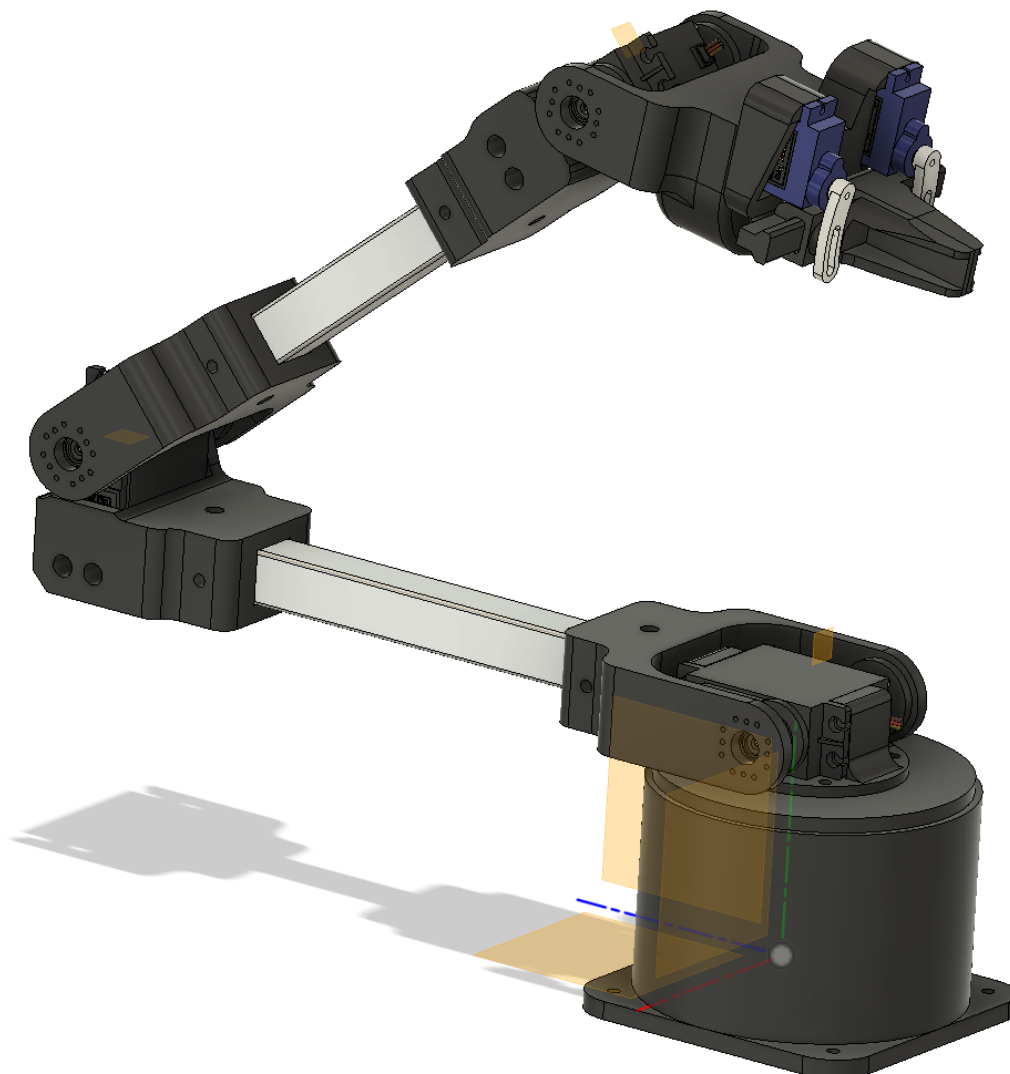


Рисунок 3.31 – Загальний вигляд робота-маніпулятора

Роботизована рука буде кріпитися на пластину разом з основною платою керування (Arduino UNO). Для живлення пристрою потрібно використовувати блок живлення 12 В не менше 4А.

Керування пристроєм може здійснюватися за допомогою провідного пульта керування або з використанням технології Bluetooth зі смартфона на базі операційної системи Android.

Код прошивки робота-маніпулятора, пульта керування та коротку інструкцію з використання пристрою наведено в додатках.

ВИСНОВКИ

Мета даної кваліфікаційної роботи полягала у створенні функціональної роботизованої рухомої руки на платформі Arduino UNO.

Створений робот-маніпулятор отримав 5 ступенів свободи та рухому платформу, що дозволяє йому охоплювати значну робочу область та забезпечувати доволі високу гнучкість при виконанні завдань. Також роботизована рука отримала можливість керування як за допомогою фізичного пульта-керування так і зі смартфона, що робить її більш універсальною.

В процесі виконання даної роботи було вирішено ряд завдань:

- проведено огляд та аналіз предметної області використання роботів-маніпуляторів та розглянуто аналогічні рішення різної конструкції та призначення;

- було сформовано вимоги до вибору системи керування пристроєм, визначено основні критерії, на які варто звернути увагу: надійність, швидкість, точність та повторюваність, гнучкість, безпека, простота використання, сумісність, наявність документації та підтримки.

- визначено основну структуру будови пристрою та підбір елементної бази: мікроконтролерної платформи, виконавчих механізмів (сервоприводів та крокових двигунів), систем комунікації, живлення маніпулятора;

- розроблено технічне завдання на розробку пристрою, представлена загальна кінематична схема робота-маніпулятора та загальна електрична схема підключення основних електронних складових;

- здійснено проектування та виготовлення плати розширення для керування двигунами роботизованої руки;

- здійснено проектування та виготовлення основних механічних вузлів робота-маніпулятора з використанням технологій 3D-друку;

- розроблено програмне забезпечення для мікроконтролерної платформи Arduino UNO, призначене для керування роботизованою рукою;

- наведена коротка інструкція користувача для керування пристроєм.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Михайлов Є.П., Лінгур В.М. Підручник : Маніпулятори та промислові роботи. Одеса: ОНПУ, 2019. 233 с.
2. С. О. Кошель, Ю. Ковальов, О. П. Манойленко. Підручник : Проектування промислових роботів та маніпуляторів. Київ : Центр навчальної літератури, 2019. 256 с.
3. Л.І. Цвіркун, Г. Грулер. Підручник : Робототехніка та мехатроніка, 3-тє видавництво. Дніпро: НГУ, 2019. 224 с.
4. Імшенецький Я. О. Підручник : Розроблення технічних засобів маніпуляційної системи мобільного робота Festo Robotino Том 3. Мукачево, 2021. 41–44 с.
5. Д.М. Проскуренко «Розробка лабораторного маніпулятора», XIII Науково-практична конференція студентів, аспірантів та молодих вчених «Погляд у майбутнє приладобудування», Київ: КПІ ім. Ігоря Сікорського, 2020, с. 121-124
6. Greer, R. Advances in Control Systems for Construction Manipulators. Austin. 2019. 615 с.
7. Д.М. Проскуренко «Система управління маніпуляторами», XIII Науково-практична конференція студентів, аспірантів та молодих вчених «Погляд у майбутнє приладобудування», Київ: КПІ ім. Ігоря Сікорського, 2020, с. 118-120
8. Матвієнко М.П. Підручник : Основи електроніки. Київ: Видавництво Ліра-К, 2020 364 с.
9. Болюх В.Ф. Основи електротехніки, електроніки та мікропроцесорної техніки. Харків: ПланетаПрінт, 2019. 248с.
10. Квітка С.О. Електроніка та мікросхемотехніка. Мелітополь: Видавничо-поліграфічний центр «Люкс», 2019. 223с.
11. Морзе Н.В. Основи робототехніки: навчальний посібник. Кам'янець-Подільський : ПП Буйницький О.А., 2022. 184 с.
12. Tutorials | Arduino Documentation. URL: <https://docs.arduino.cc/tutorials/> (дата звернення: 16.03.2023).

13. Як побудований сервопривід. Принцип роботи. URL: <https://modelistam.com.ua/ua/kakustroen-servoprivod-printsip-raboty-a-164/> (дата звернення: 16.03.2023).

14. Бучинський М. Я., Горик О. В., Чернявський А. М., Яхін С. В. Основи творення машин. Харків: Вид-во «НТМТ», 2019. 448 с.

15. Румбешта В. О. Технологія складання, регулювання та випробування приладів. – Київ: НТУУ «КПІ», 2020. 360 с.

16. Service Robotics Market Analysis By Application (Professional, Personal) And Segment Forecasts To 2020. Published: November 2021 .

17. Ельперін І.В. Автоматизація виробничих процесів. Підручник. Вид. 2-ге, виправлене. Київ: Видавництво Ліра-К, 2021. 364 с.

18. Бережна М.А. Комп'ютерні технології автоматизованого виробництва. Харків: ТОВ «Компанія СМІТ», 2019. 368 с.

19. Маляр В.С. Теоретичні основи електротехніки. Електричні кола. Львів: Видавництво Львівської політехніки, 2019. 312с.

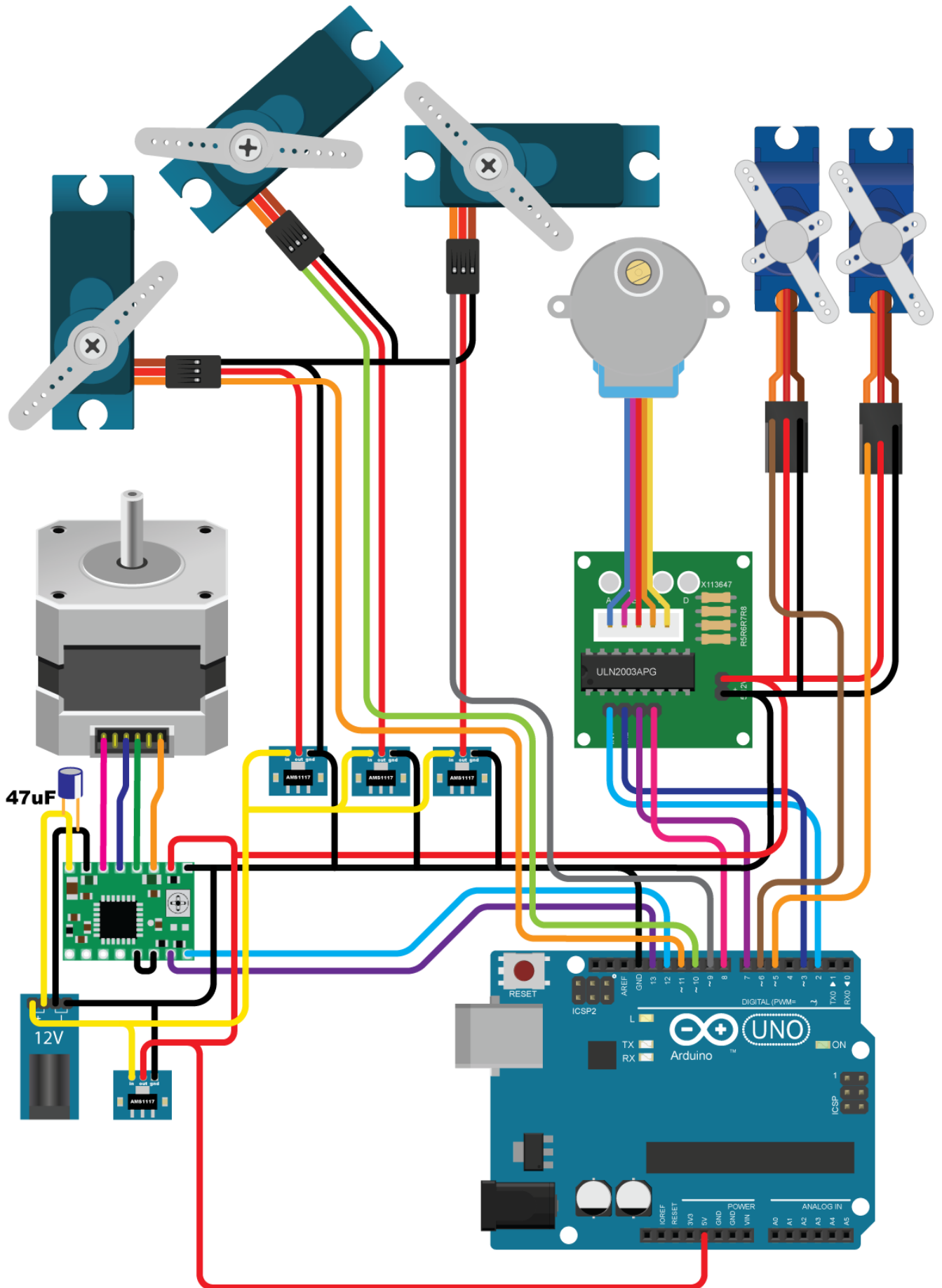
20. Гуржій А. М. Електротехніка та основи електроніки. Підручник для здобувачів професійної (професійно-технічної) освіти. Київ: Літера ЛТД, 2020. 288 с.

21. Цвіркун Л.І. Робототехніка та мехатроніка. Дніпро: НГУ, 2019. 224 с.

ДОДАТКИ

Додаток А

Загальна схема підключення елементів роботизованої руки



Додаток Б

Код прошивки роботизованої рухомої руки

```
#include <Wire.h> //бібліотека для інтерфейсу I2C
#include <Servo.h> //бібліотека для серводвигунів
#include <AccelStepper.h> //бібліотека для крокового двигуна NEMA17
#include <Stepper_28BYJ.h> //бібліотека для крокового двигуна 28-BYJ48
#include <EEPROM.h> //бібліотека для роботи з EEPROM-пам'яттю

//визначаємо піни підключення крокового двигуна NEMA17
#define STEP 13
#define DIR 12

//визначаємо піни підключення крокового двигуна 28-BYJ48
#define IN1 2
#define IN2 3
#define IN3 7
#define IN4 8

//визначаємо піни підключення серводвигунів
#define SPIN1 11
#define SPIN2 10
#define SPIN3 9
#define SPIN4 6
#define SPIN5 5

//визначаємо максимальні кути повороту серводвигунів та крокових двигунів
#define MAX_SV1 120
#define MAX_SV2 180
#define MAX_SV3 180
#define MAX_SV45 60
#define MAX_ST1 180
#define MAX_ST2 120

//визначаємо адреси для зберігання даних в EEPROM
#define SV1_ADDR 0
#define SV2_ADDR 2
#define SV3_ADDR 4
#define SV45_ADDR 6
#define ST1_ADDR 8
#define ST2_ADDR 10

//визначаємо адресу пульта керування
#define I2C_ADDR 28

//створюємо об'єкт для крокового двигуна NEMA17 та задаємо його параметри - режим та піни підключення
AccelStepper NEMA17(1, STEP, DIR);
```

```

//створюємо об'єкт для крокового двигуна 28-BYJ48 та задаємо його параметри - кількість кроків та піни підключення
Stepper_28BYJ ST28BYJ(4078, IN1, IN2, IN3, IN4);

//створюємо об'єкт Servo для кожного серводвигуна
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
Servo servo5;

//змінні для зберігання проміжних даних
int16_t sv1_angle, sv2_angle, sv3_angle, sv45_angle, st1_angle, st2_angle; //поточне значення кута повороту
byte receive[12]; //буфер для отриманих даних

void setup() {
  //задаємо параметри роботи крокового двигуна NEMA17
  NEMA17.setMaxSpeed(3000); //максимальна швидкість обертання
  NEMA17.setAcceleration(13000); //прискорення
  //задаємо максимальну швидкість обертання крокового двигуна 28-BYJ48
  ST28BYJ.setSpeed(13);
  //підключаємо керування серводвигунами
  servo1.attach(SPIN1);
  servo2.attach(SPIN2);
  servo3.attach(SPIN3);
  servo4.attach(SPIN4);
  servo5.attach(SPIN5);
  //ініціюємо інтерфейс передачі даних для Bluetooth
  Serial.begin(9600);
  //ініціюємо протокол I2C
  Wire.begin();
  //зчитуємо дані з EEPROM
  EEPROM.get(SV1_ADDR, sv1_angle);
  EEPROM.get(SV2_ADDR, sv2_angle);
  EEPROM.get(SV3_ADDR, sv3_angle);
  EEPROM.get(SV45_ADDR, sv45_angle);
  EEPROM.get(ST1_ADDR, st1_angle);
  EEPROM.get(ST2_ADDR, st2_angle);
}

void loop() {
  Wire.requestFrom(I2C_ADDR, 12); //робимо запит на отримання 12 байт даних по I2C
  int n = 0; //створюємо лічильник
  while(Wire.available()){ //поки є дані для отримання
    receive[n] = Wire.read(); //записуємо отримані дані в масив
    n++; //збільшуємо значення лічильника
  }
  analyze_data();
}

```

```

//функція запису даних до EEPROM
void writeEEPROM() {
    EEPROM.put(SV1_ADDR, sv1_angle);
    EEPROM.put(SV2_ADDR, sv2_angle);
    EEPROM.put(SV3_ADDR, sv3_angle);
    EEPROM.put(SV45_ADDR, sv45_angle);
    EEPROM.put(ST1_ADDR, st1_angle);
    EEPROM.put(ST2_ADDR, st2_angle);
}

//аналізуємо отримані дані
void analyze_data() {
    if (receive[1] != 0) {
        if (receive[0] == 0) sv1_angle -= receive[1]; else sv1_angle += receive[1];
        if (sv1_angle > MAX_SV1) sv1_angle = MAX_SV1; else
        if (sv1_angle < 0) sv1_angle = 0;
        servo1.write(sv1_angle);
    }
    if (receive[3] != 0) {
        if (receive[2] == 0) sv2_angle -= receive[3]; else sv2_angle += receive[3];
        if (sv2_angle > MAX_SV2) sv2_angle = MAX_SV2; else
        if (sv2_angle < 0) sv2_angle = 0;
        servo2.write(sv2_angle);
    }
    if (receive[5] != 0) {
        if (receive[4] == 0) sv3_angle -= receive[5]; else sv3_angle += receive[5];
        if (sv3_angle > MAX_SV3) sv3_angle = MAX_SV3; else
        if (sv3_angle < 0) sv3_angle = 0;
        servo3.write(sv3_angle);
    }
    if (receive[7] != 0) {
        if (receive[6] == 0) sv45_angle -= receive[7]; else sv45_angle += receive[7];
        if (sv45_angle > MAX_SV45) sv1_angle = MAX_SV45; else
        if (sv45_angle < 0) sv45_angle = 0;
        servo4.write(sv45_angle);
        servo5.write(MAX_SV45 - sv45_angle);
    }
    if (receive[9] != 0) {
        int angle = 0;
        if (receive[8] == 0) angle -= receive[9]; else angle += receive[9];
        if (st1_angle + angle > MAX_ST1) angle = st1_angle + angle - MAX_ST1; else
        if (st1_angle + angle < (-1) * MAX_ST1) angle = (-1) * MAX_ST1 - st1_angle;
        NEMA17.move(angle / 1.8);
        st1_angle += angle;
    }
    if (receive[11] != 0) {
        int angle = 0;

```

```
if (receive[10] == 0) angle -= receive[11]; else angle += receive[11];
if (st2_angle + angle > MAX_ST2) angle = st2_angle + angle - MAX_ST2; else
if (st2_angle + angle < (-1) * MAX_ST2) angle = (-1) * MAX_ST2 - st2_angle;
ST28BYJ.step(angle / 5.625);
st2_angle += angle;
}
writeEEPROM();
}
```

Додаток В

Код прошивки пульта керування роботизованої рухомої руки

```
#include <Wire.h>

#define EN1_A 2
#define EN1_B 3
#define EN2_A 4
#define EN2_B 5
#define EN3_A 6
#define EN3_B 7
#define EN4_A 8
#define EN4_B 9
#define EN5_A 10
#define EN5_B 11
#define EN6_A 12
#define EN7_B 13

const byte SLAVE_ADDRESS = 28;
byte data[12];
int stepEnc1 = 0, stepEnc2 = 0, stepEnc3 = 0, stepEnc4 = 0, stepEnc5 = 0, stepEnc6 = 0;
byte buffEnc1[2], buffEnc2[2], buffEnc3[2], buffEnc4[2], buffEnc5[2], buffEnc6[2];
byte flagEnc1 = 0, flagEnc2 = 0, flagEnc3 = 0, flagEnc4 = 0, flagEnc5 = 0, flagEnc6 = 0;

void setup() {
  pinMode(EN1_A, INPUT);
  pinMode(EN1_B, INPUT);
  pinMode(EN2_A, INPUT);
  pinMode(EN2_B, INPUT);
  pinMode(EN3_A, INPUT);
  pinMode(EN3_B, INPUT);
  pinMode(EN4_A, INPUT);
  pinMode(EN4_B, INPUT);
  pinMode(EN5_A, INPUT);
  pinMode(EN5_B, INPUT);
  pinMode(EN6_A, INPUT);
  pinMode(EN6_B, INPUT);
  Wire.begin(SLAVE_ADDRESS);
  Wire.onRequest(requestEvent);
}

void loop() {
  enc();
  if (flagEnc1) {
    if (stepEnc1 < 0) data[0] = 0; else data[0] = 1;
    data[1] = abs(stepEnc1);
    flagEnc1 = 0;
  } else {
    data[0] = 0;
  }
}
```

```

    data[1] = 0;
}
if (flagEnc2) {
    if (stepEnc2 < 0) data[2] = 0; else data[2] = 1;
    data[3] = abs(stepEnc2);
    flagEnc2 = 0;
} else {
    data[2] = 0;
    data[3] = 0;
}
if (flagEnc3) {
    if (stepEnc3 < 0) data[4] = 0; else data[4] = 1;
    data[5] = abs(stepEnc3);
    flagEnc3 = 0;
} else {
    data[4] = 0;
    data[5] = 0;
}
if (flagEnc4) {
    if (stepEnc4 < 0) data[6] = 0; else data[6] = 1;
    data[7] = abs(stepEnc4);
    flagEnc4 = 0;
} else {
    data[6] = 0;
    data[7] = 0;
}
if (flagEnc5) {
    if (stepEnc5 < 0) data[8] = 0; else data[8] = 1;
    data[9] = abs(stepEnc5);
    flagEnc5 = 0;
} else {
    data[8] = 0;
    data[9] = 0;
}
if (flagEnc6) {
    if (stepEnc6 < 0) data[10] = 0; else data[10] = 1;
    data[11] = abs(stepEnc6);
    flagEnc6 = 0;
} else {
    data[10] = 0;
    data[11] = 0;
}
}

void enc() {
    static unsigned long timer;
    if (timer + 5 > millis()) return;

```

```

buffEnc1[0] = buffEnc1[1];
buffEnc1[1] = digitalRead(EN1_A);
if (buffEnc1[0] != buffEnc1[1]) {
  stepEnc1 += digitalRead(EN1_B) == buffEnc1[1] ? 1 : -1;
  flagEnc1 = 1;
}
buffEnc2[0] = buffEnc2[1];
buffEnc2[1] = digitalRead(EN2_A);
if (buffEnc2[0] != buffEnc2[1]) {
  stepEnc2 += digitalRead(EN2_B) == buffEnc2[1] ? 1 : -1;
  flagEnc2 = 1;
}
buffEnc3[0] = buffEnc3[1];
buffEnc3[1] = digitalRead(EN3_A);
if (buffEnc3[0] != buffEnc3[1]) {
  stepEnc3 += digitalRead(EN3_B) == buffEnc3[1] ? 1 : -1;
  flagEnc3 = 1;
}
buffEnc4[0] = buffEnc4[1];
buffEnc4[1] = digitalRead(EN4_A);
if (buffEnc4[0] != buffEnc4[1]) {
  stepEnc4 += digitalRead(EN4_B) == buffEnc4[1] ? 1 : -1;
  flagEnc4 = 1;
}
buffEnc5[0] = buffEnc5[1];
buffEnc5[1] = digitalRead(EN5_A);
if (buffEnc5[0] != buffEnc5[1]) {
  stepEnc5 += digitalRead(EN5_B) == buffEnc5[1] ? 1 : -1;
  flagEnc5 = 1;
}
buffEnc6[0] = buffEnc6[1];
buffEnc6[1] = digitalRead(EN6_A);
if (buffEnc6[0] != buffEnc6[1]) {
  stepEnc6 += digitalRead(EN6_B) == buffEnc6[1] ? 1 : -1;
  flagEnc6 = 1;
}
timer = millis();
}

void requestEvent() {
  Wire.write(data, 12);
}

```

Додаток Г

Коротка інструкція з користування

Для правильного використання роботизованої руки необхідно:

- 1) забезпечити розміщення робота-маніпулятора та рівній твердій поверхні;
- 2) переконатися, що для руху роботизованої руки не буде створюватися зайвих перешкод;
- 3) забезпечити стабільне безперебійне живлення від комплектного блока живлення або сумісного (12В, 3А).

Підготовка до роботи:

- 1) підключити пульт керування до відповідного роз'єму на платі розширення;
- 2) підключити блок живлення до плати розширення;
- 3) увімкнути блок живлення в розетку;
- 4) пристрій готовий до роботи.

Керування пристроєм здійснюється за допомогою спеціального пульта, основними елементами якого є поворотні ручки еncoderів (рис. Г.1). Кожна ручка еncoderа відповідає за свій елемент керування. Поворот ручки еncoderа за часовою стрілкою або проти здійснює відповідний поворот сервопривода або крокового двигуна роботизованої руки.

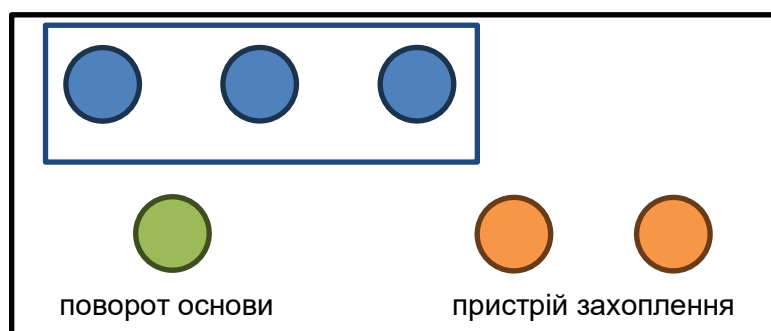


Рисунок Г.1 – Схематичне зображення пульта керування