

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: *Компютеризована двоманіпуляторна система безконтактного асистування монтажних процесів*

Виконав: студент 4 курсу, групи СІс-41
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

Шустак Д. С.

(прізвище та ініціали)

Керівник

(підпис)

Стрембіцький М. О.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Тили Є. В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль
2026

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Осухівська Г.М.
(підпис) (прізвище та ініціали)
«25» квітня 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

студента Шустака Давида Степановича
(прізвище, ім'я, по батькові)

1. Тема роботи Компютеризована двоманіпуляторна система безконтактного асистування монтажних процесів

Керівник роботи _____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від
« 24 » квітня 2026 року № 4/9-18_

2. Термін подання студентом завершеної роботи 18.06.2026

3. Вихідні дані до роботи Технічне завдання

4. Зміст роботи

Вступ. 1. Аналіз технічного завдання

2. Проектування структурної та електричної схем

3. Практична частина

4. Безпека життєдіяльності, основи охорони праці.

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Схема електрично-принципова ЕЗ (А1)

2. Схума функціональна Е2 (А1)

3. Блок-схема алгоритму роботи

4.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>			

7. Дата видачі завдання 25.04.2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Розробка технічного завдання</i>	26.01 – 02.02	
2.	<i>Робота над першим розділом «Аналіз технічного завдання»</i>	03.02 – 15.02	
3.	<i>Робота над другим розділом «Проектування структурної та електричної схем»</i>	20.04 – 23.04	
4.	<i>Робота над третім розділом «Практична частина»</i>	23.04 – 01.05	
5.	<i>Робота над четвертим розділом «Безпека життєдіяльності, основи охорони праці»</i>	01.05 – 20.05	
6.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	20.05 – 7.06	
7.	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	8.06 – 14.06	
8.	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	15.06 – 21.06	
9.	<i>Захист кваліфікаційної роботи бакалавра</i>		

Студент

(підпис)

Шустак Д. С.

(прізвище та ініціали)

Керівник роботи

(підпис)

Стрембіцький М. О.

(прізвище та ініціали)

АНОТАЦІЯ

Шустак Д. С. Комп'ютеризована двоманіпуляторна система безконтактного асистування монтажних процесів: робота на здобуття кваліфікаційного ступеня бакалавра: спец. 123 — комп'ютерна інженерія. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2026.

Ключові слова: жестове керування, комп'ютерний зір, маніпулятор, сервопривід, безконтактний інтерфейс, мікроконтролер, програмування жестів.

Кваліфікаційна робота присвячена розробці та реалізації комп'ютеризованої двоманіпуляторна система безконтактного асистування монтажних процесів. Для досягнення цієї мети використовується мікроконтролер, сервоприводи та камера технічного зору з алгоритмами розпізнавання жестів. Завдяки цій системі працівник отримує додаткову механічну кінцівку, здатну виконувати допоміжні операції без необхідності фізичного контакту з панеллю керування, що значно підвищує продуктивність та зручність праці. Дана система розпізнає жести оператора в реальному часі та перетворює їх у команди для маніпулятора, зокрема переміщення у різних напрямках, підйом та опускання, затискання і відпускання елементів монтажу. В разі необхідності виконання нестандартних операцій система передбачає режим програмування, в якому працівник може прив'язати положення маніпулятора до обраного жесту, що дозволяє адаптувати систему під конкретні виробничі задачі. Алгоритми розпізнавання побудовані з урахуванням вимог до надійності та безпеки праці. Розроблена система забезпечує ефективне асистування оператору під час монтажних процесів завдяки інтуїтивному безконтактному керуванню та функції програмування жестових команд.

ANNOTATION

Shustak D. S. Computerized Dual-Manipulator System for Contactless Assistance in Assembly Processes: thesis for a bachelor's degree: spec. 123 — Computer Engineering. Ternopil: Ternopil Ivan Puluj National Technical University, 2026.

Keywords: gesture control, computer vision, manipulator, servo drive, contactless interface, microcontroller, gesture programming.

The qualification work is devoted to the development and implementation of a computerized Dual-Manipulator System for Contactless Assistance in Assembly Processes. To achieve this goal, a microcontroller, servo drives, and a machine vision camera with gesture recognition algorithms are used. This system provides the operator with an additional mechanical limb capable of performing auxiliary operations without physical contact with the control panel, which significantly increases productivity and ease of work. The system recognizes the operator's gestures in real time and converts them into commands for the manipulator, including movement in various directions, raising and lowering, gripping and releasing assembly elements. When non-standard operations are required, the system provides a programming mode in which the operator can assign a manipulator position to a selected gesture, allowing the system to be adapted to specific production tasks. The recognition algorithms are built in accordance with reliability and occupational safety requirements. The developed system ensures effective assistance to the operator during assembly processes through intuitive contactless control and the gesture command programming feature.

ЗМІСТ

СПИСОК СКОРОЧЕНЬ.....	8
ВСТУП	9
1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	11
1.1 Аналіз вимог до комп'ютеризованої системи	11
1.2 Огляд існуючих аналогів систем допоміжних маніпуляторів.....	14
1.3 Постановка задачі та визначення технічних вимог до системи	17
2 ПРОЕКТУВАННЯ СТРУКТУРНОЇ ТА ЕЛЕКТРИЧНОЇ СХЕМ	19
2.1 Розробка узагальненої структури системи	19
2.1.1 Мікроконтролер.....	20
2.1.2 Обчислювальний модуль.....	23
2.1.3 Камера технічного зору	25
2.1.4 Сервоприводи	26
2.2 Проєктування електричної принципової схеми.....	28
2.3 Проєктування програмного забезпечення системи	32
2.3.1 Вибір алгоритму керування маніпулятором	32
2.3.2 Розробка схеми розподілення портів GPIO мікроконтролера.....	33
2.3.3 Опис електричної принципової схеми	34
3 ПРАКТИЧНА ЧАСТИНА	36
3.1 Розробка програмного забезпечення системи.....	36
3.1.1 Головна функція програми.....	38
3.1.2 Функція керування сервоприводами.....	39
3.1.3 Функція режиму програмуван	40
3.2 Програма розпізнавання жестів на Raspberry Pi	41
3.3 Інтеграція механічних елементів з мікроконтролером	43

					КС КРБ 123.275.00.00 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата	Компютеризована двоманіпуляторна система безконтактного асистування монтажних				Літ.	Арк.	Аркушів
Розроб.		Шустак Д. С.									
Перевір.		Стрембіцький								6	
Реценз.									ТНТУ, каф. КС, гр. СІс-41		
Н. Контр.		Тиш Є. В.									
Затверд.		Осухівська Г.М.									

3.3.1 Живлення та керування сервоприводами	43
3.3.2 Підключення EEPROM-пам'яті	43
4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	44
4.1 Моделювання та прогнозування небезпечних ситуацій	44
4.2 Особливості заходів захисту безпеки роботи.....	46
4.3 Організація безпеки роботи електроустановок.....	47
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	52
Додаток А Технічне завдання	55
Додаток Б Перелік елементів	74
Додаток В Код програми	76

					КС КРБ 123.275.00.00 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК СКОРОЧЕНЬ

АЦП – Аналогово-цифровий перетворювач

GPIO – General Purpose Input/Output

HAL – Hardware Abstraction Layer

HDMI – High Definition Multimedia Interface

I2C – Inter-Integrated Circuit

IDE – Integrated Development Environment

МК – Мікроконтролер

PWM – Pulse Width Modulation

RAM – Random Access Memory

RPi – Raspberry Pi

SCL – Serial Clock Line

SDA – Serial Data Line

SPI – Serial Peripheral Interface

SRAM – Static Random Access Memory

STM32 – мікроконтролер серії STM32 компанії STMicroelectronics

SWD – Serial Wire Debug

TX/RX – Transmit/Receive

UART – Universal Asynchronous Receiver-Transmitter

USB – Universal Serial Bus

ШИМ – Широтно-імпульсна модуляція

EEPROM – Electrically Erasable Programmable Read-Only Memory

CSI – Camera Serial Interface

					КС КРБ 123.275.00.00 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У сучасному виробництві монтажні роботи займають одне з ключових місць у технологічних процесах різних галузей промисловості. Незважаючи на значний розвиток автоматизації, велика кількість монтажних операцій досі виконується вручну, що вимагає від працівника одночасного утримання деталей, інструментів та виконання точних маніпуляцій. У таких умовах гостро постає потреба в допоміжних засобах, здатних звільнити руки оператора без ускладнення робочого процесу.

Існуючі на ринку рішення, зокрема механічні тримачі, лещата та промислові маніпулятори, мають суттєві обмеження. Традиційні механічні пристосування є статичними і не здатні реагувати на динамічні потреби монтажника. Промислові роботизовані маніпулятори, такі як Universal Robots UR-серія чи Fanuc CRX, хоча й забезпечують широкий функціонал, є надзвичайно дорогими, громіздкими та вимагають складного програмування спеціалістами. Більш доступні рішення по типу автоматизованих затискачів або педального керування не вирішують проблеми повністю, адже обмежують мобільність або відволікають увагу оператора.

Перспективним напрямом вирішення цієї проблеми є розробка інтелектуальних систем з безконтактним керуванням на основі комп'ютерного зору. Такий підхід дозволяє зберегти природність рухів оператора, не накладаючи жодних фізичних обмежень, і водночас забезпечити ефективне управління допоміжним маніпулятором у реальному часі.

Метою кваліфікаційної роботи є розробка та реалізація комп'ютеризованої інтелектуальної двоманіпуляторної системи для асистування в монтажних процесах з безконтактним інтерфейсом керування. Для досягнення цієї мети використовується мікроконтролер, сервоприводи та камера технічного зору з алгоритмами розпізнавання жестів. Система надає оператору додаткову механічну кінцівку, якою він керує за допомогою жестів рук, а також підтримує режим програмування, що дозволяє прив'язати

					КС КРБ 123.275.00.00 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

довільне положення маніпулятора до обраного жесту відповідно до конкретних виробничих потреб.

					КС КРБ 123.275.00.00 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Аналіз вимог до комп'ютеризованої системи

У сучасному виробництві монтажні роботи є невід'ємною частиною технологічних процесів у різних галузях промисловості — від електроніки до будівництва. Значна частина таких операцій передбачає одночасне утримання деталей, підключення проводів, кріплення елементів конструкцій тощо. Виконання подібних завдань однією особою без додаткових засобів є фізично складним і часто призводить до помилок, травм або зниження якості монтажу.

Традиційні допоміжні засоби, такі як механічні затискачі, лещата або підставки, є статичними та не здатні адаптуватися до динамічних потреб оператора в процесі роботи. Їх використання вимагає ручного налаштування і переналаштування, що суттєво уповільнює робочий процес. Водночас промислові роботизовані маніпулятори, незважаючи на широкий функціонал, є дорогими у придбанні та обслуговуванні, а також вимагають складного програмування.

Оптимальним рішенням є розробка інтелектуальної допоміжної системи механічної «третьої руки», якою оператор керує дистанційно за допомогою жестів, не відволікаючись від основного процесу монтажу. Для цього система повинна відповідати наступним вимогам:

- розпізнавання жестів оператора в режимі реального часу за допомогою камери технічного зору;
- виконання базових команд керування маніпулятором: переміщення вперед/назад, вліво/вправо, вгору/вниз, затискання та відпускання захвату;

					КС КРБ 123.275.00.00 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

- підтримка режиму програмування, в якому оператор може самостійно прив'язати положення маніпулятора до обраного жесту;
- висока надійність розпізнавання жестів в умовах виробничого середовища;
- забезпечення безпеки оператора та оточуючих під час роботи системи;
- простота та інтуїтивність керування без необхідності спеціальної підготовки.

Наведені вимоги були сформовані з урахуванням реальних умов монтажних робіт та вимог до ергономіки й безпеки праці.

Сучасні системи розпізнавання жестів активно застосовуються в різних сферах медицині, розумному домі, ігровій індустрії. Одним із поширених рішень є контролер Leap Motion, який дозволяє відстежувати рухи рук у тривимірному просторі з високою точністю. Приклад пристрою показано на рисунку 1.1.



Рисунок 1.1 – Контролер жестового керування Leap Motion

Проте подібні спеціалізовані пристрої мають обмежений кут огляду та є відносно дорогими. Альтернативою слугують стандартні камери в поєднанні з бібліотеками комп'ютерного зору, зокрема MediaPipe від Google, яка

					КС КРБ 123.275.00.00 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечує розпізнавання 21 ключової точки кисті руки в режимі реального часу, як зображено на рисунку 1.2.

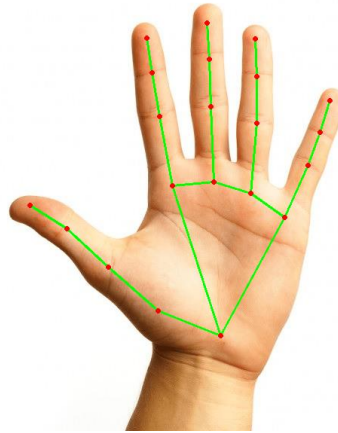


Рисунок 1.2 – Розпізнавання ключових точок кисті руки за допомогою MediaPipe

Робота системи виконується в декілька етапів:

- камера технічного зору фіксує зображення рук оператора в реальному часі;
- алгоритм розпізнавання аналізує положення та конфігурацію пальців і кисті;
- розпізнаний жест перетворюється у відповідну команду для мікроконтролера;
- мікроконтролер формує керуючі сигнали для сервоприводів маніпулятора, які виконують задану дію.

Головною метою розробки даної системи є максимальне спрощення процесу монтажних робіт шляхом надання оператору інтелектуального безконтактного інструменту керування допоміжним маніпулятором.

1.2 Огляд існуючих аналогів систем допоміжних маніпуляторів

На сьогоднішній день ринок пропонує кілька категорій пристроїв, що частково вирішують проблему асистування при монтажних роботах. Кожна з них має свої принципи роботи, переваги та суттєві недоліки, які й зумовили необхідність розробки нової системи.

Найпростішими та найпоширенішими засобами є механічні тримачі та затискачі — так звані «треті руки» (Third Hand Tool). Вони являють собою конструкцію зі штативом, гнучкими важелями та затискачами-«крокодилами», як зображено на рисунку 1.3.



Рисунок 1.3 – Механічний тримач «третя рука» Third Hand Tool

Принцип їх роботи є гранично простим: оператор вручну встановлює затискач у потрібне положення та фіксує деталь. Проте такі пристрої є повністю статичними — вони не здатні самостійно переміщуватись, змінювати кут захвату або реагувати на команди оператора. Для будь-якого переналаштування необхідно відкласти інструмент і вручну відрегулювати положення, що суттєво уповільнює роботу.

					КС КРБ 123.275.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Більш технологічним рішенням є роботизовані маніпулятори з програмним керуванням, зокрема серія Universal Robots UR3/UR5, що широко використовується у промисловості, як показано на рисунку 1.4.



Рисунок 1.4 – Промисловий маніпулятор Universal Robots UR3

Такі системи забезпечують широкий діапазон рухів, високу точність та можливість програмування складних траєкторій. Однак їх використання у контексті допоміжного монтажного інструменту має критичні недоліки:

- надзвичайно висока вартість обладнання та обслуговування;
- необхідність спеціалізованого програмування та налаштування фахівцями;
- великі габарити, що унеможливають використання на звичайному робочому місці монтажника;
- відсутність інтуїтивного безконтактного інтерфейсу керування в реальному часі.

Окрему категорію складають системи з жестовим керуванням, що використовують спеціалізовані сенсори. Одним із найвідоміших представників є контролер Leap Motion Controller, зображений на рисунку 1.5.

					КС КРБ 123.275.00.00 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.5 – Контролер жестового керування Leap Motion Controller

Leap Motion забезпечує високоточне відстеження рухів рук у тривимірному просторі та може використовуватись для керування зовнішніми пристроями. Проте вузький кут огляду сенсора, чутливість до освітлення та висока вартість обмежують його практичне застосування у виробничих умовах.

Порівняльний аналіз розглянутих аналогів наведено у таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз існуючих аналогів

Критерій	Механічний тримач	Промисловий маніпулятор	Leap Motion + маніпулятор	Розроблювана система
Безконтактне керування	Ні	Ні	Так	Так
Режим програмування жестів	Ні	Частково	Ні	Так
Вартість	Низька	Дуже висока	Висока	Низька
Простота використання	Висока	Низька	Середня	Висока
Мобільність	Середня	Низька	Середня	Висока

Підводячи підсумок, можна стверджувати, що жоден із розглянутих аналогів не поєднує в собі одночасно низьку вартість, інтуїтивне безконтактне керування та можливість гнучкого програмування жестових команд під

конкретні виробничі задачі. Саме ці переваги і покликана забезпечити розроблювана комп'ютеризована інтелектуальна двоманіпуляторна система.

1.3 Постановка задачі та визначення технічних вимог до системи

На основі проведеного аналізу вимог та огляду існуючих аналогів сформульовано основну задачу кваліфікаційної роботи: розробити та реалізувати комп'ютеризовану інтелектуальну двоманіпуляторну систему для асистування в монтажних процесах, яка керується за допомогою безконтактного жестового інтерфейсу та є доступною за вартістю і простою у використанні без спеціальної підготовки оператора.

Для досягнення поставленої задачі система повинна відповідати наступним технічним вимогам.

Вимоги до апаратної частини:

- конструкція маніпулятора повинна забезпечувати не менше чотирьох ступенів свободи для виконання повного набору монтажних операцій;
- сервоприводи повинні забезпечувати достатній крутний момент для утримання проводів, дрібних деталей та електронних компонентів;
- захватний механізм повинен надійно утримувати елементи монтажу без їх пошкодження;

камера технічного зору повинна забезпечувати роздільну здатність не менше 720p та частоту кадрів не менше 30 кадрів/с для стабільного розпізнавання жестів;

- система повинна функціонувати автономно на основі мікроконтролера без потреби у постійному підключенні до персонального комп'ютера.

Вимоги до програмної частини:

					КС КРБ 123.275.00.00 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

- алгоритм розпізнавання жестів повинен працювати в режимі реального часу із затримкою реакції не більше 200 мс;
- система повинна коректно розпізнавати не менше восьми різних жестових команд;
- режим програмування повинен дозволяти оператору самостійно прив'язати довільне положення маніпулятора до обраного жесту без втручання розробника;
- програмне забезпечення повинно зберігати запрограмовані жестові команди після вимкнення живлення системи;
- алгоритм повинен забезпечувати стабільне розпізнавання жестів за різного освітлення робочого місця.

Вимоги до надійності та безпеки:

- система повинна коректно обробляти ситуації, коли жест не розпізнано, та переходити у стан очікування без виконання непередбачуваних рухів;
- у разі втрати зображення з камери маніпулятор повинен зупинитися та утримувати поточне положення;

конструкція повинна виключати можливість травмування оператора під час роботи системи

Дотримання наведених технічних вимог є основою для проектування структурної та електричної схем системи, а також розробки програмного забезпечення, що здійснюється в наступних розділах кваліфікаційної роботи.

					КС КРБ 123.275.00.00 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЕКТУВАННЯ СТРУКТУРНОЇ ТА ЕЛЕКТРИЧНОЇ СХЕМ

2.1 Розробка узагальненої структури системи

На рисунку 2.1 зображена структурна схема комп'ютеризованої інтелектуальної двоманіпуляторної системи для асистування в монтажних процесах з безконтактним інтерфейсом керування.

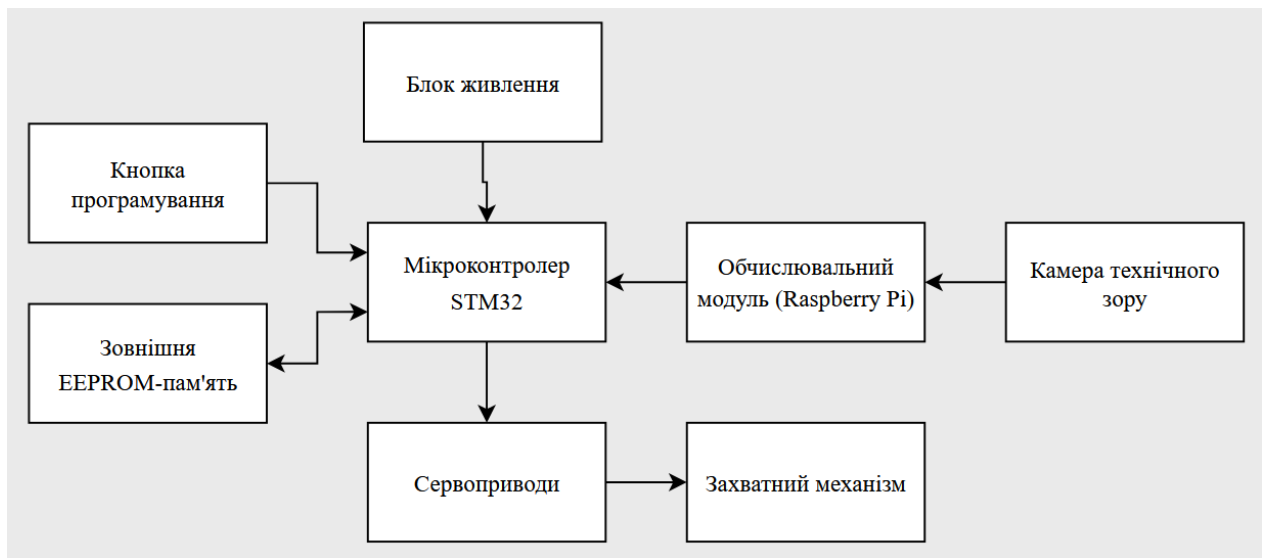


Рисунок 2.1 – Структурна схема двоманіпуляторної системи з безконтактним керуванням

Дана система складається з наступних складових:

- камера технічного зору — основний сенсор введення інформації. Здійснює захоплення відеопотоку з робочої зони оператора та передає зображення на обчислювальний модуль для подальшого аналізу;
- обчислювальний модуль (Raspberry Pi) — виконує обробку відеопотоку, запускає алгоритм розпізнавання жестів на основі бібліотеки MediaPipe, визначає тип жесту та формує відповідну команду керування, яку передає до мікроконтролера через інтерфейс UART;

- мікроконтролер STM32F103C8T6 — є центральним керуючим елементом апаратної частини системи. Отримує команди від обчислювального модуля через UART, формує керуючі ШІМ-сигнали для сервоприводів маніпулятора та керує режимом програмування жестових команд;
- сервоприводи — виконавчі механізми системи, що забезпечують переміщення ланок маніпулятора у відповідних напрямках, підйом та опускання, а також керування захватом;
- захватний механізм — кінцева ланка маніпулятора, призначена для затискання та утримання елементів монтажу, зокрема проводів та дрібних деталей;
- кнопка програмування — апаратний елемент, що ініціює режим програмування жестових команд. При першому натисканні система переходить у режим запису положення маніпулятора, при повторному — зберігає жест, прив'язаний до цього положення;
- зовнішня EEPROM-пам'ять — підключається до мікроконтролера через інтерфейс I2C та призначена для енергонезалежного зберігання запрограмованих жестових команд і відповідних положень маніпулятора після вимкнення живлення системи;
- блок живлення — забезпечує подачу стабілізованої напруги 5 В та 3.3 В на всі компоненти системи: мікроконтролер, сервоприводи та периферійні пристрої. Обґрунтування вибору апаратного забезпечення системи

2.1.1 Мікроконтролер

Для керування сервоприводами маніпулятора та обробки команд було обрано мікроконтролер STM32F103C8T6, широковідомий у середовищі розробників під назвою Blue Pill. Дана плата зображена на рисунку 2.2.

					КС КРБ 123.275.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

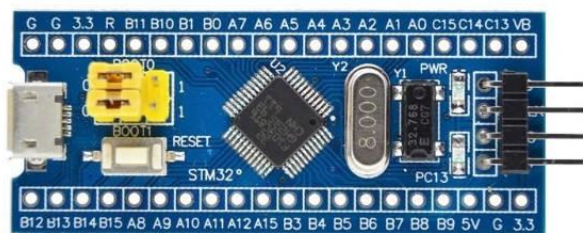


Рисунок 2.2 – Мікроконтролер STM32F103C8T6 (Blue Pill)

Вибір саме цього мікроконтролера обумовлений рядом суттєвих переваг над 8-бітними аналогами. По-перше, 32-бітне ядро ARM Cortex-M3 з тактовою частотою 72 МГц забезпечує значно вищу обчислювальну потужність, що дозволяє опрацьовувати команди керування маніпулятором у реальному часі з мінімальною затримкою. По-друге, 16-бітні апаратні таймери з підтримкою ШІМ забезпечують точне та стабільне керування сервоприводами, що є критично важливим для точного позиціонування ланок маніпулятора. По-третє, наявність апаратного інтерфейсу UART дозволяє організувати надійний зв'язок з обчислювальним модулем Raspberry Pi, а інтерфейс I2C — підключення зовнішньої EEPROM-пам'яті для зберігання запрограмованих жестових команд.

Плата містить всі необхідні елементи для реалізації системи:

- 32-бітне ядро ARM Cortex-M3 з тактовою частотою 72 МГц;
- 37 портів GPIO, з яких 12 підтримують апаратний ШІМ через таймери;
- робоча напруга 2.0–3.6 В із вбудованим стабілізатором;
- два 12-бітних АЦП для зчитування аналогових сигналів;
- апаратні інтерфейси: до 3× UART, 2× I2C, 2× SPI, 1× USB, 1× CAN;
- роз'єм micro USB для живлення та прошивки;
- підтримка програматорів ST-Link через інтерфейс SWD;
- сумісність із середовищами розробки Arduino IDE, STM32CubeIDE та PlatformIO.

На рисунку 2.2.1 Зображено умовне графічне позначення на функціональній схемі.

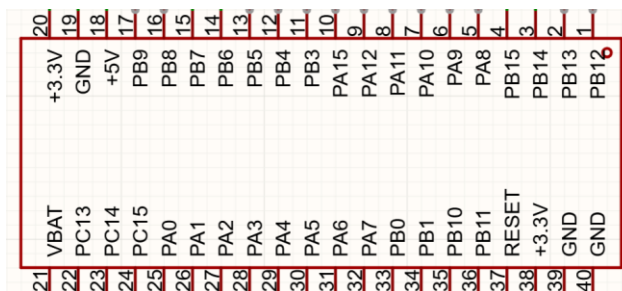


Рисунок 2.2.1 – Умовне графічне позначення на схемі

Компактний форм-фактор плати дозволяє інтегрувати мікроконтролер безпосередньо в корпус маніпуляторної системи без суттєвого збільшення її габаритів. Характеристики мікроконтролера наведено у таблиці 2.1.

Таблиця 2.1 – Характеристики мікроконтролера STM32F103C8T6

Параметр	Значення
Ядро	ARM 32-bit Cortex-M3
Тактова частота	72 МГц
Напруга живлення	2.0–3.6 В
Флеш-пам'ять	64 КБ
Оперативна пам'ять SRAM	20 КБ
Кількість портів GPIO	37
АЦП	2× 12-bit (10 каналів)
Інтерфейси	3× USART, 2× I2C, 2× SPI, 1× USB, 1× CAN
Діапазон температур	-40°C...+85°C
Підтримка ШІМ	12 каналів (через таймери TIM1–TIM4)

Завдяки наведеним характеристикам даний мікроконтролер повністю відповідає технічним вимогам системи, визначеним у підрозділі 1.3.

2.1.2 Обчислювальний модуль

Для обробки відеопотоку та виконання алгоритму розпізнавання жестів було обрано одноплатний комп'ютер Raspberry Pi 4 Model B, зображений на рисунку 2.3.



Рисунок 2.3 – Одноплатний комп'ютер Raspberry Pi 4 Model B

Raspberry Pi 4 забезпечує достатню обчислювальну потужність для запуску бібліотеки MediaPipe в режимі реального часу та передачі сформованих команд до мікроконтролера STM32 через інтерфейс UART.

На рисунку 2.3.1 Зображено умовне графічне позначення на функціональній схемі.

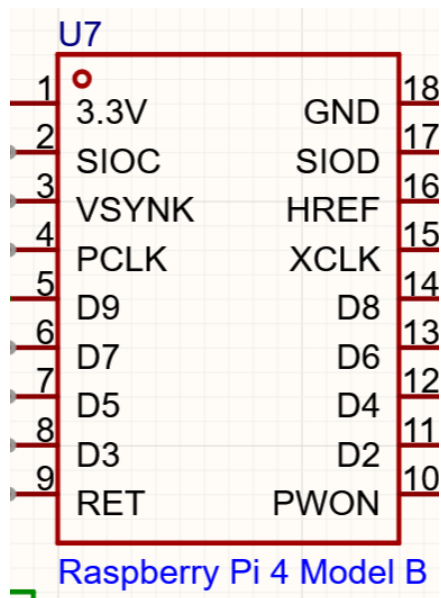


Рисунок 2.3.1 – Умовне графічне позначення на схемі

Наявність виділеного CSI-порту для підключення камери забезпечує передачу відеопотоку з мінімальною затримкою, що є критично важливим для розпізнавання жестів у режимі реального часу. Характеристики модуля наведено у таблиці 2.2.

Таблиця 2.2 – Характеристики Raspberry Pi 4 Model B

Параметр	Значення
Процесор	Broadcom BCM2711, 4-ядерний Cortex-A72, 1.8 ГГц
Оперативна пам'ять	4 ГБ LPDDR4
Операційна система	Raspberry Pi OS (Linux)
Інтерфейси	USB 3.0, HDMI, GPIO, UART, I2C, SPI
Підтримка камери	CSI-порт для Camera Module
Напруга живлення	5 В / 3 А

2.1.3 Камера технічного зору

Для захоплення зображення рук оператора обрано модуль камери Raspberry Pi Camera Module V2, що підключається безпосередньо до CSI-порту обчислювального модуля, як зображено на рисунку 2.4.



Рисунок 2.4 – Модуль камери Raspberry Pi Camera Module V2

Невеликі габарити модуля камери дозволяють розмістити її безпосередньо на робочому місці оператора під оптимальним кутом огляду без суттєвих обмежень простору. На рисунку 2.4.1 Зображено умовне графічне позначення на функціональній схемі.

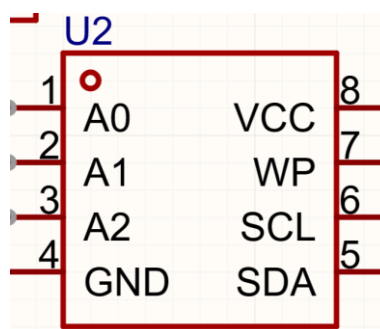


Рисунок 2.4.1 – Умовне графічне позначення на схем

Підключення через CSI-інтерфейс виключає необхідність використання драйверів та додаткового програмного налаштування, що

спрощує інтеграцію камери з обчислювальним модулем. Характеристики камери наведено у таблиці 2.3.

Таблиця 2.3 – Характеристики Raspberry Pi Camera Module V2

Параметр	Значення
Роздільна здатність	8 МП (3280 × 2464 пікселів)
Частота кадрів	до 30 кадрів/с при 1080p
Кут огляду	62.2° × 48.8°
Інтерфейс підключення	CSI
Мінімальна відстань фокусування	1 м

Дана камера повністю відповідає вимогам до роздільної здатності та частоти кадрів, визначеним у підрозділі 1.3, що забезпечує стабільне розпізнавання жестів в умовах монтажного виробництва.

2.1.4 Сервоприводи

Для реалізації рухів маніпулятора обрано сервоприводи MG996R, зображені на рисунку 2.5.



Рисунок 2.5 – Сервопривід MG996

На рисунку 2.5.1 Зображено умовне графічне позначення на функціональній схемі.

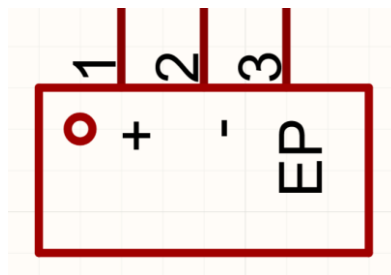


Рисунок 2.5.1 – Умовне графічне позначення на схемі

Наявність металевого редуктора суттєво підвищує довговічність сервоприводу порівняно з пластиковими аналогами, що є важливим фактором при інтенсивному використанні системи в умовах виробничого середовища. Характеристики сервоприводу наведено у таблиці 2.4.

Таблиця 2.4 – Характеристики сервоприводу MG996R

Параметр	Значення
Напруга живлення	4.8–7.2 В
Крутний момент	9.4 кг·см (при 4.8 В)
Швидкість обертання	0.17 с/60°
Кут повороту	180°
Тип редуктора	Металевий
Маса	55 г

Металевий редуктор забезпечує надійну роботу під навантаженням та довговічність конструкції. Для системи використовується чотири сервоприводи даної моделі, що керуються ШІМ-сигналами від таймерів

мікроконтролера STM32F103C8T6 та забезпечують переміщення маніпулятора по всіх необхідних осях і керування захватним механізмом.

Слід зазначити, що оскільки сервоприводи MG996R живляться від напруги 4.8–7.2 В, а мікроконтролер STM32 працює від 3.3 В, керуючі ШІМ-сигнали від STM32 є сумісними з логічним входом сервоприводів, проте живлення сервоприводів здійснюється окремо від основного блоку живлення системи, що виключає просадки напруги на мікроконтролері під час роботи приводів.

2.2 Проектування електричної принципової схеми

Електрична принципова схема двоманіпуляторної системи розроблена на основі мікроконтролера STM32F103C8T6 та відображає з'єднання всіх апаратних компонентів системи. Загальна схема зображена на рисунку 2.6.

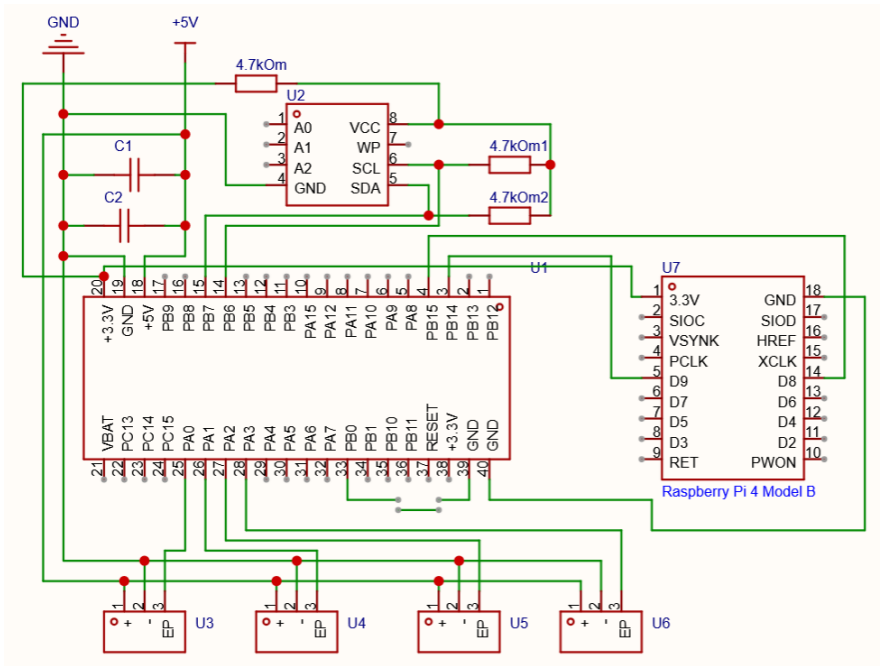


Рисунок 2.6 – Електрична принципова схема двоманіпуляторної системи

Схема складається з наступних функціональних вузлів: вузол живлення, вузол керування сервоприводами, вузол зв'язку з обчислювальним модулем, вузол програмування та вузол енергонезалежної пам'яті. Розглянемо кожен з них детальніше.

Живлення системи здійснюється від зовнішнього блоку живлення з вихідною напругою 5 В. Оскільки мікроконтролер STM32F103C8T6 працює від напруги 3.3 В, на платі встановлено вбудований лінійний стабілізатор напруги, який перетворює 5 В у 3.3 В для живлення самого МК та периферійних пристроїв з відповідним рівнем логіки. Сервоприводи MG996R підключаються безпосередньо до лінії 5 В окремим ланцюгом живлення, що виключає просадки напруги на мікроконтролері під час різких рухів маніпулятора. Між лініями живлення та землею встановлені розв'язувальні конденсатори номіналом 100 нФ та 10 мкФ для фільтрації імпульсних завад від сервоприводів.

Для керування чотирма сервоприводами MG996R використовуються чотири ШІМ-виходи мікроконтролера STM32F103C8T6, що формуються апаратними таймерами TIM2 та TIM3. Розпіновка підключення сервоприводів до МК наведена у таблиці 2.5.

Таблиця 2.5 – Підключення сервоприводів до мікроконтролера STM32

Сервопривід	Функція	Пін STM32	Таймер
Servo 1	Горизонтальне переміщення (ліво/право)	PA0	TIM2_CH1
Servo 2	Вертикальне переміщення (вгору/вниз)	PA1	TIM2_CH2
Servo 3	Переміщення вперед/назад	PA2	TIM2_CH3
Servo 4	Керування захватом	PA3	TIM2_CH4

Кожен сервопривід має три дроти: живлення (червоний, +5 В), земля (чорний/коричневий, GND) та сигнальний (жовтий/помаранчевий, ШІМ). Сигнальний дріт підключається безпосередньо до відповідного ШІМ-виходу STM32. Оскільки вихідний рівень логіки STM32 становить 3.3 В, а мінімальний поріг розпізнавання сигналу сервоприводом MG996R є 2.5 В, додаткові перетворювачі рівня не потрібні.

Зв'язок між мікроконтролером STM32 та обчислювальним модулем здійснюється через послідовний інтерфейс UART. Для цього використовується апаратний інтерфейс USART1 мікроконтролера. Підключення виконується наступним чином:

- пін TX (PA9) мікроконтролера STM32 підключається до пину RX (GPIO15) Raspberry Pi;
- пін RX (PA10) мікроконтролера STM32 підключається до пину TX (GPIO14) Raspberry Pi;
- лінії GND обох пристроїв об'єднуються.

Оскільки Raspberry Pi працює з логічними рівнями 3.3 В, а STM32F103C8T6 також використовує рівень 3.3 В на виходах UART, підключення є прямим без необхідності використання перетворювача рівнів. Швидкість передачі даних (baud rate) встановлюється на рівні 9600 бод, що є достатнім для передачі коротких командних пакетів у реальному часі.

Кнопка програмування підключається між піном PB0 мікроконтролера та землею. На піні PB0 активовано внутрішній підтягуючий резистор (pull-up) мікроконтролера, завдяки чому при відпущеній кнопці на вході присутній високий логічний рівень, а при натисканні — низький.

Для захисту від дребезгу контактів у програмному забезпеченні реалізована програмна затримка опитування кнопки тривалістю 50 мс.

Для збереження запрограмованих жестових команд після вимкнення живлення системи використовується мікросхема зовнішньої EEPROM-пам'яті

					КС КРБ 123.275.00.00 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

AT24C32, підключена до мікроконтролера через інтерфейс I2C. Підключення виконується наступним чином:

- пін SDA мікросхеми AT24C32 підключається до пину PB7 (I2C1_SDA) мікроконтролера;
- пін SCL мікросхеми AT24C32 підключається до пину PB6 (I2C1_SCL) мікроконтролера;
- лінії SDA та SCL підтягуються до напруги 3.3 В через резистори номіналом 4.7 кОм;
- живлення мікросхеми здійснюється від лінії 3.3 В.

Підключення всіх компонентів системи до мікроконтролера STM32F103C8T6 узагальнено у таблиці 2.6.

Таблиця 2.6 – Зведена таблиця підключення компонентів до STM32F103C8T6

Компонент	Пін компонента	Пін STM32	Інтерфейс
Raspberry Pi	TX (GPIO14)	PA10 (RX)	USART1
Raspberry Pi	RX (GPIO15)	PA9 (TX)	USART1
Servo 1	Сигнал	PA0	TIM2_CH1 (ШІМ)
Servo 2	Сигнал	PA1	TIM2_CH2 (ШІМ)
Servo 3	Сигнал	PA2	TIM2_CH3 (ШІМ)
Servo 4	Сигнал	PA3	TIM2_CH4 (ШІМ)
AT24C32 (EEPROM)	SDA	PB7	I2C1
AT24C32 (EEPROM)	SCL	PB6	I2C1
Кнопка програмування	—	PB0	GPIO (pull-up)

Розроблена електрична принципова схема є основою для подальшої розробки алгоритму та програмного забезпечення системи, що розглядається у наступному розділі кваліфікаційної роботи.

2.3 Проектування програмного забезпечення системи

2.3.1 Вибір алгоритму керування маніпулятором

Алгоритм керування маніпулятором полягає у безперервному зчитуванні команд, що надходять від обчислювального модуля Raspberry Pi через інтерфейс UART, та формуванні відповідних ШІМ-сигналів для керування сервоприводами. Кожна команда являє собою байтовий код, що відповідає певному жесту оператора. Отримавши код команди, мікроконтролер STM32 обчислює необхідний кут повороту кожного сервоприводу та формує відповідний ШІМ-сигнал.

Керування сервоприводом здійснюється за допомогою ШІМ-сигналу з періодом 20 мс (частота 50 Гц), де тривалість імпульсу визначає кут повороту сервоприводу відповідно до формули:

$$T = T_{\min} + (\alpha / 180^\circ) \cdot (T_{\max} - T_{\min}), \quad (2.3)$$

де:

T – тривалість керуючого імпульсу (мс),

$T_{\min}$ – мінімальна тривалість імпульсу (1 мс, відповідає куту 0°),

$T_{\max}$ – максимальна тривалість імпульсу (2 мс, відповідає куту 180°),

α – цільовий кут повороту сервоприводу (від 0° до 180°).

Для запису та зчитування положень маніпулятора у режимі програмування використовується зовнішня EEPROM-пам'ять AT24C32. Один запис містить код жесту та кути чотирьох сервоприводів і займає 5 байт пам'яті.

Описаний алгоритм виконується безперервно у головному циклі програми мікроконтролера з перевіркою наявності нових даних у буфері UART кожні 10 мс, що забезпечує своєчасну реакцію системи на команди оператора.

					КС КРБ 123.275.00.00 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

2.3.2 Розробка схеми розподілення портів GPIO мікроконтролера

Вибір портів GPIO мікроконтролера STM32F103C8T6 для вирішення задач проєкту у табличному вигляді зображено у таблиці 2.7.

Таблиця 2.7 – Розподілення портів GPIO мікроконтролера STM32F103C8T6

Змінна у програмі	Порт STM32	Підключений пристрій
SERVO_1_PWM	PA0	Servo 1 — горизонтальне переміщення
SERVO_2_PWM	PA1	Servo 2 — вертикальне переміщення
SERVO_3_PWM	PA2	Servo 3 — переміщення вперед/назад
SERVO_4_PWM	PA3	Servo 4 — керування захватом
UART_TX	PA9	TX → RX Raspberry Pi
UART_RX	PA10	RX → TX Raspberry Pi
I2C_SCL	PB6	SCL мікросхеми AT24C32
I2C_SDA	PB7	SDA мікросхеми AT24C32
BTN_PROGRAM	PB0	Кнопка режиму програмування
LED_STATUS	PC13	Індикація режиму програмування

Вбудований світлодіод PC13 використовується для індикації поточного стану системи: у штатному режимі роботи він вимкнений, а під час активного режиму програмування жестових команд — світиться, що дозволяє оператору чітко розуміти поточний стан системи без використання додаткових індикаторів.

У подальших версіях даної системи можливе розширення функціоналу шляхом підключення додаткових сервоприводів до вільних ШІМ-виходів таймерів TIM3 та TIM4, а також підключення дисплея для відображення поточного стану маніпулятора через вільні лінії інтерфейсу I2C.

2.3.3 Опис електричної принципової схеми

Електрична принципова схема комп'ютеризованої двоманіпуляторної системи розроблена на основі структурної схеми (див. рис. 2.1). Головним компонентом схеми є мікроконтролер STM32F103C8T6, який позначений на схемі під найменуванням U1. Напруга +5 В подається через зовнішній блок живлення, а напруга +3.3 В формується вбудованим стабілізатором напруги мікроконтролера. Застосування внутрішнього підтягуючого резистора дозволяє скоротити кількість зовнішніх елементів схеми та зменшити її загальну складність без втрати надійності роботи вузла введення. Вбудований індикаторний світлодіод, підключений до піну PC13, слугує засобом візуальної індикації поточного стану системи та не потребує додаткових зовнішніх елементів завдяки наявності відповідного обмежувального резистора безпосередньо на платі мікроконтролера.

На рисунку 2.8 зображено електричну принципову схему двоманіпуляторної системи з безконтактним інтерфейсом керування.

					КС КРБ 123.275.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

3 ПРАКТИЧНА ЧАСТИНА

3.1 Розробка програмного забезпечення системи

В основному циклі роботи системи передбачено два режими функціонування: штатний режим розпізнавання жестів та режим програмування нових жестових команд. Штатний режим виконується безперервно, доки система отримує коректний відеопотік з камери. Режим програмування активується натисканням апаратної кнопки на пін PB0 мікроконтролера STM32F103C8T6.

Програмне забезпечення системи складається з двох незалежних частин. Перша частина функціонує на обчислювальному модулі Raspberry Pi та відповідає за захоплення відеопотоку, обробку зображення і розпізнавання жестів оператора. Вона написана мовою програмування Python з використанням бібліотек MediaPipe та OpenCV. Друга частина функціонує безпосередньо на мікроконтролері STM32F103C8T6 та відповідає за прийом команд від Raspberry Pi через інтерфейс UART, формування керуючих ШІМ-сигналів для сервоприводів маніпулятора, а також за обробку режиму програмування жестових команд і роботу з зовнішньою EEPROM-пам'яттю AT24C32 через інтерфейс I2C. Програма мікроконтролера написана мовою C у середовищі розробки STM32CubeIDE з використанням бібліотеки апаратної абстракції HAL.

У штатному режимі роботи система виконує наступну послідовність дій: обчислювальний модуль Raspberry Pi захоплює кадр відеопотоку, передає його до бібліотеки MediaPipe для визначення координат 21 ключової точки кисті руки, класифікує жест та передає відповідний байтовий код команди до мікроконтролера через інтерфейс UART. Мікроконтролер STM32, отримавши

					КС КРБ 123.275.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	Практична частина		
Розроб.		Шустак Д. С.					
Перевір.		Стрембіцький					
Реценз.							
Н. Контр.		Тиш Є. В.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Аркушів
						36	
					ТНТУ, каф. КС, гр. СІс-41		

команду, виконує пошук відповідного жесту серед запрограмованих у EEPROM команд. Якщо жест знайдено серед запрограмованих — зчитуються збережені кути сервоприводів і маніпулятор відтворює відповідне положення. Якщо жест є стандартною командою переміщення — мікроконтролер обчислює цільові кути та формує відповідні ШІМ-сигнали через таймер TIM2. Весь цикл повторюється з частотою 30 кадрів/с, що забезпечує плавне та своєчасне реагування маніпулятора на команди оператора з затримкою не більше 200 мс.

У режимі програмування після першого натискання кнопки PB0 вбудований світлодіод PC13 вмикається, сигналізуючи оператору про активний режим запису. Система фіксує поточні значення кутів усіх чотирьох сервоприводів як цільове положення маніпулятора. Після повторного натискання кнопки система переходить в очікування нового жесту від оператора. Raspberry Pi розпізнає показаний жест та передає його байтовий код до STM32 через UART. Мікроконтролер записує пару «байтовий код жесту — кути чотирьох сервоприводів» у зовнішню EEPROM-пам'ять AT24C32 через інтерфейс I2C. Після успішного завершення запису світлодіод PC13 вимикається і система автоматично повертається у штатний режим роботи. Такий підхід дозволяє оператору самостійно розширювати набір жестових команд без втручання розробника та без перепрограмування мікроконтролера.

Весь програмний код повністю наведено у додатку А. В ньому вказано всі функції, параметри та структури даних. Також код детально описано коментарями мовою програмування C для кращого розуміння принципу роботи кожного фрагменту та спрощення подальшого доопрацювання і розширення функціоналу системи у наступних версіях.

Для коректної роботи програми мікроконтролера було розроблено функції, повний перелік та призначення яких наведено у таблиці 3.1.

					КС КРБ 123.275.00.00 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.1 – Перелік основних функцій програми мікроконтролера STM32

Функція	Призначення
void servo_set_angle(uint8_t ch, uint8_t angle)	Встановлення кута повороту сервоприводу на заданому каналі таймера TIM2
void uart_send_command(uint8_t cmd)	Передача байту команди від Raspberry Pi через UART
uint8_t uart_receive_command(void)	Прийом байту команди від Raspberry Pi через UART
void eeprom_write(uint8_t gesture, uint8_t s1, uint8_t s2, uint8_t s3, uint8_t s4)	Запис жесту та кутів сервоприводів у EEPROM через I2C
void eeprom_read(uint8_t gesture, uint8_t *angles)	Зчитування кутів сервоприводів для заданого жесту з EEPROM
void programming_mode_handler(void)	Обробка режиму програмування жестових команд
void execute_command(uint8_t cmd)	Виконання команди керування маніпулятором за кодом жесту
uint8_t button_read(void)	Зчитування стану кнопки програмування з піну PB0

3.1.1 Головна функція програми

Головною функцією програми мікроконтролера є функція `int main(void)`. Першими у даній функції виконуються наступні дії:

- ініціалізація таймера TIM2 для формування ШІМ-сигналів керування сервоприводами;
- ініціалізація всіх портів GPIO, зокрема піну PB0 для кнопки програмування та піну PC13 для індикаторного світлодіода;

- ініціалізація комунікаційних інтерфейсів USART1 для зв'язку з Raspberry Pi та I2C1 для роботи з EEPROM AT24C32;
- встановлення початкових значень кутів сервоприводів у нейтральне положення (90°);
- зчитування з EEPROM усіх раніше запрограмованих жестових команд до оперативної пам'яті для прискорення доступу під час роботи.

Після ініціалізації виконується основний цикл while(1), в якому здійснюється безперервна обробка команд від Raspberry Pi та опитування кнопки програмування. Умова циклу та структура зображені на рисунку 3.1.

```
c
/* USER CODE BEGIN WHILE */
while (1)
{
    if (button_read()) {
        programming_mode_handler();
    }
    uint8_t cmd = uart_receive_command();
    if (cmd != 0xFF) {
        execute_command(cmd);
    }
    HAL_Delay(10);
/* USER CODE END WHILE */
}
```

Рисунок 3.1 - Основний цикл програми мікроконтролера STM32

3.1.2 Функція керування сервоприводами

Функція керування сервоприводами servo_set_angle перетворює значення кута в градусах у відповідне значення регістра порівняння таймера TIM2 для формування коректного ШІМ-сигналу. Дана функція зображена на рисунку 3.2.

```

c
void servo_set_angle(uint8_t channel, uint8_t angle) {
    if (angle > 180) angle = 180;
    /* 1ms = 0°, 2ms = 180°
       при частоті таймера 72МГц та періоді 20мс:
       250 = 1ms (0°), 500 = 2ms (180°) */
    float pulse = 250 + (angle * 1.3888f);
    __HAL_TIM_SET_COMPARE(&htim2, channel, (uint32_t)pulse);
}

```

Рисунок 3.1 - Функція керування кутом повороту сервоприводу

Дана функція конвертує значення кута в діапазоні від 0° до 180° у відповідне значення лічильника таймера. Значення 250 відповідає куту 0° (тривалість імпульсу 1 мс), а значення 500 відповідає куту 180° (тривалість імпульсу 2 мс). Для підключення математичних операцій у програмі підключається стандартна бібліотека мови С "math.h".

3.1.3 Функція режиму програмуван

Функція обробки режиму програмування `programming_mode_handler` реалізує логіку запису нової жестової команди. Дана функція зображена на рисунку 3.3.

					КС КРБ 123.275.00.00 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		


```

c
void programming_mode_handler(void) {
    /* Індикація входу в режим програмування */
    status_led_set(1);
    HAL_Delay(50); // захист від дребезгу кнопки

    /* Зчитування поточних кутів сервоприводів */
    uint8_t s1 = current_angles[0];
    uint8_t s2 = current_angles[1];
    uint8_t s3 = current_angles[2];
    uint8_t s4 = current_angles[3];

    /* Очікування жесту від оператора */
    uint8_t gesture = 0xFF;
    while (gesture == 0xFF) {
        gesture = uart_receive_command();
        HAL_Delay(10);
    }

    /* Збереження команди у EEPROM */
    eeprom_write(gesture, s1, s2, s3, s4);

    /* Вихід з режиму програмування */
    status_led_set(0);
}

```

Рисунок 3.3 - Функція обробки режиму програмування жестових команд

Під час виконання режиму програмування вбудований світлодіод PC13 світитиметься, сигналізуючи оператору про активний режим запису. Після успішного збереження жестової команди у EEPROM світлодіод вимикається і система повертається у штатний режим роботи.

3.2 Програма розпізнавання жестів на Raspberry Pi

Програма обробки відеопотоку та розпізнавання жестів написана мовою Python з використанням бібліотек MediaPipe та OpenCV. Головна функція програми виконує безперервне захоплення кадрів з камери, передачу їх до модуля розпізнавання та відправку сформованих команд до

					КС КРБ 123.275.00.00 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

мікроконтролера STM32 через UART. Фрагмент головного циклу програми зображено на рисунку 3.4.

```
python

import cv2
import mediapipe as mp
import serial

ser = serial.Serial('/dev/ttyAMA0', 9600)
cap = cv2.VideoCapture(0)
hands = mp.solutions.hands.Hands()

while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break
    results = hands.process(
        cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
    if results.multi_hand_landmarks:
        gesture = classify_gesture(
            results.multi_hand_landmarks[0])
        if gesture is not None:
            ser.write(bytes([gesture]))
```

Рисунок 3.4 - Головний цикл програми розпізнавання жестів на Raspberry Pi

Функція `classify_gesture` аналізує координати 21 ключової точки кисті руки, обчислює кути між фалангами пальців та повертає байтовий код розпізнаного жесту, або `None` у разі відсутності відомого жесту у полі зору камери.

					КС КРБ 123.275.00.00 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

3.3 Інтеграція механічних елементів з мікроконтролером

3.3.1 Живлення та керування сервоприводами

Керуючий сигнал широтно-імпульсної модуляції для сервоприводів формується апаратними каналами таймера TIM2 мікроконтролера STM32F103C8T6. Практичне дослідження підтвердило, що логічного рівня 3.3 В на виходах STM32 є достатньо для коректного розпізнавання керуючого сигналу сервоприводами MG996R без використання додаткових перетворювачів рівня.

Живлення сервоприводів здійснюється окремо від лінії +5 В блоку живлення, що унеможливило просадки напруги на мікроконтролері під час різких рухів маніпулятора та підвищує стабільність роботи всієї системи.

3.3.2 Підключення EEPROM-пам'яті

Мікросхема зовнішньої EEPROM-пам'яті AT24C32 підключена до інтерфейсу I2C1 мікроконтролера по лініях PB6 (SCL) та PB7 (SDA) із підтягуючими резисторами номіналом 4.7 кОм до лінії +3.3 В. Запис та зчитування даних здійснюється за допомогою функцій HAL_I2C бібліотеки STM32 HAL. Адреса мікросхеми на шині I2C становить 0x57, що визначається апаратними пінами A0–A2 мікросхеми, підключеними до землі.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Моделювання та прогнозування небезпечних ситуацій

У кваліфікаційній роботі описується процес створення комп'ютеризованої інтелектуальної двоманіпуляторної системи для асистування в монтажних процесах з безконтактним інтерфейсом керування, яка відповідно до свого призначення буде експлуатуватися безпосередньо на робочому місці монтажника у виробничому приміщенні. У зв'язку з цим у розділі розглядаються питання моделювання та прогнозування небезпечних ситуацій, особливості заходів захисту безпеки роботи, організація безпечної роботи електроустановок. При введенні системи в експлуатацію враховано фактори безпеки й передбачення небезпечних ситуацій в автоматизованому режимі роботи.

Небезпечні події з несприятливими наслідками мають свої історичні назви, а саме: аварія, стихійне лихо, катастрофа. Аварія — це небезпечна подія техногенного характеру, що створює загрозу для життя і здоров'я людини, призводить до руйнування будівель, споруд, обладнання і транспортних засобів, порушення виробничого чи транспортного процесу, або завдає шкоди довкіллю. Стихійне лихо — це небезпечна подія природного походження, яка за своєю інтенсивністю, масштабом поширення і тривалістю вражає людей, об'єкти економіки та довкілля. Катастрофа — це великомасштабна аварія, стихійне лихо чи інша подія, що призводить до тяжких, трагічних наслідків. Стан, який виникає після небезпечних подій, є надзвичайною ситуацією.

Враховуючи специфіку роботи двоманіпуляторної системи у виробничому середовищі, необхідно моделювати і прогнозувати всі можливі

					КС КРБ 123.275.00.00 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Шустак Д. С.			Безпека життєдіяльності, основи охорони праці			Літ.	Арк.	Аркушів	
Перевірив		Стрембіцький М.								44	
Консульт.		Гурик О.Я.						ТНТУ, каф. КС, гр. СІс-41			
Н. Контр.		Тиш Є. В.									
Затверд.		Осухівська Г.М.									

небезпечні ситуації. Моделювання і прогнозування небезпек на практиці проходить три стадії.

На першій стадії визначають матеріальні носії небезпек, тобто небезпечні та шкідливі чинники і умови, за яких вони можуть призвести до небажаних наслідків.

На другій стадії визначається головна небезпечна подія і послідовність інших небезпечних подій та умов, які їй передують. На цій стадії будується логічна схема розвитку небезпеки у вигляді дерева небезпечних подій та причин. Небезпечні події можуть відбуватися послідовно одна за одною, паралельно одна одній, а найчастіше — за змішаною послідовно-паралельною схемою. Для відображення схем реалізації небезпечних подій використовують логічні оператори «І» та «АБО».

На третій стадії аналізують можливі небажані наслідки і визначають можливі шляхи зменшення їхнього негативного впливу. В основу такого аналізу покладено методику порівняння затрат, спрямованих на відвернення небезпек, і вигод, отриманих від зниження рівня ризику.

У розробленій системі можна виділити наступні матеріальні носії небезпек.

Першим є рухомі механічні елементи маніпулятора. Сервоприводи та ланки маніпулятора здійснюють активні рухи під час роботи системи, що створює ризик механічного травмування оператора або оточуючих осіб. Умовами виникнення даної небезпеки є некоректне розпізнавання жесту алгоритмом, збій у програмному забезпеченні або неочікувана команда, що призведе до непередбачуваного руху маніпулятора. Передуючою причиною може бути недостатня освітленість робочого місця, яка знижує якість розпізнавання жестів камерою технічного зору.

Другим носієм небезпеки є електрична складова системи. Система містить компоненти, що живляться від напруги 5 В, зокрема сервоприводи MG996R та обчислювальний модуль Raspberry Pi, що при порушенні ізоляції

					КС КРБ 123.275.00.00 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

або неправильному монтажі може становити загрозу ураження електричним струмом для оператора.

До небажаних наслідків у випадку механічних ризиків відносяться: защемлення пальців або кисті руки ланками маніпулятора під час руху, удар ланкою маніпулятора по тілу оператора, пошкодження елементів монтажу внаслідок неконтрольованого руху захватного механізму. До небажаних наслідків електричної небезпеки відносяться ураження електричним струмом та вихід з ладу компонентів системи внаслідок короткого замикання.

З урахуванням цих небезпек та їхнього моделювання в системі розроблено алгоритми їхнього застереження, зокрема зупинка маніпулятора при втраті відеосигналу з камери та обмеження швидкості переміщення ланок маніпулятора.

4.2 Особливості заходів захисту безпеки роботи

Електроустановка — це установка, в якій виробляється, перетворюється, передається, розподіляється та споживається електрична енергія. За видами струму електроустановки поділяють на установки змінного та постійного струму.

Вимоги до заходів захисту від прямого дотику в нормальному режимі роботи електроустановок полягають у тому, що струмовідні частини електроустановок не повинні бути доступні для випадкового прямого дотику, а доступні для дотику відкриті і сторонні провідні частини не повинні перебувати під напругою, що становить небезпеку ураження електричним струмом у нормальному режимі роботи та в разі пошкодження ізоляції.

Для запобігання ураженню електричним струмом у нормальному режимі роботи застосовуються окремо або у поєднанні такі заходи захисту від прямого дотику:

- основна ізоляція струмовідних частин;
- огорожі та оболонки в електроустановках;

					КС КРБ 123.275.00.00 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

- бар'єри в електроустановках;
- блокування безпеки в електроустановках.

Для забезпечення механічної безпеки розробленої системи передбачено наступні заходи захисту:

- обмеження робочої зони маніпулятора конструктивними упорами сервоприводів, що унеможлиблює вихід ланок за безпечні межі;
- програмне обмеження швидкості переміщення сервоприводів для запобігання різким некерованим рухам маніпулятора;
- зупинка маніпулятора у поточному положенні при втраті відеосигналу з камери технічного зору або при відсутності розпізнаного жесту протягом більше ніж 2 секунди;
- розміщення камери технічного зору таким чином, щоб робоча зона маніпулятора була чітко відокремлена від зони знаходження рук оператора.

Ізоляція — це шар діелектрика або конструкція, виконана з діелектрика, за допомогою якого струмовідні частини відокремлюються одна від одної чи від інших конструктивних елементів обладнання. Електроустановки передусім мають робочу ізоляцію, яка забезпечує протікання струму потрібним шляхом і безпечну експлуатацію обладнання.

Контроль ізоляції підрозділяють на:

- періодичний — виконують у певні проміжки часу;
- постійний — виконують впродовж усього періоду роботи електроустановки;
- передпусковий — виконують перед пуском електроустановки.

4.3 Організація безпеки роботи електроустановок

Персонал, що обслуговує електрообладнання, може потрапити під напругу внаслідок несправності, аварії або власних помилкових дій. Безпека обслуговування електрообладнання залежить від його робочої напруги, умов експлуатації і характеру середовища приміщення, в якому воно встановлене.

					КС КРБ 123.275.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

Значною мірою безпека обслуговування електрообладнання залежить від умов середовища приміщення, в якому воно встановлене, оскільки ці умови впливають на стан ізоляції і опір шкіри людини. Волога, їдкі пари або гази, струмопровідний пил і висока температура знижують опір ізоляції та руйнують її. Крім того, шкіра людини під впливом вологи і високої температури стає провідною, що зменшує опір тіла людини і підвищує небезпеку ураження електричним струмом.

Таким чином, ознаками підвищеної небезпеки є:

- волога (відносна вологість повітря вище 75%) або наявність струмопровідного пилу;
- струмопровідні підлоги;
- підвищена температура повітря (більше +35°C);
- можливість одночасного дотику людини до заземлених корпусів обладнання і до частин електрообладнання, що перебуває під напругою.

Ознакою особливої небезпеки є наявність особливої вологості (відносна вологість повітря близька до 100%) і хімічно активне середовище. За ступенем небезпеки ураження людей електричним струмом приміщення підрозділяють на класи:

- приміщення з підвищеною небезпекою (присутня лише одна ознака підвищеної небезпеки);
- особливо небезпечні приміщення (наявність хоча б однієї ознаки особливої небезпеки або одночасно двох чи більше ознак підвищеної небезпеки).

Робоче місце монтажника, де експлуатується розроблювана система, відноситься до приміщень без підвищеної небезпеки, оскільки система живиться від безпечної напруги 5 В, а всі струмовідні частини надійно ізольовані та розміщені у корпусі пристрою. Для забезпечення безпечної експлуатації системи оператором необхідно дотримуватись наступних вимог:

- не допускати потрапляння вологи на електронні компоненти системи;

					КС КРБ 123.275.00.00 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

- не виконувати монтаж та демонтаж компонентів системи під напругою;
- перевіряти цілісність ізоляції з'єднувальних кабелів перед кожним вмиканням системи;
- не допускати перевищення максимального струму споживання блоку живлення.

Вимоги безпеки розробленої комп'ютеризованої інтелектуальної двоманіпуляторної системи враховано при її проектуванні та реалізації. Дотримання наведених вимог забезпечує безпечну та ефективну експлуатацію системи в умовах монтажного виробництва.

					КС КРБ 123.275.00.00 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було виконано наступне завдання:

- Виконано огляд існуючих аналогів систем допоміжних маніпуляторів та засобів безконтактного керування, проведено їх порівняльний аналіз за критеріями вартості, функціональності та зручності використання, на основі якого обґрунтовано доцільність розробки власної системи.

- Визначено технічні вимоги до апаратної та програмної складових системи, зокрема вимоги до точності розпізнавання жестів, затримки реакції маніпулятора, надійності роботи та безпеки оператора.

- Розроблено узагальнену структурну схему комп'ютеризованої інтелектуальної двоманіпуляторної системи та обґрунтовано вибір апаратного забезпечення, зокрема мікроконтролера STM32F103C8T6, обчислювального модуля Raspberry Pi 4, камери технічного зору та сервоприводів MG996R.

- Розроблено електричну принципову схему системи з детальним описом підключення всіх компонентів, розподілу портів GPIO мікроконтролера та організації живлення.

- Розроблено алгоритм розпізнавання жестів на основі бібліотеки MediaPipe та алгоритм режиму програмування жестових команд, що дозволяє оператору самостійно прив'язати положення маніпулятора до довільного жесту.

- Написано програмне забезпечення системи: програму розпізнавання жестів мовою Python для обчислювального модуля Raspberry Pi та програму керування маніпулятором мовою C у середовищі STM32CubeIDE для мікроконтролера STM32F103C8T6.

Розроблена комп'ютеризована інтелектуальна двоманіпуляторна система для асистування в монтажних процесах з безконтактним інтерфейсом

					КС КРБ 123.275.00.00 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

керування була зібрана у вигляді тестового прототипу на основі мікроконтролера STM32F103C8T6 та обчислювального модуля Raspberry Pi 4.

За результатами тестування система показала повну працездатність та коректність роботи всіх режимів відповідно до визначених технічних вимог: розпізнавання жестів здійснюється в режимі реального часу із затримкою реакції не більше 200 мс, режим програмування жестових команд функціонує коректно із збереженням запрограмованих положень у енергонезалежній пам'яті після вимкнення живлення системи.

Розроблена комп'ютеризована система готова до виконання задач асистування оператору під час монтажних процесів завдяки інтуїтивному безконтактному керуванню та функції гнучкого програмування жестових команд під конкретні виробничі потреби.

					КС КРБ 123.275.00.00 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жаровський Р.О., Луцик Н.С., Осухівська Г.М., Паламар А.М., Тиш Є.В. Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль: ТНТУ, 2024. 39 с.
2. Universal Robots – Collaborative Robot Arms. URL: <https://www.universal-robots.com/products/>. (дата звернення: 10.02.2026)
3. Leap Motion Controller – Technical Specifications. URL: https://www.ultraleap.com/datasheets/Leap_Motion_Controller_Datasheet.pdf. (дата звернення: 12.02.2026)
4. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп'ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ, 2019. 150 с.
5. MediaPipe Hands – Google AI. URL: https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker. (дата звернення: 14.02.2026)
6. Bazarevsky V., Grishchenko I., Raveendran K., Zhu T., Zhang F., Grundmann M. BlazePalm: Real-time Hand Tracking via ML Pipeline. URL: <https://arxiv.org/abs/2006.10214>. (дата звернення: 15.02.2026)
7. Datasheet STM32F103C8T6. URL: <https://www.st.com/resource/en/datasheet/stm32f103c8.pdf>. (дата звернення: 20.02.2026)
8. Raspberry Pi 4 Model B – Technical Specifications. URL: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>. (дата звернення: 21.02.2026)

					КС КРБ 123.275.00.00 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

9. Raspberry Pi Camera Module V2 – Technical Documentation. URL: <https://www.raspberrypi.com/documentation/accessories/camera.html>. (дата звернення: 22.02.2026)

10. MG996R Servo Motor Datasheet. URL: <https://components101.com/motors/mg996r-servo-motor-datasheet>. (дата звернення: 25.02.2026)

11. AT24C32 EEPROM Datasheet – Microchip Technology. URL: <https://ww1.microchip.com/downloads/en/DeviceDoc/doc0336.pdf>. (дата звернення: 01.03.2026)

12. Getting started with UART on STM32. URL: https://wiki.st.com/stm32mcu/wiki/Getting_started_with_UART. (дата звернення: 05.03.2026)

13. Getting started with I2C on STM32. URL: https://wiki.st.com/stm32mcu/wiki/Getting_started_with_I2C. (дата звернення: 06.03.2026)

14. Getting started with PWM on STM32. URL: https://wiki.st.com/stm32mcu/wiki/Getting_started_with_PWM. (дата звернення: 07.03.2026)

15. OpenCV Python Tutorial – Computer Vision. URL: https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html. (дата звернення: 10.03.2026)

16. PySerial Documentation. URL: <https://pyserial.readthedocs.io/en/latest/>. (дата звернення: 12.03.2026)

17. Оконський М.В., Лупенко С.А., Паламар А.М. Комп'ютерна система для моніторингу метеорологічних параметрів на основі IoT. Актуальні задачі сучасних технологій: збірник тез доповідей X міжнародної науково-практичної конференції молодих учених та студентів (Тернопіль, 24–25 листопада 2021 року). Тернопіль: ТНТУ, 2021. С. 112.

					КС КРБ 123.275.00.00 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

18. Харченко О., Яцишин В. Розробка та керування вимогами до програмного забезпечення на основі моделі якості. Вісник ТДТУ. Тернопіль, 2009. Т. 14. №1. С. 201–207.

19. Слюз І., Жаровський Р. Принципи та основні етапи комплексного тестування комп'ютерної інформаційної системи. Матеріали Х науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі системи та технології» (7–8 грудня 2022 року). Тернопіль: ТНТУ, 2022. С. 93.

20. Запорожець О.І. Безпека життєдіяльності. Підручник, 2-е видання. Центр учбової літератури, 2020. 448 с.

21. Аварії, катастрофи та стихійні лиха: основні принципи і способи захисту населення. URL: <https://osvita.ua/vnz/reports/dpju/24307/>. (дата звернення: 22.05.2026)

22. ДСТУ EN IEC 31010:2022. Керування ризиками – методи оцінювання ризику. – Введено в дію 31 грудня 2023 р. – Київ: ДП «УкрНДНЦ», 2023. – 120 с. – (Гармонізований з EN IEC 31010:2019 та IEC 31010:2019).

23. IEC 60038 | Європейські стандарти змінної електричної напруги. URL: <https://mirnyk.com/ua/2024/05/17/iec-60038yevropejski-standarty-zminnoyielektrychnoyi-naprugy/>. (дата звернення: 25.05.2026)

24. Правила улаштування електроустановок. ПУЕ.

25. Правила безпечної експлуатації електроустановок споживачів: НПАОП 40.1-1.21-98. – Київ: Міністерство праці та соціальної політики України, 1998. – 15 с.

					КС КРБ 123.275.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

Додаток А

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

“Затверджую”

Завідувач кафедри КС

_____ Осухівська Г.М.

“ 2 ” лютого 2026 р.

Комп'ютеризована двоманіпуляторна система безконтактного
асистування монтажних процесів

ТЕХНІЧНЕ ЗАВДАННЯ

на 9 листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

_____ к.т.н., доц. Стрембіцький М. О.

“ 2 ” лютого 2026 р.

«ВИКОНАВЕЦЬ»

Студент групи Сіс-41

_____ Шустак Д. С.

“ 2 ” лютого 2026 р.

Тернопіль 2026

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Компютеризована двоманіпуляторна система безконтактного асистування монтажних процесів».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.275.00.00

1.2 Виконавець

Студент групи Сіс-41, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Шустак Д. С.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету №4/9-18_ від 24.04.2026 р

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 26.01.2026 р.

Плановий термін завершення виконання кваліфікаційної роботи – 21.06.2026 р.

1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ISO, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи. Попередній захист кваліфікаційної роботи відбувається при готовності роботи – наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

2 Призначення і цілі створення системи

2.1 Призначення системи

Комп'ютеризована інтелектуальна двоманіпуляторна система що розробляється призначена для:

- асистування оператору під час виконання монтажних робіт шляхом надання додаткової механічної кінцівки;
- безконтактного керування маніпулятором за допомогою жестів рук оператор;
- утримання, затискання та відпускання елементів монтажу, зокрема проводів та дрібних деталей;
- гнучкого програмування жестових команд під конкретні виробничі потреби без втручання розробника.

2.2 Мета створення системи

Метою кваліфікаційної роботи є розробка та реалізація комп'ютеризованої інтелектуальної двоманіпуляторної системи для асистування в монтажних процесах з безконтактним інтерфейсом керування на основі мікроконтролера STM32F103C8T6, обчислювального модуля Raspberry Pi 4 та камери технічного зору з алгоритмами розпізнавання жестів на базі бібліотеки MediaPipe.

2.3 Характеристика об'єкту

Розроблювана двоманіпуляторна система призначена для використання на робочому місці монтажника у виробничому приміщенні. Система є автономним пристроєм, що складається з механічної конструкції маніпулятора на сервоприводах MG996R, обчислювального модуля Raspberry Pi 4 з підключеною камерою технічного зору та мікроконтролера STM32F103C8T6, який здійснює керування виконавчими механізмами. Система функціонує в режимі реального часу та не потребує постійного підключення до зовнішніх мереж або персонального комп'ютера.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

3.1.1 Вимоги до структури та функціонування системи

Комп'ютеризована інтелектуальна двоманіпуляторна система повинна складатись з:

- апаратної частини (мікроконтролер STM32F103C8T6, сервоприводи MG996R, камера технічного зору, обчислювальний модуль Raspberry Pi 4, EEPROM AT24C32, кнопка програмування);
- програмного забезпечення мікроконтролера для керування сервоприводами та обробки команд;
- програмного забезпечення обчислювального модуля для розпізнавання жестів у реальному часі;
- механічної конструкції маніпулятора з захватним механізмом.

3.1.2 Вимоги до способів та засобів зв'язку між компонентами системи

Основна вимога, яка ставиться до способів та засобів інформаційного обміну між компонентами системи — це їх узгодженість та мінімальна затримка передачі даних. Зв'язок між обчислювальним модулем Raspberry Pi 4 та мікроконтролером STM32F103C8T6 здійснюється через послідовний інтерфейс UART зі швидкістю передачі 9600 бод. Зв'язок між мікроконтролером та зовнішньою EEPROM-пам'яттю AT24C32 здійснюється через інтерфейс I2C1. Керуючі сигнали до сервоприводів передаються у вигляді ШІМ-сигналів через канали таймера TIM2 мікроконтролера. Усі інтерфейси зв'язку повинні забезпечувати стабільну передачу даних в умовах виробничого середовища.

3.1.3 Вимоги до режимів функціонування системи

Для системи визначено два режими функціонування:

- штатний режим функціонування;
- режим програмування жестових команд.

Основним режимом функціонування є штатний режим, в якому система безперервно розпізнає жести оператора та керує маніпулятором у

реальному часі. Для забезпечення штатного режиму функціонування системи необхідно виконувати вимоги та дотримуватись умов експлуатації програмного забезпечення і комплексу технічних засобів системи, вказані у відповідних технічних документах.

Режим програмування жестових команд активується натисканням апаратної кнопки на піні PB0 мікроконтролера. В даному режимі оператор самостійно прив'язує положення маніпулятора до обраного жесту з наступним збереженням у зовнішній EEPROM-пам'яті. Після завершення програмування система автоматично повертається до штатного режиму функціонування.

3.1.4 Вимоги по діагностуванню системи

Для діагностування системи використовуються інструменти діагностування основних процесів, вбудовані в операційну систему Raspberry Pi OS, а також засоби для діагностики апаратного забезпечення мікроконтролера STM32F103C8T6. Діагностування мікроконтролера здійснюється через програматор ST-Link з використанням інтерфейсу SWD та середовища розробки STM32CubeIDE. Індикація поточного стану системи здійснюється за допомогою вбудованого світлодіода на піні PC13 мікроконтролера. Інструменти діагностування повинні забезпечувати можливість перегляду стану всіх компонентів системи та виявлення несправностей у роботі інтерфейсів UART та I2C.

3.1.5 Перспективи розвитку, проектування системи

Дана система може бути розширена завдяки використанню додаткових апаратних та програмних компонентів. Зокрема, передбачено можливість підключення додаткових сервоприводів до вільних каналів таймерів TIM3 та TIM4 мікроконтролера для розширення ступенів свободи маніпулятора. Обсяг збережених жестових команд може бути збільшений шляхом заміни мікросхеми EEPROM на модель більшої ємності з тим самим інтерфейсом I2C. Також передбачено можливість вдосконалення алгоритму розпізнавання

жестів шляхом навчання власної нейронної мережі для підвищення точності класифікації у складних умовах освітлення.

3.2 Показники призначення

Система повинна передбачати можливість масштабування. Можливості масштабування повинні забезпечуватися засобами використовуваного базового програмного та технічного забезпечення. Система повинна розпізнавати не менше восьми різних жестових команд у режимі реального часу із затримкою реакції маніпулятора не більше 200 мс. Камера технічного зору повинна забезпечувати частоту кадрів не менше 30 кадрів/с при роздільній здатності не менше 720p для стабільного розпізнавання жестів.

3.2.1 Вимоги до надійності

Система повинна забезпечувати працездатність та відновлення своїх функцій при виникненні наступних ситуацій:

- при збоях у системі електропостачання апаратної частини;
- помилках у роботі апаратних засобів, зокрема при втраті відеосигналу з камери технічного зору;
- при помилках, пов'язаних з програмним забезпеченням операційної системи та драйверів пристроїв.

У разі втрати відеосигналу з камери маніпулятор повинен автоматично зупинитися та утримувати поточне положення до відновлення сигналу. Запрограмовані жестові команди повинні зберігатися в енергонезалежній EEPROM-пам'яті та залишатися доступними після вимкнення та повторного вмикання живлення системи. Для захисту апаратури від стрибків напруги та комутаційних завад повинні застосовуватися мережні фільтри.

3.3 Вимоги до безпеки

Зовнішні елементи технічних засобів системи, що перебувають під напругою, повинні мати захист від випадкового дотику, а самі технічні засоби — занулення або захисне заземлення. Конструкція маніпулятора повинна унеможливлювати травмування оператора під час виконання рухів, зокрема шляхом конструктивного обмеження робочої зони сервоприводів.

Система електроживлення повинна забезпечувати захисне вимикання при перевантаженнях і коротких замиканнях в колах навантаження, а також аварійне ручне вимикання. Живлення сервоприводів MG996R повинно здійснюватися окремим ланцюгом від блоку живлення, відокремленим від ланцюга живлення мікроконтролера, для унеможливлення просадок напруги та нестабільної роботи системи.

Загальні вимоги пожежної безпеки повинні відповідати нормам на побутове електрообладнання. У разі пожежі не мають виділятися отруйні гази та дим. Після знеструмлення системи має бути доступне застосування будь-яких засобів пожежогасіння.

3.3.1 Вимоги до експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи

Мікроклімат у приміщеннях, де експлуатується система, повинен відповідати нормам виробничого мікроклімату по ДСН 3.3.6.042-99:

- температуру повітря в межах від $+10^{\circ}\text{C}$ до $+35^{\circ}\text{C}$;
- відносну вологість повітря при 25°C в межах від 30% до 80%;
- атмосферний тиск 760 ± 25 мм рт. ст.

Періодичне технічне обслуговування використовуваних технічних засобів має проводитися відповідно до вимог технічної документації, але не рідше ніж один раз на рік. Технічне обслуговування повинно включати перевірку цілісності з'єднувальних кабелів, перевірку стану сервоприводів та механічних елементів конструкції маніпулятора. На підставі результатів тестування технічних засобів повинні проводитися аналіз причин виникнення виявлених дефектів та прийматися заходи щодо їх ліквідації.

3.4 Вимоги до захисту інформації від несанкціонованого доступу

Система повинна забезпечувати захист від несанкціонованого доступу на рівні не нижче встановленого вимогами, що пред'являються до категорії 1Д по класифікації документа «Автоматизовані системи. Захист від несанкціонованого доступу до інформації. Класифікація автоматизованих систем».

Компоненти підсистеми захисту від несанкціонованого доступу повинні забезпечувати:

- ідентифікацію користувача системи;
- перевірку повноважень користувача при роботі з системою;
- доступу користувачів до функцій керування маніпулятором та режиму програмування жестових команд.

Рівень захищеності від несанкціонованого доступу засобів обчислювальної техніки, що здійснюють обробку інформації, повинен відповідати вимогам класу захищеності згідно з вимогами документа «Засоби обчислювальної техніки. Захист від несанкціонованого доступу до інформації. Показники захищеності від несанкціонованого доступу до інформації».

3.4.1 Вимоги по збереженню інформації при аваріях

Інформація про запрограмовані жестові команди та положення маніпулятора при виникненні аварійних ситуацій повинна бути збережена в енергонезалежній зовнішній EEPROM-пам'яті AT24C32. Дані зберігаються незалежно від стану живлення системи та залишаються доступними після усунення аварійної ситуації та повторного вмикання системи.

3.4.2 Вимоги по стандартизації і уніфікації

Система повинна відповідати вимогам ергономіки та зручності користування. Безконтактний жестовий інтерфейс керування повинен бути інтуїтивно зрозумілим та не вимагати від оператора спеціальної технічної підготовки. Алгоритм розпізнавання жестів повинен стабільно функціонувати в умовах стандартного виробничого освітлення. Усі апаратні компоненти системи повинні мати необхідні сертифікати відповідності та безпеки.

3.4.3 Вимоги до функцій (завдань), що виконуються системою:

- розпізнавання жестів оператора в режимі реального часу із затримкою реакції не більше 200 мс;
- керування положенням маніпулятора за командами жестового інтерфейсу: переміщення вгору, вниз, вліво, вправо, вперед, назад, затискання та відпускання захвату;
- програмування нових жестових команд оператором без втручання розробника;
- збереження запрограмованих жестових команд в енергонезалежній пам'яті;
- автоматична зупинка маніпулятора при втраті відеосигналу з камери;
- індикація поточного стану системи за допомогою вбудованого світлодіода

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:
 - а) Структурна схема.
 - б) Блок схема роботи комп'ютеризованої системи.
 - в) Схема електрична принципова.

5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи бакалавра	Термін виконання
1	<i>Розробка технічного завдання</i>	26.01 – 02.02
2	<i>Робота над першим розділом «Аналіз технічного завдання»</i>	03.02 – 15.02
3	<i>Робота над другим розділом «Проектування структурної та електричної схеми»</i>	20.04 – 23.04
4	<i>Робота над третім розділом «Практична частина»</i>	23.04 – 01.05
5	<i>Робота над четвертим розділом «Безпека життєдіяльності, основи охорони праці»</i>	01.05 – 20.05
6	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	20.05 – 7.06
7	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	8.06 – 14.06
8	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	15.06 – 21.06
9	<i>Захист кваліфікаційної роботи бакалавра</i>	

6 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.

Додаток Б

Перелік елементів

[illegible]

Додаток В

Код програми

Лістинг В.1 — Raspberry Pi (Python):

```
import cv2
import mediapipe as mp
import serial

ser = serial.Serial('/dev/ttyAMA0', 9600,
timeout=1)
mp_hands = mp.solutions.hands
hands = mp_hands.Hands(max_num_hands=1,
min_detection_confidence=0.7)
mp_drawing = mp.solutions.drawing_utils
cap = cv2.VideoCapture(0)

GESTURE_CODES = {
    'STOP': 0x00, 'UP': 0x01, 'DOWN': 0x02,
    'LEFT': 0x03, 'RIGHT': 0x04, 'FORWARD': 0x05,
    'BACKWARD': 0x06, 'GRIP_CLOSE': 0x07,
    'GRIP_OPEN': 0x08, 'NO_GESTURE': 0xFF
}

def get_finger_states(landmarks):
    fingers = []
    fingers.append(1 if landmarks[4].x <
landmarks[3].x else 0)
    for tip, pip in zip([8,12,16,20],
[6,10,14,18]):
        fingers.append(1 if landmarks[tip].y <
landmarks[pip].y else 0)
    return fingers

def classify_gesture(hand_landmarks):
    lm = hand_landmarks.landmark
    thumb, index, middle, ring, pinky =
get_finger_states(lm)

    if sum([thumb,index,middle,ring,pinky]) == 0:
        return GESTURE_CODES['STOP']
    if sum([thumb,index,middle,ring,pinky]) == 5:
        return GESTURE_CODES['FORWARD']
    if index==1 and middle==0 and ring==0 and
```

```

pinky==0:
    dx = lm[8].x - lm[5].x
    dy = lm[8].y - lm[5].y
    if dy < -0.10: return GESTURE_CODES['UP']
    if dy > 0.10: return GESTURE_CODES['DOWN']
    if dx < -0.10: return GESTURE_CODES['LEFT']
    if dx > 0.10: return
GESTURE_CODES['RIGHT']
    if index==1 and middle==1 and ring==0 and
pinky==0:
        return GESTURE_CODES['GRIP_OPEN']
    dist = ((lm[4].x-lm[8].x)**2 + (lm[4].y-
lm[8].y)**2)**0.5
    if dist < 0.05 and middle==0:
        return GESTURE_CODES['GRIP_CLOSE']
    if thumb==1 and pinky==1 and index==0 and
middle==0 and ring==0:
        return GESTURE_CODES['BACKWARD']
    return GESTURE_CODES['NO_GESTURE']

prev_gesture = GESTURE_CODES['NO_GESTURE']
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        ser.write(bytes([GESTURE_CODES['STOP']]))
        break
    frame = cv2.flip(frame, 1)
    results = hands.process(cv2.cvtColor(frame,
cv2.COLOR_BGR2RGB))
    gesture_code = GESTURE_CODES['NO_GESTURE']
    if results.multi_hand_landmarks:
        for hl in results.multi_hand_landmarks:
            mp_drawing.draw_landmarks(frame, hl,
mp_hands.HAND_CONNECTIONS)
            gesture_code = classify_gesture(hl)
    if gesture_code != prev_gesture:
        if gesture_code !=
GESTURE_CODES['NO_GESTURE']:
            ser.write(bytes([gesture_code]))
        prev_gesture = gesture_code
    cv2.imshow('Gesture Control', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()

```

```
cv2.destroyAllWindows()
ser.close()
```

Лістинг В.2 - STM32F103C8T6 (C):

```
#include "main.h"

#define BUTTON_PIN          GPIO_PIN_0
#define BUTTON_PORT         GPIOB
#define LED_PIN             GPIO_PIN_13
#define LED_PORT            GPIOC
#define EEPROM_ADDR         (0x57 << 1)
#define EEPROM_RECORD_SIZE  5
#define MAX_GESTURES        50
#define SERVO_STEP          10
#define SERVO_NEUTRAL       90

/* Коди жестів */
#define G_STOP              0x00
#define G_UP                0x01
#define G_DOWN              0x02
#define G_LEFT              0x03
#define G_RIGHT             0x04
#define G_FORWARD           0x05
#define G_BACKWARD          0x06
#define G_GRIP_CLOSE        0x07
#define G_GRIP_OPEN         0x08
#define G_NONE              0xFF

TIM_HandleTypeDef  htim2;
UART_HandleTypeDef huart1;
I2C_HandleTypeDef hi2c1;

uint8_t current_angles[4] = {90, 90, 90, 90};
uint8_t uart_rx_byte = 0;
uint8_t programming_mode = 0;

void servo_set_angle(uint32_t ch, uint8_t angle) {
    if (angle > 180) angle = 180;
    uint32_t pulse = 1000 + (uint32_t)((float)angle
* 1000.0f / 180.0f);
    __HAL_TIM_SET_COMPARE(&htim2, ch, pulse);
}
```

```

void status_led_set(uint8_t state) {
    HAL_GPIO_WritePin(LED_PORT, LED_PIN,
        state ? GPIO_PIN_RESET : GPIO_PIN_SET);
}

uint8_t button_read(void) {
    return HAL_GPIO_ReadPin(BUTTON_PORT,
        BUTTON_PIN) == GPIO_PIN_RESET;
}

void eeprom_write_gesture(uint8_t gesture, uint8_t
*angles) {
    uint8_t data[EEPROM_RECORD_SIZE];
    for (uint16_t i = 0; i < MAX_GESTURES; i++) {
        uint16_t addr = i * EEPROM_RECORD_SIZE;
        HAL_I2C_Mem_Read(&hi2c1, EEPROM_ADDR, addr,
            I2C_MEMADD_SIZE_16BIT,
data, 1, 100);
        if (data[0] == gesture || data[0] == 0xFF)
        {
            data[0]=gesture; data[1]=angles[0];
            data[2]=angles[1]; data[3]=angles[2];
data[4]=angles[3];
            HAL_I2C_Mem_Write(&hi2c1, EEPROM_ADDR,
addr,
I2C_MEMADD_SIZE_16BIT,
                                data,
EEPROM_RECORD_SIZE, 100);
            HAL_Delay(10);
            return;
        }
    }
}

uint8_t eeprom_read_gesture(uint8_t gesture,
uint8_t *angles) {
    uint8_t data[EEPROM_RECORD_SIZE];
    for (uint16_t i = 0; i < MAX_GESTURES; i++) {
        HAL_I2C_Mem_Read(&hi2c1, EEPROM_ADDR, i *
EEPROM_RECORD_SIZE,
                                I2C_MEMADD_SIZE_16BIT,
data, EEPROM_RECORD_SIZE,
100);
        if (data[0] == gesture) {

```



```

        angles[0]=data[1]; angles[1]=data[2];
        angles[2]=data[3]; angles[3]=data[4];
        return 1;
    }
    if (data[0] == 0xFF) break;
}
return 0;
}

void execute_command(uint8_t cmd) {
    uint8_t prog_angles[4];
    if (eeprom_read_gesture(cmd, prog_angles)) {
        for (int i = 0; i < 4; i++)
current_angles[i] = prog_angles[i];
        servo_set_angle(TIM_CHANNEL_1,
current_angles[0]);
        servo_set_angle(TIM_CHANNEL_2,
current_angles[1]);
        servo_set_angle(TIM_CHANNEL_3,
current_angles[2]);
        servo_set_angle(TIM_CHANNEL_4,
current_angles[3]);
        return;
    }
    switch (cmd) {
        case G_UP:
            if (current_angles[1]+SERVO_STEP<=180)
current_angles[1]+=SERVO_STEP;
            servo_set_angle(TIM_CHANNEL_2,
current_angles[1]); break;
        case G_DOWN:
            if (current_angles[1]>=SERVO_STEP)
current_angles[1]-=SERVO_STEP;
            servo_set_angle(TIM_CHANNEL_2,
current_angles[1]); break;
        case G_LEFT:
            if (current_angles[0]>=SERVO_STEP)
current_angles[0]-=SERVO_STEP;
            servo_set_angle(TIM_CHANNEL_1,
current_angles[0]); break;
        case G_RIGHT:
            if (current_angles[0]+SERVO_STEP<=180)
current_angles[0]+=SERVO_STEP;
            servo_set_angle(TIM_CHANNEL_1,
current_angles[0]); break;
    }
}

```

```

        case G_FORWARD:
            if (current_angles[2]+SERVO_STEP<=180)
current_angles[2]+=SERVO_STEP;
            servo_set_angle(TIM_CHANNEL_3,
current_angles[2]); break;
        case G_BACKWARD:
            if (current_angles[2]>=SERVO_STEP)
current_angles[2]-=SERVO_STEP;
            servo_set_angle(TIM_CHANNEL_3,
current_angles[2]); break;
        case G_GRIP_CLOSE:
            if (current_angles[3]+SERVO_STEP<=180)
current_angles[3]+=SERVO_STEP;
            servo_set_angle(TIM_CHANNEL_4,
current_angles[3]); break;
        case G_GRIP_OPEN:
            if (current_angles[3]>=SERVO_STEP)
current_angles[3]-=SERVO_STEP;
            servo_set_angle(TIM_CHANNEL_4,
current_angles[3]); break;
        default: break;
    }
}

```

```

void programming_mode_handler(void) {
    programming_mode = 1;
    status_led_set(1);
    uint8_t saved[4];
    for (int i = 0; i < 4; i++) saved[i] =
current_angles[i];

    while (!button_read()) HAL_Delay(10);
    HAL_Delay(50);

    /* Мигання LED - очікування жесту */
    for (int i = 0; i < 6; i++) {
status_led_set(i%2); HAL_Delay(200); }
    status_led_set(1);

    uint8_t gesture = G_NONE;
    uint32_t timeout = HAL_GetTick() + 10000;
    while (gesture == G_NONE && HAL_GetTick() <
timeout) {
        if (uart_rx_byte != G_NONE && uart_rx_byte
!= 0) {

```

```

        gesture = uart_rx_byte;
        uart_rx_byte = 0;
    }
    HAL_Delay(10);
}
if (gesture != G_NONE) {
    eeprom_write_gesture(gesture, saved);
    for (int i = 0; i < 4; i++) {
status_led_set(i%2); HAL_Delay(100); }
    }
    programming_mode = 0;
    status_led_set(0);
    HAL_UART_Receive_IT(&huart1, &uart_rx_byte, 1);
}

/* Переривання UART - прийом команди від Raspberry
Pi */
void HAL_UART_RxCpltCallback(UART_HandleTypeDef
*huart) {
    if (huart->Instance == USART1) {
        if (!programming_mode)
execute_command(uart_rx_byte);
        HAL_UART_Receive_IT(&huart1, &uart_rx_byte,
1);
    }
}

int main(void) {
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_TIM2_Init();
    MX_USART1_UART_Init();
    MX_I2C1_Init();

    HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
    HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_2);
    HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_3);
    HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_4);

    for (int i = 0; i < 4; i++)
        servo_set_angle(TIM_CHANNEL_1 + i*4,
SERVO_NEUTRAL);

    HAL_UART_Receive_IT(&huart1, &uart_rx_byte, 1);

```

```
status_led_set(0);

while (1) {
    if (button_read()) {
        HAL_Delay(50);
        if (button_read())
programming_mode_handler();
    }
    HAL_Delay(10);
}
```