

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач випускової кафедри

_____ Аліна САВЧЕНКО

«__»_____2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНОВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ МАГІСТР
ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ
«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ»

Тема: «Дистанційно керована роботизована рука на основі Arduino»

Виконавець: Дмитро КОЗЛОВЦЕВ

Керівник: к.т.н., доцент Олег ЗУДОВ

Нормоконтролер: к.т.н., доцент Олена ТОЛСТИКОВА

КИЇВ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій
Кафедра комп'ютерних інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ:
завідувач кафедри КІТ
Аліна САВЧЕНКО
(підпис)
«_____» _____ 2023 р.

ЗАВДАННЯ на виконання кваліфікаційної роботи Козловцева Дмитра Сергійовича (ПІБ випускника)

1. Тема роботи: «Дистанційно керована роботизована рука на основі Arduino» затверджена наказом ректора № 1976/ст від 29.09.2023р.
2. Термін виконання роботи: з 02 жовтня 2023 року по 31 грудня 2023 року.
3. Вихідні дані до роботи: Прототип дистанційно керованої роботизованої руки на основі Arduino для демонстрації доступних технологій роботобудовання.
4. Зміст пояснювальної записки: 1. Аналіз об'єкта та предмета дослідження. 2. Проектування роборуки та інструменти розробки. 3. Розробка та тестування роборуки.
5. Перелік обов'язкового ілюстративного матеріалу: 1. Артикульована Роборука. 2. Arduino Uno Rev3. 3. Bluetooth Модуль HC-05. 4. Стійка основа. 5. Актuator плеча вмонтований в систему. 6. Монтування ланки ліктя. 7. Основа зап'ястя роборуки. 8. Монтування маніпулятора. 9. Схема підключення роборуки до ланцюга живлення. 10. Підключення Bluetooth модуля.

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1.	Аналіз предметної області та огляд аналогів. Написання 1 розділу, аналіз та поняття технології	02.10.2023- 18.10.2023	
2.	Вибір та опис використаних технологій. Написання 2 розділу, проектування застосунку	19.10.2023- 29.10.2023	
3.	Написання 3 розділу, розробка та тестування прототипу	30.10.2023- 11.11.2023	
4.	Загальне редагування та друк пояснювальної записки	12.11.2023- 19.11.2023	
5.	Проходження нормоконтролю, перепліт пояснювальної записки.	12.11.2023- 19.10.2023	
6.	Розробка тексту доповіді. Оформлення графічного матеріалу для презентації	19.12.2023- 22.12.2023	

7. Дата видачі завдання _____ 02.10.2023 р.

Керівник кваліфікаційної роботи _____ Олег ЗУДОВ
(підпис керівника)

Завдання прийняв до виконання _____ Дмитро КОЗЛОВЦЕВ
(підпис випускника)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи на тему: «Дистанційно керована роботизована рука на основі Arduino» містить: 107 сторінок, 50 рисунків, 15 інформаційних джерел, 1 додаток.

Об'єкт дослідження - система дистанційного управління роботизованими пристроями.

Предмет дослідження - розробка та оптимізація артикульованої роботичної руки на основі Arduino з дистанційним керуванням за допомогою технології Bluetooth.

Мета кваліфікаційної роботи - це розробка доступної та ефективної моделі роботизованої руки на базі Arduino з дистанційним керуванням.

Методи дослідження - мова програмування C, інтегроване середовище розробки Arduino IDE.

Результати кваліфікаційної роботи рекомендується використовувати для демонстрацій робототехніки і артикульованих роботичних рух в навчальних закладах.

ДИСТАНЦІЙНЕ КЕРУВАННЯ, АРТИКУЛЬОВАНА РОБОТИЧНА РУКА, РОБОТОТЕХНІКА, C, ПРОТОТИП ,BLETOOTH, ARDUINO

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	2
ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ОБ’ЄКТА ТА ПРЕДМЕТА ДОСЛІДЖЕННЯ	10
1.1. Розвиток робототехніки	10
1.2. Основні характеристики і принципи роботи роботизованих механічних систем	15
1.3. Артикульована роботична рука.....	55
1.4. Порівняння існуючих аналогів	68
ВИСНОВКИ ДО РОЗДІЛУ 1	70
РОЗДІЛ 2 ПРОЕКТУВАННЯ РОБОРУКИ ТА ІНСТРУМЕНТИ РОЗРОБКИ	71
2.1. Опис вимог до прототипу.....	71
2.2. Архітектура та компоненти роботичної руки	72
ВИСНОВКИ ДО РОЗДІЛУ 2	84
РОЗДІЛ 3 РОЗРОБКА І ТЕСТУВАННЯ ПРОТОТИПА РОБОРУКИ.....	85
3.1. Конструювання роборуки.....	85
3.2. Програмне забезпечення.....	94
3.3. Інтерфейс користувача та юз-кейси.....	98
3.4. Аналіз прототипу	100
ВИСНОВКИ ДО РОЗДІЛУ 3	101
ВИСНОВКИ.....	102
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	103
ДОДАТОК А.....	105

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

<i>PWM (Pulse Width Modulation)</i>	–	Широтно-імпульсна модуляція
<i>IDE (Integrated development environment)</i>	–	Інтегроване середовище розробки
<i>DOF (Degrees of Freedom)</i>	–	Ступені свободи
<i>ПЗ (Програмне Забезпечення)</i>	–	Алгоритми, які відповідають за функціонування руки
<i>BLDC (Brushless DC electric motor)</i>	–	Безщіткові DC Мотори
<i>DC (Direct current)</i>	–	Постійний струм
<i>AC (alternating current)</i>	–	Змінний струм
<i>АЦП (Аналогово-цифровий перетворювач)</i>	–	Механізм, який перетворює аналоговий сигнал в дискретний код
<i>SCARA (Selective Compliance Articulated Robot Arm)</i>	–	Артикульована рука вибіркового виконання

ВСТУП

Сучасний світ переживає значний технологічний розквіт, особливо в області робототехніки. Це відкриває широкі можливості для автоматизації рутинних завдань, покращення якості життя людей та вирішення задач, які раніше вважалися неможливими або надто небезпечними для людей. Однак, поряд з цими можливостями, існують проблеми та виклики, які необхідно вирішити.

Однією з таких проблем є доступність та вартість робототехнічних рішень. Більшість сучасних роботизованих систем вимагає значних капіталовкладень, що робить їх недоступними для більшості користувачів, особливо для освітніх установ, малих підприємств або окремих ентузіастів.

Крім того, часто роботизовані системи розробляються під конкретні завдання, що обмежує їхнє універсальне використання та адаптацію під нові задачі. Це вимагає додаткових вкладень часу та коштів на переналаштування та модернізацію.

Ще однією проблемою є складність взаємодії людини з роботизованою системою. Для ефективного управління та взаємодії потрібно розробляти інтуїтивно зрозумілі інтерфейси, а також забезпечувати можливість дистанційного керування.

Мета дослідження - це розробка доступної та ефективної моделі роботизованої руки на базі Arduino для дистанційного керування.

Роботизовані системи знаходять застосування в ряді областей, від медицини до промисловості, проте їх вартість та складність часто обмежують доступ до них для більшості користувачів. В даному контексті основною метою цього дослідження є створення роботизованої руки, яка була б не тільки ефективною, але й доступною для широкого загалу.

Arduino, як відкрита платформа, вже довгий час служить основою для різноманітних інноваційних проектів завдяки своїй простоті, гнучкості та низькій вартості. Використання Arduino як основи для роботизованої руки

може забезпечити користувачам надійний та гнучкий інструмент для дистанційного керування.

Така роботизована рука може стати в нагоді для навчальних закладів, де студенти та викладачі могли б за її допомогою досліджувати основи робототехніки та програмування. Додатково, така доступна система може служити відмінним рішенням для хобіїстів, DIY-ентузіастів або навіть для промислових досліджень та прототипування.

Об'єкт дослідження – система дистанційного управління роботизованими пристроями.

Роботичні системи останнім часом набули широкого поширення у різних сферах. Вони дозволяють автоматизувати рутинні та складні завдання, підвищуючи ефективність та безпеку. Особливий інтерес представляють роботизовані руки, які можуть імітувати рухи людської руки, здійснюючи точні та складні маніпуляції. Дистанційне керування роботизованими руками відкриває широкі можливості для їх застосування в умовах, де безпосередня участь людини є небезпечною або неможливою.

Предмет дослідження - розробка та оптимізація артикульованої роботичної руки на основі Arduino з дистанційним керуванням за допомогою технології Bluetooth. Розробка охоплює детальний аналіз механічної структури, вивчення кінематики та дизайну руки, а також програмування мікроконтролера Arduino для ефективного контролю над рухами руки. Окрім того, важливою частиною є створення системи дистанційного керування через Bluetooth, що включає розробку користувацького інтерфейсу та забезпечення безпечного бездротового зв'язку.

Список завдань для досягнення мети:

1. Аналіз потреб та вимог.

- вивчення існуючих рішень на ринку;
- оцінка потреб користувачів та основних вимог до роборуки;
- визначення областей застосування та цільової аудиторії.

2. Вибір компонентів.

- вибір найбільш оптимальних компонентів за ціною та якістю.

3. Розробка прототипу.

- створення схеми підключення компонентів;
- збірка прототипу.

4. Програмування.

- розробка ПЗ для керування роботизованою рукою;
- інтеграція зворотний зв'язку та інтерфейсу користувача.

5. Тестування.

- проведення тестів для перевірки ефективності прототипу.

Виконання цих завдань допоможе досягти поставленої мети та створити ефективний, доступний і водночас гнучкий інструмент для дистанційного керування на базі платформи Arduino.

РОЗДІЛ 1

АНАЛІЗ ОБ'ЄКТА ТА ПРЕДМЕТА ДОСЛІДЖЕННЯ

1.1. Розвиток робототехніки

У давніх грецьких та римських цивілізаціях з'явилися одні з перших автоматизованих машин, які були більше мистецькими творами, ніж практичними пристроями. Наприклад, вони створювали автомати, що імітували дії тварин та людей. Ці машини часто використовували гідравлічні та пневматичні системи, що були досить примітивними, але вражаючими для свого часу. Вони мали велике символічне значення і часто використовувалися в релігійних церемоніях або як розваги для знаті.

Ренесанс, епоха відродження і інтенсивних інтелектуальних змін у Європі, також став важливим періодом у розвитку робототехніки. Цей час відзначився виникненням нових ідей та інновацій в науці, інженерії та мистецтві, що надихали на створення механічних пристроїв, які можна розглядати як далеких предків сучасних роботів

Найбільш відомий приклад в області робототехніки - так званий "робот-солдат" (Рис. 1.1). Це був досить складний механічний пристрій, створений приблизно в 1495 році, який міг стояти, сидіти, рухати головою та піднімати візир на шоломі. Цей "робот" керувався за допомогою низки шківів, тросів та інших механічних пристроїв.

Кафедра КІТ				НАУ 23 10 36 000 ПЗ			
	ПІБ			РОЗДІЛ 1. АНАЛІЗ ОБ'ЄКТА ТА ПРЕДМЕТА ДОСЛІДЖЕННЯ	Літ.	Аркуш	Аркушів
Розроб.	Козловцев Д.С.					10	60
Керівник	Зудов О.М.				ТП-215М - 122		
Н.Контр.	Толстікова О.В.						



Рис. 1.1 Робот-солдат да Вінчі

Період 18-го і 19-го століття в історії робототехніки відзначається значними змінами, особливо завдяки промисловій революції, яка кардинально змінила виробництво та технології.

Автомати: У 18-му столітті було створено велику кількість механічних автоматів. Ці пристрої, як правило, були складними механізмами, що імітували рухи людей або тварин. Прикладами є автомати Жака де Вокансона, який створив відомого "Флейтиста" та "Травоїдну Качка", які були здатні виконувати дивовижно реалістичні рухи.

Промислова революція: Цей період був відзначений переходом від ручного виробництва до машинного. Поява парових машин та інших форм автоматизації привела до кардинальних змін в індустрії. Це також був час, коли почали з'являтися перші концепції, які нагадували сучасних роботів.

Початок 20-го століття в історії робототехніки характеризується переходом від примітивних механічних автоматів і концепцій, які були властиві 18-му та 19-му століттям, до більш сучасного розуміння роботів, значною мірою завдяки впливу наукової фантастики.

Карел Чапек - "R.U.R." (Rossum's Universal Robots, 1920): Чеський письменник Карел Чапек вперше використав термін "робот" у своїй п'єсі

"R.U.R.". В цій п'єсі роботи були антропоморфними машинами, створеними для праці, але згодом вони повстали проти людства. Термін "робот" походить від чеського слова "robota", що означає "примусова праця" або "кріпацтво".

Перші Промислові Роботи: 1960-ті та 70-ті роки позначили собою початок ери промислових роботів. Unimate (Рис. 1.2), перший промисловий робот, був введений у виробництво в 1961 році. Цей робот був гідравлічним маніпулятором, який використовувався для виконання простих, але монотонних або небезпечних завдань на збірочних лініях, зокрема в автомобільній промисловості.



Рис. 1.2 Unimate

Роботи в Космосі:

- канадський Маніпулятор Canadarm. В 1980-ті роки, з появою космічного шатла NASA, стало можливим використання роботизованих систем у космосі. Canadarm, канадський роботизований маніпулятор, був використаний на космічних шатлах для маніпуляцій з вантажами у космосі та допомоги у будівництві Міжнародної космічної станції (МКС);

- марсоходи NASA. NASA розробила та запустила кілька марсоходів, таких як Pathfinder і марсохід Sojourner (Рис. 1.3) в 1997 році. Ці місії дозволили здійснити дистанційне дослідження поверхні Марса, збираючи важливі наукові дані.

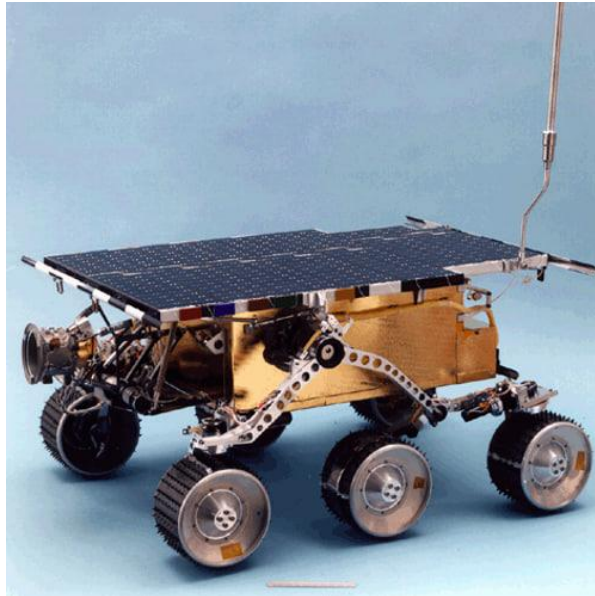


Рис. 1.3 Sojourner

21-е століття стало свідком надзвичайних проривів у робототехніці, зробивши роботів значно більш присутніми та впливовими у нашому повсякденному житті.

Розумні Домашні Помічники: Сучасні роботи, як наприклад роботи-пилососи та інші домашні помічники, стають невід'ємною частиною багатьох домогосподарств, допомагаючи в чистці, моніторингу безпеки та навіть у догляді за домашніми тваринами.

Автономні Транспортні Засоби: Розвиток автономних транспортних засобів, включаючи автомобілі, дрони для доставки та інші види транспорту, є одним з найбільш перспективних напрямків у робототехніці.

Роботизовані руки, відомі також як роботичні маніпулятори, зазнали значного розвитку від простих механічних пристроїв до складних систем, здатних до точних та складних маніпуляцій. Ось ключові моменти в їх історії:

Ранній Розвиток

20-е століття: Початок розвитку роботизованих рук сягає середини 20-го століття. Роботизовані руки були одними з перших форм роботів, що знаходили практичне застосування, особливо в промисловості.

Перший Промисловий Робот Unimate, який був представлений у 1960-х роках, мав роботизовану руку. Цей робот був здатен виконувати прості завдання, такі як зварювання чи переміщення важких предметів на лініях збору.[1]

Робототехніка в сучасному світі:

1. Промисловість: промислові роботи стали невід'ємною частиною виробничого процесу. Вони забезпечують високу точність, продуктивність та ефективність виробництва, знижуючи вартість продукції та ризик травм на робочому місці.

2. Медицина: роботизовані системи вже допомагають хірургам у проведенні складних операцій з підвищеною точністю. Також роботи активно використовуються для автоматизації лабораторних досліджень.

3. Дослідження та розвідка: роботи використовуються для дослідження глибоководних та космічних просторів, де людина може зіткнутися з небезпекою. Вони дозволяють отримувати важливу інформацію про нашу планету та космос.

4. Логістика та доставка: автоматизовані системи допомагають оптимізувати складське зберігання, сортування та доставку товарів, відповідаючи на зростаючі потреби споживачів.

5. Освіта: роботи активно використовуються для навчання студентів, допомагаючи розвивати навички в області STEM (наука, техніка, інженерія, математика).

6. Домашнє використання: роботи-пилососи, роботи для прибирання та інші допоміжні пристрої спрощують повсякденне життя, звільняючи людей від рутинних обов'язків.

7. Безпека та оборона: роботизовані системи використовуються для розмінування, розвідки та інших військових операцій, знижуючи ризик для військовослужбовців.

8. Сільське господарство: роботи допомагають автоматизувати процеси сівби, збору врожаю та догляду за рослинами, забезпечуючи стабільність та високу продуктивність харчового виробництва.

Робототехніка вже зараз змінює те, яким чином ми живемо, працюємо та взаємодіємо з навколишнім світом. Із розвитком технологій роль робототехніки тільки зростатиме, вносячи нові можливості та виклики для суспільства.[11]

1.2. Основні характеристики і принципи роботи роботизованих механічних систем

1.2.1. Механічна структура

Механічна структура роботизованих механічних систем є ключовим компонентом, який визначає функціональність, міцність, гнучкість та ефективність системи. Вона складається з різноманітних елементів, які разом створюють основу для руху та виконання завдань. Нижче наведені основні аспекти механічної структури.[3]

Ланки та суглоби (Рис. 1.4) вповнюють функцію каркасу.

Ланки - це тверді елементи, які з'єднуються суглобами. Вони формують "кістяк" робота, визначають напрямки руху та ступені свободи, можуть бути лінійними (прямий рух вздовж осі) або обертовими (рух навколо осі). Ланки зазвичай виготовляють з металів (наприклад, сталь, алюміній) або високоміцних пластиків. Їх дизайн і розміри залежать від призначення робота та його робочого навантаження.

Суглоби з'єднують різні частини механічної структури і дозволяють рух у визначених напрямках. Це можуть бути прості шарніри, поворотні або складні багатоланкові суглоби. Обертові суглоби дозволяють ланці обертатися навколо осі. Це найпоширеніший тип суглоба у роботах. Лінійні суглоби забезпечують прямолінійний рух, дозволяючи ланці переміщуватися вздовж певної осі. Складні суглоби комбінують обертові та лінійні рухи для забезпечення більшої гнучкості.

Розробка ланок та суглобів вимагає глибоких знань у галузі механіки, матеріалознавства, кінематики та динаміки. Вони є основою для створення роботизованих систем, які можуть точно та ефективно взаємодіяти з навколишнім середовищем.

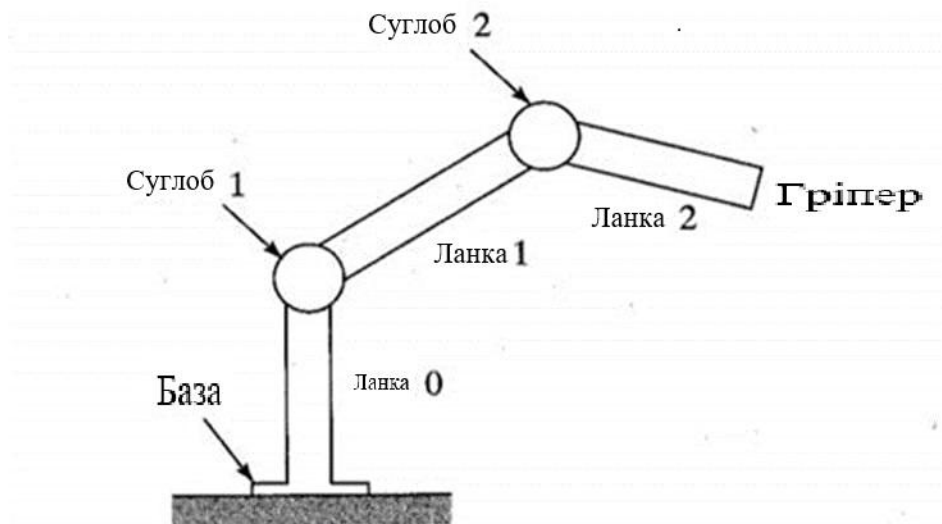


Рис. 1.4 Ланки та суглоби в роботичних системах

Актуатори (або приводи):

- електричні приводи часто використовуються через їхню точність, надійність та легкість інтеграції з електронними системами управління;
- гідравлічні та пневматичні приводи використовуються у випадках, де потрібна висока сила чи потужності, хоча вони менш точні порівняно з електричними приводами;
- п'єзоелектричні актуатори надають можливість найточнішого керування системами за рахунок зв'язку між напругою та матеріалами які використовують в основі.

Каркаси та опорні структури:

- матеріали каркаси часто виготовляються з металів (сталь, алюміній), композитних матеріалів або високоміцних пластиків. Вибір матеріалу залежить від вимог до міцності, ваги та інших експлуатаційних характеристик;
- конструкція каркаса підтримує інші компоненти та забезпечує стійкість та міцність всієї системи.

Ергономіка та дизайн:

- ергономічність: важливо, щоб механічна структура була ергономічною та відповідала вимогам конкретних завдань, що виконуються;
- дизайн: естетика також може бути важливою, особливо у споживчих або публічних застосуваннях.

Модульність та адаптивність:

- модульна конструкція дозволяє легко змінювати або оновлювати окремі компоненти для різних застосувань або під різні завдання;

Система управління:

- програмне забезпечення включає алгоритми управління, які керують роботою механічних компонентів, визначаючи їхні дії та відгук на зовнішні стимули;
- інтерфейси можуть включати мануальні контролери, комп'ютерні інтерфейси та автоматизовані системи, які дозволяють користувачам або іншим системам керувати роботизованою системою.

Сенсорна система:

- датчики використовуються для збору інформації про навколишнє середовище та стан самої системи. Включають датчики положення, швидкості, тиску, температури тощо;
- інтеграція з управлінням: інформація з датчиків використовується для адаптації та оптимізації роботи системи.

Джерела енергії:

- типи енергії можуть включати електричні акумулятори, гідравлічні або пневматичні системи, а також комбіновані джерела енергії;
- автономність- ємність джерел енергії визначає автономність системи та її здатність працювати без підзарядки або зовнішнього живлення.

Механічна структура роботизованих систем є фундаментом, на якому базуються всі інші аспекти роботи - від точності рухів до загальної продуктивності системи. Вона вимагає ретельного проектування та врахування багатьох факторів для ефективної роботи.[9]

1.2.2. Актуатори

Актуатори (приводи) – у робототехніці актуатори є ключовими компонентами, які дозволяють роботам взаємодіяти з навколишнім середовищем, забезпечуючи рух і силу. Вони є своєрідними "м'язами" робота, перетворюючи різні види енергії в механічний рух. Вибір актуаторів у робототехнічній системі значно впливає на її можливості, ефективність, точність та придатність для різних завдань. Основними актуаторами в робототехніці вважаються: Електричні, Гідравлічні, Пневматичні та П'єзоелектричні Актуатори.

Застосування в Робототехніці:

1. Маніпуляція - роботизовані руки використовують актуатори в своїх суглобах та захопленнях для маніпулювання об'єктами, виконання збіркових завдань або роботи з інструментами.

2. Пересування - у мобільних роботів актуатори використовуються для пропульсії та керування, дозволяючи роботу навігувати у своєму середовищі.

3. Сенсорика та взаємодія - деякі роботи використовують актуатори у своїх сенсорних системах для взаємодії з навколишнім середовищем, наприклад, для регулювання фокусу камери або налаштування сенсорів.

4. Контроль та інтеграція - актуатори в робототехніці часто інтегруються з сенсорами та системами управління для досягнення точного та інтелектуального руху. Ця інтеграція є ключовою для завдань, які вимагають високих рівнів точності, таких як у хірургічних роботах або автоматизованому точному виробництві.

Виклики та особливості:

- споживання енергії: актуатори можуть споживати значні кількості енергії, що є критичним аспектом, особливо для батарейних або автономних роботів;

- вага та розмір: вага та розмір актуаторів впливають на загальний дизайн та мобільність робота, особливо в додатках, таких як дослідження космосу або безпілотні літальні апарати (БПЛА);

- точність та надійність: точність та надійність актуаторів є важливими в додатках, де помилки можуть бути дорогими або небезпечними.

Електричні актуатори є одними з найбільш поширених типів актуаторів, використовуваних у різноманітних областях, від робототехніки до автоматизації виробництва. Вони перетворюють електричну енергію на механічний рух, і їхня популярність обумовлена високою точністю, контролем, ефективністю та легкістю інтеграції з електронними системами управління. Їх можна поділити на 4 типи: DC Мотори, AC Мотори, Крокові Двигуни та Сервомотори.

DC Мотори (Двигуни постійного струму) - є одними з найбільш фундаментальних та широко використовуваних типів електричних моторів. Вони перетворюють електричну енергію постійного струму на механічний рух. DC мотори мають відносно просту конструкцію, що робить їх легкими для розуміння та обслуговування. Швидкість DC моторів може бути легко контрольована зміною напруги живлення або за допомогою електронних керуючих схем. Вони забезпечують високий пусковий момент, що робить їх ідеальними для застосувань, де необхідний сильний момент з самого початку. DC мотори можна поділити на Щіткові та Безщоткові.

Щіткові DC Мотори (Рис. 1.5) складаються з ротора, статора, щіток та комутатора. Принцип роботи базується на взаємодії магнітного поля та електричного струму, що викликає обертання ротора всередині статора. Статор - нерухома частина мотора, яка створює постійне магнітне поле. Вона може містити постійні магніти або обмотки, через які проходить струм. Ротор (Якір) - обертова частина мотора, яка має обмотки, через які проходить струм. Комутатор - це електричний перемикач, який змінює напрямок струму в обмотках ротора під час обертання. Щітки забезпечують електричний контакт між стаціонарною частиною мотора та обертовим комутатором.

Коли через обмотки статора проходить струм, створюється магнітне поле. У простих DC моторах магнітне поле може бути створене за допомогою постійних магнітів. Струм подається на обмотки ротора через щітки та комутатор, в свою чергу обмотки починають створювати своє магнітне поле. Магнітне поле ротора взаємодіє з магнітним полем статора. Ця взаємодія створює силу, яка змушує ротор обертатися. Комутатор змінює напрямок струму в обмотках ротора кожного півоберту. Це необхідно для підтримання постійного обертання ротора, оскільки без зміни напрямку струму сила, яка змушує ротор обертатися, змінила б свій напрямок і зупинила обертання.

Однією з ключових переваг щіткових DC моторів є їх високий пусковий момент, що робить їх ідеальними для застосувань, де потрібен сильний момент з самого початку. Швидкість щіткових DC моторів може бути легко контрольована шляхом зміни напруги живлення. Перевагами є проста конструкція і сильний момент, недоліками – знос щіток і електромагнітні завади.

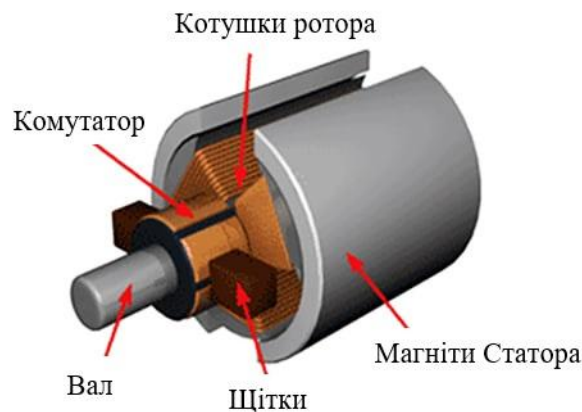


Рис. 1.5 Щітковий DC Мотор

BLDC або безщіткові DC мотори (Рис. 1.6) є сучасною альтернативою традиційним щітковим DC моторам. Вони використовують електронне управління для комутації струму, уникаючи механічного контакту щіток з комутатором. Це призводить до збільшення ефективності, надійності та тривалості життя мотора. Статор - це нерухома частина мотора, яка містить

обмотки. У BLDC моторах статор генерує магнітне поле. Ротор - обертова частина мотора, яка містить постійні магніти. Вона обертається відповідно до магнітного поля, створеного статором. Електронний Контролер - використовується для управління напрямком струму в обмотках статора. Контролер замінює механічний комутатор та щітки, які використовуються в щіткових DC моторах. Сенсори Положення Ротора - часто використовуються для визначення точного положення ротора, щоб контролер міг правильно комутувати струм.

Коли через обмотки статора проходить струм, створюється обертове магнітне поле. Напрямок струму в обмотках змінюється електронним контролером. Магнітне поле, створене статором, взаємодіє з магнітним полем постійних магнітів на роторі. Ця взаємодія створює силу, яка змушує ротор обертатися. Електронний контролер постійно змінює напрямок струму в обмотках статора, щоб підтримувати обертання ротора. Сенсори положення ротора (якщо вони є) надають контролеру інформацію про положення ротора, щоб він міг правильно синхронізувати ці зміни.

Перевагами є зменшення зносу, вища ефективність, краще управління швидкістю та моментом обертання, тиха робота і краще тепловідведення. Недоліками є вища вартість та складність контролера.



Рис. 1.6 Безщіткові DC Мотори (BLDC)

Крокові двигуни (Рис. 1.7) є електричними двигунами, які прецизійно контролюють кут обертання та позицію. Вони широко використовуються в автоматизації, робототехніці, 3D-друку та інших застосуваннях, де потрібен точний контроль руху. На відміну від багатьох інших типів двигунів, крокові двигуни не вимагають зовнішніх сенсорів для визначення положення вала, оскільки їх положення відоме на основі кількості поданих кроків. Крокові двигуни можуть утримувати вал у фіксованому положенні, коли на них подається струм, навіть без руху.

Крокові двигуни містять кілька електромагнітних котушок, які управляються послідовно. Ці котушки створюють магнітне поле, яке взаємодіє з постійними магнітами або залізними зубцями на роторі. Контролер двигуна послідовно включає та вимикає котушки, створюючи магнітні поля, які змушують ротор обертатися. Кожне включення або вимикання котушки переміщує ротор на один крок. Існують різні типи крокових двигунів, включаючи двигуни з постійними магнітами, гібридні крокові двигуни та двигуни з перемінним реактансом. Гібридні крокові двигуни забезпечують вищу точність та кращі характеристики.

Перевагами є точне позиціонування, простота управління та тримання позиції. Недоліками – обмежена швидкість, втрата кроків при перевантаженні або неправильному управлінні, а також резонанси та вібрації які можуть виникати при певних швидкостях.

Застосовуються там, де потрібне точне розташування: 3D-друк, ЧПУ-машини, робототехніка, автоматизоване виробництво.

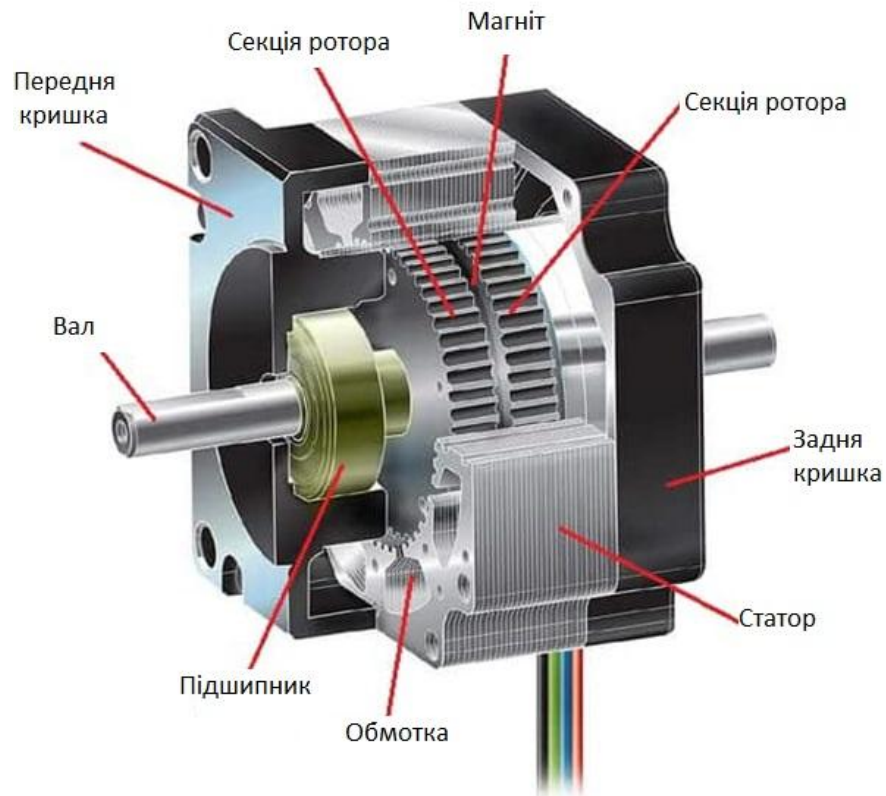


Рис. 1.7 Кроковий Мотор

Сервомотори (Рис. 1.8) є спеціалізованим типом електричного двигуна, який використовується для точного контролю положення, швидкості та кута обертання. Вони широко застосовуються в робототехніці, моделюванні, автоматизованих виробничих системах та інших областях, де потрібен точний контроль руху.

Основні Компоненти Сервомотора:

1. Двигун. Зазвичай це невеликий DC або безщітковий DC мотор.
2. Система управління (контролер) використовується для контролю положення вала двигуна.
3. Потенціометр або енкодер використовується для вимірювання фактичного положення вала та надання цієї інформації контролеру.
4. Редуктор знижує швидкість обертання вала та збільшує крутний момент.
5. Схема зворотного зв'язку забезпечує інформацію про поточне положення вала до контролера.

Сервомотор отримує керуючий сигнал, який вказує бажане положення вала. Потенціометр або енкодер, з'єднаний з валом двигуна, визначає поточне положення вала. Контролер порівнює бажане положення вала з його поточним положенням і визначає, чи потрібно змінити положення вала. Якщо поточне положення вала не відповідає бажаному, контролер відправляє сигнал на двигун, щоб змінити його положення. Двигун обертається, змінюючи положення вала до тих пір, поки потенціометр або енкодер не підтвердить, що вал досяг бажаного положення.

Перевагами є висока точність, ефективність, швидка реакція на команди. З недоліків можна згадати обмежену витривалість, високу вартість та складність управління.



Рис. 1.8 Сервомотор

Гідравлічний актуатор, зображений на рис. 1.9, є важливим компонентом у багатьох промислових та інженерних системах, де потрібно здійснювати потужні та контрольовані рухи. Вони використовують гідравлічну рідину для передачі енергії та виконання роботи, забезпечуючи високий крутний момент або силу при відносно невеликих розмірах. Гідравлічні актуатори працюють на принципі Паскаля, який стверджує, що тиск у замкненій системі рідини передається однаково у всіх напрямках

Гідравлічний Циліндр - це основний компонент, який перетворює гідравлічну енергію на механічну роботу. Циліндр складається з поршня всередині циліндричної труби. Поршень розділяє циліндр на дві камери і рухається вперед і назад, коли гідравлічна рідина надходить або відводиться.

Гідравлічна Рідина є зазвичай мінеральним маслом, яке передає тиск від насоса до поршня. Насос забезпечує рух рідини під тиском у системі. Клапани керують напрямком та потоком гідравлічної рідини. Коли гідравлічна рідина під тиском надходить у циліндр, вона штовхає поршень, який, у свою чергу, виконує механічну роботу. Зміна напрямку руху рідини змінює напрямок руху поршня.



Рис. 1.9 Гідравлічний актуатор

Гідравлічні актуатори можна поділити на декілька видів:

1. Лінійні гідравлічні актуатори:

- один з найбільш розповсюджених типів актуаторів у робототехніці;
- вони складаються з гідравлічного циліндра з поршнем всередині, який рухається вздовж осі циліндра;
- лінійні актуатори забезпечують потужні та точні поштовхи або тяги, що використовуються для переміщення або піднімання важких об'єктів.

2. Ротаційні гідравлічні актуатори:

- ці актуатори виробляють обертальний рух;
- вони часто використовуються в робототехніці для забезпечення обертання або руху частин робота, наприклад, у маніпуляторах або обертових з'єднаннях.

3. Телескопічні гідравлічні актуатори:

- ці актуатори мають декілька циліндрів різного діаметру, вкладених один в одного;

- вони забезпечують велику довжину ходу поршня з компактних розмірів, що корисно для завдань, де потрібен розширений або змінний діапазон руху.

Ці актуатори обираються на основі потреб конкретної робототехнічної системи, враховуючи необхідну силу, точність, швидкість реакції, діапазон руху та інші фактори. Важливо відзначити, що гідравлічні актуатори в робототехніці вибираються з урахуванням балансу між їхньою потужністю та точністю, а також з огляду на конкретні умови експлуатації. Гідравлічні системи ідеально підходять для задач, де потрібні великі сили або крутний момент, наприклад, у промислових роботах або в важкій автоматизації. Вони можуть ефективно переміщати великі вантажі, забезпечувати високий рівень контролю та точності, що критично важливо в деяких робототехнічних застосуваннях.

Пневматичний актуатор, зображений на рис. 1.10, є механізмом, який використовує стиснене повітря для створення руху або сили. Вони перетворюють енергію стисненого повітря на механічний рух, що може бути лінійним або обертальним. Типовий пневматичний актуатор складається з циліндра, поршня, штока поршня, клапанів для управління потоком повітря та з'єднань для підключення до пневматичної системи. Пневматичні актуатори можуть створювати як лінійний, так і обертальний рух. Лінійні актуатори переміщують поршень вздовж осі, тоді як ротаційні актуатори забезпечують обертальний рух.

Принцип роботи: повітря стискається за допомогою компресора та подається до актуатора через систему пневматичних шлангів та клапанів. У більшості пневматичних актуаторів є поршень всередині циліндра. Коли стиснене повітря потрапляє в циліндр, воно штовхає поршень, створюючи рух. Повітря може бути випущене або використано для переміщення поршня назад до його початкового положення.

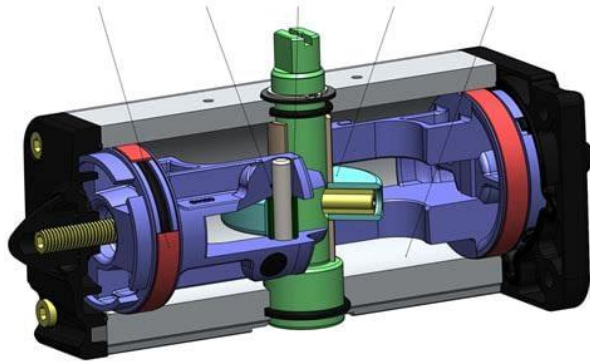


Рис. 1.10 Пневматичний актуатор

Типи Пневматичних Актуаторів:

1. Лінійні актуатори: створює прямолінійний рух. Вони використовуються у виробничих системах, автоматизації збірки та інших застосуваннях, де потрібен лінійний рух.

2. Ротаційні актуатори: створюють обертальний рух. Ці актуатори зазвичай використовуються для управління клапанами або виконання поворотних рухів у різних машинах.

3. Гріппери та маніпулятори: спеціалізовані пневматичні пристрої для захоплення та переміщення об'єктів. Вони часто використовуються в автоматизованих збірочних лініях та робототехніці.

П'єзоелектричні Актуатори, приклад котрих зображений на рис. 1.11, використовують п'єзоелектричний ефект для створення руху або сили. Цей тип актуатора знаходить застосування у ситуаціях, де потрібна висока точність та контроль над дуже малими переміщеннями.

П'єзоелектричний ефект полягає у створенні електричного заряду в кристалічних матеріалах (наприклад, кварці, цирконаті свинцю, титанаті свинцю) під дією механічного тиску. Ці матеріали можуть змінювати свої розміри (здійснювати механічне розширення або стиснення), коли до них застосовується електричне поле. Цей аспект п'єзоелектричного ефекту використовується в п'єзоелектричних актуаторах. Зміна форми матеріалу пропорційна прикладеному напругу. Ця властивість дозволяє точно

контролювати переміщення актуатора, варіюючи величину та полярність напруги.



Рис. 1.11 П'єзоелектричний актуатор

Типи П'єзоелектричних Актуаторів:

1. Стекові (Stack) актуатори:

- складаються з декількох шарів п'єзоелектричного матеріалу, скомпонованих один на одного (у формі "стеку");
- при прикладенні напруги кожен шар трохи деформується, а загальна деформація сумується, що веде до більшого зміщення;
- часто використовуються в прецизійних позиціонуючих системах та в додатках, де потрібно невелике переміщення з високою силою.

2. Стрип-актуатори:

- це довгі, тонкі смужки п'єзоматеріалу, які згинаються або скручуються при прикладенні напруги;
- вони можуть виробляти відносно великі переміщення на великій відстані, але з меншою силою, ніж стекові актуатори;
- застосовуються в таких додатках, як контроль вібрацій або у дуже компактних пристроях.

3. Трубчасті Актуатори:

- мають форму трубки, з п'єзоелектричним матеріалом, нанесеним або інтегрованим уздовж поверхні;

- такі актуатори забезпечують обертальний або лінійний рух і можуть бути використані для точного управління потоками в мікрофлюїдних системах.

У виборі між електричними, гідравлічними та пневматичними приводами враховуються такі фактори, як необхідна точність, потреба у силі чи потужності, доступність джерел енергії та загальні вимоги до системи. Кожен тип привода має свої особливості та оптимально підходить для певних застосувань у робототехніці.

Підсумовуючи, актуатори є фундаментальними для функціональності роботів, дозволяючи їм рухатися, взаємодіяти та виконувати завдання в різних середовищах. Постійний прогрес у технології актуаторів продовжує розширювати можливості та застосування робототехніки у численних галузях.

1.2.3. Системи управління роботичних систем

Система управління в роботичних системах відіграє ключову роль, оскільки вона відповідає за координацію, керування та моніторинг роботи робота.[7]

Ключові Компоненти:

- мікроконтролери або процесори використовуються для обробки даних та виконання програмного коду, який керує роботом;
- сенсорна система включає датчики, які збирають інформацію про навколишнє середовище та стан самого робота;
- інтерфейси управління можуть включати мануальні контролери, сенсорні панелі, голосове управління або дистанційне керування через комп'ютерні програми.

Функції системи управління:

- координація рухів синхронізація рухів механічних частин, таких як маніпулятори, ходові частини тощо;

- обробка даних: аналіз даних від сенсорів для визначення дій, які повинен виконати робот;
- виконання завдань: управління завданнями, програмоване чи динамічне, на основі отриманих даних та заданих алгоритмів.
- взаємодія з користувачем: забезпечення інтерфейсу для взаємодії людини з роботом, включаючи введення команд та відображення інформації.

Система управління є "мозком" роботичної системи, визначаючи її здатність адаптуватися до завдань, реагувати на зміни у навколишньому середовищі та взаємодіяти з користувачами. Ефективна система управління є ключовою для досягнення високої продуктивності, точності та надійності в роботичних системах.

Мікроконтролер - це компактний інтегрований схемний пристрій, що містить процесор (ЦПУ), пам'ять (RAM, ROM/Flash) та вхідно-вихідні порти на одному чипі. Вони виконують різні завдання, включаючи обробку даних, контроль різних пристроїв, збір інформації з сенсорів тощо. Приклад мікроконтролерів можна побачити на рис. 1.12

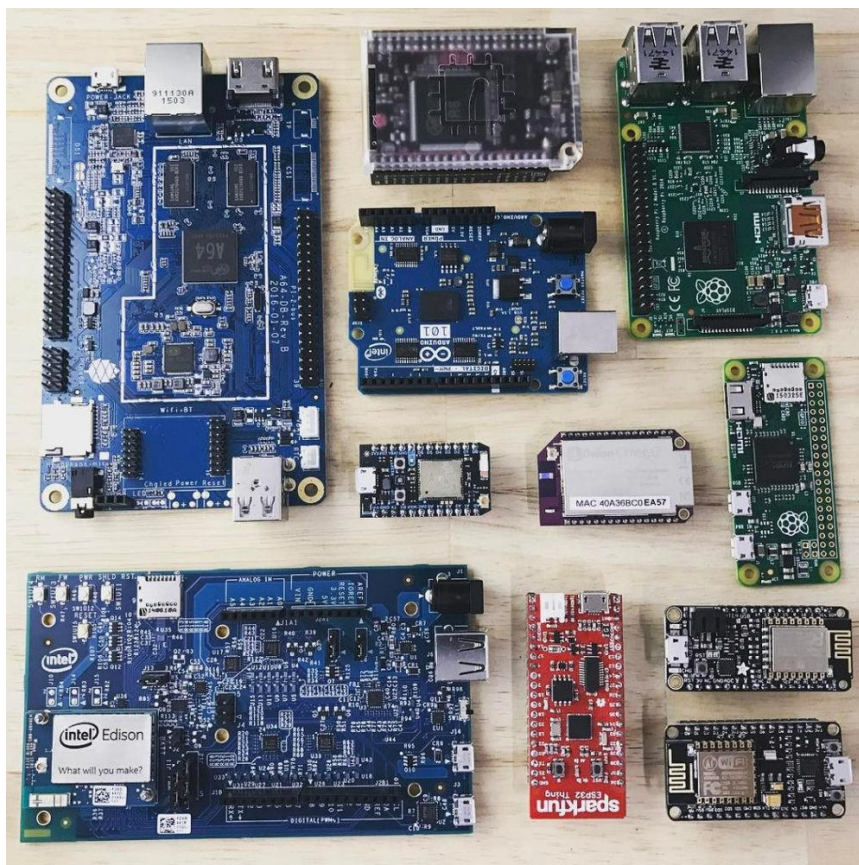


Рис. 1.12 Мікроконтролери

Основні Характеристики

- ЦПУ: серце мікроконтролера, яке виконує програмні інструкції;
- пам'ять: включає оперативну (RAM) та постійну (Flash або EEPROM) пам'ять для зберігання програмного коду та даних;
- інтерфейси: надають можливість підключення до різних зовнішніх пристроїв, наприклад, сенсорів, моторів, дисплеїв тощо;
- аналогово-цифрові перетворювачі (АЦП): для обробки аналогових сигналів;
- таймери: для вимірювання часових інтервалів та генерації періодичних подій;
- система управління живленням контролює споживану потужність, забезпечуючи енергоефективність;
- захисні механізми, такі як захист від перенапруги;
- вхідно-вихідні порти (I/O).

Мікроконтролери програмуються за допомогою різних мов програмування (наприклад, C/C++), використовуючи інтегровані середовища розробки (IDE).

Популярні Платформи

- Arduino: легкий у використанні мікроконтролер, популярний серед хобістів та у освітніх цілях;
- Raspberry Pi (у випадках, коли використовується в режимі мікроконтролера): здатний виконувати більш складні завдання, включаючи обробку відео та запуск повноцінної операційної системи;
- ESP8266/ESP32: мікроконтролери з вбудованою Wi-Fi та Bluetooth функціональністю, підходять для IoT-проектів.

Застосування:

- робототехніка: для управління роботами та автоматизованими системами;
- інтернет речей (IoT) управління розумними пристроями;

- вбудовані системи: у сферах від промисловості до споживчої електроніки;

Мікроконтролери є важливими через їхню здатність виконувати специфічні контрольні функції в широкому спектрі застосувань. Їх компактний розмір, низька вартість, енергоефективність та гнучкість роблять їх незамінними в сучасних технологіях.

Центральний процесор у мікроконтролерах є основним виконавчим компонентом, який керує всіма операціями та обробкою даних у системі.

Основні Функції ЦПУ

- обробка інструкцій: виконує машинний код або інструкції, збережені в пам'яті мікроконтролера;

- управління даними: контролює потоки даних між різними частинами мікроконтролера, включаючи пам'ять, вхідно-вихідні пристрої та інші периферійні компоненти.

Центральний процесор (ЦПУ) мікроконтролера виконує обробку інструкцій та управління даними за допомогою декількох ключових процесів.

Коли програма завантажується в мікроконтролер, вона зберігається в його постійній пам'яті, зазвичай у флеш-пам'яті. Під час роботи мікроконтролера, ЦПУ послідовно читає інструкції з цієї пам'яті та виконує їх.

Архітектура ЦПУ

Давайте розглянемо більш детально архітектуру ЦПУ та їх взаємодію.

Архітектура фон Неймана та Гарварда: в архітектурі фон Неймана інструкції та дані зберігаються в одній і тій же пам'яті. Це означає, що інструкції та дані використовують однакові шляхи для доступу до пам'яті. Через спільне використання пам'яті для інструкцій та даних, доступ до них відбувається послідовно. Це може призвести до затримок, оскільки ЦПУ може чекати доступу до пам'яті. Архітектура фон Неймана зазвичай простіша у реалізації, що робить її популярною для багатьох застосувань, особливо в минулому. Через спільне використання пам'яті для інструкцій та даних, системи на основі архітектури фон Неймана більш вразливі до вірусів та

шкідливого коду. В архітектурі Гарварда інструкції та дані зберігаються окремо. Це означає, що існують окремі шляхи доступу до пам'яті для інструкцій та даних, що збільшує швидкість обробки. Оскільки інструкції та дані зберігаються окремо, ЦПУ може одночасно зчитувати інструкції та працювати з даними, що зменшує час очікування і підвищує продуктивність. Архітектура Гарварда складніша для реалізації через необхідність управління двома окремими системами пам'яті. Розділення пам'яті для інструкцій та даних може збільшити безпеку, оскільки шкідливий код менш імовірно зможе вплинути на обидва типи пам'яті одночасно. Архітектура фон Неймана часто використовується в загального призначення комп'ютерах та простих мікроконтролерах. Архітектура Гарварда зазвичай зустрічається в спеціалізованих системах та високопродуктивних мікроконтролерах, де важлива висока швидкість обробки. Обидві архітектури мають свої переваги та недоліки. Вибір між ними залежить від конкретних вимог до системи, включаючи швидкість, складність, вартість та цілі застосування.

Лічильник програм (РС) є ключовим компонентом ЦПУ мікроконтролерів та комп'ютерів загалом. Цей регістр має важливу роль у керуванні потоком виконання програм. Лічильник програм зберігає адресу (або вказівник) наступної інструкції, яка має бути виконана ЦПУ. Кожного разу, коли інструкція виконується, лічильник програм оновлюється, щоб вказувати на наступну інструкцію. При запуску або перезавантаженні системи, лічильник програм зазвичай ініціалізується на певну стартову адресу, з якої починається виконання програми. Під час нормального виконання програми, лічильник програм зазвичай автоматично збільшується на кількість байтів, яка відповідає довжині виконаної інструкції, щоб вказувати на наступну інструкцію. У випадку виконання інструкцій управління потоком, таких як умовні або безумовні переходи, виклики функцій чи повернення з них, значення лічильника програм змінюється відповідно до адреси цільової інструкції. Розмір лічильника програм залежить від архітектури ЦПУ. Наприклад, в 32-бітній архітектурі лічильник програм також буде 32-бітним, що дозволяє адресувати до 4 ГБ пам'яті. При

обробці переривань, поточне значення лічильника програм зберігається, щоб після обробки переривання можна було продовжити виконання програми з того місця, де воно було перервано. У мікроконтролерах, де ресурси обмежені, і кожен байт пам'яті на рахунку, ефективне управління лічильником програм є критично важливим. Це дозволяє оптимізувати використання пам'яті та забезпечує ефективне виконання програм.

Регістри використовуються для тимчасового зберігання даних та інструкцій, що потрібні для поточних операцій. Вони забезпечують швидкий доступ до даних, на відміну від більш повільної основної пам'яті. Залежно від архітектури мікроконтролера, кількість та розмір регістрів можуть значно варіюватися. Наприклад, в 8-бітних мікроконтролерах регістри будуть мати розмір 8 біт, у той час як в 32-бітних - 32 біти. Деякі мікроконтролери можуть мати спеціалізовані регістри для конкретних завдань, таких як управління периферією, обробка переривань тощо.

Регістри можна поділи на типи:

- регістри загального призначення (General Purpose Registers, GPRs): ці регістри можуть використовуватися для різноманітних завдань, таких як тимчасове зберігання даних, проміжних результатів обчислень, адрес пам'яті тощо. В залежності від архітектури, їх кількість та розмір можуть варіюватися;

- лічильник програм (Program Counter, PC): цей регістр зберігає адресу наступної інструкції, яка має бути виконана;

- регістр стану (Status Register, Flag Register): він містить прапорці (флаги), які відображають результати останніх обчислень (наприклад, нуль, переповнення, від'ємне значення тощо);

- регістр інструкцій (Instruction Register, IR): зберігає поточну інструкцію, яка виконується;

- регістри базових адрес (Base Registers): використовуються для зберігання базових адрес, які є вихідними точками для адресації пам'яті;

- реєстри вказівників (Pointer Registers): включають вказівники стеку (Stack Pointer) та інші вказівники, які використовуються для роботи зі стеком та іншими структурами даних;

- індексні реєстри (Index Registers): використовуються для підтримки індексованої або базово-індексованої адресації пам'яті.

Арифметико-логічний блок (ALU) відповідає за виконання всіх математичних та логічних операцій. ALU взаємодіє з реєстрами для обробки даних. ALU виконує базові арифметичні операції, такі як додавання, віднімання, множення, ділення. Це може включати як цілочисельні, так і операції з плаваючою комою, залежно від можливостей мікроконтролера. Виконує логічні операції, такі як І (AND), АБО (OR), НЕ (NOT), XOR (виключне АБО), зсуви бітів та інші бітові операції. ALU також відповідає за встановлення різних прапорців у реєстрі стану (наприклад, прапорець переповнення, прапорець нуля, прапорець від'ємного значення), які використовуються для вказівки результатів операцій та для управління потоком програми. ALU отримує дані з реєстрів ЦПУ. Це можуть бути дані з реєстрів загального призначення, або спеціалізовані реєстри. Виконує обрану операцію на основі інструкцій, які надходять від блоку управління (Control Unit). Результати операцій повертаються назад у реєстри ЦПУ або використовуються для встановлення прапорців у реєстрі стану. Швидкість та ефективність ALU є вирішальними для загальної продуктивності мікроконтролера. У простіших мікроконтролерах ALU може бути обмежений базовими операціями, тоді як у більш складних системах він може підтримувати широкий спектр арифметичних та логічних функцій. У мікроконтролерах, призначених для вбудованих систем, ALU часто оптимізується для мінімізації розміру та споживання енергії.

Блок Управління (Control Unit, CU) в ЦПУ мікроконтролерів відіграє ключову роль у координації та керуванні всіма операціями в системі. Він функціонує як мозок процесора, керуючи потоком даних між різними компонентами та визначаючи, які операції мають бути виконані та коли. Блок управління декодує інструкції, отримані від пам'яті. Він перетворює їх на

серію сигналів, які керують іншими частинами ЦПУ та системи. CU відповідає за визначення порядку виконання інструкцій, забезпечуючи, щоб операції виконувалися в правильній послідовності. Він керує потоком даних між різними частинами мікроконтролера, такими як арифметико-логічний блок (ALU), реєстри та пам'ять. CU генерує таймінгові сигнали, які синхронізують роботу різних компонентів ЦПУ. Інструкції зазвичай зчитуються з пам'яті та завантажуються в реєстр інструкцій. Блок управління аналізує інструкцію та перетворює її на набір команд та сигналів. На основі декодованої інструкції, CU генерує відповідні сигнали для керування іншими частинами ЦПУ. Він керує виконанням операцій, направляючи дані до ALU або інших компонентів та отримуючи результати. Блок управління може бути реалізований як через мікропрограмоване керування, де логіка керування зберігається у вигляді мікропрограм, так і через жорстке керування, де логіка керування реалізована на апаратному рівні. Складність блоку управління залежить від архітектури ЦПУ та варіюється від простих до високо складних систем. У мікроконтролерах, особливо в тих, що використовуються в вбудованих системах, блок управління часто оптимізується для ефективності та енергозбереження.

Шина даних та адресна шина забезпечують транспортування інформації від ЦПУ до пам'яті та назад. Вони є основними каналами для передачі інструкцій, адрес та даних. Ширина шини даних визначає кількість бітів, які можуть бути передані одночасно. Наприклад, 8-бітна шина може передавати 8 бітів даних за один такт. Ширина шини безпосередньо впливає на пропускну спроможність, тобто на кількість даних, які можуть бути передані за одиницю часу. Шина даних зазвичай є двонаправленою, що дозволяє передавати дані як від ЦПУ до пам'яті або периферійних пристроїв, так і назад. Адресна шина використовується для вказівки адреси пам'яті або периферійного пристрою, з яким ЦПУ хоче взаємодіяти. Ширина адресної шини визначає максимальну кількість пам'яті, яку може адресувати ЦПУ. Наприклад, 16-бітна адресна шина може адресувати до 2^{16} різних адрес, що дорівнює 64 КБ. Адресна шина зазвичай є однонаправленою, оскільки вона

використовується лише для вказівки адреси в пам'яті або іншому пристрої. Через адресну шину ЦПУ вказує, з якою конкретно локацією пам'яті або периферійним пристроєм воно хоче взаємодіяти. Для правильної роботи системи важлива синхронізація між шиною даних та адресною шиною, яка часто забезпечується за допомогою таймінгових сигналів від ЦПУ.

Кеш-пам'ять призначена для прискорення доступу до даних та інструкцій, які часто використовуються. Це швидкодіюча пам'ять, розташована між основною оперативною пам'яттю (RAM) та процесором, яка зберігає копії даних з найчастіше використовуваних локацій пам'яті. Кожен з цих компонентів грає важливу роль в ефективності та швидкості обробки ЦПУ, і взаємодія між ними є ключовою для забезпечення оптимальної продуктивності мікроконтролера. Кеш-пам'ять значно швидша за основну оперативну пам'ять. Це дозволяє ЦПУ швидко отримувати доступ до найбільш використовуваних даних та інструкцій. Зазвичай кеш-пам'ять має набагато менший розмір порівняно з основною оперативною пам'яттю, оскільки вона дорожча та використовує більш швидкодіючі технології. Часто кеш-пам'ять поділяється на декілька рівнів (L1, L2, L3), де L1 є найшвидшим і найменшим, а L3 - повільнішим, але більшим.

Процес читання та виконання інструкцій:

Коли мікроконтролер запускається, ЦПУ починає читати інструкції програми, яка зберігається у флеш-пам'яті мікроконтролера. Ці інструкції зазвичай завантажуються в послідовному порядку.

Процес зчитування інструкцій з флеш-пам'яті ЦПУ мікроконтролера включає наступні кроки:

1. Адресація: ЦПУ визначає адресу наступної інструкції для виконання. Це зазвичай відбувається за допомогою лічильника програм (program counter). Після кожної інструкції, лічильник програм автоматично інкрементується (збільшується на величину, відповідну розміру інструкції), щоб вказувати на адресу наступної інструкції.

2. Запит до пам'яті: за допомогою адреси, що зберігається в лічильнику програм, ЦПУ звертається до флеш-пам'яті мікроконтролера для отримання інструкції, розташованої за вказаною адресою.

3. Читання даних: флеш-пам'ять відповідає на запит, надсилаючи дані інструкції назад до ЦПУ. Цей процес може включати взаємодію через шину даних.

4. Буферизація інструкції: після зчитування, інструкція тимчасово зберігається в одному з внутрішніх регістрів ЦПУ або в інструкційному буфері. Це забезпечує швидкий доступ до інструкції для декодування.

Кожна інструкція, яка читається з пам'яті, декодується ЦПУ для визначення необхідних дій. Цей процес включає розпізнавання типу операції (наприклад, арифметичної, логічної, управління периферією).

Процес декодування інструкції у ЦПУ мікроконтролера, після отримання інструкції, відбувається так:

1. Передача інструкції до декодера: отримана інструкція передається з регістра або буфера в декодер інструкцій.

2. Розбір бітів інструкції: декодер аналізує біти інструкції. Кожна частина бінарного коду інструкції має своє значення (наприклад, код операції, цільові регістри, значення операндів).

3. Визначення операції: декодер ідентифікує тип операції (наприклад, арифметичну, логічну, операцію зчитування/запису в пам'ять).

4. Визначення операндів: визначається, з якими даними має бути виконана операція (наприклад, дані з конкретних регістрів).

5. Генерація контрольних сигналів: на основі декодованої інструкції генеруються контрольні сигнали, які вказують іншим частинам ЦПУ чи мікроконтролера, що саме робити.

Цей процес декодування є критично важливим для правильного виконання програмного коду, оскільки він перекладає бінарні інструкції на конкретні дії, які має виконати мікроконтролер.

Після декодування, ЦПУ виконує інструкцію. Це може включати математичні обчислення, зміну стану вхідно-вихідних портів, зчитування або запис даних у пам'ять, тощо.

1. Передача контрольних сигналів: на основі декодованої інструкції, ЦПУ генерує специфічні контрольні сигнали, які направляються до відповідних блоків мікроконтролера.

2. Активація виконавчих блоків: залежно від типу інструкції, активуються різні виконавчі блоки в ЦПУ, такі як арифметико-логічний блок (ALU) для обчислень, блоки управління пам'яттю для доступу до пам'яті тощо.

3. Виконання операцій: ЦПУ виконує визначену операцію. Наприклад, це може бути арифметична операція в ALU, запис або зчитування даних із пам'яті, або управління вводом/виводом.

4. Запис результатів: результати операції записуються в певний регістр або назад у пам'ять, залежно від інструкції.

5. Оновлення лічильника програм: після завершення інструкції, лічильник програм (PC) оновлюється для вказівки на адресу наступної інструкції.

Кожен з цих кроків виконується під керівництвом ЦПУ, що забезпечує послідовне та точне виконання програмного коду.

Після виконання інструкції, ЦПУ переходить до наступної інструкції в програмі. Цей процес повторюється, поки мікроконтролер працює.

Пам'ять у мікроконтролерах є ключовим компонентом, який визначає їхню здатність зберігати та обробляти дані. Основні типи пам'яті в мікроконтролерах - це оперативна пам'ять (RAM) та постійна пам'ять (Flash або EEPROM).

Оперативна пам'ять (RAM) використовується для тимчасового зберігання даних та інструкцій, які активно використовуються мікроконтролером під час виконання програми. RAM - це волатильна пам'ять, що означає, що вона зберігає дані тільки під час подачі живлення. Коли живлення відключається, всі дані в RAM втрачаються.

Використовується для зберігання змінних, проміжних результатів обчислень, буферів для вводу/виводу тощо. ОП ділиться на два типи: SRAM (Static RAM) та DRAM (Dynamic RAM)

SRAM (Static RAM) є одним з основних типів оперативної пам'яті, який використовується в мікроконтролерах.

Основні Характеристики

1. Волатильність: SRAM є волатильною пам'яттю, що означає, що вона зберігає дані тільки під час подачі електричного живлення. При відключенні живлення всі дані втрачаються.

2. Конструкція ячейки пам'яті: кожна ячейка SRAM складається з шести транзисторів, які формують два перекреснені інвертори. Ця конструкція дозволяє ячейці зберігати біт інформації (0 або 1) без необхідності постійного оновлення.

3. Швидкість доступу: SRAM забезпечує дуже швидкий доступ до даних, що робить її ідеальною для використання як кеш-пам'ять у мікропроцесорах та мікроконтролерах.

4. Енергоспоживання: SRAM споживає менше енергії при читанні або записі порівняно з DRAM, але її статичне споживання енергії (споживання у режимі очікування) вище.

Застосування в Мікроконтролерах:

1. Кеш-пам'ять: часто використовується як кеш-пам'ять для мікропроцесорів та мікроконтролерів, забезпечуючи швидкий доступ до часто використовуваних даних та інструкцій.

2. Буферизація та тимчасове зберігання: використовується для тимчасового зберігання даних, таких як буфери вводу/виводу, стеки, змінні та проміжні результати обчислень.

3. Системи реального часу: її швидкість робить її ідеальною для систем реального часу, де швидкий доступ до даних є критично важливим.

Технічні аспекти:

- розмір: SRAM у мікроконтролерах може варіюватися від декількох кілобайт до декількох мегабайт, залежно від призначення та вартості мікроконтролера.

- інтерфейс: SRAM може бути інтегрована безпосередньо в мікроконтролер або підключена через зовнішній інтерфейс.

DRAM (Dynamic Random Access Memory) є іншим важливим типом оперативної пам'яті, який використовується в мікроконтролерах та інших комп'ютерних системах.

Основні Характеристики:

1. Волатильність: як і SRAM, DRAM є волатильною пам'яттю, що означає, що вона зберігає дані тільки під час подачі електричного живлення.

2. Конструкція ячейки пам'яті: в DRAM кожна ячейка пам'яті складається з одного транзистора та одного конденсатора. Конденсатор зберігає біт даних у формі електричного заряду, а транзистор діє як перемикач для контролю доступу до конденсатора.

3. Оновлення пам'яті: основна відмінність DRAM від SRAM полягає в необхідності регулярного оновлення. Конденсатори втрачають заряд з часом, тому дані потрібно періодично оновлювати (перезаряджати), щоб уникнути втрати інформації.

4. Швидкість доступу: швидкість доступу до DRAM повільніша порівняно з SRAM. Це пов'язано з часом, необхідним для читання та відновлення заряду в кожній ячейці.

Хоча DRAM частіше зустрічається в комп'ютерах та серверах, де потрібна велика кількість пам'яті, вона також може використовуватися в деяких мікроконтролерах, особливо в тих, що вимагають великої кількості оперативної пам'яті для складних завдань або обробки великих даних.

Технічні Аспекти

- розмір: у мікроконтролерах, які використовують DRAM, ємність може варіюватися від декількох мегабайт до більших об'ємів, залежно від потреби;

- інтерфейс: DRAM часто підключається через зовнішній інтерфейс, оскільки вона займає більше фізичного простору.

Постійна пам'ять (Flash та EEPROM) Використовується для зберігання програмного коду та даних, які не повинні втрачатися при відключенні живлення.

Flash-пам'ять у мікроконтролерах є важливим компонентом, який забезпечує зберігання програмного коду та даних, що не втрачаються при відключенні живлення.

Основні Характеристики

1. Неволатильність: головна особливість Flash-пам'яті полягає у її неволатильності, тобто вона зберігає дані навіть після відключення живлення.

2. Типи Flash-пам'яті: існують два основних типи Flash-пам'яті - NOR та NAND. NOR Flash забезпечує швидкий доступ до даних, але має меншу щільність зберігання та вищу вартість за біт. NAND Flash має вищу щільність зберігання та нижчу вартість, але швидкість доступу до даних повільніша.

3. Принцип роботи: Flash-пам'ять використовує транзистори з плаваючим затвором для зберігання бітів даних. Записування даних відбувається шляхом зміни електричного заряду в плаваючому затворі, а читання - шляхом визначення стану заряду.

4. Цикли Запису/Стирання: Flash-пам'ять має обмежену кількість циклів запису/стирання. Після певної кількості циклів пам'ять може почати втрачати свою надійність.

Застосування в мікроконтролерах:

1. Зберігання програмного коду: Flash-пам'ять часто використовується для зберігання постійного програмного коду мікроконтролера.

2. Зберігання конфігураційних даних: використовується для зберігання параметрів конфігурації, які можуть змінюватися в процесі роботи.

3. Оновлення програмного забезпечення: дозволяє оновлювати програмне забезпечення мікроконтролера без необхідності фізичного втручання.

Технічні Аспекти:

- розмір: ємність Flash-пам'яті в мікроконтролерах може варіюватися від декількох кілобайт до декількох мегабайт;

- інтерфейс: зазвичай інтегрована безпосередньо в мікроконтролер, але також може бути підключена через зовнішній інтерфейс.

EEPROM (Electrically Erasable Programmable Read-Only Memory) є важливим типом пам'яті, який використовується в мікроконтролерах для зберігання даних, які можуть змінюватися в процесі експлуатації, але які повинні зберігатися при відключенні живлення.

Основні Характеристики

1. Неволатильність: EEPROM зберігає дані навіть після відключення живлення, що робить її ідеальною для зберігання важливих конфігураційних даних та параметрів.

2. Перезаписування даних: на відміну від традиційної ROM, EEPROM дозволяє електричне стирання та записування даних, що можна виконувати багаторазово.

3. Швидкість доступу: швидкість доступу до EEPROM повільніша порівняно з Flash-пам'яттю та оперативною пам'яттю (RAM).

4. Цикли запису/стирання: EEPROM має обмежену кількість циклів запису/стирання, хоча це число зазвичай досить велике для більшості застосувань.

Застосування в мікроконтролерах

1. Зберігання конфігураційних даних: часто використовується для зберігання даних, які рідко змінюються, наприклад, налаштувань, калібрувальних даних, серійних номерів тощо.

2. Зберігання даних, що можуть змінюватися: ідеально підходить для зберігання даних, які потребують змін у процесі роботи, але не дуже часто (наприклад, лічильники циклів, логи подій).

Технічні Аспекти

- розмір: ємність EEPROM у мікроконтролерах зазвичай менша, ніж у Flash-пам'яті, і може варіюватися від декількох байт до декількох кілобайт;

- інтерфейс: EEPROM може бути інтегрована безпосередньо в мікроконтролер або доступна як зовнішній компонент.

У сучасних мікроконтролерах ці типи пам'яті інтегровані в один чіп, що забезпечує компактність, ефективність та зручність використання.

Інтерфейси в мікроконтролерах відіграють ключову роль у спілкуванні між мікроконтролером та іншими пристроями або компонентами. Ось детальний огляд основних типів інтерфейсів, які зазвичай використовуються в мікроконтролерах:

1. GPIO (General Purpose Input/Output).

- GPIO дозволяє мікроконтролеру спілкуватися з зовнішнім світом через прості сигнали вводу/виводу. Кожен пін GPIO може бути налаштований як вхід або вихід;

- керування світлодіодами, читання стану кнопок, просте цифрове спілкування тощо.

2. UART (Universal Asynchronous Receiver/Transmitter).

- UART забезпечує асинхронне серійне спілкування. Дані передаються послідовно один біт за раз через "передавальний" та "приймальний" піни;

- спілкування з комп'ютерами, модулями GPS, іншими мікроконтролерами тощо.

3. SPI (Serial Peripheral Interface).

- SPI це швидкий серійний інтерфейс, що використовує майстер/раб (master/slave) принцип. Він включає лінії MISO (Master In Slave Out), MOSI (Master Out Slave In), SCK (Serial Clock) та SS (Slave Select);

- підключення до SD-карт, дисплеїв, EEPROM тощо.

4. I²C (Inter-Integrated Circuit).

- I²C це двопровідний серійний інтерфейс, що використовує лінії SDA (Serial Data) та SCL (Serial Clock). Він підтримує кілька ведених пристроїв на одному майстрі;

- підключення до сенсорів, EEPROM, LCD-дисплеїв тощо.

5. PWM (Pulse Width Modulation).

- PWM використовується для створення аналогового сигналу за допомогою цифрових засобів, змінюючи ширину імпульсів у цифровому сигналі;

- керування швидкістю моторів, регулювання яскравості світлодіодів, генерація аудіосигналів тощо.

6. ADC (Analog-to-Digital Converter).

- ADC перетворює аналогові сигнали (наприклад, від сенсорів) в цифрові дані, які може обробляти мікроконтролер;

- читання значень від температурних сенсорів, потенціометрів, фотодіодів тощо.

7. DAC (Digital-to-Analog Converter).

- DAC перетворює цифрові дані в аналоговий сигнал;

- створення аналогових сигналів для аудіо, а також керування аналоговими пристроями.

8. CAN (Controller Area Network).

- CAN це інтерфейс, розроблений для надійного спілкування в автомобільних та промислових застосуваннях;

- автомобільні системи, промислове обладнання тощо.

9. USB (Universal Serial Bus).

- деякі мікроконтролери підтримують USB інтерфейс для спілкування з комп'ютерами та іншими USB пристроями;

- Передача даних, програмування мікроконтролерів, USB-комунікаційні пристрої тощо.

Кожен з цих інтерфейсів має свої особливості, переваги та області застосування. Вибір конкретного інтерфейсу залежить від вимог до проекту,

включаючи швидкість передачі даних, простоту реалізації, кількість доступних пінів та інші фактори.

Аналогово-цифрові перетворювачі (АЦП) в мікроконтролерах дозволяють перетворювати аналогові сигнали (наприклад, від сенсорів) в цифрові дані, які може обробляти мікроконтролер.

АЦП вимірює рівень напруги на своєму вході та перетворює його у цифровий сигнал. Вхідна напруга зазвичай повинна бути в межах допустимого діапазону для АЦП. Після починається процес квантування, суть якого полягає у відображенні аналогового сигналу на певну кількість рівнів. Кожен рівень представлений унікальним цифровим значенням. Наприклад, 10-бітний АЦП може представляти аналоговий сигнал одним із 1024 (2^{10}) можливих значень. Частота Дискретизації - це частота, з якою АЦП вимірює аналоговий сигнал. Вища частота дискретизації дозволяє точніше відтворити швидкі зміни в аналоговому сигналі. Кожному аналоговому значенню відповідає найближчий квантовий рівень. Цей процес включає округлення або обрізання аналогового значення до найближчого квантового рівня. Через обмежену кількість квантових рівнів, виникає помилка квантування - різниця між фактичним аналоговим значенням та його квантованим представленням. Ця помилка є джерелом шуму квантування. Шум квантування є випадковим і зазвичай розглядається як білий шум. Його вплив зменшується зі збільшенням роздільної здатності АЦП. Зі збільшенням кількості бітів АЦП (i , відповідно, квантових рівнів), помилка квантування зменшується, а точність перетворення підвищується. Вища роздільна здатність може призвести до зниження швидкості перетворення, тому важливо знайти баланс між швидкістю та точністю в залежності від вимог застосування.

Типи АЦП:

1. Послідовний приближений реєстр (SAR): найпоширеніший тип в мікроконтролерах. SAR АЦП швидко визначає цифрове значення аналогового сигналу, використовуючи послідовність порівнянь.

2. Інтегруючий АЦП: часто використовується у застосуваннях з низьким рівнем шуму та високою точністю, але має нижчу швидкість перетворення.

3. Дельта-Сигма ($\Delta\Sigma$) АЦП: Забезпечує високу точність та роздільну здатність, особливо корисний для аудіо застосувань.

Таймери в мікроконтролерах є фундаментальними інструментами, які використовуються для вимірювання часу, створення затримок, генерації точних часових інтервалів, управління подіями та інших завдань, де важливий час.

Таймер у мікроконтролері - це спеціалізований внутрішній блок, який рахує імпульси годинника (тактові імпульси). Ці імпульси зазвичай генеруються внутрішнім або зовнішнім джерелом годинника.

Таймер має реєстр, який інкрементується (або декрементується) на одиницю з кожним імпульсом годинника. Розмір цього реєстра (наприклад, 8-бітний, 16-бітний) визначає максимальний час вимірювання. Коли реєстр досягає свого максимального значення, він переповнюється, що може спричинити переривання або іншу подію.

Більшість таймерів мають передільник, який дозволяє зменшити частоту імпульсів годинника, що надходять до таймера, тим самим збільшуючи максимальний час вимірювання.

Типи Таймерів

1. Простий таймер: використовується для створення затримок або вимірювання часу.

2. Таймер-лічильник: може працювати як таймер, вимірюючи внутрішні тактові імпульси, або як лічильник, вимірюючи зовнішні події.

3. PWM (Pulse Width Modulation): деякі таймери можуть генерувати сигнали широтно-імпульсної модуляції, які використовуються для керування моторами, світлодіодами тощо.

4. Watchdog Timer: спеціальний тип таймера, який використовується для перезавантаження системи у випадку, якщо основна програма зависає або працює неправильно.

Застосування Таймерів:

1. Створення затримок: таймери можуть використовуватися для створення точних затримок у програмі.
2. Вимірювання часу: використовуються для вимірювання тривалості подій.
3. Генерація періодичних подій: таймери можуть використовуватися для створення періодичних переривань, що дозволяє виконувати певні завдання з регулярним інтервалом.
4. PWM для керування моторами та світлодіодами: широтно-імпульсна модуляція дозволяє контролювати швидкість моторів та інтенсивність світлодіодів.
5. Вимірювання частоти та часу зовнішніх подій: таймери-лічильники можуть використовуватися для вимірювання частоти зовнішніх сигналів або подій.

У підсумку, таймери в мікроконтролерах є універсальними інструментами, які забезпечують широкий спектр функцій, пов'язаних з часом, від простих затримок до складного керування подіями та сигналами.

Системи управління живленням в мікроконтролерах відіграють важливу роль у забезпеченні ефективного споживання енергії, особливо в пристроях, що працюють від батарей. Ці системи дозволяють оптимізувати споживання енергії, зменшуючи витрати енергії, коли висока продуктивність не потрібна, та забезпечуючи достатню потужність, коли це необхідно.

Більшість мікроконтролерів мають кілька режимів енергозбереження, які зменшують споживання енергії шляхом відключення або зниження потужності окремих блоків або функцій мікроконтролера. Система управління живленням може відключати тактовий генератор або живлення від окремих периферійних пристроїв, коли вони не використовуються. Зниження частоти годинника мікроконтролера в періоди низького навантаження може значно знизити споживання енергії. Мікроконтролер може перейти в режим глибокого сну, але залишатися здатним реагувати на

певні зовнішні або внутрішні події (наприклад, таймери або зовнішні преривання), що дозволяє йому швидко повернутися до повноцінної роботи.

Типи Режимів Енергозбереження

1. Idle Mode: в цьому режимі основний процесор зупиняється, але периферійні пристрої та таймери продовжують працювати.

2. Sleep Mode: процесор та більшість периферійних блоків вимикаються, але деякі функції, такі як таймери або зовнішні преривання, можуть залишатися активними.

3. Deep Sleep Mode: глибокий сон забезпечує ще більше зниження споживання енергії, вимикаючи майже всі функції мікроконтролера.

4. Power-Down Mode: найбільш енергоефективний режим із всіх, де вимикаються майже всі системи, крім мінімального набору функцій для пробудження.

1.2.4. Сенсорні системи в робототехніці

Сенсорні системи в робототехніці відіграють ключову роль, оскільки вони надають роботам здатність сприймати своє оточення, що є критично важливим для виконання різноманітних завдань.

Типи сенсорів (Рис. 1.13):

1. Тактильні сенсорі: використовуються для визначення сили та тиску при контакті з об'єктами. Це може бути корисним для маніпуляції об'єктами або для визначення характеристик поверхні.[2]

2. Пропріоцептивні сенсорі (датчики положення): вимірюють положення, орієнтацію та стан робота. Це включає енкодери на суглобах, гіроскопи та акселерометри.

3. Візуальні сенсори: камери та лідари, які надають візуальну інформацію про оточення. Вони можуть використовуватися для навігації, розпізнавання об'єктів та виконання завдань з високою точністю.

4. аудіосенсори: мікрофони, які дозволяють роботам реагувати на звуки або вести голосову взаємодію.

5. Датчики відстані та руху: ультразвукові сенсори, інфрачервоні сенсори, які використовуються для визначення відстані до об'єктів або їх руху.



Рис. 1.13 Сенсори

1.2.5. Джерела енергії

Джерела енергії в роботичних системах є фундаментальним компонентом, який визначає автономність, ефективність та можливості роботів.

Типи джерел енергії:

1. Акумуляторні батареї.

- літій-іонні батареї (Li-ion): найпопулярніший вибір через високу щільність енергії, довгий термін служби та низький рівень саморозряду;

- нікель-метал-гідридні батареї (NiMH): екологічно чистіші, але мають меншу щільність енергії порівняно з Li-ion;

- свинцево-кислотні батареї: економічні, але важкі та мають короткий термін служби.

2. Сонячні панелі: використовуються в деяких роботах для забезпечення додаткової енергії або як основне джерело в роботах, що працюють на відкритому повітрі.

3. Паливні елементи: виробляють електрику безпосередньо з хімічної реакції, зазвичай використовуючи водень. Вони пропонують високу щільність енергії і можуть швидко поповнюватися.

Динамо-машини: використовуються в деяких роботах для перетворення механічної енергії в електричну.

Важливі Аспекти

- щільність енергії: визначає, скільки енергії може зберігати батарея відносно її ваги або об'єму;

- тривалість роботи: важливий фактор, особливо для мобільних роботів, які потребують тривалої автономності;

- час зарядки та цикли перезарядки: впливає на загальну ефективність та готовність робота до виконання завдань;

- надійність та безпека: важливо забезпечити, щоб джерела енергії були стабільними та безпечними в різних умовах експлуатації.

Виклики:

- вага та розмір: знаходження балансу між потужністю батареї та її вагою/розміром;

- управління енергією: ефективне управління споживанням енергії для максимізації тривалості роботи;

- вартість: високоякісні джерела енергії можуть бути дорогими, що впливає на загальну вартість робота.

Забезпечення роботів надійними, ефективними та тривалими джерелами енергії є критично важливим для їхньої автономності та ефективності в широкому спектрі застосувань.

1.2.6. Інтерфейси управління

Інтерфейси управління в роботичних системах є ключовим елементом, який забезпечує взаємодію між людиною та машиною, дозволяючи операторам керувати роботами та інтерпретувати їхні дії.[8]

Типи інтерфейсів управління:

1. Фізичні контролери:

- джойстики та геймпади: часто використовуються для керування мобільними роботами або роботизованими маніпуляторами;

- клавіатура та миша: традиційні інтерфейси, корисні для точного керування або програмування роботів.

2. Графічні користувацькі інтерфейси (GUI):

- програмне забезпечення для управління роботами: часто має інтуїтивно зрозумілий інтерфейс з візуалізацією даних, картою, статусом робота тощо;

- мобільні додатки: дозволяють керувати роботами дистанційно через смартфони або планшети.

3. Голосове управління: дозволяє керувати роботами за допомогою мовних команд, забезпечуючи більш природню взаємодію.

4. Жестове управління: використовується для інтерпретації жестів оператора, наприклад, за допомогою технологій, подібних до Microsoft Kinect.

5. Інтерфейси мозок-комп'ютер (BCI): нейроінтерфейси дозволяють керувати роботами безпосередньо за допомогою мозкових сигналів.

Важливі аспекти:

- інтуїтивність та легкість використання: інтерфейс повинен бути зрозумілим та легким у використанні для операторів з різним рівнем досвіду;

- відгук та візуалізація: надання операторам зворотного зв'язку та візуальної інформації про стан та дії робота;

- надійність та безпека: забезпечення стабільності та безпеки управління, особливо в критичних застосуваннях.

Виклики:

- комплексність управління: забезпечення ефективного управління складними роботизованими системами;
- затримка та точність: мінімізація затримок у відгуку та забезпечення високої точності команд;
- адаптація до різних сценаріїв: розробка інтерфейсів, які можуть бути ефективними в різних умовах та для різних типів робіт.

1.2.7. Ступені свободи (DoF)

СВ є фундаментальним поняттям у робототехніці, що визначає кількість незалежних рухів, які може здійснити робот або його частина (наприклад, маніпулятор або рука). Розуміння та використання ступенів свободи є ключовим для проектування, програмування та управління роботичними системами. Іншими словами, ступені свободи - це кількість незалежних змінних, які необхідні для опису положення системи або її частини.

Роботична рука з одним обертовим суглобом має один ступінь свободи, оскільки може обертатися лише навколо однієї осі. Додавання до руки більшої кількості суглобів, які можуть обертатися незалежно, збільшує кількість ступенів свободи. Наприклад, рука з трьома такими суглобами матиме три ступені свободи. Якщо рука може переміщуватися в різних напрямках, наприклад, вгору-вниз, вліво-вправо, вперед-назад, це додає ще три ступені свободи.

Положення n -вимірної роботичної руки визначається за допомогою жорсткої трансформації, яка включає n трансляційних та $n(n-1)/2$ обертових ступенів свободи. Одне жорстке тіло має максимум шість ступенів свободи, що складаються з трьох трансляційних та трьох обертових ступенів. Наприклад, роботична рука з шістьма ступенями свободи може виконувати

широкий спектр рухів, включаючи обертання та перекладення у різних напрямках.

Фізичні обмеження в конструкції роботичної руки можуть зменшувати кількість можливих рухів, тобто ступенів свободи, наприклад:

1. Обмеження діапазону руху суглобів: якщо суглоби роботичної руки мають обмежений діапазон руху, це зменшує її здатність до маневрування. Наприклад, якщо ліктьовий суглоб руки може згинатися лише на 90 градусів замість 180, це обмежує можливості руки досягати різних положень.

2. Фіксовані суглоби: у деяких конструкціях роботичних рук можуть бути суглоби, які не забезпечують обертання або згинання. Наприклад, якщо зап'ясток руки фіксований і не може обертатися, це зменшує здатність руки маніпулювати об'єктами з різних кутів.

3. Обмеження через конструктивні особливості: іноді конструкція руки може включати елементи, які фізично блокують певні рухи. Наприклад, якщо частини руки занадто великі або розташовані так, що вони перешкоджають один одному.

4. Обмежена вантажопідйомність: хоча це не прямо пов'язано зі ступенями свободи, вантажопідйомність роботичної руки може обмежувати її функціональність. Наприклад, якщо рука не може піднімати важкі предмети, це обмежує спектр завдань, які вона може виконувати.

Ці обмеження впливають на загальну ефективність та універсальність роботичної руки, обмежуючи її здатність виконувати різноманітні завдання.

Формула Мобільності це рівняння для підрахунку кількості параметрів, які визначають конфігурацію набору жорстких тіл, обмежених суглобами. Вона враховує кількість тіл, суглобів та обмежень, накладених цими суглобами.

Формула мобільності в механіці та робототехніці використовується для визначення кількості ступенів свободи (DOF) в механічній системі, особливо в системах, що складаються з рухомих з'єднань, таких як кінематичні ланцюги або роботизовані механізми. Основна формула мобільності виглядає так:

$$M = 6(N - 1) - 5J_h - 4J_l$$

- M загальна кількість ступенів свободи (мобільність) системи.
- N кількість тіл у системі (включаючи базу (основу) або фіксовану раму).
- J_h кількість вищих (або повних) кінематичних пар. Вища кінематична пара зазвичай дозволяє 1 ступінь свободи (наприклад, шарнірний або обертовий суглоб).
- J_l кількість нижчих кінематичних пар. Нижча кінематична пара зазвичай дозволяє 2 або більше ступенів свободи (наприклад, суглоб, що дозволяє обертання та зсув).

Ця формула є спрощенням і може варіюватися в залежності від конкретної конструкції та обмежень системи. Вона допомагає інженерам та дизайнерам розуміти, наскільки гнучкою або обмеженою є механічна система, і використовується при проектуванні різних механізмів, включаючи роботизовані пристрої.

Вплив на Конструкцію та Функціональність.

Більша кількість ступенів свободи забезпечує більшу гнучкість, але також збільшує комплексність системи, включаючи вимоги до управління, програмування та обслуговування.

Для роботів з багатьма ступенями свободи, планування траєкторій стає складнішим, але це дозволяє виконувати більш складні та різноманітні завдання.

Підтримка високої точності та повторюваності в роботах з багатьма ступенями свободи вимагає розвинутих технологій управління та калібрування.

Роботи з багатьма ступенями свободи часто інтегруються з розширеними системами контролю, включаючи штучний інтелект і машинне навчання, для більш ефективного виконання завдань.

1.3. Артикульована роботична рука

Артикульована роботична рука - це комплексна мехатронна система, спроектована для відтворення основних характеристик та можливостей людської руки. Цей роботичний пристрій складається з кількох зчленованих сегментів, відомих як ланки, які з'єднані між собою рухомими з'єднаннями або суглобами. Кожен з цих сегментів може рухатися навколо осей, що надає роботичній руці значну гнучкість та динаміку.[14]

Артикульовані роботичні руки призначені для виконання різних завдань, що вимагають високого рівня точності, маніпуляції об'єктами та координації. Вони широко використовуються в різних галузях, включаючи:

1. Виробництво: артикульовані роботичні руки використовуються для зборки, обробки та переміщення виробів на виробничих лініях. Вони дозволяють автоматизувати рутинні завдання та підвищити продуктивність виробництва.

2. Медицина: у хірургії та реабілітації артикульовані роботичні руки можуть використовуватися для виконання хірургічних операцій з високою точністю та мінімальним втручанням.

3. Дослідження: в лабораторних дослідженнях артикульовані роботичні руки використовуються для вивчення руху, маніпулювання об'єктами та розробки нових технологій.

4. Просторова навігація: у космічних або автономних дослідженнях артикульовані роботичні руки можуть бути використані для виконання робіт у важкодоступних чи небезпечних середовищах.

Основними компонентами артикульованої роботичної руки є ланки, які з'єднують всі деталі, суглоби, серводвигуни або мотори, сенсори для вимірювання положення та сил, контролер (система управління) для керування рухом та інтеграція з іншими системами, а також програмне забезпечення для планування та керування рухом.

Механічна конструкція роботичної руки зазвичай складається з легких, але міцних матеріалів, таких як алюміній, карбонові волокна або титан. Ця конструкція повинна бути досить міцною, щоб витримувати навантаження, але при цьому легкою для забезпечення швидких та точних рухів.

Ці роботи можуть бути програмовані для виконання різних завдань, від простого підняття та переміщення предметів до складних завдань, таких як монтаж або хірургічні операції. Артикульовані роботичні руки відкривають широкі можливості для автоматизації різних галузей та вдосконалення виробничих процесів, що робить їх важливими компонентами в сучасному світі автоматизації і робототехніки.

1.3.1. Основні сегменти роборуки

Артикульована роборука з шістьма ступенями свободи (6DOF) складається з декількох ключових сегментів, які дозволяють їй імітувати рухи людської руки.

1. Основа або підстава - це фундамент роборуки, який часто кріпиться до столу або іншої поверхні. Він забезпечує стабільність для всієї конструкції.

2. Плече - це перший сегмент руки, який виходить від основи. Воно забезпечує вертикальний рух руки вгору та вниз.

3. Лікоть - сегмент, який з'єднує плече з передпліччям. Лікоть дозволяє роборуці згинатися та розгинатися, подібно до людської руки.

4. Передпліччя - це сегмент між ліктем та зап'ястком. Воно часто містить механізми для обертання, що дозволяє роборуці повертати об'єкти.

5. Зап'ясток - зап'ясток забезпечує обертання та нахил кисті, дозволяючи роборуці орієнтувати інструмент або захоплювач у потрібному напрямку.

6. Кисть або захоплювач – це кінцевий елемент роборуки, який може бути оснащений різними захоплювачами або інструментами в залежності від завдань, які вона виконує. Кисть може мати різну ступінь складності, від простих двохпальцевих захоплювачів до складних маніпуляторів, що імітують людську руку.

Кожен з цих сегментів містить різні механічні та електронні компоненти, такі як сервоприводи, датчики, та контролери, які дозволяють роботу виконувати точні та складні рухи.

1.3.2. Роль серводвигунів у руху руки

Серводвигуни відіграють критичну роль у функціонуванні роботи.

1. Точне позиціонування: серводвигуни забезпечують високу точність позиціонування. Вони можуть точно контролювати кут повороту, швидкість і прискорення, що є критично важливим для точного маніпулювання об'єктами.

2. Контроль руху: кожен ступінь свободи в роботі зазвичай керується окремим серводвигуном. Це дозволяє незалежно контролювати рух у кожному суглобі, що забезпечує гнучкість і різноманітність рухів.

3. Зворотний зв'язок: серводвигуни часто оснащені датчиками, які надають зворотний зв'язок контрольній системі роботи. Це дозволяє системі точно відстежувати і коригувати положення, швидкість і крутний момент кожного суглоба.

4. Забезпечення сили та стабільності: серводвигуни забезпечують необхідну силу для переміщення частин роботи і утримання ваги об'єктів. Вони також допомагають підтримувати стабільність роботи під час виконання завдань.

1.3.3. Загальний контроль руху роботи

Контроль руху в артикульованій є складним завданням, яке вимагає інтеграції механічних компонентів, електроніки та програмного забезпечення. Кожен ступінь свободи зазвичай керується серводвигуном або електричним приводом. Ці приводи забезпечують необхідний крутний момент для руху суглобів роботи.

Центральна система керування, часто на базі мікроконтролера або комп'ютера, відповідає за координацію рухів всіх серводвигунів. Вона отримує вхідні дані (наприклад, від оператора або програми) і перетворює їх у команди для двигунів.

Програмне забезпечення роборуки використовує різні алгоритми для розрахунку траєкторій руху, керування швидкістю та прискоренням, а також для забезпечення плавності та точності рухів.

Для точного контролю руху використовуються датчики, такі як енкодери, які вимірюють кут повороту, швидкість та положення кожного суглоба. Ця інформація використовується для корекції та точного контролю рухів.

Для взаємодії з роборукою часто використовуються спеціалізовані інтерфейси, такі як пульти управління, комп'ютерні програми або навіть системи автоматичного планування руху.

У деяких випадках роборука може інтегруватися з іншими системами, такими як машинний зір, штучний інтелект, для виконання більш складних завдань, таких як розпізнавання об'єктів або адаптивне маніпулювання. [4]

1.3.4. Апаратне та програмне забезпечення

Створення артикульованої роборуки вимагає інтеграції як апаратного, так і програмного забезпечення. Ось загальний огляд компонентів та підходів, які можуть бути використані:

Апаратне забезпечення:

1. Arduino плата: як основа для контролера, Arduino (наприклад, Arduino Uno або Mega) є популярним вибором завдяки своїй гнучкості та доступності.

2. Серводвигуни: для кожного ступеня свободи використовуються серводвигуни. Важливо вибрати серводвигуни, які можуть забезпечити достатній крутний момент та точність для конкретних завдань роборуки.

3. Зовнішнє живлення: оскільки серводвигуни можуть вимагати більше енергії, ніж може надати Arduino, зазвичай потрібне зовнішнє джерело живлення.

4. Датчики: для зворотного зв'язку можуть використовуватися різні датчики, такі як потенціометри або енкодери, встановлені на суглобах.

Програмне забезпечення:

1. Розробка коду на C: програмування Arduino зазвичай виконується на мові, схожій на C. Код буде керувати логікою руху, обробкою даних з датчиків та керуванням серводвигунами.

2. Бібліотеки Arduino: існують численні бібліотеки для Arduino, які можуть полегшити роботу з серводвигунами, датчиками та іншими компонентами.

3. Алгоритми контролю руху: це включає в себе алгоритми для керування позиціонуванням, траєкторією руху, обробкою зворотного зв'язку та, можливо, алгоритми для обробки зовнішніх команд.

4. Інтерфейс користувача: якщо потрібно, можна розробити простий інтерфейс для взаємодії з роборукою, наприклад, через комп'ютерний серійний порт.

1.3.5. Система комунікації (Bluetooth модуль)

Bluetooth модуль в даній артикульованій роботичній руці є ключовим компонентом, який дозволяє бездротово зв'язуватися з іншими пристроями, такими як смартфони, комп'ютери або інші контрольні пристрої. Нижче розглянуто деталі цієї системи комунікації:

Bluetooth є бездротовим протоколом зв'язку, розробленим для обміну даними між різними пристроями на коротких відстанях (зазвичай до 100 метрів). Він широко використовується для побудови бездротових зв'язків між різними пристроями, зокрема роботами та смартфонами.

Головне завдання Bluetooth модуля в системі роботичної руки - це забезпечити зв'язок з іншими пристроями (наприклад, смартфоном) для

керування роботичною рукою з відстані і передавати дані про стан руки і зворотний зв'язок.

Проектування та реалізація зв'язку. Перший крок - вибір відповідного Bluetooth модуля, який відповідає потребам вашого проекту. Це може бути Bluetooth Low Energy (BLE) модуль для низького споживання енергії або класичний Bluetooth модуль для більш швидкого передавання даних. Bluetooth модуль повинен бути підключений до мікроконтролера роботичної руки (наприклад, Arduino). Для цього зазвичай використовуються інтерфейси, такі як UART або SPI, для передачі та отримання даних. Для забезпечення комунікації з Bluetooth модулем потрібно написати програмне забезпечення на мікроконтролері, яке керує процесом передачі та прийому даних через Bluetooth. Використання бібліотек або SDK для Bluetooth спрощує цей процес.

Коли Bluetooth модуль вмикається, його можна парувати з іншими пристроями. Після вдалого парування можна створити з'єднання між роботичною рукою та пристроєм для взаємодії.

За допомогою спеціального програмного забезпечення або додатка на смартфоні, користувач може відправляти команди на керування роботичною рукою. Це може включати в себе переміщення сегментів, захоплення об'єктів або виконання інших завдань.

Bluetooth модуль також може передавати інформацію про стан руки, наприклад, координати сегментів, стан сенсорів і стан гріпса, до смартфона. Це дає користувачу можливість візуально контролювати та моніторити роботу роботичної руки.

Ця система комунікації дозволяє користувачам керувати і взаємодіяти з артикульованою роботичною рукою з відстані, що робить її більш гнучкою та зручною у використанні.

1.3.6. Кінематика роборук

Кінематика – це наука, яка вивчає рухи механічних систем без врахування сил, що викликають ці рухи. Для роборук, кінематика включає аналіз положення, швидкості, прискорення, і всіх можливих конфігурацій руху роботизованої руки.[12]

Типи роборук можна класифікувати за їх кінематичною структурою. Кінематична структура визначає, як рухаються різні частини роборуки відносно одна одної та як вони досягають різних положень у просторі. [15]

1. Картезіанські роборуки (Рис. 1.14) мають три лінійні осі, які рухаються вздовж координат X, Y, і Z. Це надає їм прямокутну робочу зону. Часто використовуються для точного позиціонування, наприклад, у 3D-друку, CNC-машини, або в складальних лініях.

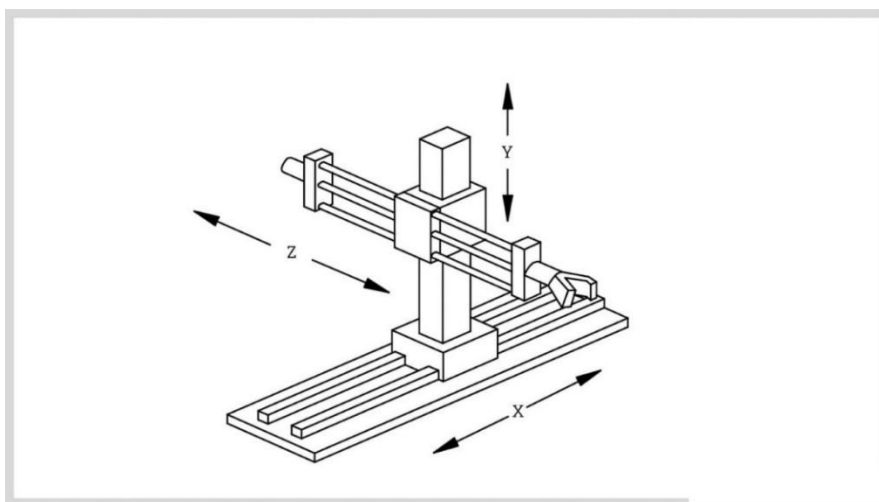


Рис. 1.14 Картезіанська роборука

2. Циліндричні роборуки (Рис. 1.15) комбінують лінійний рух з обертовим. Один суглоб дозволяє рух вгору та вниз (лінійний), а інший - обертання навколо вертикальної осі. Підходять для операцій, де потрібно комбінувати підйом та обертання, наприклад, у монтажних роботах.

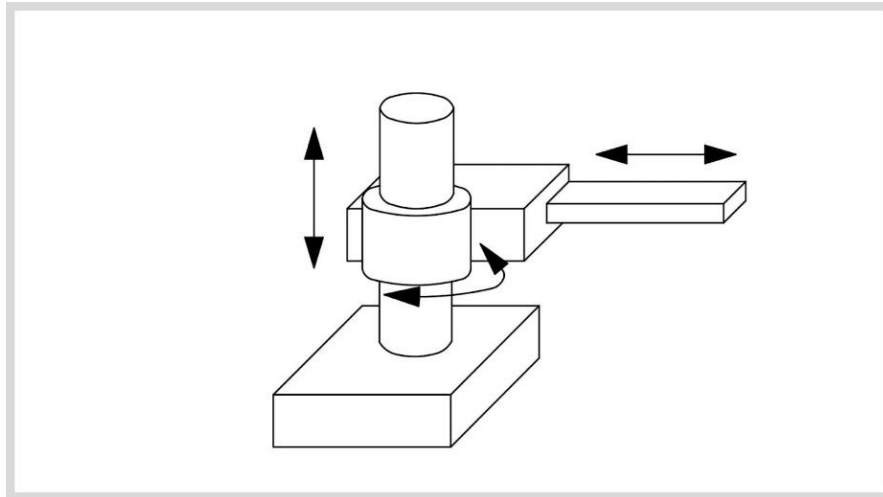


Рис. 1.15 Циліндрична роборука

3. Сферичні або полярні роборуки (Рис. 1.16) мають обертовий суглоб у основі, один або два ліктьових суглоби для вертикального руху, та додаткові суглоби для орієнтації кінцевого ефектора. Часто використовуються у зварюванні, фарбуванні, та інших застосуваннях, де потрібен широке охоплення робочої зони.

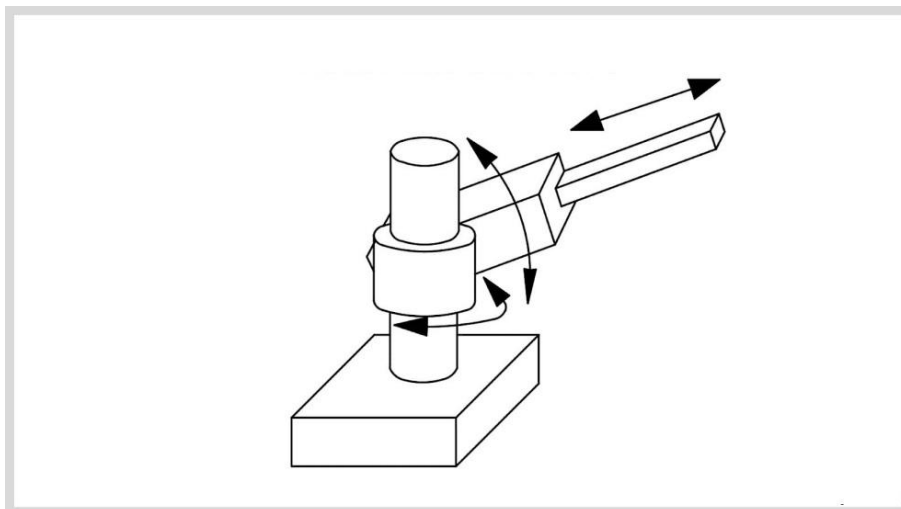


Рис. 1.16 Сферична роборука

4. SCARA (Рис. 1.17) мають два паралельні обертові суглоби для руху в горизонтальній площині, з додатковим лінійним суглобом для вертикального руху. Ідеальні для високошвидкісних монтажних та складальних операцій, особливо в електронній промисловості.

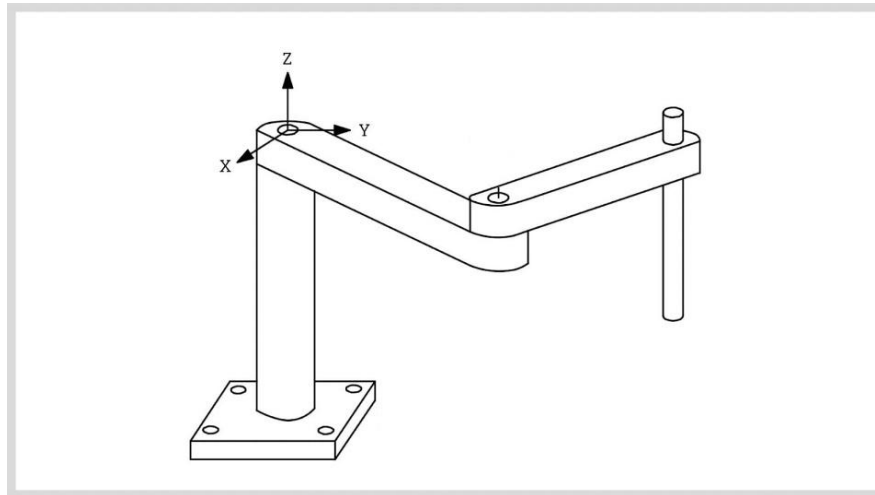


Рис. 1.17 Роботична конфігурація SCARA

5. Артикульовані роборуки (Рис. 1.18) мають кілька обертових суглобів, що надають їм структуру, схожу на людську руку. Універсальні та гнучкі, використовуються у багатьох застосуваннях, включаючи зварювання, фарбування, збірку, та навіть у медичних застосуваннях.

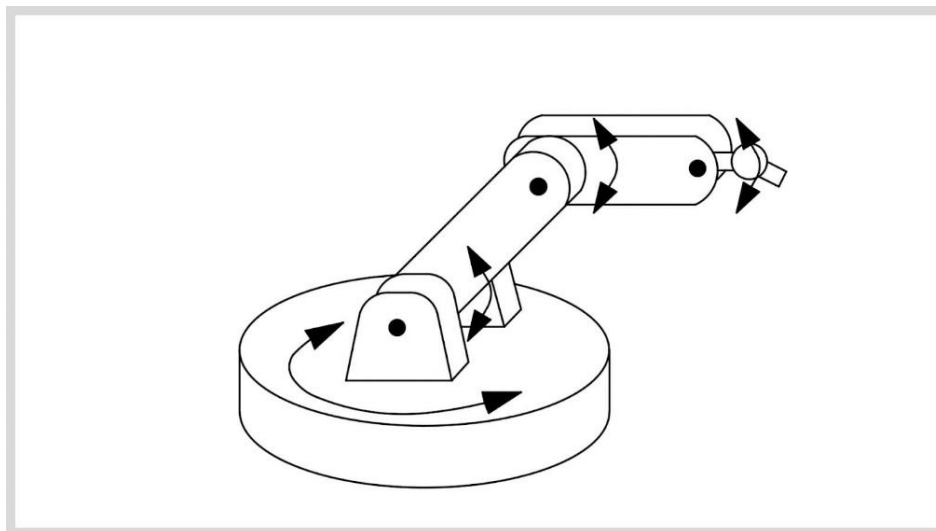


Рис. 1.18 Артикульована Роборука

6. Паралельні (Дельта) роборуки (Рис. 1.19) складаються з паралельних стрижнів, з'єднаних з основою, що контролюють рух кінцевого ефектора. Часто використовуються для швидких, точних рухів у легких складальних операціях, таких як упаковка або сортування.



Рис. 1.19 Паралельна роборука

7. Гнучкі (Soft) роборуки (Рис. 1.20) виготовлені з гнучких матеріалів, дозволяючи імітувати природні рухи, подібні до рухів живих організмів. Новий напрямок у робототехніці, з потенціалом застосування в медицині, дослідженні глибоководних середовищ, та в інших областях, де потрібна висока адаптивність та безпека.



Рис. 1.20 Гнучка роборука

Кожен тип роборуки має свої унікальні характеристики та переваги, і вибір конкретного типу залежить від специфічних потреб застосування та кінематики.

Пряма кінематика є фундаментальною частиною вивчення робототехніки, особливо в контексті роборук.

Пряма кінематика роборук зосереджується на визначенні положення та орієнтації кінцевого ефектора (наприклад, хапала) на основі відомих кутів повороту суглобів робота. Ключовим елементом тут є ступені свободи (DoF), які визначають кількість незалежних рухів, які роборука може виконувати. Наприклад, роборука з шістьма ступенями свободи може виконувати шість незалежних рухів, що дозволяє їй досягати широкого спектру положень та орієнтацій у просторі.

Математично, пряма кінематика вирішується через використання матриць перетворення. Кожна ланка роборуки асоціюється з матрицею перетворення, яка описує її положення та орієнтацію відносно попередньої ланки. Ці матриці включають інформацію про повороти та зсуви, які необхідні для переходу від однієї ланки до іншої.

Для систематизації процесу визначення цих матриць часто використовується метод Денавіт-Хартенберга (D-H). Цей метод вимагає визначення чотирьох параметрів для кожної ланки: кута повороту, відстані, довжини ланки та кута нахилу. Ці параметри дозволяють створити унікальну матрицю перетворення для кожної ланки.

Процес прямої кінематики починається з визначення параметрів D-H для кожної ланки роборуки. Після цього, для кожної ланки формується відповідна матриця перетворення. Положення та орієнтація кінцевого ефектора визначаються шляхом послідовного множення цих матриць перетворення, починаючи від бази роборуки і закінчуючи кінцевим ефектором.

Цей процес вимагає точних математичних обчислень та глибокого розуміння геометрії роборуки. Він дозволяє точно визначити, де і як буде розташований кінцевий ефектор роборуки при заданих кутах повороту суглобів, що є критично важливим для багатьох застосувань, від промислового виробництва до хірургічних операцій.

Завдяки прямій кінематиці, можливе точне програмування та контроль рухів роборуки, що дозволяє їм виконувати складні, точні та повторювані завдання в різних областях застосування.

Розглянемо приклад математичного підходу до прямої кінематики, використовуючи метод Денавіт-Хартенберга (D-H). Цей метод є стандартним для визначення положення та орієнтації кінцевого ефектора роборуки на основі кутів повороту її суглобів.

Приклад: Проста Роборука з Двома Суглобами

Уявімо, що у нас є роборука з двома суглобами (обертливими) та двома ланками. Ми хочемо визначити положення кінцевого ефектора цієї руки.

Крок 1: Визначення параметрів D-H.

Для кожної ланки роборуки визначаються чотири параметри D-H:

1. θ (theta) - кут повороту навколо осі, що передує ланці.
2. d (distance) - відстань вздовж цієї осі до початку наступної осі.
3. a (alpha) - довжина однієї ланки, відстань вздовж попередньої осі до наступної осі.
4. α (alpha) - кут нахилу осі, кут між попередньою віссю та наступною віссю.

Для спрощення припустимо, що обидві ланки мають однакову довжину a і нульові відстані d , а кути нахилу α також дорівнюють нулю.

Крок 2: Формування матриць перетворення.

Для кожної ланки роборуки формується матриця перетворення на основі її параметрів D-H. Матриця перетворення для ланки i зображена на рис 1.21.

$$T_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) & 0 & a_i \sin(\theta_i) \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Рис. 1.21 Матриця перетворення

Де θ_i , d_i , a_i , і α_i - це параметри D-H для ланки i .

Крок 3: Обчислення кінцевого положення.

Положення кінцевого ефектора обчислюється шляхом множення матриць перетворення всіх ланок:

$$T_{\text{total}} = T_1 \times T_2 \times \dots \times T_n$$

Де T_{total} - це кінцева матриця перетворення, яка визначає положення та орієнтацію кінцевого ефектора.

Кінцеве положення кінцевого ефектора знаходиться шляхом множення T_1 та T_2 .

Цей приклад демонструє основний принцип прямої кінематики. У реальних роборуках, де може бути більше ланок та складніші конфігурації, обчислення стають більш складними, але загальний принцип залишається тим самим.

Обернена кінематика відрізняється від прямої кінематики тим, що замість визначення положення кінцевого ефектора на основі відомих кутів повороту суглобів, вирішує протилежну задачу: визначення необхідних кутів повороту суглобів для досягнення заданого положення та орієнтації кінцевого ефектора.

Обернена кінематика є значно складнішою задачею порівняно з прямою кінематикою, оскільки часто існує багато можливих рішень або, в деяких випадках, рішення може і зовсім не існувати. Математично ця задача часто вирішується через методи оптимізації або чисельні методи, такі як метод Ньютона-Рафсона.

Спочатку визначається точне положення та орієнтація, які повинен досягти кінцевий ефектор. Часто вибір початкової конфігурації суглобів може вплинути на успіх або швидкість знаходження рішення. Використовуючи чисельні методи, систематично коригуються кути повороту суглобів, поки положення кінцевого ефектора не буде достатньо близьким до цільового. Оскільки можливі множинні рішення, важливо перевірити, чи рішення є практичним та безпечним для даної конкретної роборуки.

Основним викликом у оберненій кінематиці є визначення оптимального рішення з урахуванням обмежень, які накладаються конструкцією роборуки, таких як обмеження кутів повороту суглобів та можливість зіткнень. Крім того, важливо забезпечити плавність та безпеку рухів, особливо у випадках, коли роборука взаємодіє з людьми або виконує складні завдання.

1.4. Порівняння існуючих аналогів

Саморобна Arduino і Bluetooth-керована роботична рука від **Appuals**:

- у цьому проекті створюється роботична рука, якою керують за допомогою мікроконтролера Arduino та Bluetooth через Android-додаток дистанційного керування;

- зазвичай такі роботичні руки включають 6 суглобів і використовують щонайменше 4 крокових мотори, які керуються комп'ютером;

- компоненти включають Arduino Nano, Bluetooth-передавач HC-05, перемичкові дроти, макетну плату та адаптер живлення;

- процес складання включає підключення моторів і Bluetooth-модуля до плати Arduino Nano, за яким слідує програмування за допомогою Arduino IDE та додатку Android для керування рукою.[13]

Саморобна Arduino & Bluetooth-керована роботична рука від **ElectronicsHub**:

- у цьому проекті показано, як створити Arduino & Bluetooth-керовану роботичну руку, якою можна керувати вручну або програмувати для повністю автоматичного режиму;

- основні компоненти включають Arduino UNO, 4 сервомотори Tower Pro MG90S Metal Gear, Bluetooth-модуль HC-05, 3D-надруковані частини та спеціальний додаток Android;

- дизайн схеми простий, з сигналами управління 4 сервомоторів, підключеними до 4 ШІМ-контактів Arduino UNO;

- роботичною рукою можна керувати вручну або програмувати її для виконання серії завдань в автоматичному режимі.[14]

Прототип роботичної руки, описаний у цій кваліфікаційній роботі, має кілька переваг у порівнянні з іншими доступними на ринку моделями:

1. Ширший спектр знань: прототип включає 6 ступенів свободи, що перевищує можливості моделей з меншою кількістю DOF. Це забезпечує глибше вивчення механіки та кінематики роботичних рук.

2. Доступність та низька вартість: завдяки використанню доступних компонентів, цей прототип є значно дешевшим у порівнянні з комерційними аналогами, що робить його ідеальним для освітніх установ з обмеженим бюджетом.

3. Сприяння навчанню та викладанню: простота складання та програмування роботичної руки з Arduino робить її ідеальною для освітнього середовища, дозволяючи студентам легше зрозуміти основи робототехніки.

4. Приватність і незалежність: на відміну від комерційних продуктів, цей прототип забезпечує повну свободу модифікації та адаптації, сприяючи інноваційному навчальному процесу.

5. Практичність і адаптація: модель може бути адаптована до різних навчальних потреб, дозволяючи освітнім закладам налаштовувати її для досягнення конкретних учбових цілей.

Цей прототип не тільки демонструє технологічні можливості, але й відкриває нові шляхи для освітнього застосування, пропонуючи гнучкість, доступність і легкість у викладанні.

ВИСНОВКИ ДО РОЗДІЛУ 1

Роботизовані механічні системи знаходять застосування в різних сферах: промисловості, медицині, розвагах, наукових дослідженнях та ін. Вони можуть бути створені для специфічних задач, таких як збірка виробів, хірургічних операціях або дослідження важкодоступних місць.

У контексті даного дослідження, розробка роботизованої руки на базі Arduino може розглядатися як часткова реалізація такої системи, де основний акцент робиться на доступності, гнучкості та ефективності конструкції.

В даному розділі ми провели всебічний аналіз робототехніки, розглянувши ключові аспекти та тенденції її розвитку. Ми виявили, що історичний прогрес у цій галузі призвів до створення високофункціональних роботизованих механічних систем, що змінюють обличчя сучасної індустрії та технологій.

Основні характеристики та принципи роботи цих систем, включаючи механічну структуру, актуатори, системи управління, сенсорні системи, джерела енергії та інтерфейси управління, демонструють вражаючу інтеграцію та взаємодію компонентів.

Особливу увагу ми приділили артикульованій роботичній руці, аналізуючи її основні сегменти, роль серводвигунів у русі руки, загальний контроль, апаратне та програмне забезпечення, а також систему комунікації блютуз та кінематику. Ця частина дослідження підкреслює важливість кожного компонента та їхню взаємодію для досягнення точності та гнучкості в роботі роботизованих рук.

Нарешті, порівняння існуючих аналогів дозволило нам краще зрозуміти стан ринку робототехніки, виявити найсучасніші досягнення та визначити ключові напрямки для подальшого розвитку.

РОЗДІЛ 2

ПРОЕКТУВАННЯ РОБОРУКИ ТА ІНСТРУМЕНТИ РОЗРОБКИ

2.1. Опис вимог до прототипу

1. Функціональність.

- ступені свободи: рука має включати 6 ступенів свободи, щоб забезпечити максимальну гнучкість та ефективність у різних задачах;
- артикуляція: кожен сегмент руки має бути артикульованим, дозволяючи точні та різноманітні рухи.

2. Управління.

- блютуз-з'єднання: рука має підключатися до мобільного додатку через блютуз для бездротового управління;
- мобільний додаток: повинен мати інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам легко управляти рукою.

3. Конструкція.

- матеріали: використання легких, але міцних матеріалів для забезпечення довговічності та зручності у використанні;
- мінімалістичний дизайн: простота конструкції для легкості збірки та обслуговування.

4. Програмування.

- Arduino-сумісне: програмне забезпечення руки має бути сумісним з платформою Arduino, легко модифікованим та налаштовуваним;

Кафедра КІТ				НАУ 23 10 36 000 ПЗ			
	ПІБ			РОЗДІЛ 2. ПРОЕКТУВАННЯ РОБОРУКИ ТА ІНСТРУМЕНТИ РОЗРОБКИ	Літ.	Аркуш	Аркушів
Розроб.	Козловцев Д.С.					71	13
Керівник	Зудов О.М.				ТП-215М - 122		
Н.Контр.	Толстікова О.В.						

5. Освітній аспект.[10]

- доступність: проект має бути доступним для використання в освітніх закладах, з особливою увагою до простоти збірки та безпеки;
- навчальні матеріали: надання інструкцій та навчальних матеріалів для вчителів та студентів.

6. Безпека та надійність.

- безпека у використанні: конструкція та програмне забезпечення мають бути безпечними для користувачів усіх вікових категорій;
- тестування: ретельне тестування відносно всіх компонентів для забезпечення стабільної та надійної роботи.

Цей опис вимог має допомогти чітко структурувати різні аспекти проекту та забезпечити його успіх як освітнього інструменту.[5]

2.2. Архітектура та компоненти роботичної руки

2.2.1. Основа і каркас

Дерев'яне шасі - 22смх22см: Служить основою для всієї конструкції. На це шасі будуть кріпитися всі основні компоненти, включаючи сервоприводи та електроніку.

Структура руки: багатофункціональні алюмінієві стандартні кріплення для сервоприводу (кранштейни) х3 (Рис. 2.1) і алюмінієві кріплення для сервоприводу у формі букви U х3: Ці кріплення використовуються для створення суглобів руки. Вони дозволяють сервоприводам обертатися і забезпечують рухливість в різних напрямках.



Рис. 2.1 Багатофункціональні кронштейни для актуаторів

Металевий рог або ричаг (Рис. 2.2) для сервоприводу 25Т х3 встановлюються на сервоприводи і передають рух від сервоприводів до різних частин руки.



Рис. 2.2 Металевий ричаг 25Т

М3 і М4 гвинти та гайки використовуються для кріплення всіх компонентів руки разом, забезпечуючи міцність і стабільність конструкції.

Система приводу складається з сервоприводу Mg996 R з високим моментом обертання х3 і Сервопривод SG90 - 9 г міні/мікро х1. Ці сервоприводи відповідають за рух різних частин роборуки. Високомоментні сервоприводи (Mg996 R) використовуються для більш важких частин руки, тоді як міні-сервопривод (SG90) може бути використаний для більш легких і точних рухів, наприклад, в гріппері.

Гріппер (Рис. 2.3), або маніпулятор-захоплювач - це кінцевий механізм, який використовується для захоплення та маніпулювання об'єктами. Цей механізм кріпиться до кінця роборуки і може бути приведений в дію за допомогою одного або декількох сервоприводів.



Рис. 2.3 Гріппер

З'єднання та Кріплення:

1. Гвинти з плоскою головкою під викрутку різних розмірів (M3 і M4). Використовуються для з'єднання всіх механічних частин.
2. Шестигранні гайки (M3 і M4). Забезпечують міцне кріплення гвинтів.
3. Двосторонній скотч і хомут для кабелів - використовуються для фіксації проводів і деяких легких компонентів на шасі.

Ця структура забезпечує гнучкість і міцність роборуки, дозволяючи їй виконувати різні маніпуляції та рухи. Кожен суглоб і частина руки може бути налаштована і калібрована для досягнення бажаної точності та гнучкості рухів.

2.2.2. Актуатори роборуки

Сервоприводи Mg996 R (Рис. 2.4) з високим моментом обертання (High Torque Mg996 R Metal Gear Digital Servo) - ці сервоприводи є основними актуаторами роборуки. Вони мають високий момент обертання, що робить їх ідеальними для використання у важливіших вузлах руки, де потрібна більша

сила. Висока надійність і стійкість до фізичних навантажень роблять їх ідеальним вибором для проєкту, оскільки нам потрібна висока стійкість і тривалий час експлуатації.



Рис. 2.4 Сервопривод MG996R

Основні Характеристики:

1. Високий момент обертання: сервопривод Mg996 R відомий своїм високим моментом обертання. Це означає, що він може виконувати важкі завдання і управляти важкими навантаженнями, що робить його ідеальним для ключових частин роборуки, де потрібна значна сила.

2. Металева передача: використання металевої передачі замість пластикової значно підвищує довговічність і надійність сервопривода. Металеві шестерні краще витримують навантаження і знос.

3. Цифрове управління: цифрові сервоприводи, як Mg996 R, забезпечують більш точне і стабільне управління, ніж їх аналогові аналоги. Це дозволяє точно контролювати положення руки.

Технічні Дані:

- напруга: типово працює в діапазоні від 4.8 до 7.2 вольт;

- швидкість обертання: швидкість обертання варіюється залежно від напруги живлення, зазвичай близько 0.17 сек/60° при 4.8В і 0.14 сек/60° при 6В;

- момент обертання: значення моменту обертання може бути в межах 10 кг·см при 4.8В і до 12 кг·см при 6В.

Сервопривод SG90 (Рис. 2.5) - 9 г Mini/Micro (SG90 Servo - 9 gms Mini/Micro) - через свої невеликі розміри і вагу, SG90 часто використовується в кінцевих частинах роботизованих систем, таких як гріппери або маленькі механізми захоплення. Цей сервопривод добре підходить для виконання дрібних, точних рухів, необхідних в механізмах з високою точністю.



Рис. 2.5 Сервопривод SG90

Основні Характеристики:

1. Малі розміри і вага: SG90 є одним з найменших і найлегших сервоприводів, що робить його ідеальним для застосувань, де простір та вага мають велике значення.

2. Пластикова передача: на відміну від більш потужних сервоприводів з металевими шестернями, SG90 має пластикову передачу, що робить його менш витривалим, але значно знижує вартість та вагу.

3. Підходить для легких завдань: цей сервопривод призначений для застосувань, де не потрібен великий момент обертання, наприклад, для управління невеликими механізмами або як частина більш складних систем.

Технічні Дані:

- напруга: робоча напруга варіюється від 4.8 до 6 вольт;
- швидкість обертання: в середньому, 0.12 сек/60° при 4.8В і 0.10 сек/60° при 6В;
- момент обертання: приблизно 1.8 кг·см при 4.8В і 2.2 кг·см при 6В.

Сервоприводи підключаються до Arduino через спеціальні виводи і управляються за допомогою бібліотеки сервоприводів в Arduino. Управління сервоприводами може бути синхронізовано з сигналами, отриманими від Bluetooth модуля HC-05, дозволяючи дистанційно керувати рухами роборуки.

Кожен сервопривод у цій системі спеціально підібраний для виконання певних завдань, від важких навантажень до точних і дрібних рухів, забезпечуючи таким чином гнучкість і ефективність роборуки.

2.2.3. Система управління для артикульованої роборуки

Arduino Uno Rev3 (Рис. 2.6) - центральний контролер, який координує всі дії роборуки. Arduino Uno Rev3 є однією з найпопулярніших моделей в серії Arduino, використовується в різноманітних DIY проектах, освітніх програмах, і навіть у промислових застосуваннях.



Рис. 2.6 Arduino Uno Rev3

Основні характеристики:

1. Мікроконтролер: на основі Atmega328P.
2. Цифрові вводи та виводи (I/O) піни: 14 штук, з яких 6 можуть використовуватися як виходи PWM (пульс-ширинної модуляції).
3. Аналогові входи: 6 штук.
4. Робоча напруга: 5V.
5. Напруга живлення (рекомендована): 7-12V.
6. Напруга живлення (межі): 6-20V.
7. Ліміт струму для I/O пінів: 20 mA.
8. Ліміт струму для 3.3V піна: 50 mA.
9. Флеш-пам'ять: 32 KB (ATmega328P), з яких 0.5 KB використовується bootloader'ом.
10. SRAM: 2 KB (ATmega328P).
11. EEPROM: 1 KB (ATmega328P).
12. Частота Годинника: 16 MHz.

Конструкція:

- порт USB: для з'єднання з комп'ютером та завантаження програм;
- бар'єрний роз'єм: для живлення від зовнішнього джерела (наприклад, батареї);
- вбудований регулятор напруги: дозволяє платі працювати в широкому діапазоні вхідних напруг;
- LED індикатори: для відображення стану живлення, передачі даних та іншої інформації;
- Reset кнопка: для перезавантаження мікроконтролера.

Програмування та використання:

- Arduino IDE: Arduino Uno програмується через Arduino IDE, яка є дружньою до користувача і підтримує різні мови програмування, включаючи C та C++;

- Bootloader: уже встановлений bootloader дозволяє легко завантажувати нові програми через USB-порт без потреби в зовнішньому програматорі.

Переваги:

- легкість у використанні: простий для розуміння та програмування, ідеально підходить для початківців;

- гнучкість: може бути використаний у широкому спектрі проектів;

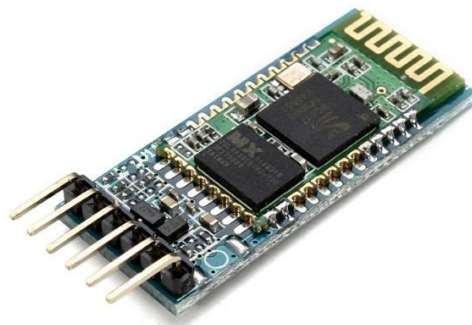
- спільнота: велика онлайн спільнота, багато навчальних матеріалів та проектів.

Недоліки:

- обмежена пам'ять та швидкість: для дуже складних проектів можуть знадобитися більш потужні плати;

- обмеження I/O пінів: для деяких проектів кількість доступних пінів може бути недостатньою.

Модуль HC-05 (Рис. 2.7) є одним з найпопулярніших Bluetooth-модулів, який використовується у багатьох проектах DIY, особливо в сфері робототехніки та автоматизації, є другим важливим елементом управління.



Banggood.com

Рис. 2.7 Bluetooth модуль HC-05

Основні характеристики:

1. Bluetooth специфікація: HC-05 підтримує Bluetooth 2.0 EDR (Enhanced Data Rate), що забезпечує швидкість передачі даних до 3 Мбіт/с.

2. Робочий режим: модуль може працювати як у режимі майстра (master mode), так і в режимі підлеглого (slave mode).

3. Дальність дії: ефективна дальність дії складає близько 10 метрів без перешкод.

4. Швидкість серійного з'єднання: від 9600 до 115200 бод.

Підключення та інтерфейси:

1. Серійний інтерфейс (UART): HC-05 використовує серійний інтерфейс для з'єднання з мікроконтролерами, такими як Arduino. Це дозволяє передавати дані між Arduino та Bluetooth-пристроєм.

2. GPIO піни: є додаткові піни для розширених функцій і налаштувань.

Програмування та конфігурація:

1. AT команди: HC-05 можна програмувати та налаштовувати за допомогою AT команд. Це дає можливість налаштувати ім'я пристрою, пароль, швидкість серійного з'єднання та інші параметри.

2. З'єднання з різними пристроями: модуль може легко з'єднуватися зі смартфонами, планшетами, комп'ютерами та іншими Bluetooth-пристроями.

Переваги:

- легкість інтеграції: простота підключення до Arduino та інших мікроконтролерів;

- гнучкість: можливість використання як у режимі майстра, так і в режимі підлеглого;

- великий діапазон налаштувань: широкий спектр налаштувань через AT команди.

Недоліки:

- обмежена дальність та швидкість: для деяких застосувань дальність в 10 метрів та швидкість Bluetooth 2.0 можуть бути недостатніми;

- безпека: Bluetooth 2.0 не має такого рівня захисту, як новіші версії Bluetooth.

Програмне забезпечення та інтерфейс: спеціально розроблена програма, яка дозволяє Arduino приймати команди через Bluetooth та керувати сервоприводами відповідно до цих команд. Інтерфейс для користувача, який дозволяє відправляти команди до роборуки. Це може бути графічний інтерфейс з кнопками та слайдерами для контролю рухів руки.

Реалізація управління: Arduino приймає сигнали від Bluetooth модуля та інтерпретує їх як команди для руху сервоприводів. Arduino використовує бібліотеку сервоприводів для генерації сигналів PWM (пульс-ширинної модуляції), які керують рухами сервоприводів.

Функціональні можливості: система дозволяє калібрувати рухи та налаштувати обмеження руху сервоприводів для запобігання механічних пошкоджень. Інтерфейс користувача спроектований таким чином, щоб управління роборукою було інтуїтивно зрозумілим і легким.

2.2.4. Потенційні джерела енергії роборуки

USB Charger (Рис. 2.8) - 2.4А: цей компонент може використовуватися для живлення Arduino Uno та інших низьковольтних компонентів роборуки, коли вона знаходиться поруч з джерелом живлення (наприклад, при підключенні до комп'ютера або мережевого адаптера).



Рис. 2.8 USB Charger

Для автономної роботи роборуки, без прив'язки до стаціонарного джерела живлення, можна розглянути наступні опції:

1. Літій-іонні або літій-полімерні акумулятори: якщо потрібна висока ємність та довгий час роботи. Вони легкі і мають високу щільність енергії.
2. Свинцево-кислотні акумулятори: якщо вага не є критичною проблемою і потрібен дешевший варіант.
3. Стандартні AA або AAA батареї: для простих або навчальних проєктів з меншими вимогами до потужності.

2.2.5. Інтерфейс управління прототипа

За інтерфейс відповідає мобільний додаток на Android від SkifiLabs, який дає змогу підключатися до Arduino через Bluetooth та керувати роборукою за допомогою ползунків.

Кнопка "CONNECT": велика синя кнопка, що використовується для ініціації підключення або відключення від роботичної руки через Bluetooth. Коли з'єднання встановлено, ця кнопка може змінювати колір або підпис, щоб показати статус з'єднання.

Ползунки для сервоприводів (Servo X, Servo Y, Servo Z, Servo G): ці елементи забезпечують інтерактивний контроль положення відповідних сервоприводів роборуки. Ползунки можуть рухатися вправо або вліво, щоб збільшити чи зменшити кут повороту або іншу керовану характеристику. Кожен ползунок відповідає за один ступінь свободи руху роборуки.

Індикатор статусу Bluetooth: текстовий рядок показує поточний статус з'єднання Bluetooth

Відображення значення ползунка (Slider value): нижче на інтерфейсі може відображатися поточне значення, вибране за допомогою ползунків

Інтерфейс чистий та простий, що сприяє легкості використання. Використовується контрастне кольорове кодування для легкого розпізнавання активних та неактивних елементів управління. Цей інтерфейс

забезпечує інтуїтивне та ефективне управління для користувачів роборуки, дозволяючи точно керувати її рухами та функціями.

2.2.6. ПЗ роботичної системи

Артикульована роботична рука в даній роботі програмується мовою C.

Мова програмування C, яка використовується для розробки роботичної руки, є однією з найбільш потужних і гнучких інструментів в руках розробника. Створена у 1972 році Деннісом Рітчі, вона залишається однією з найпопулярніших мов до сьогодні завдяки своїй здатності поєднувати простоту коду з ефективним управлінням ресурсами системи.

Програмуючи на C, розробники отримують безпосередній доступ до фізичних адрес пам'яті через вказівники, що є критично важливим для роботи з апаратним забезпеченням, як-от мікроконтролери та сенсори. Це дозволяє точно контролювати роботу обладнання та оптимізувати використання пам'яті та процесорного часу.

Для роботичної руки, програмування на C дозволяє реалізувати комплексну логіку управління сервоприводами, читання даних з сенсорів та керування комунікаціями з Bluetooth модулем. Мова C надає можливості для написання низькорівневого коду, який може безпосередньо взаємодіяти з регістрами мікроконтролера, що ідеально підходить для точного та швидкого реагування системи.

У той же час, мова C вимагає від розробника глибокого розуміння принципів роботи апаратного забезпечення та уважного управління пам'яттю, оскільки вона не має вбудованого сміттєзбору або високорівневих абстракцій, які є у більш нових мовах.

Використання C для програмування роботичної руки є прикладом того, як класичні технології можуть бути ефективно застосовані до сучасних викликів в області робототехніки.[6]

ВИСНОВКИ ДО РОЗДІЛУ 2

Проектування роботичної руки є комплексним процесом, який об'єднує механічне конструювання, електроніку, програмування та інтерфейс користувача. Основою системи стала платформа Arduino Uno Rev3, яка через свою гнучкість та доступність стала оптимальним вибором для контролю сервоприводів, що забезпечують маніпуляції рукою. Використання Bluetooth-модуля HC-05 дало можливість розробки бездротового інтерфейсу управління, що значно розширило функціонал і зручність використання руки.

Завдяки мобільному додатку, розробленому SkyfiLabs, забезпечено інтуїтивно зрозуміле управління за допомогою графічних ползунків, які дозволяють точно контролювати кожен згин руки. Цей інтерфейс не лише спрощує взаємодію з рукою, але й дозволяє оператору отримувати зворотний зв'язок про стан з'єднання та позиції сервоприводів.

Були опрацьовані варіанти живлення роборуки, що гарантує її ефективну і тривалу роботу в автономному режимі.

Таким чином, детальне проектування артикульованої роборуки не лише демонструє технічну виконаність концепції, але й створює основу для подальших розробок та інновацій у сфері робототехніки.

РОЗДІЛ 3

РОЗРОБКА І ТЕСТУВАННЯ ПРОТОТИПУ РОБОРУКИ

3.1. Конструювання роборуки

У цій частині дипломної роботи починається перехід від теоретичних основ до практичної реалізації проекту дистанційно керованої, артикульованої роботичної руки. Основна увага буде приділена процесу збирання руки та детальному аналізу роботи. Ми розглянемо кожен етап конструювання, від вибору компонентів до монтажу і налаштувань.

Головним кроком буде детальний опис процесу збірки, включаючи механічне складання частин руки, основних команд коду та інтерфейсу користувача.

Аналіз роботи та результатів допоможе зрозуміти потенціал та обмеження конструкції, а також надасть важливі інсайти для подальших вдосконалень.

Таким чином, ця частина роботи є ключовою для демонстрації практичного застосування теоретичних знань та навичок, отриманих в ході навчання, та відіграє важливу роль у загальному контексті дипломної роботи.

Розробка прототипу починається з основи (Рис. 3.1): основа має бути витривала, відповідати габаритам руки, легка для переносу, дешева, але стійка. Як було визначено в минулому розділі, за основу можна взяти дерев'яну дощечку, просверливши отвори для болтів, вкрутивши їх та закріпивши гайками.

Кафедра КІТ				НАУ 23 10 36 000 ПЗ			
	ПІБ			РОЗДІЛ 3. РОЗРОБКА ТА ТЕСТУВАННЯ РОБОРУКИ	Літ.	Аркуш	Аркушів
Розроб.	Козловцев Д.С.					85	16
Керівник	Зудов О.М.				ТП-215М - 122		
Н.Контр.	Толстікова О.В.						



Рис. 3.1 Матеріали основи

Таким чином буде створена стійка основа. Далі, використовуючи багатофункціональний кронштейн, почнемо процес монтування плеча роборуки до основи (Рис. 3.2).

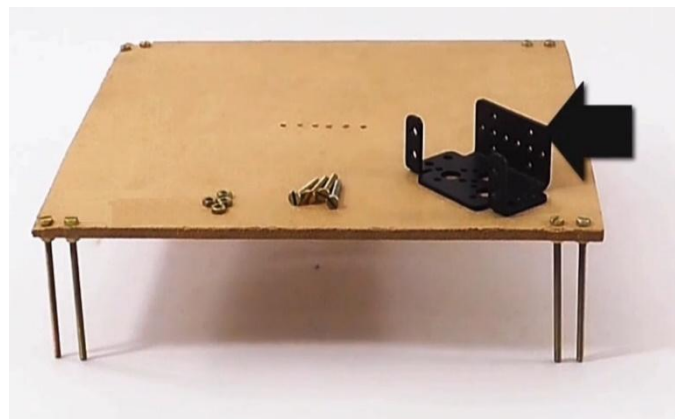


Рис. 3.2 Стійка основа

Кронштейн потрібно закріпити таким чином, щоб його «руки» дивилися вгору, а 3 основні отвори були перпендикулярні основі (Рис. 3.3).

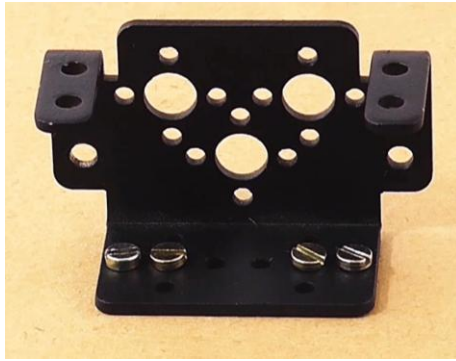


Рис. 3.3 Прикріплений кронштейн плеча

Перед кріпленням актуаторів, потрібно налаштувати нульове положення.

Нульове положення сервоприводів – це початкове або вихідне положення сервомотора, яке визначається як відлік для всіх інших положень. Для багатьох сервоприводів, особливо в моделюванні та невеликих робототехнічних проектах, нульове положення відповідає середньому положенню, зазвичай при куті 90 градусів (у сервоприводів з діапазоном руху від 0 до 180 градусів). Нульове положення сервоприводів – це початкове або вихідне положення сервомотора, яке визначається як відлік для всіх інших положень.

У цьому положенні сервопривід не виконує жодного руху і знаходиться в стані спокою. Коли система керування відправляє сигнал до сервопривода, вона використовує нульове положення як відправну точку для визначення, на який кут і в якому напрямку необхідно перемістити вал сервомотора.

Налаштування проводиться через підключення сервомоторів до Arduino і програмного коду, який описаний в додатку А.

Виконавши відладку, можна закріплювати серво, закінчуючи конструювання основи плеча (Рис. 3.4).

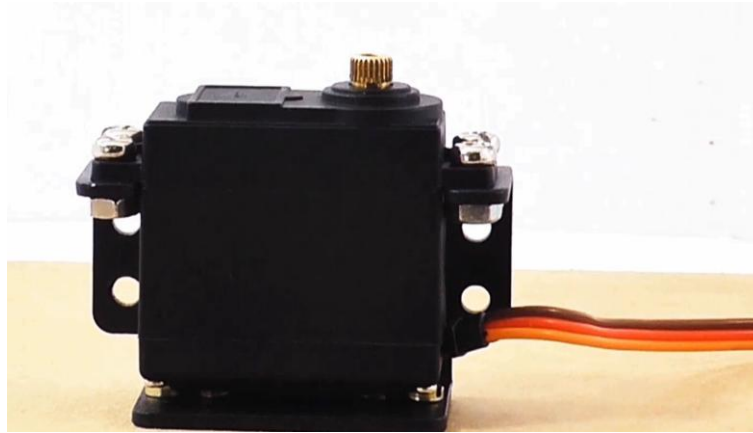


Рис. 3.4 Актуатор плеча вмонтований в систему

Наступним кроком буде монтування ліктя (Рис. 3.5). Для цього потрібно закріпити другий багатофункціональний кронштейн таким чином, щоб його «руки» були направлені вертикально вгору та виходили на праву частину основи:

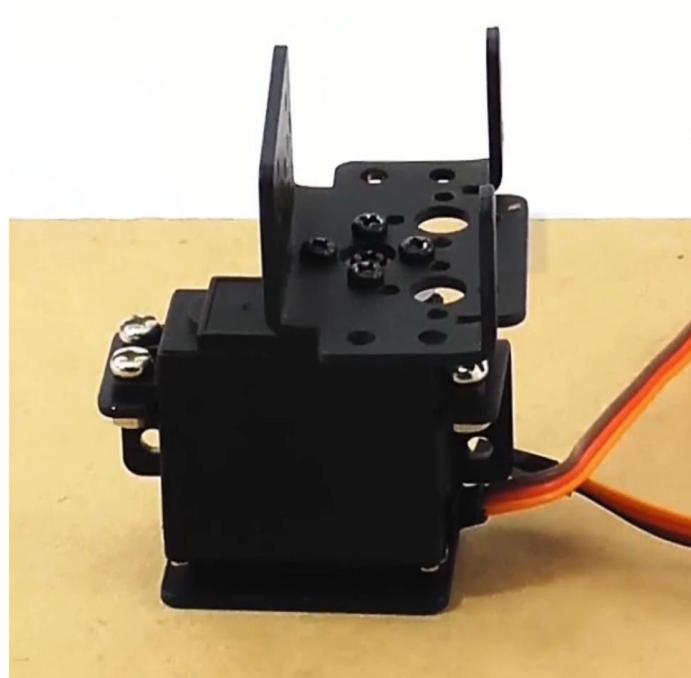


Рис. 3.5 Кронштейн основи ліктя

Важливо помітити, що потрібно вставити гвинт в перший центральний отвір, цей гвинт буде використаний як додаткове укріплення ліктя (Рис. 3.6).



Рис. 3.6 Гвинт для кріплення ліктя

Після закріплення другого сервомотору, конструювання основи ліктя (Рис. 3.7) можна вважати закінченим.

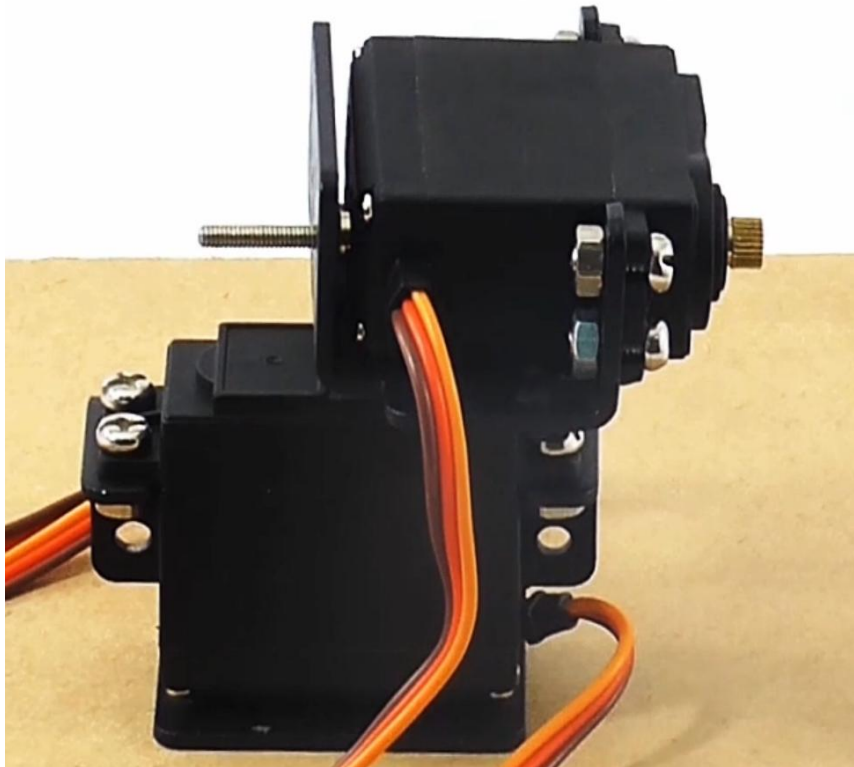


Рис. 3.7 Основа ліктя

Тепер, для самого ліктя ми використаємо конструкцію з двох U-подібних кронштейнів (Рис. 3.8).



Рис. 3.8 Конструкція ланки для ліктя

Для закріплення ліктя (Рис. 3.9), монтування буде проведене за допомогою гвинта, зазначеного вище, та рога актуатора.



Рис. 3.9 Монтування ланки ліктя

Для конструювання зап'ястя, потрібно закріпити третій багатофункціональний кронштейн на основу нашого ліктя, в тому самому положенні, що і багатофункціональний кронштейн перед ним (Рис. 3.10).



Рис. 3.10 Кронштейн зап'ястя

Вкручуємо додатковий гвинт (Рис. 3.11), як і минулого разу:



Рис. 3.11 Додавання допоміжного гвинта

Під'єднуємо третій серво з високим моментом обертання (Рис. 3.12):

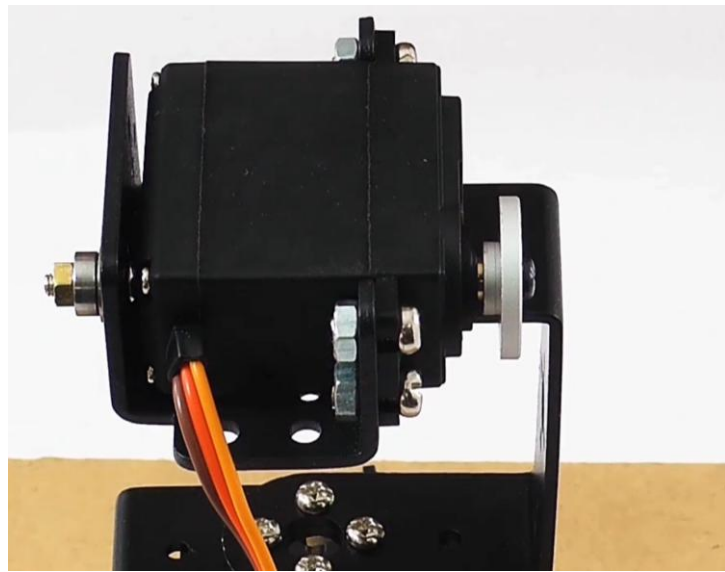


Рис. 3.12 Основа зап'ястя роборуки

Для конструювання повного маніпулятора використаємі L-кронштей, сервопривод SG90 та, власне, сам маніпулятор (Рис.3.13):



Рис. 3.13 Маніпулятор

Монтуємо маніпулятор до зап'ястя, використовуючи кронштейн у формі U (Рис. 3.14):

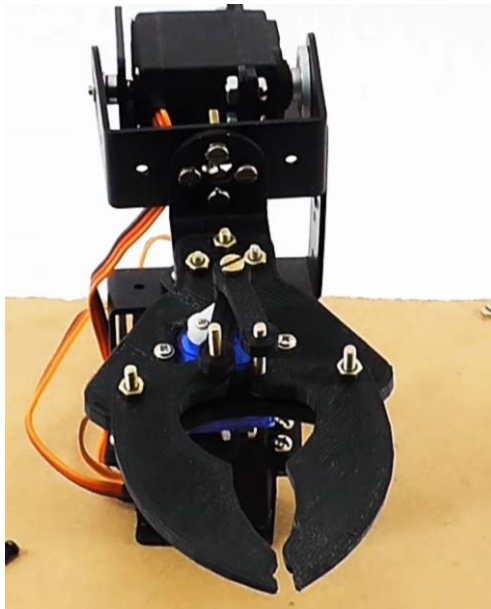


Рис. 3.14 Монтування маніпулятора

На цьому конструювання механічної структури роборуки можна вважати закінченим.

Для руху роботичної руки потрібне живлення і алгоритм, який буде сприяти цьому руху. Код дивитися в додатку А. Схема підключення до ланцюга живлення наведена на рис. 3.15:

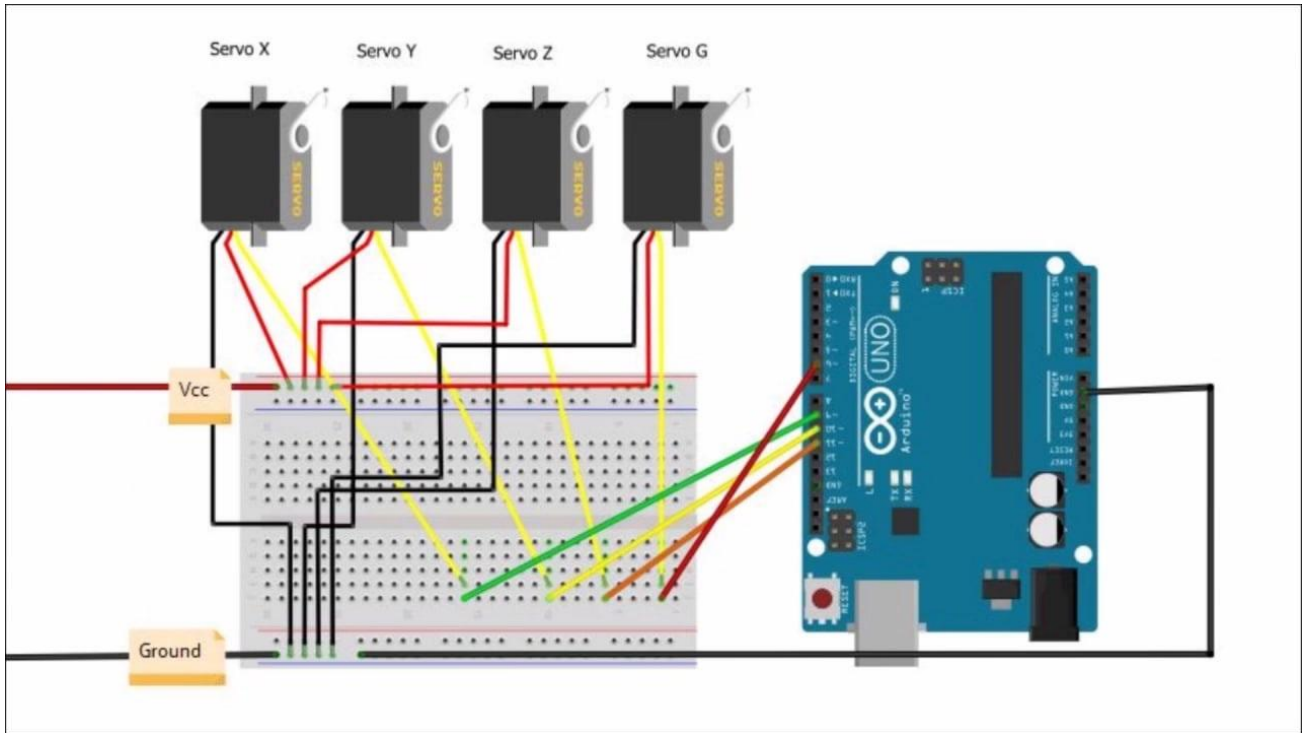


Рис. 3.15 Схема підключення роботу до ланцюга живлення

Останній етап – додавання до прототипу дистанційного керування через Bluetooth (Рис. 3.16). Алгоритм комунікації рки через модуль дивитись в додатку А.

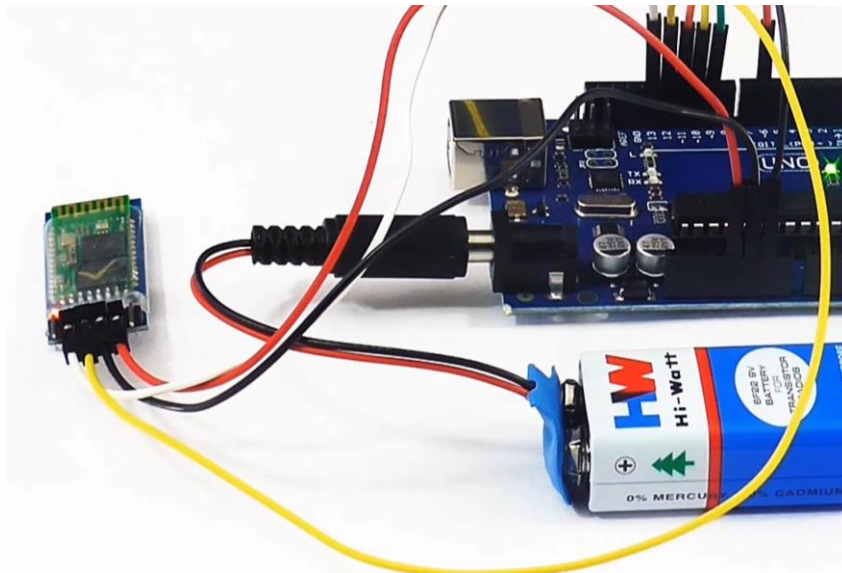


Рис. 3.16 Підключення Bluetooth модуля

3.2. Програмне забезпечення

SoftwareSerial

- функція з бібліотеки "SoftwareSerial" Arduino;
- опис: створює об'єкт для серійного зв'язку;
- синтаксис: `SoftwareSerial(rxPin, txPin);`
- параметри:
 - rxPin - пін, на який приймаються серійні дані;
 - txPin - пін, на якому передаються серійні дані;
- повертає: None;
- використання - серійний зв'язок через Bluetooth;
- створення "bluetooth" як об'єкту (може бути будь-яка назва);
- Використовуються піни 12 та 13.

bluetooth.available

- функція з бібліотеки "SoftwareSerial" Arduino;
- опис: отримує кількість байтів, доступних для читання з програмного серійного порту;
- синтаксис: `bluetooth.available();`
- параметри: жодних;
- повертає: кількість байтів, доступних для читання.

bluetooth.read

- функція з бібліотеки "SoftwareSerial" Arduino;
- опис: читає вхідні дані, які були отримані на піні RX програмного серійного порту;
- синтаксис: `bluetooth.read();`
- параметри: жодних;
- повертає: перший байт вхідних серійних даних у форматі цілого числа;

- зчитує та зберігає прочитаний байт до змінної.

`Serial.print()`

- вбудована функція;
- опис: виводить дані на серійний монітор у форматі ASCII;
- синтаксис: `Serial.print("Символи");`
- параметри: значення для друку будь-який типів даних;
- повертає: `None`;
- `Serial.println()` - для друку в наступному рядку.

`servo.attach`

- функція з бібліотеки "Servo" Arduino;
- опис: приєднує змінну сервоприводу до піна;
- синтаксис: `servo.attach(pin);`
 - `Servo` змінюється на назву об'єкта;
- параметри: цифровий вхід/вихід піна на Arduino, який буде підключений до сервоприводу;
 - пін може бути під номером 3, 5, 6, 9, 10 та 11 - цифрові вхід/вихід піни з PWM виводом;
- повертає: `None`.

`map`

- вбудована функція;
- опис: перевизначає число з одного діапазону в інший;
- синтаксис: `map(value,fromLow,fromHigh,toLow,toHigh);`
- параметри:
 - `value` - число/кількість для перевизначення;
 - `fromLow, fromHigh` - межі поточного діапазону значення
 - `toLow, toHigh` - межі цільового діапазону значення;
- повертає: перевизначене значення;

- відображає отримані дані сервоприводу в кут сервоприводу:
"servo1" (діапазон: 2000 до 2180) на "servo1" (діапазон: 0 до 180).

`servo.write()`

- функція з бібліотеки "Servo" Arduino;

- опис: записує значення до сервоприводу;

- синтаксис: `servo.write(кут)`;

- параметри:

- кут: ціле значення кута в градусах для запису в сервопривод,
від 0 до 180;

- повертає: None.

Для кращого розуміння алгоритму коду, представлена блок-схема
(Рис. 3.17):

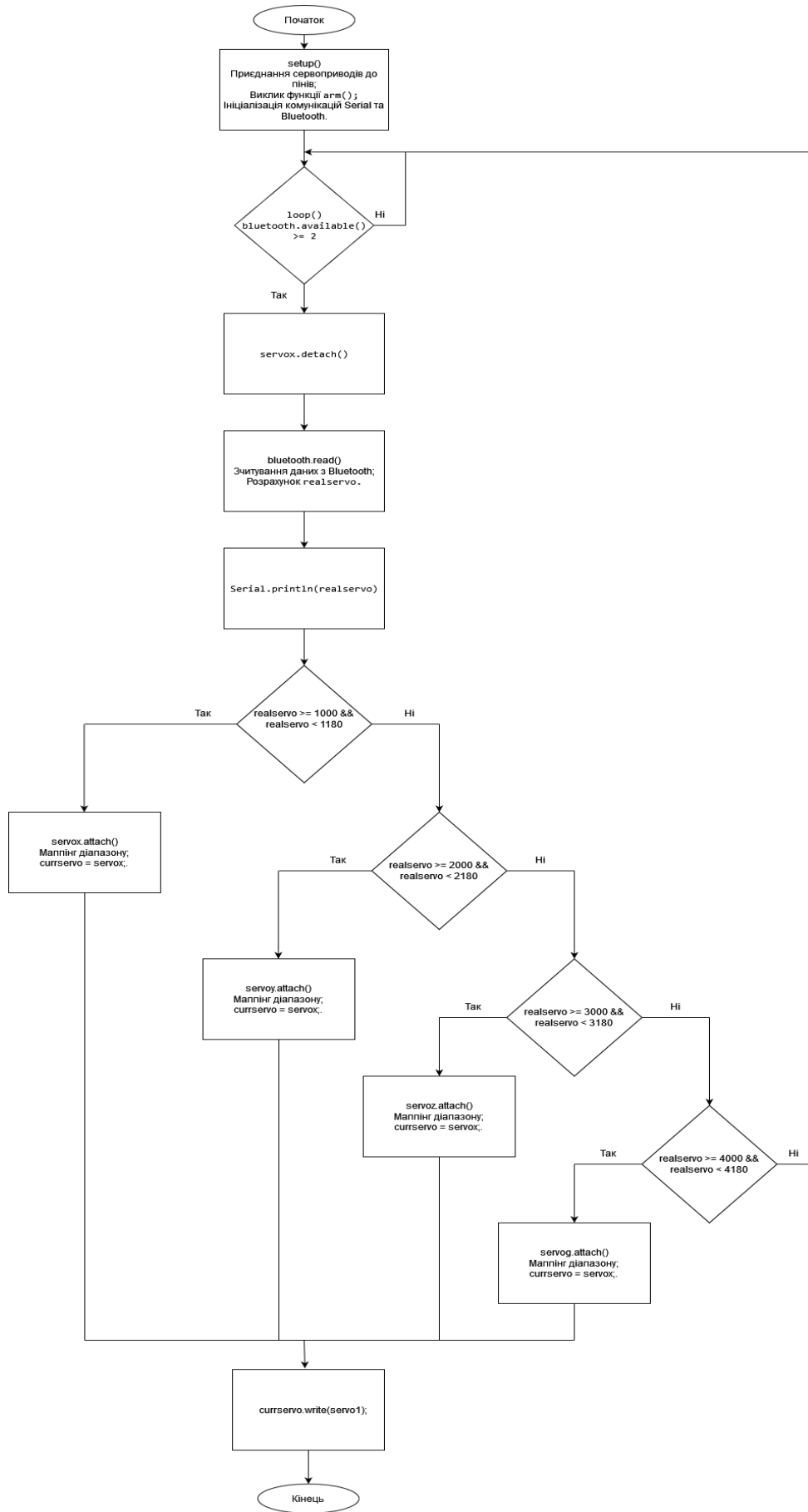


Рис. 3.17 Блок-схема алгоритму

3.3. Інтерфейс користувача та юз-кейси

Оскільки інтерфейс користувача був описаний в минулому розді, розглянемо його реалізацію і декілька юз-кейсів, котрі вказують на вплив інтерфейсу управління.

При середньому значенні слайдерів в 3079 - роборука відкриває клешню, припіднімає зап'ястя та трохи змінює положення плеча з ліктем (Рис. 3.18).

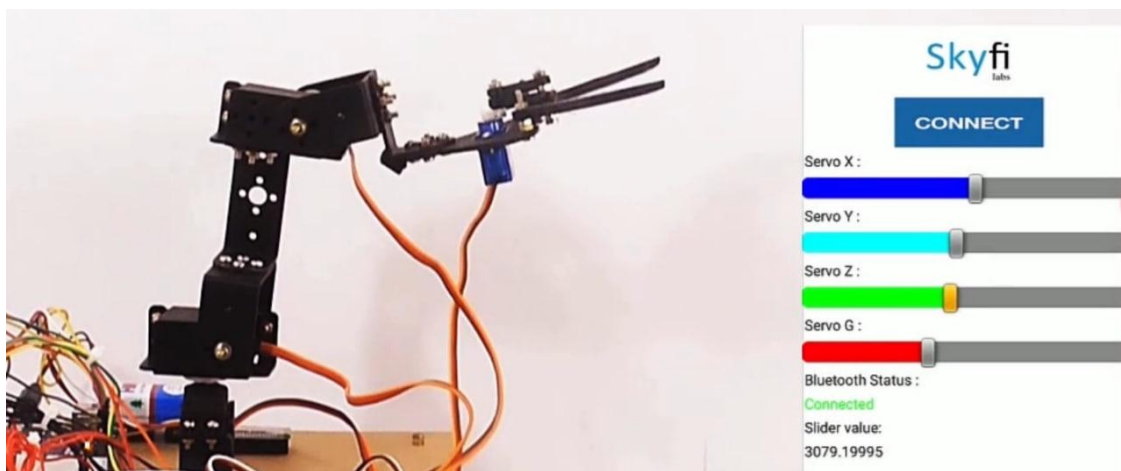


Рис. 3.18 Положення роборуки номер один

При значенні в 2111 можемо побачити сильний нахил ліктя, який також відображається у застосунку просунутим далеко вправо синім ползунком (Рис. 3.19).

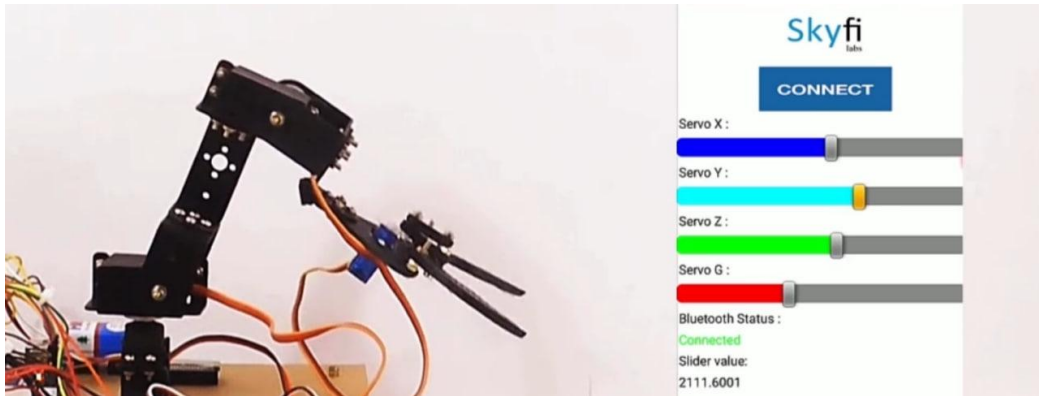


Рис. 3.19 Положення роборуки номер два

Після маніпуляцій зі слайдерами, роборука швидко захоплює легкий предмет, зажимаючи в клешні. Для цього слайдер Servo G викручується майже до нуля (Рис 3.20).

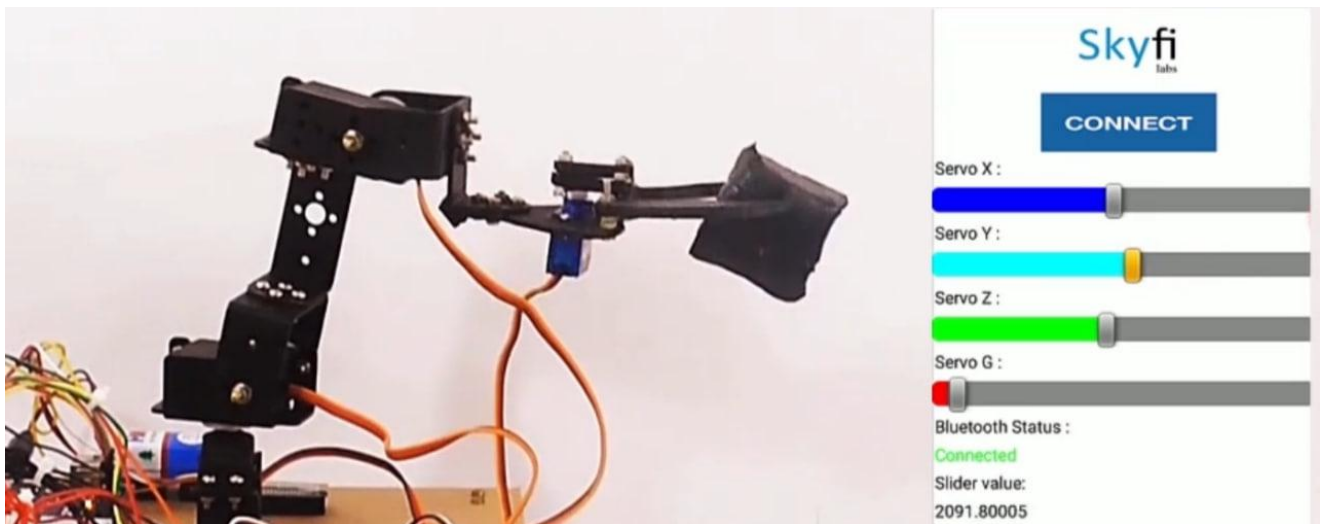


Рис. 3.20 Положення роборуки номер три

В якості останнього прикладу, повернемо плече до камери (Рис 3.21):

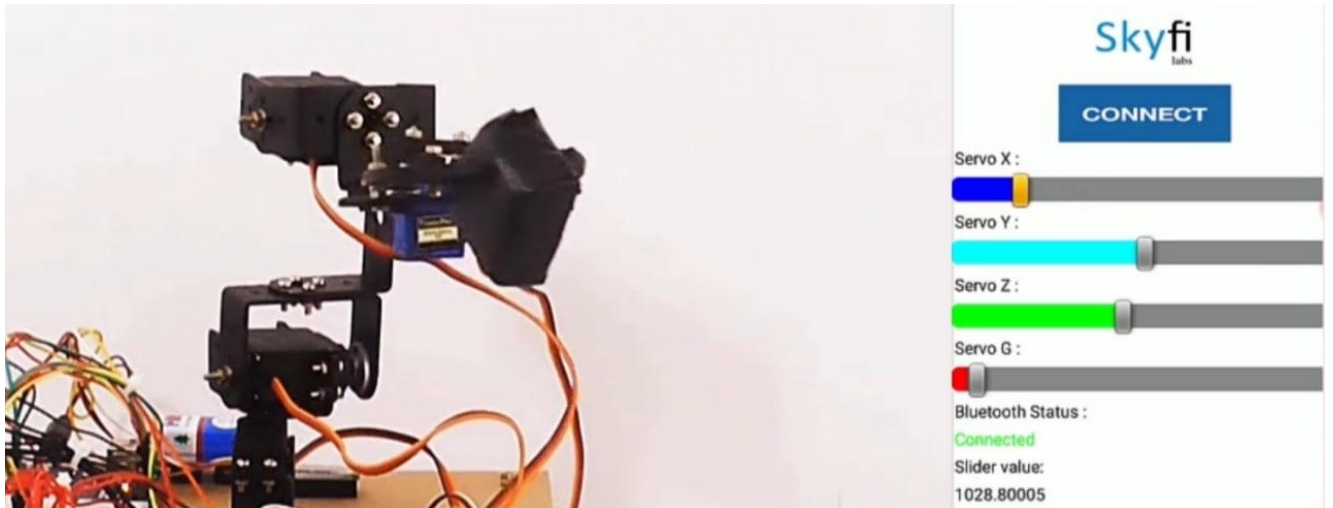


Рис. 3.21 Положення роборуки номер чотири

3.4. Аналіз прототипу

Потенціал та результати:

1. Гнучкість та адаптивність: роботична рука показала високий рівень адаптивності до різних завдань, завдяки точному контролю сервоприводів та можливості дистанційного управління.

2. Точність рухів: використання Arduino та якісних сервоприводів забезпечило високу точність у рухах, що є критично важливим для застосувань, де необхідна детальна маніпуляція.

3. Інтуїтивний інтерфейс користувача: мобільний додаток забезпечив зручне та інтуїтивно зрозуміле управління.

Обмеження конструкції:

1. Обмежена вантажопідйомність: через використання стандартних сервоприводів вантажопідйомність руки може бути обмеженою.

2. Обмеження енергії: автономність руки залежить від ємності батареї, що може стати обмежувальним фактором у тривалих або енергоємних завданнях.

3. Фізичні обмеження конструкції: механічна конструкція має обмеження щодо максимальних кутів повороту та довжини руху.

Інсайти для вдосконалень:

1. Використання розширених сенсорів: інтеграція додаткових сенсорів, таких як силачів або торк-сенсорів, може значно підвищити функціональність та можливості руки.

2. Удосконалення дизайну для більшої вантажопідйомності: перегляд конструкції з метою збільшення вантажопідйомності, можливо за рахунок використання більш потужних сервоприводів або зміцнення каркасу.

3. Оптимізація енергоспоживання: розробка більш енергоефективних алгоритмів керування може допомогти збільшити час автономної роботи.

ВИСНОВКИ ДО РОЗДІЛУ 3

У третьому розділі кваліфікаційної роботи, присвяченому конструюванню артикульованої роботичної руки на базі Arduino з дистанційним управлінням через Bluetooth, було розглянуто комплексний процес від задуму до втілення. Було досягнуто успішної інтеграції ключових компонентів: сервоприводів для забезпечення рухливості руки, плати Arduino Uno як центрального контролера та Bluetooth модуля HC-05 для забезпечення дистанційного зв'язку. Конструкція була ретельно спланована для забезпечення оптимальної роботи та ефективності, з урахуванням механічної стійкості та електронного управління.

Були продемонстровані головні команди ПЗ, їх інтеграція та властивості. Також було визначено основні алгоритми за якими здійснюється рух прототипу.

Аналіз прототипу виявив важливі властивості та можливі області вдосконалення. Точність рухів та надійність управління через мобільний додаток демонстрували високу ефективність системи. Водночас були виявлені обмеження, зокрема у вантажопідйомності та тривалості автономної роботи, які вказують на потребу подальших технічних вдосконалень.

ВИСНОВКИ

Кваліфікаційна робота надає всебічний огляд розвитку робототехніки, висвітлюючи ключові характеристики та принципи роботи роботизованих механічних систем, включаючи їх механічну структуру та актуатори. Значна увага приділяється системам управління, сенсорним системам, джерелам енергії, інтерфейсам управління та ступеням свободи, що є фундаментальними для розуміння та ефективного проектування роботичних систем.

В підрозділі про артикульовану роботичну руку розглянуто її основні сегменти, роль серводвигунів у русі руки, загальний контроль руху, апаратне та програмне забезпечення, систему комунікації, зокрема Bluetooth модуль, і кінематику роборуки. Також представлено порівняльний аналіз існуючих аналогів, що демонструє місце розробки в контексті сучасного стану робототехніки.

У другому розділі розглядається процес проектування роборуки, включаючи вимоги до прототипу, архітектуру та компоненти.

Особлива увага приділяється системі управління, потенційним джерелам енергії, інтерфейсу управління та програмному забезпеченню роботичної системи.

У третьому розділі зосереджено увагу на практичному конструюванні роборуки та глибокому аналізі її прототипу. Цей аналіз включає в себе оцінку функціональності, ефективності, потенційних обмежень та можливостей для подальших вдосконалень.

В цілому, ця кваліфікаційна робота демонструє глибоке розуміння принципів та практик у сфері робототехніки, а також підкреслює важливість інноваційного підходу до проектування та розвитку роботизованих систем.

Розробка артикульованої роботичної руки на базі Arduino з Bluetooth управлінням є свідченням потенціалу та можливостей сучасної робототехніки, відкриваючи шлях для нових досліджень і технічних інновацій в цій області.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Навчальний посібник: Ford M. Rise of the Robots: Technology and the Threat of a Jobless Future. New York: Basic Books, 2019. 352 с.
2. Навчальний посібник: Sergiyenko O., Flores-Fuentes W., Mercorelli P. Machine Vision and Navigation. Berlin: Springer, 2019. 280 с.
3. Навчальний посібник: Mckinnon P. Robotics: Everything You Need to Know About Robotics from Beginner to Expert. New York: CreateSpace Independent Publishing Platform, 2019. 198 с.
4. Навчальний посібник: Colomé A., Torras C. Reinforcement Learning of Bimanual Robot Skills. Berlin: Springer, 2019. 150 с.
5. Навчальний посібник: Ceceri K. Making Simple Robots: Exploring Cutting-Edge Robotics with Everyday Stuff. San Francisco: Maker Media, Inc, 2019. 225 с.
6. Навчальний посібник: Murphy R.R. Introduction to AI Robotics. 2nd ed. Cambridge: MIT Press, 2019. 544 с.
7. Навчальний посібник: Spong M.W., Hutchinson S., Vidyasagar M. Robot Modeling and Control. New York: Wiley, 2019. 496 с.
8. Навчальний посібник: Joshi R., Ziegler M., Kumar A., Alarcon E. From Artificial Intelligence to Brain Intelligence. New York: Springer, 2019. 320 с.
9. Навчальний посібник: Ceceri K. Robotics: Discover The Science and Technology of the Future with 20 Projects. Beverly: Nomad Press, 2019. 144 с.
10. Стаття: Струтинська О. Використання робототехніки та 3D-технологій у розвитку STEM-освіти. Київ: Наукова думка, №7. С. 1-14.
11. Навчальний посібник: Струтинська О. Актуальність впровадження освітньої робототехніки в українські школи. Київ: Наукова думка, 2019. С. 324-341.
12. Стаття: Palli G., Melchiorri C., Meattini R., editors. Human-Friendly Robotics 2021. Springer, 2022. No. 14. С. 1-15.

13. <https://appuals.com/how-to-make-a-diy-arduino-and-bluetooth-controlled-robotic-arm/>
14. <https://lambdageeks.com/articulated-robots/>
15. <https://robocademy.com/2020/04/21/robot-kinematics-in-a-nutshell/>

Код функціонування роборуки.

```
#include <Servo.h>
#include <SoftwareSerial.h>

SoftwareSerial bluetooth(12, 13);
Servo servox;
Servo servoy;
Servo servoz;
Servo servog;

void arm(){
  servox.write(90);
  servoy.write(90);
  servoz.write(40);
  servog.write(0);
  delay(1000);
}

void setup() {
  servox.attach(9);
  servoy.attach(10);
  servoz.attach(11);
  servog.attach(6);
  arm();
  Serial.begin(9600);
  bluetooth.begin(9600);
}
```

```

void loop() {

  if (bluetooth.available() >= 2){
    servox.detach();
    servoy.detach();
    servoz.detach();
    servog.detach();

    unsigned int servopos = bluetooth.read();
    unsigned int servopos1 = bluetooth.read();
    unsigned int realservo = (servopos1 * 256) + servopos;

    Serial.println(realservo);
    int servo1 = realservo;
    Servo currservo;

    if (realservo >= 1000 && realservo < 1180) {
      servox.attach(9);
      servo1 = map(servo1, 1000, 1180, 0, 180);
      currservo = servox;
    }

    if (realservo >= 2000 && realservo < 2180) {
      servoy.attach(10);
      servo1 = map(servo1, 2000, 2180, 0, 180);
      currservo = servoy;
    }
  }
}

```

```
if (realservo >= 3000 && realservo < 3180) {
  servoz.attach(11);
  servo1 = map(servo1, 3000, 3180, 0, 180);
  currservo = servoz;
}

if (realservo >= 4000 && realservo < 4180) {
  servog.attach(6);
  servo1 = map(servo1, 4000, 4180, 0, 180);
  currservo = servog;
}

currservo.write(servo1);

}
}
```