

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: *Комп'ютерна система керування стабілізаційною
платформою для наземної турелі*

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

(шифр і назва спеціальності)

	<u>Вітовщик П.В.</u> (підпис) (прізвище та ініціали)
Керівник	<u>Лецишин Ю.З.</u> (підпис) (прізвище та ініціали)
Нормоконтроль	<u>Луцик Н.С.</u> (підпис) (прізвище та ініціали)
Завідувач кафедри	<u>Осухівська Г.М.</u> (підпис) (прізвище та ініціали)
Рецензент	<u></u> (підпис) (прізвище та ініціали)

Тернопіль
2025

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)
«27» січня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

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

студента Вітовщика Павла Васильовича
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система керування стабілізаційною платформою для наземної турелі

Керівник роботи к. т. н. Лецишин Юрій Зіновійович
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 27 » січня 2025 року № 4/7-53

2. Термін подання студентом завершеної роботи 16.06.2025р

3. Вихідні дані до роботи Технічне завдання

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз технічного завдання на розробку кс керування стабілізаційною платформою турелік

2. Проєктування КС керування стабілізаційною платформою турелі

3. Програмування та тестування програмного забезпечення КС

4. Безпека життєдіяльності, основи охорони праці

Висновки

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

1. Структурна схема

2. Блок схема алгоритму роботи модуля керування двигунами

3. Блок схема алгоритму роботи модуля веб-сервера

4. Схема електрична принципова

5. Схема електрична монтажна (з'єднань)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Пилипець М. І. д.т.н. проф. каф. МТ</i>		

7. Дата видачі завдання 27.01.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Розробка технічного завдання.</i>	<i>27.01.2025р. – 02.02.2025р.</i>	<i>Викон.</i>
2.	<i>Робота над розділом аналіз технічного завдання на розробку кс керування стабілізаційною платформою турелі</i>	<i>02.02.2025р. – 10.03.2025р.</i>	<i>Викон.</i>
3.	<i>Робота над основним розділом проектування КС керування стабілізаційною платформою турелі</i>	<i>11.03.2025р. – 01.05.2025р.</i>	<i>Викон.</i>
4.	<i>Робота над основним розділом програмування та тестування програмного забезпечення КС</i>	<i>02.05.2025р. – 28.05.2025р.</i>	<i>Викон.</i>
5.	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>29.05.2025р. – 02.06.2025р.</i>	<i>Викон.</i>
6.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>03.06.2025р. – 09.06.2025р.</i>	<i>Викон.</i>
7.	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	<i>9.06.2025р. – 15.06.2025р.</i>	<i>Викон.</i>
8.	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	<i>16.06.2025р. – 22.06.2025р.</i>	<i>Викон.</i>
9.	<i>Захист кваліфікаційної роботи бакалавра</i>	<i>23.06.2025р.</i>	<i>Викон.</i>

Студент

(підпис)

Вітовцик Павло Васильович

(прізвище та ініціали)

Керівник роботи

(підпис)

к. т. н. Лецишин Юрій Зіновійович

(прізвище та ініціали)

АНОТАЦІЯ

Вітовщик П. В. Комп'ютеризована система для керування стабілізаційною платформою для наземної турелі: робота на здобуття кваліфікаційного ступеня бакалавра: спец. 123 — комп'ютерна інженерія. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2025.

Ключові слова: комп'ютерна система, керування, безщітковий двигун, stm32, raspberry pi, веб-інтерфейс, камера.

Кваліфікаційна робота присвячена дослідженню методів керування стабілізаційною платформою, з використанням веб-інтерфейсу. Робота охоплює аналіз наявних рішень у сфері розробки підвісу, а також наголошує на можливості дистанційного керування та моніторингу. Для створення системи керування стабілізаційною платформою використовується модуль керування двигунами, безщіткові двигуни постійного струму, драйвер для керування двигунами, raspberry pi.

Надіслані команди через веб-інтерфейс передаються до raspberry pi, що в свою чергу надсилає команди через UART до модуля керування двигунами, яка генерує ШІМ для драйвера двигунів. Як результат двигун повертаються в задане положення, що дає змогу точно керувати кутом повороту. Результати цього дослідження можуть мати практичне значення у керування положенням об'єктів в просторі.

ANNOTATION

Vitovshchyk P. V. Computerized control system for a stabilization platform for a ground turret: Bachelor's Graduation Thesis: speciality 123 — computer engineering. Ternopil: Ternopil Ivan Puluj National Technical University, 2025.

Keywords: computer system, control, motors, encoders, stm32, raspberry pi, web interface.

The qualification work is devoted to the study of methods for controlling a stabilization platform using a web interface. The work includes an analysis of existing solutions in the field of gimbal development, and also focuses on the possibility of remote control and monitoring. To create a control system for the stabilization platform, a printed circuit board with an stm32, BLDC motors, a driver for controlling the motors, and a Raspberry Pi are used.

The commands sent via the web interface are transmitted to the raspberry pi, which in turn sends commands via UART to the stm32 PCB, which generates PWM for the motor driver. As a result, the motors are rotated to a predetermined position, which allows for precise control of the angle of rotation. The results of this study can be of practical importance in controlling the position of objects in space.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ НА РОЗРОБКУ КС КЕРУВАННЯ СТАБІЛІЗАЦІЙНОЮ ПЛАТФОРМОЮ ТУРЕЛІ.....	11
1.1 Аналіз вимог до КС	11
1.2 Розгляд наявних рішень	14
1.3 Аналіз промислового рішення OTUS-U135.....	18
РОЗДІЛ 2 ПРОЄКТУВАННЯ КС КЕРУВАННЯ СТАБІЛІЗАЦІЙНОЮ ПЛАТФОРМОЮ ТУРЕЛІ.....	22
2.1 Розробка структурної схеми стабілізаційної платформи	22
2.2 Опис особливостей модуля для керування двигунами.....	24
2.3 Схема з'єднань КС.....	30
2.4 Аргументація вибору компонентів	32
РОЗДІЛ 3 ПРОГРАМУВАННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КС	38
3.1 Реалізація програмної частини	38
3.1.1 Код для модуля керування двигунами.....	38
3.1.2 Програмування роботи модуля веб-сервера	42
3.2 Тестування та запуск КС керування стабілізаційною платформою турелі 48	
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ..	53
4.1 Заходи щодо захисту від ураження електричним струмом.....	53
4.2 Заходи, що покращують умови праці оператора.....	54

					КС КРБ 123.196.00.00 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Вітовщик П.В.			Комп'ютеризована система керування стабілізаційною платформою для наземної турелі			Літ.	Арк.	Аркушів	
Перевір.		Лецишин Ю.З.							6		
Реценз.								ТНТУ, каф. КС, гр. СІ-41			
Н. Контр.		Луцик Н.С.									
Затверд.		Осухівська Г.М.									

ВИСНОВКИ.....58

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....59

Додаток А Технічне завдання

Додаток Б Програмний код

Додаток В Перелік елементів

					КС КРБ 123.196.00.00 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК СКОРОЧЕНЬ

ARM – Advanced RISC Machine

USB – Universal Serial Bus

UART – Universal asynchronous receiver/transmitter

CAN - Controller Area Network

RAM - Random-access memory

ROM - Read-only memory

PWM – Pulse-width modulation

ШИМ – Широтно-імпульсна модуляція

КС – Комп'ютеризована система

ПЗ – Програмне забезпечення

RPi - Raspberry Pi

					КС КРБ 123.196.00.00 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Розвиток сучасних технологій в галузі робототехніки, автономних систем та військової техніки висуває нові вимоги до точності та надійності систем позиціювання. Особливо важливою стає потреба у високоточних системах стабілізації, які забезпечують стабільне наведення різноманітного обладнання — від камер спостереження до спеціалізованих сенсорів та озброєння на рухомих платформах.

Традиційні методи стабілізації часто не забезпечують необхідної гнучкості та точності керування, особливо в умовах динамічного середовища. Сучасні можливості інтеграції комп'ютерних систем керування, веб-технологій та мікроконтролерів відкривають нові перспективи для створення адаптивних систем реального часу.

Інтеграція мікроконтролерів STM32, одноплатних комп'ютерів Raspberry Pi та безщіткових двигунів постійного струму дозволяє створювати комплексні системи, здатні до автоматичного адаптування до змін зовнішніх умов. Особливо перспективним є використання веб-інтерфейсів для дистанційного керування, що значно розширює функціональні можливості та доступність таких систем.

Практичне значення подібних розробок підтверджується широкими можливостями застосування в оборонній промисловості, системах безпеки, сільському господарстві та наукових дослідженнях. Створення комп'ютеризованої системи керування стабілізаційною платформою, яка поєднує переваги сучасних веб-технологій з можливостями мікроконтролерів, відповідає актуальним потребам ринку.

Мета роботи полягає в розробленні комп'ютеризованої системи керування стабілізаційною платформою турелі з веб-інтерфейсом для дистанційного управління, що забезпечує високоточне позиціювання та стабілізацію в режимі реального часу.

Наукова новизна роботи полягає в розробці оригінальної архітектури системи керування, яка вперше об'єднує веб-інтерфейс користувача з

					КС КРБ 123.196.00.00 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

низькорівневим апаратним забезпеченням через протокол UART з подальшою генерацією ШІМ-сигналів. Запропонована система реалізує безперервну взаємодію між веб-додатком, мінікомп'ютером Raspberry Pi та мікроконтролером STM32, що забезпечує підвищену інтеграцію програмних і апаратних компонентів.

Практичне значення розробленої системи полягає в її універсальності та можливостях застосування в автоматизованих комплексах відеоспостереження, робототехнічних системах, охоронних та військових об'єктах, а також в освітніх закладах. Реалізована система забезпечує можливість прецизійного дистанційного позиціонування об'єктів та може служити основою для подальшого розвитку автономних стабілізаційних платформ.

					КС КРБ 123.196.00.00 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ НА РОЗРОБКУ КС КЕРУВАННЯ СТАБІЛІЗАЦІЙНОЮ ПЛАТФОРМОЮ ТУРЕЛІ

1.1 Аналіз вимог до КС

Комп'ютеризована система керування стабілізаційною платформою призначена для забезпечення точного утримання об'єктів у фіксованій орієнтації незалежно від рухів базової платформи. Система базується на технології підвісу (див. рис. 1.1) з активним електронним керуванням у реальному часі та здатна забезпечувати точне позиціонування завдяки поєднанню високоточних двигунів та керування через веб-інтерфейс.

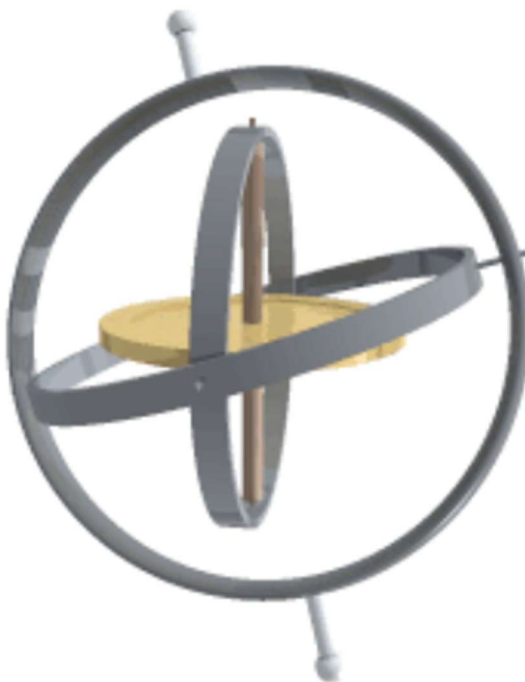


Рисунок 1.1 – Ілюстрація простого набору тривісних підвісів

					КС КРБ 123.196.00.00 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Вітовщик П.В.			Аналіз технічного завдання на розробку КС керування стабілізаційною платформою турелі			Літ.	Арк.	Аркушів	
Перевір.		Лецишин Ю.З.								11	
Реценз.								ТНТУ, каф. КС, гр. СІ-41			
Н. Контр.		Луцик Н.С.									
Затверд.		Осухівська Г.М.									

Основною функцією системи є багатовісне позиціювання, що підтримує двох- або тривісні конфігурації з окремим керуванням кожної осі. Система повинна забезпечувати точне наведення платформи згідно з командами оператора через веб-інтерфейс, а також відстеження об'єктів за допомогою трекера на камері після їх вибору користувачем.

Керування системою здійснюється виключно через вбудований веб-інтерфейс, який дозволяє дистанційне управління з будь-якого пристрою, підключеного до локальної або глобальної мережі. Оператор може задавати напрямки повороту платформи за осями, а також активувати режим автоматичного відстеження після виділення цілі на камері. Це забезпечує високу мобільність оператора та можливість оперативного реагування на події.

Адаптивність системи проявляється в її здатності перемикатись від ручного керування оператора до автоматичного відстеження об'єктів за допомогою камерного трекера. Модульна архітектура дозволяє інтегрувати різні типи обладнання та розширювати функціональність залежно від специфічних потреб застосування.

Архітектура системи базується на гібридному підході, що поєднує STM32 мікроконтролер для низькорівневого керування двигунами та обробки сигналів у реальному часі з Raspberry Pi як центральним комп'ютером для логіки системи, обробки команд, мережевого доступу та візуалізації. Така конфігурація забезпечує високу продуктивність, масштабованість та розширюваність системи.

Силові компоненти включають безщіткові двигуни постійного струму, які забезпечують активне керування рухом осей для позиціювання об'єкта. Генерація широтно-імпульсних сигналів дозволяє досягти точного позиціювання платформи, зменшити споживання енергії та рівень шуму, а також реалізувати плавний і швидкий рух, необхідний для точного наведення.

Система позиціювання базується виключно на командах від оператора

					КС КРБ 123.196.00.00 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

через веб-інтерфейс, де користувач вказує напрямок повороту платформи. Додатково система використовує трекер на камері для відстеження об'єктів після їх первинного виявлення та вибору оператором.

Система повинна забезпечувати швидку реакцію на команди оператора з мінімальною затримкою між отриманням команди через веб-інтерфейс та виконанням руху платформи. Плавність руху є критичною для точного позиціонування без ривків, що досягається через ефективну обробку команд керування та генерацію сигналів у реальному часі. Точність позиціонування досягається завдяки використанню прецизійних приводів та ефективних алгоритмів керування двигунами, при цьому трекер на камері забезпечує точне відстеження об'єктів після їх виділення оператором, дозволяючи автоматично утримувати ціль у полі зору без постійного ручного корегування.

Модульна архітектура системи забезпечує гнучкість та розширюваність, дозволяючи додавання нових компонентів та функцій без кардинальної перебудови основної конструкції. Така архітектурна концепція підтримує різні конфігурації та типи навантаження, а інтеграційні можливості гарантують сумісність з наявними системами та протоколами.

Управління та моніторинг реалізовані через повноцінне віддалене керування за допомогою мережевих інтерфейсів, при цьому система має можливість моніторингу власного стану, а користувацький інтерфейс розроблений інтуїтивно зрозумілим для операторів різного рівня підготовки.

Ключові інноваційні особливості системи включають:

- Веб-орієнтований підхід до дистанційного керування як сучасна альтернатива традиційним методам управління.
- Гібридна архітектура, що оптимально поєднує продуктивність та функціональність через розподіл завдань між спеціалізованими обчислювальними модулями.

Енергоефективність системи досягається через мінімізацію енергоспоживання при максимальній продуктивності, що особливо важливо

					КС КРБ 123.196.00.00 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

для мобільних платформ з обмеженими енергетичними ресурсами. Адаптивність системи проявляється в її здатності автоматично пристосовуватися до змінюваних умов експлуатації, забезпечуючи стабільну роботу як у режимі ручного керування, так і в режимі автоматичного відстеження цілей.

1.2 Розгляд наявних рішень

Для порівняння з розробленою системою розглянемо існуючу реалізацію двовісного підвісу-контролера, яка демонструє альтернативний підхід до стабілізації. Дана система побудована на базі мікроконтролера STM32, що працює у зв'язці з модулем бездротового зв'язку HC-06 для забезпечення передачі даних через Bluetooth-з'єднання. Основною особливістю цієї реалізації є отримання інформації про орієнтацію безпосередньо з Android-додатка, встановленого на смартфоні.

На відміну від запропонованої системи з веб-керуванням, даний контролер функціонує в режимі повної автоматичної стабілізації без можливості ручного втручання користувача. Принцип роботи базується на безперервній передачі даних з акселерометра та гіроскопа смартфона на мікроконтролер, який у режимі реального часу обробляє ці значення та генерує відповідні команди для сервоприводів.

Архітектура системи відрізняється своєю простотою та складається з трьох взаємопов'язаних компонентів: Android-додатка, що генерує дані про просторове положення телефону; мікроконтролера STM32, який виконує обробку отриманих даних; та сервоприводів, які безпосередньо змінюють положення телефону відповідно до розрахованих кутів компенсації. Загальна архітектура системи та взаємодія її компонентів показана на рисунку 1.2, де чітко видно потік даних від сенсорів смартфона до виконавчих механізмів.

					КС КРБ 123.196.00.00 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

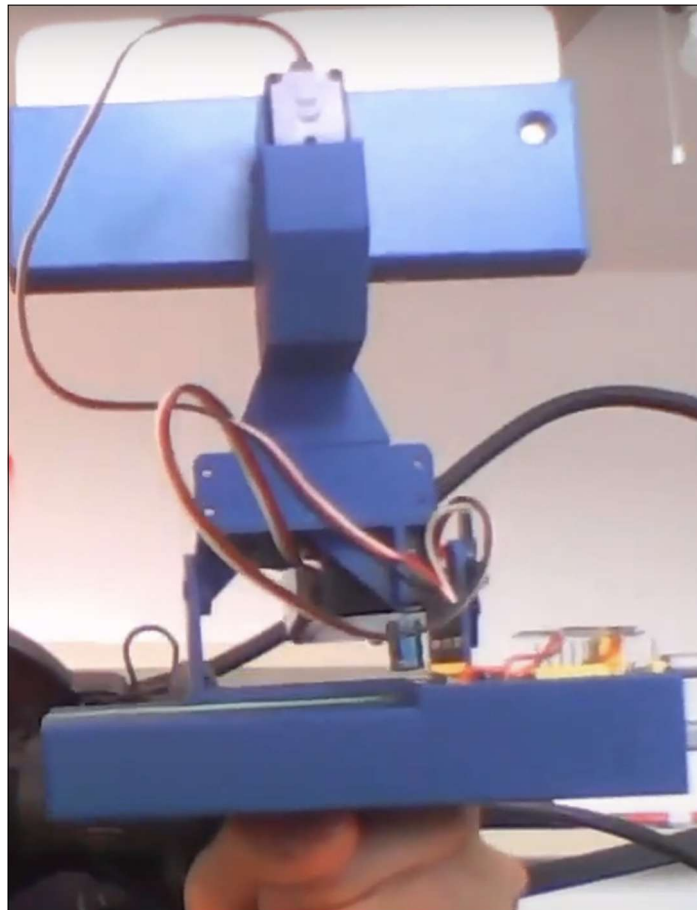


Рисунок 1.2 – Вигляд аналога системи в зібраному стані

Функціональні можливості системи обмежуються стабілізацією по двох основних осях — pitch (нахил вперед/назад) та roll (нахил вліво/вправо), що дозволяє компенсувати найбільш критичні коливання пристрою під час руху або ходьби. Хоча це рішення є базовим порівняно з багатофункціональними системами, воно забезпечує помітне покращення стабільності відео, особливо в умовах динамічного використання.

Зовнішній вигляд готового пристрою, представлений на рисунку 1.3, демонструє практичну реалізацію концепції стабілізації. Смартфон надійно закріплений у спеціальному тримачі, який механічно з'єднаний з конструкцією, що містить сервоприводи. Така конфігурація дозволяє користувачеві зручно утримувати пристрій у руках або встановлювати його на

штатив, при цьому система автоматично забезпечує стабілізацію положення телефону відповідно до змін його орієнтації в тривимірному просторі.



Рисунок 1.3 – Аналогова система зі сторони користувача

Користувацький інтерфейс системи характеризується мінімалізмом та простотою використання — оператору достатньо лише активувати Android-додаток та встановити Bluetooth-з'єднання, після чого система переходить в автономний режим роботи. Це суттєво відрізняється від запропонованої веб-орієнтованої системи, де користувач має повний контроль над процесом позиціонування.

Алгоритм роботи включає автоматичне визначення початкового положення пристрою як опорної точки, на основі якої система розраховує необхідні кути повороту сервоприводів для кожної з осей стабілізації. Це забезпечує коректну початкову орієнтацію та стабільну роботу системи навіть після перезапуску або зміни базового положення пристрою, що є важливою перевагою для практичного використання.

Android-додаток забезпечує візуалізацію роботи системи, відображаючи в реальному часі координати пристрою по трьох просторових осях (X, Y, Z), а також критичні кути нахилу — pitch та roll, як показано на рисунку 3. Дані з

					КС КРБ 123.196.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

вбудованих датчиків смартфона (акселерометра та гіроскопа) обробляються безперервно, навіть коли додаток функціонує у фоновому режимі, що особливо важливо під час запису відео або виконання інших завдань, коли стабілізація повинна залишатися активною без додаткової взаємодії з користувачем.

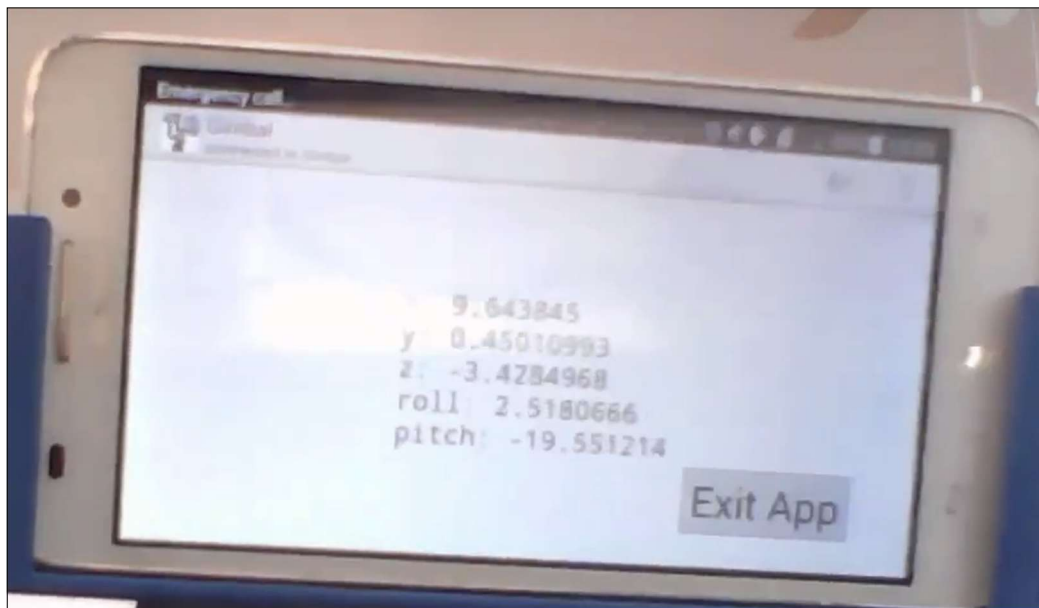


Рисунок 1.4 – Вигляд інтерфейсу додатка для Android

Завершальним етапом обробки є перетворення розрахованих кутів компенсації у відповідні ШІМ-сигнали (широтно-імпульсна модуляція), які передаються на сервоприводи для точного позиціювання. Ці сигнали визначають конкретне положення кожного сервоприводу, що забезпечує плавне та прецизійне керування положенням мобільного телефону. Результатом такого підходу є ефективна стабілізація зображення, значне зменшення тремтіння камери та загальне покращення якості відео порівняно з нестабілізованими системами.

1.3 Аналіз промислового рішення OTUS-U135

Стабілізація OTUS-U135 — це сучасна, високоефективна гіростабілізована система, що демонструє баланс між компактністю, точністю та функціональністю. Завдяки direct-drive двигунам, slip-rings, потужній імпульсній системі та багатому набору опцій (авто-трекінг, геоприв'язка), вона представляє технологічний орієнтир для розробки малих стабілізаційних платформ. У порівнянні з низькобюджетними моделями, OTUS-U135 — це приклад промислового рішення зі збалансованими технічними параметрами для UAV/UGV застосувань, хоча й з функціональними обмеженнями, що властиві двовісним платформам. Дану систему можна розглянути на рисунку 1.5.



Рисунок 1.5 – Зовнішній вигляд системи OTUS-U135

Враховуючи необхідність застосування на мобільних платформах обмеженого розміру, корпус OTUS-U135 (див. рис. 1.6) має діаметр приблизно 135 мм, що дозволяє легко інтегрувати його навіть на платформах з обмеженим простором. Загальна маса пристрою складає близько 1 кг, що робить його придатним для встановлення на дрони малого та середнього розміру без значного впливу на тривалість польоту або маневровість. Попри

ці скромні габарити, попри свої компактні розміри, платформа забезпечує надзвичайно точну стабілізацію з максимальним рівнем вібраційного зсуву до 100 мікрорадіан, що відповідає приблизно 0.0057 градуса. Це дає змогу зберігати чіткість і фокус зображення навіть під час різких маневрів або роботи в умовах високої турбулентності.

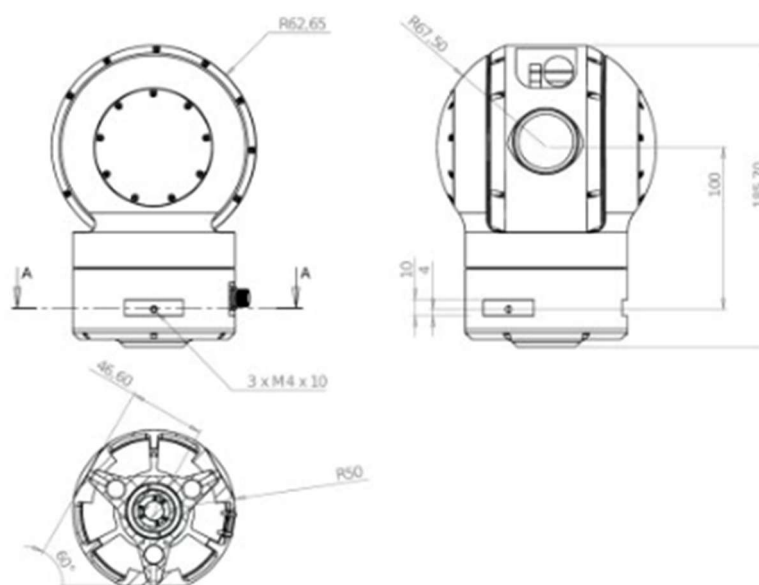


Рисунок 1.6 – Розміри корпусу OTUS-U135

Така висока точність стабілізації досягається завдяки використанню передових технічних рішень [2]. Стабілізація системи досягається за допомогою інтегрованих безщіткових електроприводів прямого приводу, що забезпечують високу динаміку відгуку і точне позиціонування в межах обох осей. Ключовою особливістю конструкції є використання так званих slip-rings — обертових контактів, які дозволяють необмежено обертати модуль у панорамній осі без заплутування кабелів, що вкрай важливо для безперервного стеження або кругового огляду. Максимальна швидкість обертання платформи може досягати 120–180 градусів на секунду, що забезпечує швидке наведення на нову ціль.

Для забезпечення високої точності позиціювання та швидкості реакції, інтелектуальне ядро платформи складається з високоточного інерціального вимірювального блоку (IMU), розміщеного безпосередньо на гіростабілізованій частині. Це дозволяє зменшити ефекти паразитної інерції та підвищити точність корекції кута нахилу та обертання. Розширюючи можливості навігації та позиціювання, OTUS-U135 підтримує інтеграцію з різними зовнішніми модулями GPS і цифровими компасами, що дозволяє реалізувати функції геоприв'язки і координатного наведення в режимі реального часу.

Гнучкість системи забезпечується її модульною архітектурою. Платформа може бути оснащена різними типами сенсорів, зокрема денними камерами (EO), тепловізійними модулями (IR), лазерними далекомірами та комбінаціями кількох каналів спостереження. Наприклад, у стандартній конфігурації OTUS-U135 використовуються сенсори FLIR або Sony з підтримкою відеовиходів у форматах NTSC/PAL, а також можливістю трансляції зображення через цифрові інтерфейси типу Ethernet. Для забезпечення комплексної інтеграції з різними системами, пристрій підтримує інтерфейс керування через RS-485, що дає змогу реалізувати інтеграцію з наземними або бортовими системами керування, а також здійснювати дистанційне управління положенням платформи і параметрами відео.

Окрім базових можливостей стабілізації та керування, серед додаткових функцій OTUS-U135 передбачено можливість автоматичного відстеження об'єктів у відеопотоці (video auto-tracking) та геопросторового трекінгу, що особливо важливо для місій із динамічним супроводженням цілі. Платформа підтримує визначення координат об'єкта в полі зору і здатна автоматично утримувати фокус на ньому, навіть при його русі або зміні просторового положення. Ця функція реалізується через обчислення зворотної геопозиції за допомогою телеметричних даних та IMU.

					КС КРБ 123.196.00.00 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

З погляду практичного використання, живлення OTUS-U135 здійснюється від джерела постійного струму в діапазоні 18–36 В, а споживана потужність коливається в межах 15–30 Вт. Робочий температурний діапазон становить від 0 до +50 °С, але за потреби може бути розширений до -40 °С шляхом використання додаткового обігріву та захисту. Конструкція герметична, захищена від пилу та вологи відповідно до військових стандартів, що дозволяє використовувати OTUS-U135 у складних польових умовах.

Завдяки поєднанню високих технічних характеристик та надійності, OTUS-U135 часто використовується у військових і цивільних застосуваннях, включаючи моніторинг кордонів, охорону об'єктів, картографування, розвідку, супровід підозрілих об'єктів та участь у системах автономного наведення. Його відмінною рисою є винятковий баланс між компактністю та функціональністю, що дає змогу вважати цю систему зразком промислової реалізації високоточних стабілізаційних технологій.

Підсумовуючи аналіз системи OTUS-U135, основними перевагами промислової системи є стабільність у жорстких умовах, глибока інтеграція електроніки, наявність функцій геотрекінгу і автотрекінгу, а також висока точність механіки. Разом з тим, дані системи залишаються недоступними для широкого аматорського використання через високу вартість і складність інтеграції, що підкреслює розрив між дослідницькими прототипами й готовими промисловими рішеннями.

					КС КРБ 123.196.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

РОЗДІЛ 2 ПРОЄКТУВАННЯ КС КЕРУВАННЯ СТАБІЛІЗАЦІЙНОЮ ПЛАТФОРМОЮ ТУРЕЛІ

2.1 Розробка структурної схеми стабілізаційної платформи

На рисунку 2.1 представлена структурна схема розробленої системи керування стабілізаційною платформою, яка демонструє організацію взаємодії між усіма компонентами системи та потоки даних між ними.

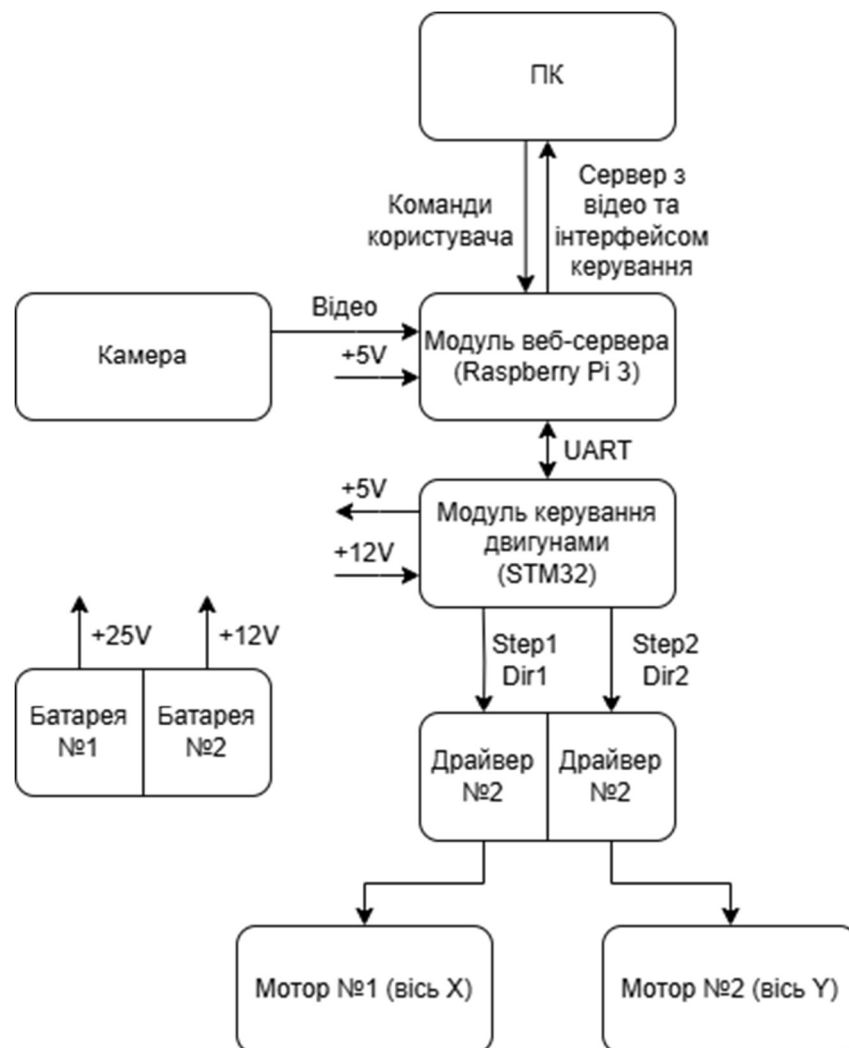


Рисунок 2.1 – Структурна схема комп’ютерної системи керування стабілізаційною платформою для наземної турелі

					КС КРБ 123.196.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Вітовцик П. В.			Проекткування КС керування стабілізаційною платформою турелі	Літ.	Арк.
Перевір.		Лецишин Ю.З.					22
Реценз.						ТНТУ, каф. КС, гр. CI-41	
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					

Система побудована за ієрархічним принципом, де верхній рівень представлений персональним комп'ютером (ПК), який забезпечує інтерфейс користувача та відображення відеопотоку. Взаємодія між оператором та системою здійснюється через сервер з відео та інтерфейсом керування, який передає команди користувача та забезпечує передачу відеосигналу у реальному часі.

Центральним елементом системи є модуль веб-сервера, одноплатний комп'ютер Raspberry Pi 3, який функціонує як головний контролер та координує роботу всіх підсистем. Модуль веб-сервера отримує команди керування від ПК через мережеве з'єднання та забезпечує обробку відеосигналу від підключеної камери. Відеопотік передається на ПК для відображення оператору, створюючи замкнутий контур відеоспостереження.

Камера підключена до Raspberry Pi через стандартний відеоінтерфейс, що забезпечує стабільне функціонування та якісний відеосигнал. Така конфігурація дозволяє реалізувати режим реального часу для відеоспостереження з можливістю дистанційного керування.

Зв'язок між модулем веб-сервера та модулем керування двигунами здійснюється через інтерфейс UART, який забезпечує надійну передачу команд керування з контролем цілісності даних. Модуль керування двигунами отримує команди позиціонування та генерує відповідні керуючі сигнали для драйверів двигунів.

Модуль керування двигунами представляє рівень безпосереднього керування виконавчими механізмами та забезпечує генерацію сигналів Step1/Dir1 та Step2/Dir2 для керування безщітковими двигунами. Ці сигнали передаються на відповідний драйвер, які перетворюють логічні команди у силові сигнали, необхідні для керування.

Виконавча частина системи включає два безщіткові двигуни постійного струму: двигун №1 забезпечує обертання по осі X, а двигун №2 - по осі Y, що дозволяє здійснювати повноцінне двовісне позиціонування камери у просторі. Така конфігурація забезпечує можливість панорамування та нахилу для

					КС КРБ 123.196.00.00 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

охоплення широкого діапазону зон спостереження.

Система живлення організована за принципом багаторівневого електропостачання. Два акумулятори забезпечують напруги +25В та +12В для силових компонентів системи. Акумулятор №1 (+25В) призначений для живлення високопотужних компонентів, а акумулятор №2 (+12В) забезпечує живлення проміжних вузлів. Додатково система включає стабілізовані напруги +12В та +5В для живлення логічних схем, мікроконтролера та Raspberry PI.

Представлена структурна схема демонструє комплексний підхід до побудови системи відеоспостереження з активним позиціюванням, забезпечуючи інтеграцію відеозахоплення, обробки команд керування та прецизійного механічного позиціювання. Така архітектура забезпечує високу функціональність системи, надійність роботи та можливість дистанційного керування через мережеві інтерфейси.

2.2 Опис особливостей модуля для керування двигунами

Модуль керування двигунами є критично важливим етапом створення системи керування стабілізаційною платформою, оскільки саме вона забезпечує інтеграцію всіх електронних компонентів та їх надійну взаємодію. Основою модуля керування двигунами є мікроконтролер STM32G0B1RBT6 (див. рис. 2.2), який належить до сімейства високопродуктивних 32-розрядних мікроконтролерів на базі ARM Cortex-M0+ ядра.

Вибір мікроконтролера STM32G0B1RBT6 для реалізації системи керування, представленої на схемі з'єднань, обґрунтований комплексом технічних характеристик та функціональних можливостей, які ідеально відповідають специфічним вимогам проєкту робототехнічної системи з керуванням кроковими двигунами.

					КС КРБ 123.196.00.00 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

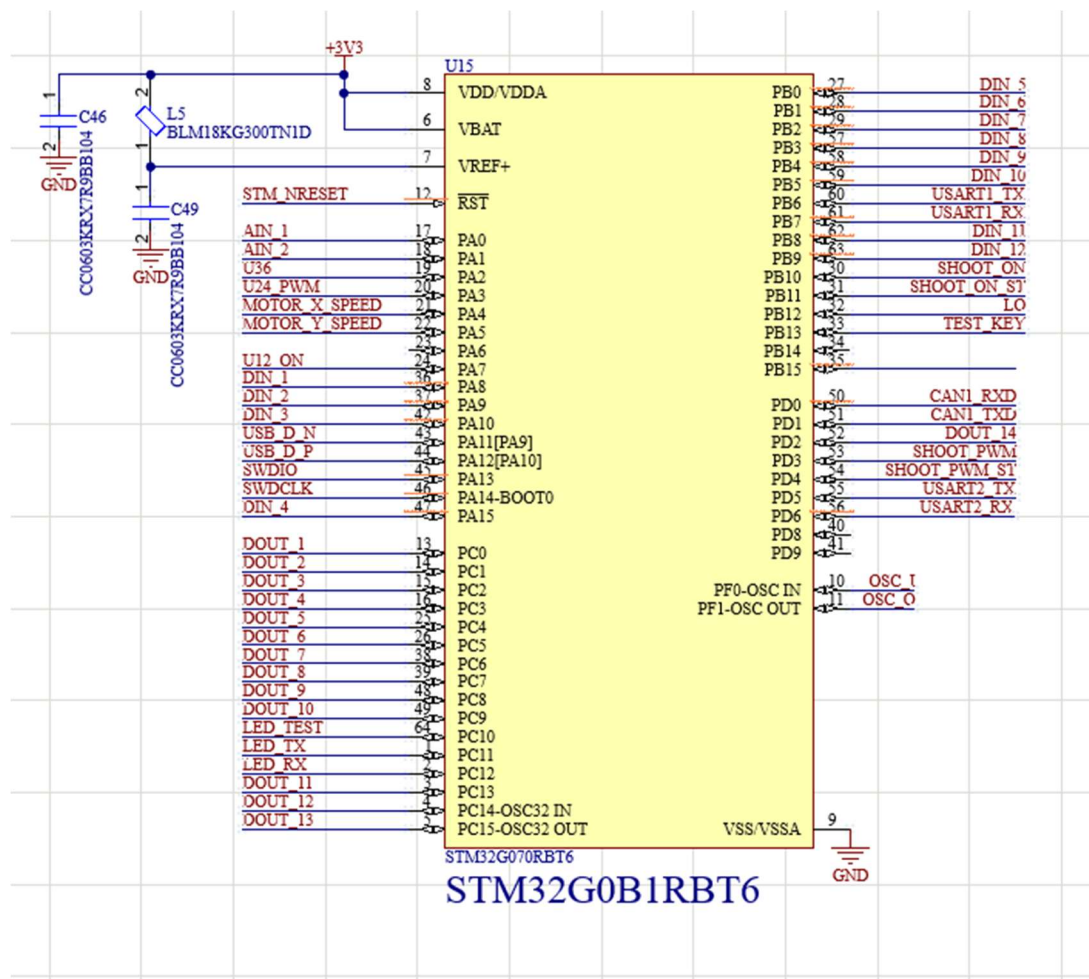


Рисунок 2.2 – Принципова електрична схема STM32G0B1RBT6

Таким чином, STM32G0B1RBT6 представляє оптимальний вибір для реалізації представленої схеми керування двигунами, забезпечуючи необхідну функціональність при мінімальних витратах та максимальній надійності [3].

Ключовою перевагою STM32G0B1RBT6 є наявність чотирьох 16-бітних таймерів загального призначення (TIM1, TIM2, TIM3, TIM16) з розширеними можливостями генерації ШІМ-сигналів. Кожен таймер здатен генерувати до 4 незалежних ШІМ-каналів з роздільністю до 16 біт та частотою до 1 МГц. Це дозволяє одночасно керувати двома кроковими двигунами (Motor №1 та Motor №2, згідно зі схемою) з високою точністю та плавністю.

На відміну від простіших мікроконтролерів типу Arduino Uno (ATmega328P), які мають лише 6 ШІМ-каналів з 8-бітною роздільністю, STM32G0B1RBT6 забезпечує значно вищу точність позиціонування - до 1/65536

від повного кроку замість 1/256.

Мікроконтролер оснащений трьома UART модулями, двома SPI інтерфейсами та двома I2C контролерами, що забезпечує гнучкість підключення різноманітних периферійних пристроїв. Згідно зі схемою, один з UART використовується для зв'язку з Raspberry Pi, а другий перероблений в RS485 та третій, а саме CAN зарезервований для підключення додаткових сенсорів або модулів зв'язку.

Порівняно з популярним STM32F103, який має лише 3 UART та 2 SPI, STM32G0B1RBT6 пропонує більшу кількість комунікаційних інтерфейсів при нижчому енергоспоживанні.

Мікроконтролер має 55 ліній GPIO, що повністю покриває потреби схеми з'єднань, показаної на рисунку. Важливою особливістю є можливість перемаршрутизації (remapping) функцій альтернативних портів, що дозволяє оптимально розмістити сигнальні лінії на модулі керування двигунами та мінімізувати довжину провідників.

Кожен GPIO порт здатен працювати в різних режимах: цифровий вхід/вихід, аналоговий вхід, альтернативна функція з підтримкою швидкості перемикання до 50 МГц. Це забезпечує надійну передачу швидких PULSE сигналів до драйвера DM556SX2 без спотворень.

Система живлення модуля організована за принципом багаторівневого стабілізованого живлення. На модулі передбачені окремі піни для підключення трьох різних напруг живлення: 12В для силових компонентів та драйверів двигунів, 5В для периферійних пристроїв та інтерфейсних схем, та 3.3В для живлення мікроконтролера та цифрової логіки. Така організація живлення забезпечує оптимальні умови роботи для кожного типу компонентів та мінімізує взаємний вплив між різними підсистемами.

Комунікаційні можливості реалізовані через комплексну систему інтерфейсів, а саме UART, CAN, RS485, що забезпечує гнучкість підключення, масштабованість системи та інтеграцію з різноманітними периферійними пристроями та мережевими архітектурами. Система включає кілька рівнів

					КС КРБ 123.196.00.00 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

комунікації, кожен з яких оптимізований для специфічних завдань передачі даних.

Інтерфейс UART забезпечує основний канал зв'язку між модем керування двигунами та модулем веб-сервера. Цей інтерфейс працює на стандартних швидкостях від 9600 до 2 Мбіт/с, що дозволяє передавати команди керування в реальному часі з мінімальними затримками.

Через UART канал передаються структуровані пакети даних, що включають команди позиціонування для безщіткових двигунів, параметри швидкості та прискорення, статусну інформацію про поточне положення виконавчих механізмів, телеметричні дані про споживання струму, температуру драйверів та напругу живлення. Протокол обміну реалізований з використанням контрольних сум CRC16 для забезпечення цілісності даних.

Особливістю реалізації є використання апаратного UART з підтримкою DMA, що дозволяє передавати великі масиви даних без навантаження на центральний процесор мікроконтролера. Це критично важливо при одночасному керуванні двома двигунами та обробці сигналів від множинних датчиків.

Додатково на модулі реалізований інтерфейс RS485, який забезпечує можливість організації розподіленої системи з декількома модулями керування або підключення додаткових периферійних пристроїв. RS485 інтерфейс особливо корисний для створення мережі з кількох стабілізаційних платформ або для інтеграції з наявними промисловими системами автоматизації.

Інтерфейс RS485 реалізований з використанням диференційної передачі сигналів, що забезпечує високу завадостійкість та можливість передачі даних на відстані до 1200 метрів при швидкостях до 10 Мбіт/с. Топологія мережі підтримує до 32 пристроїв на одній лінії з можливістю каскадного з'єднання для створення більших мереж.

Протокол RS485 включає адресацію пристроїв, що дозволяє створювати складні автоматизовані комплекси з централізованим керуванням. Кожен

					КС КРБ 123.196.00.00 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

модуль має унікальний мережевий адрес, що дозволяє головному контролеру селективно звертатися до конкретних виконавчих модулів. Реалізовано механізми виявлення колізій, автоматичного перевідправлення пакетів та діагностики стану мережі.

CAN інтерфейс - автомобільний стандарт для критичних додатків

Для забезпечення максимальної надійності та відмовостійкості в системі реалізований CAN (Controller Area Network) інтерфейс, який є промисловим стандартом для критичних застосувань у автомобільній, аерокосмічній та промисловій автоматизації. Мікроконтролер STM32G0B1RBT6 оснащений вбудованим CAN контролером, що дозволяє інтегрувати модуль в наявні CAN-мережі без додаткових компонентів.

CAN інтерфейс працює на швидкостях від 10 Кб/с до 1 Мб/с та забезпечує детерміновану передачу даних з гарантованими часовими характеристиками. Протокол CAN має вбудовані механізми виявлення та виправлення помилок, включаючи CRC контроль, біт-стаффінг та автоматичне перевідправлення пошкоджених повідомлень.

Особливістю CAN є пріоритетна арбітража, яка дозволяє критично важливим повідомленням (наприклад, командам екстреної зупинки) отримувати пріоритет над рутинними даними телеметрії. Кожне повідомлення має 11-бітний ідентифікатор, який одночасно служить адресою та пріоритетом.

У контексті робототехнічної системи CAN інтерфейс використовується для:

- синхронізації роботи кількох платформ у складних маніпуляційних комплексах;
- передачі критично важливих команд безпеки;
- інтеграції з системами технічного зору та навігації;
- підключення до промислових контролерів типу SCADA.

Модуль керування двигунами включає систему індикації стану, що складається з світлодіодів, які відображають наявність живлення, активність

					КС КРБ 123.196.00.00 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

комунікаційних інтерфейсів та стан керування двигунами. Це значно спрощує процес налагодження та діагностики системи під час розробки та експлуатації.

Конструктивно модуль керування двигунами має стандартні розміри та кріпильні отвори, що дозволяє легко інтегрувати її в різні механічні конструкції. Всі сигнальні та силові контакти захищені від короткого замикання та перенапруг за допомогою відповідних захисних елементів.

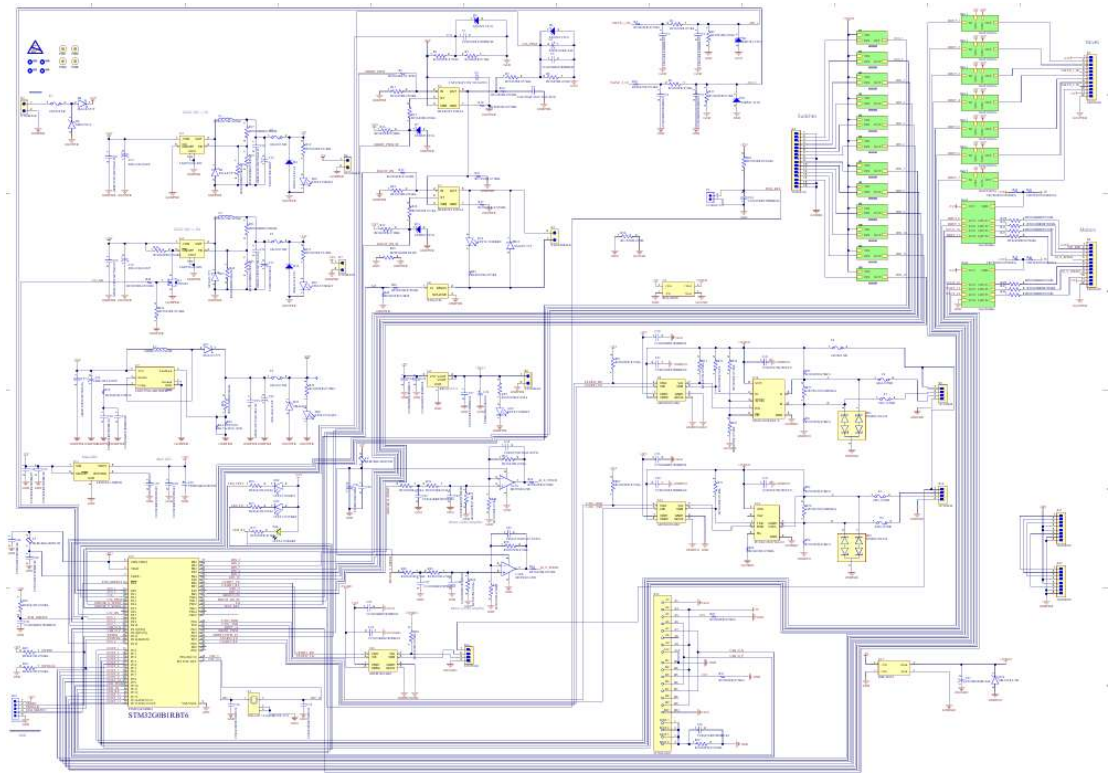


Рисунок 2.3 – Електрична принципова схема модуля керування двигунами

На рисунку 2.3 представлено принципову електричну схему модуля керування двигунами, яка демонструє детальну організацію всіх функціональних блоків та їх взаємозв'язки. Схема показує реалізацію системи живлення, комунікаційних інтерфейсів та допоміжних функцій.

Розміщення компонентів на платі виконано з урахуванням теплового режиму роботи та електромагнітної сумісності. Силові компоненти розташовані з одного боку модуля та забезпечені достатньою площею мідних провідників для ефективного відведення тепла. Аналогові схеми ізолювані від

цифрових за допомогою окремих земляних площин, що мінімізує цифрові завади в аналогових сигналах, також є можливість реалізувати фільтр описаний у [4], або корелювадані як у [5].

2.3 Схема з’єднань КС

Даний розділ містить детальний опис електричних з’єднань між основними компонентами робототехнічної системи керування. Схема (див. рис. 2.4) демонструє архітектуру багаторівневої системи, що базується на модулі керування двигунами керування двигунами та центральному обчислювальному вузлі модулем веб-сервера для виконання складних обчислювальних завдань.

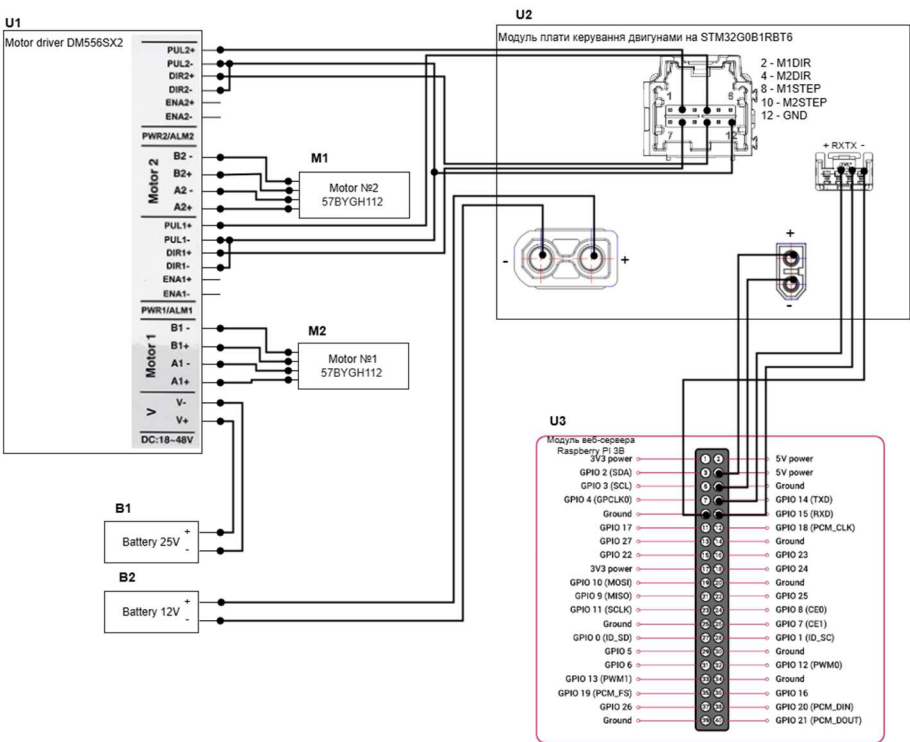


Рисунок 2.4 – Схема з’єднань КС керування стабілізаційною платформою турелі

Модуль керування двигунами виконує функції головного керуючого елемента системи реального часу. Він забезпечує безпосереднє керування всіма периферійними пристроями через систему GPIO портів, таймерів та

спеціалізованих інтерфейсів. Контролер обробляє сигнали від датчиків з мілісекундною точністю, генерує керуючі імпульси для приводів та забезпечує синхронізацію роботи всіх підсистем.

До основного модуля керування двигунами під'єднано широкий спектр сигнальних ліній, включаючи цифрові входи/виходи для керування логікою системи, та спеціалізовані порти для високошвидкісної передачі даних.

Драйвер двигунів DM556SX2 являє собою двоканальний контролер безщіткових двигунів з мікрокроковим керуванням. Кожен канал здатний керувати двигуном з номінальним струмом до 5.6А, що забезпечує високий крутний момент та точність позиціонування.

З'єднання з мікроконтролером включає:

- лінії PULSE1/PULSE2 для передачі тактових імпульсів, що визначають швидкість обертання;
- лінії DIR1/DIR2 для встановлення напрямку обертання;
- лінії PWR1/PWR2 для моніторингу стану живлення драйверів.

Безщіткові двигуни постійного струму 57BYGH112 підключені до відповідних виходів драйвера через чотири кабелі. Ці двигуни забезпечують крок 1.8° з можливістю мікрокрокового ділення до 1/256, що дозволяє досягти високої точності позиціонування механічних вузлів системи.

Raspberry Pi 3B інтегрована у систему як високорівневий обчислювальний модуль веб-сервера, відповідальний за обробку відеопотоку, реалізацію алгоритмів комп'ютерного зору, мережеву взаємодію та координацію роботи з віддаленими системами керування.

Підключення здійснюється через 40-контактний GPIO роз'єм, що забезпечує:

- двонаправлений обмін даними з основним мікроконтролером через UART/SPI інтерфейси;
- передачу команд та отримання телеметричних даних;
- синхронізацію роботи між швидкими операціями реального часу та складними алгоритмічними обчисленнями;

					КС КРБ 123.196.00.00 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

– живлення модуля веб-сервера здійснюється стабілізованою напругою 5В.

Акумулятор 25В забезпечує живлення силових компонентів системи, включаючи драйвери двигунів та високопотужні виконавчі механізми. Ця лінія живлення розрахована на струми до 10А.

Акумулятор 12В призначений для живлення електронних модулів, датчиків та допоміжного обладнання. З цієї лінії через DC-DC перетворювачі формуються стабілізовані напруги 5В та 3.3В для живлення цифрових схем, використовуючи [6].

Передбачено гальванічну розв'язку між силовими та сигнальними ланцюгами для підвищення завадостійкості системи.

2.4 Аргументація вибору компонентів

Вибір компонентної бази для розробки системи автоматизованого керування базується на комплексному аналізі технічних вимог, функціональних характеристик та економічної доцільності кожного елемента. Архітектура системи побудована на принципі ієрархічного розподілу обчислювальних задач між різними рівнями керування, що забезпечує оптимальне використання ресурсів та високу надійність функціонування.

Як основний обчислювальний модуль веб-сервера обрано одноплатний комп'ютер Raspberry Pi 3B (див. рис. 2.5), що оснащений чотириядерним процесором ARM Cortex-A53 з тактовою частотою 1.2 ГГц та 1 ГБ оперативної пам'яті LPDDR2 [7]. Даний вибір обґрунтовується достатньою обчислювальною потужністю для виконання складних алгоритмів та обробки користувацького інтерфейсу, наявністю вбудованих модулів Wi-Fi та Bluetooth для забезпечення бездротового зв'язку, а також 40-піновим GPIO інтерфейсом для підключення периферійних пристроїв. Суттєвими перевагами Raspberry Pi 3B є низька вартість при високій функціональності, широка підтримка спільноти розробників, сумісність з різними операційними

					КС КРБ 123.196.00.00 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

системами та можливість програмування на мовах Python і C++, що особливо важливо для робототехнічних застосувань.

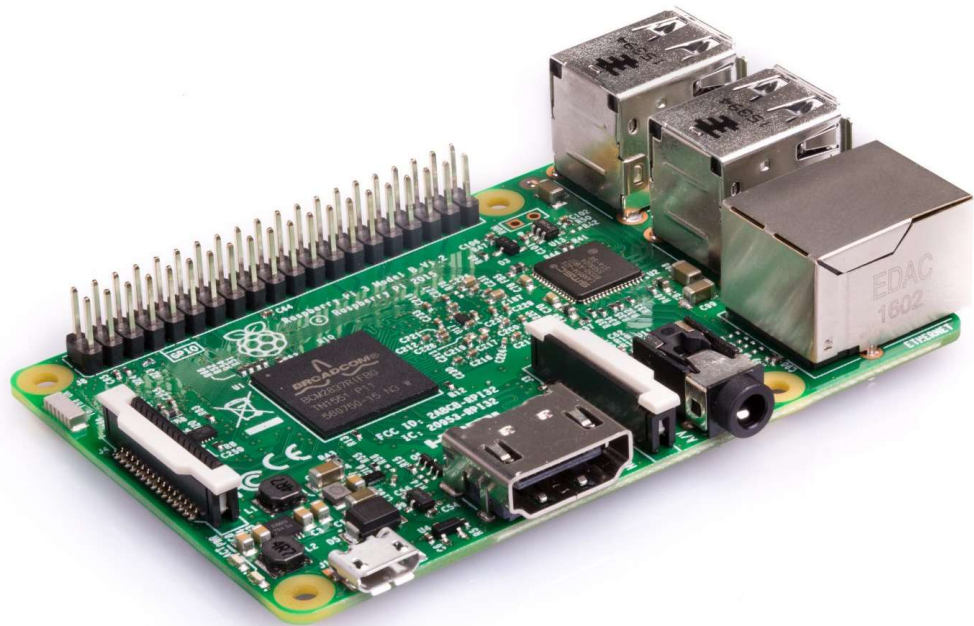


Рисунок 2.5 – Вигляд Raspberry Pi 3B

Для керування безщітковими двигунами використовуються драйвер DM556SX2 (див. рис. 2.6), які підтримують широкий діапазон струму від 1.0A до 5.6A при робочій напрузі живлення 24-50V постійного струму. Ці драйвери забезпечують мікрокроковий режим з роздільною здатністю до 25600 імпульсів на оборот, що гарантує надзвичайно високу точність позиціонування. Вбудована система захисту включає функції запобігання перенапрузі, пониженій напрузі та короткому замиканню.

Вибір даних драйверів обґрунтовується їх здатністю забезпечувати прецизійне мікрокрокове керування з низьким рівнем шуму та вібрацій, простим інтерфейсом керування через сигнали PUL/DIR/ENA.



Рисунок 2.6 – Драйвер DM556SX2

В якості виконавчих механізмів застосовуються безщіткові двигуни NEMA 23 моделі 57BYGH112 (див. рис. 2.7), що характеризуються кроком 1.8° , номінальним струмом фази 2.8A при напрузі 3.36В та моментом утримання 1.12 Н·м. Використання двох таких двигунів забезпечує достатній крутний момент для ефективного переміщення робототехнічних механізмів, високу точність позиціювання без необхідності використання додаткових редукторів, універсальність монтажу завдяки стандартному розміру NEMA 23 та повну сумісність з обраними драйверами [8].



Рисунок 2.7 – Безщіткові двигуни постійного струму 57BYGH112

Обрана комбінація компонентів створює систему з розподілом функцій між різними рівнями керування. Модуль веб-сервера виконує роль центрального обчислювального вузла для складних алгоритмів та інтерфейсних функцій, STM32G0B1RBT6 концентрується на задачах реального часу та низькорівневому керуванні, драйвери DM556SX2 забезпечують прецизійне перетворення цифрових команд у механічні рухи, а двигуни 57BYGH112 реалізують фізичні переміщення з необхідними параметрами.

Вибір Raspberry Pi HQ Camera (див. рис. 2.8) на базі сенсора Sony IMX477 для системи автоматизованого керування обґрунтовується необхідністю забезпечення високоякісного візуального зворотного зв'язку та реалізації алгоритмів машинного зору. Дана камера високої роздільної здатності оснащена 12.3-мегапіксельним стековим CMOS-сенсором зі зворотним підсвічуванням, що має діагональ 12.3" та максимальну роздільну здатність 4056×3040 пікселів.

					КС КРБ 123.196.00.00 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.8 – Raspberry Pi HQ Camera

Технічні характеристики та переваги Raspberry Pi HQ Camera

Розмір пікселя становить $1.55 \text{ нм} \times 1.55 \text{ нм}$, що є більшим за попередні модулі Raspberry Pi High Quality Camera Review & Specs | Arrow.com, забезпечуючи суттєво покращену світлочутливість та значне зменшення цифрового шуму в умовах слабкого освітлення. Ця характеристика особливо важлива для систем моніторингу, які повинні функціонувати цілодобово в різних умовах освітленості.

Порівняно з Camera Module v1 на базі сенсора OmnivisionOV5647 з роздільною здатністю 5 Мп, HQ Camera демонструє втричі вищу деталізацію зображення. Camera Module v2 та роздільною здатністю 8 Мп також поступається HQ Camera за кількістю пікселів, а світлочутливість у нього приблизно на 50% гірша через меншу площу пікселя Raspberry Pi High Quality HQ Camera.

Ключовою відмінністю HQ Camera від усіх інших модулів є можливість роботи зі змінними об'єктивами в форматах C- та CS-mount. Це дозволяє адаптувати камеру до специфічних завдань спостереження, використовувати

					КС КРБ 123.196.00.00 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

професійні оптичні системи, досягати різних кутів огляду та фокусних відстаней, а також здійснювати точне налаштування діафрагми та фокуса відповідно до умов експлуатації.

Використання архітектури back-illuminated sensor Buy a Raspberry Pi High Quality Camera – Raspberry Pi забезпечує підвищену квантову ефективність на 50%, що значно покращує роботу камери в умовах змінного освітлення та зменшує цифровий шум при високих значеннях ISO. Ця технологія зворотного підсвічування є особливо корисною для систем автоматичного спостереження, які повинні адаптуватися до різних світлових умов протягом доби.

Щодо продуктивності відеозапису, HQ Camera здатна записувати Full HD відео при 50 fps, що цілком достатньо для алгоритмів відстеження об'єктів у реальному часі та забезпечує плавність відтворення для операторів систем дистанційного керування. Високий динамічний діапазон та низький рівень шумів гарантують стабільну роботу при різних світлових умовах робочого середовища.

Для систем дистанційного спостереження HQ Camera забезпечує оптимальний баланс між професійними можливостями та доступністю. Висока роздільна здатність гарантує якісне цифрове збільшення, змінні об'єктиви дозволяють оптимізувати систему під конкретні умови експлуатації, а професійний сенсор забезпечує стабільну роботу в різноманітних умовах навколишнього середовища. Хоча модуль має більші розміри та вищу вартість порівняно з компактними альтернативами, його технічні можливості повністю виправдовують ці компроміси для професійних застосувань у сфері дистанційного моніторингу та керування.

					КС КРБ 123.196.00.00 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 ПРОГРАМУВАННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КС

3.1 Реалізація програмної частини

3.2 Код для модуля керування двигунами

Для кращого розуміння роботи розробленої системи, код якої написано на мові C [9]. На наступному рисунку представлено блок-схему основного алгоритму функціонування модуля керування двигунами (див. рис. 3.1). Блок-схема відображає послідовність виклику функцій після початкової ініціалізації, а також логіку безперервної обробки даних в режимі реального часу [10].

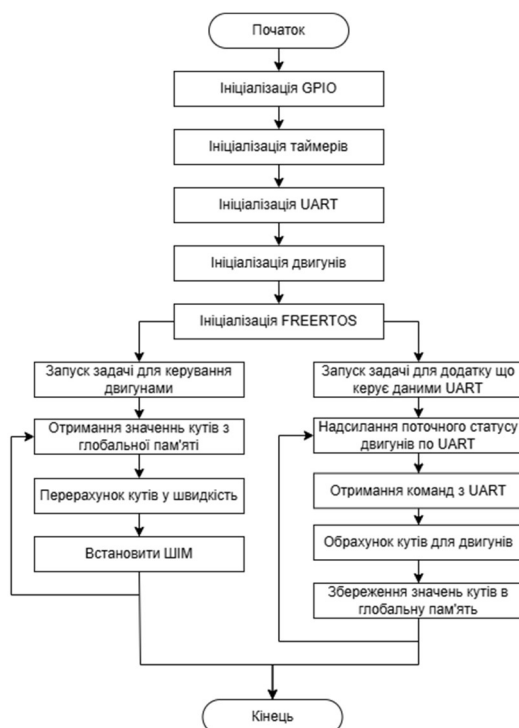


Рисунок 3.1 – Блок схема алгоритму роботи модуля керування двигунами

					КС КРБ 123.196.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Вітовцик П. В.			Програмування та тестування програмного забезпечення КС	Літ.	Арк.
Перевір.		Лецишин Ю.З.					38
Реценз.						ТНТУ, каф. КС, гр. CI-41	
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					

Алгоритм починається з початкової ініціалізації системи, яка включає послідовну настройку GPIO портів, таймерів для ШІМ-сигналів, UART для зв'язку з модулем веб-сервера, параметрів двигунів та запуск операційної системи FREERTOS [11].

Після завершення ініціалізації система переходить до основного циклу роботи, який умовно поділено на два паралельні напрями виконання. Перший напрям відповідає за запуск задачі для керування двигунами, що реалізується у вигляді безперервного циклу `while(true)`. В рамках цієї задачі послідовно виконуються наступні операції: отримання значень кутів з глобальної пам'яті, де зберігаються поточні команди керування, перерахунок з кутів в швидкість для адаптації команд до формату, зрозумілого системі керування двигунами, та встановлення ШІМ для безпосереднього керування швидкістю обертання двигунів. ШІМ проаналізовано за допомогою методу описаного у [12].

Другий паралельний напрям забезпечує запуск задачі для додатка, що керує додатком UART, також реалізованої у формі безперервного циклу `while(true)`. Ця задача включає надсилання поточного статусу двигунів по UART для інформування Raspberry Pi про стан системи, зчитування команд з UART для отримання нових інструкцій керування, обрахунок кутів для двигунів на основі отриманих команд, та збереження значень кутів в глобальну пам'ять для подальшого використання першою задачею.

Для забезпечення надійного обміну даними між модулем веб-сервера та модулем керування двигунами було реалізовано функцію зчитування даних з UART інтерфейсу [13], використовуючи метод описаний у [14]. Функція перевіряє відповідність розміру отриманих даних із запитуваним та повертає статус операції (див. рис. 3.2).

					КС КРБ 123.196.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

```

RET RPi_recive(U8 *pData, U16 size)
{
    U32 tsize = halUSART_recive(pUSART_app, pData, size);
    if (size == tsize)
        return RET_OK;
    else
        return RET_BUSY;
}

```

Рисунок 3.2 – Лістинг функції для читання UART

Керування швидкістю та напрямком обертання двигунів здійснюється через обробку повідомлень від керуючого пристрою. У відповідному кейсі команди MSG_SET_SPEED виконується розбір отриманих даних та встановлення параметрів для кожного з двигунів (див. рис. 3.3).

```

case MSG_SET_SPEED:
{
    data_pars_t data_pars[MAX_NUM_MOTORS];
    for(uint8_t i = 0; i < MAX_NUM_MOTORS; i++)
    {
        data_pars[i].direction = ((parcer_buf[2]
& (i+1)) ? 1 : 0);
        data_pars[i].speed = parcer_buf[i + 3];

        if(data_pars[i].speed == 0 &&
data_pars[i].direction == 0)
            asm("nop");
    }
    set_data_pars_motor(data_pars,
MAX_NUM_MOTORS);
}break;

```

Рисунок 3.3 – Лістинг для встановлення значення PWM на двигуни

Основну задачу керування двигунами виконує цикл, що безперервно зчитує дані з глобальної пам'яті та встановлює відповідні значення ШІМ [15] для керування швидкістю та напрямком обертання двигунів (див. рис. 3.4).

					КС КРБ 123.196.00.00 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

```

void StartDefaultTask(void const * argument)
{
    enc_states = get_encVAL();
    enc_stat = get_encVAL();
    for(;;)
    {
        setCWsw(M1_ID, enc_states[0].arr[0]);
        setCCWsw(M1_ID, enc_states[0].arr[1]);
        setCWsw(M2_ID, enc_states[1].arr[1]);
        setCCWsw(M2_ID, enc_states[1].arr[0]);
        osDelay(20);
    }
}

```

Рисунок 3.4 – Лістинг задачі для встановлення значень двигунів

Для забезпечення безперервної роботи UART інтерфейсу створено окрему задачу, яка виконує ініціалізацію та постійне обслуговування протоколу зв'язку з модулем веб-сервера (див. рис. 3.5).

```

void StartTask02(void const * argument)
{
    RPi_uart_app_init();
    for(;;)
    {
        RPi_uart_app_run();
    }
}

```

Рисунок 3.5 – Лістинг коду задачі для читання UART

Представлені програмні модулі у сукупності формують цілісну систему керування двигунами з двоступеневою архітектурою. Модуль керування двигунами виконує функції низькорівневого керування апаратними компонентами: отримує команди через UART інтерфейс, обробляє отримані дані для керування двигунами, генерує ШІМ сигнали для керування швидкістю та напрямком обертання двигунів. Багатозадачна архітектура на базі FreeRTOS [16] забезпечує паралельну обробку команд керування та виконання дій в реальному часі.

3.2.1 Програмування роботи модуля веб-сервера

На блок-схемі (див. рис. 3.6) зображено загальну логіку роботи системи, яка забезпечує передавання відео з камери, обробку даних та взаємодію з користувачем через сервер.

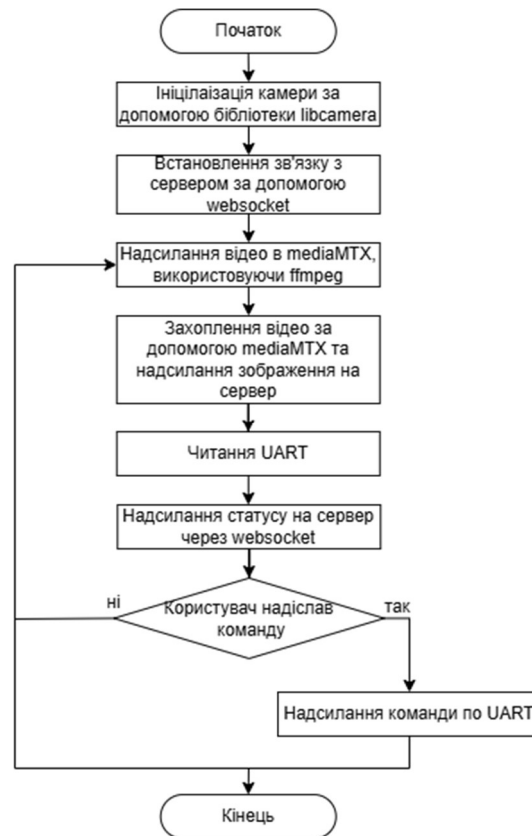


Рисунок 3.6 – Блок схема алгоритму роботи веб-сервера

Процес починається з ініціалізації камери за допомогою бібліотеки libcamera [17]. Після цього відбувається встановлення зв'язку з сервером за допомогою websocket [18].

Далі система переходить у нескінченний цикл обробки даних. У цьому циклі спочатку відбувається надсилання відео в mediaMTX, використовуючи ffmpeg, де відеопотік від камери передається через ffmpeg до медіа-сервера mediaMTX. Одночасно виконується захоплення відео та надсилання зображення на сервер, коли система отримує оброблений відеопотік та передає його на сервер, як аналог для передачі даних використовується [19].

Далі система здійснює читання UART для зчитування даних з послідовного порту, після чого відбувається надсилання статусу на сервер через websocket, де отримані UART дані передаються на сервер. Дана архітектура була обрана за допомогою [20].

Таким чином, реалізується повноцінна циклічна взаємодія між пристроєм, сервером і користувачем з обробкою відео- та UART-даних у режимі реального часу, з можливістю передачі команд користувача до пристрою через UART інтерфейс.

Для захоплення відеопотоку використовується бібліотека Picamera2 (див. рис. 3.7), яка забезпечує налаштування роздільної здатності, частоти кадрів та формату зображення. Конфігурація камери оптимізована для роботи в режимі реального часу.

```
with Picamera2() as picam2:
    picam2.preview_configuration.main.size = RESOLUTION
    picam2.preview_configuration.main.format = 'RGB888'
    picam2.configure("preview")
    picam2.set_controls({'FrameRate': FRAMERATE})
    picam2.configure()
    picam2.start()
```

Рисунок 3.7 – Лістинг коду для отримання відео з камери

Функція генерації кадрів забезпечує обробку відеопотоку та його підготовку для передачі через веб-інтерфейс [21]. Реалізовано механізм буферизації кадрів [22] та обробки випадків відсутності нових даних (див. рис. 3.8).

```

def generate_frame():
    global last_frame
    current_frame = get_newest_and_lclear_queue()
    if current_frame is None:
        if last_frame is None:
            blank_frame = np.zeros((480, 1280, 3),
dtype=np.uint8)
            current_frame = blank_frame
        else:
            current_frame = last_frame
    last_frame = current_frame
    _, jpeg = cv2.imencode('.jpg', current_frame)
    return jpeg.tobytes()

```

Рисунок 3.8 – Лістинг коду для зміни розміру зображення та надсилання через flask

Для асинхронної роботи з послідовним портом використовується бібліотека `aioserial` з налаштуванням параметрів зв'язку (див. рис. 3.9).

```

ser = aioserial.AioSerial(
    port='/dev/' + port,
    baudrate=115200,
    xonxoff=False,
    parity=aioserial.PARITY_NONE,
    stopbits=aioserial.STOPBITS_ONE,
    bytesize=aioserial.EIGHTBITS,
    timeout=0.1)

```

Рисунок 3.9 – Лістинг коду `ws_server` налаштування послідовного порту

Парсинг команд керування швидкістю здійснюється через розбір JSON повідомлень та перетворення їх у бінарний протокол для передачі мікроконтролеру. Функція включає обчислення напрямку руху та контрольної суми (див. рис. 3.10).

```

def parse_speed_msg_to_bytes(message: str):
    msg_header = 0xff
    msg_type = 0x00
    zooming_step_coefficient = 0.0019
    try:
        string_x, string_y =
message.strip('{ "comm": ["MV_HLD:DX:'] .strip('[]') .split(":DY:")
        float_x = float(string_x)
        float_y = float(string_y)
        dir_x = 0 if float_x > 0 else 1
        dir_y = 0 if float_y > 0 else 1
        speed_on_x = lookupSpeed(current_zoom, abs(float_x))
        speed_on_y = lookupSpeed(current_zoom, abs(float_y))
        speed_on_x = 255 if abs(speed_on_x) > 255 else
abs(speed_on_x)
        speed_on_y = 255 if abs(speed_on_y) > 255 else
abs(speed_on_y)
        speed_direction = int.from_bytes(bitarray([0, 0, 0,
0, 0, 0, dir_y, dir_x]), "little")
        _msg = [msg_header, msg_type, speed_direction,
speed_on_x, speed_on_y, 0, 0, 0, 0, 0, 0]
        crc8 = sum(_msg[1:]) % 256
        _msg.append(crc8)
        return _msg
    except Exception as e:
        print(e)
        return None

```

Рисунок 3.10 – Лістинг коду надсилання швидкості двигунів по UART

Raspberry Pi виконує роль інтелектуального центру обробки, який об'єднує відеопотік від камери з системою керування механікою. Модуль захоплення відео забезпечує отримання кадрів у високій роздільності, їх буферизацію та стиснення для передачі через мережу. Одночасно асинхронний серверний додаток обробляє команди користувача, отримані через WebSocket з'єднання, перетворює їх у бінарний протокол та передає мікроконтролеру через послідовний інтерфейс.

Веб-сервер на основі Node.js [23] та Express забезпечує обслуговування статичних файлів інтерфейсу та надає доступ до основної сторінки керування системою через HTTP протокол (див. рис. 3.11).

```

const express = require('express'),
      app = express(),
      http = require('http').createServer(app);
const host = '127.0.0.1';
const port = 8081;
app.use(express.static(__dirname));
app.get('/', (req, res) => {

res.sendFile(`${__dirname}/src/server_files/index.html`);
});
app.get('/test', (req, res) => {
  res.sendFile(`${__dirname}/src/server_files/webrtc_test
.html`);
});
http.listen(port, host, () => console.log(`Server listens
http://${host}:${port}`));

```

Рисунок 3.11 – Лістинг коду app.js для запуску сервера nodejs

Функція step формує команди керування рухом у форматі JSON та передає їх через WebSocket з'єднання до серверної частини системи для подальшої обробки (див. рис. 3.12).

```

function step(_dx, _dy) {
  let st = '{"comm":["MV_HLD:DX:' + _dx + ':DY:' + _dy +
  '"]}' ;
  ws2.send(st);
}

```

Рисунок 3.12 – Лістинг коду nodejs, для надсилання сигналів до ws_server

Для організації відеопотоку використовується протокол WHER (WebRTC-HTTP Egress Protocol), який забезпечує передачу SDP пропозицій, що вираховані за допомогою метода аналізу поведінки платформи [24] та встановлення WebRTC з'єднання через HTTP запити до медіа-сервера (див. рис. 3.13).

```

const sendOffer = (offer) => {
    fetch(new URL('whep', VIDEO_HREF) + window.location
.search, {
        method: 'POST',
        headers: {
            'Content-Type': 'application/sdp',
        },
        body: offer.sdp,
    })
    .then((res) => {
        switch (res.status) {
            case 201:
                break;
            case 404:
                throw new Error('stream not found');
            case 400:
                return res.json().then((e) => { throw new
Error(e.error); });
            default:
                throw new Error(`bad status code
${res.status}`);
        }
        sessionUrl = new URL(res.headers.get
('location'), VIDEO_HREF).toString();
        return res.text()
            .then((sdp) => onRemoteAnswer(sdp));
    })
    .catch((err) => {
        onError(err.toString());
    });
};

```

Рисунок 3.13 – Лістинг коду для надсилання відео з mediaMTX на сервер

Веб-інтерфейс на базі Node.js та WebRTC технологій створює зручний канал взаємодії з користувачем, забезпечуючи відображення відеопотоку в реальному часі та прийняття команд керування з мінімальними затримками. Результуючий функціонал системи дозволяє здійснювати точне дистанційне позиціонування камери з візуальним контролем через веб-браузер, забезпечуючи при цьому стабільну роботу всіх підсистем та надійну передачу даних між компонентами.

3.3 Тестування та запуск КС керування стабілізаційною платформою турелі

Підключення тестового стенда зображено на рисунку 3.14. Тестовий стенд представлений у відкритому вигляді без захисного корпусу, що дозволяє наочно продемонструвати всі компоненти системи та їх взаємодію. На стенді чітко видно основні елементи конструкції: модуль веб-сервера, модуль керування двигунами, безщіткові двигуни постійного струму для позиціонування камери та з'єднувальні кабелі. Таке компонування забезпечує зручний доступ для налагодження та демонстрації функціональності системи.



Рисунок 3.14 – Вигляд тестового стенда КС керування стабілізаційною платформою для наземної турелі

					КС КРБ 123.196.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

Для автоматичного запуску всіх необхідних програм розроблено спеціальні скрипти, які забезпечують ініціалізацію компонентів при завантаженні операційної системи. Лістинг скриптів для запуску сервісів наведено у додатку 1-4Б.

Після підключення живлення загораються індикатори на модулі керування двигунами та на Raspberry Pi, що свідчить про успішну ініціалізацію обладнання. Через декілька хвилин за допомогою сервісів Linux система автоматично запускає всі необхідні компоненти.

```
ffmpeg.service      loaded active running Runs ffmpeg for pass
main.service        loaded active running Starts turret diagnos
mediamtx.service    loaded active running Runs mediamtx server
rear_sight.service  loaded active running starting an turret re
video_processor.service loaded active running Processed video from
websocket_server.service loaded active running starting an turret we
ws_storage.service  loaded active running for autostarting web
```

Рисунок 3.15 – Сервіси написані для автозапуску системи

Процес ініціалізації включає запуск веб-сервера для обробки відеопотоку, WebSocket використовується для обміну командами з мікроконтролером та веб-сервера на базі Node.js для забезпечення користувацького інтерфейсу (див. рис. 3.16). А також досвід і методи провайдерів [25] застосовано у проєкті для аналізу стабільності з'єднання між клієнтською частиною та сервером управління.

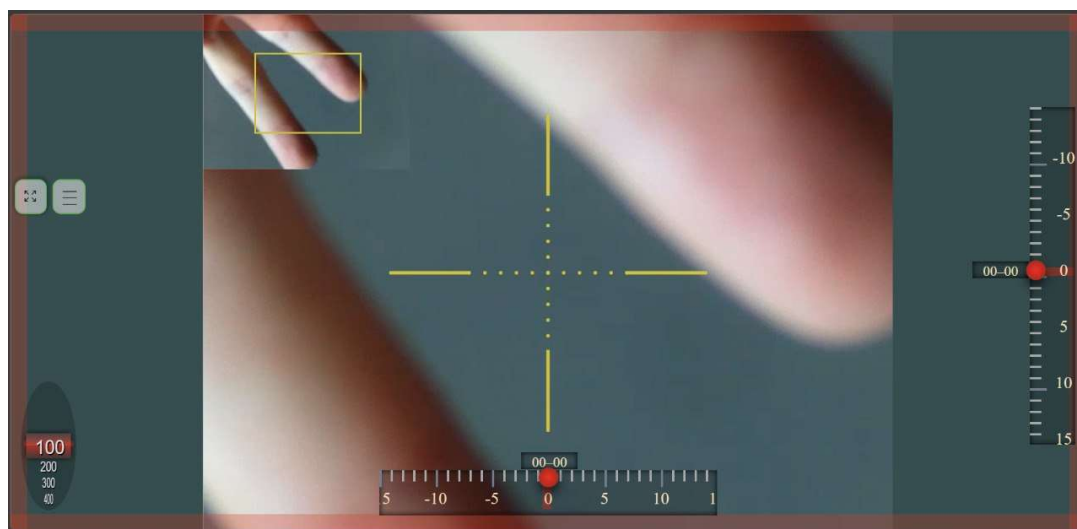


Рисунок 3.16 – Демонстрація опрацювання відео сервером

Веб-інтерфейс системи реалізовано у вигляді інтуїтивно зрозумілої сторінки, яка відображає відеопотік від камери у режимі реального часу та надає елементи керування для позиціонування. Інтерфейс містить область відображення відео, кнопки або джойстик для керування рухом камери в горизонтальній та вертикальній площинах, а також елементи керування цифровим зумом. Система підтримує декілька режимів цифрового зуму, що дозволяє користувачу наближати зображення для детального огляду об'єктів або віддаляти для отримання широкого кута огляду. Перемикання між режимами зуму здійснюється через відповідні кнопки клавіатури, а коефіцієнт зуму впливає на чутливість керування позиціонуванням для забезпечення більш точного наведення при збільшенні. Користувач може взаємодіяти з системою через стандартний веб-браузер, не потребуючи встановлення додаткового програмного забезпечення. Протестовано ефективність з'єднання між користувачем та веб-сервером, за допомогою методу описаного у [26].

Демонстрація роботи системи починається з моменту старту системи, коли камера та двигуни встановлені у початковому положенні (див. рис. 3.17).

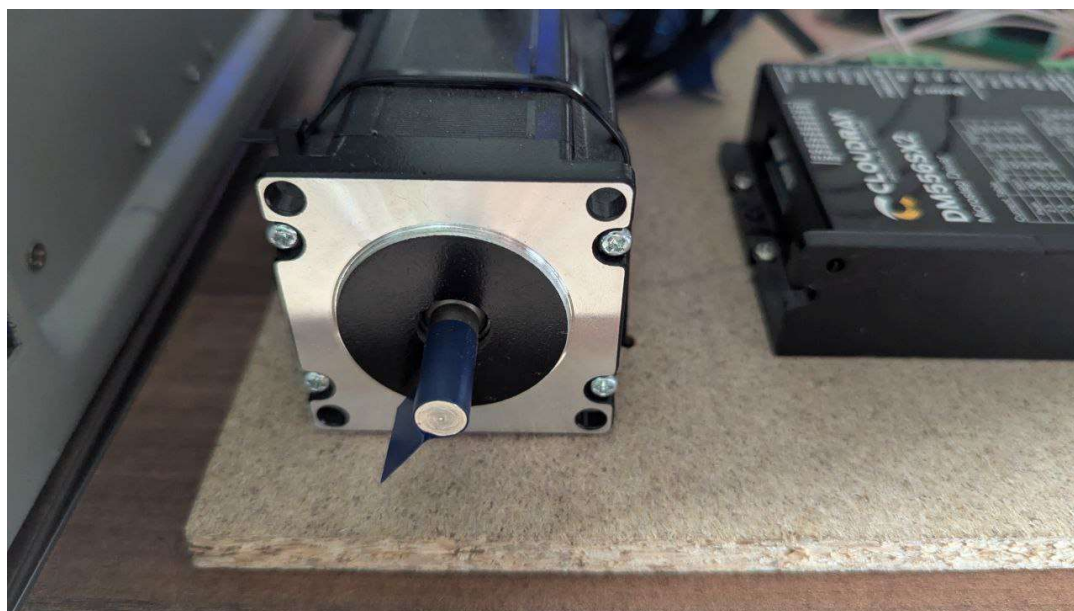


Рисунок 3.17 – Двигун у стартовому положенні

Користувач через веб-інтерфейс встановлює кут повороту вліво,

					КС КРБ 123.196.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

активуючи відповідний елемент керування (див. рис. 3.18). Команда миттєво передається через WebSocket з'єднання до серверної частини системи.

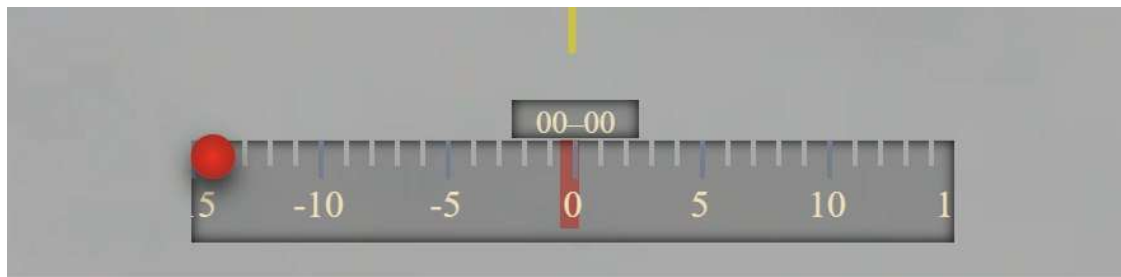


Рисунок 3.18 – Виставлення кута повороту вліво користувачем

Після обробки команди сервером та її передачі мікроконтролеру через UART інтерфейс, серводвигун виконує поворот у задану позицію (див. рис. 3.19). Система плавно переміщується у ліву сторону, що підтверджує коректну роботу всього ланцюга передачі команд.

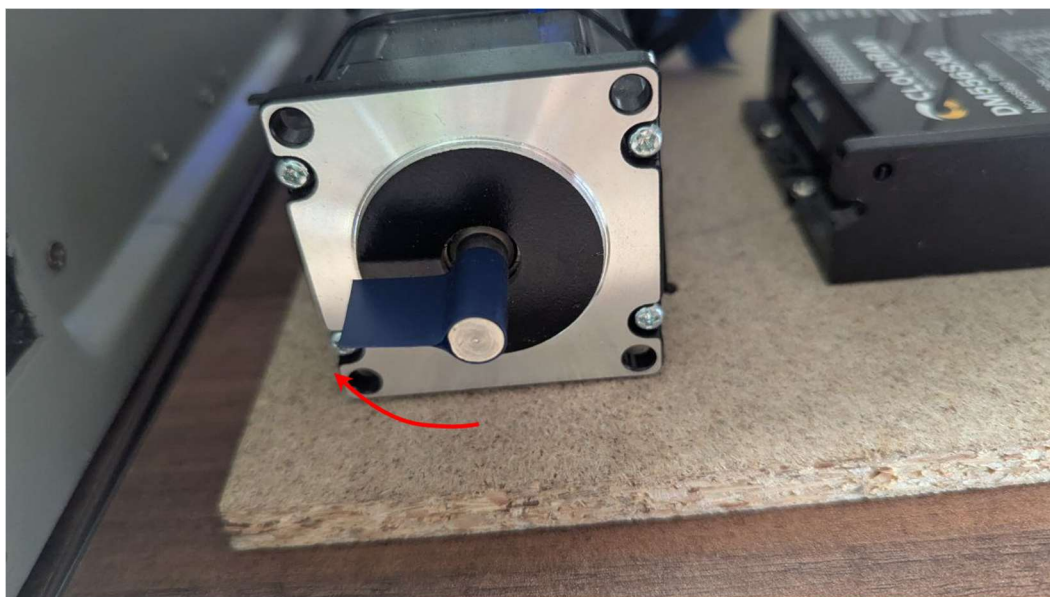


Рисунок 3.19 – Двигун у лівому положенні після надсилання команди

Аналогічно демонструється робота системи при повороті у протилежному напрямку. Користувач через веб-інтерфейс ініціює команду повороту вправо (див. рис. 3.20).

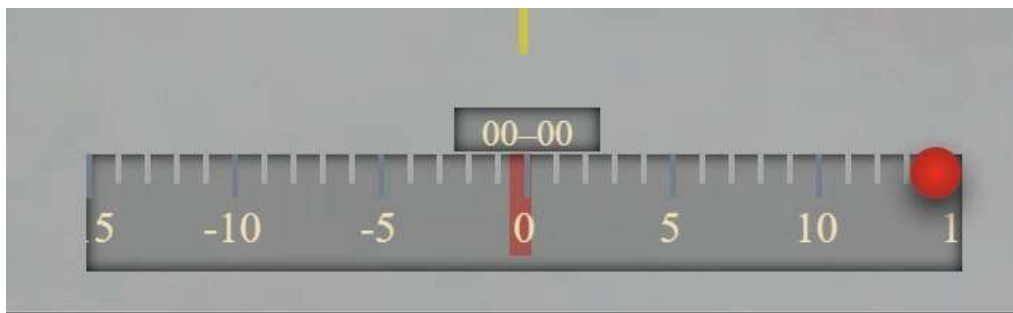


Рисунок 3.20 – Виставлення кута повороту вправо користувачем

Система швидко реагує на команду користувача, і серводвигун переводить камеру у праве положення (див. рис. 3.21), демонструючи точність позиціювання та стабільність роботи механічної частини.

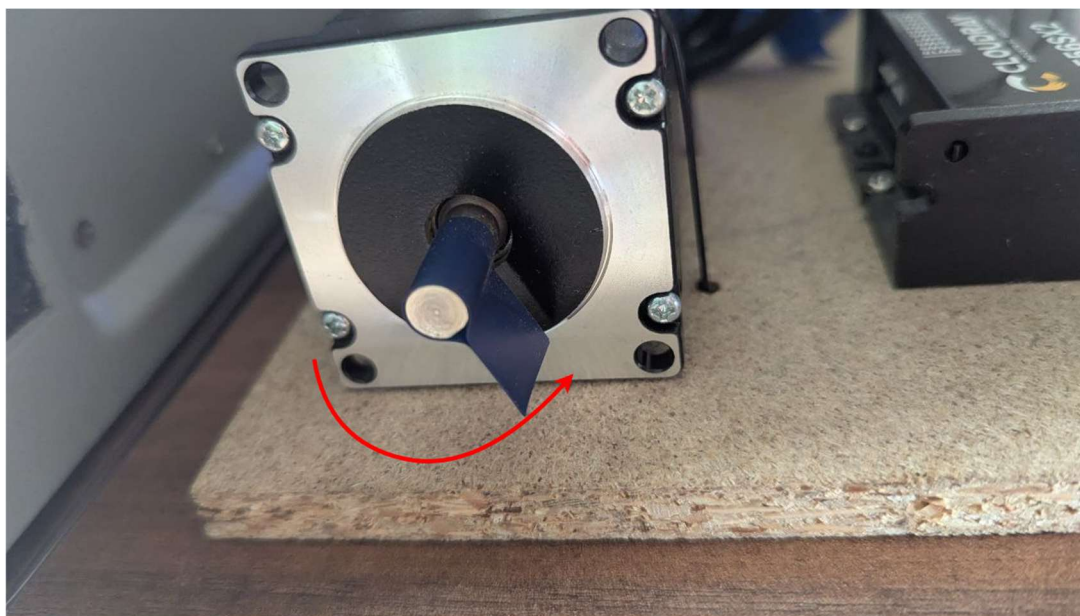


Рисунок 3.21 – Двигун у правому положенні після надсилання команди

Проведені тести демонструють успішну реалізацію всіх ключових функцій системи керування стабілізаційною платформою. Система показала стабільну роботу в режимі реального часу з мінімальними затримками між подачею команд користувачем та їх виконанням механічними компонентами.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Заходи щодо захисту від ураження електричним струмом

У сучасних високотехнологічних системах, таких як комп'ютеризована стабілізаційна платформа для наземної турелі, безпека експлуатації електронного та електротехнічного обладнання є критично важливою. Одним з основних ризиків є ураження електричним струмом, яке може виникнути через несправності обладнання, помилки в експлуатації або недоліки проєктування. Тому впровадження ефективних заходів електробезпеки є обов'язковим етапом розробки та використання системи.

Система включає різні рівні напруг, а саме 5В, 12В, 25В постійного струму – для живлення контролерів, сенсорів, мікропроцесорів, приводів стабілізації.

Усі елементи відповідають вимогам безпеки згідно з ДСТУ EN 61140:2019 "Захист від ураження електричним струмом".

Захист від прямого дотику (первинного):

- запобігання контакту зі струмопровідними частинами за допомогою ізоляції, корпусів, захисних кожухів;
- використання малих напруг для критичних компонентів.

Захист від непрямого дотику (вторинного):

- заземлення та зрівнювання потенціалів металевих частин, які можуть опинитися під напругою при пробіі ізоляції;
- автоматичне вимкнення (узо, запобіжники) у разі несправностей;
- подвійна або посилена ізоляція провідників, особливо в місцях, доступних для персоналу;

					КС КРБ 123.196.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Вітовщик П.В.			Безпека життєдіяльності, основи охорони праці		
Перевірив		Лецишин Ю.З.					
Консульт.		Пилипець М.І.					
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Аркушів
						53	
					ТНТУ, каф. КС, гр. СІ-41		

Забезпечення електробезпеки системи реалізується через комплекс технічних рішень та організаційних заходів вказаних у [27]. З технічної точки зору система використовує промислові роз'єми з ключами для запобігання помилковому підключенню, а також кабелі з подвійною ізоляцією та надійною фіксацією для уникнення механічних пошкоджень. Особливу увагу приділено моніторингу електричних параметрів через аналого-цифрові перетворювачі для виявлення перенапруг, коротких замикань та перегріву, при цьому у разі виникнення аварійних ситуацій система автоматично переходить у безпечний режим або повністю відключається. Критично важливим елементом є єдиний контур заземлення для усієї системи, що запобігає виникненню небезпечних різниць потенціалів між компонентами.

Організаційні заходи включають обов'язковий інструктаж та навчання персоналу з правил електробезпеки, а також проведення регламентного обслуговування з перевіркою ізоляції, заземлення та роботи захисних пристроїв. Доступ до електрообладнання суворо обмежений і дозволяється лише кваліфікованим фахівцям з відповідною підготовкою. Для забезпечення контролю та відстеження стану системи ведеться спеціальний журнал електробезпеки з фіксацією всіх проведених перевірок та виявлених несправностей.

Електробезпека в системі керування стабілізаційною платформою забезпечується комплексом технічних та організаційних заходів [28]. Відповідність нормативним вимогам, надійне заземлення, автоматичний захист та регулярне обслуговування дозволяють мінімізувати ризики для персоналу та обладнання, забезпечуючи стабільну та безпечну роботу системи.

4.2 Заходи, що покращують умови праці оператора

Оператор комп'ютеризованої системи керування стабілізаційною платформою для наземної турелі виконує складну та відповідальну роботу в

					КС КРБ 123.196.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

умовах підвищеного психоемоційного навантаження, потреби високої концентрації уваги та швидкої реакції. У зв'язку з цим, створення комфортних, безпечних та ергономічно правильних умов праці є ключовим фактором збереження здоров'я оператора, зниження стомлюваності, підвищення ефективності виконання завдань та запобігання помилкам у роботі описаних у [29].

Робоче місце оператора організовано з урахуванням принципів ергономіки вказаних у [30]:

- висота столу і стільця повинна відповідати антропометричним даним користувача. Стілець має бути регульованим по висоті, з опорою для попереку;
- монітор має розміщуватись на рівні очей або трохи нижче (15–20°), на відстані 50–70 см. Це зменшує напруження очей і шийного відділу хребта;
- клавіатура і мишка на одному рівні з ліктями оператора;
- поверхня неслизька, а руки мають опору;
- кабелі від пристроїв мають бути укладені в кабель-канали, щоб уникнути сплутування та запобігти випадковому зачепленню;
- усі елементи інтерфейсу джойстики, кнопки, панелі розміщені в межах зони комфорту оператора, без потреби надмірних рухів;
- найбільш використовувані елементи управління під рукою.

Бажано передбачити можливість змінювати положення тіла під час тривалого чергування (через регульований стіл, або альтернативне місце для короткотривалої роботи стоячи), що позитивно впливає на кровообіг та знижує ризик опорно-рухових порушень.

Мікроклімат має суттєвий вплив на працездатність оператора. Забезпечуються наступні параметри згідно з ДСН 3.3.6.042-99 "Санітарні норми мікроклімату виробничих приміщень":

- температура повітря: 20–24°C (залежно від сезону);
- вологість: 40–60%;

					КС КРБ 123.196.00.00 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

- швидкість повітря: не більше 0.2 м/с;
- регулярне провітрювання або використання припливно-витяжної вентиляції.

Для уникнення локального перегріву електронного обладнання та підтримання здорового середовища використовується система кондиціонування із фільтрацією повітря. Вентиляційні отвори не направлені безпосередньо на робоче місце оператора.

Правильне освітлення є одним з найважливіших факторів, що впливає на зорову систему оператора та загальне самопочуття під час роботи з системою. Загальне освітлення робочого місця повинно бути рівномірним, без пульсацій, з рівнем освітленості не менше 300 лк для забезпечення комфортних умов праці. Локальне освітлення обладнується регульованим рівнем яскравості та розміщується таким чином, щоб не створювати відблисків на моніторі або клавіатурі, що може призвести до втомлюваності очей оператора, згідно з ДБН В.2.5-28-2006 "Організація робочих місць".

Для освітлення використовуються енергоефективні LED лампи з температурою світла в діапазоні 4000–5000 К, що відповідає нейтрально-білому спектру та найкраще підходить для тривалої роботи з відеоінформацією. Такий спектр освітлення забезпечує оптимальний контраст та чіткість зображення без надмірного навантаження на зорову систему.

Природне освітлення регулюється за допомогою жалюзів на вікнах, що дозволяє запобігти засліпленню оператора прямими сонячними променями та уникнути створення контрастних тіней на робочій поверхні. Така система освітлення забезпечує стабільні та комфортні умови роботи протягом усього робочого дня незалежно від зовнішніх погодних умов.

Тривала робота в умовах шуму (>50 дБ) негативно впливає на психоемоційний стан, викликає втому, знижує точність дій оператора.

Основними джерелами шуму в системі є вентилятори охолодження стабілізаційної платформи, які генерують постійний шум для відведення тепла від електронних компонентів, та серводвигуни приводу, що створюють

					КС КРБ 123.196.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

акустичні коливання під час позиціювання камери. Додатковими джерелами є комутатори та блоки живлення, які можуть генерувати високочастотні звуки через роботу внутрішніх компонентів. Сукупний вплив цих джерел може впливати на комфорт оператора, тому важливо передбачати заходи для зниження шумового навантаження.

Для мінімізації акустичного впливу використовуються безшумні вентилятори з низьким рівнем акустичного тиску та оптимізованою аеродинамікою лопатей. Основне обладнання встановлюється в звукоізовьованому корпусі з багатошаровими панелями, а найбільш шумні елементи розміщуються в окремому технічному відсіку, ізовьованому від робочої зони оператора.

Для зниження вібраційних шумів застосовуються гумові антивібраційні прокладки під механічними компонентами системи. Додатково на стінах приміщення встановлюються поглинаючі акустичні панелі, які створюють комфортні акустичні умови для тривалої роботи оператора з системою керування.

Програмний інтерфейс оператора витриманий у нейтральних тонах. Контрастні кольори використовуються тільки для сигналів тривоги. Застосування "нічного режиму" в умовах обмеженої видимості знижує навантаження на очі.

					КС КРБ 123.196.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було розроблено та реалізовано повнофункціональну систему дистанційного керування камерою на базі Raspberry Pi з можливістю передачі відеопотоку в режимі реального часу. Досягнуті результати:

1) Розроблено програмне забезпечення серверної частини з використанням технологій WebSocket для двостороннього обміну даними, асинхронної обробки UART команд та інтеграції з медіа-сервером mediaMTX для ефективного передачі відеопотоку.

2) Створено веб-інтерфейс користувача на базі Node.js та WebRTC технологій з інтуїтивними елементами керування, підтримкою декількох режимів цифрового зуму та адаптивною чутливістю керування залежно від рівня збільшення.

3) Впроваджено систему автоматичного запуску всіх сервісів через systemd демони Linux, що забезпечує автономну роботу системи після подачі живлення без необхідності ручного втручання.

4) Проведено комплексне тестування системи, яке підтвердило коректну роботу всіх підсистем, мінімальні затримки передачі відеопотоку, точність позиціонування камери та стабільність роботи в тривалому режимі експлуатації.

5) Продемонстровано практичну придатність розробленої системи для вирішення завдань дистанційного спостереження, моніторингу та керування в різних галузях застосування, де потрібен віддалений візуальний контроль з можливістю активного позиціонування камери.

					КС КРБ 123.196.00.00 ПЗ	Арк.
						58
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. Segway Robotics. Gimbal Design and Control Principles. Technical White Paper, 2020. – [Електронний ресурс]. – Режим доступу: <https://www.segwayrobotics.com/resources/whitepapers/gimbal-design>
3. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп'ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ. 2019. 150 с.
4. Tymkiv P., Leshchyshyn Y. Algorithm Reliability of Kalman Filter Coefficients Determination for Low-Intensity Electroretinosignal. IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM). 2019. P.1-5.
5. Leshchyshyn Y., Semchyshyn O. Periodically correlated heart rate variability detection by Neyman - Pearson criterion. 9th International Conference - The Experience of Designing and Applications of CAD Systems in Microelectronics. 2007. P. 139–140.
6. Voloskyi V., Leshchyshyn Y., Romanyshyn N., Palamar A., Tarasenko L. Method and algorithm for efficient cell balancing in the lithium-ion battery control system. CEUR Workshop Proceedings, The 1st International Workshop on Bioinformatics and Applied Information Technologies (BAIT 2024), Zboriv, Ukraine, October 02-04, 2024. Vol. 3842. P. 258-267.
7. Raspberry Pi Foundation. Official Raspberry Pi Documentation. – [Електронний ресурс]. – Режим доступу: <https://www.raspberrypi.com/documentation/>

					КС КРБ 123.196.00.00 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

8. Джонсон Т. BLDC Motor Control with Microcontrollers. Texas Instruments Application Report, 2020. – [Електронний ресурс]. – Режим доступу: <https://www.ti.com/lit/an/bldc-control>
9. Гладкий Ю. В. Цифрові системи керування: проектування та моделювання. – Київ: Наук. думка, 2019. – 278 с.
10. Olexa M. Programming Embedded Systems with C and GNU Development Tools. – O'Reilly Media, 2022. – 440 p.
11. Мельничук В. В. Мікроконтролери STM32 у системах реального часу. – Вінниця: ВНТУ, 2021. – 192 с.
12. Leschyshyn Y., Scherbak L., Nazarevych O., Gotovych V., Tymkiv P., Shymchuk G. Multicomponent Model of the Heart Rate Variability Change-point. IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH). 2019. P. 110–113.
13. Агафонов С. Ю. UART, I2C та SPI: серійні інтерфейси у мікроконтролерах. – М.: ДМК Пресс, 2020. – 256 с.
14. Kharchenko A., Bodnarchuk I., Yatecsyn V. The Method for Comparative Evaluation of Software Architecture with Accounting of Trade-offs. American Journal of Information Systems. 2014. Vol. 2, No. 1. P. 20-25.
15. GitHub. STM32 Motor Control SDK. — [Електронний ресурс]. – Режим доступу: <https://github.com/STMicroelectronics/x-cube-mcsdk>
16. Луцків А., Лупенко С., Пасічник В. Паралельні та розподільнені обчислення. Навчальний посібник. Львів: Видавництво «Магнолія 2006», 2024. 566 с.
17. Upton E., Halfacree G. Raspberry Pi User Guide. – 4th ed. – Wiley, 2016. – 312 p.
18. Мазур О. С. Технології інтернету речей та вбудованих систем. – Харків: ХНУРЕ, 2021. – 276 с.
19. Лецишин Ю., Герасименко А., Герасименко О. Комп'ютерна система моніторингу радіозв'язку від БПЛА // Матеріали 12-ї науково-

технічної конференції „Інформаційні моделі, системи та технології “. Тернопіль, 2024. С.140.

20. Глушко М. С. Веб-технології для вбудованих систем. – Вінниця: ВНТУ, 2022. – 174 с.

21. Flask Documentation. Welcome to Flask Documentation (2.3.x). – [Електронний ресурс]. – Режим доступу: <https://flask.palletsprojects.com/en/2.3.x/>

22. Кардаш І., Лецишин Ю., Варавін А. Критерії ефективності роботи для задачі моніторингу локальної мережі / Матеріали XI науково-технічної конференції «Інформаційні моделі, системи та технології» Тернопіль: ТНТУ. 2023. С.154

23. IEEE Xplore. Survey of Web-Based Control Systems. – [Електронний ресурс]. – Режим доступу: <https://ieeexplore.ieee.org/document/7422156>

24. Method of Gas Consumption Change-point Detection based on Seasonality Multicomponents Model / Leshchyshyn Yu., Nazarevych O., Lupenko S., Gotovych V., Shymchuk G., Shabliy N. 2020 10th International Conference on Advanced Computer Information Technologies (ACIT). 2020. P. 152-155.

25. Залісковий Ю., Лецишин Ю., Варавін А. Методи проведення моніторингу і аналізу мережевої інфраструктури Інтернет провайдерами, Матеріали XI науково-технічної конференції «Інформаційні моделі, системи та технології». Тернопіль: ТНТУ. 2023. С.152

26. Ярмусь О., Лецишин Ю., Варавін А. Методи та засоби моделювання швидкості Wi-Fi мережі, Матеріали XI науково-технічної конференції «Інформаційні моделі, системи та технології» Тернопіль: ТНТУ. 2023. С.11

27. Правила охорони праці під час роботи з обладнанням, що працює під напругою. – Наказ Міністерства соціальної політики України № 1404 від 19.12.2016.

28. Бойчук М. Т. (ред.). Безпека життєдіяльності : навч. посіб. – К. : Лібра, 2018. – 320 с.

					КС КРБ 123.196.00.00 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

29. Кулагін Ю. П., Федоренко М. П. Основи охорони праці : навч. посіб. – К. : Каравела, 2019. – 320 с.
30. ISO 6385:2016. Ergonomic principles in the design of work systems. – International Organization for Standardization, 2016.

					КС КРБ 123.196.00.00 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

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.

“ ____ ” _____ 2025 р

Комп'ютеризована система керування стабілізаційною платформою для
наземної турелі

ТЕХНІЧНЕ ЗАВДАННЯ

на 9 листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

«ВИКОНАВЕЦЬ»

Керівник кваліфікаційної роботи

Студент групи СІ-41

_____ к.т.н., доц. Лещишин Ю.З.

_____ Вітовщик П.В.

« ____ » _____ 2025 р.

« ____ » _____ 2025 р.

Тернопіль 2025

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютеризована система керування стабілізаційною платформою для наземної турелі».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.196.00.00

1.2 Виконавець

Студент групи СІ-41, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Вітовщик П.В.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/7-53 від 27.01.2025 р.)

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 27.01.2025 р.

Плановий термін завершення виконання кваліфікаційної роботи – 23.06.2025 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'ютеризован система що розробляється призначенна для:

- Автоматична стабілізація положення платформи при зовнішніх впливах;
- Дистанційне керування через веб-браузер з будь-якого пристрою, підключеного до мережі;
- Реалізація плавних рухів платформи в двох площинах;
- Моніторинг поточного стану системи в режимі реального часу;
- Відтворення заданих позицій платформи.

2.2 Мета створення системи

Метою кваліфікаційної роботи є розробка та реалізація комп'ютеризованої системи керування стабілізаційною платформою турелі, що поєднує веб-інтерфейс для дистанційного управління з мікроконтролерним забезпеченням для високоточного позиціонування та стабілізації в режимі реального часу.

2.3 Характеристика об'єкту

Розробляювану комп'ютерну систему призначено для використання в системах відеоспостереження, охоронних системи та оборонній промисловості.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

3.1.1 Вимоги до структури та функціонування системи

Система керування стабілізаційною платформою повинна забезпечувати високоточне позиціонування та стабілізацію з можливістю дистанційного керування через веб-інтерфейс.

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

Основна вимога, яка ставиться до способів та засобів інформаційного обміну – це їх узгодженість та швидкодія.

3.1.3 Вимоги до режимів функціонування системи

Для системи керування стабілізаційною платформою турелі визначено два основні режими функціонування: нормальний та аварійний режими роботи.

Нормальний режим функціонування є основним робочим режимом системи, при якому забезпечується повний функціонал керування. В цьому режимі веб-інтерфейс забезпечує дистанційне керування, Raspberry Pi обробляє команди користувача та здійснює високорівневі обчислення, а

мікроконтролер STM32 генерує ШІМ-сигнали для точного позиціонування двигунів. Система підтримує всі заявлені характеристики точності, швидкості відгуку та функціональні можливості.

Аварійний режим функціонування активується при відмові одного або декількох компонентів програмного чи технічного забезпечення системи. В цьому режимі система автоматично переходить до базових налаштувань та обмежує функціонал для забезпечення безпечної роботи. При втраті зв'язку з веб-інтерфейсом платформа фіксується в поточному положенні, при збої Raspberry Pi мікроконтролер STM32 продовжує підтримувати стабілізацію за останніми отриманими параметрами, а при відмові одного з двигунів система блокує рух по відповідній осі.

3.1.4 Вимоги по діагностуванню системи

Для діагностування системи використовуються інструменти діагностування основних процесів системи, які вмонтовані в операційну систему і програмне забезпечення, а також засоби для діагностики апаратного забезпечення.

Інструменти повинні забезпечувати зручний інтерфейс для можливості перегляду діагностичних подій, моніторингу процесу виконання програм.

3.1.5 Перспективи розвитку, проєктування системи

Дана система може бути розширена завдяки використанню додаткових програмних та апаратних компонентів. Модульна архітектура системи дозволяє інтегрувати різноманітне обладнання без суттєвих змін базової структури.

3.2 Показники призначення

Система повинна передбачати можливість масштабування. Можливості масштабування повинні забезпечуватися засобами використовуваного базового програмного і технічного забезпечення.

3.2.1 Вимоги до надійності

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

- при збоях в системі електропостачання апаратної частини;
- при помилках в роботі апаратних засобів;
- при помилках, пов'язаних з програмним забезпеченням (ОС і драйвери пристроїв).

Для захисту апаратури від стрибків напруги і 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 мати занулення або захисне заземлення .

Система електроживлення повинна забезпечувати захисне вимикання при перевантаженнях і коротких замиканнях в колах навантаження, а також аварійне ручне вимикання.

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

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

Мікроклімат в приміщеннях повинен відповідати нормам виробничого мікроклімату по ДСН 3.3.6.042-99:

- температуру повітря в межах від +10°C до +35°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.

Періодичне технічне обслуговування і тестування технічних засобів повинні включати обслуговування і тестування всіх використовуваних засобів, датчики, контролери, системи передачі даних, пристрої безперебійного живлення.

На підставі результатів тестування технічних засобів повинні проводитися аналіз причин виникнення виявлених дефектів і прийматися заходи по 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, що діє, “Автоматизовані системи. Захист від несанкціонованого доступу до інформації. Класифікація автоматизованих систем”.

Компоненти підсистеми захисту від НСД повинні забезпечувати:

- ідентифікацію користувача;
- перевірку повноважень користувача при роботі з системою;
- розмежування доступу користувачів.

Рівень захищеності від несанкціонованого доступу засобів обчислювальної техніки, що здійснюють обробку конфіденційної інформації, повинен відповідати вимогам класу захищеності згідно вимогам документу “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, при виникненні аварійних ситуацій повинна бути збережена на резервних носіях.

3.4.2 Вимоги по стандартизації і уніфікації

Система повинна відповідати вимогам ергономіки і зручності користування за умови комплектування високоякісним обладнанням (ЕОМ, монітор і інше обладнання), що має необхідні сертифікати відповідності і безпеки.

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

- забезпечення точного позиціонування платформи;
- забезпечення стабілізації в режимі реального часу;
- забезпечення дистанційного керування через веб-інтерфейс;
- забезпечення високої швидкості відгуку на команди;
- забезпечення діагностики та моніторингу стану системи.

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 модуля керування двигунами.
- в) Блок схема роботи модуля веб-сервера.
- г) Схема електрична принципова.
- д) Схема електрична монтажна (з'єднань).

*Примітка: У комплект документації можуть вноситися зміни та доповнення в процесі розробки.

5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка технічного завдання.	27.01.2025р. – 02.02.2025р.
2	Робота над розділом аналіз технічного завдання на розробку кс керування стабілізаційною платформою турелі	02.02.2025р. – 10.03.2025р.
3	Робота над основним розділом проектування КС керування стабілізаційною платформою турелі	11.03.2025р. – 01.05.2025р.
4	Робота над основним розділом програмування та тестування програмного забезпечення КС	02.05.2025р. – 28.05.2025р.
5	Безпека життєдіяльності, основи охорони праці	29.05.2025р. – 02.06.2025р.
6	Оформлення пояснювальної записки і графічного матеріалу	03.06.2025р. – 09.06.2025р.
7	Перевірка на академічний плагіат, перевірка керівником та консультантами	9.06.2025р. – 15.06.2025р.
8	Попередній захист кваліфікаційної роботи бакалавра	16.06.2025р. – 22.06.2025р.
9	Захист кваліфікаційної роботи бакалавра	23.06.2025р.

6 Додаткові умови виконання кваліфікаційної роботи

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

Додаток Б

Лістинг програм

Лістинг 1Б – Код сервісу websocket_server.service

```
[Unit]
Description=starting websocket server
After=network-online.target
Wants=network-online.target

[Service]
Type=simple
Restart=on-failure
Environment=PYTHONPATH=/home/pi/.local/lib/python3.9/site-packages
WorkingDirectory=/home/pi/websocket_server
ExecStart=python3 ws_server.py
StandardOutput=journal
StandardError=journal
Environment=PYTHONUNBUFFERED=1

[Install]
WantedBy=multi-user.target
```

Лістинг 2Б – Код сервісу main.service

```
[Unit]
Description=Starts diagnostic and services
After=network-online.target
Wants=network-online.target
StartLimitIntervalSec=60
StartLimitBurst=6
StartLimitAction=reboot

[Service]
Type=simple
StandardOutput=journal
```

```
StandardError=journal
Environment=PYTHONUNBUFFERED=1
WorkingDirectory=/home/prog _main
ExecStart=python3 main.py
ExecStopPost=systemctl stop rear_sight.service
ExecStopPost=systemctl stop websocket_server.service
ExecStopPost=systemctl stop video_processor.service
TimeoutSec=10
Restart=on-failure
[Install]
WantedBy=multi-user.target
```

Лістинг 3Б – Код сервісу rear_sight.service

```
[Unit]
Description=starting an rear sight node js server
[Service]
Type=simple
Restart=on-failure
WorkingDirectory=/home/pi/rear_sight
ExecStart=node app.js
[Install]
WantedBy=multi-user.target
```

Лістинг 4Б – Код сервісу video_processor.service

```
[Unit]
Description=Processed video from Raspberry Pi Camera
After=syslog.target
After=network.target
[Service]
Type=simple
WorkingDirectory=/home/pi/video_processor
Environment=PYTHONPATH=/home/pi/.local/lib/python3.9/site-
```

packages

```
ExecStart=python3 video_processor.py
Restart=on-failure
RestartSec=5
[Install]
WantedBy=multi-user.target
```

Лістинг 5Б – Код main.c для stm32

```
void stepConfig() {
    M1.motor_ID = M1_ID;
    M1.dir_now = MDIR_CW;
    M1.turn_state = M_STATE_SPEED_ZERO;
    M1.end_cnt = START_MAX_END_CNT;
    M1.max_end_cnt = START_MAX_END_CNT;
    M1.GPIO_Dir_Port = M1_DIR_GPIO_Port;
    M1.GPIO_Dir_Pin = M1_DIR_Pin;
    M1.GPIO_End_CW_Port = M1_ENC2_GPIO_Port;
    M1.GPIO_End_CW_Pin = M1_ENC2_Pin;
    M1.GPIO_End_CCW_Port = M1_ENC1_GPIO_Port;
    M1.GPIO_End_CCW_Pin = M1_ENC1_Pin;
    stepInit(&M1, M1.motor_ID);
    M2.motor_ID = M2_ID;
    M2.dir_now = MDIR_CW;
    M2.turn_state = M_STATE_SPEED_ZERO;
    M2.end_cnt = START_MAX_END_CNT;
    M2.max_end_cnt = START_MAX_END_CNT;
    M2.GPIO_Dir_Port = M2_DIR_GPIO_Port;
    M2.GPIO_Dir_Pin = M2_DIR_Pin;
    M2.GPIO_End_CW_Port = M2_ENC1_GPIO_Port;
    M2.GPIO_End_CW_Pin = M2_ENC1_Pin;
    M2.GPIO_End_CCW_Port = M2_ENC2_GPIO_Port;
    M2.GPIO_End_CCW_Pin = M2_ENC2_Pin;
    stepInit(&M2, M2.motor_ID);
}
```

```

int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_TIM7_Init();
    MX_TIM14_Init();
    MX_TIM6_Init();
    MX_CRC_Init();
    MX_USART1_UART_Init();
    MX_USART2_UART_Init();
#ifdef UART5V
    sUSART_RPi.instance = RPI_USE_UART;
#else
    sUSART_RPi.instance = RS485_USE_UART;
#endif
    halUSART_init(&sUSART_RPi);
    RPi_app_init(&sUSART_RPi);

#ifdef UART5V
    NVIC_SetPriority(USART2_LPUART2_IRQn, 0); // priority
(0...15), 0 - highest.
    NVIC_EnableIRQ(USART2_LPUART2_IRQn);
#else
    NVIC_SetPriority(USART1_IRQn, 0); // priority (0...15),
0 - highest.
    NVIC_EnableIRQ(USART1_IRQn);
#endif

    stepConfig();
    HAL_GPIO_WritePin(M_EN_GPIO_Port, M_EN_Pin, 0);
    HAL_TIM_Base_Start_IT(&htim14); // запуск таймера
    HAL_TIM_Base_Start_IT(&htim6); // запуск таймера
    HAL_TIM_Base_Start_IT(&htim7); // запуск таймера
    __enable_irq();

```

```

    MX_FREERTOS_Init();
    osKernelStart();
    while (1) {
        }
    }
}

```

Лістинг 6Б – Код програми RPI_app.c

```

#include <stdint.h>
#include <string.h>
USART_t *pUSART_app;
modbus_answ_t *enc_state;
StepMotors *Motors_p;
int32_t current_angle1 = 0;
int32_t prev_angle1 = 0;
int32_t current_angle2 = 0;
int32_t prev_angle2 = 0;
int32_t shif_angle1 = 0;
int32_t shif_angle2 = 0;
typedef enum
{
    RX_CHECK_DATA = 0, RX_READ_DATA, RX_PARS_DATA
} state_machine_rx_t;

void RPi_app_init(USART_t *pUSART)
{
    pUSART_app = pUSART;
}

RET RET_USART_send(USART_t *pUSART, U8 *pData, U16 size)
{
    return halUSART_transmit(pUSART, pData, size);
}

```

```

RET RET_USART_recive(USART_t *pUSART, U8 *pData, U16 size)
{
    U32 tsize = halUSART_recive(pUSART, pData, size);
    if (size == tsize)
        return RET_OK;
    else
        return RET_BUSY;
}

void RPi_uart_app_init()
{
    Motors_p = get_StepMotors_pointer();
    enc_state = get_encVAL();
}

int32_t adjust_angle(int32_t angle, int32_t *prev_angle,
int32_t *shift_ang) {
    const int32_t threshold = 120000;
    int32_t angle_dif = angle - *prev_angle;
    if (angle_dif > threshold) {
        *shift_ang -= 360000;
    }
    if (angle_dif < -threshold) {
        *shift_ang += 360000;
    }
    *prev_angle = angle;
    int32_t angle_ret = angle + *shift_ang;
    return angle_ret;
}

void RPi_uart_status_send(
    uint8_t enc1_v1,
    uint8_t enc1_v2,
    uint8_t enc2_v1,
    uint8_t enc2_v2,
    int32_t angle1,
    int32_t angle2){
    HAL_Delay(80);
}

```

```

uint8_t switch_states = 0;
uint8_t motor0_switch_CW_state = (enc1_v1 > 0) ? 1: 0;
uint8_t motor0_switch_CCW_state = (enc1_v2 > 0) ? 1: 0;
uint8_t motor1_switch_CW_state = (enc2_v1 > 0) ? 1: 0;
uint8_t motor1_switch_CCW_state = (enc2_v2 > 0) ? 1: 0;
switch_states |= (motor0_switch_CCW_state << 0);
switch_states |= (motor0_switch_CW_state << 1);
switch_states |= (motor1_switch_CW_state << 2);
switch_states |= (motor1_switch_CCW_state << 3);
current_angle1 = angle1;
current_angle2 = angle2;
int32_t a1 = adjust_angle(current_angle1, &prev_angle1,
&shif_angle1);
int32_t a2 = adjust_angle(current_angle2, &prev_angle2,
&shif_angle2);
gen_buff();
RET_USART_send(pUSART_app, buffl, sizeof(buffl));
}

```

Лістинг 7Б – Код для генерації та надсилання відео на вебсервер

```

def generate_frame():
    global last_frame
    current_frame = get_newest_and_lclear_queue()
    if current_frame is None:
        if last_frame is None:
            blanck_frame = np.zeros((480, 640, 3),
dtype=np.uint8)
            current_frame = blanck_frame
        else:
            current_frame = last_frame
    last_frame = current_frame
    _, jpeg = cv2.imencode('.jpg', current_frame)
    return jpeg.tobytes()
last_generate_time = datetime.now()

```

```

def generate_mjpeg():
    global last_generate_time
    while True:
        frame = generate_frame()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame +
               b'\r\n')

        normal_delay_ms = 1 / FRAMERATE * 1000
        now_time = datetime.now()
        real_delay_ms = math.floor((now_time -
last_generate_time).total_seconds()*1000)
        if real_delay_ms < normal_delay_ms:
            add_wait_ms = normal_delay_ms - real_delay_ms
            #print(f"Addition wait: {add_wait_ms} ms")
            time.sleep(add_wait_ms / 1000)
        last_generate_time = now_time

app = Flask(__name__)
frame_queue = queue.LifoQueue()
settings_queue = queue.LifoQueue()

async def handler(websocket, path):
    print("Client connected")
    try:
        async for message in websocket:
            #print(f"Income message: {message}")
            settings_queue.put(message)
            #await websocket.send(f"Echo: {message}")
    except websockets.ConnectionClosed:
        print("Client disconnected")

def websocket_start():
    loop = asyncio.new_event_loop()
    asyncio.set_event_loop(loop)
    server = websockets.serve(handler, WEBSOCKET_IP,
WEBSOCKET_PORT)

```

```
    print(f"Websocket started at port {WEBSOCKET_PORT}")
    loop.run_until_complete(server)
    loop.run_forever()
@app.route('/video_feed')
def video_feed():
    return Response(generate_mjpeg(),
                    mimetype='multipart/x-mixed-replace;
boundary=frame')
```

Додаток В

Перелік елементів

<i>Поз. позначення</i>	<i>Найменування</i>	<i>Кіл.</i>	<i>Примітка</i>
	<u><i>Конденсатори</i></u>		
C1, C5, C14, C19, C21, C25, C27, C30, C31, C33, C35, C36, C40, C46, C49, C50, C51, C52, C56, C57A, C57B, C57C, C57D, C57E, C57F, C57G, C57H, C57I, C57J, C57K, C57L	CC0603KRX7R9BB104, 100нФ 50В ±10% 0603	31	
C10, C11, C15, C16, C23, C29	GRM31CR71H475MA12L, 4,7мкФ 50В ±10% 1206	6	
C12, C17	TAJD227K010RNJ, 220мкФ 10В ±10% 2917	2	
C13, C18, C22, C28	EEE-1HA101UP, 100мкФ 50В ±20% 0805	4	
C2, C3, C8, C9	CC0603KRX5R6BB105, 1мкФ 10В ±10% 0603	4	
C20, C24, C34, C37	C0805C475K3PACTU, 4,7мкФ 25В ±10% 0805	4	
C26, C38	TPSB106K016R0800, 10мкФ 16В ±10% 0805	2	
C32, C45	C0603C102J2GACAUTO, 1нФ 200В ±5% 0603	2	
C39	CC0603KRX7R9BB103, 10нФ 50В ±10% 0603	1	
C4, C53	T491B106K016AT, 10мкФ 16В ±10% 1411	2	
C41, C42	TAJB476M010RNJ, 47мкФ 10В ±20% 1210	2	
C43, C44, C47, C48	CC0603KRX7R7BB105, 1мкФ 16В ±10% 0603	4	
C54, C55	CC0603JRNPO9BN120, 12пФ 50В ±5% 0603	2	
C58A, C58B, C59A, C59B	CC0603KRX7R7BB104, 100нФ 16В ±10% 0603	4	
C6, C7	CAP 470nF 250V 1812(4532), 470нФ 250В ±20% 1812	2	

					КС КРБ 123.196.00.01 ПЕ			
<i>Зм</i>	<i>Арк</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>	<i>Вітовцик П.В.</i>				Комп'ютеризована система керування стабілізаційною платформою для наземної турелі Перелік елементів	<i>Лім</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевір.</i>	<i>Лецишин Ю.З.</i>					<i>Н</i>		
<i>Консульт.</i>						ТНТУ, каф. КС, гр. СІ-41		
<i>Н. контр.</i>	<i>Туцик Н.С.</i>							
<i>Зав. каф.</i>	<i>Осухівська Г.М.</i>							

Поз. позначення		Найменування		Кіл.	Примітка
		<u>Конектори</u>			
X1, X6	346900120, 12-контактний роз'єм	2			
X11, X12	346900080, 8-контактний роз'єм	2			
X13	217182-0001, USB 2.0 Type-C 24-контактний	1			
X3	346900160, 16-контактний роз'єм	1			
X4, X7, X8	XT30	3			
X5, X2	XT60, XT60	2			
X9, X10, X14	347920040, 4-контактний роз'єм	3			
X15A, X15B	61300511121, 5-контактний роз'єм 2.54мм	2			
		<u>Світлодіоди</u>			
D10, D16, D19, D20, D22	LTST-C190GKT, зелений світлодіод SMD	5			
D12, D24	LTST-C190KRKT, червоний світлодіод SMD	2			
D23	LTST-C191TBKT, синій світлодіод SMD	1			
		<u>DC-DC перетворювачі</u>			
U5, U17	ROE-0505S, DC-DC перетворювач 5В 1Вт	2			
		<u>Запобіжники</u>			
F1, F2, F3, F4, F5	0466003.NR, запобіжник 3А 32В 1206	5			
F6, F7, F8, F9	0466.125NR, запобіжник 125мА 125В	4			
F10A, F10B, F10C, F10D, F10E, F10F, F10G	0468.500NR, запобіжник 500мА 63В	7			
		<u>Транзистори</u>			
Q1	2N7002, N-канальний MOSFET 60В 1.8Ом 0.35А SOT-23	1			
Q2A, Q2B, Q2C, Q2D, Q2E, Q2F, Q2G	SQ2318AES-T1_GE3, N-канальний MOSFET 40В 8А SOT-23	7			

Зм

Арк

№ докум

Підпис

Дата

КС КРБ 123.196.00.01 ПЕ

Арк

Поз. позначення		Найменування		Кіл.	Примітка
		<u>Індуктори</u>			
L1, L2, L3		CDRH127NP-100МС, 10μГн ±20%		3	
L4, L5		BLM18KG300TN1D, феритовий фільтр 30Ом 100МГц 0603		2	
		<u>Мікросхеми</u>			
U1, U3		BTS432E2 E3062A, розумний високовольтний ключ 5А		2	
U10		MAX13487EESA+T, трансивер RS-485		1	
U11		LP2985A-33DBVR, стабілізатор напруги 3.3В 150мА		1	
U12		MCP6002-I/SN, подвійний операційний підсилювач		1	
U14		PCA82C250T/YM,112, CAN трансивер		1	
U15		STM32G070RBT6, 32-бітний мікроконтролер ARM Cortex-M0+		1	
U19A, U19B		CD40109BPWR, перетворювач рівнів напруги		2	
U2, U4		LM2596HV-ADJ, понижуючий стабілізатор 3А 4.5-40В		1	
U6		BTS141TC, розумний низьковольтний ключ 12А		1	
U7		LM2577SX-ADJ/NOPB, підвищуючий стабілізатор 3А		1	
U8		REG1117-3.3, стабілізатор напруги 3.3В 800мА		1	
U9, U13, U16		ADUM1201ARZ, подвійний цифровий ізолятор 2.5кВ		3	
		<u>Оптопарі</u>			
U18A, U18B, U18C, U18D, U18E, U18F, U18G, U18H, U18I, U18J, U18K, U18L		HMHA281, фототранзисторна оптопара 6В 50мА		12	
		<u>Резонатори</u>			
Y1		TSX-3225 16.0000MF09Z-AC0, кварц 16МГц 9пF SMD		1	
					КС КРБ 123.196.00.01 ПЕ Арк
Зм	Арк	№ докум	Підпис	Дата	

Поз. позначення	Найменування	Кіл.	Примітка
	<u>Діоди</u>		
D1, D2, D3, D5, D7, D11, D27A, D27B, D27C, D27D, D27E, D27F, D27G, D27H, D27I, D27J, D27K, D27L	ESD5Z3.3T1G, діод ECP 3.3B	18	
D21, D25	PESD1CAN-UX, діод CAN 24B	2	
D26	SMAJ5.0A-TR, TVS діод 5B	1	
D4, D9, D13, D15, D17	SS34-E3/57T, діод 40B 3A	5	
D6, D18	SMAJ48A, TVS діод 48B	2	
D8, D14	SMAJ12A-E3/61, TVS діод 12B	2	
	<u>Резистори</u>		
R1, R3, R5, R8, R9, R13, R15, R28, R65, R68, R69, R70, R73, R74, R75, R77, R79, R82, R83, R84, R85	RC0603FR-071KL, 1кОм ±1% 1/10Вт 0603	21	
R14, R27	RC0603FR-07150RL, 150Ом ±1% 1/10Вт 0603	2	
R17, R31	RT0603BRE076K04L, 6,04кОм ±0,1% 1/10Вт 0603	2	
R19, R21, R41, R42, R60	RT0603BRD072K05L, 2,05кОм ±0,1% 1/10Вт 0603	5	
R2, R10, R12, R23, R43, R66, R81, R87, R91, R46	RC0603FR-0715KL, 15кОм ±1% 1/10Вт 0603	10	
R20, R36, R92	RC0603FR-07100KL, 100кОм ±1% 1/10Вт 0603	3	
R25, R26, R39, R40, R96A, R96B, R96C, R96D, R96E, R96F, R96G, R97A, R97B, R97C, R97D, R97E, R97F, R97G	CRCW06030000Z0EA, 0Ом 0603	18	
R29, R30, R33, R35, R45, R47, R48	RT0603BRD07220RL, 220Ом ±0,1% 1/10Вт 0603	7	
R34	RC1206JR-070RL, 0Ом 1/4Вт 1206	1	
R38	MCR50JZHFLR100, 0,1Ом ±1% 0,5Вт 2010	1	
Зм	Арк	№ докум	Підпис
			Дата
			КС КРБ 123.196.00.01 ПЕ
			Арк

<i>Поз. позначення</i>	<i>Найменування</i>	<i>Кіл.</i>	<i>Примітка</i>
R4, R7, R11, R16, R18, R24, R32, R37, R50, R51, R52, R53, R54, R56, R59, R63, R67, R71, R86, R88, R89, R90, R93A, R93B, R93C, R93D, R93E, R93F, R93G, R93H, R93I, R93J, R93K, R93L, R99A, R99B, R99C, R99D, R99E, R99F, R99G	RC0603FR-0710KL, 10кОм ±1% 1/10Вт 0603	41	
R49, R55, R61, R72, R78, R100A, R100B, R100C, R100D, R100E, R100F, R100G	RC0603FR-074K7L, 4,7кОм ±1% 1/10Вт 0603	12	
R57, R76	CRCW1206120RJNEA, 120Ом ±5% 0,25Вт 1206	2	
R58, R80	RC0603FR-072K2L, 2,2кОм ±1% 1/10Вт 0603	2	
R6, R22, R44, R94A, R94B, R94C, R94D, R94E, R94F, R94G, R94H, R94I, R94J, R94K, R94L, R95A, R95B, R95C, R95D, R95E, R95F, R95G, R95H, R95I, R95J, R95K, R95L	RC0603FR-07100RL, 100Ом ±1% 1/10Вт 0603	27	
R62, R64	2K 1% 0603(1608), 2кОм ±1% 1/10Вт 0603	2	
R98A, R98B, R98C, R98D, R98E, R98F, R98G	RT0805FRE074K7L, 4,7кОм ±1% 1/8Вт 0805	7	

					КС КРБ 123.196.00.01 ПЕ	Арк
Зм	Арк	№ докум	Підпис	Дата		

