

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет

імені Петра Могили

Факультет комп'ютерних наук

Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

**Завідувач кафедри,
д-р техн. наук, проф.**

_____ **І. М. Журавська**

« __ » _____ 2023 р.

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

**Шасі безпілотного наземного транспортного
засобу високої проходимості на базі моторів-коліс**

Спеціальність 123 Комп'ютерна інженерія

123 – КБР.ПЗ.00 – 405.21910518

Студент

_____ **П.М. Лосіцький**

підпис

« __ » _____ 202__ р.

Керівник ст. викладач

_____ **І. С. Бурлаченко.**

підпис

« __ » _____ 202__ р.

Миколаїв – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Зав. кафедри _____ І. М. Журавська

« _____ » _____ 2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної бакалаврської роботи

Видано студенту групи 405 факультету комп'ютерних наук

Лосіцький Павло Михайлович
(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

Шасі безпілотного наземного транспортного засобу високої проходимості на базі моторів-коліс

Затверджена наказом по ЧНУ від « _____ » _____ 20__ р. № _____

2. Строк представлення кваліфікаційної роботи « _____ » _____ 20__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом роботи є: апаратне та програмне забезпечення систем дистанційного керування шасі транспортного засобу за допомогою бездротового зв'язку.

4. Перелік питань, що підлягають розробці

1) огляд сучасних систем безпілотних транспортних засобів, моторів-коліс та типів підвісних систем автомобілів що розробляються;

2) аналіз переваг та недоліків існуючих систем;

3) побудова математичної моделі та розрахунки характеристик безпілотного наземного транспортного засобу;

4) розробка апаратно-програмної частини безпілотного наземного транспортного засобу;

5. Перелік графічних матеріалів

слайди презентації

6. Завдання до спеціальної частини

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
А. О. Алексеєва, к.т.н., доцент	кафедра екології Медичного інституту ЧНУ імені Петра Могили	Спеціальна частина з охорони праці

Керівник роботи

ст. викладач Бурлаченко Іван Сергійович

(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

Лосіцький Павло Михайлович

(прізвище, ім'я, по батькові студента)

(підпис)

Дата видачі завдання « ____ » _____ 20 ____ р.

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної бакалаврської роботи

Тема: Шасі безпілотного наземного транспортного засобу високої прохідності на базі моторів-коліс.

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КР	11.12.22	12.12.22	Виконав
2.	Огляд літератури за темою роботи	12.01.23	15.01.23	Виконав
3.	Огляд існуючих рішень системи, що розробляється	15.01.23	18.02.23	Виконав
4.	Формування вимог до апаратно-програмного забезпечення	19.02.23	04.03.23	Виконав
5.	Формування математичної моделі БНТЗ	05.04.23	10.04.23	Виконав
6.	Формування методів керування	10.04.23	15.04.23	Виконав
7.	Формування моделі енергоживлення	15.04.23	20.04.23	Виконав
8.	Формування моделі прохідності	20.04.23	25.04.23	Виконав
9.	Перший передзахист	30.04.23	30.04.23	Виконав
10.	Вибір програмних і апаратних компонентів	1.05.23	5.05.23	Виконав
11.	Побудова схеми підключення компонентів	5.05.23	7.05.23	Виконав
12.	Побудова блок схеми алгоритму керування	7.05.23	8.05.23	Виконав
13.	Написання скетчу	8.05.23	15.05.23	Виконав
14.	Перевірка працездатності апаратно-програмного забезпечення	16.05.23	19.05.23	Виконав
15.	Дослідження та формування розділу з охорони праці	19.05.23	31.05.23	Виконав
16.	Оформлення КБР та презентації	1.06.23	12.06.23	Виконав
17.	Попередній захист	13.06.23	13.06.23	Виконав
18.	Завершення оформлення КБР та презентації	14.06.23	18.06.06	Виконав
19.	Відгук керівника КБР	18.06.23	18.06.23	Виконав
20.	Рецензування	18.06.26	19.06.23	Виконав
21.	Захист кваліфікаційної роботи	27.06.23	27.06.23	Виконав

Розробив здобувач ВО Лосіцький Павло Михайлович

(прізвище, ім'я, по батькові)

(підпис)

«__» _____ 20__ р.

Керівник роботи ст. викладач Бурлаченко Іван Сергійович

(підпис)

«__» _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної роботи
«Шасі безпілотного наземного транспортного засобу
високої прохідності на базі моторів-коліс»
Студент 405 гр.: Лосіцький Павло Михайлович
Керівник: старший викладач кафедри КІ, Бурлаченко І. С.

У даній роботі проведено дослідження шасі безпілотного наземного транспортного засобу (БНТЗ) високої прохідності на базі моторів коліс. Запропоноване дослідження є актуальним у контексті потреби в автоматизованих рішеннях для робіт у сферах, де важлива висока прохідність транспортних засобів.

Об'єктом дослідження є цифрове керування шасі безпілотного наземного транспортного засобу високої прохідності. Предметом дослідження є технології бездротового зв'язку для організації взаємодії Android застосунків з апаратним забезпеченням на базі Arduino.

Метою даної роботи є розроблення шасі безпілотного наземного транспортного засобу високої прохідності, дослідження параметрів БНТЗ та особливостей роботи в умовах, які відрізняються від умов, при яких працюють звичайні автомобілі.

Практична значимість даного дослідження полягає у тому, що його результати можуть сприяти автоматизації сільського господарства, промисловості, військових та інших галузей, де вимагається висока прохідність транспортних засобів та є небезпека для життя чи здоров'я людини.

Результати роботи були апробовані на XXV Всеукраїнській науково-практичній конференції "Могилянські читання–2022: досвід та тенденції розвитку суспільства в Україні: глобальний, національний та регіональний аспекти" у місті Миколаїв.

В першому розділі роботи розглянуто основні сучасні системи безпілотних транспортних засобів, мотори-колеса, типи приводів та підвісних систем транспортних засобів. В другому розділі проведено розрахунки технічних

характеристик безпілотного наземного транспортного засобу високої прохідності. В третьому розділі представлено результати розроблення апаратно-програмного комплексу для керування безпілотним наземним транспортним засобом. Отримані результати підтверджують значимість шасі безпілотного наземного транспортного засобу на базі моторів коліс у вимогливих умовах експлуатації.

Робота складається з 56 сторінок(без джерел посилань та додатків), 4 таблиці, 24 рисунки та 4 додатків. В процесі дослідження було використано 24 джерел посилання.

Ключові слова: шасі, безпілотний наземний транспортний засіб, мотор-коліс, прохідність, підвісна система, контролер, драйвер.

ABSTRACT

of the Bachelor's Thesis
"Chassis of an unmanned ground vehicle
with high off-road capability based on wheel motors"

Student.: Pavlo Mykhailovych Lositskyi

Supervisor: senior lecturer of the CE Department, Ivan S. Burlachenko

In this work, a study of the chassis of an unmanned ground vehicle of high off-road capability based on wheel motors was performed. The proposed study is relevant in the context of the need for automated solutions for work in areas where high vehicle traffic is important.

The object of research is the digital control of the chassis of an unmanned ground vehicle of high off-road capability. The subject of research is wireless communication technologies for organizing the interaction of Android applications with Arduino-based hardware.

The purpose of this work is to study the chassis of an unmanned ground vehicle of high off-road capability and to study their parameters and features of operation in conditions that differ from the conditions under which ordinary cars operate.

The practical significance of this research is that its results can contribute to the automation of agriculture, industry, the military and other industries where high vehicle traffic is required and there is a danger to human life or health.

The results of the work were tested at the 25th All-Ukrainian scientific and practical conference "Mohyla readings-2022: experience and trends of the development of society in Ukraine: global, national and regional aspects" in the city of Mykolaiv.

In the first chapter of the study, the main modern systems of unmanned ground vehicles, wheel motors, types of drivetrains, and suspension systems of vehicles are discussed. The second chapter involves calculations of the technical characteristics of a high-mobility unmanned ground vehicle. The third chapter presents the results of the development of a hardware and software complex for controlling the unmanned ground

vehicle. The obtained results confirm the significance of the chassis of the wheel-based unmanned ground vehicle in demanding operating conditions

The work consists of 56 pages (excluding references and appendices), 4 tables, 24 figures, and 4 appendices. During the research, 27 sources were used.

Key words: chassis, unmanned ground vehicle, motor-wheel, cross-country ability, suspension system, controller, driver.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	3
ВСТУП	4
1 АНАЛІТИЧНА ЧАСТИНА	6
1.1 Огляд існуючих рішень системи, що розробляється.....	6
1.2 Формування вимог до апаратно-програмного забезпечення	17
Висновки до розділу 1	19
2 МАТЕМАТИЧНІ МЕТОДИ ТА МОДЕЛІ	21
2.1 Математична модель наземного транспортного засобу	21
2.2 Методи керування та оптимізації.....	26
2.3 Математичні моделі енергоживлення.....	30
2.4 Математична модель коливань та прохідності підвіски транспортного засобу	37
Висновки до розділу 2	41
3 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	42
3.1 Вибір програмних і апаратних компонентів	42
3.2 Реалізація проекту.....	47
3.3 Завантаження скетчу на ESP32.....	55
Висновки до розділу 3	57
ВИСНОВКИ.....	58
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	59
ДОДАТОК А ДОВІДКА	62
ДОДАТОК Б БЛОК-СХЕМА	63
ДОДАТОК В СКЕТЧ 1	64
ДОДАТОК Г СКЕТЧ 2	68

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

АКБ – Аккумуляторна батарея

БНТЗ – Безпілотні наземні транспортні засоби

ТЗ – Транспортні засоби

AWD – All-Wheel Drive

FWD – Front Wheel Drive

RWD – Rear-Wheel Drive

ВСТУП

Сьогодні в епоху швидкого розвитку технологій все більше уваги приділяється розробці автономних наземних транспортних засобів. Одним з таких засобів є БНТЗ високої прохідності.

Шасі безпілотних транспортних засобів є важливою складовою системи, яка відповідає за стабільність, маневреність та прохідність засобу. Особливу увагу необхідно приділяти розробці шасі для безпілотних транспортних засобів високої прохідності, які можуть працювати в умовах, які є вкрай складними для людини, наприклад, в гірських районах, в бездоріжжі або в умовах, що несприятливо впливають на здоров'я людини.

Об'єктом дослідження є цифрове керування шасі БНТЗ високої прохідності.

Предметом дослідження є технології бездротового зв'язку для організації взаємодії Android застосунків з апаратним забезпеченням на базі Arduino.

Метою даної роботи є розроблення шасі безпілотного наземного транспортного засобу високої прохідності, дослідження параметрів БНТЗ та особливостей роботи в умовах, які відрізняються від умов, при яких працюють звичайні автомобілі.

Поставлена мета передбачає собою вирішення наступних завдань:

1. огляд літератури та аналіз існуючих рішень;
2. підбір апаратних компонентів;
3. розробка математичної моделі БНТЗ;
4. реалізація апаратно-програмного комплексу;
5. аналіз та оформлення результатів.

Практична значущість дослідження полягає в тому, що розробка шасі БНТЗ високої прохідності на базі моторів-коліс може мати значний вплив на різноманітні сфери, такі як військова, громадська та промислова.

За військової діяльності, безпілотні транспортні засоби з високим рівнем прохідності можуть використовуватися для розвідки, пошуку та рятування,

доставки вантажів та виконання місій у зоні бойових дій. Це може зменшити ризик для життя та здоров'я військовослужбовців та допомогти їм виконувати різні завдання.

У громадському секторі, безпілотні ТЗ можуть використовуватися для пошуку та рятування, моніторингу стану доріг, управління та нагляду за лісовими ділянками та заповідниками, а також для розвідки та пошуку жертв при природних катастрофах.

У промисловості, безпілотні ТЗ можуть використовуватися для вантажоперевезень, моніторингу та діагностики важко завантажених машин, а також для автоматизації роботи на великих відкритих майданчиках, таких як кар'єри та гірничі підприємства.

Крім того, розробка БНТЗ є актуальною темою в галузі автоматизації та робототехніки, оскільки може знизити витрати на транспортування, поліпшити безпеку та ефективність перевезення товарів та пасажирів.

Таким чином, розробка шасі БНТЗ високої прохідності є науково новим та перспективним напрямком дослідження, який може мати важливі практичні застосування в різних галузях та покращити ефективність транспорту в цілому.

1 АНАЛІТИЧНА ЧАСТИНА

Аналітична частина містить огляд існуючих рішень системи, що розробляється, а також формування вимог до апаратно-програмного забезпечення. Цей розділ є ключовим етапом у дослідженні, оскільки він дозволяє ознайомитися з попередніми розробками та здійснити аналіз їх переваг та недоліків.

Огляд існуючих рішень системи дозволяє визначити те, що вже було зроблено в даній області та які технології вже застосовуються. Цей аналіз сприяє формуванню бази знань та визначенню прогалів, які ще потребують подальшого дослідження.

Далі, у цьому розділі будуть сформульовані вимоги до апаратно-програмного забезпечення системи. Це включає вимоги до апаратної частини шасі, програмного забезпечення для безпілотного управління, а також системи навігації та керування. Цей етап є важливим для визначення основних параметрів та характеристик, які система має виконувати, а також для визначення потреб у подальшому дослідженні та розробці.

Аналітична частина є фундаментальною складовою нашої роботи, оскільки вона ставить базові принципи та вимоги, на яких буде базуватися подальший дослідницький процес.

1.1 Огляд існуючих рішень системи, що розробляється

У сьогоденні розробляється безліч автономних та безпілотних транспортних засобів, які знаходять застосування в різних сферах та галузях. Ці інноваційні технології перетворюють спосіб, яким ми пересуваємось, та мають потенціал змінити наш погляд на автомобільну індустрію та громадський транспорт. Від автономних автомобілів, що здатні перевозити пасажирів без участі водія, до безпілотних дронів для доставки товарів, ці розробки відкривають нові перспективи в ефективності, безпеці та сталості транспортних систем.

Відомі компанії, такі як Tesla, Waymo, Uber, General Motors та багато інших, активно працюють над розробкою та вдосконаленням автономних та безпілотних рішень. Ці проекти включають в себе розробку спеціальних алгоритмів, використання штучного інтелекту, розпізнавання об'єктів, сенсорну технологію та інші інноваційні елементи, які допомагають створювати автономні транспортні засоби, що забезпечують безпеку та ефективність.

Окрім того, деякі проекти та платформи, такі як Apollo від Baidu, надають відкриті рішення для розробки автономних систем, що сприяє спільному вдосконаленню технологій та сприяє зростанню автономного транспорту як глобального явища.

Ці розробки відкривають нові можливості для майбутнього транспорту, забезпечуючи покращену мобільність, зменшення впливу на навколишнє середовище та покращення безпеки на дорогах. Використання автономних та безпілотних транспортних засобів в міському, логістичному та сільському господарстві, медичному транспорті та багатьох інших галузях може привести до значних переваг у сфері транспорту та забезпечити більш сталу та зручну майбутню транспортну систему.

1.1.1 Огляд безпілотних наземних транспортних засобів

БНТЗ є активно розвиваючоюся галуззю, що має потенціал змінити транспортну систему майбутнього. Ці автономні системи можуть бути застосовані в різних сферах, включаючи логістику, громадський транспорт, сільське господарство та військову сферу. Огляд безпілотних транспортних засобів дозволяє визначити різноманітність систем та їхні функціональні можливості.

Одним з основних видів БНТЗ є автономні автомобілі. Ці транспортні засоби оснащені датчиками, системами візуального сприйняття, лідарами, радарми та GPS-навігацією, що дозволяють їм реагувати на дорожні умови та навколишнє середовище без участі водія. Такі автомобілі проводять обробку

даних в режимі реального часу, розпізнають дорожні знаки, перешкоди та інші транспортні засоби, а також приймають рішення про безпечний рух.

Крім автономних автомобілів, існують інші типи БНТЗ, такі як безпілотні мікро пересувні засоби (наприклад, електричні скутери та велосипеди), безпілотні автобуси та безпілотні вантажні автомобілі. Кожен з цих типів має свої особливості та застосування залежно від конкретної сфери використання.

Огляд БНТЗ є актуальним завданням у сучасному світі. В даний час існує велика кількість таких транспортних засобів, в тому числі традиційні автомобілі, електромобілі, велосипеди та електросамокати з автоматичним керуванням.

Одним з найбільш відомих проектів безпілотних транспортних засобів є проект Waymo [1], який належить компанії Alphabet, материнської компанії Google. Компанія Waymo розробила безпілотний автомобіль, який вже проходить деякі тестові випробування в США.

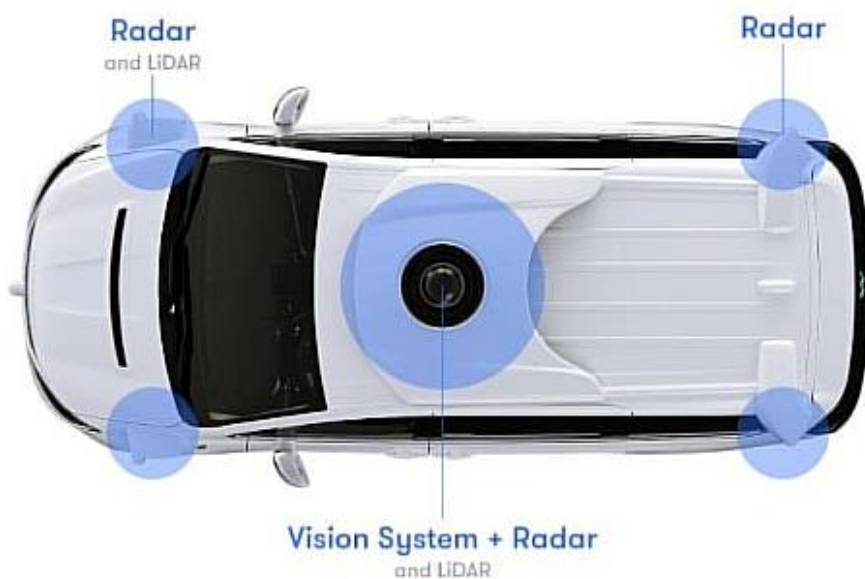


Рисунок 1.1 – технологія Waymo

Технологія Waymo базується на використанні штучного інтелекту та глибокого навчання для навігації та безпечного керування автономними автомобілями. Основні складові технології Waymo включають:

Система спостереження та сприйняття: Waymo використовує різноманітні сенсори, такі як лідари, радары, камери та датчики, для отримання високоякісної

інформації про навколишнє середовище. Ці сенсори забезпечують чітке спостереження за дорогами, транспортними засобами та перешкодами.

Обробка та аналіз даних: Waymo використовує потужні обчислювальні системи для обробки та аналізу великого обсягу даних, що надходять від сенсорів. Застосування глибокого навчання та інших алгоритмів штучного інтелекту дозволяє системі розпізнавати об'єкти, прогнозувати їхню поведінку та приймати рішення на основі цих даних.

Автономне управління: Waymo розробляє алгоритми та системи управління, які дозволяють автомобілям самостійно приймати рішення про рух, швидкість, маневри та виконувати безпечну навігацію на дорозі.

Іншим відомим проектом є Tesla[3], який розробляє електричний автомобіль з автоматичним керуванням. Цей автомобіль може переключатися між різними режимами керування, включаючи автопілот та повністю автоматичне керування.



Рисунок 1.2 – автопілот Tesla

Технологія автопілота Tesla є однією з провідних систем автоматизованого водіння на ринку автомобілів. Автопілот Tesla є еволюційним підходом до автономного водіння, що поступово набуває нових функцій та можливостей з

кожним оновленням програмного забезпечення. Основні складові технології автопілота Tesla включають:

Камери та сенсори: автомобілі Tesla оснащені набором камер, радарів, ультразвукових датчиків та інших сенсорів, які забезпечують спостереження за дорогою та навколишнім середовищем. Ці сенсори збирають велику кількість даних про дорожній рух, перешкоди та інші об'єкти.

Обробка даних та машинне навчання: зібрані дані з камер та сенсорів передаються до системи обробки даних, яка застосовує методи машинного навчання для аналізу та розпізнавання об'єктів на дорозі. Це дозволяє системі розпізнавати автомобілі, пішоходів, дорожні знаки, світлофори та інші елементи інфраструктури.

Автоматичне керування: система автопілота Tesla використовує отримані дані та результати аналізу, щоб приймати рішення про рух, швидкість та маневри автомобіля. Вона забезпечує автоматичне керування, включаючи утримання автомобіля у полосі руху, контроль швидкості, розпізнавання та об'єктивне реагування на дорожні ситуації.

Деякі інші компанії, такі як Uber [3], також працюють над розробкою безпілотних транспортних засобів. У 2018 році Uber випустив перший безпілотний таксі в місті Піттсбург.

Загалом, розвиток безпілотних транспортних засобів є перспективним та актуальним напрямком в автомобільній промисловості. В даний час багато компаній працюють над розробкою нових технологій та вдосконаленням існуючих систем для покращення безпеки та ефективності безпілотних наземних транспортних засобів.

1.1.2 Огляд технологій моторів-колес

У сучасному автомобільному та робототехнічному просторі, мотори-колеса займають центральне місце в розробці безпілотних наземних транспортних засобів. Мотори-колеса є важливим компонентом системи шасі і

відіграють ключову роль у забезпеченні рухомості, маневреності та ефективності таких транспортних засобів.

Існує кілька різних типів технологій моторів-коліс, які застосовуються у безпілотних транспортних засобах. Декілька з них включають:

- електричні мотори-колеса;
- гібридні мотори-колеса;
- гідравлічні мотори-колеса.

Електричні мотори-колеса [4] є популярними у БНТЗ, оскільки вони пропонують екологічно чистий привід, високу ефективність та миттєвий крутний момент. Такі мотори-колеса зазвичай використовуються в електромобілях та гібридних транспортних засобах.

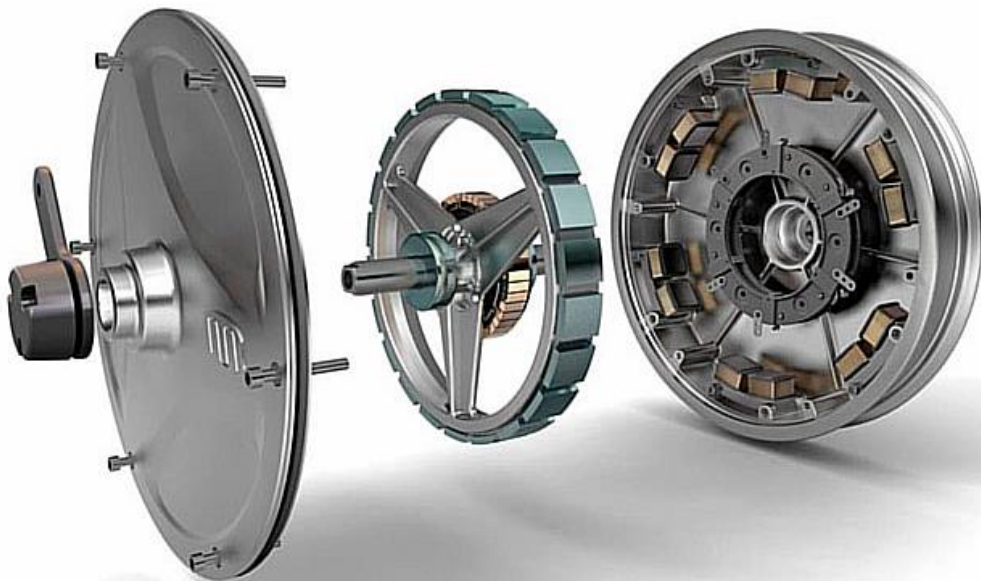


Рисунок 1.3 – Електричне мотор-колесо

Перевагами використання електричних моторів-коліс є ефективність та компактність. Вони дозволяють знизити вагу транспортного засобу, що зменшує витрати на паливо або збільшує запас ходу в електромобілі. Також електричні мотори-колеса забезпечують більш точне керування, адже кожне колесо може бути незалежно контролюване за допомогою програмного забезпечення.

Однак, є деякі виклики та обмеження, пов'язані з використанням електричних моторів-коліс. Наприклад, їх потужність та швидкість обертання

можуть бути обмежені в порівнянні з традиційними двигунами внутрішнього згоряння. Також, вони можуть бути дорожчими у виробництві та ремонті в порівнянні з іншими видами приводу.

Гібридні мотори-колеса [5] поєднують електричні та традиційні горючим паливом системи приводу. Ця технологія дозволяє поєднувати переваги електричних моторів-коліс (екологічність, висока ефективність) з довготривалим діапазоном пробігу традиційних автомобілів.



Рисунок 1.4 – Гібридне мотор-колесо

Основними перевагами гібридних моторів-коліс є ефективне використання енергії, зниження витрат палива та викидів шкідливих речовин. Гібридні системи дозволяють використовувати електричний привід для маневрування та руху на низькій швидкості, тоді як традиційний двигун активується при високих навантаженнях або для зарядки батареї. Це дозволяє знизити споживання палива та шкідливі викиди, особливо у міському середовищі.

Гібридні мотори-колеса також мають інші переваги, такі як рекуперація енергії під час гальмування, що допомагає зберігати та використовувати енергію, яка зазвичай втрачається. Вони також забезпечують покращену маневреність та контроль за рухом транспортного засобу завдяки можливості незалежного контролю кожного колеса.

Проте, гібридні мотори-колеса також мають свої мінуси. Вони можуть бути складнішими в проектуванні та виготовленні порівняно зі звичайними двигунами. Крім того, їх вартість може бути вищою, що впливає на загальну ціну транспортного засобу.

Гідравлічні мотори-колеса [6] використовують рідину під високим тиском для створення руху. Вони можуть бути використані в тяжких умовах експлуатації, таких як вантажні автомобілі.



Рисунок 1.5 – Гідравлічне мотор-колесо

Гідравлічні мотори-колеса є інноваційною технологією, яка поєднує гідравлічну систему з колісним приводом транспортного засобу. Вони використовують гідравлічний привід для передачі обертового моменту на колеса, забезпечуючи рух транспортного засобу.

Основною перевагою гідравлічних моторів-колес є їх висока потужність і здатність до роботи при великих навантаженнях. Вони забезпечують значну силу тяги та мають високий крутний момент, що робить їх ефективними для використання в транспортних засобах, які потребують великої потужності і високої прохідності, наприклад, важкі вантажівки або військові транспорти.

Гідравлічні мотори-колеса також відзначаються високою маневреністю та контролем. Завдяки індивідуальному керуванню кожним колесом, транспортний

засіб може виконувати різні маневри, включаючи повороти на місці, зміну напрямку та регулювання швидкості кожного колеса окремо. Це дозволяє покращити стійкість, керованість та адаптацію до різних умов дороги.

Однак, гідравлічні мотори-колеса мають свої обмеження і мінуси. Вони можуть бути важкими та займати багато місця, що може обмежувати їх застосування в деяких типах транспортних засобів. Крім того, вони вимагають складних гідравлічних систем, які потребують додаткового обладнання, такого як насоси та розподільники потоку, що може підвищити вартість і складність обслуговування.

1.1.3 Аналіз сучасних систем шасі високої прохідності

Аналіз сучасних систем шасі високої прохідності відіграє важливу роль в проектуванні нових безпілотних наземних транспортних засобів. Система шасі є однією з найбільш важливих частин будь-якого автомобіля або іншого транспортного засобу, особливо якщо йдеться про транспорт з високою прохідністю.

Сьогодні на ринку представлено багато різних систем шасі високої прохідності. Одними з найбільш популярних є системи на базі гідропневматичної підвіски, системи залежної підвіски, системи з незалежною підвіскою та системи з кріпленням коліс на важільний підвісці. Кожна з цих систем має свої переваги та недоліки.

Система гідропневматичної підвіски [7], наприклад, дозволяє регулювати жорсткість підвіски та висоту підлоги, що робить її досить універсальною та зручною в експлуатації. Однак вона має досить складну конструкцію та високі витрати на обслуговування.

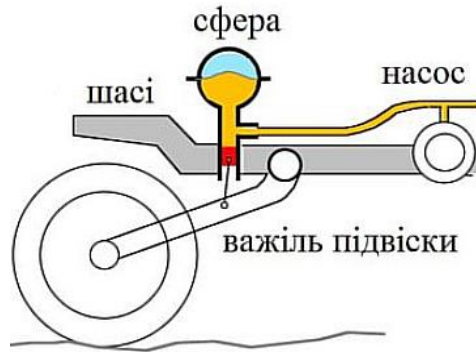


Рисунок 1.6 – Гідропневматична підвіска

Система залежної підвіски [8], у свою чергу, є більш простою в конструкції та менш вимогливою до обслуговування, але має обмежену прохідність та меншу здатність до адаптації до різних умов дорожнього покриття.

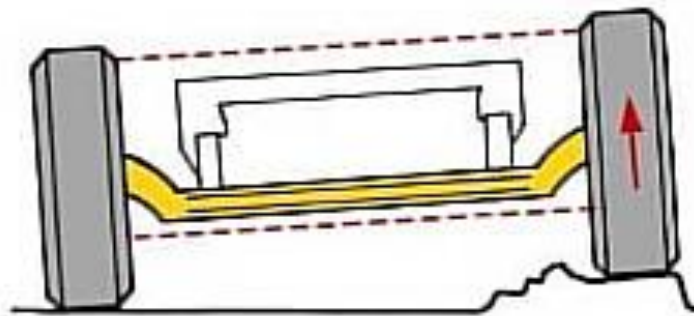


Рисунок 1.7 – Система залежної підвіски

Система з незалежною підвіскою [9], зокрема, є досить універсальною та здатною працювати в різних умовах. Однак вона також має свої недоліки, такі як більш складна конструкція та високі витрати на обслуговування.

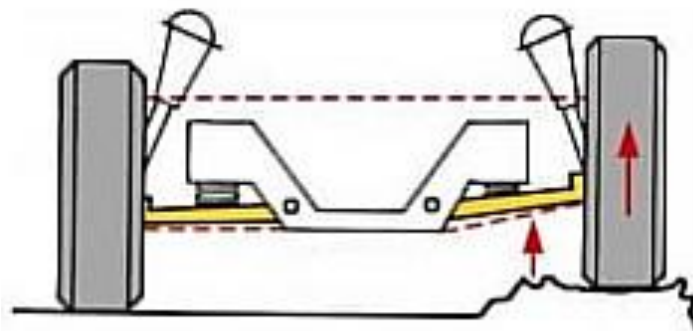


Рисунок 1.8 – Система незалежної підвіски

Також варто враховувати тип приводу транспортного засобу [10]. Транспортні засоби можуть мати різні типи приводів, які визначають, яким чином потужність передається від джерела енергії до коліс або гусениць, забезпечуючи рухомість транспортного засобу. Типи приводів, які широко використовуються:

Задній привід: у задньому приводі потужність передається до задніх коліс транспортного засобу. Цей тип приводу забезпечує краще розподілення ваги на задню частину автомобіля, що робить його стійким на дорозі. Він часто використовується в спортивних автомобілях і транспортних засобах, які мають велику потужність.

Передній привід: у передньому приводі потужність передається до передніх коліс транспортного засобу. Цей тип приводу забезпечує краще управління і керування на дорозі, а також кращу економію палива. Він часто використовується в пасажирських автомобілях та маленьких вантажівках.

Повний привід: повний привід означає, що потужність передається до всіх коліс транспортного засобу. Цей тип приводу надає кращу прохідність та контроль на бездоріжжі, снігу, льоду та інших складних дорожніх умовах. Він часто використовується в позашляховиках, вантажівках та великих SUV.

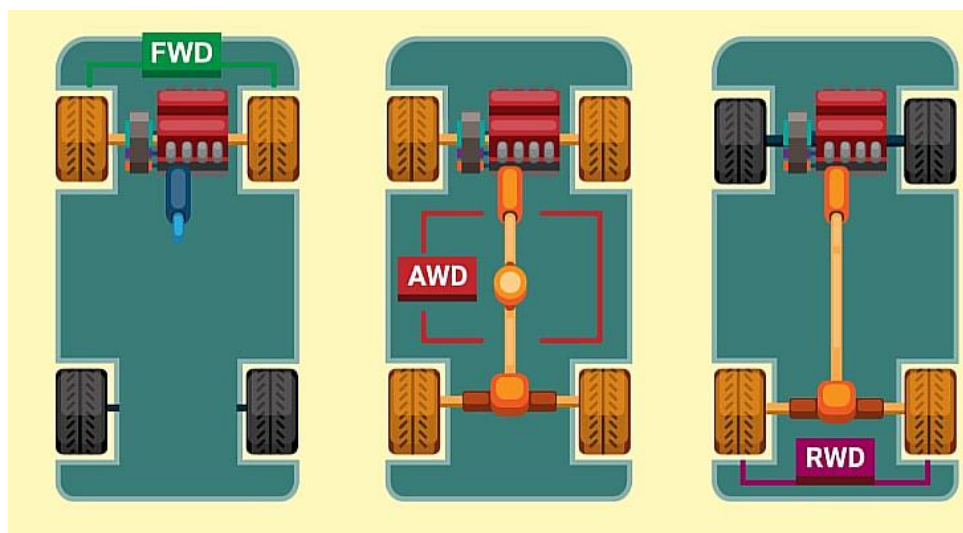


Рисунок 1.9 – Типи приводів ТЗ. FWD – передній,
AWD – повний, RWD – задній

Ще одним показником прохідності є колеса які використовуються на транспортному засобі.



Рисунок 1.10 – Колеса для бездоріжжя з грубим протектором

Використання великих коліс з грубим протектором є одним із способів покращення прохідності. Вони забезпечують краще утримання на бездоріжжі, мають кращу амортизацію і здатні подолати перешкоди, такі як камені, бруд або вода.

1.2 Формування вимог до апаратно-програмного забезпечення

Функціональні вимоги:

- можливість точного керування рухом транспортного засобу, включаючи рух вперед, назад та повороти;
- здатність до дотримання швидкісного режиму;
- використання камер для визначення дорожньої ситуації;
- взаємодія з іншими системами;
- можливість обміну даними з центральною системою управління.

Нефункціональні вимоги:

- здатність системи до працездатності в різних погодних умовах та на різних типах доріг;
- забезпечення резервування системи для забезпечення продовження руху у випадку відмови окремих компонентів;

– використання енергозберігаючих алгоритмів та оптимізація споживання електроенергії системою.

1.2.1 Вимоги до апаратної частини шасі

Вимоги до апаратної частини шасі безпілотного наземного транспортного засобу високої прохідності включають наступні аспекти:

1) механічна міцність:

– стійкість до впливу дорожніх умов: шасі повинно витримувати нерівності дороги, кам'янисті та нерівні поверхні з високою стійкістю та забезпечувати стабільну роботу системи безпілотного управління;

– захист від пошкоджень: шасі повинно бути здатним захищати внутрішні компоненти від зовнішніх впливів, таких як удари, вібрації та знос;

2) енергоефективність:

– енергозберігаючі режими: шасі повинно мати можливість переходу в енергозберігаючі режими роботи для ефективного використання енергії та збільшення тривалості роботи;

– підтримка акумуляторних батарей: Шасі повинно бути сумісним з різними типами акумуляторних батарей, забезпечуючи надійне живлення та швидке зарядження;

3) маневреність:

– радіус повороту: шасі повинно мати здатність здійснювати повороти на малих радіусах для ефективного керування та руху вузькими просторами;

– здатність до маневрування: шасі повинно мати достатню маневреність для виконання складних маневрів, таких як об'їзд перешкод;

4) система керування:

– інтегрована система керування: шасі повинно бути оснащене інтегрованою системою керування, яка забезпечує точне та стабільне керування транспортним засобом;

– реакція на сигнали керування: шасі повинно миттєво реагувати на сигнали керування, забезпечуючи швидке та плавне переключення напрямку руху та швидкості;

5) стійкість та управління:

– стабільність на дорозі: шасі повинно мати стійкість на дорозі, забезпечуючи безпечний рух та уникнення перекидання під час маневрів або руху по нерівностях.

Ці вимоги до апаратної частини шасі допоможуть забезпечити маневреність та надійність транспортного засобу, забезпечуючи безпеку та ефективність його руху.

1.2.2 Вимоги до програмного забезпечення для безпілотного управління

Програмне забезпечення повинно мати можливість встановлення надійного та стабільного зв'язку з безпілотним транспортним засобом. Це може бути досягнуто за допомогою бездротових технологій, таких як радіо, супутниковий зв'язок або мережі передачі даних.

Візуалізація та моніторинг:

– графічний інтерфейс: система повинна мати зрозумілий та інтуїтивний графічний інтерфейс;

– відображення в реальному часі: інтерфейс повинен надавати оператору актуальну інформацію в режимі реального часу про положення, швидкість та інші параметри системи;

– команди та контроль: інтерфейс має надавати засоби для відправлення команд керування транспортним засобом та контролю його роботи.

Висновки до розділу 1

В результаті дослідження та аналізу різних аспектів безпілотних наземних транспортних засобів, включаючи системи шасі та мотори-колеса були виявлені

різноманітні технології та рішення, що сприяють покращенню прохідності, маневреності та безпеки таких транспортних засобів. Огляд існуючих рішень показав, що електричні мотори-колеса є ефективними та екологічно чистими, забезпечуючи низькі викиди та економію палива.

Системи шасі високої прохідності, які використовують незалежний контроль кожного колеса та активне управління динамікою руху, забезпечують поліпшену стійкість, керованість та здатність до подолання перешкод. Це особливо важливо в умовах поганої дорожньої інфраструктури або складного ґрунтового покриття.

Взаємодія з оператором вимагає зрозумілого та інтуїтивного інтерфейсу, що дозволяє оператору ефективно контролювати транспортний засіб та отримувати необхідну інформацію.

Також для шасі високої прохідності було вирішено використовувати повний привід 4×4 так як він є більш ефективним для використання по міській інфраструктурою. В якості підвіски було вирішено використовувати гібридну, а саме задню вісь з системою залежної підвіски, передню вісь з системою незалежної підвіски. Всю цю ходову систему будуть підкреслювати тяжкі, широкі колеса з грубим протектором.

2 МАТЕМАТИЧНІ МЕТОДИ ТА МОДЕЛІ

У цьому розділі розглядаються математичні методи та розрахунки, які використовуються для розробки систем керування БНТЗ. Ці методи є критично важливими для моделювання, оптимізації та аналізу всіх аспектів функціонування системи, забезпечення її ефективності та безпеки. Цей розділ охоплює ключові аспекти математичного моделювання, методи керування та оптимізації, обробку сигналів, математичне моделювання та моделювання, а також розрахунки безпеки та надійності. У наступних підрозділах цієї глави надається детальний огляд цих методів та їх практичних наслідків.

2.1 Математична модель наземного транспортного засобу

Математична модель ТЗ є важливим інструментом для розробки безпілотних систем керування. Вона дозволяє описувати фізику та поведінку ТЗ за допомогою математичних рівнянь і параметрів. У рамках такої моделі можна включити різні компоненти, такі як кінематика (опис руху), динаміка (взаємодія зовнішніх сил і масово-інерційних властивостей), енергетика (використання енергії та палива), системи керування (алгоритми керування та реакції на зміни навколишнього середовища) та інші аспекти, які впливають на функціональність автомобіля.

Одним з важливих елементів математичної моделі є таблиця, яка містить параметри та характеристики ТЗ. Наприклад, така таблиця може включати параметри, які впливають на функціональність автомобіля, такі як маса, довжина, ширина, висота, потужність двигуна, максимальна швидкість, радіус повороту та витрата палива. Ця таблиця може бути використана для подальшого аналізу та розрахунків в рамках математичної моделі ТЗ.

Таблиця 2.1 – Характеристики ТЗ

Параметр	Опис	Значення
Маса, кг	Маса транспортного засобу	0,735
Довжина, м	Довжина транспортного засобу	0,300
Ширина, м	Ширина транспортного засобу	0,240
Висота, м	Висота транспортного засобу	0,150
Радіус повороту, м	Мінімальний радіус повороту	0,300
Привід	На які колеса передається крутний момент з двигуна (двигунів)	4 × 4
Радіус коліс, м	Радіус коліс транспортного засобу з урахуванням шини	0,035
Ширина коліс, м	Ширина коліс транспортного засобу	0,030
Глибина протектору, мм	Глибина протектору шини	3,000

Розробка та використання математичної моделі ТЗ дозволяє здійснювати аналіз, прогнозування та оптимізацію його роботи, враховуючи різні впливи та умови експлуатації. Це допомагає вдосконалювати функціональність транспортного засобу, підвищувати ефективність та безпеку його руху

2.1.1 Визначення математичної моделі для системи безпілотного управління

Математична модель БНТЗ може бути складною і включати різні аспекти, такі як керування, динаміка руху, взаємодія з оточуючим середовищем та інші. Однак, для спрощення можна розглянути базову математичну модель, що описує деякі основні аспекти БНТЗ. Ось кілька складових, які можуть входити до математичної моделі:

- 1) кінематика [11]: Математична модель може включати розрахунки шляху, швидкості та прискорення БНТЗ. Це може включати рух у прямій лінії, повороти, зміну швидкості та інші рухи;
- 2) динаміка [12]: Модель може враховувати фізичні закони динаміки, такі як другий закон Ньютона, щоб описати рух транспортного засобу відповідно до сили, що діє на нього;
- 3) система керування: Математична модель може включати алгоритми керування, які визначають поведінку транспортного засобу на основі вхідних даних, таких як інформація з датчиків, вимоги маршруту, обмеження швидкості тощо;
- 4) датчики та взаємодія з оточуючим середовищем: Математична модель може описувати роботу датчиків, таких як відеокамери, лазерні сканери, радари, які збирають дані про оточуюче середовище та перешкоди. Модель може також включати алгоритми обробки цих даних та врахування їх впливу на рух транспортного засобу.

Це лише загальні принципи, і конкретна математична модель БНТЗ буде залежати від його типу, функцій, сфери в якій планується використовувати та інших важливих факторів.

2.1.2 Розрахунок динаміки руху, кінематики та поведінки транспортного засобу

Математична модель БНТЗ є важливим інструментом для розробки систем керування. Він дозволяє описувати фізику та поведінку транспортних засобів за допомогою математичних рівнянь і параметрів.

Математичні моделі можуть включати такі компоненти, опис руху, взаємодія зовнішніх сил і масово-інерційних властивостей, енергетика, системи керування тощо. та інші аспекти, які впливають на функціональність автомобіля.

Моделі можуть бути представлені у вигляді систем диференціальних рівнянь, матриць і векторів, стохастичних процесів або інших математичних форм залежно від конкретних властивостей ТЗ та поставленої задачі.

Розробка та використання математичної моделі БНТЗ дозволяє здійснювати аналіз, прогнозування та оптимізацію його роботи, враховуючи різні впливи та умови експлуатації. Вона дозволяє розробникам та інженерам проводити віртуальні експерименти, здійснювати вдосконалення та валідацію системи без необхідності фізичних прототипів.

Застосування математичних формул у розрахунках динаміки руху, кінематики та поведінки ТЗ дозволяє нам здійснювати точні розрахунки і прогнозувати рухові характеристики ТЗ з високою точністю. Надалі, ми розглянемо конкретні математичні формули, що застосовуються для визначення різних параметрів.

Кінематика руху транспортного засобу [13]:

Швидкість (V):

$$V = \frac{S}{t}, \quad (2.1)$$

де V – швидкість;

S – відстані, м;

t – час, с.

$$V = \frac{1}{1,1} \approx 0,91 \text{ м/с} \times 3,6 = 3,276 \text{ км/год}$$

Прискорення (a):

$$a = \frac{V}{t}, \quad (2.2)$$

де a – прискорення;

ΔV – зміна швидкості, м/с;

Δt – зміна часу, с.

$$a = \frac{0,91}{0,1} \approx 9,1 \text{ м/с}^2$$

Динаміка руху транспортного засобу:

Сила тяги ($F_{traction}$):

$$F_{traction} = m \times a, \quad (2.3)$$

де $F_{traction}$ – сила тяги;

m – маса транспортного засобу, кг;

a – прискорення, м/с².

$$F_{traction} = 0,735 \times 9,1 \approx 6,69 \text{ Н.}$$

Гальмівний шлях (d_{brakes}):

$$d_{brakes} = V^2 / (2 \times \mu \times g), \quad (2.4)$$

де d_{brakes} – гальмівний шлях;

V – швидкість, м/с;

μ – коефіцієнт тертя;

g – прискорення вільного падіння, м/с².

$$d_{brakes} = \frac{0,91^2}{(2 \times 0,7 \times 9,8)} \approx 0,057 \text{ м} \approx 5,7 \text{ см}$$

Ці математичні формули дозволяють нам розраховувати та аналізувати рухові характеристики ТЗ, враховуючи різні впливи та фактори. Вони є важливими інструментами для вивчення динаміки руху ТЗ і визначення його поведінки. Застосування цих формул дозволяє нам розрахувати швидкість, прискорення, відстань, сили, що діють на ТЗ, а також передбачити його рухові характеристики за різних умов.

Наприклад, за допомогою формул для кінематики руху, ми можемо визначити швидкість ТЗ, враховуючи зміну відстані і часу. Це дозволяє нам оцінити, з якою швидкістю здатен рухатись ТЗ.

У випадку динаміки руху, формули для сил тяги, опору повітря і тертя дають нам можливість визначити величину цих сил і їх вплив на рух ТЗ. Це важливо при проектуванні та оптимізації для досягнення кращої ефективності та економії таких ресурсів як паливо, електроенергія тощо.

Нарешті, формули для поведінки ТЗ дозволяють нам розрахувати гальмівний шлях, враховуючи швидкість, коефіцієнт тертя та прискорення вільного падіння. Це допомагає визначити відстань, яка потребується для повної зупинки при різних умовах.

Отже, використання математичних формул є важливою складовою для розрахунку динаміки руху, кінематики та поведінки ТЗ. Вони надають нам розуміння та можливість прогнозувати рухові процеси, допомагають вдосконалювати ТЗ та забезпечують безпеку та ефективність у транспортній системі.

2.2 Методи керування та оптимізації

Методи керування та оптимізації є важливою складовою системою безпілотного управління, дозволяючи досягти ефективності, точності та безпеки в їх функціонуванні.

Оскільки управління відбувається через оператора, використання ефективних методів керування та оптимізації стає особливо важливим для досягнення максимальної продуктивності та безпеки руху ТЗ.

Операторське керування дозволяє вести нагляд та приймати рішення щодо руху БНТЗ засновано на зібраних даних та інформації про стан системи. У даному розділі розглядаються наступні методи керування та оптимізації:

Регулювання швидкості один з основних аспектів управління БНТЗ. Оператор може контролювати швидкість транспортного засобу в залежності від умов дороги, обмежень швидкості та потреб пасажирів. Методи оптимізації швидкості включають розробку алгоритмів, що враховують прогнозування руху, оптимальну швидкість для зменшення витрат палива та забезпечення комфортного руху.

Керування маршрутом: оператор має можливість планувати та змінювати маршрут БНТЗ в реальному часі. Це включає врахування обмежень доріг, трафіку, умов навколишнього середовища тощо.

Управління поведінкою: Оператор може встановлювати правила та параметри для поведінки БНТЗ, забезпечуючи безпеку та ефективність руху. Це включає управління режимами руху (наприклад, режимами економії енергії або швидкого руху), врахування правил дорожнього руху та реакцію на небезпечні ситуації.

Оптимальне використання ресурсів: Методи оптимізації дозволяють забезпечити ефективне використання ресурсів, таких як паливо, електроенергія або заряд БНТЗ. Оператор може встановлювати пріоритети щодо використання ресурсів та розробляти стратегії для мінімізації витрат та забезпечення тривалої роботи ТЗ.

Ці методи керування та оптимізації допомагають оператору забезпечити безпеку, ефективність та комфорт при управлінні безпілотним наземним транспортним засобом. Враховуючи особливості кожної конкретної ситуації, оператор може використовувати ці методи для досягнення найкращих результатів в управлінні БНТЗ.

Використання методів керування та оптимізації сприяє покращенню продуктивності, безпеки та надійності систем безпілотного управління. Вони дозволяють забезпечити ефективне та економічно вигідне функціонування транспортного засобу, сприяючи розвитку автономної та інтелектуальної мобільності.

2.2.1 Визначення стратегій керування для безпілотного управління

У керування для безпілотного управління даному розділі розглядаються стратегії ТЗ з участю оператора. У таких системах оператор виконує роль наглядача та приймає рішення на основі інформації, отриманої з безпілотної системи.

Оператор використовує різні стратегії для керування безпілотним транспортним засобом. Деякі з них включають:

1) стратегія віддаленого керування [14]: оператор знаходиться віддалено від транспортного засобу і керує ним за допомогою дистанційних засобів керування, таких як джойстики, рульові колеса або комп'ютерні інтерфейси. Оператор отримує відеопотік та іншу інформацію з безпілотної системи для прийняття рішень;

2) стратегія навчання з учителем [15]: оператор надає безпілотній системі зразки правильних дій і навчає її виконувати певні завдання. Система використовує ці зразки для навчання та покращення свого алгоритму управління;

3) стратегія гібридного керування у цій стратегії оператор співпрацює з безпілотною системою, виконуючи певні завдання, а решту роботи доручає системі. Наприклад, оператор може виконувати завдання з навігації та вирішення складних ситуацій, тоді як система відповідає за базові рухи та дотримання правил дорожнього руху;

4) стратегія екстреного втручання: оператор втручається у безпілотну систему тільки у випадку непередбачених ситуацій або аварійних ситуацій, коли система не може самостійно прийняти рішення. Оператор має можливість швидко реагувати та прийняти необхідні дії для забезпечення безпеки та успішного виконання місії.

Ці стратегії керування з участю оператора можуть бути комбіновані та адаптовані в залежності від конкретних умов, завдань та потреб безпілотної системи. Вони дозволяють поєднати переваги автоматизованого безпілотного управління з експертними знаннями та рішеннями оператора, забезпечуючи надійність, безпеку та ефективність роботи ТЗ.

2.2.2 Оптимізація параметрів та функцій керування

У даному розділі розглядаються методи оптимізації параметрів та функцій керування БНТЗ з участю оператора. Управління БНТЗ під час експлуатації може здійснюватись оператором, який має можливість втручатись у роботу системи та приймати рішення залежно від ситуації.

Оптимізація параметрів та функцій керування з участю оператора передбачає врахування його ролі та можливостей при прийнятті рішень. Оператор може використовувати різні вхідні дані, такі як сенсорна інформація, дані з картографічних систем, та власний досвід, для забезпечення ефективного та безпечного управління БНТЗ.

Стратегії оптимізації з участю оператора можуть включати інтерактивне взаємодію з системою керування, де оператор може вносити корективи у вироблені алгоритми та рішення. Оптимізація такого роду забезпечує гнучкість та адаптивність управління, дозволяючи оператору враховувати непередбачувані обставини та змінні умови експлуатації.

Для забезпечення оптимального управління БНТЗ з участю оператора використовуються різні підходи, такі як гібридні системи керування, де комбінуються автономне та операторське управління, а також системи підтримки прийняття рішень, які надають оператору інформацію та рекомендації для прийняття оптимальних рішень.

Оптимізація параметрів та функцій керування з участю оператора має на меті покращення продуктивності, безпеки та ефективності управління БНТЗ. Вона дозволяє враховувати людський фактор та експертні знання при процесі прийняття рішень, що сприяє оптимальному функціонуванню системи у різних умовах.

Існують методи та підходи до оптимізації параметрів та функцій керування з участю оператора в управлінні БНТЗ:

- 1) еволюційні алгоритми [16]: цей підхід базується на імітації природного еволюційного процесу. Використовуються генетичні алгоритми, алгоритми частинок та інші методи, що дозволяють шукати оптимальні рішення шляхом множинного пошуку у просторі параметрів. Оператор може впливати на процес оптимізації шляхом встановлення обмежень, переваг або накладання обмежень на деякі параметри;

2) методи зворотного зв'язку: використовуються алгоритми, які враховують відгуки оператора та виконують корекцію параметрів та функцій керування на основі цих відгуків. Оператор може здійснювати натиск на систему або коригувати її роботу, спираючись на відхилення від бажаного результату;

3) методи машинного навчання [17]: використовуються алгоритми навчання на основі даних, які дозволяють системі безпілотного управління самостійно вивчати та оптимізувати свою роботу. Оператор може брати участь у процесі навчання, надавати початкові дані, контролювати процес та оцінювати результати;

4) експертні системи [18]: використовуються бази знань та правил, розроблені експертами у галузі безпілотного управління. Оператор взаємодіє з системою, використовуючи експертні знання та правила, що дозволяють системі приймати оптимальні рішення в реальному часі;

5) гібридні системи: використовуються комбінації різних методів та підходів з метою досягнення найкращих результатів оптимізації. Оператор може впливати на вибір та налаштування різних компонентів гібридної системи.

Вибір конкретного методу або підходу залежить від вимог до системи, обмежень, доступних ресурсів та експертного досвіду оператора. Розуміння принципів та особливостей різних методів допоможе оператору визначити оптимальну стратегію керування та оптимізації для управління БНТЗ.

2.3 Математичні моделі енергоживлення

Модель енергоживлення системами БНТЗ враховує різні аспекти, такі як потужність системи (P), тривалість руху (t) та коефіцієнт ефективності (η). Ця модель дозволяє визначити загальну енергію, яку витрачає система керування протягом певного періоду часу.

Потужність системи керування вимірюється в одиницях енергії, яку споживає система за одиницю часу (наприклад, ваттах або кіловаттах). Це

включає споживання енергії всіма компонентами системи керування, такими як процесори, сенсори, комунікаційні модулі тощо.

Тривалість руху (t) вимірюється в одиницях часу (наприклад, секундах або годинах) і представляє собою період, протягом якого система безперервно працює або знаходиться в режимі руху. Цей параметр відображає загальний час, протягом якого система споживає електроенергію.

Коефіцієнт ефективності (η) враховує втрати енергії, пов'язані з роботою системи керування. Це можуть бути втрати енергії внаслідок опору проводів, некерованих процесів, неефективності елементів системи тощо. Коефіцієнт ефективності знаходиться в межах від 0 до 1, де 1 відповідає ідеальній ефективності, а 0 - повній втраті енергії.

Математичні моделі споживання електроенергії в системі БНТЗ відіграють важливу роль у розрахунках та оптимізації енергетичної ефективності. Нижче наведено кілька математичних моделей, які використовуються для оцінки споживання електроенергії:

- 1) модель споживання електроенергії моторами коліс;
- 2) модель споживання електроенергії системами керування;
- 3) модель загального споживання електроенергії.

Ці математичні моделі дозволяють оцінити споживання електроенергії різними компонентами системи безпілотного управління. Це допомагає виявити енергоефективність системи та здійснити оптимізацію з метою зниження споживання електроенергії і підвищення тривалості роботи системи на одному заряді акумулятора.

2.3.1 Розрахунок енергоживлення систем безпілотного управління

Для розрахунків енергоживлення [19] систем БНТЗ варто враховувати P електро модулів (повний перелік енергоспоживачів та їх характеристики занесені в таблицю (2.2). Для необхідно знати значення напруги (U) та сили

струму (I), які споживає певний модуль. P може бути обчислена за допомогою наступної формули:

$$P = U \times I, \quad (2.5)$$

де P – потужність;
 U – напруга, В;
 I – сила струму, А.
Використовуючи

Напруга та сила струму певних елементів були внесені в таблицю 2.2. також туди були внесені результати розрахунків потужності.

$$P_{motor} = 8 \times 0,25 = 2 \text{ Вт}$$

$$P_{control} = 5 \times 0,35 = 1.75 \text{ Вт}$$

Математична модель споживання електроенергії у системі безпілотного управління дозволяє оцінити енергетичні витрати, пов'язані з рухом транспортного засобу. Основною формулою для розрахунку споживання енергії є:

$$W = P \times t \times \eta, \quad (2.6)$$

де W – споживана енергія, вимірюється у джоулях (Дж або Вт);
 P – потужність, вимірюється у ваттах (Вт);
 t – час руху транспортного засобу, вимірювань у секундах (с);
 η – коефіцієнт ефективності, який відображає співвідношення між вихідною потужністю та витратою електроенергії, виражений у відсотках (%). Коефіцієнт ефективності знаходиться в межах від 0 до 1, де 1 відповідає ідеальній ефективності, а 0 - повній втраті енергії.

Знаючи потужність двигунів можна вирахувати енергію яку вони споживають:

$$W_{motor} = 2 \times 3600 \times 1 = 7200 \text{ Дж}$$

Далі потрібно визначити обсяг електроенергії, який буде спожитий системою керування під час руху або функціонування. За тією самою формулою розрахуємо споживання енергії контролера та драйверу:

$$W_{control} = 1,75 \times 3600 \times 1 = 6300 \text{ Дж}$$

$$W_{driver} = 25 \times 3600 \times 1 = 90000 \text{ Дж}$$

Далі враховуються сумарне споживання енергії всіх компонентів системи безпілотного управління, включаючи мотори коліс, системи керування та інші електричні пристрої.

$$W_{total} = W_{consumed} + W_{control} + W_{driver} \quad (2.7)$$

$$W_{total} = 5 \times 7200 + 6300 + 2 \times 90000 = 222300 \text{ Дж} = 222,3 \text{ кДж}$$

Отже при використанні системи на максимальній потужності вона потребує щонайменше 222,3 кДж, або 61,75 Вт/год. Чим більша потужність системи керування, час руху та коефіцієнт ефективності, тим більше енергії буде спожито. Оптимізація цих параметрів може допомогти знизити споживання електроенергії та покращити енергетичну ефективність системи БНТЗ.

Таблиця 2.2 – Короткі характеристики комплектуючих

Назва	Максимальна напруга, В	Максимальна сила струму, А	Максимальне споживання енергії, Вт,	Кількість
ESP32-CAM	5 В	0,35 А	1,75 Вт	1
Драйвер L298N	46 В	3,00 А	25,00 Вт	2
Електродвигун з редуктором(для мотор-колеса)	8 В	0,25 А	2 Вт	4
Електродвигун(для системи поворотів коліс)	8 В	0,25 А	2 Вт	1

Відомості про напругу та струм елементів зазначених в таблиці було взято з онлайн сторінки магазину радіотехніки.

2.3.2 Визначення оптимальних стратегій енергозбереження та ємності акумуляторної батареї.

Визначення оптимальних стратегій енергозбереження включає математичні розрахунки, що дозволяють встановити оптимальні параметри та режими роботи системи з метою мінімізації споживання енергії. Для зменшення енергоживлення можна забезпечити оптимальне використання енергії шляхом розробки ефективних алгоритмів керування та оптимізації параметрів функцій керування. Це може включати розрахунок оптимальних траєкторій руху, вибір оптимальних швидкостей, активне управління споживанням енергії.

Використовуючи формулу (2.5) та враховуючи характеристики комплектуючих можна вирахувати їх оптимальна потужність. Наприклад головний контролер здатен працювати від двох напруг 3,3 В та 5 В. Так як використовується батарея на 6 В можна роз оптимальну потужність для дарайвера L298N та двигунів.

$$P_{control} = 3,3 \times 0,35 = 1,155 \text{ Вт}$$

$$P_{driver} = 12 \times 2 = 24 \text{ Вт}$$

$$P_{motor} = 8 \times 0,25 = 2 \text{ Вт}$$

Наступним кроком знайдемо оптимальне енергоспоживання двигунів та систем керування за формулою (2.6) , після чого розрахуємо сумарне енергоспоживання за формулою(2.7):

$$W_{motor} = 2 \times 3600 \times 0,9 = 6480 \text{ Дж}$$

$$W_{control} = 1,155 \times 3600 \times 0,9 \approx 3745 \text{ Дж}$$

$$W_{control} = 24 \times 3600 \times 0,7 = 60480 \text{ Дж}$$

$$W_{total} = 5 \times 6480 + 3745 + 60480 = 96625 \text{ Дж} = 96,6 \text{ кДж}$$

Отже при обмежені напруги було значно зменшено споживання систем БНТЗ. Тепер система потребує 96,6 кДж, або 26,6 Вт/год. Що становить приблизно 38 % від споживання енергії на максимальній потужності.

Для мінімізації споживання електроенергії в системі безпілотного управління [20] також можна використовувати наступні стратегії:

Використання енергоефективного обладнання: використання компонентів та пристроїв, які мають низький споживання енергії, може допомогти знизити загальний розхід енергії системи. Наприклад, вибір енергоефективних процесорів, сенсорів та інших елементів управління.

Управління енергією в режимах очікування: встановлення стратегій управління енергією в режимах очікування або бездіяльності системи. Наприклад, перехід в режим сну або вимкнення окремих компонентів, коли вони не використовуються, може знизити розхід енергії.

Енергетична оптимізація алгоритмів: аналіз та оптимізація алгоритмів керування з метою зниження споживання енергії. Це може включати заміну складних обчислень на більш ефективні алгоритми, використання кешування даних або оптимізацію протоколів передачі даних.

Акумуляторна батарея (АКБ) є одним з найпоширеніших та важливих джерел енергії в сучасному світі. Вона забезпечує мобільність та незалежність в різних галузях, включаючи електроніку, транспорт, промисловість та багато іншого. АКБ дозволяє зберігати та постачати електричну енергію в різних пристроях та системах, надаючи їм можливість працювати без прив'язки до зовнішнього джерела живлення.

Використання АКБ є особливо важливим в контексті безперебійного живлення, мобільних пристроїв, електромобілів, сонячних енергетичних систем та багатьох інших застосувань, де надійне та ефективне живлення є критично важливим. Для досягнення оптимальної роботи та тривалого часу автономної роботи необхідно правильно визначити параметри та властивості АКБ, такі як ємність, напруга, струм, технологія, температурні умови та інші фактори.

У світі існує кілька типів акумуляторних батарей [21] з різними характеристиками та застосуваннями. Деякі з них включають:

1) свинцево-кислотні акумулятори (Lead-Acid Batteries): Це найпоширеніший тип АКБ, який широко використовується в автомобілях, мотоциклах та системах резервного живлення. Вони мають високу ємність, низьку вартість і можуть працювати в широкому діапазоні температур;

2) літій-іонні акумулятори (Lithium-Ion Batteries): Цей тип АКБ використовується в різних пристроях, таких як мобільні телефони, ноутбуки, електронні скутери та електромобілі. Вони мають високу енергетичну щільність, низьку вагу і довгий термін служби, але вимагають спеціального управління та контролю;

3) нікель-метал-гідридні акумулятори (Nickel-Metal Hydride Batteries): Ці батареї використовуються в різних електронних пристроях, таких як фотоапарати, портативні інструменти та електричні іграшки. Вони мають помірну ємність, добру стійкість до перезарядки і меншу вразливість до екологічних проблем порівняно з іншими типами батарей;

4) літій-полімерні акумулятори (Lithium-Polymer Batteries): Цей тип АКБ широко використовується в різних портативних пристроях, таких як планшетні комп'ютери, смартфони та Bluetooth-гарнітури. Вони мають гнучку форму, високу енергетичну щільність і можуть бути виготовлені у різних розмірах і конфігураціях.

Маючи дані отримані під час попередніх розрахунків можна розрахувати ємність (АКБ) [22]. Щоб розрахувати потрібну ємність батареї, можна використати наступну формулу:

$$Q = \frac{P \times t}{V \times k}, \quad (2.8)$$

де Q – ємність батареї (А год);

P – потужність в Вт;

t – час використання в годинах;

V_{battery} – напруга батареї В;

k – коефіцієнт, що відображає яка частина ємності АКБ використовується(%). Знаходиться в межах від 0 до 1, де 1 відповідає повному розряду батареї, а 0 – повному збереженню заряду.

$$Q = \frac{26,8 \times 1}{12 \times 0,95} = \frac{26,8}{12,6} \approx 2,1 \text{ А год} = 2100 \text{ мА год}$$

Отже при використанні АКБ з напругою 12 В протягом години її ємність повинна становити не менше 2100 мА год

2.4 Математична модель коливань та прохідності підвіски транспортного засобу

Математична модель коливальної підвіски [23] транспортного засобу є важливим інструментом для дослідження та аналізу поведінки системи в умовах коливань. Вона базується на фізичних законах та математичних принципах, що дозволяють моделювати та прогнозувати рух транспортного засобу на нерівних дорогах та в умовах зовнішніх збурень.

Основою математичної моделі є рівняння руху, яке враховує такі параметри, як маса транспортного засобу, жорсткість пружини, амортизація та зовнішні сили, що діють на систему. Рух системи описується як коливання, яке може бути гармонійним або не гармонійним залежно від умов руху та властивостей підвіски

У математичній моделі використовуються різні підходи та методи, такі як диференціальні рівняння, системи лінійних алгебраїчних рівнянь, методи чисельного моделювання та інші. Вона дозволяє аналізувати вплив різних параметрів, таких як жорсткість пружини, амортизація, маса транспортного засобу та інші фактори, на характеристики коливань та поведінку системи.

Математична модель коливальної підвіски транспортного засобу має велике практичне застосування в автомобільній промисловості, де вона допомагає вдосконалювати та оптимізувати конструкцію підвіски для забезпечення кращої стійкості, комфорту та безпеки руху. Крім того, вона є важливим інструментом для вивчення та аналізу динамічних характеристик

транспортних засобів. За допомогою математичної моделі можна встановити оптимальні параметри підвіски, які забезпечують стабільність та керованість автомобіля при різних умовах руху

Математична модель дозволяє проводити чисельні симуляції та аналізувати вплив різних факторів на динаміку колісного транспорту. За допомогою моделювання можна вивчати поведінку системи при різних швидкостях, нахилених поверхнях дороги, зміні навантаження та інших умовах руху. Це дозволяє виявити потенційні проблеми та удосконалити конструкцію підвіски для покращення її характеристик.

Окрім цього, математична модель може бути використана для прогнозування впливу різних зовнішніх збурень, таких як нерівності дороги, гальмування, прискорення та інші, на динаміку транспортного засобу. Це дозволяє підвищити безпеку руху, забезпечити оптимальне функціонування підвіски та знизити знос та пошкодження елементів підвіски.

Таким чином, математична модель коливальної підвіски транспортного засобу є незамінним інструментом для дослідження та оптимізації підвісної системи. Вона дозволяє аналізувати та прогнозувати динамічні характеристики транспортного засобу, покращувати його стійкість, комфорт та безпеку руху, а також ефективно використовувати ресурси підвіски.

2.4.1 Визначення частоти коливань

Частота коливань є однією з основних характеристик коливальних систем, включаючи коливання в підвісці автомобіля. Вона визначає кількість повних коливань, які відбуваються за одиницю часу. Частоту коливань позначають символом « ν » і вимірюють в герцах (Гц).

Частота коливань (ν) в коливальній системі може бути розрахована з використанням математичної формули:

$$\nu = \frac{1}{T}, \quad (2.9)$$

де T – період коливань, який визначається як час, необхідний для виконання одного повного коливання.

Частота коливань залежить [23] від характеристик системи, таких як жорсткість пружин, маса системи та амортизаційні властивості. Період коливань може бути обчислений за формулою:

$$T = 2\pi \times \sqrt{\frac{m}{k}}, \quad (2.10)$$

де T – період коливань;

m – маса системи;

k – жорсткість пружини.

Знання частоти коливань є важливим для проектування та налаштування підвіски автомобіля. Правильне підбір частоти коливань допомагає досягти балансу між комфортом пасажирів та стійкістю автомобіля на дорозі. Частота коливань також впливає на динамічні властивості автомобіля, його керованість та маневреність.

Розрахунки періоду коливань:

$$T = 2\pi \times \sqrt{\frac{0,2 \text{ кл}}{6 \text{ Н/м}}} \approx 0,728 \text{ с}$$

Частота коливань:

$$\nu = \frac{1}{0,728} \approx 1,37$$

Таким чином, частота коливань колеса становить приблизно 1,37 Гц

Параметри прохідності для автомобілів [24] включають ряд характеристик, які визначають їх здатність подолати перешкоди та рухатися в різних умовах терену. Основними параметрами прохідності є:

Таблиця 2.3 – Показники прохідності

Найменування	Опис	Показник
Відстань від землі (кліренс):	це відстань від нижньої точки автомобіля до поверхні дороги. Більший кліренс дозволяє автомобілю проїжджати перешкоди висотою, такі як валуни, камені, високі бордюри та нерівності. (м)	0,04
Кути нахилу	це кути, які автомобіль може подолати без ризику перекидання або впадання. Це включає кут нахилу вперед (максимальне підйомне здатність), кут нахилу назад (максимальне здатність до спуску) і кут бокового нахилу.	40
Кут з'їзду та кут підйому:	ці параметри вказують на максимальний кут нахилу, який автомобіль може подолати при з'їзді або підйомі на дорогу чи перешкоду. Чим більше ці кути, тим краща прохідність автомобіля.	Відсутні
Глибина переходу води:	це максимальна глибина води, яку автомобіль може безпечно переїхати без негативних наслідків для його компонентів і систем (м).	0,02
Система повного приводу	Наявність системи повного приводу дозволяє автомобілю розподіляти потужність на всі колеса, що поліпшує прохідність та зчеплення з поверхнею.	4 × 4
Система блокування диференціала	блокування диференціала дозволяє заблокувати роботу диференціала та забезпечити однаковий крутний момент на	Відсутня

Найменування	Опис	Показник
	обох колесах однієї осі, що поліпшує прохідність у важких умовах	

Ці параметри визначаються виробниками автомобілів і можуть варіюватися залежно від типу транспортного засобу, його призначення та характеристик.

Висновки до розділу 2

У другому розділі були розглянуті математичні методи та їх застосування в системі безпілотного управління транспортним засобом. Було визначено математичну модель системи, що дозволяє описати рух та поведінку безпілотного транспортного засобу.

Також було досліджено визначення стратегій керування. Це дозволяє оператору здійснювати керування транспортним засобом з віддаленої локації з використанням оптимальних стратегій та алгоритмів.

Застосування математичних моделей та формул дозволяє аналізувати та прогнозувати рух транспортного засобу, враховуючи різні фактори та умови експлуатації. Це дозволяє розробляти оптимальні стратегії керування, що сприяють досягненню бажаних результатів у безпілотному управлінні.

В даному розділі були проведені розрахунки такі як знаходження: швидкості, сили тяги чи гальмівного шляху. Також було вираховано енергоживлення системи та потрібну ємність батареї.

3 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Вибір програмних і апаратних компонентів

Існує велика різноманітність програмного забезпечення та апаратні компоненти. Серцем нашого проекту є обладнання обчислювальна платформа. Вибір платформи безпосередньо залежить від вимоги до проекту. Існує 2 основні марки мікроконтролерів друковані плати: Arduino і Raspberry Pi. Так само доведеться вибирати і інші компоненти:

- тип двигуна;
- тип датчика.

Залежно від обраної платформи буде також залежати вибір середовища програмування. Управління також відіграє величезну роль. Тож ми повинні вирішити, якою буде система керування роботом. Вибір невеликий:

- ПК;
- мобільна платформа;
- спеціальні програмовані пристрої.

3.1.1 Вибір апаратної обчислювальної платформи

У даному підрозділі розглядається вибір апаратної обчислювальної платформи для системи БНТЗ. Після аналізу різних варіантів, було прийняте рішення використовувати ESP32-CAM [25] як основну апаратну платформу.

ESP32-CAM є мікроконтролером, який поєднує в собі потужність мікропроцесора ESP32 та можливості камери. Ця платформа має ряд переваг, які відповідають вимогам системи БНТЗ.



Рисунок 3.1 – ESP32-CAM

Однією з ключових переваг ESP32-CAM є її потужний мікропроцесор ESP32, який забезпечує високу продуктивність обчислень та швидку обробку даних. Він має достатньо обчислювальних ресурсів для виконання складних алгоритмів, а також може працювати в реальному часі, що дозволяє системі безпілотного наземного транспортного засобу швидко реагувати на зміни в оточенні.

Таблиця 3.1 – Характеристики ESP32-CAM

Контролер	ESP32-S, 2 ядра, 32-біт
Робоча частота процесора	240 МГц
Модуль бездротового зв'язку	WiFi 802.11 b / g / n , Bluetooth 4.2 LE
Флеш Пам'ять	Вбудована: 520 КБ / Зовнішня: 4 МБайт
Підтримувані інтерфейси	UART/SPI/іес/PWM/ADC/DAC
Напруга живлення	5В, 3.3В
Камера:	OV2640 2 Мп

Крім того, ESP32 CAM має вбудовану камеру, що є необхідною функціональністю для системи БНТЗ. Це дозволяє отримувати зображення з оточення та використовувати їх для аналізу, розпізнавання об'єктів та прийняття рішень з управління.

ESP32 CAM також володіє низьким споживанням енергії, що є важливим фактором для безперебійної роботи системи БНТЗ. Це дозволяє забезпечити тривалу автономну роботу транспортного засобу без частого заряджання акумуляторів.

Загалом, ESP32 CAM відповідає вимогам системи БНТЗ щодо продуктивності, обробки даних в реальному часі, наявності камери та енергоефективності. Його використання як апаратної обчислювальної платформи дозволить ефективно реалізувати функції безпілотного управління та обробку даних в системі БНТЗ.

3.1.2 Вибір двигунів

При проектуванні БНТЗ необхідно розуміти, що він у своїй конструкції передбачає двигун (двигуни), який може обертати та/або переміщувати частину або всю конструкцію.

Двигун - цей пристрій перетворює будь-який вид енергії в механічну.

Різні конструкції БНТЗ віддають перевагу різним типам двигунів. Технічно вигідно розглядати двигуни постійного струму, які перетворюють електричну енергію в механічне обертання, оскільки вони спрощують програмне керування, схеми та конструкцію безпілотних ТЗ. В даній роботі будуть використовуватись колекторні двигуни з редуктором 1:48 [26]. Також для контролю двигунів був підібраний драйвер L298N [27].

Колекторний двигун - це електрична машина, в якій датчик положення ротора і перемикачі струму в обмотках є одним пристроєм - щітковий колекторний блок.

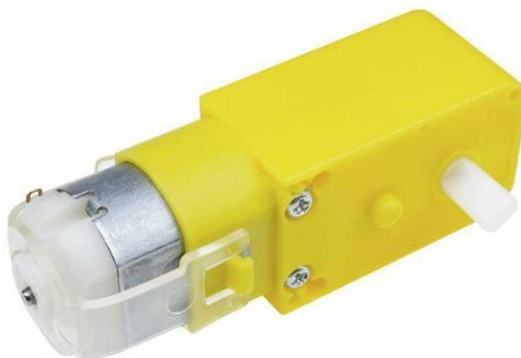


Рисунок 3.2 – Колекторний двигун з ротором 1:48

Дані двигуни мають просте управління, високий початковий крутний момент та доступну ціну. Однак, він також має недоліки, такі як знос щіток, електромагнітні шуми та обмежену швидкість обертання. Порівняно з безколекторними двигунами або кроковими двигунами, колекторні двигуни мають просту конструкцію та нижчу вартість.

Драйвер L298N – це інтегральна схема, яка використовується для керування двигунами, зокрема постійними струму та кроковими двигунами. Він забезпечує можливість зміни напрямку обертання та керування швидкістю обертання двигуна.

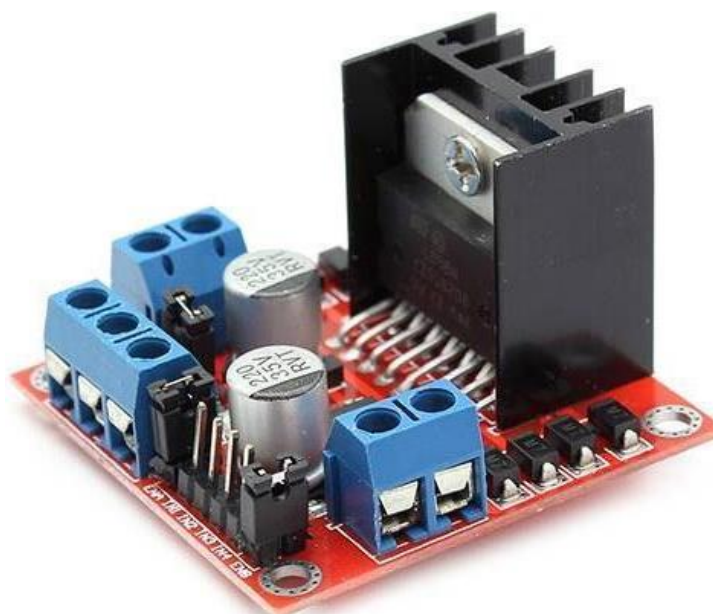


Рисунок 3.3 – Драйвер L298N

Драйвер L298N має два Н-мости, що дозволяє керувати двома незалежними двигунами. Кожен Н-міст містить два транзистори, які працюють у режимі перемикачів, що дозволяє керувати напрямком струму через двигун. Драйвер може працювати з напругою живлення від 5 до 35 Вольт і здатний постачати до 2 Ампер струму на кожен канал.

Для керування драйвером L298N необхідно подати відповідні сигнали на його керуючі входи. Для керування напрямком руху та зупинки двигуна використовуються два керуючі входи, а для керування швидкістю обертання - один аналоговий або шим сигнал.

Драйвер L298N має кілька переваг, зокрема високий струмовий потужність, можливість керування двома двигунами, простота використання та доступна ціна. Однак, він також має деякі обмеження, такі як велика втрата потужності через внутрішні діоди, обмежена швидкість перемикачів транзисторів та великі габаритні розміри.

Драйвер L298N часто використовується в робототехніці, автоматизації та інших проектах, де потрібне керування двигунами зі змінним напрямком обертання та швидкістю. Він є популярним вибором серед ентузіастів та розробників, які працюють з рухомими систем

3.1.3 Вибір середовища програмування

Програми для ESP32-CAM написані на звичайному C++, доповнені простими функціями керування контактами введення-виводу. Для того, щоб полегшити подальшу роботу можна використовувати безкоштовне середовище програмування «Arduino IDE», яке може працювати на Windows, Mac OS і Linux.

Середовище розробки Arduino складається з вбудованого текстового редактора, області повідомлень, текстового вікна (консолі), панелі інструментів з кнопками для часто використовуваних команд і кількох меню. Для завантаження програм і спілкування середовище розробки підключається до апаратного забезпечення Arduino. Програми, написані в середовищі Arduino,

називаються скетчами. Ескізи пишуться в текстовому редакторі з інструментами для вирізання/вставки, пошуку/заміни тексту. Під час збереження та експорту проектів в області сповіщень з'являються інструкції та можуть з'являтися помилки. Вікно виведення тексту (консоль) відображає повідомлення Arduino, включаючи повні звіти про помилки та іншу інформацію. Кнопки панелі інструментів дозволяють перевіряти та зберігати програми, створювати, відкривати та зберігати ескізи, а також відкривати моніторинг послідовної шини.

3.1.4 Вибір середовища управління роботом

Даний БНТЗ буде керуватися через мобільний пристрій на платформі Android.

Android – операційна система для смартфонів, інтернет-планшетів, електронних книг, цифрових плеєрів, годинників, ігрові консолі, нетбуки, смартбуки, Google Glass, телевізори і інші пристрої. Заснований на ядрі Linux і його власної реалізації віртуальна машина Java від Google..

3.2 Реалізація проєкту

Для реалізації цього проєкту необхідно розробити апаратну частину, реалізувати скетч та системи управління.

Для реалізації апаратної частини необхідно розмістити всі потрібні нам компоненти на шасі та підключити їх.

Щоб реалізувати скетч, необхідно бути знайомі з функціями середовище програмування, використовувати необхідні бібліотеки і Алгоритм.

Усім механізмом керуватиме мобільний додаток на платформі Android. Для цього потрібно створити програму та завантажити її мобільний телефон.

3.2.1 Реалізація апаратної частини керування

Апаратною частиною є в першу чергу механізм складання фізичних компонентів майбутнього проєкту.

Необхідно зібрати всі компоненти майбутньої платформи, мотор постійного струму, мотори з редукторами, мікроконтролер, драйвер для управління двигунами, блок живлення або АКБ та додаткові компоненти такі як лінійні стабілізатори чи UART адаптер.

3.2.2 Вимоги до підключення двигунів, драйверів та контролеру

Для підключення колекторних двигунів постійного струму до драйвера L298N потрібно виконати наступні кроки:

Для початку варто переконатись, що напруга живлення драйвера L298N відповідає вимогам вашого двигуна. Зазвичай драйвер L298N працює з напругою живлення в діапазоні від 5 до 35 В.

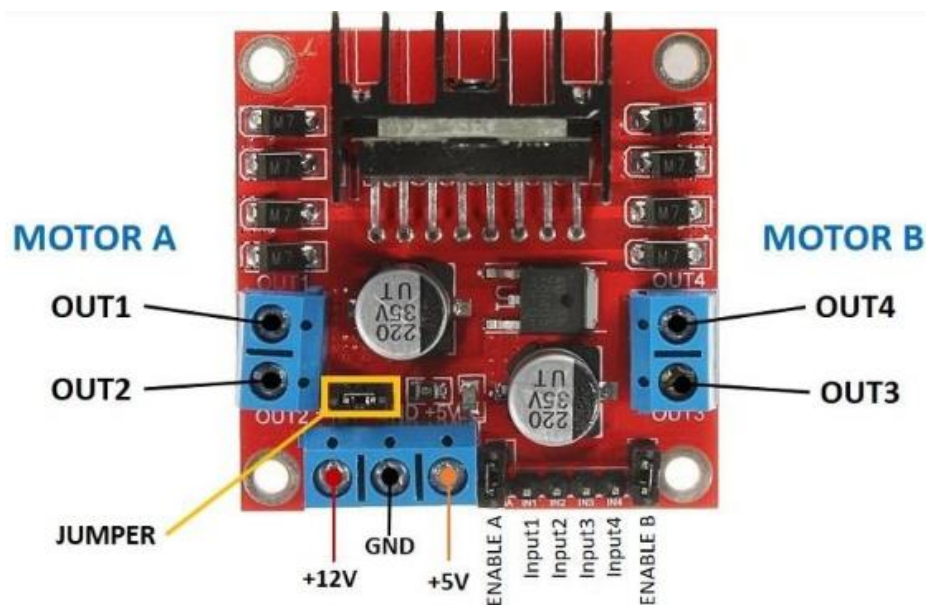


Рисунок 3.4 – Найменування пінів драйверу L298N

Підключаємо джерело живлення (наприклад, акумулятор або джерело постійного струму) до контактів «+12V» та «GND» на драйвері L298N. Варто зазначити що при використанні двигунів та джерел живлення з напругою більше 12 В варто зняти перемичку «JUMPER» та подати напругу в 5 В на відповідний контакт для живлення логіки драйверу.

Наступним кроком підключаємо двигуни до драйвера L298N. Для кожного двигуна використовуємо два контакти на драйвері. Зазвичай це контакти

«OUT1» і «OUT2» для першого двигуна, а також «OUT3» і «OUT4» для другого двигуна.

Підключаємо сигнальні контакти драйвера L298N до нашого керуючого пристрою, наприклад мікроконтролера. Для керування кожним двигуном використовуйте два сигнальні контакти. Зазвичай це контакти «ENABEL A», «IN1» і «IN2» для першого двигуна, а також «ENABEL B», «IN3» і «IN4» для другого двигуна.

Для підключення драйвера L298N до ESP32-Cam потрібно виконати наступні кроки:

Для підключення ESP32-Cam до драйвера L298N. Потрібно підключити виходи ESP32-Cam до вхідних клем драйвера для керування двигунами. Використовуйте вільні GPIO-піни ESP32-Cam для керування напрямком руху (IN1, IN2, IN3, IN4) та швидкістю (ENA, ENB) двигунів.



Рисунок 3.5 – Розміщення пінів ESP32-CAM

Також потрібно заземлити спільний негативний полюс ESP32-Cam та драйвера L298N. Тобто підключити GND ESP32-Cam до GND драйвера L298N. Підключаючи живлення до ESP32 потрібно враховувати що дана плата не любить зміни напруг, тому потрібно використовувати стабілізатори.



Рисунок 3.6 – Стабілізатор напруги AMS1117

Даний стабілізатор є модульним. На вхід приймає від 6 до 18 В при цьому на виході буде стабільні 3,3 В, та підтримує силу струму до 800 мА год.

Розібравшись з пінами елементів та можливостями їх підключення варто зробити макетну схему. Це без проблем можна зробити в інструменті Fritzing.

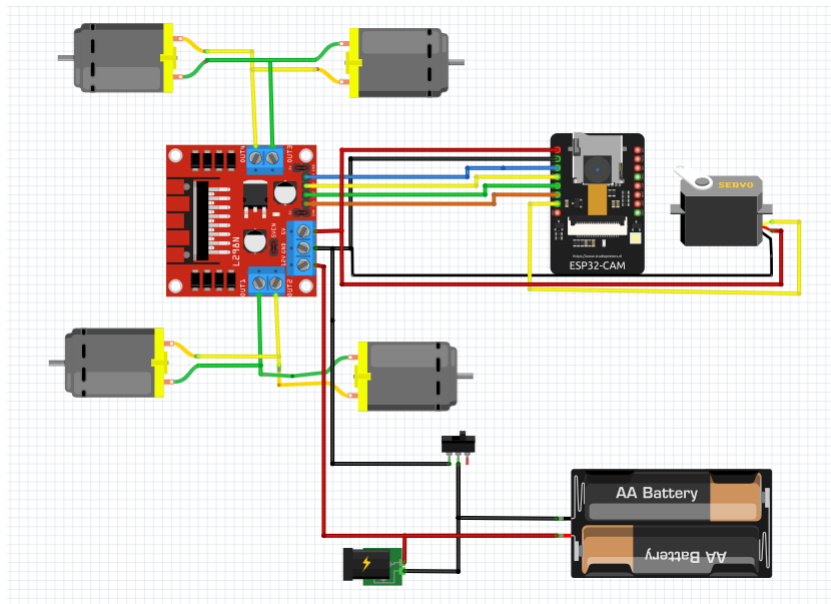


Рисунок 3.7 – Схема підключення компонентів розроблена в Fritzing

Fritzing – це візуальний інструмент для проектування електронних схем, розроблений спеціально для мейкерів, студентів та ентузіастів. Він надає можливість створювати схеми електричних кіл, макетувати плати та створювати схематичні діаграми. Fritzing має простий інтуїтивно зрозумілий інтерфейс, який дозволяє використовувати графічні символи компонентів і з'єднувати їх за допомогою проводів.

За допомогою Fritzing можна створювати схеми електричних колів з використанням різних електронних компонентів, таких як резистори, конденсатори, транзистори, мікроконтролери тощо. Також можна макетувати плати, розміщуючи компоненти на віртуальній дошці та з'єднуючи їх проводами. Fritzing також надає можливість створювати схематичні діаграми, що використовуються для подання зв'язків між компонентами у більш абстрактному вигляді.

Fritzing є корисним інструментом для розробки електроніки, дозволяючи створювати, документувати і ділитися схемами та проектами. Він підтримується активною спільнотою користувачів і має велику колекцію готових символів компонентів для використання.

3.2.3 Реалізація скетча

Перед початком роботи над скетчем слід налаштувати Arduino IDE та як воно не має прямого доступу до роботи з потрібними нам платами. Для цього заходимо в налаштування та додаємо посилання як зображено на рисунку 3.8:



Рисунок 3.8 – Налаштування Arduino IDE

Після цього йдемо в Boards Manager та завантажуюємо набір потрібних плат. Вибираємо потрібну нам плату в нашому випадку AI Thinker ESP32-CAM та вказуємо COM порт. Також варто підвантажити стандартний скетч CameraWebServer який підвантажить конфігурацію та налаштування камер які працюють з нашою платою.

В скетчі буде зазначено декілька моделей камери. Залишаємо ту де вказано нашу модель плати.

```
#define CAMERA_MODEL_AI_THINKER
```

Наступним кроком вказуємо дані нашого Wi-Fi для підключення.

```
const char* ssid = "*****";  
const char* password = "*****";
```

Визначаємо контакти двигуна.

```
const int IN1 = 12;  
const int IN2 = 13;  
const int IN3 = 14;  
const int IN4 = 15;
```

Далі створимо функцію *initMotors()* та налаштуємо ШИМ-сигнали для моторів і прикріпимо до їх каналів відповідні піни.

```
void initMotors()  
{  
    ledcSetup(3, 2000, 8);  
    ledcSetup(4, 2000, 8);  
    ledcSetup(5, 2000, 8);  
    ledcSetup(6, 2000, 8);  
    ledcAttachPin(IN1, 3);  
    ledcAttachPin(IN2, 4);  
    ledcAttachPin(IN3, 5);  
    ledcAttachPin(IN4, 6);  
}
```

Також потрібно проробити ці кроки для сервоприводу.

```
const int ServoPin = 2;  
void initServo()  
{  
    ledcSetup(8, 50, 16);  
    ledcAttachPin(ServoPin, 8);  
}
```

Та ініціалізуємо ці дві функції в функції *setup()*.

```
initMotors();  
initServo();
```

Наступним кроком переходимо в налаштування веб-інтерфейсу нашої ESP32-CAM. Для цього перейдемо в файл *app_httpd.cpp* та внесемо декілька змін в його структуру. В першу чергу перепишемо HTML код, щоб змінити стандартний інтерфейс на той що нам потрібно.

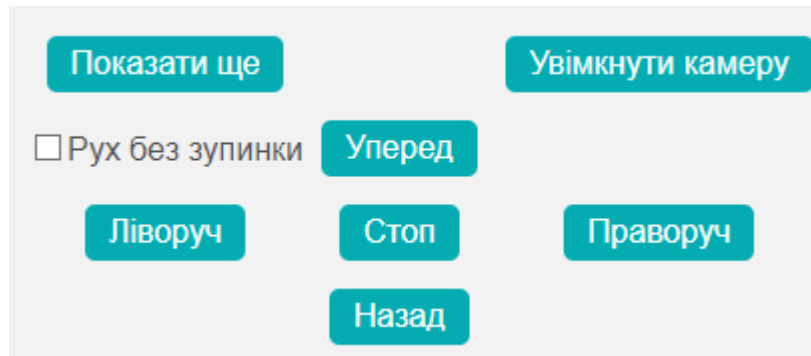


Рисунок 3.9 – Інтерфейс керування шасі

Далі створимо алгоритм керування ТЗ.

```
else if(!strcmp(variable, "car")) {
    if (val==1) {
        Serial.println("Forward");
        actstate = fwd;
        ledcWrite(4,speed);
        ledcWrite(3,0);
        ledcWrite(5,speed);
        ledcWrite(6,0);
        delay(2000);
    }
    else if (val==2) {
        Serial.println("TurnLeft");
        ledcWrite(3,0);
        ledcWrite(5,0);
        ledcWrite(4,speed);
        ledcWrite(6,speed);
        delay(2000);
        ledcWrite(4,0);
        ledcWrite(6,0);
    }
}
```

```
}  
else if (val==3) {  
    Serial.println("Stop");  
    actstate = stp;  
    ledcWrite(4,0);  
    ledcWrite(3,0);  
    ledcWrite(5,0);  
    ledcWrite(6,0);  
}  
else if (val==4) {  
    Serial.println("TurnRight");  
    ledcWrite(4,0);  
    ledcWrite(6,0);  
    ledcWrite(3,speed);  
    ledcWrite(5,speed);  
    delay(2000);  
    ledcWrite(3, 0);  
    ledcWrite(5, 0);  
}  
else if (val==5) {  
    Serial.println("Backward");  
    actstate = rev;  
    ledcWrite(4,0);  
    ledcWrite(3,speed);  
    ledcWrite(5,0);  
    ledcWrite(6,speed);  
    delay(2000);  
}  
if (noStop!=1)  
{  
    ledcWrite(3, 0);  
    ledcWrite(4, 0);  
    ledcWrite(5, 0);  
    ledcWrite(6, 0);  
}
```

```
}  
}
```

Ця частина коду перевіряє, чи змінна *variable* має значення «car». Якщо так, перевіряється значення *val*. Блок-схема алгоритму керування винесена в додаток Б.

Якщо *val* дорівнює 1, виконується рух вперед. Виводиться повідомлення «Forward», встановлюється значення *actstate* в *fwd*, і викликаються функції *ledcWrite()* для керування швидкістю руху вперед на певних пінах.

Якщо *val* дорівнює 2, виконується поворот вліво. Виводиться повідомлення «TurnLeft», викликаються функції *ledcWrite()* для керування рухом вліво.

Якщо *val* дорівнює 3, виконується зупинка. Виводиться повідомлення «Stop», і всі значення швидкості встановлюються в 0.

Якщо *val* дорівнює 4, виконується поворот вправо. Виводиться повідомлення «TurnRight», викликаються функції *ledcWrite()* для керування рухом вправо.

Якщо *val* дорівнює 5, виконується рух назад. Виводиться повідомлення «Backward», викликаються функції *ledcWrite()* для керування швидкістю руху назад на певних пінах.

Якщо *noStop* не дорівнює 1, то всі значення швидкості встановлюються в 0.

3.3 Завантаження скетчу на ESP32

Перед початком потрібно підключити UART до ESP32 так як показано на Рисунку 3.1. Обов'язково потрібно замкнути контакти GND та IO0.

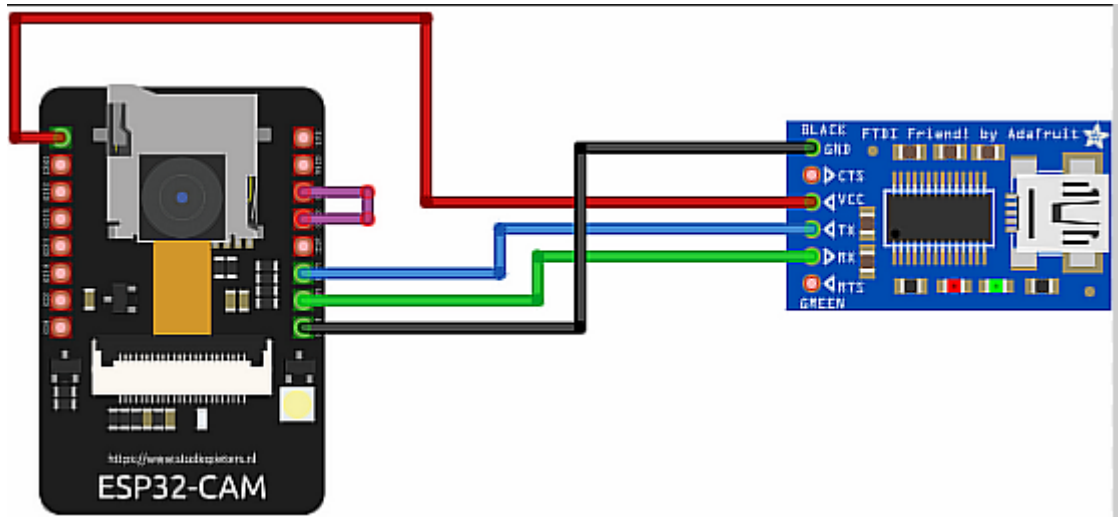


Рисунок 3.11 – Підключення UART до ESP32

В даній схемі червоний дріт це живлення 5 В, чорний GND, синій U0T та RXD, зелений U0R та TXD, фіолетовий перемичка між контактами GND та IO.

Після чого компілюємо скетч. В кінці завантаження повинно з'явитися наступне повідомлення.

```
1. Compressed 3072 bytes to 119...
2. Wrote 3072 bytes (119 compressed) at 0x00008000 in 0.0 seconds
3. Hash of data verified.
4.
5. Leaving...
6. Hard resetting via RTS pin...
```

Рисунок 3.12 – Результат завантаження скетчу.

Після цього потрібно відкрити монітор порту, вибрати 115200 baud, зняти перемичку на платі та на ній же натиснути кнопку RTS.

В моніторі порту отримаємо наступне повідомлення.

```
.....
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.0.9' to connect
```

Рисунок 3.13 – Звіт з монітору порту

Перейшовши за отриманою адресою з'явиться інтерфейс керування транспортним засобом.

Висновки до розділу 3

В розділі даному розділі розглядаються аспекти, пов'язані з апаратурою та програмним забезпеченням системи. Він охоплює вибір необхідних компонентів, налаштування інтерфейсів та програмування мікроконтролерів.

Також у розділі детально розглядається процес вибору програмних і апаратних компонентів для системи. Цей процес включає в себе опис мікроконтролера, драйверів та інших пристроїв, що задовольняють функціональні вимоги проекту.

До того ж розділ описує процес реалізації проекту. Це включає створення алгоритмів, програмування функцій і модулів, побудову блок схеми та налаштування параметрів системи та перевірку функціональності.

Було розглянуто процес завантаження скетчу (програмного коду) на мікроконтролер ESP32. Цей процес включає підключення ESP32 до комп'ютера, правильне підключення та послідовність програмування через UART адаптер.

Усі ці розділи спрямовані на досягнення належної функціональності системи, правильного вибору компонентів, їх налаштування та успішне завантаження програмного коду на мікроконтролер для досягнення проектних цілей.

ВИСНОВКИ

В ході дослідження шасі безпілотного наземного транспортного засобу високої прохідності на базі моторів коліс було проведено комплексний аналіз та розробка системи з використанням математичних методів та моделей.

Огляд існуючих рішень дозволив виявити основні переваги та недоліки таких систем як коливальна система транспортного засобу чи моторів-коліс, також визначити вимоги до апаратно-програмного забезпечення системи. Формування вимог було здійснено з урахуванням потреб високої прохідності, ефективного керування та оптимізації енергоживлення.

Результатом дослідження є розробка математичних моделей, які враховують особливості наземних транспортних засобів. Ці моделі дозволяють визначити технічні характеристики прототипу та включають моделі енергоживлення та коливань підвіски. Вони є основою для подальшої реалізації системи.

Вибір програмних і апаратних компонентів був здійснений з урахуванням вимог до системи, забезпечуючи високу надійність, швидкодію та можливості масштабування. Реалізація проекту включає в себе розробку апаратно-програмного комплексу на базі контролера ESP32, драйверу L298N, та колекторних двигунів постійного струму .

Загальний висновок полягає в тому, що проведені дослідження та розробка системи шасі БНТЗ високої прохідності на базі моторів коліс дозволили досягти поставлених цілей. Отримані результати дають підстави для використання розробленої системи у практичних умовах та можуть зробити внесок в розвиток сучасних транспортних технологій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. The World's Most Experienced Driver. URL: <https://waymo.com/> (Last accessed: 16.01.2023).
2. Tesla автопілот. URL: <https://teslaone.com.ua/news/tesla/pokoleniya-avtopilota-tesla-i-ikh-otlichiya> (дата звернення: 18.01.2023).
3. Uber запустив перші безпілотні таксі в США. URL: https://news.infocar.ua/uber_zapustil_pervye_bespilotnye_taksi_v_ssha_109842.html (дата звернення: 19.01.2023).
4. Електричні мотори-колеса. URL: <https://mybro.com.ua/news/kak-rabotaet-motor-koleso.html> (дата звернення: 7.06.2023).
5. Гібридні автомобілі. URL: <https://kk-auto.com.ua/novyny/shho-take-gibridni-avtomobili-yak-voni-vlashtovani-ta-yakimi-buvayut/> (дата звернення: 18.01.2023).
6. Vehicle Drive with Hydraulic Motors – Part 1. URL: <https://www.womackmachine.com/engineering-toolbox/data-sheets/vehicle-drive-with-hydraulic-motors-part-1/> (Last accessed: 23.01.2023).
7. Hydro-pneumatic suspension systems: faster and more cost-effective development. URL: <https://www.argo-hytos.com/news/archive/archive/hydro-pneumatic-suspension-systems-faster-and-more-cost-effective-development.html> (Last accessed: 26.01.2023).
8. Dependent suspension system. URL: <https://www.autodeal.com.ph/articles/car-features/beginners-guide-car-suspension-types-and-why-they-matter> (Last accessed: 29.01.2023).
9. Independent Suspension Explained And If It's Right For You. URL: <https://www.carandbike.com/news/independent-suspension-explained-and-if-its-right-for-you-2972628> (Last accessed: 3.02.2023).
10. The 4 Types Of Drivetrains And Why They Matter. URL: <https://www.cbac.com/waxahachie/media-center/blog/2016/september/the-4-types-of-drivetrains-and-why-they-matter/> (Last accessed: 6.02.2023).

11. Кінематичне описання руху. Матеріальна точка. URL: <https://zfftt.kpi.ua/images/Chizska/Lecture1.pdf> (дата звернення: 6.04.2023).
12. Основні поняття розділу "Динаміка". URL: <https://naukozavr.info/fizuka/formula-shvydkosti/> (дата звернення: 6.04.2023).
13. Формула швидкості — позначення, одиниці виміру, приклади. URL: http://document.kdu.edu.ua/info_zab/274_468.pdf (дата звернення: 8.04.2023).
14. Системи віддаленого управління. URL: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwiaw8Owy8r_AhUYxgIHHQL1CkoQFnoECBQQAQ&url=https%3A%2F%2Fjvestnik-sss.donnu.edu.ua%2Farticle%2Fview%2F10001%2F9928&usg=AOvVaw222-uP-aWkCIyMSDbIQX2P&opi=89978449 (дата звернення: 10.04.2023).
15. Машине навчання з вчителем. URL: <http://www.mmf.lnu.edu.ua/en/aren/1739> (дата звернення: 12.04.2023).
16. Мурашині алгоритми та інші види еволюційних алгоритмів. URL: https://learn.ztu.edu.ua/pluginfile.php/247489/mod_resource/content/1/СІШІ_Л-10_ГА2_22.pdf (дата звернення: 13.04.2023).
17. How Machine Learning Works. URL: <https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-machine-learning>. (Last accessed 13.04.2023).
18. Експертні системи: особливості застосування. Реферат. URL: <https://osvita.ua/vnz/reports/management/13574/> (дата звернення: 14.04.2023).
19. Формула розрахунку потужності однофазного та трифазного навантаження. URL: <https://energokvant.com/rozrakhunok-strumu-za-potuzhnistyuyak-pravilno-rozrakhuvati> (дата звернення: 16.04.2023)
20. Енергозбереження та способи економії електроенергії. URL: <https://org2.knuba.edu.ua/mod/book/tool/print/index.php?id=30905&chapterid=432> (дата звернення: 17.04.2023)

21. Different Types Of Batteries. URL: <https://www.electronicsforu.com/technology-trends/learn-electronics/different-types-of-batteries> (Last accessed: 19.04.2023)
22. Suspension design: definitions and effects on vehicle behavior. URL: <https://www.car-engineer.com/suspension-design-definitions-and-effects-on-vehicle-behavior/> (Last accessed: 20.04.2023)
23. Войтенко В. Математичне моделювання пружної підвіски транспортного засобу. *Automation Technological and Business Processes*. 2014. Vol. 11. DOI: 10.15673/2312-3125.11-12/2012.33452.
24. ПРОХІДНІСТЬ АВТОМОБІЛЯ. URL: <https://budtehnika.pp.ua/5921-prohdnst-avtomoblya.html> (дата звернення: 24.04.2023)
25. Модуль Wi-Fi ESP32-CAM з камерою 2MP. URL: <https://arduino.ua/prod3458-modyl-wi-fi-esp32-s-kameroi-2mp> (дата звернення: 03.05.2023)
26. Колекторний двигун з ротором 1:48. URL: <https://arduino.ua/prod3195-motor-s-reduktorom-148-odno-osevoi> (дата звернення: 03.05.2023)
27. Драйвер L298N. URL: <https://arduino.ua/prod406-draiver-dvyh-dvigateli-na-l298n> (дата звернення: 03.05.2023)

ДОДАТОК А

ДОВІДКА

про перевірку на унікальність пояснювальної записки
кваліфікаційної бакалаврської роботи
на тему: «Шасі безпілотного наземного транспортного засобу
високої прохідності на базі моторів-коліс»
студента спеціальності 123 «Комп'ютерна інженерія»,
405 групи
Лосіцького Павла Михайловича
прізвище, ім'я, по-батькові

Перевірку тексту здійснено сервісом: онлайн-сервіс Unicheck

Результат перевірки тексту бакалаврської кваліфікаційної роботи: схожість складає 1%.



User name:
Іван Бурлаченко

Check ID:
1015633772

Check date:
17.06.2023 18:17:56 EEST

Check type:
Doc vs Internet + Library

Report date:
17.06.2023 18:23:25 EEST

User ID:
100000130

File name: Лосіцький П. М. Комп'ютерна інженерія КБР 2023

Page count: 18 Word count: 8969 Character count: 69261 File size: 776.95 KB File ID: 1015280250

1% Matches

Highest match: 0.31% with Internet source (<http://elar.khmnu.edu.ua/jspui/bitstream/123456789/12253/1/%d0%9c%d0%b0%d0%>)

1% Internet sources 72

Page 20

0.31% Library sources 9

Page 20

0% Quotes


Exclusion of quotes is off

Exclusion of references is off

0% Exclusions

No exclusions

Здобувач:


підпис П. М. Лосіцький
ініціали, прізвище

Дата: «__» _____ 2023 р

Керівник:

ст. викладач

_____ І. С. Бурлаченко
підпис ініціали, прізвище

ДОДАТОК Б

БЛОК-СХЕМА

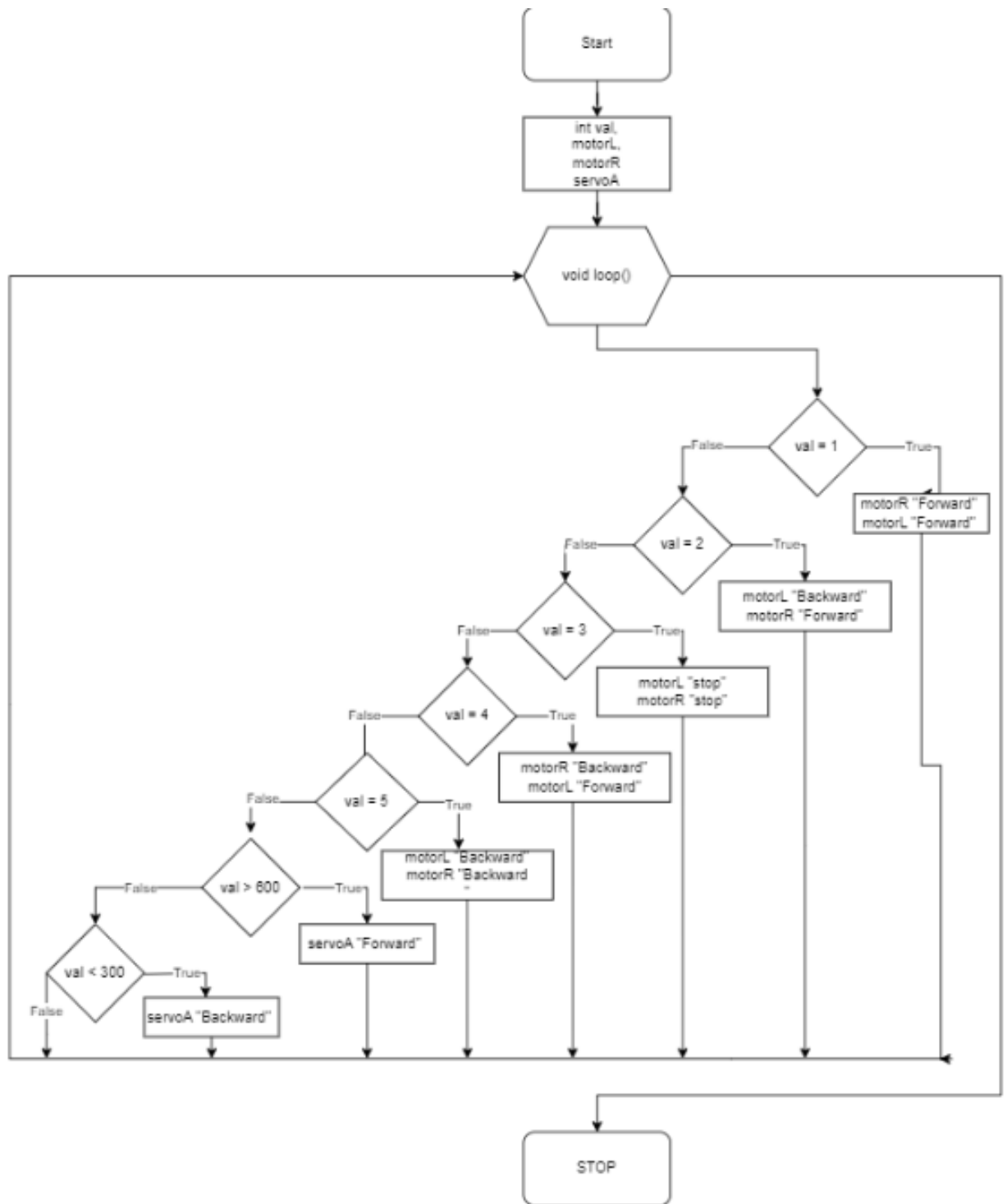


Рисунок Б.1 – Блок-схема алгоритму керування

ДОДАТОК В

СКЕТЧ 1

```
const char* ssid = "TOTOLINK_N300RT";  
const char* password = "*****";
```

```
#include "esp_wifi.h"  
#include "esp_camera.h"  
#include <WiFi.h>  
#include "soc/soc.h"  
#include "soc/rtc_cntl_reg.h"
```

```
#define CAMERA_MODEL_AI_THINKER
```

```
#elif defined(CAMERA_MODEL_AI_THINKER)
```

```
#define PWDN_GPIO_NUM    32  
#define RESET_GPIO_NUM  -1  
#define XCLK_GPIO_NUM    0  
#define SIOD_GPIO_NUM    26  
#define SIOC_GPIO_NUM    27  
#define Y9_GPIO_NUM      35  
#define Y8_GPIO_NUM      34  
#define Y7_GPIO_NUM      39  
#define Y6_GPIO_NUM      36  
#define Y5_GPIO_NUM      21  
#define Y4_GPIO_NUM      19  
#define Y3_GPIO_NUM      18  
#define Y2_GPIO_NUM      5  
#define VSYNC_GPIO_NUM   25  
#define HREF_GPIO_NUM    23  
#define PCLK_GPIO_NUM    22
```

```
#else
```

```
#error "Camera model not selected"
```

```
#endif
```

```
void startCameraServer();
```

```
const int IN1 = 12;
```

```
const int IN2 = 13;
```

```
const int IN3 = 14;
```

```
const int IN4 = 15;
```

```
void initMotors()
{
  ledcSetup(3, 2000, 8);
  ledcSetup(4, 2000, 8);
  ledcSetup(5, 2000, 8);
  ledcSetup(6, 2000, 8);
  ledcAttachPin(IN1, 3);
  ledcAttachPin(IN2, 4);
  ledcAttachPin(IN3, 5);
  ledcAttachPin(IN4, 6);
}

const int ServoPin = 2;
void initServo()

{
  ledcSetup(8, 50, 16);
  ledcAttachPin(ServoPin, 8);
}

void setup()
{
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); // prevent brownouts by silencing them
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
```



```
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;

//init with high specs to pre-allocate larger buffers
if(psramFound()){
  config.frame_size = FRAMESIZE_QVGA;
  config.jpeg_quality = 10;
  config.fb_count = 2;
} else {
  config.frame_size = FRAMESIZE_QVGA;
  config.jpeg_quality = 12;
  config.fb_count = 1;
}
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
  Serial.printf("Camera init failed with error 0x%x", err);
  return;
}
sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_QVGA);
s->set_vflip(s, 1);
s->set_hmirror(s, 1);
initMotors();
initServo();
ledcSetup(7, 5000, 8);
ledcAttachPin(4, 7);
Serial.println("ssid: " + (String)ssid);
Serial.println("password: " + (String)password);
WiFi.begin(ssid, password);
delay(500);
long int StartTime=millis();
while (WiFi.status() != WL_CONNECTED)
{
  delay(500);
  if ((StartTime+10000) < millis()) break;
}
```

```
startCameraServer();
if (WiFi.status() == WL_CONNECTED)
{
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
} else {
  Serial.println("");
  Serial.println("WiFi disconnected");
  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.softAPIP());
  Serial.println("' to connect");
  char* apssid = "ESP32-CAM";
  char* appassword = "12345678";
  WiFi.softAP((WiFi.softAPIP().toString()+"_"+(String)apssid).c_str(), appassword);
}
for (int i=0;i<5;i++)
{
  ledcWrite(7,10);
  delay(50);
  ledcWrite(7,0);
  delay(50);
}
}
void loop() {
  delay(1000);
  Serial.printf("RSSI: %ld dBm\n",WiFi.RSSI());
}
```

ДОДАТОК Г

СКЕТЧ 2

```
#include "dl_lib_matrix3d.h"
#include <esp32-hal-ledc.h>

int speed = 255;
int noStop = 0;

#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "img_converters.h"
#include "Arduino.h"

typedef struct {
    httpd_req_t *req;
    size_t len;
} jpg_chunking_t;

#define PART_BOUNDARY "1234567890000000000000987654321"
static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary="
PART_BOUNDARY;
static const char* _STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";
static const char* _STREAM_PART = "Content-Type: image/jpeg\r\nContent-Length:
%u\r\n\r\n";
httpd_handle_t stream_httpd = NULL;
httpd_handle_t camera_httpd = NULL;
static size_t jpg_encode_stream(void * arg, size_t index, const void* data, size_t
len){
    jpg_chunking_t *j = (jpg_chunking_t *)arg;
    if(!index){
        j->len = 0;
    }
    if(httpd_resp_send_chunk(j->req, (const char *)data, len) != ESP_OK){
        return 0;
    }
    j->len += len;
    return len;
}
```

```
static esp_err_t capture_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    int64_t fr_start = esp_timer_get_time();
    fb = esp_camera_fb_get();
    if (!fb) {
        Serial.println("Camera capture failed");
        httpd_resp_send_500(req);
        return ESP_FAIL;
    }

    httpd_resp_set_type(req, "image/jpeg");
    httpd_resp_set_hdr(req, "Content-Disposition", "inline;
filename=capture.jpg");
    size_t out_len, out_width, out_height;
    uint8_t * out_buf;
    bool s;
    {
        size_t fb_len = 0;
        if(fb->format == PIXFORMAT_JPEG){
            fb_len = fb->len;
            res = httpd_resp_send(req, (const char *)fb->buf, fb->len);
        } else {
            jpg_chunking_t jchunk = {req, 0};
            res = frame2jpg_cb(fb, 80, jpg_encode_stream,
&jchunk)?ESP_OK:ESP_FAIL;
            httpd_resp_send_chunk(req, NULL, 0);
            fb_len = jchunk.len;
        }
        esp_camera_fb_return(fb);
        int64_t fr_end = esp_timer_get_time();
        Serial.printf("JPG: %uB %ums\n", (uint32_t)(fb_len), (uint32_t)((fr_end -
fr_start)/1000));
        return res;
    }

    dl_matrix3du_t *image_matrix = dl_matrix3du_alloc(1, fb->width, fb->height,
3);

    if (!image_matrix) {
        esp_camera_fb_return(fb);
    }
}
```

```
Serial.println("dl_matrix3du_alloc failed");
httpd_resp_send_500(req);
return ESP_FAIL;
}
out_buf = image_matrix->item;
out_len = fb->width * fb->height * 3;
out_width = fb->width;
out_height = fb->height;
s = fmt2rgb888(fb->buf, fb->len, fb->format, out_buf);
esp_camera_fb_return(fb);
if(!s){
    dl_matrix3du_free(image_matrix);
    Serial.println("to rgb888 failed");
    httpd_resp_send_500(req);
    return ESP_FAIL;
}
jpg_chunking_t jchunk = {req, 0};
s = fmt2jpg_cb(out_buf, out_len, out_width, out_height, PIXFORMAT_RGB888, 90,
jpg_encode_stream, &jchunk);
dl_matrix3du_free(image_matrix);
if(!s){
    Serial.println("JPEG compression failed");
    return ESP_FAIL;
}
int64_t fr_end = esp_timer_get_time();
return res;
}

static esp_err_t stream_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    size_t _jpg_buf_len = 0;
    uint8_t * _jpg_buf = NULL;
    char * part_buf[64];
    dl_matrix3du_t *image_matrix = NULL;
    static int64_t last_frame = 0;
    if(!last_frame) {
        last_frame = esp_timer_get_time();
    }
    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
    if(res != ESP_OK){
```

```
    return res;
}
while(true){
    fb = esp_camera_fb_get();
    if (!fb) {
        Serial.println("Camera capture failed");
        res = ESP_FAIL;
    } else {
        {
            if(fb->format != PIXFORMAT_JPEG){
                bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf,
&_jpg_buf_len);

                esp_camera_fb_return(fb);
                fb = NULL;
                if(!jpeg_converted){
                    Serial.println("JPEG compression failed");
                    res = ESP_FAIL;
                }
            } else {
                _jpg_buf_len = fb->len;
                _jpg_buf = fb->buf;
            }
        }
    }
    if(res == ESP_OK){
        size_t hlen = sprintf((char *)part_buf, 64, _STREAM_PART,
_jpg_buf_len);
        res = httpd_resp_send_chunk(req, (const char *)part_buf, hlen);
    }
    if(res == ESP_OK){
        res = httpd_resp_send_chunk(req, (const char *)_jpg_buf,
_jpg_buf_len);
    }
    if(res == ESP_OK){
        res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY,
strlen(_STREAM_BOUNDARY));
    }
    if(fb){
        esp_camera_fb_return(fb);
        fb = NULL;
        _jpg_buf = NULL;
    }
}
```

```
} else if(_jpg_buf){
    free(_jpg_buf);
    _jpg_buf = NULL;
}
if(res != ESP_OK){
    break;
}
int64_t fr_end = esp_timer_get_time();
int64_t frame_time = fr_end - last_frame;
last_frame = fr_end;
frame_time /= 1000;
Serial.printf("MJPG: %uB %ums (%.1ffps)\n",
    (uint32_t)(_jpg_buf_len),
    (uint32_t)frame_time, 1000.0 / (uint32_t)frame_time
);
}

last_frame = 0;
return res;
}
enum state {fwd,rev,stp};
state actstate = stp;
static esp_err_t cmd_handler(httpd_req_t *req)
{
    char* buf;
    size_t buf_len;
    char variable[32] = {0,};
    char value[32] = {0,};
    buf_len = httpd_req_get_url_query_len(req) + 1;
    if (buf_len > 1) {
        buf = (char*)malloc(buf_len);
        if(!buf){
            httpd_resp_send_500(req);
            return ESP_FAIL;
        }
        if (httpd_req_get_url_query_str(req, buf, buf_len) == ESP_OK) {
            if (httpd_query_key_value(buf, "var", variable, sizeof(variable)) ==
ESP_OK &&
                httpd_query_key_value(buf, "val", value, sizeof(value)) == ESP_OK)
            {
                } else {
```

Шасі безпілотного наземного транспортного засобу
високої прохідності на базі моторів-коліс

```
        free(buf);
        httpd_resp_send_404(req);
        return ESP_FAIL;
    }
} else {
    free(buf);
    httpd_resp_send_404(req);
    return ESP_FAIL;
}
free(buf);
} else {
    httpd_resp_send_404(req);
    return ESP_FAIL;
}
int val = atoi(value);
sensor_t * s = esp_camera_sensor_get();
int res = 0;

if(!strcmp(variable, "framesize"))
{
    Serial.println("framesize");
    if(s->pixformat == PIXFORMAT_JPEG) res = s->set_framesize(s,
(framesize_t)val);
}
else if(!strcmp(variable, "quality"))
{
    Serial.println("quality");
    res = s->set_quality(s, val);
}
else if(!strcmp(variable, "flash"))
{
    ledcWrite(7, val);
}
else if(!strcmp(variable, "speed"))
{
    if (val > 255) val = 255;
    else if (val < 0) val = 0;
    speed = val;
}
else if(!strcmp(variable, "nostop"))
{
```



```
noStop = val;
}
else if(!strcmp(variable, "servo"))
{
    if      (val > 650) val = 650;
    else if (val < 225) val = 325;
    ledcWrite(8,10*val);
}
else if(!strcmp(variable, "car")) {
    if (val==1) {
        Serial.println("Forward");
        actstate = fwd;
        ledcWrite(4,speed);
        ledcWrite(3,0);
        ledcWrite(5,speed);
        ledcWrite(6,0);
        delay(2000);
    }
    else if (val==2) {
        Serial.println("TurnLeft");
        ledcWrite(3,0);
        ledcWrite(5,0);
        ledcWrite(4,speed);
        ledcWrite(6,speed);
        delay(2000);
        ledcWrite(4,0);
        ledcWrite(6,0);
    }
    else if (val==3) {
        Serial.println("Stop");
        actstate = stp;
        ledcWrite(4,0);
        ledcWrite(3,0);
        ledcWrite(5,0);
        ledcWrite(6,0);
    }
    else if (val==4) {
        Serial.println("TurnRight");
        ledcWrite(4,0);
        ledcWrite(6,0);
        ledcWrite(3,speed);
    }
}
```

```
    ledcWrite(5, speed);
    delay(2000);
    ledcWrite(3, 0);
    ledcWrite(5, 0);

}
else if (val==5) {
    Serial.println("Backward");
    actstate = rev;
    ledcWrite(4, 0);
    ledcWrite(3, speed);
    ledcWrite(5, 0);
    ledcWrite(6, speed);
    delay(2000);
}
if (noStop!=1)
{
    ledcWrite(3, 0);
    ledcWrite(4, 0);
    ledcWrite(5, 0);
    ledcWrite(6, 0);
}
}
else
{
    Serial.println("variable");
    res = -1;
}
if(res){ return httpd_resp_send_500(req); }
httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
return httpd_resp_send(req, NULL, 0);
}

static esp_err_t status_handler(httpd_req_t *req){
    static char json_response[1024];
    sensor_t * s = esp_camera_sensor_get();
    char * p = json_response;
    *p++ = '{';
    p+=sprintf(p, "\"framesize\":%u,", s->status.framesize);
    p+=sprintf(p, "\"quality\":%u,", s->status.quality);
    *p++ = '}';
    *p++ = 0;
}
```

```
httpd_resp_set_type(req, "application/json");
httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "");
return httpd_resp_send(req, json_response, strlen(json_response));
}
static const char PROGMEM INDEX_HTML[] = R"rawliteral(
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <title>ESP32 OV2460</title>
    <style>
      body{font-family:Arial,Helvetica,sans-
serif;background:#f2f2f2;color:#4F4E4E;font-size:16px}h2{font-
size:18px}section.main{display:flex}#menu,section.main{flex-
direction:column}#menu{display:none;flex-wrap:nowrap;min-
width:340px;background:#f2f2f2;padding:8px;border-radius:4px;margin-top:-10px;margin-
right:10px}#content{display:flex;flex-wrap:wrap;align-
items:stretch}figure{padding:0;margin:0;-webkit-margin-before:0;margin-block-start:0;-
webkit-margin-after:0;margin-block-end:0;-webkit-margin-start:0;margin-inline-start:0;-
webkit-margin-end:0;margin-inline-end:0}figure
img{display:block;width:100%;height:auto;border-radius:4px;margin-top:8px}@media (min-
width: 800px) and (orientation:landscape){#content{display:flex;flex-wrap:nowrap;align-
items:stretch}figure      img{display:block;max-width:100%;max-height:calc(100vh
-
40px);width:auto;height:auto}figure{padding:0;margin:0;-webkit-margin-before:0;margin-
block-start:0;-webkit-margin-after:0;margin-block-end:0;-webkit-margin-start:0;margin-
inline-start:0;-webkit-margin-end:0;margin-inline-
end:0}}section#buttons{display:flex;flex-wrap:nowrap;justify-content:space-
between}#nav-toggle{cursor:pointer;display:block}#nav-toggle-
cb{outline:0;opacity:0;width:0;height:0}#nav-toggle-
cb:checked+#menu{display:flex}.input-group{display:flex;flex-wrap:nowrap;line-
height:22px;margin:5px      0}.input-group>label{display:inline-block;padding-
right:10px;min-width:47%}.input-group  input,.input-group  select{flex-grow:1}.range-
max,.range-min{display:inline-block;padding:0
5px}button{display:block;margin:5px;padding:0      12px;border:0;line-
height:28px;cursor:pointer;color:#fff;background:#04ACB1;border-radius:5px;font-
size:16px;outline:0}button:hover{background:#1abc9c}button:active{background:#1abc9c}bu
tton.disabled{cursor:default;background:#a0a0a0}input[type=range]{-webkit-
appearance:none;width:100%;height:22px;background:#ffffff;cursor:pointer;margin:0}input
[type=range]:focus{outline:0}input[type=range]:::-webkit-slider-runnable-
track{width:100%;height:2px;cursor:pointer;background:#EFEFEF;border-radius:0;border:0
```

```
solid #EFEFEF}input[type=range]::-webkit-slider-thumb{border:1px solid
rgba(0,0,30,0);height:22px;width:22px;border-
radius:50px;background:#04ACB1;cursor:pointer;-webkit-appearance:none;margin-top:-
11.5px}input[type=range]:focus::-webkit-slider-runnable-
track{background:#EFEFEF}input[type=range]::-moz-range-
track{width:100%;height:2px;cursor:pointer;background:#EFEFEF;border-radius:0;border:0
solid #EFEFEF}input[type=range]::-moz-range-thumb{border:1px solid
rgba(0,0,30,0);height:22px;width:22px;border-
radius:50px;background:#04ACB1;cursor:pointer}input[type=range]::-ms-
track{width:100%;height:2px;cursor:pointer;background:0 0;border-
color:transparent;color:transparent}input[type=range]::-ms-fill-
lower{background:#EFEFEF;border:0 solid #EFEFEF;border-radius:0}input[type=range]::-ms-
fill-upper{background:#EFEFEF;border:0 solid #EFEFEF;border-
radius:0}input[type=range]::-ms-thumb{border:1px solid
rgba(0,0,30,0);height:22px;width:22px;border-
radius:50px;background:#04ACB1;cursor:pointer;height:2px}input[type=range]:focus::-ms-
fill-lower{background:#EFEFEF}input[type=range]:focus::-ms-fill-
upper{background:#363636}.switch{display:block;position:relative;line-height:22px;font-
size:16px;height:22px}.switch
input{outline:0;opacity:0;width:0;height:0}.slider{width:50px;height:22px;border-
radius:22px;cursor:pointer;background-color:grey}.slider,.slider:before{display:inline-
block;transition:.4s}.slider:before{position:relative;content:"";border-
radius:50%;height:16px;width:16px;left:4px;top:3px;background-
color:#fff}input:checked+.slider{background-
color:#04ACB1}input:checked+.slider:before{-webkit-
transform:translateX(26px);transform:translateX(26px)}select{border:1px solid
#363636;font-size:14px;height:22px;outline:0;border-radius:5px}.image-
container{position:relative;min-
width:160px}.close{position:absolute;right:5px;top:5px;background:#04ACB1;width:16px;he-
ight:16px;border-radius:100px;color:#fff;text-align:center;line-
height:18px;cursor:pointer}.hidden{display:none}
</style>
</head>
<body>
<figure>
<div id="stream-container" class="image-container hidden">
<div class="close" id="close-stream"></div>
<img id="stream" src="">
</div>
</figure>
<section class="main">
```

```
<section id="buttons">
  <table>
    <tr><td align="center"><button id="get-still">Показати
ще</button></td><td align="center"><button id="toggle-stream">Увімкнути
камеру</button></td></tr>
    <tr><td><input type="checkbox" id="nostop" onclick="var
noStop=0;if
noStop=1;fetch(document.location.origin+'/control?var=nostop&val='+noStop);">Рух без
зупинки</td><td align="center"><button id="forward"
onclick="fetch(document.location.origin+'/control?var=car&val=1');">Уперед</button></td>
<td></td></tr>
    <tr><td align="center"><button id="turnleft"
onclick="fetch(document.location.origin+'/control?var=car&val=2');">Ліворуч</button></t
d><td align="center"><button id="stop"
onclick="fetch(document.location.origin+'/control?var=car&val=3');">Стоп</button></td><
td align="center"><button id="turnright"
onclick="fetch(document.location.origin+'/control?var=car&val=4');">Праворуч</button></
td></tr>
    <tr><td align="center"><button id="backward"
onclick="fetch(document.location.origin+'/control?var=car&val=5');">Назад</button></td>
<td></td></tr>
    <tr><td>Світлодіод</td><td align="center" colspan="2"><input
type="range" id="flash" min="0" max="255" value="0"
onchange="try{fetch(document.location.origin+'/control?var=flash&val='+this.value);}cat
ch(e)}"></td></tr>
    <tr><td>Швидкість</td><td align="center" colspan="2"><input
type="range" id="speed" min="0" max="255" value="255"
onchange="try{fetch(document.location.origin+'/control?var=speed&val='+this.value);}cat
ch(e)}"></td></tr>
    <tr><td>Поворот коліс</td><td align="center" colspan="2"><input
type="range" id="servo" min="325" max="650" value="487"
onchange="try{fetch(document.location.origin+'/control?var=servo&val='+this.value);}cat
ch(e)}"></td></tr>
    <tr><td>Якість</td><td align="center" colspan="2"><input
type="range" id="quality" min="10" max="63" value="10"
onchange="try{fetch(document.location.origin+'/control?var=quality&val='+this.value);}c
atch(e)}"></td></tr>
    <tr><td>Розширення</td><td align="center" colspan="2"><input
type="range" id="framesize" min="0" max="6" value="5"
onchange="try{fetch(document.location.origin+'/control?var=framesize&val='+this.value);
}catch(e)}"></td></tr>
```

```
</table>
</section>
</section>
<script>
    document.addEventListener('DOMContentLoaded',function(){function
b(B){let C;switch(B.type){case'checkbox':C=B.checked?1:0;break;case'range':case'select-
one':C=B.value;break;case'button':case'submit':C='1';break;default:return;}const
D=`${C}/control?var=${B.id}&val=${C}`;fetch(D).then(E=>{console.log(`request to ${D}
finished, status: ${E.status}`)})}var c=document.location.origin;const
e=B=>{B.classList.add('hidden')},f=B=>{B.classList.remove('hidden')},g=B=>{B.classList.
add('disabled'),B.disabled=!0},h=B=>{B.classList.remove('disabled'),B.disabled=!1},i=(B
,C,D)=>{D!=(null!=D)||D;let
E;'checkbox'===B.type?(E=B.checked,C=!!C,B.checked=C):(E=B.value,B.value=C),D&&E!==C?b(
B):!D&&('aec'===B.id?C?e(v):f(v):'agc'===B.id?C?(f(t),e(s)):(e(t),f(s)):'awb_gain'===B.
id?C?f(x):e(x):'face_recognize'===B.id&&(C?h(n):g(n))});document.querySelectorAll('.clo
se').forEach(B=>{B.onclick=(()=>{e(B.parentNode)}),fetch(`${C}/status`).then(function(B
){return B.json()}).then(function(B){document.querySelectorAll('.default-
action').forEach(C=>{i(C,B[C.id],!1)}))});const
j=document.getElementById('stream'),k=document.getElementById('stream-
container'),l=document.getElementById('get-still'),m=document.getElementById('toggle-
stream'),n=document.getElementById('face_enroll'),o=document.getElementById('close-
stream'),p(())=>{window.stop(),m.innerHTML='
    Увімкнути камеру'},q(())=>{j.src=`${C}+:81'/stream`,f(k),m.innerHTML='Вимкнути
камеру'};l.onclick=(()=>{p(),j.src=`${C}/capture?_cb=${Date.now()}`,f(k)},o.onclick=(()=>
{p(),e(k)},m.onclick=(()=>{const B='Stop
Stream'===m.innerHTML;B?p():q()},n.onclick=(()=>{b(n)},document.querySelectorAll('.defau
lt-action').forEach(B=>{B.onChange=(()=>b(B)}));const
r=document.getElementById('agc'),s=document.getElementById('agc_gain-
group'),t=document.getElementById('gainceiling-
group');r.onChange=(()=>{b(r),r.checked?(f(t),e(s)):(e(t),f(s))});const
u=document.getElementById('aec'),v=document.getElementById('aec_value-
group');u.onChange=(()=>{b(u),u.checked?e(v):f(v)});const
w=document.getElementById('awb_gain'),x=document.getElementById('wb_mode-
group');w.onChange=(()=>{b(w),w.checked?f(x):e(x)});const
y=document.getElementById('face_detect'),z=document.getElementById('face_recognize'),A=
document.getElementById('framesize');A.onChange=(()=>{b(A),5<A.value&&(i(y,!1),i(z,!1))}
,y.onChange=(()=>{return 5<A.value?(alert('Please select CIF or lower resolution before
enabling this feature!'),void
i(y,!1)):void(b(y),!y.checked&&(g(n),i(z,!1)))},z.onChange=(()=>{return
5<A.value?(alert('Please select CIF or lower resolution before enabling this
feature!'),void i(z,!1)):void(b(z),z.checked?(h(n),i(y,!0)):g(n))}));
```

```
</script>
</body>
</html>
)rawliteral";

static esp_err_t index_handler(httpd_req_t *req){
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, (const char *)INDEX_HTML, strlen(INDEX_HTML));
}

void startCameraServer()
{
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    httpd_uri_t index_uri = {
        .uri      = "/",
        .method   = HTTP_GET,
        .handler  = index_handler,
        .user_ctx = NULL
    };
    httpd_uri_t status_uri = {
        .uri      = "/status",
        .method   = HTTP_GET,
        .handler  = status_handler,
        .user_ctx = NULL
    };
    httpd_uri_t cmd_uri = {
        .uri      = "/control",
        .method   = HTTP_GET,
        .handler  = cmd_handler,
        .user_ctx = NULL
    };
    httpd_uri_t capture_uri = {
        .uri      = "/capture",
        .method   = HTTP_GET,
        .handler  = capture_handler,
        .user_ctx = NULL
    };
    httpd_uri_t stream_uri = {
        .uri      = "/stream",
        .method   = HTTP_GET,
        .handler  = stream_handler,
        .user_ctx = NULL
    };
}
```

```
};  
Serial.printf("Starting web server on port: '%d'\n", config.server_port);  
if (httpd_start(&camera_httpd, &config) == ESP_OK) {  
    httpd_register_uri_handler(camera_httpd, &index_uri);  
    httpd_register_uri_handler(camera_httpd, &cmd_uri);  
    httpd_register_uri_handler(camera_httpd, &status_uri);  
    httpd_register_uri_handler(camera_httpd, &capture_uri);  
}  
config.server_port += 1;  
config.ctrl_port += 1;  
Serial.printf("Starting stream server on port: '%d'\n", config.server_port);  
if (httpd_start(&stream_httpd, &config) == ESP_OK) {  
    httpd_register_uri_handler(stream_httpd, &stream_uri);  
}  
}
```