

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Радіотехнічний факультет

(повна назва інституту/факультету)

Кафедра радіоконструювання та виробництва радіоапаратури

(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

Е.А. Нелін
(підпис) (ініціали, прізвище)

« 13 » 06 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

За напрямом підготовки 6.050902 Радіoeлектронні апарати

(код та назва спеціальності)

на тему: Робот - маніпулятор з голосовим керуванням

Виконав (-ла): студент (-ка) IV курсу, групи РК-51
(шифр групи)

Чабай Дмитро Вячеславович
(прізвище, ім'я, по батькові)

Керівник старший викладач, к.т.н., Навроцький Д.О.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

Консультант з охорони праці доцент, к.т.н., Каштанов С.Ф.
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

Рецензент асистент, МIRONCHUK O.O.
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць ін-
ших авторів без відповідних посилань.

Студент Чабай Дмитро
(підпис)

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»

Факультет (інститут) радіотехнічний
(повна назва)

Кафедра радіоконструювання та виробництва радіоапаратури
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

За напрямом підготовки 6.050902 Радіoeлектронні апарати
(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

С.А. Нелін
(підпис) (ініціали, прізвище)

«10» травня 2019 р.

ЗАВДАННЯ

на дипломний проект (роботу) студенту

Чубаю Дмитру Вячеславовичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Робот-маніпулятор з голосовим керуванням

керівник проекту (роботи) Навроцький Дмитро Олександрович ст. викл. к.т.н.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «27» травня 2019 р. № 1399-С

2. Строк подання студентом проекту (роботи) 14.06.2019р.

3. Вихідні дані до проекту (роботи) Напрямок шийовий 220В, голосове керування, керування за допомогою розрахованого програмного забезпечення.

4. Зміст (дипломної роботи) розрахунково-пояснювальної записки (перелік завдань, які потрібно розробити) Вступ: огляд існуючих рішень, вступ до та аналіз технічного рішення; опис конструкції та вибір схематичного рішення; розрахунки

алгоритму, Проектування примісту та аналіз його продуктивності
охорони праці.

5. Перелік (ілюстративного) графічного матеріалу (з точним зазначенням обов'язкових креслеників, плакатів тощо) Структурна схема

6. Консультанти розділів проекту (роботи)*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
3 охорони праці	к.т.н., доцент Каштанов С.Ф.		

7. Дата видачі завдання 15 квітня 2019 року

Календарний план

№ з/п	Назва етапів виконання дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Оцінка існуючих ризиків	13.04. - 16.04.2019р.	виконано
2	Розробка та аналіз технічного завдання	18.04. - 21.04.2019р.	виконано
3	Структурування та вибір схемних рішень	22.04. - 23.04.2019р.	виконано
4	Вибір та аргументування елементів	25.04. - 29.04.2019р.	виконано
5	Розробка проєкційного розроблення	30.04. - 02.05.2019р.	виконано
6	Проектування примісту	03.05. - 06.05.2019р.	виконано
7	Аналіз продуктивності	07.05. - 09.05.2019р.	виконано
8	Охорона праці	10.05. - 14.05.2019р.	виконано
9	Аргументація ризиків та узаг. документів	14.05. - 16.05.2019р.	виконано

Студент


(підпис)

О.В. Чобот
(ініціали, прізвище)

Керівник проекту (роботи)


(підпис)

Д.О. Навроцький
(ініціали, прізвище)

* Консультантом не може бути зазначено керівника дипломного проекту (роботи)

АНОТАЦІЯ

У дипломному проєкті розроблено робот-маніпулятор з голосовим керуванням, який може переміщувати легкі предмети з одного місця на інше.

Метою дипломного проєкта є розробка програмного забезпечення для робота-маніпулятора з голосовим керуванням.

Проводиться аналіз готових рішень, розроблено структурну схему, обрано електронні модулі. Розроблено програмне забезпечення на мовах програмування C/C++, C#, XAML. Проведено перевірку на працездатність програмного забезпечення.

Ключові слова: Arduino Uno, робот-маніпулятор з голосовим керуванням, мікроконтролер, програмне забезпечення.

ANNOTATION

The graduation project presents the elaboration of voice-activated manipulator which can move light objects from one place to another.

The aim of the graduation project is to develop software for the voice-activated manipulator.

The analysis of ready solutions was conducted, a block diagram was developed and electronic modules were selected. A software was elaborated on such programming languages as C/C++, C#, XAML. A software performance check has been performed.

Key words: Arduino Uno, voice-activated manipulator, micro controller unit, software.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
до дипломного проекту**

на тему: Робот-маніпулятор з голосовим керуванням

Київ — 2019 року

ЗМІСТ

Перелік скорочень.....	2
Вступ.....	3
1 Огляд існуючих рішень. Розробка та аналіз технічного завдання.....	4
1.1 Огляд та аналіз аналогів на ринку.....	4
1.2 Аналіз технічного завдання	7
2 Обґрунтування та вибір схемотехнічного рішення.....	13
2.1 Структурна схема.....	13
2.2 Вибір апаратної платформи	14
2.3 Вибір сервоприводів	18
2.4 Вибір блоку живлення.....	23
2.5 Вибір перетворювача напруги	25
3 Розроблення алгоритму	27
3.1 Вибір мови програмування для Arduino	27
3.1.1 Структурна схема для написання програми для Arduino	28
3.1.2 Розробка програми для Arduino.....	28
3.2 Програма для голосового керування.....	32
3.3 Інтерфейс програми голосового керування	42
3.3.1 Розробка програми для інтерфейсу.....	44
4 Проектування приладу та аналіз його працездатності.....	48
4.1 Опис макету	48
4.2 Перевірка працездатності.....	49

					РК51.453321.001 ПЗ			
ЗМ.	Лис	№ докум.	Підпис	Дата				
Розробив	Чабай Д.В.				Робот-маніпулятор з го- лосовим керуванням	Літ.	Лист	Листів
Перевірів	Навроцький Д.О.						1	
Н. Контр.						РК-51 РТФ		
Затведив	Навроцький Д.О.							

5 Охорона праці.....	51
5.1 Визначення основних потенційно шкідливих та небезпечних виробничих факторів	51
5.2 Технічні рішення та організаційні заходи з безпеки і гігієни праці на робочих місцях користувачів ПК	52
5.2.1 Вимоги з охорони праці при роботі з ПК.....	52
5.2.2 Оптимізації параметрів повітряного середовища в робочій зоні	53
5.2.3 Вимоги до робочого місця	54
5.2.4 Організація оптимального режиму праці та відпочинку при використанні ПК.....	55
5.2.5 Виробниче освітлення	56
5.2.6 Виробничий шум.....	56
5.2.7 Електробезпека у виробничих приміщеннях	57
5.2.8 Розрахунок електромережі на вимикаючу здатність	58
5.3 Пожежна безпека та профілактика.....	59
Висновки	61
Перелік джерел посилань	62
Додаток А. Технічне завдання.....	65
Додаток Б. Лістинг програми для Arduino	72
Додаток В. Лістинг програми для голосового керування.....	75
Додаток Г. Лістинг програми для інтерфейсу.....	82

					РК51.453321.001 ПЗ						
ЗМ.	Лис	№ докум.	Підпис	Дата	Робот-маніпулятор з го- лосовим керуванням			Літ.	Лист	Листів	
Розробив	Чабай Д.В.										
Перевірів	Навроцький Д.О.									1	
								РК-51 РТФ			
Н. Контр.											
Затведив	Навроцький Д.О.										

ПЕРЕЛІК СКОРОЧЕНЬ

COM-порт — communications port — двонаправлений послідовний інтерфейс, призначений для обміну байтовою інформацією;

EEPROM — electrically erasable programmable read-only memory — постійний запам'ятовувальний пристрій, один з видів енергонезалежної пам'яті;

GUI — graphical user interface — тип інтерфейсу, який дозволяє користувачам взаємодіяти з електронними пристроями через графічні зображення та візуальні вказівки;

HTML — hypertext markup language — стандартна мова розмітки для створення веб-сторінок і веб-додатків;

IDE — integrated development environment — крос-платформний додаток, що використовується для запису та завантаження програм на Arduino;

SRAM — static random access memory — напівпровідникова оперативна пам'ять;

UART — universal asynchronous receiver-transmitter — асинхронний приймач-передавач, компонентів комп'ютерів;

XAML — eXtensible application markup language — це варіант мови XML від Microsoft, який описує GUI;

МК — мікроконтролер;

ПК — персональний комп'ютер;

ПЗ — програмне забезпечення;

ТЗ — технічне завдання;

ШІМ — широтно-імпульсна модуляція;

					РК51.453321.001 ПЗ	Лист
Зм.	Лис	№ докum.	Підпис	Дата		2

ВСТУП

Одною з невід’ємних частин заводів, великих підприємств є робот. Робот в перекладі з чеської означає «працювати», а робот-маніпулятор це пристрій, який примусово можна обертати в різні боки, змушуючи виконувати ті чи інші операції. Розроблені вони, щоб виконувати дії замість людини, а часом небезпечні операції.

Ручна праця стає все більше не приваблива та не престижна, тому великі підприємства економлять на так званій «робочій силі» та замість малокваліфікованих робітників ставлять роботів. Найбільш розповсюджений вид роботів на підприємствах є робот-маніпулятор.

В даному дипломному проекті розглянуто та зроблено робот-маніпулятор з голосовим керуванням. З’ясовано недоліки та переваги сучасних роботів, проведено їх порівняльний аналіз.

					PK51.453321.001 ПЗ	Лист
						3
Зм.	Лист	№ докум.	Підпис	Дата		

1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ. РОЗРОБКА ТА АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Огляд та аналіз аналогів на ринку

Маніпулятор — це автоматизована машина, що створена за принципом роботи людських кінцівок для виконання однотипних функцій.

На даний час роботи заміняють сотні, а то й тисячі людей. Значне зростання попиту на роботів в виробництві спричинило необхідність їх масовому виробництву, тому на торговому ринку представлено багато роботів-маніпуляторів, але роботів з голосовим керуванням не так же і багато. Приклад роботи маніпулятора показано на рис. 1.1 [1]:



Рисунок 1.1 — Робот-маніпулятор

Найчастіше маніпуляторів використовують на сучасних автоматизованих заводах, де відбувається складання різноманітних механізмів.

Технічні можливості роботів-маніпуляторів дуже великі: вантажопідйомність від 100 грамів до 1 тони, з необмеженою складністю траєкторії та швидкістю рухів. Але в дипломному проекті розглянуто найпростіший варіант робота-маніпулятора з голосовим керуванням, такий робот може стати корисним людям з обмеженими можливостями.

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
4

Сучасні промислові маніпулятори вже мало чим нагадують людську кінцівку, щоб підвищити їх мобільність та універсальність.

В результаті аналізу аналогів на ринку не було знайдено робота-маніпулятора на голосовому керуванні. Тому як аналоги взято роботи-маніпулятори на батарейках CIC 21-535N (рис. 1.2) [2], на гідравліці CIC 21-632 (рис. 1.3) [3] та робот-маніпулятор Ewood Robotics (рис. 1.4) [4].

Особливості CIC 21-535N:

- працює на акумуляторах;
- вантажопідйомність 100г;
- 5 окремих двигунів для управління;
- можливість використання в темноті завдяки влаштованому підсвічуванню;
- джойстик для управління клешнею відкривання/закривання, поворот, обертання;
- збирається швидко та не потребує пайки.



Рисунок 1.2 — Робот CIC 21-535N

Недоліки:

- велика вартість;
- якість матеріалів;

Зм.	Лис	№ докum.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
5

– можливість перепрограмування, та доопрацювання механізмів, двигунів;

– немає голосового керування.

Головною перевагою даного аналога є те що він продається в кожному магазині, та не являється таким складним в зборі.

Особливості СІС 21-632:

– працює без двигунів та акумуляторів;

– вантажопідйомність 50г;

– висока точність виконання команд завдяки системі гальм в рукоятці контролера;

– змога міняти насадки замість клешні присоску;

– збирається швидко та не потребує пайки.



Рисунок 1.3 — Робот СІС 21-632

Головною перевагою так же є доступність в магазинах та простота в зборі, а також екологічність, оскільки виробник запевняє що даний робот є екологічно чистий та не шкодить навколишньому середовищу. Недоліки такі ж як у першого аналога.

Зм.	Лис	№ докum.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
6

Особливості Ewood Robotics:

- працює на батареях або від комп'ютера;
- виготовлений з екологічних матеріалів без використання клею та фарб;
- пристрій може працювати не тільки в ручному режимі а й в автоматичному;

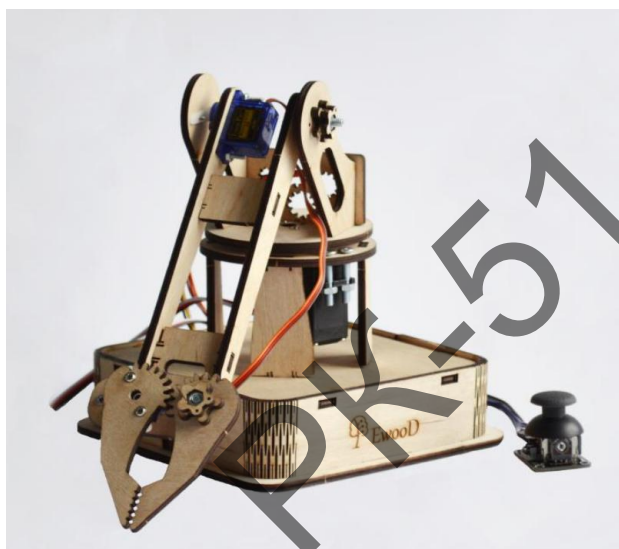


Рисунок 1.4 — Робот-маніпулятор Ewood Robotics

Маніпулятор керується за допомогою платформи Arduino, тому має змогу перепрограмування, на відміну від попередніх аналогів. Головним недоліком даного робота є відсутність голосового керування.

Проаналізувавши аналоги роботів-маніпуляторів, було вирішено зробити робот маніпулятор, але з програмним доопрацюванням порівняно з аналогами.

1.2 Аналіз технічного завдання

У дипломному проекті розробляється робот-маніпулятор з голосовим керуванням. Основною функцією робота є переміщення речей за допомогою голосового керування.

Робот-маніпулятор визначається спеціальним типом апаратури, тому його використання наближено до лабораторних. Зроблений робот з пластику

					РК51.453321.001 ПЗ	Лист
						7
Зм.	Лис	№ доквм.	Підпис	Дата		

надрукованого на 3D принтері, вага даного робота не більше 400 грамів. Робот побудований на базі Arduino, та запрограмований за допомогою програмного забезпечення Visual Studio 2019 на мові C#/C++. Живиться робот від блоку живлення на 12 В.

Проектований робот-маніпулятор буде використовуватись в умовах помірного і холодного клімату. Згідно таких вимог обираємо кліматичне виконання УХЛ 4.2 за ГОСТ 15150-69 [5].

В (табл. 1.1), (табл. 1.2) наведенні параметри, обраним умовам.

Таблиця 1.1 — Температурні вимоги

	Верхнє значення, °С	Нижнє значення, °С
Робочі температури	+35	+10
Граничні температури	+40	+1

Таблиця 1.2 — Вимоги до вологості

Відносна вологість		Абсолютна вологість, г·м ³
Середнє значення	Верхнє значення	
60% при 20°C	80% при 25°C	10

Згідно ГОСТ 16019-2001 [6], умови експлуатації приладу С1: стаціонарна апаратура, що експлуатується в опалювальних та підземних приміщеннях.

Таблиця 1.3 — Вимоги стійкості до механічних і кліматичних факторів

Фактор впливу	Характеристика	Значення фактору для групи апаратури (С1)	Допустимі відхилення норм режимів
1. Синусоїдальна вібрація	Діапазон частот, Гц	10–70	±1

Зм.	Лист	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
8

Таблиця 1.3 (Продовження)

Фактор впливу		Характеристи- ка	Значення фактору для групи апарату- ри (C1)	Допустимі відхилен- ня норм режимів
1. Синусоїдальна вібрація		Амплітуда прискорення, м/с ² (g)	19,6 (2)	±2 (0,2)
		Тривалість дії, хв	90	
2. Механічні удари	При експлуатації	Пікове ударне прискорення, м/с (g)		±20%
		Тривалість удару, м/с (g)		
		К-сть ударів в кожному на- прямку		
	При транспор- туванні	Пікове ударне прискорення, м/с (g)	147 (15)	±20%
		Тривалість удару, м/с (g)	6	
		К-сть ударів в кожному на- прямку	4000	

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист

9

Таблиця 1.3 (Продовження)

Фактор впливу		Характеристика	Значення фактору для групи апаратури (C1)	Допустимі відхилення норм режимів
3. Знижена температура для виконання за ступенем міцності	1	Робоча температура, °C	+5	±3
		Гранична температура, °C	-40	±3
		Час витримки, год	2	
	2	Робоча температура, °C	+5	±3
		Гранична температура, °C	-55	±3
		Час витримки, год	2	
4. Понижений атмосферний тиск		Атмосферний тиск, кПа	55	±10%
		Час витримки в камері, хв	60	
5. Підвищена температура		Робоча температура, °C	+40	±3
		Гранична температура, °C	+55	±3
		Час витримки, год	2	

Зм.	Лист	№ докум.	Підпис	Дата

PK51.453321.001 ПЗ

Лист

10

Робот-маніпулятор в експлуатаційних умовах повинен бути стійкий до зовнішніх чинників, характеристики які наведені у таблиці 1.3.

В даному дипломному проєкті розробляється пристрій, який повинен бути безпечним, простим та зручним у застосуванні і відповідати вимогам безпеки.

До електричних пристроїв висувають вимоги, виконання яких є запорукою безпечної експлуатації виробу. Відповідно до вимог такі пристрої повинні відповідати правилам:

1. У конструкції електротехнічних виробів повинні бути засоби шумозахисту і віброзахисту, що забезпечують рівні шуму і вібрації на робочих місцях відповідно до затверджених санітарних норм. Допустимі значення шумових і вібраційних характеристик електротехнічних виробів повинні бути встановлені в стандартах і технічних умовах, зазначених в ГОСТ 12.1.003-83 та ГОСТ 12.1.012-78;

2. Електрична схема виробу повинна виключати можливість його довільного включення і відключення;

3. Розташування та з'єднання частин виробів повинні бути виконані з урахуванням зручності та безпеки спостереження за виробом при виконанні складальних робіт, проведенні огляду, випробувань та обслуговування;

4. Пожежна безпека повинна забезпечуватися як у нормальному так і в аварійному режимах роботи.

Пристрій який розробляється, повинен бути сконструйований та виготовлений таким чином, щоб ним можна було користуватись за нормальних умов експлуатації та за умов несправності, при цьому повинні бути забезпечені:

- захист користувача від враження струмом;
- захист користувача від впливу високих температур;

					РК51.453321.001 ПЗ	Лист
						11
Зм.	Лист	№ док-м.	Підпис	Дата		

- захист користувача від впливу випромінювання;
- захист користувача від наслідків порушення механічної рівноваги апарату і його рухомих частин.

Цабай Д.В. РК-51, 2019

					РК51.453321.001 ПЗ	Лист
						12
Зм.	Лист	№ докум.	Підпис	Дата		

2 ОБГРУНТУВАННЯ ТА ВИБІР СХЕМОТЕХНІЧНОГО РІШЕННЯ

Проаналізувавши аналоги в попередньому розділі та визначившись із функціональними особливостями робота, було вирішено створити структурну схему керування роботом.

2.1 Структурна схема

На рис. 2.1 показана структурна схема робота-маніпулятора з голосовим керуванням на базі Arduino Uno.

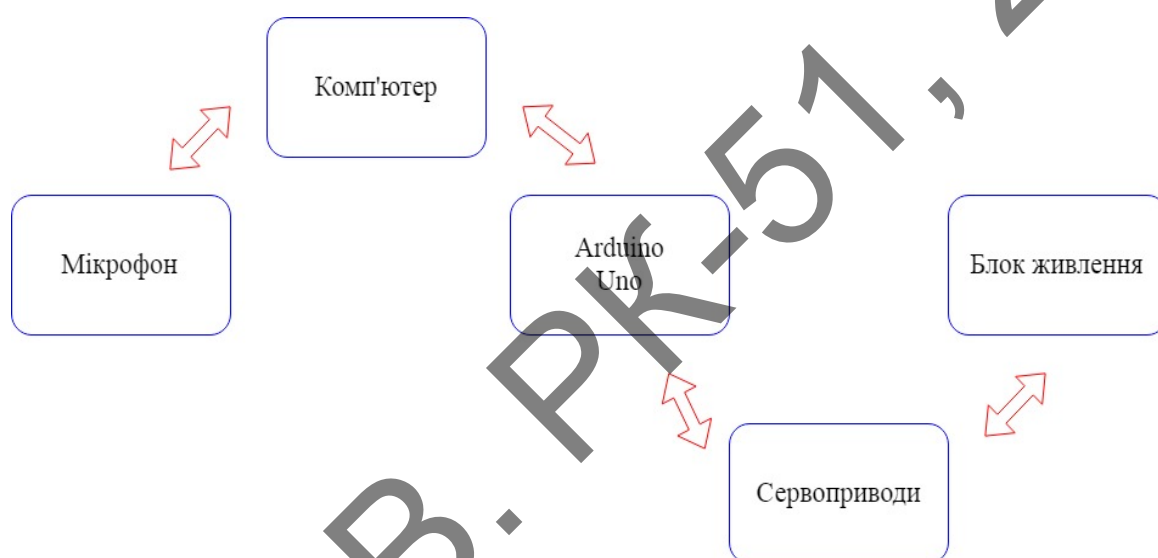


Рисунок 2.1 — Структурна схема робота-маніпулятора з голосовим керуванням

На структурній схемі показано з'єднання елементів. Комп'ютер зв'язує мікрофон та Arduino Uno, побудованої на базі мікроконтролера (МК) ATmega 328P. Arduino Uno є основною складовою блоку керування роботом, за допомоги мікрофона передається команда на Arduino Uno, з якого іде сигнал на сервоприводи і він виконує певний рух. З блоку живлення подається напруга 12 В на понижувач напруги, який на виході має 5 В. Далі з перетворювача напруги йде живлення на сервоприводи, за допомогою яких і працює робот-маніпулятор.

Зм.	Лис	№ докum.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
13

2.2 Вибір апаратної платформи

У даному дипломному проекті представлено розробку робота-маніпулятора з голосовим керуванням на базі Arduino Uno.

Arduino Uno — це широко використовувана плата мікроконтролерів з відкритим кодом на базі мікроконтролера ATmega328P [7]. Зовнішній вигляд Arduino Uno зображено на рис. 2.2:

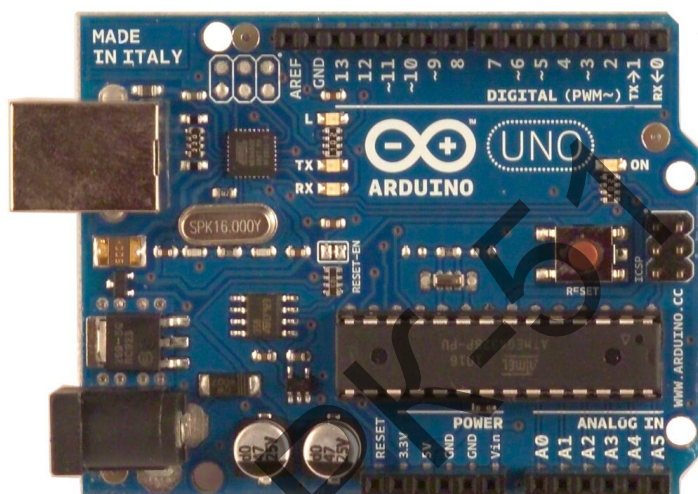


Рисунок 2.2 — Зовнішній вигляд Arduino Uno

У його склад входить все необхідне для зручної роботи з мікроконтролером: 14 цифрових входів/виходів, з них 6 можуть використовуватися в якості ШІМ-виходів, 6 аналогових входів, кварцовий резонатор на 16 МГц, роз'єм USB, роз'єм живлення, роз'єм для програмування всередині схеми і кнопка скидання. ШІМ (Широтно-імпульсна модуляція) це операція керуванням зміни вихідного сигналу за допомогою зміни часу тривалості імпульсу. Для початку роботи з пристроєм досить просто подати живлення від АС/DC-адаптера або батарейки, або підключити його до комп'ютера за допомогою USB-кабелю [7].

У таблиці 2.1 наведено характеристики Arduino Uno [7].

Зм.	Лист	№ док-м.	Підпис	Дата

PK51.453321.001 ПЗ

Лист

14

Таблиця 2.1 — Характеристики Arduino Uno

Параметр	Значення
Мікроконтролер	ATmega328P
Робоча напруга	5 В
Напруга живлення (рекомендована)	7–12 В
Напруга живлення (гранична)	6–20 В
Цифрові входи / виходи	14 (з них 6 можуть використовуватися в якості ШІМ-виходів)
Аналогові входи	6
Максимальний струм одного виведення	40 мА
Максимальний вихідний струм виводу 3.3V	50 мА
Flash-пам'ять	32 КБ (ATmega328P) з яких 0.5 КБ використовуються завантажувачем
SRAM	2 КБ (ATmega328P)
EEPROM	1 КБ (ATmega328P)
Тактова частота	16 МГц

Arduino Uno може житися від USB або від зовнішнього джерела живлення — тип джерела вибирається автоматично [7].

В якості зовнішнього джерела живлення може використовуватися мережевий AC/DC-адаптер або акумулятор/батарея [7].

Напруга зовнішнього джерела живлення може бути в межах від 6 до 20 В. Однак, зменшення напруги живлення нижче 7 В призводить до зменшення напруги на виході 5V, що може стати причиною нестабільної роботи пристрою. Використання напруги більше 12 В може призводити до перегріву стабілізатора напруги і виходу плати з ладу. З огляду на це, рекомен-

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист

15

дується використовувати джерело живлення з напругою в діапазоні від 7 до 12 В [7].

Обсяг флеш-пам'яті ATmega328P становить 32 КБ (з яких 0.5 КБ використовуються завантажувачем). Мікроконтролер також має 2 КБ пам'яті SRAM (static random access memory) — напівпровідникова оперативна пам'ять і 1 КБ EEPROM (electrically erasable programmable read-only memory) — постійний запам'ятовувальний пристрій, один з видів енергонезалежної пам'яті (з якої можна зчитувати або записувати інформацію за допомогою бібліотеки EEPROM) [7].

З використанням функцій pinMode (), digitalWrite () і digitalRead () кожен з 14 цифрових виводів може працювати в якості входу або виходу. Рівень напруги на виводах обмежений 5 В. Максимальний струм, який може споживати один вивід, становить 40 мА. Всі виводи пов'язані з внутрішніми резисторами номіналом 20-50 кОм. Крім цього, деякі виводи Arduino можуть виконувати додаткові функції [7]:

- **Послідовний інтерфейс:** виводи 0 (RX) і 1 (TX). Використовуються для отримання (RX) і передачі (TX) даних по послідовному інтерфейсу. Ці виводи з'єднані з відповідними виводами мікросхеми ATmega8U2, яка виконує роль перетворювача USB-UART. UART (Universal Asynchronous Receiver-Transmitter) — асинхронний приймач-передавач, компонентів комп'ютерів. [6];

- **Зовнішні переривання:** виводи 2 і 3. Можуть служити джерелами переривань, що виникають при фронті, спаді або при низькому рівні сигналу на цих виводах;

- **ШІМ:** виводи 3, 5, 6, 9, 10 та 11. За допомогою функції analogWrite () можуть виводити 8-бітові аналогові значення в вигляді ШІМ-сигналу;

- **Інтерфейс SPI:** виводи 10(), 11(), 12(), 13(). Із застосуванням бібліотеки SPI дані виводи можуть здійснювати зв'язок по інтерфейсу SPI;

					PK51.453321.001 ПЗ	Лист
						16
Зм.	Лис	№ докum.	Підпис	Дата		

- **Світлодіод:** 13. Вбудований світлодіод, приєднаний до виводу 13. При відправці значення HIGH світлодіод включається, при відправці LOW — вимикається.

В Arduino Uno є 6 аналогових входів (A0–A5), кожен з яких може уявити аналогову напругу у вигляді 10-бітного числа (1024 різних значення). За умовчанням, вимір напруги здійснюється щодо діапазону від 0 до 5 В. Проте, верхню межу цього діапазону можна змінити, використовуючи вивід AREF і функцію `analogReference()`. Крім цього, деякі з аналогових входів мають додаткові функції [7]:

- **TWI:** вивід A4 або SDA і вивід A5 або SCL. З використанням бібліотеки Wire дані виводи можуть здійснювати зв'язок по інтерфейсу TWI.

Крім перерахованих на платі існує ще кілька виводів:

- **AREF.** Опорна напруга для аналогових входів. Може бути задіяний функцією `analogReference()`;

- **Reset.** Формування низького рівня (LOW) на цьому виводі призведе до перезавантаження мікроконтролера.

На рис. 2.3 зображено конфігурацію виводів Arduino Uno.

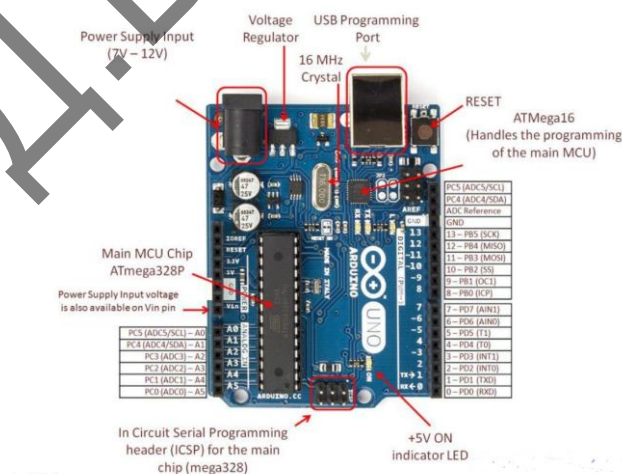


Рисунок 2.3 — Конфігурація виводів Arduino Uno та зображення його корпусу

Arduino Uno надає ряд можливостей для здійснення зв'язку з комп'ютером, ще одним Arduino або іншими мікроконтролерами.

					PK51.453321.001 ПЗ	Лист
						17
Зм.	Лис	№ доквм.	Підпис	Дата		

АТmega328 має приймач UART, що дозволяє здійснювати послідовний зв'язок за допомогою цифрових виводів 0 (RX) і 1 (TX). Мікроконтролер АТmega16U2 на платі забезпечує зв'язок цього приймача з USB-портом комп'ютера, і при підключенні до ПК дозволяє Arduino визначатися як віртуальний COM-порт.

COM-порт (Communications port) — двонаправлений послідовний інтерфейс, призначений для обміну байтовою інформацією. Прошивка мікросхеми 16U2 використовує стандартні драйвера USB-COM, тому установка зовнішніх драйверів не потрібна [7].

Бібліотека SoftwareSerial дозволяє реалізувати послідовний зв'язок на будь-яких цифрових виводах Arduino Uno [7].

Arduino Uno працює на базі програмного забезпечення Arduino IDE. Мікроконтролер АТmega328P що вмонтований в Arduino Uno випускається з прошитим boot loader (завантажувачем), що дозволяє завантажувати програми в мікроконтролер. Взаємодія з ним відбувається за допомогою протоколу STK500 [7].

2.3 Вибір сервоприводів

Розроблювальний робот рухається за допомогою сервоприводів, яка являється одною з основних частин конструкції. На ринку існує багато сервоприводів, тому було проведено аналіз та вибір сервоприводів, основними критеріями були ціна та якість.

Перший варіант сервопривода це 3.7g DS37, — це легкий мікро-сервопривід, який керується зазвичай за допомогою Arduino або інших мікроконтролерів [8].

В таблиці 2.2 наведено характеристики 3.7g DS37.

					PK51.453321.001 ПЗ	Лист
						18
Зм.	Лис	№ доквм.	Підпис	Дата		

Таблиця 2.2 — Характеристики 3.7g DS37

Параметр	Значення
Розміри, мм	20,1 × 9,1 × 17,6
Вага, г	3,7
Напруга живлення, В	4,8–6
Кут повороту, градуси	180

Ціна даного сервоприводу являється середнім показником в цінній категорії серед обраних сервоприводів.

На рис. 2.4 зображено сервопривід 3.7g DS37:



Рисунок 2.4 — Сервопривід 3.7g DS37

Другий варіант, самий дорогий з трьох обраних сервоприводів, це Tower Pro MG90S 14g [9] — легкий та якісний сервопривід, який керується від Arduino або від мікроконтролерів. Головною перевагою даного сервопривода є металевий редуктор, що дає змогу підіймати більш важкі предмети. Характеристики Tower Pro MG90S 14g наведені в таблиці 2.3:

Таблиця 2.3 — Характеристики Tower Pro MG90S 14g

Параметр	Значення
Розміри, мм	22,8×12,2×28,5
Вага, г	13,4
Напруга живлення, В	4,8–6
Кут повороту, градуси	180

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист

19

На рис. 2.5 зображено Tower Pro MG90S 14g:



Рисунок 2.5 — Tower Pro MG90S 14g та кріплення

Головним недоліком Tower Pro MG90S 14g — це ціна, хоча цей сервопривід більш надійніший за своїх конкурентів.

Отже порівнявши двох основних аналогів було обрано сервоприводи Tower Pro SG90 g9, тому що вони найдешевші в своєму сегменті та відповідають характеристикам які потрібні для робота.

Сервопривід Tower Pro SG90 g9 — це мотор-редуктор, здатний повертати вихідний вал в задане положення (на заданий кут) і утримувати його в цьому положенні, всупереч опорам [10].

В корпусі сервопривода знаходиться невеликий модуль управління, який під дією вхідного сигналу подає живлення відповідної полярності на електродвигун. Вхідний сигнал управління містить дані про необхідне положення валу. Для визначення поточного положення валу редуктор з'єднаний з двигуном змінного резистора.

Електроніка Tower Pro SG90 g9 обчислює різницю між поточним становищем редуктора і необхідним. Модуль управління орієнтуючись на опір змінного резистора подає живлення необхідної полярності на двигун для повороту редуктора [11].

					РК51.453321.001 ПЗ	Лист
						20
Зм.	Лис	№ доквм.	Підпис	Дата		

На рис. 2.6 зображено загальний вигляд Tower Pro SG90 g9:



Рисунок 2.6 — Загальний вигляд корпусу Tower Pro SG90 g9 та кріплення

Даний сервопривід Tower Pro SG90 g9 має 3-х контактний кабель. Червоний дріт підключається до позитивного полюса джерела живлення, коричневий до негативного, а помаранчевий провід підключається до джерела подачі сигналів управління. З легкістю працює з різними контролерами [12].

Сервопривід керується за допомогою імпульсів змінної тривалості. У таких імпульсів виділяється всього три параметри: мінімальна і максимальна тривалість імпульсу, а також його частота повторення [12].

Будь-який сервопривід очікує надходження імпульсу кожні 20 мс. Від тривалості імпульсу залежить кут повороту мотора [12].

Іншим важливим параметром сервоприводу є його швидкість обертання. Під швидкістю обертання розуміється час, який необхідний сервоприводу, щоб зробити поворот з одного положення в інше [12].

Швидкість сервоприводу вимірюється інтервалом часу, який потрібно важелю сервоприводу, щоб повернутися на 60° . Характеристика $0,1 \text{ с} / 60^\circ$ означає, що сервопривід повертається на 60° за $0,1 \text{ с}$. З неї нескладно вирахувати швидкість в більш звичній величині, оборотах на хвилину, але так склалося, що при описі сервоприводів найчастіше використовують таку одиницю [12].

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
21

В таблиці 2.5 представлені основні характеристики сервопривода Tower Pro SG90 g9 [13]:

Таблиця 2.5 — Характеристики сервопривода Tower Pro SG90 g9

Параметр	Значення
Розміри, см	$3,2 \times 3 \times 1,2$
Вага, г	9
Напруга живлення, В	4,8–6
Напруга живлення (рекомендована), В	4,8
Кут повороту, градуси	180
Час повороту на 60° (при 4,8 В без навантаження), с	0,12
Пусковий момент, кг·см	до 1,8
Споживаний струм в русі, мА	50–80
Споживаний струм в утриманні, мА	5–10

Схема підключення сервоприводу до Arduino показана рисунку нижче.

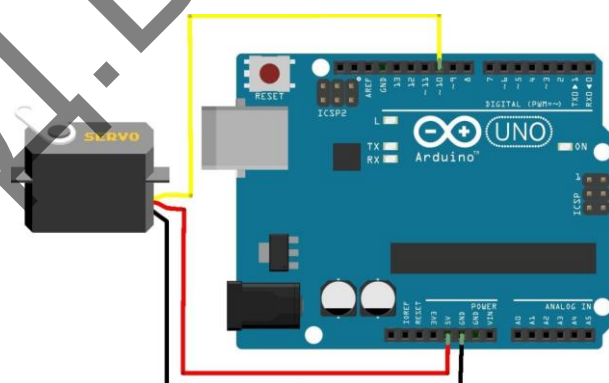


Рисунок 2.7 — Схема увімкнення сервопривода

Живлення сервопривода здійснюється від Arduino, або від окремого блоку живлення, батареї тощо.

2.4 Вибір блоку живлення

Вибір блоку живлення був сконцентрований на двох варіантах: ND-60W-24V-2.5A [14] та PL-S-25W-12V-2A [15]. Основними критеріями при виборі блока живлення: ціна, розміри та вихідний струм не більше 3 А.

Імпульсний блок живлення ND-60W-24V-2.5A використовується для потужних світлодіодів, світлодіодного освітлення, аудіо підсилювачів тощо [14]. Характеристики ND-60W-24V-2.5A наведені в таблиці 2.6:

Таблиця 2.6 — Характеристики ND-60W-24V-2.5A

Параметр	Значення
Потужність, Вт	60
Вхідна напруга, В	220
Вихідна напруга, В	24
Вихідний струм, А	2,5
Робоча температура, градуси	-20–50
Розміри, мм	100×77×35

На рис. 2.8 зображено імпульсний блок живлення ND-60W-24V-2.5A:



Рисунок 2.8 — Імпульсний блок живлення ND-60W-24V-2.5A

Головним недоліком даного блоку живлення є ціна, та розміри які більше ніж у PL-S-25-12.

Зм.	Лис	№ докum.	Підпис	Дата

PK51.453321.001 ПЗ

Оскільки робот є макетом для показу працездатності програми, тому обрано імпульсний блок живлення Prolum PL-S-25-12, якого достатньо по заданим характеристикам та ціною що менше ніж у ND-60W-24V-2.5A, що є великою перевагою над його аналогом.

Взагалі даний блок живлення найчастіше використовується в системах світлодіодного освітлення.

На рис. 2.9 зображено імпульсний блок живлення Prolum PL-S-25-12:



Рисунок 2.9 — Імпульсний блок живлення Prolum PL-S-25-12

Оскільки блок живлення підходить по параметрам (див. табл. 2.7) для дипломного проекту, тому був обраний саме імпульсний блок живлення Prolum PL-S-25-12 [15].

Таблиця 2.7 — Параметри Prolum PL-S-25-12

Параметр	Значення
Потужність, Вт	25
Вхідна напруга, В	220
Вихідна напруга, В	12
Вихідний струм, А	2,08
Робоча температура, градуси	-20–50
Розміри, мм	70×41×29

Також головними критеріями при виборі блоку живлення були: розміри, захист від короткого замикання та захист від перенавантаження.

2.5 Вибір перетворювача напруги

В минулому підрозділі було обрано імпульсний блок живлення PL-S-25-12 в параметрах якого зазначено, що вихідна напруга становить 12 В. Для сервоприводів достатньо 5 В, тому було вирішено додати понижуючий перетворювач напруги. Було розглянуто два варіанти: DC-DC LM2596 LED Voltmeter [16] та DC-DC LM2596 [17]. Перетворювач напруги обирався по критеріям ціни, характеристики порівнювати немає сенсу тому що вони однакові, оскільки базуються на LM2596.

DC-DC LM2596 LED Voltmeter це простий та надійний понижувач напруги, з влаштованим цифровим вольтметром вхідного/вихідного напруження, який зображений на рис. 2.10 [16]:

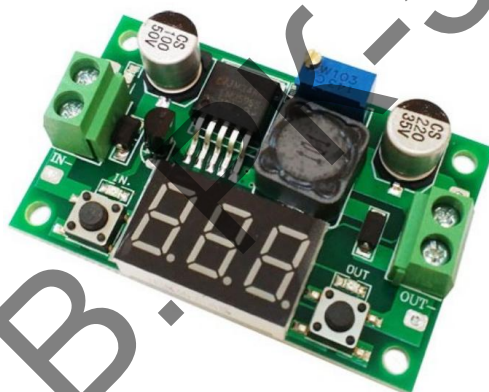


Рисунок 2.10 — Понижуючий перетворювач DC-DC LM2596 LED Voltmeter

Головною перевагою даного перетворювача є цифровий вольтметр, за допомогою якого легко налаштувати вхідне/вихідне напруження. Головним недоліком являється ціна.

Оскільки перетворювач напруги DC-DC LM2596 LED Voltmeter занадто дорогий, хоча задовольняє по характеристикам, було обрано більш простий перетворювач напруги, а саме DC-DC LM2596, він відрізняється лиш тим що у нього немає цифрового показника.

Перетворювач напруги DC-DC LM2596 призначений для пониження напруги шляхом регулювання підстроюваним резистором, в межах від 3 В до 35 В.

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
25

На рис. 2.11 зображено понижаючий перетворювач напруги DC-DC LM2596 [17]:



Рисунок 2.11 — Понижуючий перетворювач напруги DC-DC LM2596

В таблиці 2.8 наведено параметри перетворювача напруги DC-DC LM2596.

Таблиця 2.8 — Параметри перетворювача напруги DC-DC LM2596

Параметр	Значення
Вхідна напруга, В	4,5–40
Вихідна напруга, В	3–35
Вихідний струм, А	2
ККД, %	92
Робоча температура, градуси	-40–85
Розміри, мм	44×22×15

Застосована мікросхема LM2596 має високий ККД і малі втрати на нагрів що дозволило застосувати її без додаткового радіатора. У модулі застосований сучасний фільтр на оригінальних конденсаторах Sanyo, що знизило рівень пульсацій до рівня 50мВ. Перетворювач відмінно підходить для живлення світлодіодів, зарядки акумуляторів і як лабораторний джерело живлення з регульованим обмеженням максимального струму [17].

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
26

3 РОЗРОБЛЕННЯ АЛГОРИТМУ

3.1 Вибір мови програмування для Arduino

Мови програмування для платформи Arduino основані на C/C++ та скомпільовані з бібліотекою AVR Libc що дозволяє використовувати будь-які її функції. На даний час програмування на платформі Arduino — це найзручніший спосіб програмування пристроїв на мікроконтролерах.

Сучасні мікроконтролери можуть програмуватися на різних мовах програмування, які за допомогою компіляторів перекладаються на мову Assembler.

Для сучасних мікроконтролерів, що мають більшу швидкість, розроблено мови програмування високого рівня, що полегшують роботу програміста та написання програмного забезпечення (ПЗ).

В даному дипломному проекті використовується мова програмування C++, обґрунтування чому саме ця мова програмування обумовлена декількома факторами:

- розроблювана система не є складною та вимогливою до апаратної частини модуля;
- платформа Arduino має достатній об'єм пам'яті та достатню швидкість, що не обмежує у виборі мови програмування;
- існування середовища програмування Arduino IDE (integrated development environment) — крос-платформний додаток, що використовується для запису та завантаження програм на Arduino;
- мова програмування C++ не потребує високої кваліфікації програміста, що також не займає багато часу для написання програмного забезпечення та його модернізації.

					РК51.453321.001 ПЗ	Лист
						27
Зм.	Лист	№ докв.	Підпис	Дата		

3.1.1 Структурна схема для написання програми для Arduino

Для написання програми для Arduino, яка керує роботом-маніпулятором, була розроблена структура схема, зображена на рис. 3.1.



Рисунок 3.1 — Структурна схема для написання програми

Програма що розроблена на даній структурній схемі дозволяє керувати роботом-маніпулятором.

3.1.2 Розробка програми для Arduino

Текст програми:

```
#include <Servo.h> //підключення бібліотек для сервоприводів
```

```
Servo servopovorot;
```

```
Servo servocatch;
```

```
Servo servoverh; //присвоєння імен для кожного з сервоприводів
```

```
Servo servovpered;
```

Зм.	Лис	№ докum.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
28

```

int pos1 = 0;
int pos2 = 0;
int pos3 = 0; //змінні для передачі кута
int pos4 = 0;
int pos5 = 0;
int angle = 90;
int k = 0;

//налаштування програми (прописання контактів для Arduino)
void setup() {
    Serial.begin(9600); //встановлення швидкості передачі даних
    servopovorot.attach(3);
    servocatch.attach(6);
    servoverh.attach(5);
    servovpered.attach(9);
}

//задачі для програми (рух сервоприводів)
void loop() {
    if (Serial.available() > 0)
    {
        k = Serial.parseInt();
        switch (k)
        {
            case 10: //тест
                for (pos5 = 90; pos5 >=40; pos5 -= 1) { //open
                    servocatch.write(pos5);
                    delay(20);}

                for (pos5 = 40; pos5 <=90; pos5 += 1) { //close
                    servocatch.write(pos5);
                    delay(20); }
        }
    }
}

```

Зм.	Лис	№ докum.	Підпис	Дата

PK51.453321.001 ПЗ

Лист

29

```
for (pos5 = 0; pos5 <= 180; pos5 += 1) { //left
servopovorot.write(pos5);
delay(20);}
```

```
for (pos5 = 180; pos5 >= 0; pos5 -= 1) { //right
servopovorot.write(pos5);
delay(20);}
break;
```

```
case 1: // відкриття клешні
for (pos2 = 90; pos2 >= 40; pos2 -= 1) {
servocatch.write(pos2);
delay(20);}
break;
```

```
case 2: // закриття клешні
for (pos2 = 40; pos2 <= 90; pos2 += 1) {
servocatch.write(pos2);
delay(20); }
break;
```

```
case 3: // поворот на ліво
for (pos1 = angle; pos1 <= 180; pos1 += 1) {
servopovorot.write(pos1);
delay(20);}
angle = 180;
break;
```

```
case 4: // поворот на право
for (pos1 = angle; pos1 >= 0; pos1 -= 1) {
servopovorot.write(pos1);
delay(20);}
angle = 0;
```

Зм.	Лис	№ докum.	Підпис	Дата

PK51.453321.001 ПЗ

Лист

30

```
break;
```

```
case 5:// центр
```

```
if(angle > 90)// якщо крутить з лівого положення
```

```
{
```

```
for (pos1 = angle; pos1 >= 90; pos1 -= 1) { //right
```

```
servopovorot.write(pos1);
```

```
delay(20);}
```

```
angle = 90;
```

```
}
```

```
if (angle < 90) //якщо крутить з правого положення
```

```
{
```

```
for (pos1 = angle; pos1 <= 90; pos1 += 1) { //left
```

```
servopovorot.write(pos1);
```

```
delay(20);}
```

```
angle = 90;
```

```
}
```

```
servopovorot.write(90);
```

```
delay(3000);
```

```
break;
```

```
case 6: // вверх
```

```
for (pos3 = 60; pos3 <= 130; pos3 += 1) {
```

```
servoverh.write(pos3);
```

```
delay(20);}
```

```
break;
```

```
case 7: //вниз
```

```
for (pos3 = 130; pos3 >= 60; pos3 -= 1) {
```

```
servoverh.write(pos3);
```

```
delay(20);}
```

```
break;
```

```
case 8: //вперед
```

Зм.	Лис	№ докum.	Підпис	Дата

PK51.453321.001 ПЗ

Лист

31

```

for (pos4 = 20; pos4 <= 60; pos4 += 1) {
servovpered.write(pos4);
delay(20);}
break;

case 9: // назад
for (pos4 = 60; pos4 >= 20; pos4 -= 1) {
servovpered.write(pos4);
delay(20);}
break;
}
}
}

```

Дана програма дозволяє виконувати команди роботу-маніпулятору. За допомогою програмного забезпечення для голосового керування, яке буде розглянуто пізніше, Arduino отримує сигнал та передає його на сервоприводи, які в свою чергу виконують команду.

3.2 Програма для голосового керування

Для голосового керування була розроблена програма на програмному забезпеченні Visual Studio 2019 на мові програмування C#.

Visual Studio — серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight [18].

Дана програма розбита на чотири блоки, які відповідають за певні функції:

					PK51.453321.001 ПЗ	Лист
						32
Зм.	Лис	№ доквм.	Підпис	Дата		

1. Робота з COM-портом

Текст програми:

```
public partial class MainWindow : Window
{
    System.Timers.Timer aTimer; //об'ява змінних для роботи з
COM-портом Arduino
    SerialPort currentPort;
    private delegate void updateDelegate(string txt); // об'ява делега-
та (ссылки на функцію) для виводу текста з COM-порта
    SpeechRecognitionEngine sre;
    static Label l; //об'ява змінних для роботи з COM-портом
Arduino
    static SerialPort myPort;
    public MainWindow() // функція задає вікно
    {
        InitializeComponent();
        this.Left = SystemParameters.PrimaryScreenWidth - this.Width;
        //об'ява розміру основного вікна
        this.Height = SystemParameters.PrimaryScreenHeight;
        Instance = this;
    }
    private void OnTimedEvent(object sender, ElapsedEventArgs e) //
функція яка відповідає за роботу з COM-портом
    {
        // у разі не доступності COM-порту воно вертає помилку, а
не закриває програму
        try
        {
            if (!currentPort.IsOpen)
            {
                return;
            }
        }
        catch (Exception ex)
```

Зм.	Лис	№ докum.	Підпис	Дата

PK51.453321.001 ПЗ

Лист

33

```

        {
            bt_connect.Visibility = Visibility.Visible;
        }
        // оновлює текст мітки згідно інформації на COM-порті
        try
        {
            currentPort.DiscardInBuffer(); // видаляє стару інформацію
            string strFromPort = currentPort.ReadLine(); // зчитує остан-
            нє значення
            lblPortData.Dispatcher.BeginInvoke(new
            updateDelegate(updateTextBox), strFromPort);
        }
        catch (Exception ex)
        {
            lblPortData.Dispatcher.BeginInvoke(new
            updateDelegate(updateTextBox), ex.Message);
        }
    }
    private void updateTextBox(string txt) // допоміжна функція для
    передачі даних з COM-порта
    {
        lblPortData.Content = txt;
    }
    bool FindPort() // пошук доступного COM-порта
    {
        bool ArduinoPortFound = false;

        try
        {
            string[] ports = SerialPort.GetPortNames();
            if (ports.Length != 0)
            foreach (string port in ports)
            {

```

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
34

```

        currentPort = new SerialPort(port, 9600);
        ArduinoPortFound = true;
    }
}
catch { }

if (ArduinoPortFound == false) return false;
else return true;
}

void PortConnect() //основна функція для підключення COM-
порту
{
    System.Threading.Thread.Sleep(500); // затримка
    currentPort.BaudRate = 9600;
    currentPort.DtrEnable = true;
    currentPort.ReadTimeout = 1000;
    try
    {
        currentPort.Open();
    }
    catch { }

    aTimer = new System.Timers.Timer(1000);
    aTimer.Elapsed += OnTimedEvent;
    aTimer.AutoReset = true;
    aTimer.Enabled = true;
    //my_Rec_setup("ru-ru");
    //my_Rec_setup("en-en");
}

```

В даному блоці програми відбувається робота з COM-портом за допомогою основної функції void PortConnect().

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист

35

void — це псевдонім для типу System.Void в .NET Framework. При використанні в якості типу значення, що повертається для методу ключове слово void вказує, що метод не повертає значення. Ключове слово void також використовується в небезпечному контексті для оголошення покажчика на невідомий тип [19].

2. Відправлення сигналів на Arduino через COM-порт

Текст програми:

```
private void btn_Click(Button btn) // допоміжна функція для відправ-
лення керуючих сигналів на COM-порт
```

```
{
    if (currentPort == null) return;
    if (!currentPort.IsOpen) return;
    //Button btn = sender as Button;
    string str = btn.Name;
    if (str == "bt_test") currentPort.Write("10");
    if (str == "bt_open") currentPort.Write("1");
    if (str == "bt_close") currentPort.Write("2");
    if (str == "bt_left") currentPort.Write("3");
    if (str == "bt_right") currentPort.Write("4");
    if (str == "bt_center") currentPort.Write("5");
    if (str == "bt_up") currentPort.Write("6");
    if (str == "bt_down") currentPort.Write("7");
    if (str == "bt_forward") currentPort.Write("8");
    if (str == "bt_rotate") currentPort.Write("9");
}
```

```
public static MainWindow Instance { get; private set; } // обявлен-
ня інтерфейсу форми
```

```
static void sre_SpeechRecognized(object sender,
SpeechRecognizedEventArgs e) //допоміжна функція для пере-
дачі керуючих сигналів на COM-порт
```

```
{
    if (e.Result.Confidence > 0.7)
```

Зм.	Лис	№ докum.	Підпис	Дата

PK51.453321.001 ПЗ

Лист

36

```

{
    Button somebutton = MainWindow.Instance.bt_test; ;

    I.Content = e.Result.Text; // ступінь відповідності

    string str = I.Content.ToString();
    if (str == "тест") somebutton = MainWindow.Instance.bt_test;
    // myPort.Write("0");
    if (str == "открыть") somebutton =
    MainWindow.Instance.bt_open; //myPort.Write("1");
    if (str == "закрыть") somebutton =
    MainWindow.Instance.bt_close; // myPort.Write("2");
    if (str == "лево") somebutton = MainWindow.Instance.bt_left;
    // myPort.Write("3");
    if (str == "право") somebutton =
    MainWindow.Instance.bt_right; // myPort.Write("4");
    if (str == "центр") somebutton =
    MainWindow.Instance.bt_center; // myPort.Write("5");
    if (str == "вверх") somebutton = MainWindow.Instance.bt_up;
    // myPort.Write("6");
    if (str == "вниз") somebutton =
    MainWindow.Instance.bt_down; // myPort.Write("7");
    if (str == "вперед") somebutton =
    MainWindow.Instance.bt_forward; // myPort.Write("8");
    if (str == "назад") somebutton =
    MainWindow.Instance.bt_rotate; // myPort.Write("9");

    ButtonAutomationPeer peer = new
    ButtonAutomationPeer(somebutton);
    IInvokeProvider invokeProv =
    peer.GetPattern(PatternInterface.Invoke) as IInvokeProvider;
    invokeProv.Invoke();
}
}

```

Зм.	Лис	№ докum.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
37

Основною дією даного блоку є відправлення сигналів на Arduino за допомогою ButtonAutomationPeer.

ButtonAutomationPeer — це спеціальний робочий клас для Button клас елемента управління. Однорангові класи приблизно схожі на типи елементів управління автоматизації призначеного для користувача інтерфейсу, але відносяться до елементів WPF. Код автоматизації, який звертається до додатків WPF через інтерфейс автоматизації призначеного для користувача інтерфейсу, не використовує однорангові класи автоматизації безпосередньо, але код автоматизації в тому ж просторі процесу може використовувати їх безпосередньо [20].

3. Функція для розпізнавання голосових команд

Текст програми:

```
void my_Rec_setup(string language = "ru-ru") //основна функція для  
розпізнавання голосових команд
```

```
{  
    I = this.LabelVoice;  
    myPort = this.currentPort;  
    //System.Globalization.CultureInfo ci = new  
    System.Globalization.CultureInfo("ru-ru");  
    System.Globalization.CultureInfo ci = new  
    System.Globalization.CultureInfo(language);  
    sre = new SpeechRecognitionEngine(ci);  
    sre.SetInputToDefaultAudioDevice();  
  
    sre.SpeechRecognized += new  
    EventHandler<SpeechRecognizedEventArgs>(sre_SpeechRec  
    ognized); //отримувач який ссилається на функцію для пере-  
    дачі керуючих команд  
    Choices numbers = new Choices(); //порівняння записаних звуків  
    з потрібними керуючими командами  
    if (language == "ru-ru")
```

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
38

```

//numbers.Add(new string[] { "ноль", "один", "два", "три",
"четыре", "пять", "шесть", "семь", "восемь", "девять" });
numbers.Add(new string[] { "тест", "открыть", "заккрыть",
"лево", "право", "центр", "вверх", "вниз", "вперед", "назад" });
if (language == "en-en")
numbers.Add(new string[] { "zero", "one", "two", "three",
"four", "five", "six", "seven", "eight", "nine" });
GrammarBuilder gb = new GrammarBuilder(); //розпізнавання
граматики
gb.Culture = ci;
gb.Append(numbers);
Grammar g = new Grammar(gb);
sre.LoadGrammar(g);
sre.RecognizeAsync(RecognizeMode.Multiple);
}

```

Даний блок програми являється основним, оскільки в ньому реалізовується голосове керування, яке розпізнається за допомогою класів SpeechRecognized та GrammarBuilder.

GrammarBuilder використовується, щоб побудувати ієрархічне дерево, що складається з об'єктів Choices, які містять альтернативні фрази, змішані з preamble і post-ample в кожному вузлі, і посіяні семантичними значеннями, які передають сенс назад до програми [21].

SpeechRecognized екземпляр цього класу можна створити для будь-якого з встановлених розпізнавань мови. Цей клас під керуванням speech recognition ядр в процесі дозволяє контролювати різні аспекти розпізнавання мови [22].

Розпізнавання мовлення може бути двох типів на основі грамматики, на якій базується розпізнавання. Іншими словами, граматики — це список можливих результатів розпізнавання, які можуть бути сформовані. Програма може обмежити можливу комбінацію слів, сказаних вибором відповідної грамматики [23].

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
39

У сценарії команд і керування розробник надає обмежений набір можливих комбінацій слів, і механізм розпізнавання мовлення збігається зі словами, висловленими користувачем, до обмеженого списку. У команді і контролі точність розпізнавання дуже висока. Програми завжди краще реалізувати команду та управління, оскільки більш висока точність розпізнавання робить додаток краще реагує [23].

У режимі диктування механізм розпізнавання порівнює вхідну мову з усім списком словника. Для режиму диктування, щоб мати високу точність розпізнавання, важливо, щоб користувач попередньо навчив двигун розпізнавання, виступаючи в ньому [23].

4. Функція натискання кнопки

Текст програми:

```
private void Bt_open_Click(object sender, RoutedEventArgs e) // опрацьовувач подій на натискання кнопки
```

```
{
    image_open.Visibility = Visibility.Visible;
    image_close.Visibility = Visibility.Collapsed;
    btn_Click(sender as Button);
}
```

```
private void Bt_close_Click(object sender, RoutedEventArgs e)
```

```
{
    image_open.Visibility = Visibility.Collapsed;
    image_close.Visibility = Visibility.Visible;
    btn_Click(sender as Button);
}
```

```
private void Bt_rotate_Click(object sender, RoutedEventArgs e)
```

```
{
    btn_Click(sender as Button);
}
```

```
private void Bt_test_Click(object sender, RoutedEventArgs e)
```

```
{
    btn_Click(sender as Button);
}
```

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
40

```

    }
    private void Bt_up_Click(object sender, RoutedEventArgs e)
    {
        image_up.Visibility = Visibility.Visible;
        image_forward.Visibility = Visibility.Collapsed;
        image_down.Visibility = Visibility.Collapsed;
        btn_Click(sender as Button);
    }
    private void Bt_forward_Click(object sender, RoutedEventArgs e)
    {
        image_up.Visibility = Visibility.Collapsed;
        image_forward.Visibility = Visibility.Visible;
        image_down.Visibility = Visibility.Collapsed;
        btn_Click(sender as Button);
    }
    private void Bt_down_Click(object sender, RoutedEventArgs e)
    {
        image_up.Visibility = Visibility.Collapsed;
        image_forward.Visibility = Visibility.Collapsed;
        image_down.Visibility = Visibility.Visible;
        btn_Click(sender as Button);
    }
    private void Bt_left_Click(object sender, RoutedEventArgs e)
    {
        image_left.Visibility = Visibility.Visible;
        image_center.Visibility = Visibility.Collapsed;
        image_right.Visibility = Visibility.Collapsed;
        btn_Click(sender as Button);
    }
    private void Bt_center_Click(object sender, RoutedEventArgs e)
    {
        image_left.Visibility = Visibility.Collapsed;
        image_center.Visibility = Visibility.Visible;
        image_right.Visibility = Visibility.Collapsed;
    }

```

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист

41

```

        btn_Click(sender as Button);
    }
    private void Bt_right_Click(object sender, RoutedEventArgs e)
    {
        image_left.Visibility = Visibility.Collapsed;
        image_center.Visibility = Visibility.Collapsed;
        image_right.Visibility = Visibility.Visible;
        btn_Click(sender as Button);
    }
}

```

Головною дією даної функції являється виконання команд за допомогою натиску кнопки. Функція button click при натисканні кнопки змінює рисунок залежно від команди, яка переадресована на Arduino.

Найпростіший спосіб використання форм Windows Button — це коли елемент управління повинен виконувати певний код, при натисканні кнопки. Натиснувши Button створює елемент управління також ряд інших подій, таких як MouseEnter , MouseDown , і MouseUp події [24].

3.3 Інтерфейс програми голосового керування

Інтерфейс програми виконаний на мові програмування XAML (eXtensible Application Markup Language), що в перекладі означає "Розширювана Мова Розмітки для Додатків", це варіант мови XML від Microsoft, який описує GUI (Graphical user interface) — тип інтерфейсу, який дозволяє користувачам взаємодіяти з електронними пристроями через графічні зображення та візуальні вказівки, на відміну від текстових інтерфейсів, заснованих на використанні тексту, текстовому наборі команд та текстовій навігації. Дана мова дуже схожа на мову HTML (Hypertext Markup Language) — стандартна мова розмітки для створення веб-сторінок і веб-додатків, і з його допомогою стає можливим легко розробляти і редагувати графічний інтерфейс.

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
42

На рис. 3.2 зображено інтерфейс програми голосовим керуванням.

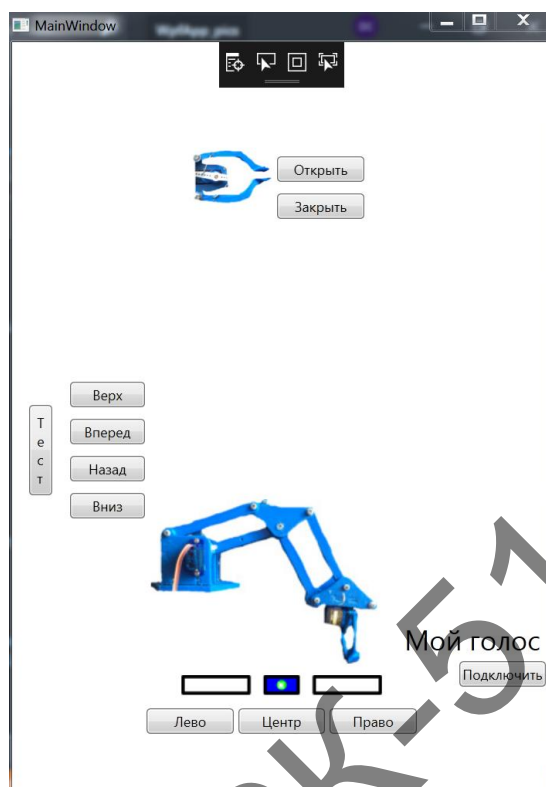


Рисунок 3.2 — Інтерфейс програми голосового керування

Основою програми є функції StackPanel та Grid. Панель StackPanel — один з найпростіших контейнерів компонування. Вона просто укладає свої дочірні елементи в один рядок або колонку. За замовчуванням панель StackPanel має елементи зверху вниз, встановлюючи висоту кожного з них такий, яка необхідна для відображення його вмісту. StackPanel може також використовуватися для впорядкування елементів в горизонтальному напрямку за рахунок установки властивості Orientation [25].

Grid підтримує впорядкування по рядках і стовпцях уявлень. Рядки і стовпці можна привласнити пропорційні розміри або абсолютні розміри. Макет не слід плутати з допомогою традиційних таблиць він не призначений для представлення табличних даних, не підтримує концепцію рядка, стовпця або комірки. На відміну від таблиць HTML призначений виключно для розташування вмісту. Також на відміну від традиційних таблиць Grid не виводить число і розміри рядків і стовпців на основі вмісту [26].

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
43

3.3.1 Розробка програми для інтерфейсу

Текс програми:

```
//стандарт описуючий групу атрибутів
<Window x:Class="WpfApp_pics.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:WpfApp_pics"
        mc:Ignorable="d"
        Title="MainWindow" Height="auto" Width="480"
        WindowStartupLocation="Manual" Top="0" Loaded="Window_Loaded">
//задання параметрів вікна інтерфейсу
    <Window.Resources>
    </Window.Resources>
    <Grid VerticalAlignment="Center"> //задання сітки
        <Grid.RowDefinitions>
            <RowDefinition Height="auto"></RowDefinition>
            <RowDefinition Height="auto"></RowDefinition>
            <RowDefinition Height="auto"></RowDefinition>
        </Grid.RowDefinitions>
        <StackPanel Grid.Row="0" Orientation="Horizontal"
            HorizontalAlignment="Center" Height="auto" VerticalAlignment="Bottom"
            Margin="10">
//функція яка відповідає за відкриття/закриття картинок
            <StackPanel>
                <Image Name="image_open" Source="Res/robot WPF
open.jpg" Visibility="Collapsed"/>
                <Image Name="image_close" Source="Res/robot WPF
close.jpg" Visibility="Visible"/>
//функція яка відповідає за положення кнопок на екрані ін-
терфейсу
            </StackPanel>
```

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист

44

```
<StackPanel Orientation="Vertical" VerticalAlignment="Center"
HorizontalAlignment="Left">
```

```
<Button Name="bt_open" Content="Открыть" Width="75"
Margin="5" Click="Bt_open_Click" />
```

```
<Button Name="bt_close" Content="Закрыть" Width="75"
Margin="5" Click="Bt_close_Click"/>
```

```
</StackPanel>
```

```
</StackPanel> //функція яка відповідає за положення та назву
кнопки «ТЕСТ»
```

```
<StackPanel Grid.Row="1" Orientation="Horizontal"
HorizontalAlignment="Center" Height="auto" VerticalAlignment="Center">
```

```
<StackPanel HorizontalAlignment="Left" Height="auto"
VerticalAlignment="Center">
```

```
<Button Name="bt_test" Height="auto" Width="auto"
Margin="10" Padding="5" Click="Bt_test_Click" >
```

```
<Button.Content>
```

```
<StackPanel Orientation="Vertical">
```

```
<TextBlock Text="T"/>
```

```
<TextBlock Text="e"/>
```

```
<TextBlock Text="c"/>
```

```
<TextBlock Text="т"/>
```

```
</StackPanel>
```

```
</Button.Content>
```

```
</Button>
```

```
</StackPanel> // функція яка відповідає за назви кнопок:
«вперед», «назад», «вниз», «вверх»
```

```
<StackPanel Orientation="Horizontal"
HorizontalAlignment="Center" Height="auto" VerticalAlignment="Center">
```

```
<StackPanel HorizontalAlignment="Left"
VerticalAlignment="Center" Height="auto" Width="75">
```

```
<Button Name="bt_up" Content="Вверх" Margin="5"
Click="Bt_up_Click"/>
```

```
<Button Name="bt_forward" Content="Вперед" Margin="5"
Click="Bt_forward_Click"/>
```

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
45

<Button Name="bt_rotate" Content="Назад" Margin="5"
Click="Bt_rotate_Click"/>

<Button Name="bt_down" Content="Вниз" Margin="5"
Click="Bt_down_Click"/>

</StackPanel>//функція відповідає за відкриття рисунків на
екрані інтерфейсу

<Grid HorizontalAlignment="Center" Height="auto"
VerticalAlignment="Center" MinHeight="369" MinWidth="341">

<Image Name="image_up" Source="Res/robot WPF 3.jpg"
Visibility="Collapsed" Stretch="None" VerticalAlignment="Bottom"
HorizontalAlignment="Left"/>

<Image Name="image_forward" Source="Res/robot WPF
2.jpg" Visibility="Collapsed" Stretch="None" VerticalAlignment="Bottom"
HorizontalAlignment="Left"/>

<Image Name="image_rotate" Source="Res/robot WPF
0.jpg" Visibility="Collapsed" Stretch="None" VerticalAlignment="Bottom"
HorizontalAlignment="Left"/>

<Image Name="image_down" Source="Res/robot WPF
1.jpg" Visibility="Visible" Stretch="None" VerticalAlignment="Bottom"
HorizontalAlignment="Left" />

<Label x:Name="lblPortData" FontSize="24"
HorizontalAlignment="Right" VerticalAlignment="Bottom" Width="auto"
Visibility="Collapsed">No data</Label>

<Label x:Name="LabelVoice" FontSize="24"
HorizontalAlignment="Right" VerticalAlignment="Bottom" Width="auto"
Visibility="Visible">Мой голос</Label>

<Button x:Name="bt_connect" Content="Подключить"
HorizontalAlignment="Right" VerticalAlignment="Bottom" Margin="0,0,0,-20"
Width="75" Click="Bt_connect_Click"/>//кнопка підключення

</Grid>

</StackPanel>

</StackPanel>// функція відповідає за відкриття рисунків для
команд «вліво», «вправо», «центр»

Зм.	Лис	№ докum.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
46

```

        <StackPanel            Grid.Row="2"            Orientation="Vertical"
HorizontalAlignment="Center" Height="auto" VerticalAlignment="Top">
            <StackPanel Margin="10">
                <Image Name="image_left" Source="Res/robot WPF left.gif"
Visibility="Collapsed" Stretch="None"/>
                <Image Name="image_center" Source="Res/robot WPF
centr.gif" Visibility="Visible" Stretch="None"/>
                <Image Name="image_right" Source="Res/robot WPF
right.gif" Visibility="Collapsed" Stretch="None"/>
            </StackPanel> // функція кнопок «вліво», «вправо», «центр»
        <StackPanel            Grid.Row="2"            Orientation="Horizontal"
HorizontalAlignment="Center" Height="auto" VerticalAlignment="Top">
            <Button            Name="bt_left"            Content="Ліво"
HorizontalAlignment="Left" Width="75" Margin="2" Click="Bt_left_Click"/>
            <Button            Name="bt_center"            Content="Центр"
HorizontalAlignment="Center" Width="75" Margin="2"
Click="Bt_center_Click"/>
            <Button            Name="bt_right"            Content="Право"
HorizontalAlignment="Right" Width="75" Margin="2" Click="Bt_right_Click"/>
        </StackPanel>
    </StackPanel>
</Grid>
</Window>

```

Дана програма реалізована на мові програмування XAML, являється найпростішою для роботи з інтерфейсом. Інтерфейс тут поділений на три частини за допомогою функції Grid. Принцип роботи даної програми в відкритті та закритті рисунків, та передачі даних кнопками для виконання команд на Arduino.

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
47

4 ПРОЕКТУВАННЯ ПРИЛАДУ ТА АНАЛІЗ ЙОГО ПРАЦЕЗДАТНОСТІ

4.1 Опис макету

Даний прилад проектувався як макет для показу працездатності програми, тому проектування самого робота було зроблено найпростішим шляхом.

Деталі робота були надруковані на 3D принтері, тому точність складових робота була мінімальною. Сама конструкція, деталі робота були розроблені в програмному середовищі Sketch Up. Деталі з'єднані між собою за допомогою кріплень для плати, гвинтів та гайок. Для скріплення конструкції були обрані гвинти: M3×8, M3×10, M3×15, кріплення для плати TFM-M3/8, TFM-M3/10, TFM-M3/20, TFM-M3/30, та гайки M3 DIN934.

На рис. 4.1 зображено все необхідне для збору робота-маніпулятора з голосовим керуванням:



Рисунок 4.1 — Деталі для збору робота-маніпулятора з голосовим керуванням

Для даних сервоприводів зроблено окреме живлення, а саме обрано блок живлення на 12В, та до нього перетворювач напруги, у якого на виході отримуємо 5В.

Зм.	Лис	№ докum.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
48

Для під'єднань модулів між собою, використано 3 типи дротів. На вході блоку живлення, де подається напруга живлення, використано 2-х жильний дріт із перерізом 1 мм^2 . З блоку живлення використано 2-х жильний дріт із перерізом $0,5 \text{ мм}^2$, який припаяний до перетворювача напруги. З перетворювача напруги йде підключення на сервоприводи, немає потреби у дроті із таким перерізом. В цьому випадку можна використати звичайні дроти для плат Arduino.

Робот сам рухається за допомогою сервоприводів, які встановлені у спеціальні отвори на деталях робота, які запрограмовані за допомогою програмного забезпечення Arduino IDE.

Зібраний робот-маніпулятор показаний на рис. 4.2:



Рисунок 4.2 — Готовий робот-маніпулятор з голосовим керуванням

4.2 Перевірка працездатності

Перевірка працездатності полягає в роботі сервоприводів та програмного забезпечення. Для нормальної роботи сервоприводів потрібно в коді програми заздалегідь виставити кут повороту, що забезпечує рух робота. Також

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
49

сервоприводи не повинні працювати в одному положенні, це також прописується в коді програми, вразі роботи сервопривода в одному положенні він перегрівається, оскільки вал крутиться, а руху немає.

Оскільки даний дипломний проект є всього лиш прототипом, то програмне забезпечення написано найпростішим шляхом, тому головною проблемою програмного забезпечення є використання безкоштовних бібліотек голосового керування, оскільки вони реагують не на голос, а на інтонацію.

Також виконана перевірка на підключення модуля керування до комп'ютера, коли модуль підключений на головному вікні інтерфейсу не показується кнопка «Підключити», а коли підключений показується рядок з написом виконання команди.

Внаслідок аналізу працездатності отримано практичне підтвердження робити застосованих рішень, які в цілому задовольняють завданням цього дипломного проекту.

					PK51.453321.001 ПЗ	Лист
						50
Зм.	Лист	№ докum.	Підпис	Дата		

5 ОХОРОНА ПРАЦІ

Охорона праці — система законодавчих актів, соціально-економічних, гігієнічних та лікувально-профілактичних заходів і засобів, які забезпечують безпеку, збереження здоров'я і працездатності людини в процесі праці.

Завдання охорони праці — звести до мінімуму імовірність ураження або захворювання працюючого з одночасним забезпеченням комфорту при максимальній продуктивності праці. Реальні виробничі умови характеризуються, як правило, наявністю певних небезпечних і шкідливих виробничих факторів.

Оскільки даний пристрій призначений для використання на базі існуючих рішень, які мають сертифікати згідно з ДСНіП 239-96, то питання захисту від електромагнітного впливу не розглядається у даному розділі.

Основну увагу приділено питанням електробезпеки та питанням, які пов'язані із створенням безпечних та комфортних умов праці користувачів ПК.

У дипломному проекті проводиться розробка робота-маніпулятора з голосовим керуванням, тому в даному розділі запропоновано відповідні технічні рішення та організаційні заходи з безпеки і гігієни праці та виробничої санітарії, а також визначені основні заходи з пожежної безпеки та профілактики.

5.1 Визначення основних потенційно шкідливих та небезпечних виробничих факторів

Основними потенційно небезпечними та шкідливими факторами, які мають місце при застосуванні ПК (персональний комп'ютер), є:

- можливість ураження електричним струмом;
- недостатній рівень освітленості або підвищена яскравість освітленості на робочих місцях;

					РК51.453321.001 ПЗ	Лист
						51
Зм.	Лист	№ доквм.	Підпис	Дата		

– невідповідних параметрів мікроклімату робочої зони санітарним нормам;

– можливе виникнення пожежі;

– значне розумове навантаження тощо.

5.2 Технічні рішення та організаційні заходи з безпеки і гігієни праці на робочих місцях користувачів ПК

5.2.1 Вимоги з охорони праці при роботі з ПК

При роботі з комп'ютером робітник піддається впливу наступних небезпечних та шкідливих факторів як:

- наявність шуму вентиляторів;
- невідповідність освітлення;
- невідповідна організація робочого місця;
- можливість ураження електричним струмом;
- монотонність праці, тощо.

Важливу роль грає планування робочого місця, що повинна задовольняти вимогам зручності виконання робіт й економії енергії й часу працівника, зручності обслуговування комп'ютера. Нераціональна конструкція й розташування робочих місць приводить до змушеної робочої пози й до напруги кістково-м'язової системи. При тривалій роботі за екраном дисплея в операторів спостерігається виражена напруга органів зору з появою скарг на незадоволеність роботою, дратівливість, порушення сну, хворобливі відчуття в очах, області шиї й у руках. У зв'язку з цим для працівників повинні забезпечуватися оптимальні умови праці й відпочинку. Праця операторів комп'ютера ставиться до I і II класу по гігієнічних умовах праці.

Умови при роботі із моніторами ПК повинні відповідати «Вимогам щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» та ДСанПіН-3.3.2.007-98.

Зм.	Лист	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист

52

5.2.2 Оптимізації параметрів повітряного середовища в робочій зоні

Роботи з розробки програми розроблялись протягом холодної та теплої пори року, категорії виконуваних робіт - Іа.

Згідно ДСН.3.3.6-042-99 під мікрокліматом виробничих приміщень розуміють: клімат внутрішнього середовища, що визначається дією на організм людини в сполученні температури, вологості, швидкості руху повітря й теплових випромінювань.

Параметри мікроклімату можуть змінюватися в дуже широких межах і впливати на самопочуття й здоров'я працюючих, продуктивність і якість їхньої праці. Значне відхилення параметрів мікроклімату від оптимального або припустимих може бути причиною ряду фізіологічних порушень в організмі людини й привести до різкого зниження працездатності і у підсумку до професійних захворювань.

Норми на оптимальні й припустимі значення температури, відносній вологості й швидкості руху повітря встановлюються для робочої зони виробничих приміщень залежно від періоду року й категорії виконуваних робіт.

Оптимальні і припустимі параметри мікроклімату виробничих приміщень наведені в таблиці 5.1 і 5.2 відповідно.

Оптимальні норми температури, відносні вологості й швидкості руху повітря в робочій зоні виробничих приміщень зведені в таблицю 5.1.

Таблиця 5.1 — Оптимальні норми температури

Сезон року	Категорія робіт	Температура, °С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодний	Легка Іа	22-24	40-60	0,1
Теплий	Легка Іа	23-25	40-60	0,1

Зм.	Лист	№ док-м.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
53

Припустимі норми температури, відносній вологості й швидкості руху повітря в робочій зоні виробничих приміщень зведені в таблицю 5.2.

Таблиця 5.2 — Припустимі норми температури

Сезон року	Категорія робіт	Температура, °C	Відносна вологість, %	Швидкість руху повітря, м/с
Холодний	Легка 1а	21-25	75	не більше 0,1
Теплий	Легка 1а	21-28	75 (при 24°C і нижче)	0,1...0...0,2

У виробничих умовах параметри мікроклімату варто вимірювати на початку, середині і кінці холодного й теплого періодів року не менш трьох разів у зміну.

У приміщенні, де проводиться розробка програми, підтримується температура 22–23°C при вологості повітря, що не перевищує 75%, а швидкість потоку повітря – близько 0.2 м/с. Перепад температур у робочій зоні:

1. по висоті – не більше 2°C;
2. у горизонтальній площині – не більше 3°C.

Виконання норм за ДСН 3.3.6.042-99 досягається проведенням наступних мір:

1. природна вентиляція в холодний сезон року;
2. штучна вентиляція в теплий сезон;
3. опалення приміщень у холодний сезон центральним опаленням.

Умови мікроклімату в робочому приміщенні повністю відповідають вимогам ДСН 3.3.6.042-99, тому проведення додаткових заходів по їхньому поліпшенню не потрібно.

5.2.3 Вимоги до робочого місця

Вимоги до робочого місця робіт з використанням ПК регламентує нормативний документ ДСанПіН 3.3.2.007-98. Конструкція робочого столу відповідає сучасним вимогам ергономіки і забезпечувати оптимальне розміщен-

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

ня на робочій поверхні використовуваного обладнання (дисплея, клавіатури) і документів.

В якості робочого стільця використовується підйомно-поворотне крісло, регульоване за висотою, з кутом і нахилу сидіння та спинки. Регулювання за кожним із параметрів здійснюється незалежно, легко і надійно фіксується. Для зниження статичного напруження м'язів використовуються стаціонарні підлокітники, що регулюються за висотою над сидінням. Поверхня сидіння і спинки крісла напівм'яка з нековзним покриттям.

Дисплей розташований на оптимальній відстані від очей користувача з урахуванням розміру літеро-цифрових знаків і символів. Розташування дисплея забезпечує зручність зорового спостереження у вертикальній площині. Клавіатура розташована на поверхні столу на відстані 100–300 мм від краю. У конструкції клавіатури передбачений опорний пристрій, який дає змогу змінювати кут нахилу поверхні клавіатури у межах 5–150. Таке положення клавіатури зручне для праці обома руками.

Працівники з ПК підлягають обов'язковим попереднім медичним оглядам під час влаштування на роботу і періодичним — один раз на два роки комісією у складі терапевта, невропатолога та офтальмолога. Медогляди мають на меті перевірку стану здоров'я працівника і виявлення відсутності у нього медичних протипоказань, які перешкоджають виконанню роботи з персональним комп'ютером.

5.2.4 Організація оптимального режиму праці та відпочинку при використанні ПК

При організації праці, що пов'язана з використанням ПК, для збереження здоров'я працюючих, запобігання професійним захворюванням і підтримки працездатності слід передбачити регламентовані перерви для відпочинку.

Згідно з ДСанПіН 3.3.2.007-98 характер трудової діяльності відповідає роботі оператора персонального комп'ютера — роботі, яка пов'язана з обліком інформації одержаної з дисплея за попереднім запитом, або тієї, що над-

Зм.	Лис	№ докum.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
55

ходить з нього, супроводжується перервами різної тривалості, пов'язана з виконанням іншої роботи і характеризується як робота з напруженням зору, невеликими фізичними зусиллями, нервовим напруженням середнього ступеня та виконується у вільному темпі.

Тривалість регламентованих перерв під час роботи з ПК становить 15 хвилин через кожні дві години роботи.

5.2.5 Виробниче освітлення

Приміщення для роботи з ПК має природне та штучне освітлення. Природне освітлення має здійснюватися через світлові прорізи.

Для збереження тривалої працездатності, підвищення продуктивності праці суттєвим є забезпечення норм освітленості на робочому місці. Величина освітленості регламентується нормами ДБН В.2.5-28-2006. Приміщення з ПК, відповідно до ДБН В.2.5-28-2006. «Природне і штучне освітлення. Норми проектування», відноситься до класу кабінетів інформатики та обчислювальної техніки. Розряд і під розряд зорової роботи Б – 2.

Нормування штучного освітлення також здійснюється згідно ДБН В.2.5-28-2006. Для загального освітлення використовують головним чином люмінесцентні лампи, що обумовлено їхніми перевагами.

5.2.6 Виробничий шум

Для умов, що розглядаються в проекті і характеру роботи, який можна класифікувати як роботу програміста персонального комп'ютера у лабораторії для теоретичних робіт та обробки даних, рівні шуму визначені ДСН 3.3.6.037-99. «Санітарні норми виробничого шуму, ультразвуку та інфразвуку».

Джерелами шуму в умовах робочого приміщення, що розглядається в роботі є вентилятори охолодження внутрішніх систем персонального комп'ютера (вентилятори блоку живлення, радіатора процесора та відеокарти) і система кондиціонування повітря.

					PK51.453321.001 ПЗ	Лист
						56
Зм.	Лис	№ доквм.	Підпис	Дата		

Очікувані рівні звукового тиску і рівень звуку відповідно до шумових характеристик цих джерел :

- рівень шуму, створюваний внутрішніми елементами ПК дорівнює 35 дБ;
- рівень шуму системи кондиціонування на низьких/високих частотах дорівнює 25/30 дБ.

Оскільки рівні звуку потенційних джерел шуму нижчі допустимих, умови робочого приміщення відповідають вимогам.

5.2.7 Електробезпека у виробничих приміщеннях

Згідно ДСТУ ІЕС 61.140:2015 «Захист проти ураження електричним струмом. Загальні аспекти щодо установок та обладнання», все електрообладнання, що знаходиться в робочому приміщенні можна віднести до І класу по електрозахисту (І клас — обладнання, що приєднується до ланцюга живлення трьох контактними вилками, один з цих контактів з'єднується з заземленим контактом розетки).

Робоче приміщення за ступенем небезпеки ураження людей електричним струмом можна віднести, згідно ПУЕ 2017 та ДБН В.2.5-27-2006 «Захисні заходи електробезпеки в електроустановках будинків і споруд», до приміщень без підвищеної небезпеки, так як:

- відносна вологість повітря не перевищує 75%;
- матеріал підлоги (паркет) є діелектриком;
- температура повітря не досягає значень, більших 35°C;
- відсутня можливість одночасного торкання людини до елементів металоконструкцій будівель, технологічних апаратів, механізмів, що мають з'єднання з землею, з одного боку і до металевих корпусів електрообладнання з іншого.

При нормальному (неаварійному) режимі роботи електрообладнання, гранично припустимі рівні напруги дотику і струмів через людину не повинні перевищувати значень, які приведені в табл. 5.3.

					РК51.453321.001 ПЗ	Лист
						57
Зм.	Лист	№ доквм.	Підпис	Дата		

Таблиця 5.3 — Граничні значення при нормальному режимі

№ п/п	Рід струму	U, В	I, мА
1	Змінний 50Гц, не більше	2	0,3
2	Постійний, не більше	8	1

5.2.8 Розрахунок електромережі на вимикаючу здатність

Сила струму при виникненні однофазного короткого замикання на корпус електроприладу, у разі, якщо знехтувати індуктивною складовою опорів фазного та нульового дротів та еквівалентним опором трансформатору електромережі:

$$I_{кз} = \frac{U_{\phi}}{R_{\phi} + R_0 + \frac{Z_T}{3}}, \quad (5.1)$$

де $I_{кз}$ — струм короткого замкнення;

R_{ϕ} — активний опір фазового дроту;

R_0 — активний опір нульового дроту;

Z_T — активний опір трансформатора;

U_{ϕ} — напруга фази.

Максимальне значення напруги, яка появиться на корпусах електрообладнання при його аварійному режимі роботи, буде дорівнювати:

$$U_{\max} = I_{кз} \cdot R_0; \quad (5.2)$$

Таким чином при часі спрацювання автоматів максимального струмового захисту менше 0,1 секунди, напруга дотику на корпусах електрообладнання не буде перевищувати допустиме значення.

$$U_{\max} < U_{\text{дон}}, \text{ при } t < 0,1 - U_{\text{дон}}, \text{ згідно ПУЕ 2017.}$$

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист

58

Для надійної роботи автоматів струмового захисту номінальний струм їх спрацювання повинен задовольняти умові:

$$\frac{I_{кз}}{I_{сн}} > 1,4; \quad (5.3)$$

У робочому приміщенні лабораторії ця вимога виконується.

Безпека експлуатації електричного устаткування і захист від можливого ураження електричним струмом у приміщенні забезпечується застосуванням ряду захисних заходів:

- персонал, що працює з приладами повинний бути технічно грамотним;
- забезпечена недоступність і відповідна робоча ізоляція струмоведучих частин обладнання;
- ремонтний персонал, що виконує ремонтні роботи в електричних установках, має необхідну групу допуску;
- регулярно проводиться перевірка знань та інструктаж з надання першої медичної допомоги при ураженні електричним струмом.

5.3 Пожежна безпека та профілактика

Причини виникнення пожежі в робочому приміщенні, де використовуються комп'ютер, можуть носити електричний і неелектричний характер.

До причин електричного характеру ставляться: короткі замикання, перевантаження, іскріння від порушення ізоляції, електрична дуга, що виникає між контактами комутаційних апаратів, незадовільні контакти в місцях з'єднання проводів (скрутки) і їхнє сильне нагрівання внаслідок великого перехідного опору при протіканні електричного струму.

Причини неелектричного характеру: порушення режимних вимог, халатне й необережне поводження з вогнем, порушення правил пожежної безпеки.

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
59

Категорія робочого приміщення згідно ДСТУ Б.В.1.1-36:2016, відно-
ситься до класу «В». Клас робочої зони згідно ДНАОП 0.00-1.32-0, П-П.

Приміщення відповідно до ДБН В.2.5-56-2014 обладнане чотирма по-
жежними датчиками типу ДПС-038 (площа, що перебуває під захистом одно-
го датчика становить до 25 м², відстань між датчиками становить 4м). Відпо-
відно до ДСТУ 3675-98 і ISO 3941-77 як первинний засіб гасіння пожежі ви-
користовується чотири вогнегасника ОУ-3 – вуглекислий (клас пожежі “Е”).
Вибір речовини вогнегасника обґрунтовується тим, що у вогні можуть вияви-
тися електричні пристрої, що перебувають під напругою. Кількість, розмі-
щення і зміст первинних засобів гасіння пожежі повністю задовольняє всім
вимогам ДСТУ 3675-98 й ISO 3941-77. Крім цього, у коридорі є 2 пожежних
крани і ящик з піском.

Витримано всі умови ДБН В.1.1-7-2016, СНіП 2.01.02-85 і СНіП 2.09.02-
85 по вогнестійкості будинків, часу евакуації у випадку пожежі, ширині ева-
куаційних проходів і виходів із приміщень. Витримано всі умови НАПБ
А.01.001-2004 «Правила пожежної безпеки в Україні».

Зм.	Лис	№ докum.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
60

ВИСНОВКИ

1. Провели аналіз існуючих аналогів на ринку, було виявлені такі основні недоліки, як невідповідність якості до ціни виробів та відсутність голосового керування.

2. Обґрунтовано аналіз вибору схемотехнічних рішень, згідно з якими обрано апаратну платформу, сервоприводи, блок живлення та перетворювач напруги.

3. Розроблено програмне забезпечення написане за допомогою інтегрованого середовища Visual Studio 2019 на мові програмування C++/C, та розроблено інтерфейс на мовою програмування XAML.

4. Створено макет для демонстрації працездатності програмного забезпечення та проведено тестування програмного забезпечення, в результаті якого отримано практичне підтвердження роботи застосованих рішень, які в цілому задовольняють завданням цього дипломного проекту.

5. Розроблений робот-маніпулятор з голосовим керуванням задовольняє вимоги технічного завдання.

					PK51.453321.001 ПЗ	Лист
						61
Зм.	Лист	№ док-м.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Акриловый робот-манипулятор [Електронний ресурс]. — Режим доступу: <https://ru.aliexpress.com/i/32859958012.html> — Назва з екрану.
2. Робот CIC 21-535N [Електронний ресурс]. — Режим доступу: <https://parts4.net/p816146782-konstruktor-cic-535n.html> — Назва з екрану.
3. Робот CIC 21-632 [Електронний ресурс]. — Режим доступу: <http://umbum.ua/konstruktor-cic-21-632-gidravlicheskiy-manipulyator/> — Назва з екрану.
4. Робот-манипулятор Ewood Robotics [Електронний ресурс]. — Режим доступу: <http://arduino.ua/prod3074-rozvivachii-konstruktor-robot-manipulyator-ewood-robotics> — Назва з екрану.
5. Машины, приборы и другие технические изделия. Термін та видання: ГОСТ 15150-69. — [Введ. в дію 01.01.1971]. — М. : Государственный стандарт союза ССР, 1969. — 81с.
6. Аппаратура сухопутной подвижной радиосвязи. Термін та видання: ГОСТ 16019-2001. — [Введ. в дію 01.01.2002]. — М. : Межгосударственный стандарт, 2001. — 11с.
7. Arduino Uno [Електронний ресурс]. — Режим доступу: <https://doc.arduino.ua/ru/hardware/Uno> — Назва з екрану.
8. Сервопривод 3.7g DS37 [Електронний ресурс]. — Режим доступу: https://diyshop.com.ua/index.php?route=product/product&product_id=161 — Назва з екрану.
9. Сервопривод Tower Pro MG90S14g [Електронний ресурс]. — Режим доступу: <https://www.mini-tech.com.ua/servomotor-mg90> — Назва з екрану.

Зм.	Лис	№ доквм.	Підпис	Дата

PK51.453321.001 ПЗ

Лист
62

10. Tower Pro SG90 g9 [Електронний ресурс]. — Режим доступу: https://autohome.org.ua/market/moto_servo/servo-sg90-9g-detail — Назва з екрану.
11. Принцип роботи Tower Pro SG90 g9 [Електронний ресурс]. — Режим доступу: <https://arduino-kit.ru/product/servoprivod-towerpro-sg90> — Назва з екрану.
12. Характеристики Tower Pro SG90 g9 [Електронний ресурс]. — Режим доступу: <https://freedelivery.com.ua/arduino-100/dvigateli-i-servomotory-132/servoprivod-serva-tower-pro-9g-sg90-arduino-pic-2684.html> — Назва з екрану.
13. Принцип роботи Tower Pro SG90 g9 [Електронний ресурс]. — Режим доступу: <https://ukrbukva.net/page,9,100972-Razrabotka-avtomatizirovannogo-mobil-nogo-robot.html> — Назва з екрану.
14. Импульсный блок питания 24В 2.5А (60Вт) [Електронний ресурс]. — Режим доступу: https://www.mini-tech.com.ua/blok-pitaniya-60w-24v-2_5a-metall-perforirovanniy — Назва з екрану.
15. Блок живлення імпульсний PROLUM 25W 12V (IP20, 2,08A) Standard PL-S-25-12 [Електронний ресурс]. — Режим доступу: <https://prolum.com.ua/ua/blok-pitaniya-impulsnyy-prolum-negermetichnyy-25w-12v-ip20-208a-standart/> — Назва з екрану.
16. DC-DC конвертер понижающий на LM2596 со встроенным вольтметром [Електронний ресурс]. — Режим доступу: <https://www.mini-tech.com.ua/dc-dc-konverter-ponizhayushchij-lm2596-so-vstroennym-voltmetrom> — Назва з екрану.
17. DC-DC понижающий конвертер LM2596 с 4.5 – 40В до 3 – 35В [Електронний ресурс]. — Режим доступу: <https://arduino.ua/prod650-DC-DC-ponijaushhii-konverter-s-4-5-60V-do-3-35V> — Назва з екрану.
18. Microsoft Visual Studio [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio — Назва з екрану.

					PK51.453321.001 ПЗ	Лист
						63
Зм.	Лис	№ доквм.	Підпис	Дата		

19. void (справочник по C#) [Электронный ресурс]. — Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/void> — Назва з екрану.
20. Модель автоматизации пользовательского интерфейса пользовательского элемента управления WPF [Электронный ресурс]. — Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/framework/wpf/controls/ui-automation-of-a-wpf-custom-control> — Назва з екрану.
21. GrammarBuilder Class [Электронный ресурс]. — Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/api/system.speech.recognition.grammarbuilder?view=netframework-4.8> — Назва з екрану.
22. SpeechRecognizer Class [Электронный ресурс]. — Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/api/system.speech.recognition.speechrecognitionengine?view=netframework> — Назва з екрану.
23. Speech Recognition using C# [Электронный ресурс]. — Режим доступа: <https://www.c-sharpcorner.com/article/speech-recognition-using-C-Sharp/> — Назва з екрану.
24. Обработка события нажатия кнопки в Windows Forms [Электронный ресурс]. — Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/controls/how-to-respond-to-windows-forms-button-clicks> — Назва з екрану.
25. StackPanel [Электронный ресурс]. — Режим доступа: https://professorweb.ru/my/WPF/base_WPF/level3/3_3.php — Назва з екрану.
26. Сетка Xamarin.Forms [Электронный ресурс]. — Режим доступа: <https://docs.microsoft.com/ru-ru/xamarin/xamarin-forms/user-interface/layouts/grid> — Назва з екрану.

ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ

Чабай Д.В. РК-51, 2019

ДОДАТОК Б. ЛІСТИНГ ПРОГРАМИ ДЛЯ ARDUINO

```
#include <Servo.h> //підключення бібліотек для сервоприводів

Servo servopovorot;

Servo servocatch;

Servo servoverh; //присвоєння імен для кожного сервопривода

Servo servovpered;

int pos1 = 0;

int pos2 = 0;

int pos3 = 0; //змінні для передачі кута

int pos4 = 0;

int pos5 = 0;

int angle = 90;

int k = 0;

//налаштування програми (прописання пінов для Arduino)

void setup() {

    Serial.begin(9600); //встановлення швидкості передачі даних

    servopovorot.attach(3);

    servocatch.attach(6);

    servoverh.attach(5);

    servovpered.attach(9);

}

//задачі для програми (рух сервоприводів)

void loop() {

    if (Serial.available() > 0)

    {

        k = Serial.parseInt();

        switch (k)

        {

            case 10: //тест

                for (pos5 = 90; pos5 >=40; pos5 -= 1) { //open

                    servocatch.write(pos5);

                    delay(20);}

                for (pos5 = 40; pos5 <=90; pos5 += 1) { //close

                    servocatch.write(pos5);

                    delay(20); }

                for (pos5 = 0; pos5 <= 180; pos5 += 1) { //left

                    servopovorot.write(pos5);

                    delay(20);}
```

```
for (pos5 = 180; pos5 >= 0; pos5 -= 1) { //right
servopovorot.write(pos5);
delay(20);}
break;
case 1: // відкриття клешні
for (pos2 = 90; pos2 >= 40; pos2 -= 1) {
servocatch.write(pos2);
delay(20);}
break;
case 2: // закриття клешні
for (pos2 = 40; pos2 <= 90; pos2 += 1) {
servocatch.write(pos2);
delay(20); }
break;
case 3: // поворот на ліво
for (pos1 = angle; pos1 <= 180; pos1 += 1) {
servopovorot.write(pos1);
delay(20);}
angle = 180;
break;
case 4: // поворот на право
for (pos1 = angle; pos1 >= 0; pos1 -= 1) {
servopovorot.write(pos1);
delay(20);}
angle = 0;
break;
case 5:// центр
if(angle > 90)// якщо крутить з лівого положення
{
for (pos1 = angle; pos1 >= 90; pos1 -= 1) { //right
servopovorot.write(pos1);
delay(20);}
angle = 90;
}
if (angle < 90) //якщо крутить з правого положення
{
for (pos1 = angle; pos1 <= 90; pos1 += 1) { //left
servopovorot.write(pos1);
delay(20);}
```

```
angle = 90;
}
servopovorot.write(90);
delay(3000);
break;
case 6: // вверх
for (pos3 = 60; pos3 <= 130; pos3 += 1) {
servoverh.write(pos3);
delay(20);}
break;
case 7: //вниз
for (pos3 = 130; pos3 >= 60; pos3 -= 1) {
servoverh.write(pos3);
delay(20);}
break;
case 8: //вперед
for (pos4 = 20; pos4 <= 60; pos4 += 1) {
servovpered.write(pos4);
delay(20);}
break;
case 9: // назад
for (pos4 = 60; pos4 >= 20; pos4 -= 1) {
servovpered.write(pos4);
delay(20);}
break;
}
}
}
```

ДОДАТОК В. ЛІСТИНГ ПРОГРАМИ ДЛЯ ГОЛОСОВОГО КЕРУВАННЯ

```
using System;
using System.Windows;
using System.Windows.Controls; //підключення бібліотек
using System.Timers;
using System.IO.Ports;
using Microsoft.Speech.Recognition;
using System.Windows.Automation.Peers;
using System.Windows.Automation.Provider;
namespace WpfApp_pics
{
    public partial class MainWindow : Window
    {
        System.Timers.Timer aTimer; //об'ява змінних для роботи з COM портом Arduino
        SerialPort currentPort;
        private delegate void updateDelegate(string txt); // об'ява делегата (ссылки на функцію) для виводу
текста з COM порта
        SpeechRecognitionEngine sre;
        static Label l; //об'ява змінних для роботи з COM портом Arduino
        static SerialPort myPort;
        public MainWindow()
        {
            InitializeComponent();
            this.Left = SystemParameters.PrimaryScreenWidth - this.Width; //об'ява розміру основного вік-
на
            this.Height = SystemParameters.PrimaryScreenHeight;
            Instance = this;
        }
        private void OnTimedEvent(object sender, ElapsedEventArgs e) // функція яка відповідає за роботу з
COM портом
        {
            // у разі не доступності ком порта воно вертає помилку, а не закриває програму
            try
            {
                if (!currentPort.IsOpen)
                {

```

```
        return;
    }
}
catch (Exception ex)
{
    bt_connect.Visibility = Visibility.Visible;
}
// обновляє текст мітки згідно інформації на ком порті
try
{
    currentPort.DiscardInBuffer(); // видаляє стару інформацію з буфера
    string strFromPort = currentPort.ReadLine(); // зчитує останнє значення
    lblPortData.Dispatcher.BeginInvoke(new updateDelegate(updateTextBox), strFromPort);
}
catch (Exception ex)
{
    lblPortData.Dispatcher.BeginInvoke(new updateDelegate(updateTextBox), ex.Message);
}
}
private void updateTextBox(string txt) // допоміжна функція для передачі даних з ком порта
{
    lblPortData.Content = txt;
}
bool FindPort() // пошук доступного ком порта
{
    bool ArduinoPortFound = false;
    try
    {
        string[] ports = SerialPort.GetPortNames();
        if (ports.Length != 0)
        foreach (string port in ports)
        {
            currentPort = new SerialPort(port, 9600);
            ArduinoPortFound = true;
        }
    }
    catch { }
    if (ArduinoPortFound == false) return false;
    else return true;
}
```

```

    }
    void PortConnect() // основна функція для підключення ком порту
    {
        System.Threading.Thread.Sleep(500); // затримка
        currentPort.BaudRate = 9600;
        currentPort.DtrEnable = true;
        currentPort.ReadTimeout = 1000;
        try
        {
            currentPort.Open();
        }
        catch { }
        aTimer = new System.Timers.Timer(1000);
        aTimer.Elapsed += OnTimedEvent;
        aTimer.AutoReset = true;
        aTimer.Enabled = true;
    }
    private void Window_Loaded(object sender, RoutedEventArgs e) // функція викликається при заван-
таженні сторінки
    {
        if (FindPort() == false) return;
        else bt_connect.Visibility = Visibility.Collapsed;
        PortConnect();
    }
    private void Bt_connect_Click(object sender, RoutedEventArgs e) //опрацьовувач події настикування
на кнопку коннект
    {
        bt_connect.Visibility = Visibility.Collapsed;
        PortConnect();
    }
    private void btn_Click(Button btn) // допоміжна функція для відправлення керуючих сигналів на
ком порт
    {
        if (currentPort == null) return;
        if (!currentPort.IsOpen) return;
        //Button btn = sender as Button;
        string str = btn.Name;
        if (str == "bt_test") currentPort.Write("10");
        if (str == "bt_open") currentPort.Write("1");
    }

```

```

        if (str == "bt_close") currentPort.Write("2");
        if (str == "bt_left") currentPort.Write("3");
        if (str == "bt_right") currentPort.Write("4");
        if (str == "bt_center") currentPort.Write("5");
        if (str == "bt_up") currentPort.Write("6");
        if (str == "bt_down") currentPort.Write("7");
        if (str == "bt_forward") currentPort.Write("8");
        if (str == "bt_rotate") currentPort.Write("9");
    }

```

```

    public static MainWindow Instance { get; private set; } // оголошення інтерфейсу форми

```

```

    static void sre_SpeechRecognized(object sender, SpeechRecognizedEventArgs e) // допоміжна функція

```

для передачі керуючих сигналів на ком порт

```

    {
        if (e.Result.Confidence > 0.7)
        {
            Button somebutton = MainWindow.Instance.bt_test; ;
            I.Content = e.Result.Text; // ступінь відповідності
            string str = I.Content.ToString();
            if (str == "тест") somebutton = MainWindow.Instance.bt_test; // myPort.Write("0");
            if (str == "открыть") somebutton = MainWindow.Instance.bt_open; //myPort.Write("1");
            if (str == "закрыть") somebutton = MainWindow.Instance.bt_close; // myPort.Write("2");
            if (str == "лево") somebutton = MainWindow.Instance.bt_left; // myPort.Write("3");
            if (str == "право") somebutton = MainWindow.Instance.bt_right; // myPort.Write("4");
            if (str == "центр") somebutton = MainWindow.Instance.bt_center; // myPort.Write("5");
            if (str == "вверх") somebutton = MainWindow.Instance.bt_up; // myPort.Write("6");
            if (str == "вниз") somebutton = MainWindow.Instance.bt_down; // myPort.Write("7");
            if (str == "вперед") somebutton = MainWindow.Instance.bt_forward; // myPort.Write("8");
            if (str == "назад") somebutton = MainWindow.Instance.bt_rotate; // myPort.Write("9");
            ButtonAutomationPeer peer = new ButtonAutomationPeer(somebutton);
            IInvokeProvider invokeProv = peer.GetPattern(PatternInterface.Invoke) as IInvokeProvider;
            invokeProv.Invoke();
        }
    }

    void my_Rec_setup(string language = "ru-ru") //основна функція для розпізнавання голосових ко-
манд
    {
        I = this.LabelVoice;
        myPort = this.currentPort;

        //System.Globalization.CultureInfo ci = new System.Globalization.CultureInfo("ru-ru");
    }

```

```

System.Globalization.CultureInfo ci = new System.Globalization.CultureInfo(language);
sre = new SpeechRecognitionEngine(ci);
sre.SetInputToDefaultAudioDevice();
sre.SpeechRecognized += new EventHandler<SpeechRecognizedEventArgs>(sre_SpeechRecognized);
//отримувач який ссилається на функцію для передачі керуючих команд
Choices numbers = new Choices(); //порівня записаних звуків з потрібними керуючими коман-
дами
if (language == "ru-ru")
    //numbers.Add(new string[] { "ноль", "один", "два", "три", "четыре", "пять", "шесть", "семь",
"восемь", "девять" });
    numbers.Add(new string[] { "тест", "открыть", "заккрыть", "лево", "право", "центр", "вверх",
"вниз", "вперед", "назад" });
if (language == "en-en")
    numbers.Add(new string[] { "zero", "one", "two", "three", "four", "five", "six", "seven", "eight",
"nine" });

GrammarBuilder gb = new GrammarBuilder(); //розпізнавання граматики
gb.Culture = ci;
gb.Append(numbers);
Grammar g = new Grammar(gb);
sre.LoadGrammar(g);
sre.RecognizeAsync(RecognizeMode.Multiple);
}
private void Bt_open_Click(object sender, RoutedEventArgs e) // опрацьовувач подій на натискання
кнопки
{
    image_open.Visibility = Visibility.Visible;
    image_close.Visibility = Visibility.Collapsed;
    btn_Click(sender as Button);
}
private void Bt_close_Click(object sender, RoutedEventArgs e)
{
    image_open.Visibility = Visibility.Collapsed;
    image_close.Visibility = Visibility.Visible;
    btn_Click(sender as Button);
}
private void Bt_rotate_Click(object sender, RoutedEventArgs e)
{
    btn_Click(sender as Button);
}

```

```
private void Bt_test_Click(object sender, RoutedEventArgs e)
{
    btn_Click(sender as Button);
}
private void Bt_up_Click(object sender, RoutedEventArgs e)
{
    image_up.Visibility = Visibility.Visible;
    image_forward.Visibility = Visibility.Collapsed;
    image_down.Visibility = Visibility.Collapsed;
    btn_Click(sender as Button);
}
private void Bt_forward_Click(object sender, RoutedEventArgs e)
{
    image_up.Visibility = Visibility.Collapsed;
    image_forward.Visibility = Visibility.Visible;
    image_down.Visibility = Visibility.Collapsed;
    btn_Click(sender as Button);
}
private void Bt_down_Click(object sender, RoutedEventArgs e)
{
    image_up.Visibility = Visibility.Collapsed;
    image_forward.Visibility = Visibility.Collapsed;
    image_down.Visibility = Visibility.Visible;
    btn_Click(sender as Button);
}
private void Bt_left_Click(object sender, RoutedEventArgs e)
{
    image_left.Visibility = Visibility.Visible;
    image_center.Visibility = Visibility.Collapsed;
    image_right.Visibility = Visibility.Collapsed;
    btn_Click(sender as Button);
}
private void Bt_center_Click(object sender, RoutedEventArgs e)
{
    image_left.Visibility = Visibility.Collapsed;
    image_center.Visibility = Visibility.Visible;
    image_right.Visibility = Visibility.Collapsed;
    btn_Click(sender as Button);
}
```

```
private void Bt_right_Click(object sender, RoutedEventArgs e)
{
    image_left.Visibility = Visibility.Collapsed;
    image_center.Visibility = Visibility.Collapsed;
    image_right.Visibility = Visibility.Visible;
    btn_Click(sender as Button);
}
}
}
```

Цабай Д.В. РК-51, 2019

ДОДАТОК Г. ЛІСТИНГ ПРОГРАМИ ДЛЯ ІНТЕРФЕЙСУ

```
<Window x:Class="WpfApp_pics.MainWindow"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:local="clr-namespace:WpfApp_pics"
mc:Ignorable="d"
Title="MainWindow" Height="auto" Width="480" WindowStartupLocation="Manual" Top="0"
Loaded="Window_Loaded">
<Window.Resources>
</Window.Resources>
<Grid VerticalAlignment="Center">
<Grid.RowDefinitions>
<RowDefinition Height="auto"></RowDefinition>
<RowDefinition Height="auto"></RowDefinition>
<RowDefinition Height="auto"></RowDefinition>
</Grid.RowDefinitions>
<StackPanel Grid.Row="0" Orientation="Horizontal" HorizontalAlignment="Center" Height="auto"
VerticalAlignment="Bottom" Margin="10">
<StackPanel>
<Image Name="image_open" Source="Res/robot WPF open.jpg" Visibility="Collapsed"/>
<Image Name="image_close" Source="Res/robot WPF close.jpg" Visibility="Visible"/>
</StackPanel>
<StackPanel Orientation="Vertical" VerticalAlignment="Center" HorizontalAlignment="Left">
<Button Name="bt_open" Content="Открыть" Width="75" Margin="5" Click="Bt_open_Click" />
<Button Name="bt_close" Content="Закрыть" Width="75" Margin="5" Click="Bt_close_Click"/>
</StackPanel>
</StackPanel>
<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Center" Height="auto"
VerticalAlignment="Center">
<StackPanel HorizontalAlignment="Left" Height="auto" VerticalAlignment="Center">
<Button Name="bt_test" Height="auto" Width="auto" Margin="10" Padding="5"
Click="Bt_test_Click">
<Button.Content>
<StackPanel Orientation="Vertical">
<TextBlock Text="T"/>
<TextBlock Text="e"/>
<TextBlock Text="c"/>
<TextBlock Text="t"/>
</StackPanel>
</Button.Content>
</Button>
</StackPanel>
<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Height="auto"
VerticalAlignment="Center">
<StackPanel HorizontalAlignment="Left" VerticalAlignment="Center" Height="auto" Width="75">
<Button Name="bt_up" Content="Вверх" Margin="5" Click="Bt_up_Click"/>
<Button Name="bt_forward" Content="Вперед" Margin="5" Click="Bt_forward_Click"/>
<Button Name="bt_rotate" Content="Назад" Margin="5" Click="Bt_rotate_Click"/>
<Button Name="bt_down" Content="Вниз" Margin="5" Click="Bt_down_Click"/>
</StackPanel>
<Grid HorizontalAlignment="Center" Height="auto" VerticalAlignment="Center" MinHeight="369"
MinWidth="341">
```

```

        <Image Name="image_up" Source="Res/robot WPF 3.jpg" Visibility="Collapsed" Stretch="None"
VerticalAlignment="Bottom" HorizontalAlignment="Left"/>
        <Image Name="image_forward" Source="Res/robot WPF 2.jpg" Visibility="Collapsed"
Stretch="None" VerticalAlignment="Bottom" HorizontalAlignment="Left"/>
        <Image Name="image_rotate" Source="Res/robot WPF 0.jpg" Visibility="Collapsed"
Stretch="None" VerticalAlignment="Bottom" HorizontalAlignment="Left"/>
        <Image Name="image_down" Source="Res/robot WPF 1.jpg" Visibility="Visible" Stretch="None"
VerticalAlignment="Bottom" HorizontalAlignment="Left" />
        <Label x:Name="lblPortData" FontSize="24" HorizontalAlignment="Right"
VerticalAlignment="Bottom" Width="auto" Visibility="Collapsed">No data</Label>
        <Label x:Name="LabelVoice" FontSize="24" HorizontalAlignment="Right"
VerticalAlignment="Bottom" Width="auto" Visibility="Visible">Мой голос</Label>
        <Button x:Name="bt_connect" Content="Подключить" HorizontalAlignment="Right"
VerticalAlignment="Bottom" Margin="0,0,0,-20" Width="75" Click="Bt_connect_Click"/>
    </Grid>
</StackPanel>
</StackPanel>
<StackPanel Grid.Row="2" Orientation="Vertical" HorizontalAlignment="Center" Height="auto"
VerticalAlignment="Top">
    <StackPanel Margin="10">
        <Image Name="image_left" Source="Res/robot WPF left.gif" Visibility="Collapsed"
Stretch="None"/>
        <Image Name="image_center" Source="Res/robot WPF centr.gif" Visibility="Visible"
Stretch="None"/>
        <Image Name="image_right" Source="Res/robot WPF right.gif" Visibility="Collapsed"
Stretch="None"/>
    </StackPanel>
    <StackPanel Grid.Row="2" Orientation="Horizontal" HorizontalAlignment="Center" Height="auto"
VerticalAlignment="Top">
        <Button Name="bt_left" Content="Лево" HorizontalAlignment="Left" Width="75" Margin="2"
Click="Bt_left_Click"/>
        <Button Name="bt_center" Content="Центр" HorizontalAlignment="Center" Width="75"
Margin="2" Click="Bt_center_Click"/>
        <Button Name="bt_right" Content="Право" HorizontalAlignment="Right" Width="75" Margin="2"
Click="Bt_right_Click"/>
    </StackPanel>
</StackPanel>
</Grid>
</Window>

```