

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Інститут інформатики та радіоелектроніки,
факультет радіоелектроніки та телекомунікацій
(повне найменування інституту, назва факультету)

Кафедра інформаційних технологій електронних засобів
(повна назва кафедри)

Пояснювальна записка

до магістерської роботи

магістр

(рівень вищої освіти (освітньо-кваліфікаційний рівень))

на тему «Методи моделювання робота-маніпулятора»

Виконав: студент VI курсу, групи РТ-112м
спеціальності (напряму підготовки)
_172 «Телекомунікації та
радіотехніка»

(код і назва напрямку підготовки, спеціальності)

Шапгала С.В.

(прізвище та ініціали)

Керівник Пирожок А.В.

(прізвище та ініціали)

Рецензент Чорнобородов М.П.

(прізвище та ініціали)

м. Запоріжжя
2017 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет
(повне найменування вишого навчального закладу)

Інститут, факультет Інститут інформатики та радіоелектроніки, факультет радіоелектроніки та телекомунікацій
Кафедра інформаційних технологій електронних засобів
Рівень вищої освіти (освітній ступінь) магістр
Спеціальність 172 Телекомунікації та радіотехніка
(код і назва)
Напрямок підготовки Радіоелектронні апарати і засоби
(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри К.ф.т.ч. ШТЕЗ
Т. М. Шенко
"19" 12 2017 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ (РОБОТУ) СТУДЕНТУ

Шайтала Євген Вікторович
(прізвище, ім'я, по батькові)

Тема проекту (роботи) Методи моделювання роботи маніпулятора

Виконав проекту (роботи) Тітонюк Андрій Володимирович, доцент кафедри ШТЕЗ, к.т.н.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вишого навчального закладу від "25" жовтня 2017 року № 415

Термін подання студентом проекту (роботи) 20.12.2017

Вимоги до проекту (роботи) важконадійність не більше 500 кг, площа роботи довж. не більше 3500 мм, висота роботи не більше 2500 мм

Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розв'язати) класифікація робіт, опис вибраного робота JRB 460, розгляд способів творення різноманітної моделі, голосове управління, охорона праці та безпека та безпека в надзвичайних ситуаціях

Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

16 слайдів

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконання завдання
Основний	доцент кафедри Тироненко А.В.	<i>Тироненко</i>	<i>Тироненко</i>
Спеціалізо-організаційно-методичне	ст. викл. Остапенко В.В.	05.11.09.17	05.15.12
охорони праці та безпеки у складі сист.	ст. викл. Коробко О.В.	05.11.17	05.11.17
Корпоративна	ст. викл. ЗОселева Т.Є.		05.11.17

7. Дата видачі завдання 02 вересня 2017 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітки
1.	Аналіз типів робіт	09.09.17 - 15.09.17	
2.	Вибір роботи для дослідження	15.09.17 - 20.09.17	
3.	Опис вибраного промислової роботи	20.09.17 - 03.10.17	
4.	Розробка математичної та фізичної моделей	03.10.17 - 25.10.17	
5.	Обґрунтування необхідності голосового управління	25.10.17 - 28.10.17	
6.	Реалізація програмного забезпечення	20.11.17 - 23.11.17	
7.	Тестування програмного забезпечення	23.11.17 - 10.12.17	
8.	Охорона праці	23.11.17 - 10.12.17	
9.	Економічне обґрунтування	01.05.17 - 19.12.17	
10.	Консультації по фізичній роботі з керівником	20.12.17	
11.	Захист магістерської роботи		

Студент

Тироненко
(підпис)Шарман С.В.
(прізвище та ініціали)

Керівник проекту (роботи)

Тироненко
(підпис)Тироненко А.В.
(прізвище та ініціали)

РЕФЕРАТ

Дипломний проект: 108 с., 28 рис., 12 табл., 12 джерел, 2 додатків.

Темою дипломної роботи є методи моделювання робота-маніпулятора.

Мета роботи – створення моделей та розробка системи голосового керування роботом - маніпулятором на базі Arduino.

В магістерській роботі створено фізичну та математичну модель промислового робота маніпулятора IRB 460. Система керування роботом базується на платформі Arduino. Розроблено програмне забезпечення для голосового керування роботом. Для розпізнавання речі використано стандартний сервіс Google. Для передачі інструкцій керування до платформи Arduino використовується Bluetooth. Така система може встановлюватися на промислові роботи та використовуватися у небезпечних середовищах або для віддаленого управління роботом через мережу інтернет.

РОБОТ - МАНІПУЛЯТОР, ARDUINO, СИСТЕМА УПРАВЛІННЯ ГОЛОСОМ, ФІЗИЧНА МОДЕЛЬ, ПРОГРАМУВАННЯ.

ЗМІСТ

РЕФЕРАТ.....	4
ВСТУП.....	7
1 ОСНОВИ РОБОТОТЕХНІКИ	8
1.1 Основні поняття і визначення роботів і робото технічних пристроїв.....	8
1.2 Класифікація промислових роботів.....	10
1.3 Рука робота	17
1.4 Класифікація роботів.....	18
1.5 Структура і геометрія маніпуляторів	23
1.6 Механіка промислових роботів	25
1.7 Виконавчі механізми і приводи промислових роботів.....	26
1.8 Побудова рухових систем роботів.....	28
1.9 Захватні пристрої та передавальні механізми.....	31
2 ОПИС ПРОМИСЛОВОГО РОБОТА IRB 460	35
3 РОГЛЯД МОЖЛИВОСТІ СТВОРЕННЯ ФІЗИЧНОЇ МОДЕЛІ РОБОТА ЗА ДОПОМОГОЮ ОПЕНСОРС ПРОЕКТУ MEARM	40
3.1 MeArmRobotArm	40
3.2 Сервоприводи SG-90	41
3.3 Складання	43
4 ГОЛОСОВЕ УПРАВЛІННЯ	45
4.1 Аналіз середовищ програмування Arduino	45
4.2 Принцип роботи	48
5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	53
5.1 Аналіз потенційних небезпечних чинників при проведенні науково-дослідної роботи.....	53
5.2 Заходи щодо забезпечення безпеки.....	54
5.3 Заходи щодо забезпечення виробничої санітарії та гігієни праці	56

5.4 Заходи з пожежної безпеки	60
5.5 Заходи з цивільного захисту	62
6 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА	65
6.1 Планування розробки програмного виробу	66
6.2 Визначення витрат на розробку програми	68
6.2.1 Розрахунок основної заробітної плати	68
6.2.2 Розрахунок додаткової заробітної плати	69
6.2.3 Відрахування на єдиний соціальний внесок	70
6.2.4 Визначення витрат на матеріали	70
6.2.5 Витрати на спеціальне устаткування	71
6.2.6 Накладні витрати	75
6.3 Бальна оцінка економічної ефективності науково-дослідної роботи	76
6.4 Оцінка економічної ефективності	79
ВИСНОВОК	80
ПЕРЕЛІК ПОСИЛАНЬ	81
Додаток А	83
Додаток Б	103

ВСТУП

В даний час робототехніка набирає все більшої популярності у всьому світі і більшість країн вже давно використовує її в своєму виробництві, будівництві, військовій справі, наукових дослідженнях і навіть у повсякденному житті. Завдяки виходу технології нейронних-мереж до ринку робототехніки очікується скачок як у якості виготовлення роботів, так і збільшення їх можливостей що до виконання неосновних задач, тобто універсальності. Але це може призвести і до збільшення складності їх управління.

Тому особливо актуальною є тема розробки промислових роботів та маніпуляторів які можуть виконувати команди за допомогою інтуїтивно зрозумілих голосових команд.

Також данні наробітки можуть використовуватися для управління роботами які працюють у небезпечних середовищах або для віддаленого управління роботом через мережу інтернет.

Мета роботи – створення моделей та розробка системи голосового керування роботом - маніпулятором на базі Arduino.

Таким чином, для досягнення мети необхідно виконати наступні задачі:

- дослідження типів роботів;
- вибір конструкції робота маніпулятора;
- дослідження фізичної та математичної моделі робота;
- побудова макета для подальших тестувань;
- розробка алгоритму для взаємодії(прийому, обробці, конвертації інформації) мобільного телефона на базі Android з Arduino;
- розробка програмного забезпечення системи виконання голосового керування.

1 ОСНОВИ РОБОТОТЕХНІКИ

1.1 Основні поняття і визначення роботів і робото технічних пристроїв

У довідковій літературі [1] робот описується як автоматична машина, що включає програмований пристрій управління та інші технічні засоби, що забезпечують виконання тих чи інших дій, властивих людині в процесі його трудової діяльності і найбільш досконалий робот здатний самостійно вирішувати завдання самоврядування, адаптації до умов зовнішнього середовища і виконувати комплекс трудових дій. Загальною ознакою роботів є можливість швидкого переналагодження або перепрограмування для автоматичного виконання різних дій, які були передбачені програмою. В Японії, під промисловим роботом розуміють автомати зі змінною програмою, які використовуються для автоматизації ручної роботи з метою покращення трудових показників. За цим визначенням промисловий робот, як і людина, наділений трьома можливостями: функціональними, розумовими та фізичними. При цьому під функціональними можливостями розуміють універсальність, здатність пересуватися в просторі(для роботів усіх поколінь), а під розумовими можливостями - здатності відчуття і сприйняття, пам'ять і логіку, а також здатність до навчання(для роботів 2 та 3 покоління). Наведемо визначення промислового робота, прийняте в нашій країні. Промисловий робот (ПР) - це автономно функціонуюча автоматична машина, що складається з маніпулятора, що має кілька ступенів рухливості, і пристрої програмного управління, і призначена для виконання основних і допоміжних операцій виробничого циклу без участі людини. У технічній літературі часто використовується коротке визначення: промисловий робот - перепрограмований автоматичний маніпулятор промислового застосування. А також, маніпулятор - оснащене робочим органом виконавчий пристрій для

виконання рухових функцій, аналогічних функціям руки людини при переміщенні об'єктів в просторі. Об'єктом маніпулювання називають тіло, що переміщується в просторі маніпулятором. До об'єктів маніпулювання відносять заготовки, деталі, інструмент, технологічне оснащення і т.д.

Конструкція маніпулятора, виконаного у вигляді механічної руки, кінематично аналогічна руці людини (рис. 1.1).

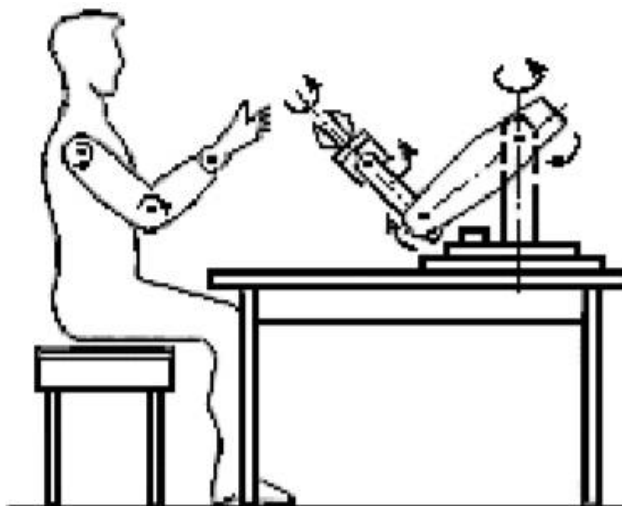


Рисунок 1.1- Схема кінематичної аналогії руки людини і механічної руки маніпулятора промислового робота

Промислові роботи є дуже важливими для тих підприємств які намагаються покращити свої трудові показники. Шляхом організації комплексно-автоматизованих ділянок і цехів вони можуть не тільки виконувати якісь поодинокі завдання а і об'єднувати технологічне обладнання в координовано-працюючі комплекси різного рівня. Такі комплекси можуть забезпечити оптимальну структуру технологічних процесів в широкому діапазоні серійності виробів. Відкриваються нові можливості ефективної організації та управління серійного виробництва на основі комплексної автоматизації інформаційних та матеріальних потоків з використанням промислових роботів і комп'ютерів. Така форма комплексної автоматизації отримала назву інтегрованої виробничої системи. Такі системи

мають швидку зміну моделей виробництва, тобто переходу від одного типорозміру до іншого і тому мають володіти технологічною гнучкістю яка надасть можливість як найшвидше настроїти виробництво під необхідні вимоги. Робототехнічні пристрої складають основу таких систем автоматизації, які отримали назву гнучкої виробничої системи. Гнучка виробнича система (ГВС) може бути складена з модулів двох типів: модуль у вигляді роботизованого технічного комплексу (РТК), в якому робот обслуговує основне технологічне обладнання, до них в основному належать роботи першого покоління, які виконують функції завантаження-розвантаження, і модуль роботизованого виробничого комплексу (РВК), в якому робот може виконувати основні чи допоміжні операції технологічного процесу, такі як зварювання, фарбування, напилення, складання та інші.

Ми можемо з впевненістю сказати що промислові роботи є одними з найважливіших компонентів у автоматизованих комплексах, таких як гнучкі виробничі системи (ГВС), тому що вони надають багата переваг у виробництві:

- а) поліпшення економічних показників;
- б) мінімізація робочого простору;
- в) підвищення продуктивності;
- г) підвищення рівня безпеки;
- д) підвищення якості обробки;
- е) мінімальне обслуговування по з рівнянню з людським ресурсом.

1.2 Класифікація промислових роботів

На даний час у світі прийнято розділяти промислових роботів на три основні етапи, по принципу їх взаємодії з навколишньою середою, здатності пристосовуватися до зміни задач або навіть до розуміння навколишнього середовища та взаємодії с ним, не виключаючи с цього навколишнього середовища и людину. Ці етапи оцінки «можливостей» промислових роботів

мають назву: першого, другого та третього покоління. Роботи першого покоління не відрізняються кмітливістю і лише виконують наперед задані команди, не здатні адаптуватися до умов виробництва і зовнішнього середовища та взагалі є дуже складними у зміні направленості виробництва. Деталі які вони завантажують або розвантажують мають бути розмішені тільки упорядковано і по заданій схемі. У той же час завдяки простоті конструкції, їх невеликій ціні та збільшеними можливостям(відносно людини, у довгостроковій перспективі) роботи першого покоління широко застосовуються в промисловості (завантаження - розвантаження, складування деталей, транспортування та інші).

Роботи же другого покоління оснащені досить розвиненим сенсорним апаратом та працюють за гнучкою програмою, що виконує роботу за принципом «ситуація - дія». Ця програма може самостійно вибирати оптимальній алгоритм для виконання промислових робіт в залежності від реального стану виробничого процесу. Такі роботи використовуються у тих галузях де необхідний повний контроль над процесом виробництва для того щоб мінімізувати можливі помилки у виготовляемому виробі.

Роботи третього покоління намагаються повністю копіювати людину, не тільки у способу виконанні якоїсь справи, а її спосіб мислення. Завдяки оснащенню новими способами адаптації, які включають у себе заохочення до виконання будь якої дії, яку задала людина роботи третього покоління набули здатності до самонавчання та розпізнавання образів, що є найдивовижнішим із елементів штучного інтелекту. З розвитком штучного інтелекту роботи та автоматичні роботизовані системи отримали можливість моделювання зовнішнього середовища, планування власних дій, прийняття рішень та аналізу виробничої обстановки з метою покращення виробничого процесу. В деяких розвинених країнах світу навіть ведуться роботи по переходу систем управління роботизованими комплексами на інтелектуальний інтерфейс, що дозволяє управляти роботами засобами природної мови.

Загальна класифікація промислових роботів може містити від 7 до 12 класифікаційних ознак (в залежності від інформаційного джерела [1]). Наведемо приклади приватних класифікацій по найбільш важливим класифікаційними ознаками. Поділ за характером виконуваних операцій показано на рис. 1.2.



Рисунок 1.2 - Класифікація промислових роботів за характером виконуваних операцій

Технологічні роботи(першого або другого покоління) виконують основні операції технологічного процесу: складання, зварювання, обробку, фарбування та інші. Допоміжні ПР можуть виконувати роботу обслуговування основного технологічного устаткування: установка-зняття заготовок, деталей, інструменту та оснащення, на транспортно-складських та інших операціях. Згідно с необхідністю універсальні роботи можуть виконують як основні, так і допоміжні операції, тобто поєднують в собі ознаки перших двох груп. За ступенем спеціалізації можуть розділятися на спеціальні, спеціалізовані та багатоцільові (рис. 1.3).

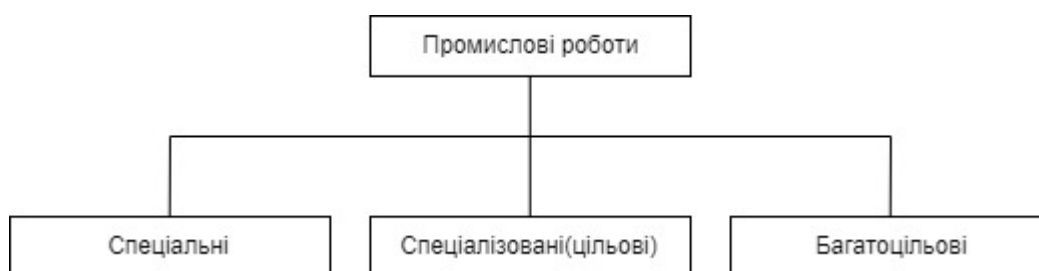


Рисунок 1.3 - Класифікація ПР за ступенем спеціалізації

Спеціалізований промисловий робот має можливість виконувати обидві із запропонованих раніше функцій в залежності від напрямку виробництва, або виходячи із того що на даний момент необхідно. Спеціалізовані ПР можуть виконувати технічні операції тільки одного виду (фарбування, штабелювання, зварювання, збірка, гнуття та інші). Вони можуть використовуватися для обслуговування широкої номенклатури моделей основного технічного обладнання, які знаходяться в одній, спільній маніпуляційній лінії. Якщо промисловий робот поєднує у собі можливість виконання декількох дій різних за своєю важливістю для підприємства, то він має назву багатоцільового робота. Такі роботи можуть виконувати основні або допоміжні операції. Якщо робот поєднує ці два фактори в один і той же час то він відноситься до числа універсальних. Також промислові роботи поділяються за системою координатних переміщень(рис. 1.4):



Рисунок 1.4 - Класифікація ПР по системі координатних переміщень

Прямокутна система координат підрозділяється на плоску і просторову. У виробництві роботів є дуже важливим вибір силового приводу(рис. 1.5), тому що саме від нього залежить точність, швидкість та ступінь свободи.



Рисунок 1.5 - Класифікація ПР за типом силового приводу

Як промислові роботи поділяються за характер відпрацювання програми показано на рис. 1.6.



Рисунок 1.6 - Класифікація ПР за характером відпрацювання програми

Більшість робіт які зараз використовуються у промисловості є першого покоління, тобто відносяться до числа жорстко програмованих. Програма в таких роботах задається заздалегідь і не змінюється протягом усього виробництва. Адаптивні же промислові роботи(другого покоління) містять у собі задану програму у якій зберігається про об'єкти навколишнього середовища та умові за якими між ними може здійснюватися взаємодія. Завдяки сенсорному оснащенню та гнучкій системі управління система може корегувати свою програму і тим самим зменшувати кількість помилок у виготовленні виробу. Гнучко програмовані промислові роботи здатні формувати свою програму на основі поставленої мети та інформації про навколишнє середовище. За характером програмування промислові роботи поділяються за швидкостями і дискретності переміщень(рис. 1.7).



Рисунок 1.7 - Класифікація ПР за характером програмування швидкостей і дискретності переміщень (ПУ - програмне управління)

Існує декілька видів переміщення робочого органу робота, такі як від точки до точки та по безперервній траєкторії. Такі види переміщення характерні в основному тільки для певних систем керування маніпуляціями робіт. Наприклад такий вид переміщення як від точки до точки характерний для позиційної системи керування і в основному застосовується у промислових роботах які виконують точне переміщення деталі за яким не обхідно слідкувати. А по безперервній траєкторії для контурної системи керування, яка застосовується для управління спільним рухом декількох робочих органів промислового робота. По типу представлення заданої інформації системи програмного керування (СПК) можуть поділятися на цифрові, аналогові, аналого-цифрові та циклові. До цифрових систем програмного керування відносяться ті у яких інформація зберігається на швидкозмінному носії і представлена у вигляді цифрових кодів. В аналогових системах програмного керування інформація задається у вигляді безперервно змінюваних значень фізичних (аналогових) величин. Аналого-цифрові системи програмного керування поєднують переваги попередньо перерахованих систем. Циклові системи програмного керування задаються настроюванням упорів, що впливають на кінцеве розміщення перемикачів. Також до класифікаційних ознак промислового робота відносять області застосування за видами виробництва (складання, зварювання та інші.), мобільність (стаціонарні, пересувні), число ступенів рухливості, конструктивне виконання (підлогові, підвісні або вбудовані в устаткування),

схема розташування приводів (в єдиному блоці, на виконавчих органах). За методом управління всі маніпулятори можна розділити на біотехнічні (з ручним керуванням), інтерактивні (зі змішанням управління) і автоматичні(рис. 1.8).



Рисунок 1.8 - Класифікація маніпуляторів за методом управління

У біотехнічному маніпуляторі управління може здійснюватися дистанційно або безпосередньо за рахунок переміщення робочого органу. Біотехнічні маніпулятори можуть розділятися на 3 підгрупи: копіюючі, командні, напівавтоматичні. У копіюючих маніпуляторів рух робочого органу повторює переміщення руки оператора. У командному маніпуляторі рух робочого органу не пов'язані кінематично з заданим пристроєм, а управління здійснюється окремо за кожної зі ступенів рухливості за допомогою кнопок, джойстиків та інших пристроїв. У напівавтоматичному маніпуляторі задана система включає в себе рукоятку, керуючу декількома ступенями рухливості, і спеціальний чип, який перетворює сигнали з пульта управління в сигнали, що управляють рухами роботи. Всі біотехнічні маніпулятори відрізняються від автоматичних відсутністю пристроїв пам'яті, і тому вимагають безперервної участі людини в процесі управління. Інтерактивні маніпулятори на відміну від біотехнічних мають пристрої пам'яті для збереження певних наперед запрограмованих команд руху. Залежно від форми участі людини інтерактивне управління може бути двох видів: автоматизоване, коли має місце чергування в часі автоматичних

режимів управління з біотехнічними. В автоматичному маніпуляторі виконання робочих функцій забезпечується без участі оператора. До числа автоматичних в ряді випадків можуть бути віднесені маніпулятори з інтерактивним управлінням.

1.3 Рука робота

Інша назва руки робота це механізована рука, тому що при її створенні люди намагалися відтворити такий механізм який зміг би виконувати таку ж саме роботу яку може і людина. На даний момент ми можемо зі впевненістю сказати що ми не тільки виконали план, але і навіть перевиконали. На даний момент роботи не просто аби як виконують задану їм роботу але і роблять це набагато якісніше за людину. Вже неможливо представити підприємства на яких у великих кількостях виконується такі роботи як фарбування, зварювання, складання та інші без роботизованих систем, особливо таких як руки-маніпулятори. Вони приводяться до руху за допомогою гідравлічними, електричними, а іноді і пневматичними приводами і мають певну запрограмовану заздалегідь послідовності рухів або напряду керуватися за допомогою пульта управління.

У наш час починають здобувати популярність роботи на голосовому управлінні. Звичайно це ще не ті роботи які повністю розуміють людину и можуть підтримувати розмову. Зараз промислові роботи лише можуть виконувати заздалегідь задані команди, які задаються оператором. Але це вже є великий крок уперед від звичайного управління роботами пультами. Найкраще себе показують в цьому напрямку роботи другого та третього покоління, які оснащені різноманітними датчиками і тому можуть реагувати на зміни навколишнього середовища в теперішній час, з мінімальною паузою на обробку отриманої інформації.

1.4 Класифікація роботів

Промислові роботи класифікуються не лише по принципу руки робота, є і інші класифікаційні принципи [2,3]:

Роботи зі змінною і жорсткою послідовність переміщень. Пристрої першого покоління які в основному виконуються роботу «взяти-покласти», за заданою командою, які мають механічні жорсткі упори завдяки яким і регулюється початкова і кінцева точки руку пристрою, але не проміжні. Такі пристрої не можуть виконувати одразу нові програми, а потребують попереднього перепрограмування. Якщо принцип роботи змінюється не тільки по принципу зміни положення початкової та кінцевої точки то робот потребує не тільки перепрограмування але і зміни механічних частин, таких як захватний пристрій та інші. На поточний час вже існують роботи із змінною послідовністю які можуть одразу змінювати послідовність операція і виконувати нову програму. Все це можливо завдяки тому що в них включено декілька жорстких упорів, які відповідають кожен за свою операцію. Більшість роботів на даний час виготовляється по даній схемі.

Роботи із стежачою системою і без неї. Будь який робот який змінює положення свого робочого органу має бути обладнаний можливістю зупинити окремий вузол у будь-якій точці своєї траєкторії. Для того щоб це зробити треба або вимкнути окремий вузол, або подати йому команду на зупинку. Крокові двигуни мають змогу зупинитися при завершення свого кроку, але не всі промислові роботи оснащені ними. Ще одним із недостатків цього способу є те що ми не можемо із упевненістю сказати де саме закінчиться крок і чи зможе ця аварійна зупинка запобігти аварійній ситуації. Тому на всіх промислових роботах вирішено було встановлювати на кожний вузол сервомеханізма, який буде ефективно керувати кожний вузол, для того щоб програма була виконана із заданою точністю і не створювала аварійних ситуацій. Зазвичай таких роботів також називають сервороботами, або роботами із стежачою системою.

Роботи з контурними або позиційними системами руху.

Контурними є системи в яких робочий орган рухається від початкової точки до кінцевої точки за не постійною траєкторією. Траєкторією руху не те що не переймалися, на той час її було важко передбачити. Також не останньою причиною в цьому було і те що на той час запам'ятовувальні пристрої мали малі об'єми та велику ціну. Коли запам'ятовувальні пристрої почали отримувати збільшені об'єми пам'яті з'явилася можливість збільшити число запам'ятовуються точок. Завдяки цій події виробники ввели новий термін, як багато-точкове управління, яким описується зберігати у пам'яті промислового робота дуже великого числа дискретних точок, які у деталях описують як саме повинен рухатися робочий орган.

Для робіт які виконують прості дії, як наприклад перекладання продукції з одного конвеєра на інший або що складають їх для майбутнього пересування точний контроль по пересуванню робочого органа та всіх його вузлів є не доцільним. Тому таці системи використовуються лише в зварювальних або фарбувальних роботах. Безпосередньо для дугового зварювання або фарбування розпиленням. Такі роботи розділяють увесь рух на багату кількість точок, кількість яких залежить від його запам'ятовувальних можливостей, та безпосередньої необхідності у точності руху робочого органу. Ці точки можна обчислювати при фактичному русі шляхом інтерполяції, тобто створення між двома точками прямої лінії та поділу цієї лінії на точки.

Роботи першого, другого, третього поколінь. До робіт першого покоління відносяться роботи у яких повністю відсутні слідкуюча система або якісь датчики. Мають мінімальну кількість пам'яті та жорстку систему переміщень. Із-за мінімальної комплектації мають невелику ціну і тому широко використовуються на підприємствах.

Роботи другого покоління відрізняються від першого тип що вони мають необхідну кількість датчиків для того щоб зібрати всю необхідно інформацію із навколишнього середовища та свого виробу для того щоб

найбільш якісно виконати задану їм роботу. Вони не тільки здатні отримувати інформацію із навколишнього середовища на початку роботи, такі як: який прийшов об'єкт, якою стороною, що і де необхідно зробити але і можуть корегувати свої дії під час самого виконання. Можуть мати не тільки технічні пристрої які імітують зір, вимірюють відстань, перевіряють температуру, внутрішній склад отриманої деталі і навіть наслідувати такі складні до імітації людські якості як відчуття дотику.

Під час створення роботів третього покоління людина намагається повністю імітувати себе, тобто створювати роботів гуманоїдного типу які не тільки мають схожу з нами будову а і є «інтелектуальними», тобто можуть самі вивчати навколишнє середовище та навчатися на своїх помилках. На даний час роботи вже навчили пересуватися та долати перешкоди без сторонньої допомоги завдяки самонавчанню, використовуючи технологію нейронних мереж яку на даний час підтримує та розвиває така відома компанія як «Нвидиа».

Незважаючи на всю різноманітність класифікаційних ознак, існують так звані «сірі зони». Наприклад, додавання одного простого датчика ще не робить пристрій роботом другого покоління. Необхідно, щоб датчик повністю змінював та покращував дії робота. Цілком можливо, що тільки роботів другого покоління можна вважати справжніми роботами, відносячи перше покоління до програмованим пристроїв, або звичайним маніпуляторам.

Також роботів маніпуляторів класифікують за такими ознаками[2]:

1) За сферами застосування:

- промислові;
- медичні;
- побутового обслуговування;
- спеціальні;
- військові;
- космічні;

- підводні та інші.
- 2) За функціональним призначенням:
 - мобільні (стаціонарні і рухомі);
 - інформаційні (вимірювально-інформаційні);
 - маніпуляційні (автоматично діючі, дистанційного керування).
 - 3) За точністю руху вихідної ланки:
 - мала - похибка позиціонування від 1 мм і вище;
 - середня - похибка позиціонування від 0,1 мм до 1 мм;
 - висока - похибка позиціонування від 0,01 мм до 0,1 мм;
 - надвисока - похибка позиціонування менше 0,01 мм.
 - 4) За місцем розміщення джерела електроживлення:
 - автономні;
 - неавтономні.
 - 5) За місцем розміщення центру управління рухом:
 - самонавідні;
 - із зовнішнім наведенням.
 - 6) За типом каналу зв'язку з центром зовнішнього управління:
 - провідний;
 - радіоканал;
 - інфрачервоний;
 - ультразвуковий.
 - 7) За типом системи управління:
 - управління оператором (дистанційно і безпосередньо);
 - система з програмним управлінням;
 - система з адаптивним керуванням;
 - супервізорне управління;
 - комбіновані системи дистанційно-автоматичного керування.
 - 8) За кількістю рухливостей:
 - з однією, двома, трьома і чотирма рухливостями;
 - з рухливостями більше чотирьох.

- 9) За видами інтеграції мехатронних модулів, що входять до складу мехатронної машини:
- модулі руху;
 - мехатронні модулі руху;
 - інтелектуальні мехатронні модулі.
- 10) За типом мехатронних модулів:
- електромеханічні;
 - пневматичні;
 - гідравлічні;
 - комбіновані.
- 11) За місцем розміщення мехатронних модулів:
- в єдиному блоці;
 - на рухомих ланках;
 - комбінована компоновка.

Мехатронна система - мехатронні машини або кілька однотипних мехатронних пристроїв, об'єднаних енергетичними, технологічними та інформаційними зв'язками, які здійснюють необхідні функціональні перетворення.

Залежно від рівня інтеграції мехатронних пристроїв мехатронних система може бути частиною мехатронної машини (наприклад, приводи ступенів рухливості одного робота), або самої мехатронної машиною (наприклад, робот), або представляти собою кілька однотипних машин, об'єднаних відповідними зв'язками, що виконують один технологічний процес (наприклад, група верстатів, які здійснюють механічну обробку деталі).

Мехатронний комплекс - одна і чи кілька мехатронних машин з інструментом та інше технологічне обладнання, яке виконує єдиний технологічний процес.

Таким чином, мехатонні пристрої приходять на зміну механічних пристроям, які вже не відповідають якісним вимогам, що пред'являються до сучасного обладнання.

1.5 Структура і геометрія маніпуляторів

Структура побудови маніпуляторів є досить різноманітною і розрізняється розташуванням видами і розташуванням кінематичних пар з різною рухливістю та числом ланок. Найбільш поширені види побудови маніпуляторів зображені на рис. 1.9, а – г.

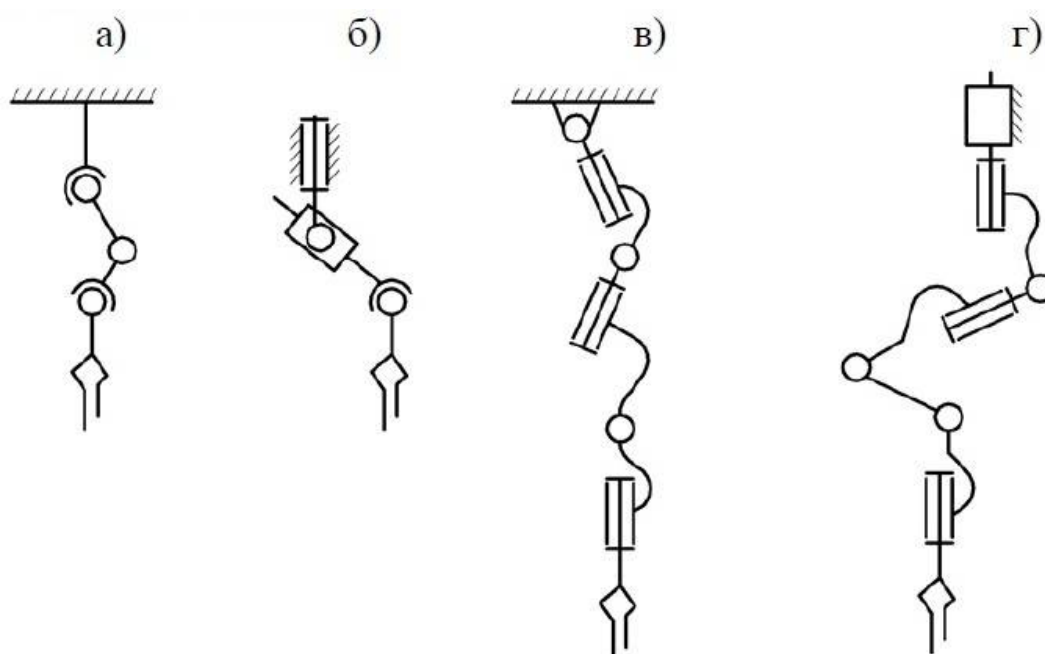
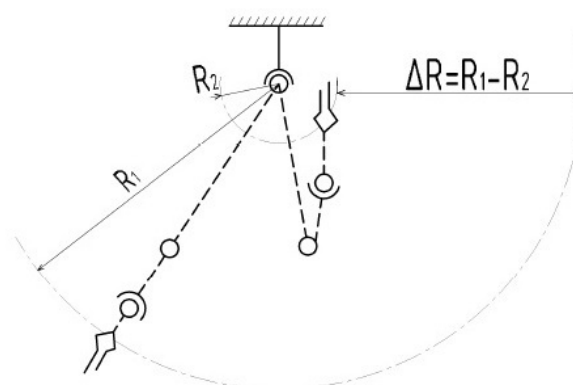


Рисунок 1.9 - Схема маніпуляторів

Маніпулятор може складатися з поступальних та обертальних пар [4], а також сферичних пар. Сферичні пари є більш використовуваними у виробництві роботів наукового типу бо вони є більш рухомими і має більшу ступінь свободи. Іноді сферичну пару можуть замінити трьома обертовими парами. Також кожен маніпулятор характеризується зоною обслуговування, тобто максимальною відстанню на якій маніпулятор може виконувати задану йому програму. Наприклад, робоча зона маніпулятора (рис. 1.9, а)

обмежується сферою з радіусом, рівним максимальній довжині кінематичного ланцюга, що показано на рис. 1.10. Маніпулятор за схемою рис. 1.9, а має $m = 1$; за схемою на рис. 1.9, б, в - $m = 0$. Завдяки підвищеній маневреності робот може виконувати більш складні рухи що збільшує свободу дій оператора при управлінні. Фактично, маневреність що дорівнює одиниці означає, що до заданої точки робочий орган може підійти з любого положення.



Зона обслуговування маніпулятора

Рисунок 1.10 - Зона обслуговування маніпулятора

Зона в якій маніпулятор може виконувати попередньо задану програму називають робочою зоною. Так для маніпулятора зображеного на рис. 1.9, а робоча зона - є простором між сферами радіуса R_1 і R_2 (рис. 1.10). Усі точки робочої зони характеризується кутом сервісу ψ , який можна пояснити як деякий кут всередині якого робочий орган має доступ до будь якої точки.

Відношення $\frac{\Psi}{4\pi} = \Theta$ називається коефіцієнтом сервісу, значення якого

може змінюватися $0 \leq \Theta \leq 1$ (рис. 1.11). За графіками $\theta(R)$ можна оцінити можливості тієї чи іншої кінематичної схеми маніпулятора.

Також до основних технічних показників маніпуляторів та промислових роботів відносять: швидкість, вантажопідйомність, точність

позиціонування, та енергетичні витрати. Більш точно характеристики роботів та маніпуляторів було розписано у розділі 1.4.

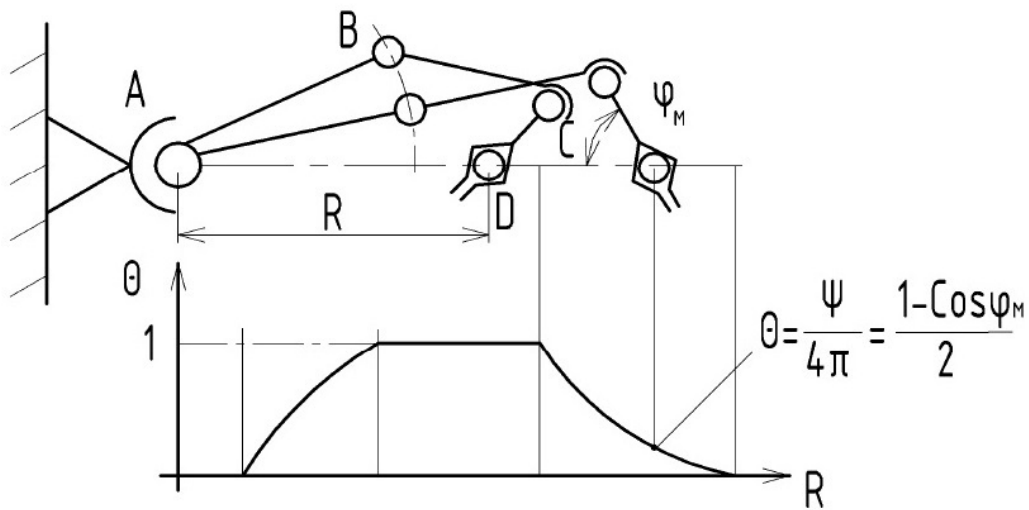


Рисунок 1.11 – Графік залежності θ (R)

1.6 Механіка промислових роботів

Рухові можливості промислового робота в основному визначається видом і розташуванням кінематичних пар. Загалом існує три системи руху, такі як: локальні, регіональні та глобальні. Локальна система руху забезпечує точність позиціонування. Регіональний рух відноситься до внутрішньо операційного, тобто переміщення між операціями. Глобальне переміщення же забезпечує мобільність робота, тобто його переміщення з одного місця на інше. Промислові роботи мають декілька видів робочих зон, наприклад таких як сферична, циліндрична та прямокутна (рис. 1.12).

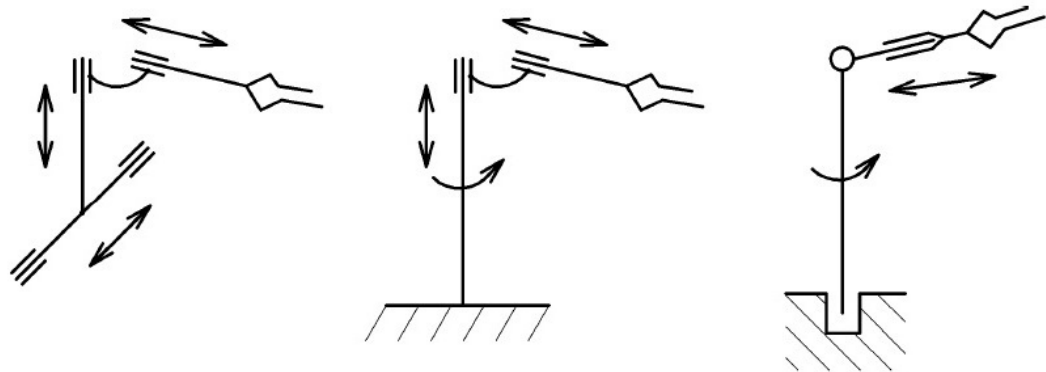


Рисунок 1.12 – Схеми промислових роботів

Основною вимогою до побудови промислового робота є забезпечити цьому роботу виконувати задану роботу з будь якого положення, тобто щоб робочий орган міг дістати до будь якої точки у заданій робочій зоні та міг виконувати завдання в необхідною точністю. Для цього перед тим як робити робота спочатку вирішують для чого він потрібен, тобто яку працю він буде виконувати. Це потрібно для того щоб вибрати потрібну кількість ступенів рухливості. Важливою перевагою великої кількості ступенів рухливості є те що промисловий робот може дістати до будь якої точки у робочій зоні, з будь-якого місця та під будь-яким кутом. Але чим більше ступенів рухливості тим меншу вагу може підіймати робот, тому треба зберігати певний баланс, або робити вузько спеціальний робота який може виконувати тільки один тип завдань.

Промислові роботи в основному складаються з основних таких конструктивних елементів(модулів):основа; колона, закріплена на підставі; рука (руки), що кріпиться до колони; кисть, що кріпиться до руки; технологічні механізми, що кріпляться до кисті; захватні пристрої або інші робочі органи, що кріпляться до кисті. Для рухливих промислових роботів додається візок підлогового або підвісного Виконання.

Для того щоб промисловий робот буд універсальним в його конструкції можуть використовувати уніфіковані елементи, такі як наприклад рука, кисть, технологічний механізм, захватний пристрій. Тобто елементи можна відключати або знімати за необхідністю.

1.7 Виконавчі механізми і приводи промислових роботів

Виконавчі механізми розділяються на 3 класи: гідравлічні, пневматичні, електричні.

Гідравлічний привід це такай механізм який перекачує через насосрідину до гідродвигуна, тиск створений у гідродвигуні і двигач гідроциліндр, тобто перетворює гідравлічну енергію у механічну. Гідравлічні приводи мають непогані показники майже у всіх напрямках, тобто забезпечують високі швидкості, високу вантажопідйомність та зручність управління.

Однак саму високу швидкість забезпечує тільки пневматичний привід, який зараз застосовується майже у 50% випущених промислових роботах.

Останнім часом дуже великої популярності здобули електричні приводи. Електричні приводи в основному у промислових роботах для яких є не дуже важливою велика вантажопідйомність та не потрібне часте переналагодження.

Для виконавчих механізмів циклової дії використовуються механічні приводні пристрої зворотно-поступального або ротаційної дії з программоносіями у вигляді профільованих кулачків або барабана, де обертальний рух перетворюється в зворотно-поступальний вихідних ланок маніпуляторів (рис. 1.12, а).

Широкі можливості відкриваються з використанням високочастотних вібродвигунів, що мають хороші динамічні параметри вперехідних режимах (пуск, зупинка, кроковий режим), тому що віброелементи з моменту відключення живлення стають гальмуючими. Вони забезпечують високу точність маніпулювання. Принцип дії заснований на перетворенні косих ударних коливань високої частоти (більше 20 тис. Гц) в обертальний рух ротора за рахунок сил сухого тертя (рис. 1.12, б).

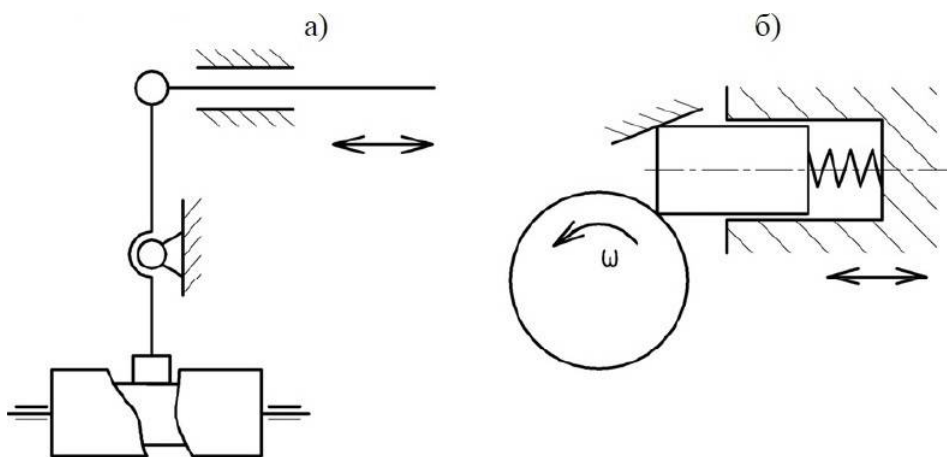


Рисунок 1.12 - Привідні пристрої:

а – кулачкові, б - вібраційні

1.8 Побудова рухових систем роботів

При побудові промислових роботів дуже важливим є місце розташування двигунів які надалі будуть приводити в рух ланок, бо від варіанту їх компоновки в майбутньому буде залежати область застосування швидкість, керованість маніпулятора.

Існують три типи розташовування приводів: безпосередньо в шарнірах механізму, на підставі і комбіноване. Також існують такі види конструкцій, в яких всі рухи від двигуна через перемикаючи муфти передаються одразу на всі ланки.

Найчастіше у виготовленні промислових роботів використовують комбінований тип розташування двигунів. Це зумовлено тим, що розташування двигунів прямо на ланках зменшує кінематичні помилки передачі руху, але виникають і недоліки, збільшуються габарити і маса. Якщо же компонувати приводи на основі то ми зможемо зменшити вагу рухомих ланок, але тим самим ми ускладнюємо будову маніпулятора, що приводить до збільшення числа ланок та появи помилок у передачі руху. Також у багатоланкових кінематичних ланцюгах можуть виникають люфти, з якими також необхідно боротися.

На рис. 1.13 а, б, зображені приклади розміщених на основі. Передача руху на них може здійснюється зубчасто-важільним або роликів-стрічковим диференціальним механізмом. Тобто ми маємо можливість керувати швидкістю або вантажопідйомністю, створюючи різницю передавального механізму від приймаючого. В більшості серійно випущених роботів передача руху від приводів до ланок здійснюється за допомогою трубчастих валів, на кінцях яких закріплені конічні колеса.

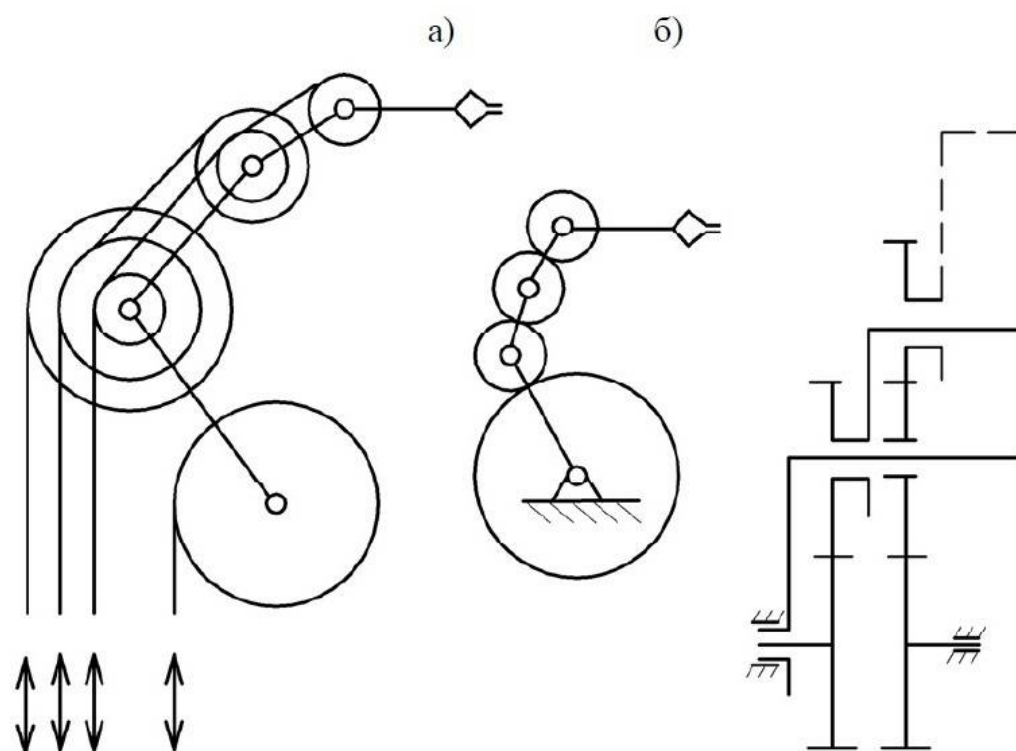


Рисунок 1.13 - Приводи роботів
а – роликів, б – диференційні

Такі ж само колеса ми можемо зустріти і у триланкових диференціальних механізмах (TDM). На рис. 1.14 зображено шарнір у вигляді TDM, побудова якого дозволяє згин щодо осі Y і обертання по осі Z.

Існують і інші конструкції які схожу за призначенням і забезпечують транзитну передачу руху. Для реалізації цього механізму у середину основних ланок встановлюються автономні TDM.

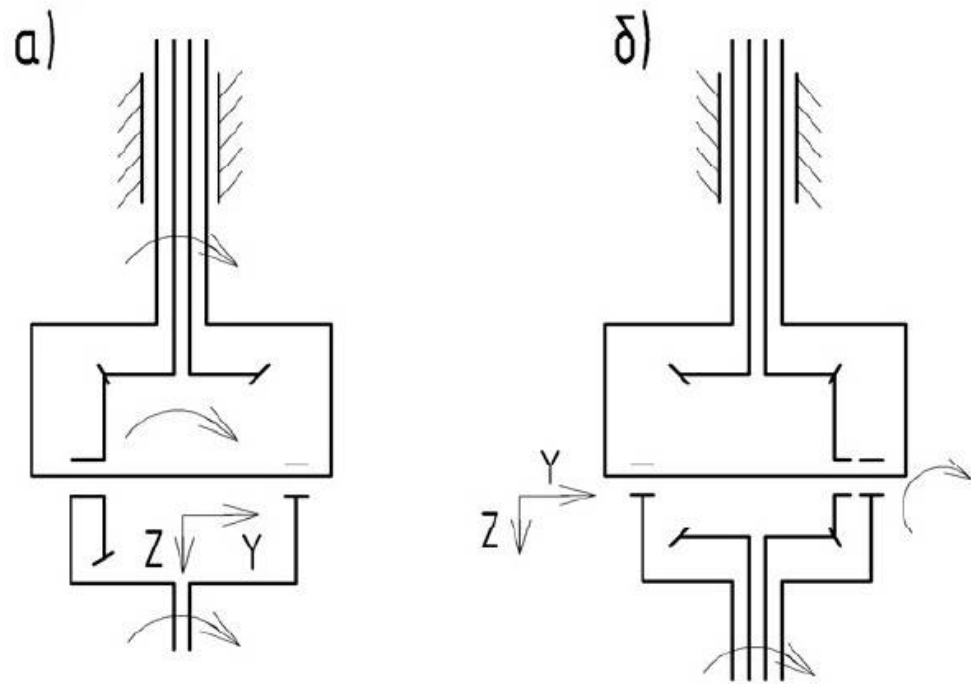


Рисунок 1.14 – Схеми TDM

а – шарнірні, б - шарнірно-автономні

Ще одним недоліком механізмів з великою кількістю ланок є те що при передачі руху з основних данок маніпулятора можуть виникати зазори які будуть причиною виникнення помилок руху, що є неприйнятно для промислових роботів. Для того щоб позбавитися цього недоліку було розроблено спеціальний механізм міцно замикаючий ланцюги. Він містить у своїй конструкції пружний елемент (торсіон), який попередньо закручується у з'єднувальні отвори (рис. 1.15).

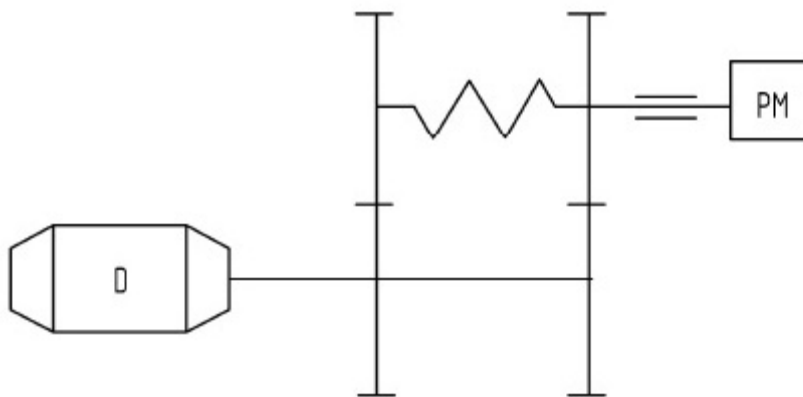


Рисунок 1.15 - Кінематична ланцюг з пружним елементом

1.9 Захватні пристрої та передавальні механізми

Одна із основних задач хватних пристроїв це утримувати у своєму робочому органі об'єктів які можуть відрізнятися за розміром, вагою та конфігурацією. Для виконання цієї функції існує три основних типи хватів: механічні, магнітні, вакуумні. Також можуть зустрічатися інші не розповсюджені типи хватів, але вони використовуються у промислових виробництвах дуже не часто. Такі хвати діють за принципом прилипання, проколювання, притягання.

Механічні хват в основному використовують для виробів які мають спеціальні, наперед розраховані розмір, вагу, властивості. Тобто хват розробляється спеціально для заданої деталі, і не завжди може використовуватися у транспортуванні інших об'єктів, інакше кажучи не є уніфікованим. Для зменшення інерційних навантажень масу та габарити хватів прагнуть зменшити.

Ті підприємства яким необхідні універсальні хват можуть використовувати хват с регульованими губками. Такі хват мають легко переналагоджувальну конструкцію.

Також дуже важливою якістю є само настройка захвату для компенсації неточності положення об'єкта. Існує декілька видів механічних захватів, такі як односторонні, двосторонні та об'ємні. За конструкцією поділяються зубчасті, ланцюгові, кулачкові (клинові), шарнірні, гвинтові, та комбіновані.

Для захватів частіше використовують пневмо- або гідропривід.

У прикладах захватів (рис. 1.16), захоплення «а» забезпечує найвищу силу, надійність, швидкість спрацьовування, компактність. В основному застосовується в токарних маніпуляторах.

Захват за схемою «б» використовується для захоплення деталей з паралельними гранями. Схема захвата з паралельними губками використовується в основному з невеликими об'єктами «в».

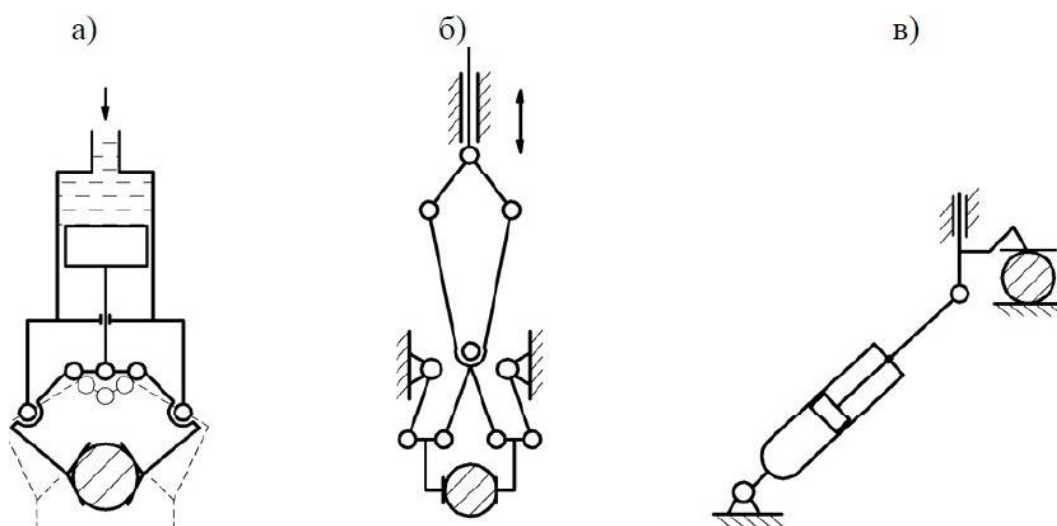


Рисунок 1.16 – Конструкції захватів

а – з гідроприводом, б – з електроприводом, в – з пневмоприводом

Для утримання деталі «пальці» роботів роблять не повністю твердими, а такими щоб вони при дотику до об'єкта деформувались, облягаючи деталь і утримуючи її.

З метою скорочення часу на виробництві встановлюється два роботи, один з яких відповідає за забирання виробленої деталі, а інший за щоб класти сировину. Таким чином можна в 2-3 рази знизити час обслуговування.

Найрозповсюдженішою схемою захватів є схеми з двома жорсткими пальцями, які здійснюють обертальний або поступальний рух (рис. 1.17, а, б).

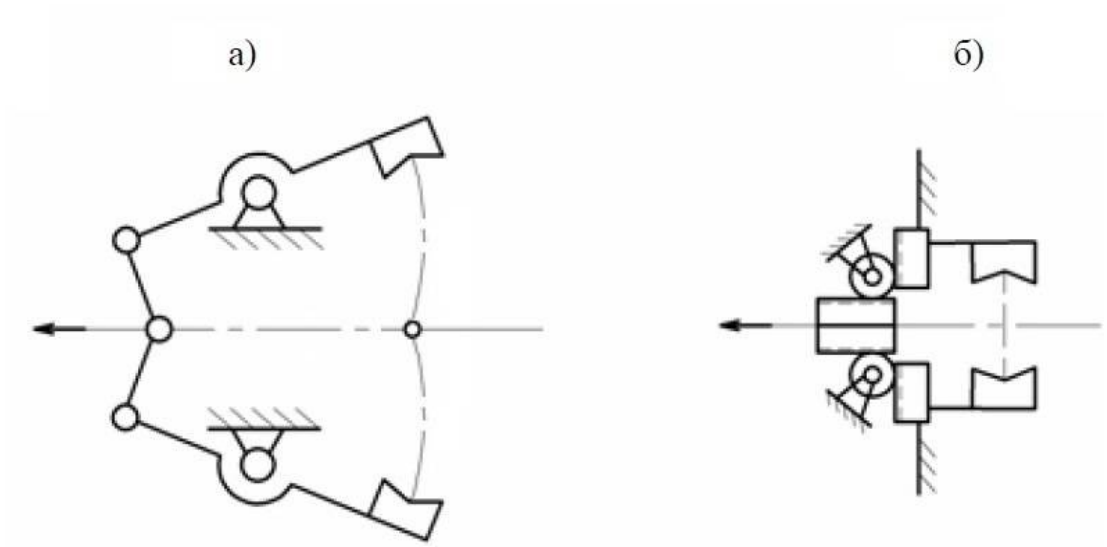


Рисунок 1.17 – Кінематичні системи захватів
 а - з обертаним рухом, б - з поступальним рухом

Схема на рис. 1.17, а - більш проста і надійна, але не може взаємодіяти з великими об'єктами. Схема б - більш складна але дозволяє працювати в широкому діапазоні розмірів.

Кінематичний аналіз механізмів захвату може проводитися за відомими швидкостями і прискорень вхідних ланок. Також можна використовувати метод планів, який показано на рис. 1.19, а - в.

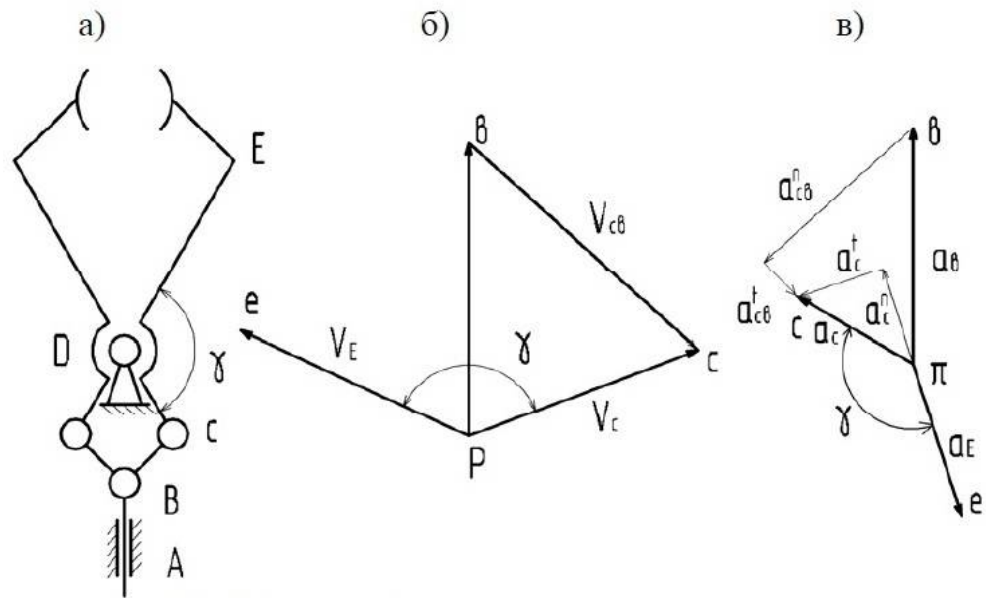


Рисунок 1.19 – Кінетичний аналіз захвату маніпулятора
 а – схема, б – план швидкостей, в – план прискорень

2 ОПИС ПРОМИСЛОВОГО РОБОТА IRB 460

Робот - це електромеханічний блок, який може виконувати певні операції автономно (згідно заданою програмою або під контролем людини). Промислові роботи вже багато років присутні в цій галузі, де вони використовуються для виконання небезпечних, важких або важких робіт для людей.

Роботи серії IRB 460 є роботами багатонаціональної корпорації AVV, призначеної для промислових робіт. Robot IRB 460 [5] заснований на простій і надійній конструкції, яка дозволяє здійснювати складне управління рухом з високою продуктивністю. Робот AVV IRB 460 (рис.2.1 і 2.2) має чотири ступені свободи, тобто чотири обертових з'єднання, які живляться від електричних сервомоторів. Робот призначений для високопродуктивної упаковки і обробки готової продукції, він є одним з найшвидших роботів з групи роботів для обробки піддонів. Він відрізняється міцними сполуками, які забезпечують надійну обробку вантажів . Таким чином, ця конструкція дозволяє скоротити час циклу, зменшити вагу руки робота, а також збільшити вантажопідйомність і незначні додаткові переваги.

Цей робот з чотирма свободами пересування (чотири з'єднання з діапазоном 3,180 м і навантаженням 460 кг, деякі ключові дані показані в таблиці 2.1.

Таблиця 2.1 Основні дані робота AVV IRB 460 Вантажопідйомність

Вантажопідйомність	460 кг
Діапазон	3 180 мм
Висота робота	2 504 мм
вага робота	2 310 кг
захист	IP67



Рисунок 2.1 - Робот AVV IRB 460

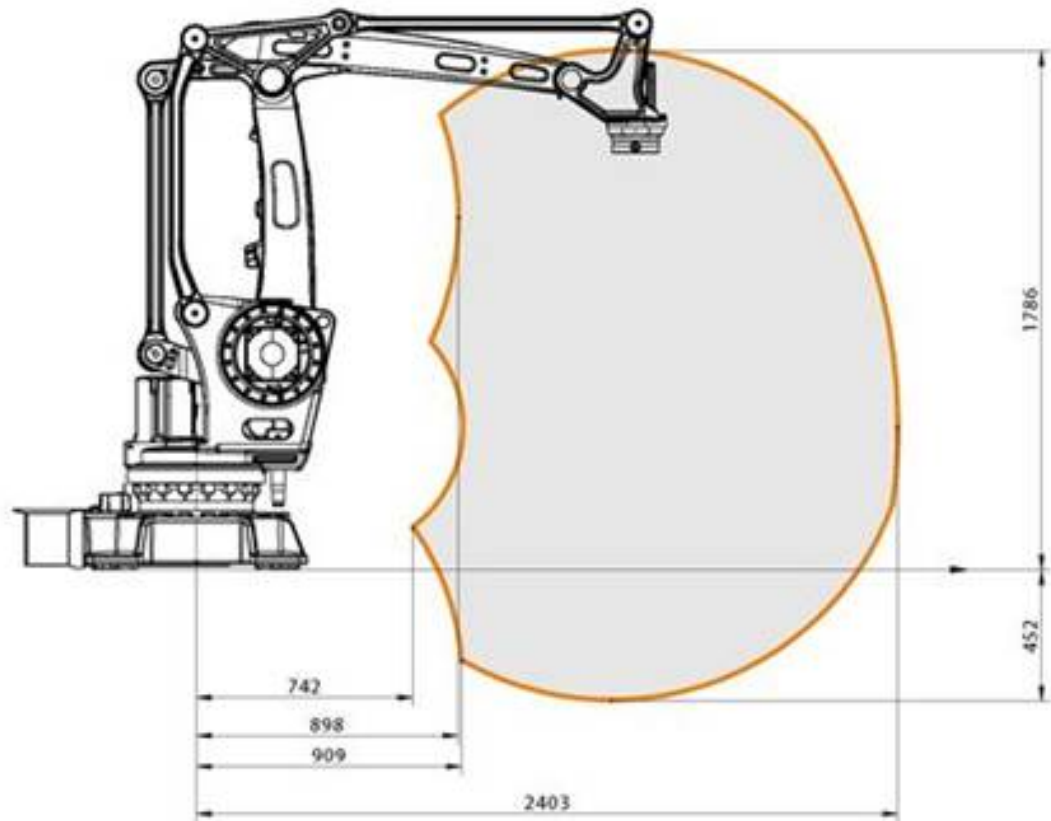


Рисунок 2.2 – Діапазон робочої зони роботів ABB IRB 460

Першим кроком в моделюванні роботів ABB IRB 460 є визначення параметрів ДН. Таблиця параметрів ДН призначена для призначення ортогональних координат парі суміжних з'єднань у відкритій кінематичній системі. Він використовується в робототехніці, де один робот може бути змодельований як багато зв'язкові тверді тіла (сегменти), де параметри ДН використовуються для визначення зв'язку між двома сусідніми сегментами. Першим кроком у визначенні параметрів ДН є визначення суглобів і визначення типу руху (обертання або трансляції) для кожного суглоба. Як уже згадувалося, і як видно з малюнка вище, робот ABB IRB 460 має чотири обертових з'єднання O_1 , O_2 , O_3 і O_4 , кожен з яких має дозволений кут повороту Θ_1 , Θ_2 , Θ_3 і Θ_4 . Суглоби, осі і кути повороту показані на наведеній схемі на малюнку 2.3 .

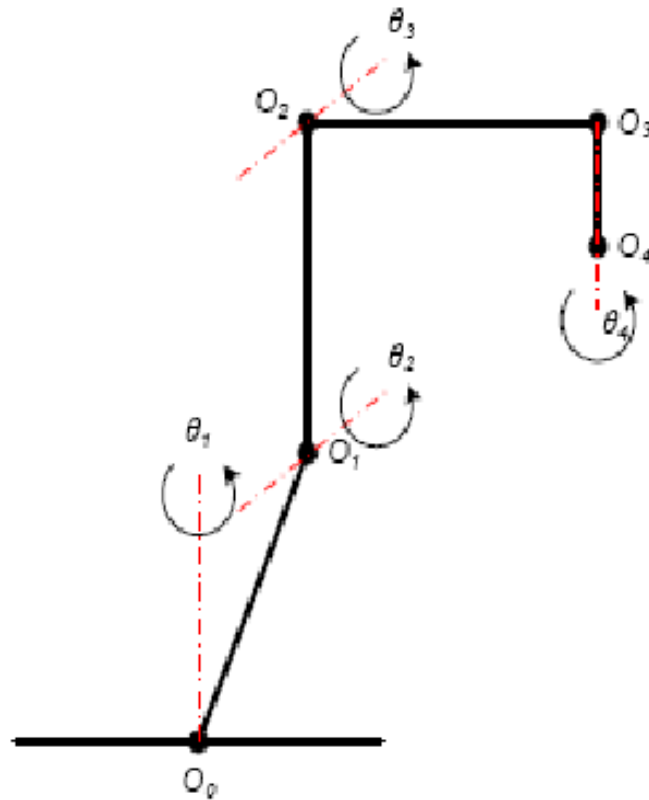


Рисунок 2.3 - Вали і кути обертання роботів

Наступним кроком буде визначення параметрів a_i і d_i на основі фізичних вимірів робота (рис.2.4). Параметр a_i представляє довжину з'єднання, тобто нормальна відстань між осями вісь + 1. Параметр d_i є відстань між сегментами уздовж осі z_{i-1} (рис.2.5).

Таблиця 2.2 – Параметри ДН

Кількість осей	Кут повороту	Довжина зв'язку a_i (m)	Відстань між сегментами d_i (m)	Кут повороту θ_i	Зсув кута повороту θ_i
1	$\pi / 2$	0,3	0,814	змінна θ_1	0
2	0	1,26	0	змінна θ_2	$+\pi / 2$
3	$+\pi / 2$	1,64	0	змінна θ_3	$-\pi / 2$
4	0	0	0,27	змінна θ_4	0

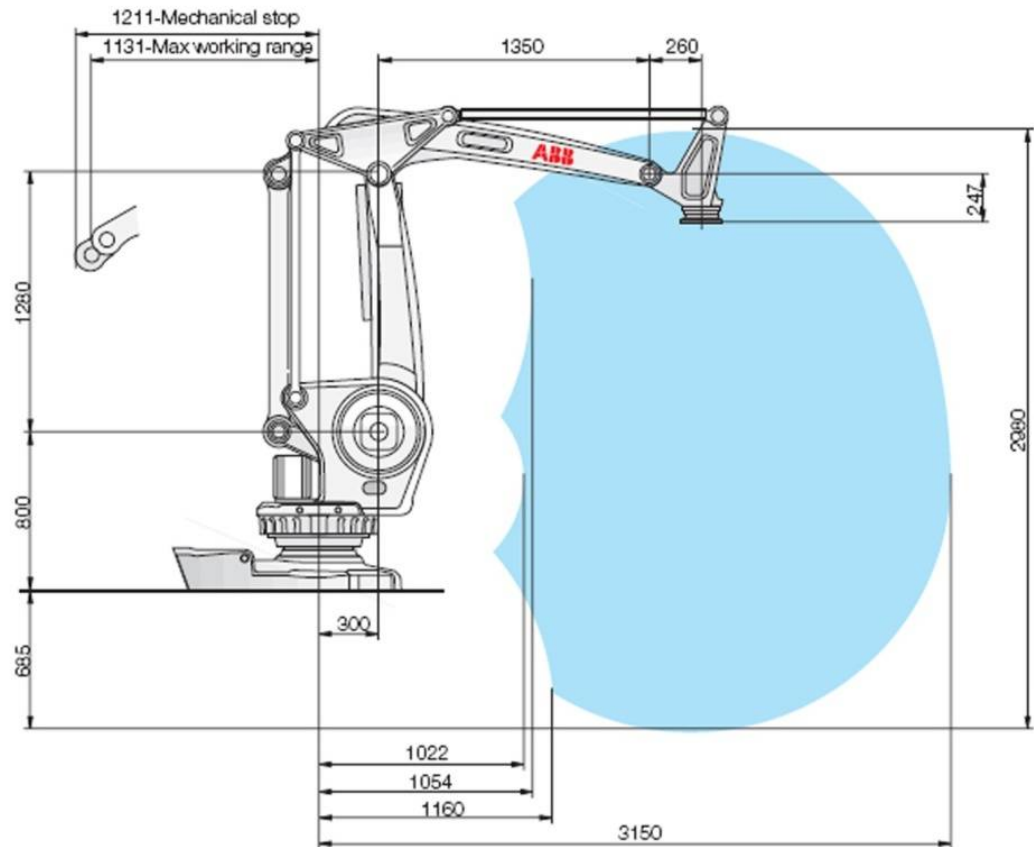
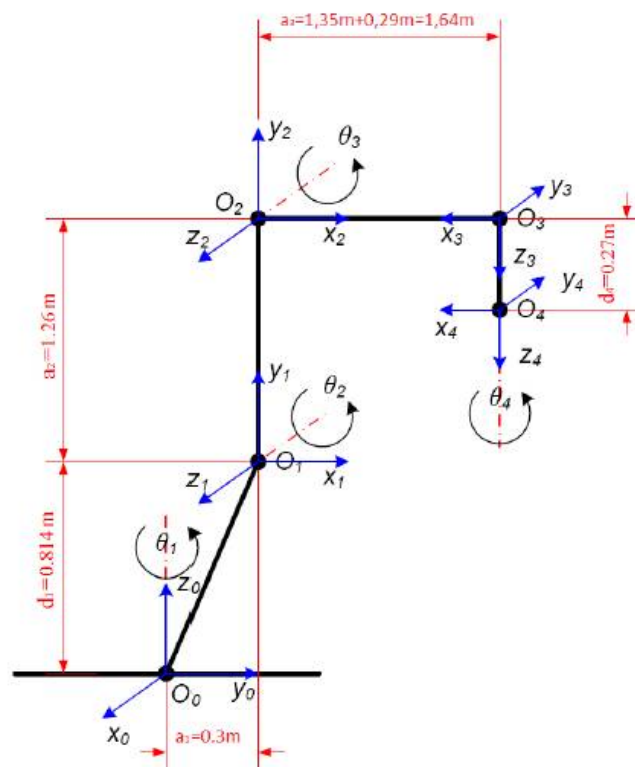


Рисунок 2.4 - Розміри робота IRB 460

Рисунок 2.5 - Параметри a_i і d_i

З РОГЛЯД МОЖЛИВОСТІ СТВОРЕННЯ ФІЗИЧНОЇ МОДЕЛІ РОБОТА ЗА ДОПОМОГОЮ ОПЕНСОРС ПРОЕКТУ MEARM

MeArm - кишенькова версія промислового маніпулятора [6]. Це опенсорсний проект - всі дані про дизайн і файли програми доступні для вільного скачування під ліцензією CreativeCommons, який є сприятливим середовищем для розробки і тестування програмного забезпечення, яке буде в подальшому використовуватися на промислових роботах, таких як IRB 460 .

3.1 MeArmRobotArm

В якості основної системи використовувався мініатюрний варіант промислового робота-маніпулятора MeArm (pic.3.1) .

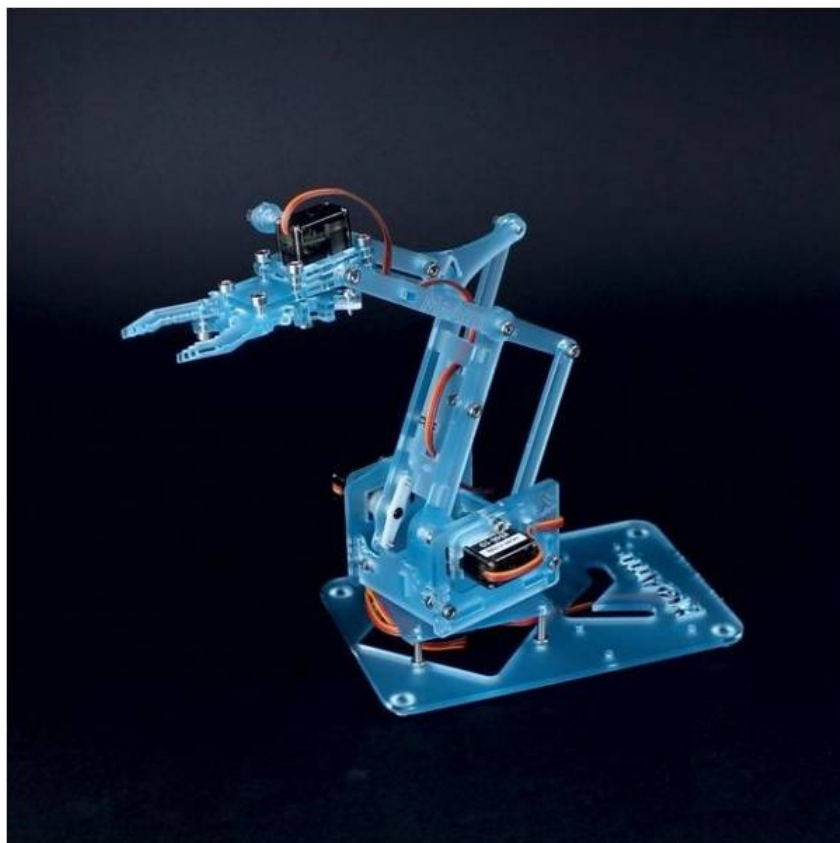


Рисунок 3.1 - маніпулятор MeArm в зібраному вигляді.

Каркас даного маніпулятора ми вирізали з листа акрилової паперу формату А4 (рис. 3.2) .

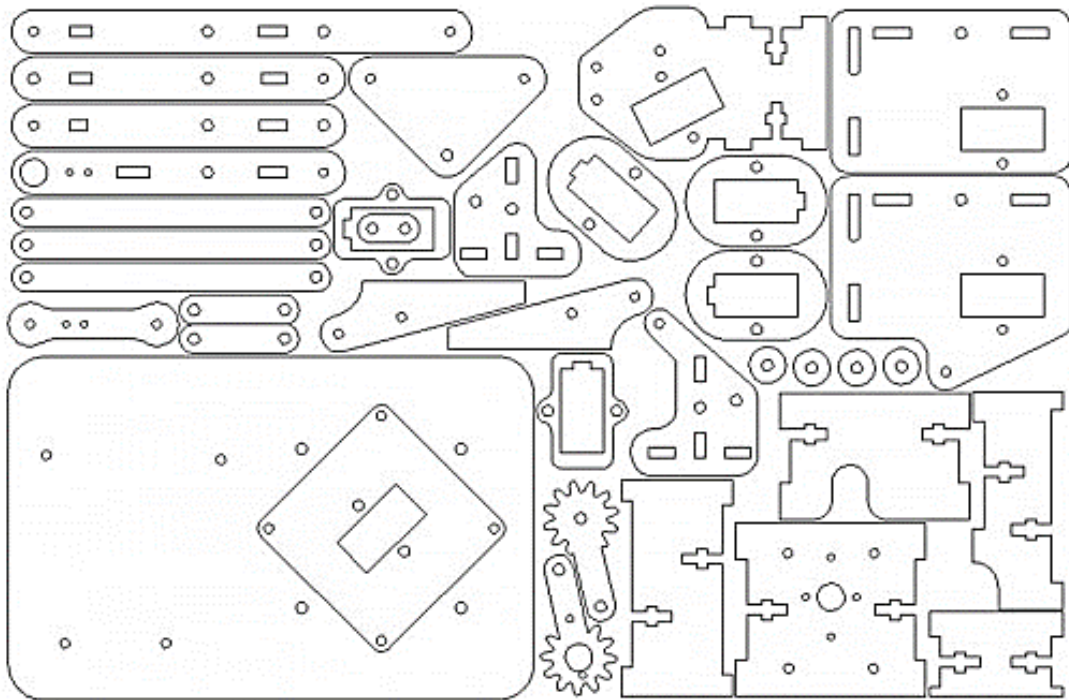


Рисунок 3.2 - Каркас маніпулятора MeArm.

Так само були використані сервоприводи SG-90, які видно на рис. 3.1.

Для управління цим пристроєм було вирішено використовувати джойстики та голосове керування через Android телефон за допомогою Bluetooth модуля.

3.2 Сервоприводи SG-90

Для побудови рухомої частини маніпулятора MeArm було вирішено використовувати сервоприводи SG-90, які повністю підходять нам по типорозмірам, показнику вантажопідйомність та не є дуже дорогими.

Набір сервоприводів SG-90 складається з:

- Сервопривод SG-90 9g - 1 шт

– Пластикові плечі для сервоприводу - 3 шт

– Гвинт для кріплення - 3 шт

Специфікація SG-90 зображена в таб. 3.1:

Таблиця 3.1 - Специфікація

Крутний момент	1,80 кг/см при 4,8 В
Швидкість	0.1сек / 60 ° (4.8В)
Напруга	4.0 В до 7.2 В, 4.6 В - 5.2 В номінальна
Запуск струму з напругою 5 В (без механічної навантаження)	220 ± 50мА
Струмівий струм з напругою 5 В (замком заглушений)	650 ± 80 мА
Потужність холостого ходу з напругою 5 В	6 ± 10 мА
Розміри	23mm x 12.2mm x 29mm
Ширина мертвої смуги	10 мкс
Вага	9g
Діапазон робочих температур	від -30 ° С до +60 ° С

Розміри сервоприводу SG-90 зображені на рис.3.3

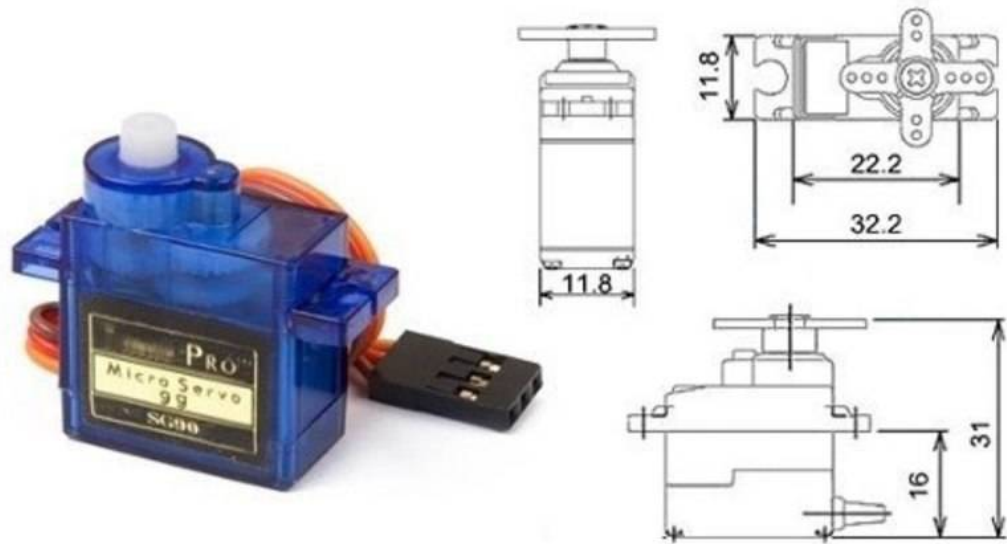


Рисунок 3.3 - Розміри сервоприводу SG-90

3.3 Складання

Перед початком збирання була розроблена попередня схема підключення сервоприводів і джойстиків до Arduino (рис. 3.4) а також підключення Bluetooth модуля (рис 3.5) для подальшого його використання для подачі голосових команд з мобільного телефону.

Після отримання основних схем побудови необхідно відкалібрувати мотори і тільки після цього починати збірку маніпулятора.

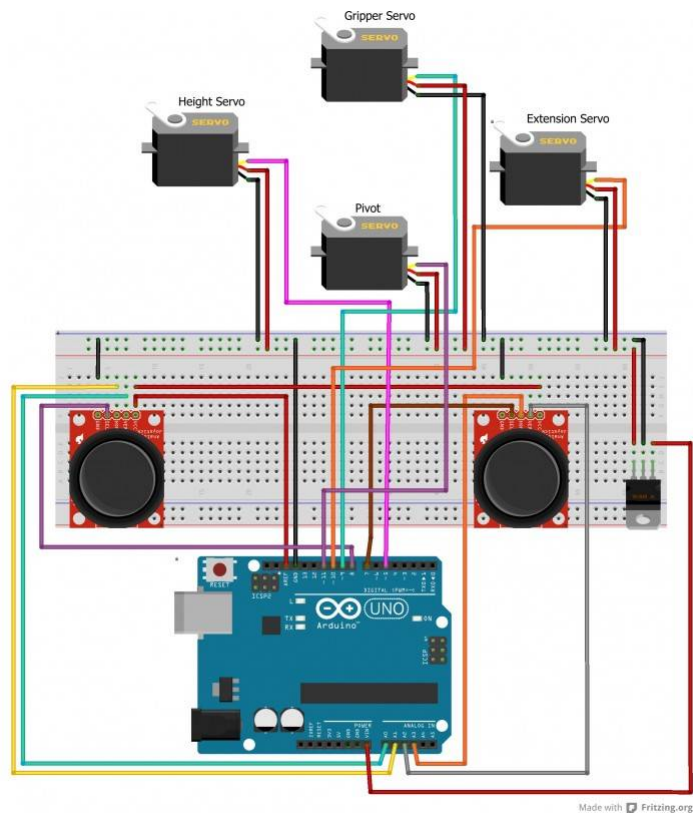


Рисунок 3.4 - Схема підключення сервоприводів і джойстиків до Arduino

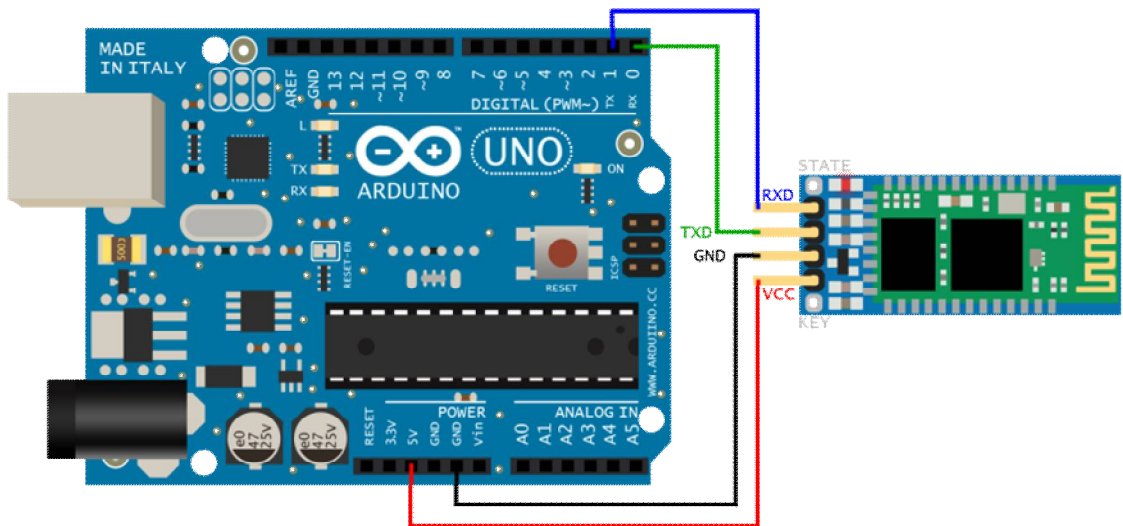


Рисунок 3.5 - Схема підключення Bluetooth модуля

4 ГОЛОСОВЕ УПРАВЛІННЯ

4.1 Аналіз середовищ програмування Arduino

Програмування мікроконтролерів є невід'ємною частиною розробки самостійного електронного пристрою. На даному етапі розвитку електроніки, найбільш популярними мікроконтролерами (МК) є: PIC, MSP, AVR, STM, ARM (процесор).

Для кожного виду МК є вузьконаправлене середовище програмування. Це пов'язано з внутрішньою структурою МК і технічного забезпечення запису програми в пам'ять МК. Якщо проаналізувати існуючі середовища програмування МК, то знайти універсальні середовища програмування МК дуже складно. При аналізі середовищ програмування було виявлено тільки одну універсальну середу програмування мікроконтролерів - FlowCode, функції якої дозволяють програмувати МК PIC, AVR, ARM.

FlowCode - графічна універсальне середовище програмування МК. Програмування здійснюється завдяки побудові логічної структури, тобто блок-схем, аналогічно середовищі HiAsm. Функція експорту дозволяє експортувати написаний код PIC МК в програму AVR МК і навпаки. Доповненням даного середовища програмування є створення HEX-коду, який може бути використувати при прошиванні МК, або при проектуванні схеми з підтримкою МК, наприклад, в середовищі Proteus.

Більшість проаналізованих середовищ програмування МК були призначені для AVR. Серед Algorithm Builder - це графічне середовище програмування МК AVR. Тип графічної розробки в даному середовищі, відрізняється від FlowCode. Як каже автор програми, написання коду здійснюється в 3-5 разів швидше. Серед русифікована і підтримує автоматичне перекодування рядків в ANSI-коду Windows в код русифікованого буквено-цифрового ПКІ. Середовище програмування

об'єднує в собі графічний редактор, компілятор алгоритму, симулятор мікроконтролера, внутрісхемний програматор, функції роботи з EEPROM.

Серед платних середовищ програмування AVR МК є AVR Studio, IAR Systems, Image Craft, WinAVR, CodeVisionAVR.

CodeVisionAVR-популярна умовно-безкоштовне середовище програмування AVR МК. Об'єднує в собі Сі-подібна мова програмування і асемблер. Функції програми дозволяють самостійно прошивати МК і встановлювати fuse-бити і ПЗУ. Кінцевим результатом розробки програми під МК є створення HEX, BIN або ROM-файлу, для прошивки МК за допомогою програматора.

Серед IAR Systems підтримує програмування МК AVR і MSP430, але функції програмування двох МК не є об'єднані в одному середовищі. Для кожного МК були розроблені окремі середовища програмування. Аналогічним чином була розроблена середовище програмування Image Craft (ICC). ICC підтримує Сі-подібний синтаксис і асемблер. IAR Systems і Image Craft в їх склад входить цілеспрямовані бібліотеки по роботі з окремими частинами МК. До складу ICC додана утиліта для генерації коду і ініціалізації периферії МК, впроваджений ANSI Terminal Emulator, який надає можливість працювати з COM-портом.

Серед Code Composer Studio (CCS) для програмування МК MSP заснована на базі стокової універсальної середовища програмування Eclipse. CCS як перераховані вище середовища програмування, має Сі-подібна мова програмування. Дане середовище програмування підтримує розробку для DSP (Digital signal processor) мікроконтролерів, процесорів АРМ сімейства TMS320 і MSP430. Дане середовище в порівнянні з ICC або CCS використовує Java-машину і вимагає більше системних ресурсів.

Energia - середовище програмування для МК MSP430, яка найбільш популярною середовищем програмування серед початківців. Має Сі-подібна мова програмування, але він відрізняється від мов, які використовуються в перерахованих вище середовищах. Мова Energia (і Arduino IDE) більш

зрозумілий, подібний англійським словам. Energia підтримує додаткові бібліотеки, до складу яких входять драйвера для підключення платформи LaunchPad MSP430 на базі ARM Cortex, FraunchPad і lm4f120 StallerPad. Дане середовище програмування є модифікованою версією середовища Arduino IDE. Працює с 1,16МГц МК MSP430 і 80 МГц lm4f120. Впроваджено функція перегляду COM-порту.

Virtual breadboard - середа розробки з підтримкою проектування електричної схеми на базі платформи Arduino під керуванням МК ATMEЛ AVR. При аналізі використовувалася програма версії 4.2.9, яка підтримує розробку програм під МК PIC. Синтаксис подібний середовищі програмування Energia і Arduino IDE. В програму впроваджені приклади, і моделі електронних елементів на основі яких проектується пристрій.

Для розробки програм під МК PIC існує середовище MPLAB IDE. Створення програми проходить в структурованому вигляді, вбудований програмний симулятор моделювання виконання програм в МК з урахуванням стану портів введення-виведення, емулятор роботи МК в масштабі реального часу безпосередньо в пристрої користувача. Розробка проекту ведеться на мові С або асемблер. Ще однією середовищем програмування для МК PIC є мікгоС. Середа мікгоС включає в себе бібліотеки, які підтримують такі пристрої і інтерфейси:

- вбудований аналого-цифровий перекодовщик (АЦП) мікроконтролера;
- вбудовану незалежну пам'ять EEPROM мікроконтролера;
- вбудовані широтно-імпульсні модулятори (PWM) мікроконтролера;
- зовнішні змінні карти пам'яті типу MMC, SD і Compact Flash;
- файлову систему FAT;
- алфавітно-цифровий рідкокристалічні індикатори (LCD PKI);
- графічні рідкокристалічні індикатори (GLCD, РК-дисплей);

– інтерфейси I2C, SPI, 1-Wire, USART, RS-485, CAN, PS / 2, USB (HID) і Ethernet.

Для подальшого програмування була виграна середа Virtual breadboard, із за її зручності та інтуїтивності програмування. Було розроблено код (код представлений у Додатку Б) який дозволяє Arduino [7,8] взаємодіяти з мобільним телефоном.

4.2 Принцип роботи

Після завершення збирання і перевірки працездатності маніпулятора починається розробка голосового управління. Для того щоб відправляти голосові команди до Arduino треба мати пристрій, який обладнаний мікрофоном(для розпізнавання голосу), пристроєм пам'яті(для запам'ятовування команд), процесором (для перетворення звукових хвиль у електричний сигнал) та Bluetooth модулем (для передачі сигналу до Arduino). Усіма цими пристроями володіє мобільний телефон, тому у теорії ми можемо використовувати і його. Одними із переваг розробки програмного забезпечення на пристрої мобільних телефонів є то що вони є дуже поширені і ними володіє більшість людей, не велика кількість платформ на які треба програмувати програму(такі як iOS та Android). Так як у виконавця цього дипломного проекту телефон на базі Android, то було вирішено розробляти програмне забезпечення для голосового управління маніпулятором саме на ньому. Код програми для Android представлений у Додатку А.

Голосове управління буде виконуватися у вигляді виконання заздалегідь закладених команд (схема показана на рис.4.1) , які виконуються по команді, що надходить від Bluetooth модуля, який підключений до мобільного телефону.

На рис. 4.1 зображено зовнішній вигляд програми при її старті. Кнопка «Вибрати устройство» переносить до меню вибору Bluetooth модуля, яке зображене на рим. 4.2.

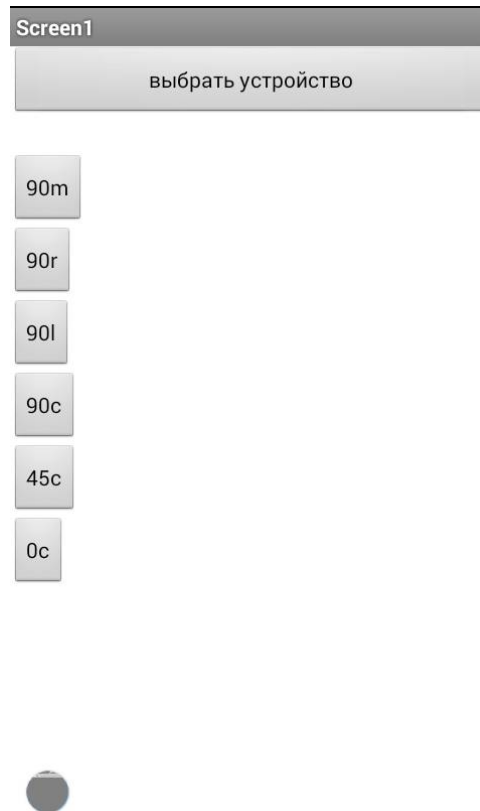


Рисунок 4.1 – Головне меню програмі управління голосом

При переході у меню вибору пристрою треба заздалегідь увімкнути на телефоні Bluetooth модуль, та зробити перше тестове підключення Bluetooth модуля Arduino с телефоном на базі Android стандартними засобами самого мобільного телефона. Це необхідно зробити із-за того що при першому з'єднанні модулів, Bluetooth модуль Arduino буде просити пароль для з'єднання з ним, у нашому випадку пароль був 1234. Якщо не зробити тестового підключення то телефон може зовсім не реагувати на Bluetooth модуль Arduino, ніби він не увімкнений, відобразатися на екрані програми це буде як пустий список(рис.4.3).

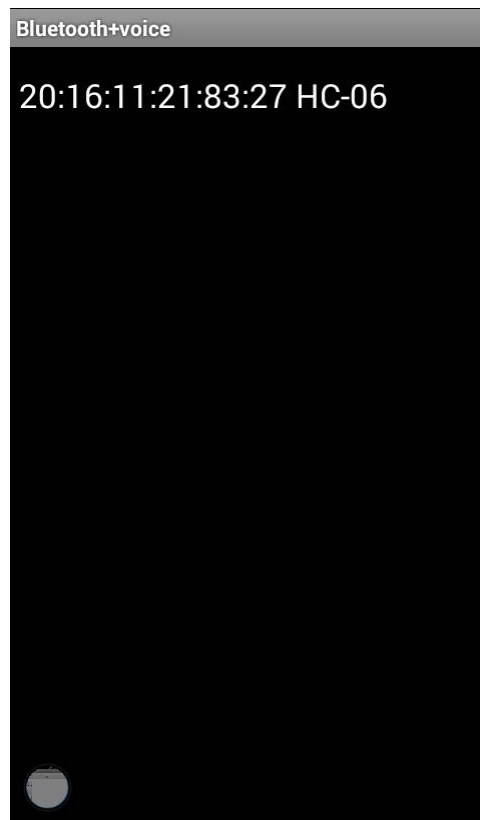


Рисунок 4.2 – Екран підключення Bluetooth модулів

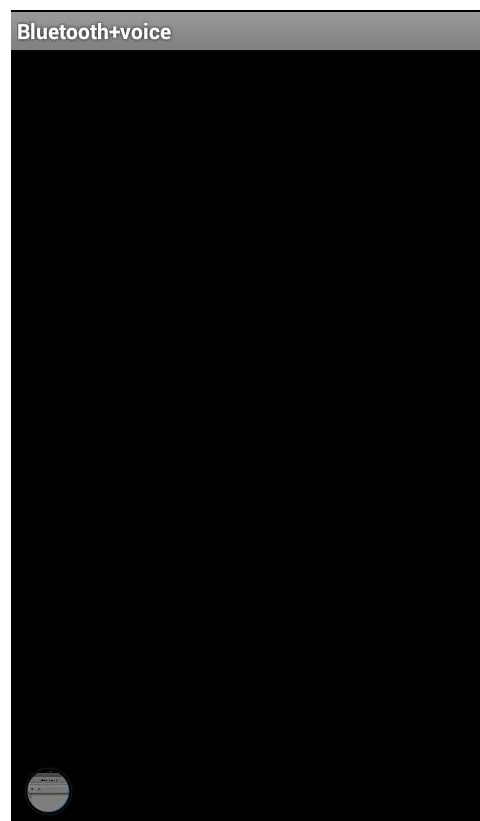


Рисунок 4.3 – Екран телефону при не виконанні першого тестового підключення

При подачі току та відсутності підключення Bluetooth модуль підключений до Arduino буде мигати. При вдалому з'єднанні світлодіоди будуть просто горіти, а у меню з'явиться додаткова кнопка «Голосовая команда»(рис 4.4).



Рисунок 4.4 – Экран телефона при вдалому з'єднанні з Bluetooth модулем

При натисканні на клавішу «Голосовая команда» телефон буде переходити до розпізнавання голосу за допомогою технології Google Talk(рис.4.5).

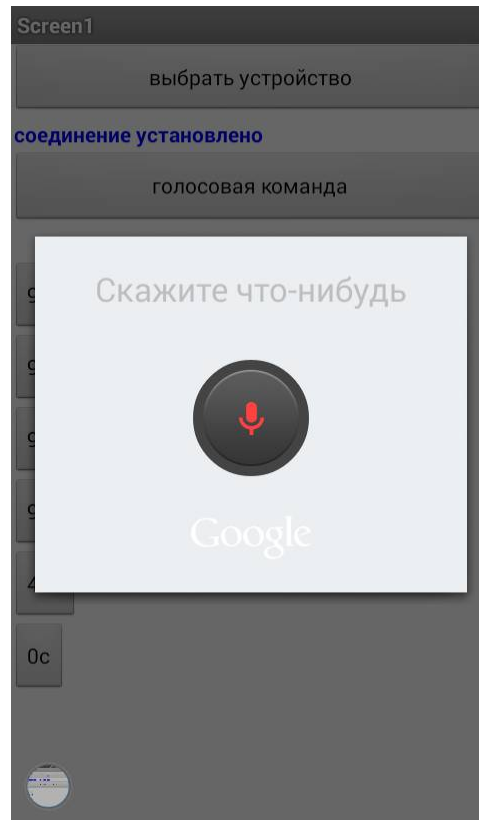


Рисунок 4.5 – Розпізнавання голосу

Після розпізнання слова, програма звіряє його з базою в якій заздалегідь збережені задані команди, які будуть задіяні при вимові певних, наперед заданих голосових команд. При співпаданні слова з командую по Bluetooth модулю передається сигнал який приводить в рух маніпулятор. Схема роботи програми показана на рис 4.6.



Рисунок 4.1 - Схема голосового управління.

5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1 Аналіз потенційних небезпечних чинників при проведенні науково-дослідної роботи

При проведенні розробки голосового управління, в середовищі розробки arduino, проектування проводиться з використанням персонального комп'ютера. На інженерів, які працюють на персональних комп'ютерах (ПК), постійно або періодично діють наступні небезпечні і шкідливі фактори [9-11]:

- забруднення повітря шкідливими речовинами, пилом і позитивними аерофонами;
- невідповідність нормам параметрів мікроклімату;
- широкий спектр випромінювання від дисплея;
- підвищений рівень електромагнітних випромінювань;
- недостатня освітленість робочої зони;
- підвищена яскравість монітора (недостатня освітленість);
- знижена контрастність;
- тривале перебування в тому ж положенні, і повторення тих же рухів призводить до синдрому тривалих статичних навантажень (СДСН);
- невідповідність ергономічних характеристик обладнання нормованим величинам;
- монотонність праці;
- нервово-емоційні стресові навантаження;
- небезпека виникнення пожежі.

Крім того, відповідно до ГОСТ 12.2.003-74 "Оборудование производственное. Общие требования безопасности" та ГОСТ 12.1.019-79

"Електробезпека" можна виділити небезпечні виробничі фактори, характерні для робочого місця оператора: небезпечний рівень напруги в електричному ланцюзі, замикання якої може відбутися через тіло оператора, а також підвищений рівень статичного електрики при неправильно спроектованій робочій зоні.

5.2 Заходи щодо забезпечення безпеки

Заходи щодо техніки безпеки - це система організаційних заходів і технічних засобів, що запобігає або зменшує вплив на працюючих небезпечних і шкідливих виробничих факторів.

Відповідно до класифікації приміщень за ступенем безпеки ураження електричним струмом ГОСТ 12.1.019-79 "Електробезпека", приміщення конструкторського бюро (КБ) відносяться до категорії приміщень без підвищеної небезпеки, тому що відносна вологість повітря не перевищує 75%, температура не більше 27°C, відсутня хімічно агресивне середовище.

Робочі місця інженерів відповідають ГОСТ 12.2.003-74 "Оборудование производственное. Общие требования безопасности" та ГОСТ 12.1.019-79 "Електробезпека". Складові частини виробничого обладнання (у тому числі кабелі, проводи тощо) виконані з таким розрахунком, щоб виключити можливість їх випадкового пошкодження, яке викликає небезпеку. Конструкцією виробничого обладнання передбачений захист від ураження електричним струмом. Для доступу до струмоведучих частин при ремонті або огляді необхідно розкрутити гвинти і зняти кришку. Металеві частини виробничого обладнання, які можуть в результаті пошкодження ізоляції опинитися під електричною напругою небезпечної величини, заземлені або занулені. Провідники занулення обрані так, щоб при замиканні на корпус або на нульовий провід виникав струм короткого замикання, сила якого перевищує не менше ніж у три рази номінальну силу струму плавкої вставки найближчого запобіжника.

Дисплей комп'ютера здатний накопичувати статичний заряд. Небезпека зарядів статичного електрики виявляється в можливості виникнення електричних зарядів в просторі, вплив на обслуговуючий персонал і порушенні нормального ходу робочого процесу. Вплив статичного електрики на обслуговуючий персонал позначається в розрядах статичних зарядів через людину і вплив електростатичного поля. Безпосередньо струм розряду не є небезпечним, тому що час його протікання через тіло людини мале (дорівнює декількох мілісекунд). Такі короточасні імпульси можуть викликати електричні удари, які супроводжуються болючим уколом, що призводять до переляку.

Для зниження рівня статичної електрики передбачено екранування джерела статичної електрики (монітора) захисним екраном і заземлення екрана відповідно до ДСанПіН 3.3.2.007-98 "Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин".

У схемі електричних ланцюгів передбачено пристрій, який централізовано відключає від живильної мережі всі електричні ланцюги. У відділі всі мережеві розетки мають написи зі значенням напруги і застережливий знак.

До основних ергономічних завданням організації робочого місця згідно ДСТУ ГОСТ 12.2.061: 2009 "Загальні вимоги безпеки до робочих місць" відносяться:

- визначення просторових параметрів робочого місця та його елементів, які відповідають антропометричним характеристикам контингенту працюючих;

- оптимальне розміщення елементів робочого місця щодо користувача на основі аналізу його діяльності.

Устаткування і організація робочого місця працюючих з ПК забезпечує відповідність конструкції всіх елементів робочого місця та їх взаємного розташування ергономічним вимогам з урахуванням характеру і

особливостей трудової діяльності (ГОСТ 12.2.032-78 "Система стандартів безпеки праці. Робоче місце при виконанні робіт сидячи. Загальні ергономічні вимоги ", ГОСТ 22.269-76" Система "людина-машина". Робоче місце оператора. Взаємне розташування елементів робочого місця. Загальні ергономічні вимоги ", ГОСТ 21.889-76" Система "людина-машина" Крісло людини-оператора. Загальні ергономічні вимоги ").

5.3 Заходи щодо забезпечення виробничої санітарії та гігієни праці

Для нормальної експлуатації машин електронної та обчислювальної техніки і високопродуктивної праці робітників та інженерно-технічного персоналу в процесі проектування пристрою передбачений наступний комплекс заходів:

- а) забезпечення оптимального мікроклімату;
- б) застосування оптимального освітлення;
- в) створення оптимальних умов праці на робочому місці;
- г) зниження психоемоційного та фізичного навантаження.

Загальні санітарно-гігієнічні вимоги до повітря приміщення, в якому розташоване КБ відповідає вимогам ГОСТ 12.1.005-88. У приміщенні застосована система кондиціонування, а також регулярно проводиться провітрювання. У приміщенні регулярно здійснюється вологе прибирання.

Згідно ДБН В.2.5-28-2006 в приміщенні застосований змішаний тип освітлення, що відповідає II-III розряду зорової роботи. Як штучного джерела освітлення застосовані лампи денного світла, які разом зі світильниками вмонтовані в стелю. У світлий час доби в основному застосовується природне освітлення, яке забезпечується наявністю вікон. При необхідності (захист від прямих сонячних променів), вікна закривають жалюзі. Увечері, або в похмуру погоду персонал включає штучне освітлення.

Правильна організація робочого місця - створення на робочому місці необхідних умов для продуктивної роботи і виконання роботи високої якості,

при найбільш повному використанні обладнання, малих розтратах фізичної та емоційної енергії працівника, підвищення змістовності та привабливості роботи, збереження здоров'я працюючих.

При розміщенні обладнання витримуються необхідні проміжки між обладнанням, відстані від стін, які забезпечують свободу пересування людей, зручність виконання робіт і безпеку працюючих. Робочі місця операторів ПК, а так само ділянки підготовки технічних носіїв інформації розташовані в ряди. Розташування місць може бути дворядним, трирядним, чотирирядним, а також прямим і поперечним.

На організацію роботи впливає конструкція і параметри основного і допоміжного обладнання, які відповідають вимогам ергономіки: оптимальному розподілу функцій в системі людина - машина. Конструкція обладнання відповідає антропометричним і психофізіологічним даними організму працюючого. Витримані допустимі параметри санітарно-гігієнічних умов праці, а також безпеки експлуатаційного обладнання.

Конструкція робочого столу забезпечує оптимальне розміщення на робочій поверхні використовуваного обладнання, враховуючи його масогабаритні і конструктивні особливості.

Висота робочої поверхні столу регулюється в межах 680-800 мм, при відсутності такої можливості висота робочої поверхні столу складає 725 мм. Модульними розмірами робочої поверхні столу для ПК, розраховуються конструктивні розміри, слід вважати: ширину 800, 1000, 1200, 1400 мм, глибину 800 і 1000 мм при нерегульованій його висоті, що дорівнює 725 мм.

Робоче місце обладнане підставкою для ніг, має ширину не менше 300 мм, глибину не менше 400 мм, регулювання по висоті в межах до 150 мм і за кутом нахилу опорної поверхні підставки до 200. Підставка має рифлену поверхню.

Конструкція робочого стільця (крісла) підтримує раціональну робочу позу при роботі з ПК, дозволяє змінювати позу з метою зниження статистичного напруги м'язів шийно-плечової області і спини для

попередження стомлення. Робочий стілець (крісло) є підйомно - поворотним і регульованим по висоті і кутам нахилу сидіння і спинки, а також відстані від переднього краю сидіння.

Конструкція стільця повинна забезпечувати максимум результатів праці. Досягти цього можна, лише освоївши раціональні методи і прийоми праці на робочому місці. Тільки вони дозволяють виконати задану роботу якісно, в мінімальні терміни і без зайвої напруги.

Специфіка роботи таких працівників полягає в великих зорових навантаженнях разом з маленькою активністю, монотонністю виконаних операцій, вимушеної робочої позою. Ці фактори негативно відбиваються на самопочутті працюючого.

При організації робочого місця дуже важливим фактором є робоча поза працівника, тобто положення його корпусу, голови, рук і ніг по відношенню до знарядь праці. Якщо працівник працює сидячи, йому необхідно забезпечити правильну і зручну посадку, яка досягається пристроєм опори для спини, рук, ніг, правильною конструкцією сидіння, яке сприяє рівномірному розподілу маси тіла.

Всі матеріальні елементи робочого столу ділять на предмети постійного, тимчасового користування і з урахуванням цього розташовують у видатному порядку на місцях постійного зберігання, це економить трудові руху і сили працюючого.

Клавіатуру комп'ютера краще розташовувати на відстані 10-15 мм від краю столу, тоді зап'ястя рук будуть спиратися на стіл. Бажано придбати спеціальну підставку під зап'ястя, що, як стверджують медики, допоможе уникнути хвороби кистей.

Для ефективного використання маніпулятора типу «миша» застосовується спеціальний «килимок» - планшет. Килимок - планшет задовольняє основним критеріям: по-перше, добре тримається на поверхні столу, по-друге, матеріал верхньої поверхні планшета забезпечує вільне ковзання миші.

Введення текстової інформації з клавіатури полегшують підставки для документів. Вони можуть або кріпитися до монітора, або встановлюватися безпосередньо на столі. Багато підставки оснащені лінійками для виділення набираемого рядка.

Інструмент, обладнання та предмети праці знаходяться на відстані 560-750 мм на рівні рук працівника, так їх використання не призводить до зайвих рухомим нахилам. Важливим елементом раціонального планування робочого місця є облік індивідуальних антропометричних психофізіологічних даних працюючого.

Робочі місця обладнані відповідними меблями та інвентарем, які відповідають найбільш комфортабельним умовам роботи і вимогам фізіології, психологія та естетики.

Нижче представлений розрахунок повітрообміну в приміщенні КБ.

У прикладі необхідно визначити необхідний повітрообмін і його кратність для вентиляційної системи КБ, що має довжину $a = 10$ м, ширину $b = 6$ м, висоту $h = 3,5$ м. У повітряне середовище КБ виділяється пил в кількості $W = 1,5$ г / год (для даного виду пилу ГДК = 4 мг/м³), концентрація пилу в робочій зоні $C_{p.z.} = 2,8$ мг/м³, в припливно повітрі $C_{п} = 0,3$ мг/м³, концентрація пилу в повітрі, що видаляється з КБ дорівнює концентрація її в робочій зоні ($C_{yx} = C_{p.z.}$), тобто пил рівномірно розподілена в повітрі. Кількість повітря, що забирається з робочої зони місцевими відсмоктувачами, так само $G_m = 50$ м³/ год.

Рішення даного завдання має наступний алгоритм:

Розрахунок обсягу КБ здійснюється за формулою 6.1:

$$V = a \times b \times h, \text{ м}^3 \quad (6.1)$$

$$V = 10 \times 6 \times 3,5 = 210 \text{ м}^3$$

Розрахунок необхідного повітрообміну здійснюється за формулою 6.2:

$$G_{пр} = G_m + \frac{W - G_m(C_{p.z.} - C_{п})}{C_{p.z.} - C_{п}}, \text{ м}^3/\text{ч} \quad (6.2)$$

$$G_{\text{пр}} = 50 + \frac{1500 - 50(2,8 - 0,3)}{2,8 - 0,3} = \frac{1375}{2,5} = 550 \text{ м}^3/\text{ч}$$

Кратність повітрообміну в КБ розраховується за формулою 6.3:

$$K = \frac{G_{\text{пр}}}{V}, \text{ 1/ч} \quad (6.3)$$

$$K = \frac{550}{210} = 2,6 \text{ 1/ч}$$

Тобто за 1 годину повітря в КБ повинно змінитися 2,6 рази.

5.4 Заходи з пожежної безпеки

Відповідно до ДБН В.2.5-13-98 "Пожежна автоматика будинків і споруд" найбільш частими причинами пожеж є наступні:

- порушення режимних вимог;
- самозаймання і самозапалювання;
- розряди статичної та атмосферної електрики;
- несправність і неправильна експлуатація обладнання.

Джерелами загоряння можуть бути коротке замикання, перевантаження проводів, перегріті опірні поверхні, несправність апаратури. Окислювачем, як правило, служить кисень. Для електронних пристроїв характерно часта поява джерел відкритого вогню при коротких замиканнях, пробоях і перевантаженнях. Однак потужність і тривалість дії цих джерел запалення порівняно малі, тому горіння, як правило, не розвивається.

Для відводу надмірної теплоти від ПК передбачені системи вентиляції і кондиціонування повітря.

Одним із важливих заходів щодо пожежної безпеки є установка в приміщенні КБ пожежної сигналізації. Вона дає можливість швидко погасити вогнище загоряння. У приміщенні встановлені променеві електричні системи сигналізації ЕРС типу ТОЛ 10/100 на 10-100 променів. Як сповіщувачі застосовані теплові і димові сповіщувачі типу КД-1.

Приміщення КБ відноситься до категорії «Д» по вибух про - і пожежобезпеки відповідно «СНиП II-2-80 "Противопожежні норми проектування будівель і споруд"».

Будівля, в якому зазвичай знаходиться КБ, є спорудою з несучими і загороджували конструкціями з природного або штучного каміння, бетону, залізобетону з використанням негорючих матеріалів. З огляду на конструктивні характеристики споруди і категорію виробництва за вибух про - і пожежобезпеки, ступінь вогнестійкості повинна бути не нижче III.

У разі виникнення пожежі, евакуація працівників може проводитися через пожежні виходи. Для уникнення пожеж на виробництві:

- розроблені комплексні заходи щодо забезпечення пожежної безпеки;
- відповідно до нормативних актів з пожежної безпеки розроблені положення та інструкції, які діють в межах підприємства, установи та організації, здійснюється постійний контроль за їх виконанням;

- забезпечується дотримання протипожежних вимог, норм, правил, а також виконання вимог, розпоряджень і постанов органів державного пожежного нагляду;

- утримуються в справному стані засоби протипожежного захисту, пожежна техніка, обладнання та інвентар, не допускається їх використання не за призначенням;

- представляються за вимогою державної пожежної охорони відомості та документи про стан пожежної безпеки об'єктів і виробленої ними продукції;

- проводяться заходи щодо впровадження автоматичних засобів виявлення та гасіння пожеж, і використання для цієї мети виробничої автоматики.

Усі працівники при прийнятті на роботу щорічно за місцем роботи проходять інструктажі з питань пожежної безпеки.

Приміщення обладнане наступними елементами пожежегасіння:

- Вогнегасник ОУ- 3;

- Вогнегасник ОП-1 "Момент".

Така кількість вогнегасників відповідає вимогам ISO3941 - 77 "Пожежі. Класифікація", якими передбачене обов'язкова наявність двох вогнегасників на 100 м² площі для приміщень типу конструкторських бюро.

5.5 Заходи з цивільного захисту

Заходами щодо евакуації називаються розосередження і евакуація населення з категорійних міст у заміську зону. Вони організовуються і проводяться відповідно до плану ЦО району (об'єкта) та розпоряджень старшого начальника ЦО.

Розосередження - організований вивіз з міст та інших населених пунктів і розміщення в заміській зоні, вільній від роботи зміни робітників і службовців тих об'єктів, які продовжують роботу в умовах НС.

Розосереджується робітники і службовці після розселення в заміській зоні позмінно виїжджають в город для роботи на підприємствах, а після роботи повертаються в заміську зону для відпочинку.

Евакуація - організований вивіз або вивід з населених пунктів і розміщення в заміській зоні решти населення. На відміну від розосереджених, евакуйовані постійно проживають в заміській зоні до особливого розпорядження.

Заміська зона - територія, розташована за межами зон можливого ураження в містах.

Кожному підприємству і установі міста, з якого планується розосередження і евакуація, в заміській зоні призначається район розміщення населення, який включає один або кілька розташованих поруч населених пунктів.

Видалення районів розосередження від міста має забезпечувати безпеку робітників і службовців, але час на переїзд в город і повернення назад повинен бути мінімальний.

Відповідальність за евакуацію робітників і службовців та їх сімей в призначені райони покладається на начальників ЦО і штабів ЦО підприємств і установ.

Розселення робітників і службовців виконується з дотриманням виробничого принципу, що забезпечує цілісність підприємства, полегшує відправку робочих змін на об'єкт і забезпечення населення харчуванням і медичним обслуговуванням.

Евакуйоване населення розміщують в більш віддалених районах заміської зони, а населення, евакуйоване з зон можливих стихійних лих, - в населених пунктах, що знаходяться поблизу цих зон.

Розосередження і евакуація в багато разів знижують щільність населення і, отже, можливі втрати.

Розосередження і евакуація здійснюються:

- робітників і службовців і членів їх сімей - за виробничим принципом (по лінії об'єктів);
- населення не пов'язаного з виробництвом - за територіальним принципом (тобто за місцем проживання через домоуправління і житлово - експлуатаційні контори).

Щоб евакуація пройшла організовано, її завчасно планують місцеві адміністративні органи і створені при них евакуаційні комісії; попередньо визначаються склад, місця розташування і порядок роботи збірних евакуаційних пунктів (ЗЕП), а в сільській місцевості, куди вивозиться населення, створюються прийомні евакуаційні комісії та приймальні евакуаційні пункти (ПЕП).

ЗЕП призначені для збору, реєстрації та організованого відправлення населення. Коли людей вивозять залізничним і водним транспортом, ЗЕП розміщують поблизу станцій, портів і на підприємствах, які мають залізничні під'їзні шляхи, морські, річкові причали.

На кожному підприємстві і установі повинні бути евакуаційні списки, які разом з паспортами є основними документами для обліку, розміщення і забезпечення всім необхідним в районах розселення.

Евакуація може здійснюватися усіма видами транспорту і пішки. Для швидкої евакуації застосовується комбінований спосіб її проведення, тобто перший порядок об'єднується з максимальним використанням усіх видів транспорту.

Коли надійшло розпорядження про початок розосередження і евакуації, начальник ГО повідомляє про це начальникам виробничих підрозділів, останні оповіщають робітників і службовців, а ті - членів своїх сімей.

Для підтримки громадського порядку на об'єктах ЗЕП і ПЕП, на станціях (пристанях, пунктах) посадки і висадки, в місцях розселення в заміській зоні виставляються пости охорони громадського порядку, організовується патрулювання. До виконання цих заходів залучаються формування охорони громадського порядку (команди, групи), що створюються з відомчої воєнізованої і сторожової охорони та добровільних народних дружин.

6 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

Проблема представленого дослідження носить актуальний характер в сучасних умовах. Про це свідчить активізація активності досліджень і нових розробок в даному напрямі.

Тема "Системи управління роботою - маніпулятором на базі arduino" вивчається на стику відразу декількох взаємопов'язаних дисциплін, таких як програмування, розпізнавання голосу і радіоелектроніка.

Висока значимість і недостатня практична розробленість проблеми визначають безсумнівну новизну даного дослідження, а, отже, і підвищують ступінь його значущості.

Подальша увага до питання необхідна з метою більш глибокого і обґрунтованого вирішення приватних актуальних проблем тематики дослідження.

Актуальність цієї роботи зумовлена, з одного боку, досить великим інтересом до теми в сучасній науці з боку державних і приватних підприємств радіоелектронного напрямку, з іншого боку, її недостатньою розробленістю. Розгляд питань пов'язаних з даною тематикою носить як теоретичну, так і практичну значимість.

Розроблений модуль дозволить управляти роботом маніпулятором будь-якої моделі за допомогою плати, що в свою чергу відкриє нові способи керування роботів. Плануються продаж 300 копій по 500 гривень за копію.

6.1 Планування розробки програмного виробу

Основні роботи, необхідні для розробки програмного продукту і їх трудомісткість представлені в таб. 6.1.

Необхідні початкові дані:

- час на розробку - до 2 міс.;
- програміст працює - до 2 міс. (40дн.) із заробітною платою (ЗП) 11000 грн./міс.;
- консультант працює - до 0,25 міс. (5дн.) із ЗП 7000 грн./міс.

Завершуючим етапом планування розробки програмного продукту [12] є звідний стрічковий графік розробки програмного продукту, побудований за даними таблиці. 6.1. Лінійний графік Ганта представлений в таблиці. 6.2.

Таблиця 6.1 - Характеристика робіт по розробці програми

Найменування роботи	Трудомісткість		Виконавці		Тривалість
	люд.- дн.	%до підсумку	спеціальність	кіл. люд.	днів
1. Дослідження вимог до проєктованого продукту	10	22,22	програміст консультант	2	5
2. Розробка ТЗ	5	11,11	програміст	1	5
3. Проєктування загальної структури ПО	5	11,11	програміст	1	5
4. Розробка схеми функціонування програми	5	11,11	програміст	1	5
5. Створення програмного коду	14	31,11	програміст	1	14

Продовження табл. 6.1

6. Тестування та налагодження програмного продукту	3	6,66	програміст	1	3
7. Складання програмної документації	3	6,66	програміст	1	3
Всього	45	100,00			40

Таблиця 6.2 - Лінійний графік Ганта розробки програмного продукту

Найменування роботи	Календарний період, дні							
	09.10 - 13.10	16.10 - 20.10	23.10 - 26.8	30.10 - 03.11	06.11 - 10.11	13.11 - 16.9	20.11 - 24.11	26.9 - 01.12
1. Дослідження і аналіз вимог до проєктованого продукту	■							
2. Розробка ТЗ		■						
3. Проєктування загальної структури ПО			■					
4. Розробка схеми функціонування програми				■				
5. Створення програмного коду					■	■	■	
6. Тестування та налагодження програмного продукту							■	
7. Складання програмної документації								■

- програміст
 - консультант

6.2 Визначення витрат на розробку програми

6.2.1 Розрахунок основної заробітної плати

Витрати по цій статті складаються з планового фонду заробітної плати усіх категорій працівників, зайнятих в розробці програмного продукту.

Розрахунок заробітної плати ведеться на основі відомостей про трудомісткість (див. табл.6.1). Результати розрахунків зведені в таблиці 6.3.

Таблиця 6.3 - Розрахунок основної заробітної плати

Посада виконавця	К-ть, люд.	Місячний оклад	К-ть, міс. роботи	Премії і доплати, грн.	Середн. міс. зарплата, грн.	Суммарн. зарплата, грн.
Програміст	1	11000	2	–	11000	22000
Консультант	1	7000	0,25	–	7000	1750
Всього	2					23750

6.2.2 Розрахунок додаткової заробітної плати

Додаткову заробітну плату приймають рівною 10% від основної заробітної плати. Розрахунок основної і додаткової заробітної плати приведені в таблицю. 6.4.

Таблиця 6.4 - Основна і додаткова заробітна плата

Посада виконавця	Додаткова заробітна плата, грн.	Сума основної і додат. заробітної плати, грн.
Програміст	2200	24200
Консультант	175	1925
Разом	2375	26125

6.2.3 Відрахування на єдиний соціальний внесок

Ці відрахування визначають в процентному відношенні від суми основної і додаткової зарплат з урахуванням премій і доплат. Премії і доплати в нашому випадку дорівнюють 0. Відрахування на соціальне страхування складають 22%.

$$\text{ОТЧ} = (\text{ЗП}_{\text{осн}} + \text{ЗП}_{\text{дод}}) \cdot 0,22 \quad (6.1)$$

де $\text{ЗП}_{\text{осн}}$ - основна заробітна плата, грн.

$\text{ЗП}_{\text{дод}}$ - додаткова заробітна плата, грн.

$$\text{ОТЧ} = (23750 + 2375) \cdot 0,22 = 5747,5 \text{ грн.}$$

6.2.4 Визначення витрат на матеріали

У цю статтю входить вартість основних і допоміжних матеріалів, комплектуючих виробів, необхідних для розробки програми.

Матеріали, які знадобилися для розробки представлені в таблиці. 6.5.

Таблиця 6.5 - Витратні матеріали

Матеріал	Кількість	Вартість, грн.
Папір формату А4	500 листів	40
CD - RW диск	1 шт.	5,0
Тонер для принтера	1 шт.	120

Витрати на матеріали розраховуються по формулі 6.2.

$$M = \sum_{i=1}^n (C_i \cdot N_i \cdot (1 + K_{m.z.}) - C_{io} \cdot N_{io}) \quad (6.2)$$

де M - витрати на матеріали, покупні напівфабрикати і комплектуючі вироби, грн.

$K_{m.z.}$ – коефіцієнт, що враховує транспортно-заготівельні витрати (0,03 - 0,1)

C_i – ціна i -го найменування матеріалу і що комплектує, грн.

N_i – ціна в i -му матеріалі, напівфабрикаті, що комплектує

C_{io} – ціна поворотних відходів i -го найменування матеріалу, грн.

N_{io} – кількість поворотних відходів i -го найменування

n – кількість найменувань матеріалів, напівфабрикатів, комплектуючих.

Розрахуємо витрати на матеріали:

$$M = (40+5,0+120) \cdot (1+0,05) = 173,25 \text{ грн.}$$

Разом витрати на матеріали склали 173,25 грн.

6.2.5 Витрати на спеціальне устаткування

У цю статтю входять витрати на оплату машинного часу ЕОМ.

Витрати на оплату машинного часу ЕОМ для написання і відладки цієї програми визначають по формулі:

$$C_{мо} = P_{ЭКС} \cdot t_{мо}, \quad (6.3)$$

де $C_{мо}$ – витрати на оплату машинного часу, грн.;

$P_{ЭКС}$ – експлуатаційні витрати на одну годину маш. часу, грн.;

$t_{мо}$ – машинний час ЕОМ для написання і відладки цього програмного продукту, ч.

Для розрахунку експлуатаційних витрат на одну годину машинного часу складемо кошторис витрат на зміст і експлуатацію цього устаткування.

Амортизаційні відрахування визначають по формулі:

$$A = \Phi_{б} \cdot \frac{H_a}{100}, \quad (6.4)$$

де $\Phi_{б}$ – балансова вартість обчислювальної техніки, грн.;

H_a – норма амортизаційних відрахувань на повне відновлення обчислювальної техніки.

$$A = 12000 \cdot \frac{25}{100} = 3000 \text{ грн.}$$

Балансова вартість машини визначається як вартість машини плюс 5% на транспортні витрати і монтаж.

Витрати на електроенергію:

$$C_{э} = N_n \cdot \Phi_{эф} \cdot K_{зв} \cdot K_{зм} \cdot Э_{мо}, \quad (6.5)$$

де N – номінальна потужність ЕОМ, кВт;

$\Phi_{\text{эф}}$ – річний ефективний фонд часу роботи ЕОМ, ч.;

$K_{\text{зв}}$ – середній коефіцієнт завантаження за часом;

$K_{\text{зм}}$ – коефіцієнт завантаження по потужності;

$C_{\text{ел.е}}$ – ціна одного кВт·ч електроенергії, грн.

$$C_3 = 0,2 \cdot 1800 \cdot 0,9 \cdot 0,7 \cdot 1,765 = 400,31 \text{ грн.}$$

Зарплата обслуговуючого персоналу розраховується по формулі 6.6:

$$ЗП_{\text{обсл}} = \Phi ЗП_{\text{з}} \cdot (1 + K_{\text{от}}) \cdot \frac{t_{\text{обсл}}}{\Phi_{\text{эф.обсл}}}, \quad (6.6)$$

де $\Phi ЗП_{\text{з}}$ – річний фонд зарплати обслуговуючих працівників, грн.;

$K_{\text{от}}$ – коефіцієнт відрахувань на соціальне страхування і до інших фондів, %;

$t_{\text{обсл}}$ – час впродовж року, необхідний на техобслуговування ЕОМ, г/рік.

Машину обслуговує один електронщик з основною заробітною платою 3200 грн./міс.

$$\Phi ЗП_{\text{г}} = 3200 \cdot 12 = 38400 \text{ грн.};$$

$$K_{\text{от}} = 0,22; \quad t_{\text{обсл}} = 10 \text{ г./міс.}; \quad \Phi_{\text{эф.обсл}} = 1750 \text{ г.};$$

$$ЗП_{\text{обсл}} = 38400 \cdot (1 + 0,22) \cdot 10 / 1750 = 267,71 \text{ грн.}$$

Сума витрат по статті "Поточний ремонт устаткування" дорівнює 3% від балансової вартості устаткування, що дорівнює 360,00 грн.

Інші витрати складають 5% від суми попередніх.

Зведемо отримані дані в таблиці. 6.6.

Таблиця 6.6 - Кошторис витрат на зміст і експлуатацію устаткування

Найменування статей витрат	Сума, грн.
Амортизація устаткування	3000,00
Експлуатація устаткування	400,31
Зарплата обслуговуючого персоналу з відрахуваннями	267,71
Поточний ремонт устаткування	360,00
Інші витрати	201,41
Всього	4229,43

Експлуатаційні витрати на одну годину машинного періоду складають:

$$P_{\text{екс}} = 4229,43/1800 = 2,35 \text{ грн./ч.}$$

Розробка і відладка програмного продукту займає 640 годин машинного часу (40 днів по 8 робочих годин), тобто $t_{\text{мо}} = 320$ ч.

Витрати на оплату машинного часу склали:

$$C_{\text{мо}} = 2,35 \cdot 320 = 752,00 \text{ грн.}$$

6.2.6 Накладні витрати

До накладних витрат відносяться витрати на загальне управління і загальногосподарські потреби (заробітна плата апарату управління, канцелярські витрати і так далі), зміст і експлуатацію будинків. Накладні витрати включаються у вартість розробки програми непрямым шляхом - у відсотках до основної заробітної плати розробників. В даному випадку накладні витрати складають 40% до основної заробітної плати розробників, що складає 9500 грн.

Таблиця 6.7 - Калькуляція кошторисної вартості робіт по розробці програми і ціна програмного продукту

Найменування статей витрат	Сума, грн.	Питома вага до підсумку, %
1. Основна заробітна плата	23750,00	56,15
2. Додаткова заробітна плата	2375,00	5,61
3. Соціальні відрахування	5747,50	13,59
4. Витрати на матеріали	173,25	0,41
5. Витрати на оплату машинного часу	752,00	1,78
6. Накладні витрати	9500,00	22,46
7. Разом витрати	42297,25	100,00
8. Прибуток	12689,18	-
9. Всього	54986,43	-
10. НДС (20%)	10997,29	-
Всього	65983,72	-

6.3 Бальна оцінка економічної ефективності науково-дослідної роботи

Наукові дослідження, прямий підрахунок економічної ефективності яких неможливий, оцінюють за допомогою бальної системи.

Бальна оцінка проводиться за такими показниками:

- важливість розробки $Do_1 = 1$, вибирається з табл. 6.8 ;

Таблиця 6.8 - Шкала для оцінки важливості розробки K_1 .

п / п	показник	бали
1	Ініціативна робота, яка не є частиною комплексної програми або завданням відомчих органів	1
2	Робота, яка виконується за договором про науково-технічної допомоги	3
3	Робота є частиною відомчої програми	5
4	Робота є частиною відомчої комплексної програми	7
5	Робота є частиною міжнародної комплексної програми	8

- можливість використання результатів розробки $K_2 = 5$ (табл. 6.9);

Таблиця 6.9 - Шкала оцінки можливості використання результатів розробки K_2 .

п / п	показник	бали
1	В даному підрозділі	1
2	У даній організації	3
3	У багатьох організаціях	5
4	У масштабах країни	8

- теоретична значущість і рівень новизни дослідження $Do_3 = 2$ (табл. 6.10);

Таблиця 6.10 - Шкала оцінки теоретичної значущості і рівня новизни дослідження Do_3 .

п / п	показник	бали
1	Аналіз, узагальнення та класифікація відомої інформації. Подібні результати були відомі в досліджуваній області	2
2	Отримання нової інформації, яка доповнює знання про сутність досліджуваних процесів, невідомих в досліджуваній області	3
3	Отримання нової інформації, яка змінює уявлення про сутність досліджуваних процесів, невідомих раніше	5
4	Створення нових теорій, методик	6
5	Отримання інформації, яка сприяє формуванню напрямків, невідомих раніше	8

- складність розробки $Do_4 = 3$ (табл. 6.11).

Таблиця 6.11 - Шкала оцінки показників складності дослідження До 4.

п / п	показник	бали
	Робота виконується одним підрозділом, витрати менше 10000 грн.	1
	Робота виконується одним підрозділом, витрати 10000-50000 грн.	3
	Робота виконується одним підрозділом, витрати 50000-100000 грн.	5
	Робота виконується одним підрозділом, витрати 100000-500000 грн.	7
	Робота виконується декількома організаціями, витрати більше 500000 грн.	8

Загальна оцінка встановлюється за твором коефіцієнтів:

$$B = K_1 \cdot K_2 \cdot K_3 \cdot K_4 \quad (6.7)$$

$$B = 1 * 5 * 2 * 3 = 30 \text{ балів}$$

Питома ефект (УЕ) на кожен бал - 20000 грн.

Загальний ефект (Е) від розробки становить:

$$E = 20000 \cdot K_1 \cdot K_2 \cdot K_3 \cdot K_4 \quad (6.8)$$

$$E = 20000 * 1 * 5 * 2 * 3 = 60000 \text{ грн}$$

Економічна проекту визначається за допомогою коефіцієнта ефективності, який характеризує частину загального ефекту від розробки, який припадає на одну грн. витрат (собівартості проекту):

$$K_e = \frac{E}{K_{np}} \quad (6.9)$$

$$K_{\epsilon} = \frac{60000}{42297,25} = 1,42$$

Термін окупності $T_{ок}$ проекту розраховується за формулою:

$$T_{ок} = \frac{1}{K_{\epsilon}} \quad (6.10)$$

$$T_{ок} = \frac{1}{1,42} = 0,7$$

6.4 Оцінка економічної ефективності

Плануються продаж 300 копій по 500 гривень за копію. Таким чином очікується отримати надходження від продажів 150000 гривень. З урахуванням ризику 50% надходження від продажів складуть 75000 гривень, що більше вартості програмного забезпечення, в якій вже була врахована рентабельність 30 відсотків.

Таким чином, розробка даного програмного забезпечення є економічно доцільною.

ВИСНОВОК

У роботі показана розробка пристрою, який дозволить віддалено управляти роботом маніпулятором. Даний прилад може використовуватися шляхом установки програмного забезпечення на промисловий маніпулятор, в нашому випадку це IRB 460, і на мобільний телефон, для віддаленого контролю даного маніпулятора, що дасть можливість в екстреній ситуації зупинити або перенаправити маніпулятор. Також це дасть можливість управляти маніпулятором людям, які в даній ситуації або часу не можуть управляти маніпулятором через джойстик.

ПЕРЕЛІК ПОСИЛАНЬ

1. **Иванов А.А.** Основы робототехники: учебное пособие [Текст] / А. А. Иванов. - М. : "ФОРУМ", 2015. - 224 с.
2. **Егоров О.Д.** Прикладная механика мехатронных устройств [Текст] / О.Д. Егоров; М.: ФГБОУ ВПО МГТУ "СТАНКИН", 2013. - 229 с.
3. **Егоров О.Д.** Конструирование механизмов роботов [Текст] / О.Д. Егоров; М.: ФГБОУ ВПО МГТУ "СТАНКИН", 2012. - 444 с.
4. Основы проектирования машин: учебное пособие [Текст] / И.Ф. Дьяков, В.Я. Недоводеев, В.Н. Демокритов, А.В. Олешкевич. – Изд. 2-е, перераб. И доп. – Ульяновск: УлГТУ, 2012. – 127 с.
5. Довідкова інформація IRB 460 [Електронний ресурс]. – Режим доступу : <http://new.abb.com/products/robotics/industrial-robots/irb-460>
6. Довідкова інформація MeArm [Електронний ресурс]. – Режим доступу : <https://www.mearm.com>
7. Довідкова інформація Arduino [Електронний ресурс]. – Режим доступу : <https://www.arduino.cc>
8. **Гололобов В.Н.** С чего начинаются роботы. О проекте Arduino для школьников и не только [Текст] / В.Н. Голобок. - Москва: 2011.— 189 с.
9. Методичні вказівки до дипломного проектування розділу «Охорона праці» [Текст] / Укл.: Г.І. Дуднік, В.П. Порохненко, А.А. Потуремець, А.О. Писарський, О.В. Коваленко, О.М. Савчук. – Запоріжжя: ЗНТУ, 2000. – 60 с.
10. **Коберник, В. Ф.** Охрана труда [Текст] / В.Ф. Коберник; – К.: Вища школа, 1990. – 286 с.
11. **Депутат О.П.** та ін. Цивільна оборона. Підручник [Текст] / О.П.Депутат, І.В.Коваленко, І.С.Мужик; За ред. полковника П. І. Кашина. - Львів: ІП «Василькевич К.І.», 2005. - 340 с.

12. Методичні вказівки до економічного обґрунтування дипломних проектів для студентів спеціальностей 7.080203 «Системний аналіз та управління» [Текст] / Укл.: Р.Г Бобровникова, Т.М.Тіховська, О.Б. Борисенко, Н.Л. Онуфрієнко. – Запоріжжя: ЗНТУ, 2011. – 20 с.

Додаток А Код програми на Android

```

<xml xmlns="http://www.w3.org/1999/xhtml" > <block type="component_event" id="9"
x="728" y="3695"> <mutation component_type="Button" instance_name="Button3"
event_name="Click"></mutation> <field name="COMPONENT_SELECTOR">Button3</field>
<statement name="DO"> <block type="component_method" id="10"> <mutation
component_type="SpeechRecognizer" method_name="GetText" is_generic="false"
instance_name="РаспознавательРечи1"></mutation> <field
name="COMPONENT_SELECTOR">РаспознавательРечи1</field> </block> </statement> </block>
<block type="component_event" id="1" x="1496" y="3812"> <mutation component_type="ListPicker"
instance_name="ВыборИзСписка1" event_name="BeforePicking"></mutation> <field
name="COMPONENT_SELECTOR">ВыборИзСписка1</field> <statement name="DO"> <block
type="component_set_get" id="2" inline="false"> <mutation component_type="ListPicker"
set_or_get="set" property_name="Elements" is_generic="false"
instance_name="ВыборИзСписка1"></mutation> <field
name="COMPONENT_SELECTOR">ВыборИзСписка1</field> <field name="PROP">Elements</field>
<value name="VALUE"> <block type="component_set_get" id="3"> <mutation
component_type="BluetoothClient" set_or_get="get" property_name="AddressesAndNames"
is_generic="false" instance_name="BluetoothClient1"></mutation> <field
name="COMPONENT_SELECTOR">BluetoothClient1</field> <field
name="PROP">AddressesAndNames</field> </block> </value> </block> </statement>
</block> <block type="component_event" id="4" x="2211" y="3812"> <mutation
component_type="ListPicker" instance_name="ВыборИзСписка1"
event_name="AfterPicking"></mutation> <field
name="COMPONENT_SELECTOR">ВыборИзСписка1</field> <statement name="DO"> <block
type="component_set_get" id="5" inline="false"> <mutation component_type="ListPicker"
set_or_get="set" property_name="Selection" is_generic="false"
instance_name="ВыборИзСписка1"></mutation> <field
name="COMPONENT_SELECTOR">ВыборИзСписка1</field> <field name="PROP">Selection</field>
<value name="VALUE"> <block type="component_method" id="6" inline="false"> <mutation
component_type="BluetoothClient" method_name="Connect" is_generic="false"
instance_name="BluetoothClient1"></mutation> <field
name="COMPONENT_SELECTOR">BluetoothClient1</field> <value name="ARG0"> <block

```

```

type="component_set_get" id="7">      <mutation component_type="ListPicker" set_or_get="get"
property_name="Selection" is_generic="false" instance_name="ВыборИзСписка1"></mutation>
<field name="COMPONENT_SELECTOR">ВыборИзСписка1</field>      <field
name="PROP">Selection</field>      </block>      </value>      </block>      </value>      <next>
<block type="component_set_get" id="8" inline="false">      <mutation component_type="Label"
set_or_get="set" property_name="Visible" is_generic="false" instance_name="Надпись2"></mutation>
<field name="COMPONENT_SELECTOR">Надпись2</field>      <field name="PROP">Visible</field>
<value name="VALUE">      <block type="logic_boolean" id="118">      <field
name="BOOL">TRUE</field>      </block>      </value>      <next>      <block
type="component_set_get" id="111" inline="false">      <mutation component_type="Button"
set_or_get="set" property_name="Visible" is_generic="false" instance_name="Button3"></mutation>
<field name="COMPONENT_SELECTOR">Button3</field>      <field name="PROP">Visible</field>
<value name="VALUE">      <block type="logic_boolean" id="119">      <field
name="BOOL">TRUE</field>      </block>      </value>      </block>      </next>
</block>      </next>      </block>      </statement>      </block>      <block type="component_event"
id="wP=Z*!uCMbXs?y@DA-7s" x="748" y="4021">      <mutation component_type="SpeechRecognizer"
instance_name="РаспознавательРечи1" event_name="AfterGettingText"></mutation>      <field
name="COMPONENT_SELECTOR">РаспознавательРечи1</field>      <statement name="DO">      <block
type="controls_if" id="uB$N^5vL/^K(!H^?x:/3" inline="false">      <value name="IF0">      <block
type="logic_compare" id="k9EtMy/dhu4DJzQs~*Ed">      <field name="OP">EQ</field>      <value
name="A">      <block type="component_set_get" id="D[w5)!Mmvo+K(8!ndLK|">
<mutation component_type="SpeechRecognizer" set_or_get="get" property_name="Result"
is_generic="false" instance_name="РаспознавательРечи1"></mutation>      <field
name="COMPONENT_SELECTOR">РаспознавательРечи1</field>      <field
name="PROP">Result</field>      </block>      </value>      <value name="B">      <block
type="text" id="};E7VOuixu$64[,7e-jX">      <field name="TEXT">основа 15</field>      </block>
</value>      </block>      </value>      <statement name="DO0">      <block
type="component_method" id="uf@YkULuUZcK[:z+%j:2" inline="false">      <mutation
component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>      <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>      <value name="ARG0">      <block
type="text" id="ZHIG$K:hp87KjNS|u^P)">      <field name="TEXT">1</field>      </block>
</value>      </block>      </statement>      <next>      <block type="controls_if"
id="cSP`;7OX)w,7uGgY2uSV" inline="false">      <value name="IF0">      <block
type="logic_compare" id="V:Mc[$hOA+bmsZkxZXqB">      <field name="OP">EQ</field>
<value name="A">      <block type="component_set_get" id="LO;x2XL;~()C}1IWOFwC">

```

```

<mutation component_type="SpeechRecognizer" set_or_get="get" property_name="Result"
is_generic="false" instance_name="РаспознавательРечи1"></mutation>      <field
name="COMPONENT_SELECTOR">РаспознавательРечи1</field>      <field
name="PROP">Result</field>      </block>      </value>      <value name="B">
<block type="text" id="+Fc;=8~6JeuW_sAx6-o">      <field name="TEXT">основа 30</field>
</block>      </value>      </block>      </value>      <statement name="DO0">
<block type="component_method" id="bt?v.egHB_0G{!.+U}" inline="false">      <mutation
component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>      <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>      <value name="ARG0">
<block type="text" id="$NHC8y,iFh72KhcdyLR">      <field name="TEXT">2</field>
</block>      </value>      </block>      </statement>      <next>      <block
type="controls_if" id="mc1!wQ{P!|ZNfqUmN3S;" inline="false">      <value name="IF0">
<block type="logic_compare" id="..TomFt@,N)B9RABb$V,">      <field name="OP">EQ</field>
<value name="A">      <block type="component_set_get" id="XQWTIKbe(;sRSm$?Md;">
<mutation component_type="SpeechRecognizer" set_or_get="get" property_name="Result"
is_generic="false" instance_name="РаспознавательРечи1"></mutation>      <field
name="COMPONENT_SELECTOR">РаспознавательРечи1</field>      <field
name="PROP">Result</field>      </block>      </value>      <value name="B">
<block type="text" id="!+:-fM1ND[0YHXg+yk2f">      <field name="TEXT">основа 45</field>
</block>      </value>      </block>      </value>      <statement name="DO0">
<block type="component_method" id="(rnQ]+i%26tWJjdMk{kF" inline="false">      <mutation
component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>      <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>      <value name="ARG0">
<block type="text" id="m~ooMjps@o4tOgtPf|XK">      <field name="TEXT">3</field>
</block>      </value>      </block>      </statement>      <next>      <block
type="controls_if" id="{yKHo!DFgG%4V@cDHb$7" inline="false">      <value name="IF0">
<block type="logic_compare" id="!*Xd+!cDqw7NxaDWaD1#">      <field
name="OP">EQ</field>      <value name="A">      <block
type="component_set_get" id="3]:RxZi7s(V5(iY-S6v">      <mutation
component_type="SpeechRecognizer" set_or_get="get" property_name="Result" is_generic="false"
instance_name="РаспознавательРечи1"></mutation>      <field
name="COMPONENT_SELECTOR">РаспознавательРечи1</field>      <field
name="PROP">Result</field>      </block>      </value>      <value
name="B">      <block type="text" id=";93KDqWe-}NX:Jq5.etL">      <field

```

```

name="TEXT">основа 60</field>                </block>                </value>                </block>
</value>                <statement name="DO0">                <block type="component_method"
id="b~_dv~G|U18b2uj0G*xQ" inline="false">                <mutation
component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>                <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>                <value name="ARG0">
<block type="text" id="/Dwp4c^DPXHuhhdUITt:">                <field name="TEXT">4</field>
</block>                </value>                </block>                </statement>                <next>
<block type="controls_if" id="VSO?|Ck5q9kyuSb@4.IS" inline="false">                <value name="IF0">
<block type="logic_compare" id="*v!pM(6U^Wb?Q9%!B?4q">                <field
name="OP">EQ</field>                <value name="A">                <block
type="component_set_get" id=",e$,@3M$NZ|j|OOMc=MB">                <mutation
component_type="SpeechRecognizer" set_or_get="get" property_name="Result" is_generic="false"
instance_name="РаспознавательРечи1"></mutation>                <field
name="COMPONENT_SELECTOR">РаспознавательРечи1</field>                <field
name="PROP">Result</field>                </block>                </value>                <value
name="B">                <block type="text" id="apVMKsrBvq*+CRGF;z0}">                <field
name="TEXT">основа 75</field>                </block>                </value>
</block>                </value>                <statement name="DO0">                <block
type="component_method" id="[:4]:R`w/iD6[bcyao+" inline="false">                <mutation
component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>                <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>                <value name="ARG0">
<block type="text" id="Z.,xN$OvPaFzv+Gjs7V8">                <field name="TEXT">5</field>
</block>                </value>                </block>                </statement>                <next>
<block type="controls_if" id="O3L|3#Z^e+SLV3$vog3D" inline="false">                <value
name="IF0">                <block type="logic_compare" id="$OWo`hdne74^KmicNNC#">
<field name="OP">EQ</field>                <value name="A">                <block
type="component_set_get" id="NpY~iilTOV{gt9Ecl}#T">                <mutation
component_type="SpeechRecognizer" set_or_get="get" property_name="Result" is_generic="false"
instance_name="РаспознавательРечи1"></mutation>                <field
name="COMPONENT_SELECTOR">РаспознавательРечи1</field>                <field
name="PROP">Result</field>                </block>                </value>
<value name="B">                <block type="text" id="Uizn8:${%*`4a/k7}x?i">
<field name="TEXT">основа 90</field>                </block>                </value>
</block>                </value>                <statement name="DO0">                <block

```

```

type="component_method" id="PG0%-/FzwjD;Zh6JK,y_" inline="false">                                <mutation
component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>                                <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>                                <value name="ARG0">
<block type="text" id="C]hwM{iNspa1h%w$U)lI">                                <field name="TEXT">6</field>
</block>                                </value>                                </block>                                </statement>
<next>                                <block type="controls_if" id="QRN}^|Ok7_8@v]*dTQ0$" inline="false">
<value name="IF0">                                <block type="logic_compare" id="({_2AE!do6lfpO-Ahy|:.">
<field name="OP">EQ</field>                                <value name="A">                                <block
type="component_set_get" id="~y7s^OwL3DHR3,:Lh2=v">                                <mutation
component_type="SpeechRecognizer" set_or_get="get" property_name="Result" is_generic="false"
instance_name="РаспознавательРечи1"></mutation>                                <field
name="COMPONENT_SELECTOR">РаспознавательРечи1</field>                                <field
name="PROP">Result</field>                                </block>                                </value>
<value name="B">                                <block type="text" id="L$(jUfSFJ.GwO$YZ+_ua">
<field name="TEXT">основа 105</field>                                </block>                                </value>
</block>                                </value>                                <statement name="DO0">
<block type="component_method" id="jq1+lrvb?[ybxu*n;y(w" inline="false">
<mutation component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>                                <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>                                <value name="ARG0">
<block type="text" id="-N+bu3ac9YnYR%$xLb@N">                                <field name="TEXT">7</field>
</block>                                </value>                                </block>                                </statement>
<next>                                <block type="controls_if" id="qG6w9~M[]OcuO?lzc;" inline="false">
<value name="IF0">                                <block type="logic_compare" id="4|p0PWx*}Hp9g2ws6pZQ">
<field name="OP">EQ</field>                                <value name="A">                                <block
type="component_set_get" id="|!-tmFN@.cv:=K-[l@y-">                                <mutation
component_type="SpeechRecognizer" set_or_get="get" property_name="Result" is_generic="false"
instance_name="РаспознавательРечи1"></mutation>                                <field
name="COMPONENT_SELECTOR">РаспознавательРечи1</field>                                <field
name="PROP">Result</field>                                </block>                                </value>
<value name="B">                                <block type="text" id="QpKzF`^8bldNH1y//BSC">
<field name="TEXT">основа 120</field>                                </block>                                </value>
</block>                                </value>                                <statement name="DO0">
<block type="component_method" id="8KoXdNYVx*ET}lLb}2|K" inline="false">
<mutation component_type="BluetoothClient" method_name="SendText" is_generic="false"

```

```

instance_name="BluetoothClient1"></mutation>                                <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>                                <value
name="ARGO">                                <block type="text" id="Jqa*3q|PNHld*%F%Ty+!">
<field name="TEXT">8</field>                                </block>                                </value>
</block>                                </statement>                                <next>                                <block
type="controls_if" id="tw:MF?-SFu4=gSv-1iie" inline="false">                                <value name="IF0">
<block type="logic_compare" id="[%z|--)t%{_KbgtGfDgp">                                <field
name="OP">EQ</field>                                <value name="A">                                <block
type="component_set_get" id="0`Q[6qU3j:kv#yM0FRAU">                                <mutation
component_type="SpeechRecognizer" set_or_get="get" property_name="Result" is_generic="false"
instance_name="РаспознавательРечи1"></mutation>                                <field
name="COMPONENT_SELECTOR">РаспознавательРечи1</field>                                <field
name="PROP">Result</field>                                </block>                                </value>
<value name="B">                                <block type="text" id="F/gBJRExFFri82cV=FfU">
<field name="TEXT">Открой захват</field>                                </block>
</value>                                </block>                                </value>
<statement name="DO0">                                <block type="component_method" id="(N_-
,8oD[.vWuLSu)еQq" inline="false">                                <mutation
component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>                                <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>                                <value
name="ARGO">                                <block type="text" id="P1Aw_u,9;;K3s),ukzss">
<field name="TEXT">9</field>                                </block>                                </value>
</block>                                </statement>                                <next>                                <block
type="controls_if" id="A-zJu#*tSlx;r~Qlh}S{" inline="false">                                <value name="IF0">
<block type="logic_compare" id="]5f#~/+yC01r[!jU2mC">                                <field
name="OP">EQ</field>                                <value name="A">                                <block
type="component_set_get" id="R5+8DUX]6`nog:Ah)0|!">                                <mutation
component_type="SpeechRecognizer" set_or_get="get" property_name="Result" is_generic="false"
instance_name="РаспознавательРечи1"></mutation>                                <field
name="COMPONENT_SELECTOR">РаспознавательРечи1</field>                                <field
name="PROP">Result</field>                                </block>                                </value>
<value name="B">                                <block type="text" id="CEsx_b(qz-liBI0oA(AQ">
<field name="TEXT">захват 90</field>                                </block>
</value>                                </block>                                </value>
<statement name="DO0">                                <block type="component_method" id="04*Q5dg-

```



```

0</field>                                </block>                                </value>
</block>                                </value>                                <statement name="DO0">
<block type="component_method" id="U/^[hU_jQ2KS@N27^I%A" inline="false">
<mutation component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>                                <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>                                <value
name="ARG0">                                <block type="text" id="m+:%iH6o2s=Sc36C^C0,">
<field name="TEXT">a</field>                                </block>
</value>                                </block>                                </statement>
<next>                                <block type="controls_if" id="I_~.lv]YnXXiY)JeNT?a" inline="false">
<value name="IF0">                                <block type="logic_compare"
id="!x|~}B[G0)D,;,^!xbv">                                <field name="OP">EQ</field>
<value name="A">                                <block type="component_set_get"
id="*o~~6lInn4XEX`y,-F{gE">                                <mutation
component_type="SpeechRecognizer" set_or_get="get" property_name="Result" is_generic="false"
instance_name="РаспознавательРечи1"></mutation>                                <field
name="COMPONENT_SELECTOR">РаспознавательРечи1</field>
<field name="PROP">Result</field>                                </block>
</value>                                <value name="B">                                <block
type="text" id="XGu}SxeOf`P*e=UaINbj">                                <field name="TEXT">левый
15</field>                                </block>                                </value>
</block>                                </value>                                <statement
name="DO0">                                <block type="component_method"
id="BH[j9rQjlz]BH2*n]B(|" inline="false">                                <mutation
component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>                                <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>                                <value
name="ARG0">                                <block type="text" id="@s-/(yOwu,5eu?{P3~@q">
<field name="TEXT">b</field>                                </block>
</value>                                </block>                                </statement>
<next>                                <block type="controls_if" id="oeh7Oc-IRp~]rnX5jmHQ"
inline="false">                                <value name="IF0">
<block type="logic_compare" id="Its/e6YtQtxsJnlybOYn">                                <field
name="OP">EQ</field>                                <value name="A">
<block type="component_set_get" id="bbE/E1@],tuGeyP2NWx/">
<mutation component_type="SpeechRecognizer" set_or_get="get" property_name="Result"

```

```

is_generic="false" instance_name="РаспознавательРечи1"></mutation>
<field name="COMPONENT_SELECTOR">РаспознавательРечи1</field>
<field name="PROP">Result</field> </block>
</value> <value name="B">
<block type="text" id="CX~HQYs{PmCktgh*}61:"> <field
name="TEXT">левый 30</field> </block>
</value> </block> </value>
<statement name="DO0"> <block type="component_method"
id="O4SIFBcyf[Lm0*K$/;5c" inline="false"> <mutation
component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation> <field
name="COMPONENT_SELECTOR">BluetoothClient1</field> <value
name="ARG0"> <block type="text" id="*6NiA^jx5U*ul`OR}4,w">
<field name="TEXT">c</field> </block>
</value> </block> </statement>
<next> <block type="controls_if" id="Eca}6~vRSZumL2br?!%b"
inline="false"> <value name="IF0">
<block type="logic_compare" id="lkXoxlpfTdrchqmWWw?">
<field name="OP">EQ</field> <value name="A">
<block type="component_set_get" id="d.Z;=e/r/^itt:rF1JZC">
<mutation component_type="SpeechRecognizer" set_or_get="get" property_name="Result"
is_generic="false" instance_name="РаспознавательРечи1"></mutation>
<field name="COMPONENT_SELECTOR">РаспознавательРечи1</field>
<field name="PROP">Result</field> </block>
</value> <value name="B">
<block type="text" id="q:l?mM`x~.m{9u25sKT3"> <field
name="TEXT">левый 45</field> </block>
</value> </block> </value>
<statement name="DO0"> <block type="component_method"
id="_6DY)lt4gtHTLInkEo1-" inline="false"> <mutation
component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation> <field
name="COMPONENT_SELECTOR">BluetoothClient1</field> <value
name="ARG0"> <block type="text"
id="lly=GbB1y=}D1Osn19MH"> <field name="TEXT">d</field>
</block> </value> </block>

```



```

</value>                                <statement name="DO0">
<block type="component_method" id="tr[^xF0]902{DZ]+yC9[" inline="false">
<mutation component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>                                <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>
<value name="ARG0">                                <block type="text"
id="bN#J*Y%`%`%{5w8BIG}6">                                <field
name="TEXT">f</field>                                </block>
</value>                                </block>
</statement>                                <next>
<block type="controls_if" id="aGkieJJ!=!~)urFN_Vyd" inline="false">
<value name="IF0">                                <block type="logic_compare"
id="R[dru+cSsylezP4]=8;1">                                <field name="OP">EQ</field>
<value name="A">                                <block type="component_set_get"
id="1{ib_fdY*d9_WL2H-D4?}">                                <mutation
component_type="SpeechRecognizer" set_or_get="get" property_name="Result" is_generic="false"
instance_name="РаспознавательРечи1"></mutation>
<field name="COMPONENT_SELECTOR">РаспознавательРечи1</field>
<field name="PROP">Result</field>                                </block>
</value>                                <value name="B">
<block type="text" id="nNd;3C{s;5H,sTPGq;}~">                                <field
name="TEXT">левый 90</field>                                </block>
</value>                                </block>
</value>                                <statement name="DO0">
<block type="component_method" id="[/bQBo[ta)W{=|5!sQeH" inline="false">
<mutation component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>                                <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>
<value name="ARG0">                                <block type="text"
id=":Bxsa6;o+$GzR@}fyQmF">                                <field
name="TEXT">g</field>                                </block>
</value>                                </block>
</statement>                                <next>
<block type="controls_if" id="CO]x.5ie{IHIPA/IE(#@" inline="false">
<value name="IF0">                                <block type="logic_compare"
id="#l6R-;cls,%8/[h]l5Ss">                                <field name="OP">EQ</field>

```

```

<value name="A">                                     <block type="component_set_get"
id="Mx1rYaLKEZljW,A[k+xk">                             <mutation
component_type="SpeechRecognizer" set_or_get="get" property_name="Result" is_generic="false"
instance_name="РаспознавательРечи1"></mutation>
<field name="COMPONENT_SELECTOR">РаспознавательРечи1</field>
<field name="PROP">Result</field>                             </block>
</value>                                             <value name="B">
<block type="text" id="I4,:JUBK{2:lv`nFdKI5">                 <field
name="TEXT">правый 15</field>                             </block>
</value>                                             </block>
</value>                                             <statement name="DO0">
<block type="component_method" id="!)V_WhZUd|.))s+ojL=^" inline="false">
<mutation component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>                 <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>
<value name="ARG0">                                         <block type="text"
id=",L,RGvcL@x%;BA.SiF.c">                                 <field
name="TEXT">h</field>                                     </block>
</value>                                             </block>
</statement>                                           <next>
<block type="controls_if" id="]tklv3_0Wn-]C8p;m.Gs" inline="false">
<value name="IF0">                                         <block type="logic_compare"
id="V.larUA]b~AxWeDB#Zf_">                               <field
name="OP">EQ</field>                                     <value name="A">
<block type="component_set_get" id="vU*~Cx[9c]K,69?UjkoB">
<mutation component_type="SpeechRecognizer" set_or_get="get" property_name="Result"
is_generic="false" instance_name="РаспознавательРечи1"></mutation>
<field name="COMPONENT_SELECTOR">РаспознавательРечи1</field>
<field name="PROP">Result</field>                             </block>
</value>                                             <value name="B">
<block type="text" id="g}9ns0Wtb}cBX43yRKNS">
<field name="TEXT">правый 30</field>                       </block>
</value>                                             </block>
</value>                                             <statement name="DO0">
<block type="component_method" id="JaqJZEg,?dPM^!93}C/B" inline="false">
<mutation component_type="BluetoothClient" method_name="SendText" is_generic="false"

```

```

instance_name="BluetoothClient1"></mutation> <field>
name="COMPONENT_SELECTOR">BluetoothClient1</field>
<value name="ARG0"> <block type="text"
id="y~x3#m,[X_]U=GC_PQ"> <field
name="TEXT">i</field> </block>
</value> </block>
</statement> <next>
<block type="controls_if" id="a,*e8]/,r`2gb_);2M6?" inline="false">
<value name="IF0"> <block type="logic_compare"
id="[/QCut~7#Y/MTI0t]N]7"> <field
name="OP">EQ</field> <value name="A">
<block type="component_set_get" id="[0{Bexa=WCg1u;oc^Tn6">
<mutation component_type="SpeechRecognizer" set_or_get="get" property_name="Result"
is_generic="false" instance_name="РаспознавательРечи1"></mutation>
<field name="COMPONENT_SELECTOR">РаспознавательРечи1</field>
<field name="PROP">Result</field> </block>
</value> <value name="B">
<block type="text" id="e#6lqw(lcmCm]ATH=,4D">
<field name="TEXT">правый 45</field> </block>
</value> </block>
</value> <statement name="DO0">
<block type="component_method" id="/l6K%d/uUE)ZA#VDvl%$" inline="false">
<mutation component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>
<field name="COMPONENT_SELECTOR">BluetoothClient1</field>
<value name="ARG0"> <block type="text"
id="/N7BVjFtnYGx8;UqF=*3"> <field
name="TEXT">j</field> </block>
</value> </block>
</statement> <next>
<block type="controls_if" id="EZ~dZ%7tWj#hQW.gEFz" inline="false">
<value name="IF0"> <block type="logic_compare"
id="}ahOm-#T6}R4zG1qA*LS"> <field
name="OP">EQ</field> <value name="A">
<block type="component_set_get" id="/4EjsvID}u69tE?M~Ukw">
<mutation component_type="SpeechRecognizer" set_or_get="get" property_name="Result"

```

```

is_generic="false" instance_name="РаспознавательРечи1"></mutation>
<field name="COMPONENT_SELECTOR">РаспознавательРечи1</field>
<field name="PROP">Result</field>                                     </block>
</value>                                                             <value name="B">
<block type="text" id="=A)ol!t.t6%rKka$Fy2q">
<field name="TEXT">правый 60</field>                                     </block>
</value>                                                             </block>
</value>                                                             <statement name="DOO">
<block type="component_method" id="CH;~@bs=(~`{Fz}^u}M5" inline="false">
<mutation component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>
<field name="COMPONENT_SELECTOR">BluetoothClient1</field>
<value name="ARG0">                                                 <block type="text"
id="Wwc*eQ9^s.cR=k8U98Gs">                                         <field
name="TEXT">k</field>                                               </block>
</value>                                                             </block>
</statement>                                                         <next>
<block type="controls_if" id="_MG~*B^pt#AG*YVcf!xL" inline="false">
<value name="IF0">                                                 <block type="logic_compare"
id="@ED)F.{N`z5fhz.77#/S">                                         <field
name="OP">EQ</field>                                               <value name="A">
<block type="component_set_get" id="0A/NsE}Vt#rt_J68k]7f">
<mutation component_type="SpeechRecognizer" set_or_get="get" property_name="Result"
is_generic="false" instance_name="РаспознавательРечи1"></mutation>
<field name="COMPONENT_SELECTOR">РаспознавательРечи1</field>
<field name="PROP">Result</field>                                     </block>
</value>                                                             <value name="B">
<block type="text" id="6za!d7`_Jvx{!8LdLp!1">
<field name="TEXT">правый 75</field>
</block>                                                             </value>
</block>                                                             </value>
<statement name="DOO">                                             <block
type="component_method" id="0pJ%#4fghLc{m~NGlr[k" inline="false">
<mutation component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>
<field name="COMPONENT_SELECTOR">BluetoothClient1</field>

```



```

<value name="ARG0">
id="5Nv}+=2C;q9=Ey,za!{}">
name="TEXT">|</field>
</value>
</statement>
<block type="controls_if" id="r)=ZmlsN5TYqB1m|:Hja" inline="false">
<value name="IF0">
type="logic_compare" id="Fi$J[S;co%p^X8Yp@Ssl">
<field name="OP">EQ</field>
name="A">
type="component_set_get" id="80BYA=)PT%kCuzPsUrAN">
<mutation component_type="SpeechRecognizer" set_or_get="get" property_name="Result"
is_generic="false" instance_name="РаспознавательРечи1"></mutation>
<field name="COMPONENT_SELECTOR">РаспознавательРечи1</field>
<field name="PROP">Result</field>
</value>
<block type="text" id="KIDcxq~ie84~lphKjJ9b">
<field name="TEXT">правый 90</field>
</block>
<block type="statement" name="DO0">
type="component_method" id="|c!_7%5cY[:a?jG!wRm" inline="false">
<mutation component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>
<field name="COMPONENT_SELECTOR">BluetoothClient1</field>
<value name="ARG0">
id="Wrs13P|(1leHv~8`c|ml">
name="TEXT">m</field>
</value>
</statement>
<block type="controls_if" id="Ol;XH%`CClywJz`_M@hH" inline="false">
<value name="IF0">
type="logic_compare" id="):6g@vljPVT/MdqA|I?,">
<field name="OP">EQ</field>
name="A">
type="component_set_get" id="rS5=86ELgk=of~d_i=0R">

```

```

<mutation component_type="SpeechRecognizer" set_or_get="get" property_name="Result"
is_generic="false" instance_name="РаспознавательРечи1"></mutation>
<field name="COMPONENT_SELECTOR">РаспознавательРечи1</field>
<field name="PROP">Result</field>
</block>                                     </value>
<value name="B">                               <block type="text"
id="[=/6?t%5BNXbH]l338X1">                       <field
name="TEXT">правый 105</field>
</block>                                     </value>
</block>                                     </value>
<statement name="DO0">                           <block
type="component_method" id="090od4+apu#19y:#:%Ed" inline="false">
<mutation component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>
<field name="COMPONENT_SELECTOR">BluetoothClient1</field>
<value name="ARG0">                               <block type="text"
id="bl:Es#G{IX:V,L~z$a6G">                       <field
name="TEXT">n</field>                               </block>
</value>                                     </block>
</statement>                                   <next>
<block type="controls_if" id=".F#b#9vvsO,Bg8;DoAq0" inline="false">
<value name="IF0">                               <block
type="logic_compare" id="t)KNFlx7.nu@L8ShbQ5`">
<field name="OP">EQ</field>                       <value
name="A">                                       <block
type="component_set_get" id="/#qm}9[*3uTP|@3B{Wb5`">
<mutation component_type="SpeechRecognizer" set_or_get="get" property_name="Result"
is_generic="false" instance_name="РаспознавательРечи1"></mutation>
<field name="COMPONENT_SELECTOR">РаспознавательРечи1</field>
<field name="PROP">Result</field>
</block>                                     </value>
<value name="B">                               <block type="text"
id="TUu5Kls5dJZsy/-3f,vc">                       <field
name="TEXT">правый 120</field>
</block>                                     </value>
</block>                                     </value>

```



```

</next>                </block>                </next>                </block>
</next>                </block>                </next>                </block>
</next>                </block>                </next>                </block>                </next>
</block>                </next>                </block>                </next>                </block>                </next>                </block>
</statement> </block> <block type="component_event" id=".F05}Rc9{5!EFIfNWy!y" x="1396"
y="4183"> <mutation component_type="Button" instance_name="Button5"
event_name="Click"></mutation> <field name="COMPONENT_SELECTOR">Button5</field>
<statement name="DO"> <block type="component_method" id="r0Z{RdH.YQ6q5L}2Bw[=">
<mutation component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation> <field
name="COMPONENT_SELECTOR">BluetoothClient1</field> <value name="ARG0"> <block
type="text" id=";v~HxoQq^(nWye[SnGkw"> <field name="TEXT">2</field> </block>
</value> </block> </statement> </block> <block type="component_event"
id="c_=NeC+BMK~!]T?Rr-tO" x="1421" y="4780"> <mutation component_type="Button"
instance_name="Button6" event_name="Click"></mutation> <field
name="COMPONENT_SELECTOR">Button6</field> <statement name="DO"> <block
type="component_method" id="H#JF?!][8,P]M4A#8snC"> <mutation
component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation> <field
name="COMPONENT_SELECTOR">BluetoothClient1</field> <value name="ARG0"> <block
type="text" id="Gu=.sx^5_q)U1BB/p}Y;"> <field name="TEXT">8</field> </block>
</value> </block> </statement> </block> <block type="component_event"
id="XV!LLbT}dat+TK.io10d" x="1428" y="4988"> <mutation component_type="Button"
instance_name="Button10" event_name="Click"></mutation> <field
name="COMPONENT_SELECTOR">Button10</field> <statement name="DO"> <block
type="component_method" id="p]](EqWXXjc;A]3C,dmL"> <mutation
component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation> <field
name="COMPONENT_SELECTOR">BluetoothClient1</field> <value name="ARG0"> <block
type="text" id="w7Hst~iK$Se;jzLF(r:F"> <field name="TEXT">9</field> </block> </value>
</block> </statement> </block> <block type="component_event" id="WDAFu2zN_/iUAep7{W)A"
x="1442" y="5185"> <mutation component_type="Button" instance_name="Button11"
event_name="Click"></mutation> <field name="COMPONENT_SELECTOR">Button11</field>
<statement name="DO"> <block type="component_method" id="x!iV]dFdT+9wuPkHh?P6">
<mutation component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation> <field

```

```

name="COMPONENT_SELECTOR">BluetoothClient1</field>    <value name="ARG0">    <block
type="text" id="@fZQ-f2QUU~~jKw+y`(m">    <field name="TEXT">a</field>    </block>
</value>    </block>    </statement>    </block>    <block type="component_event"
id="uc5@+6A#CKnV~5Hu!gr," x="1447" y="5386">    <mutation component_type="Button"
instance_name="Button7" event_name="Click"></mutation>    <field
name="COMPONENT_SELECTOR">Button7</field>    <statement name="DO">    <block
type="component_method" id="oFx3n%+btW69$VS1OM*z">    <mutation
component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>    <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>    <value name="ARG0">    <block
type="text" id="l~!O=Y03X(4m%/v`_A.B">    <field name="TEXT">c</field>    </block>
</value>    </block>    </statement>    </block>    <block type="component_event"
id=";t^0enCl=RluWu2}dnkr" x="1467" y="5800">    <mutation component_type="Button"
instance_name="Button4" event_name="Click"></mutation>    <field
name="COMPONENT_SELECTOR">Button4</field>    <statement name="DO">    <block
type="component_method" id="WN2!Mk8_cU0*RnTt5F:R">    <mutation
component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>    <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>    <value name="ARG0">    <block
type="text" id="iSs,AigQ@A9Y$IQ2CKXk">    <field name="TEXT">g</field>    </block>
</value>    </block>    </statement>    </block>    <block type="component_event"
id="u7D79HWFm=k+h0gfiKA?" x="1459" y="5994">    <mutation component_type="Button"
instance_name="Button8" event_name="Click"></mutation>    <field
name="COMPONENT_SELECTOR">Button8</field>    <statement name="DO">    <block
type="component_method" id="q0wEYbWpa=kAXy[0[85=">    <mutation
component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>    <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>    <value name="ARG0">    <block
type="text" id="rw(SnQ`JQsj!sK,F8f.J">    <field name="TEXT">i</field>    </block>    </value>
</block>    </statement>    </block>    <block type="component_event" id="8}cF;*B%/|v^+bJ%d{;-
x="1456" y="6397">    <mutation component_type="Button" instance_name="Button9"
event_name="Click"></mutation>    <field name="COMPONENT_SELECTOR">Button9</field>
<statement name="DO">    <block type="component_method" id="K=Z78;cWM(uGqp{fg%16">
<mutation component_type="BluetoothClient" method_name="SendText" is_generic="false"
instance_name="BluetoothClient1"></mutation>    <field
name="COMPONENT_SELECTOR">BluetoothClient1</field>    <value name="ARG0">    <block

```

```
type="text" id="-hmjd=2/gIYMkSCoC~d?>    <field name="TEXT">m</field>    </block>
</value>    </block>    </statement>    </block> <yacodeblocks ya-version="159" language-
version="20"></yacodeblocks></xml>
</xml>
```

Додаток Б Код програми Arduino

```

#include <Servo.h>
#include <SPI.h>
#define XPOS 0
#define YPOS 1
char incomingByte; // входящие данные
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
int servoPin1 = 5;
int servoPin2 = 9;
int servoPin3 = 10;
int servoPin4 = 11;
const int SERVOS = 4;
const int ACC = 10; // the accurancy of the potentiometer value before idle
starts counting
int PIN[SERVOS], value[SERVOS], idle[SERVOS],
currentAngle[SERVOS], MIN[SERVOS], MAX[SERVOS],
INITANGLE[SERVOS], previousAngle[SERVOS],ANA[SERVOS];
Servo myservo[SERVOS];
void setup() {
  Serial.begin(9600);
  //Middle Servo
  PIN[0] = 10;
  MIN[0] = 40;
  MAX[0] = 150;

```

```

INITANGLE[0] = 90;
ANA[0] = 1;
//Left Side
PIN[1] = 9;
MIN[1] = 55;
MAX[1] = 140;
INITANGLE[1] = 60;
ANA[1] = 0;
//Right Side
PIN[2] = 11;
MIN[2] = 50;
MAX[2] = 160;
INITANGLE[2] = 45;
ANA[2] = 2;
//Claw Servo
PIN[3] = 5;
MIN[3] = 0;
MAX[3] = 179;
INITANGLE[3] = 25;
ANA[3] = 3;
  for (int i = 0; i < SERVOS; i++){
    myservo[i].attach(PIN[i]);
    myservo[i].write(INITANGLE[i]);
    value[i] = 0;
    idle[i] = 0;
    previousAngle[i]=INITANGLE[i];
  }
void loop() {
  delay(10); // Delay function to slow movement of arm down
// if (mode == LOW){

```



```

for (int i = 0; i < SERVOS; i++){
  value[i] = analogRead(ANA[i]);
  currentAngle[i] = myservo[i].read();

  if (value[i] > 612) {
    idle[i] = 0;

    if (currentAngle[i] < MAX[i]) ++currentAngle[i];
    if (!myservo[i].attached()){
      myservo[i].attach(PIN[i]);
    }
    myservo[i].write(currentAngle[i]);
  }

  else if (Serial.available() > 0) {
    incomingByte = Serial.read();
    if (incomingByte == '1'){ // midl
servo3.attach(servoPin3);
servo3.write(135);
delay(1000);
servo3.detach();}
    if (incomingByte == '2'){
servo3.attach(servoPin3);
servo3.write(45);
delay(1000);
servo3.detach();}
    if (incomingByte == '3'){
servo3.attach(servoPin3);
servo3.write(90);

```

```
delay(1000);
servo3.detach();}
    if (incomingByte == '4'){ // left
servo2.attach(servoPin2);
servo2.write(135);
delay(1000);
servo2.detach();}
    if (incomingByte == '5'){
servo2.attach(servoPin2);
servo2.write(45);
delay(1000);
servo2.detach();}
    if (incomingByte == '6'){
servo2.attach(servoPin2);
servo2.write(90);
delay(1000);
servo2.detach();}
    if (incomingByte == '7'){ // right
servo4.attach(servoPin4);
servo4.write(135);
delay(1000);
servo4.detach();}
    if (incomingByte == '8'){
servo4.attach(servoPin4);
servo4.write(45);
delay(1000);
servo4.detach();}
    if (incomingByte == '9'){
servo4.attach(servoPin4);
servo4.write(90);
```

```

delay(1000);
servo4.detach();}
    if (incomingByte == 'a'){ // claw
servo1.attach(servoPin1);
servo1.write(135);
delay(1000);
servo1.detach();}
    if (incomingByte == 'b'){
servo1.attach(servoPin1);
servo1.write(45);
delay(1000);
servo1.detach();}
    if (incomingByte == 'c'){
servo1.attach(servoPin1);
servo1.write(90);
delay(1000);
servo1.detach();}
    if (incomingByte == 'd'){
servo1.attach(servoPin1);
servo1.write(0);
delay(1000);
servo1.detach();}

}

else if (value[i] < 412) {
idle[i] = 0;
if (currentAngle[i] > MIN[i]) --currentAngle[i];
if (!myservo[i].attached()){

```

```
    myservo[i].attach(PIN[i]);  
  }  
  myservo[i].write(currentAngle[i]);  
}  
  
else {  
  ++idle[i];  
}  
if (idle[i] > 100){  
  myservo[i].detach();  
  idle[i] = 0;  
}  
}
```