

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

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

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

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Комп'ютеризована система орієнтування і паркування безпілотних транспортних засобів

Виконав(ла): студент(ка) IV курсу, групи СІс-42
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

Якобчук Р.А.

(прізвище та ініціали)

Керівник

(підпис)

Лещишин Ю.З.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Луцик Н.С.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

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

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

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« ____ » _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

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

студенту Якобчуку Роману Анатолійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система орієнтування і паркування безпілотних транспортних засобів

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

Затверджені наказом ректора від « 26 » 04 2024 року № 4/7-468

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи Технічне завдання

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

1. Аналіз технічного завдання

2. Проектна частина

3. Практична частина

4. Безпека життєдіяльності, основи охорони праці

Висновки

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

6. Консультанти розділів роботи

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

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Розробка і затвердження технічного завдання</i>		
2	<i>Аналіз технічного завдання</i>		
3	<i>Аналіз вимог та принципів організації комп'ютеризованої системи орієнтування і паркування безпілотних транспортних засобів</i>		
4	<i>Проектування комп'ютеризованої системи орієнтування і паркування безпілотних транспортних засобів</i>		
5	<i>Розробка схем і програмного забезпечення комп'ютеризованої системи орієнтування і паркування безпілотних транспортних засобів</i>		
6	<i>Розробка інструкцій з використання системи</i>		
7	<i>Безпека життєдіяльності, основи охорони праці</i>		
8	<i>Оформлення кваліфікаційної роботи</i>		
9	<i>Попередній захист кваліфікаційної роботи</i>		
10	<i>Захист кваліфікаційної роботи</i>		

Студент

_____ (підпис)

Якобчук Р.А.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Лецишин Ю.З.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Комп'ютеризована система орієнтування і паркування безпілотних транспортних засобів // Кваліфікаційна робота бакалавра // Якобчук Роман Анатолійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних систем та мереж, група СІс-42 // Тернопіль, 2024 // с. – 75, рис. – 16, додат. – 3, бібліогр. – 20.

Ключові слова: комп'ютеризована система орієнтування і паркування, stm32f407vgt6, ультразвуковий датчик відстані hc-sr04, lcd дисплей, rpxhawk, mavlink.

В кваліфікаційній роботі бакалавра розроблено комп'ютеризовану систему орієнтування і паркування безпілотних транспортних засобів на базі мікроконтролера STM32F407VGT6.

Кваліфікаційна робота складається з чотирьох розділів.

У першому розділі проводиться аналіз технічного завдання, а саме вимог до комп'ютеризованої системи та аналіз можливих рішень.

В другому розділі описується процес проєктування та реалізації проєкту, як вбудованої системи.

В третьому розділі приводиться розробка програмного забезпечення для функціонування пристрою. Також проводиться тестування комп'ютеризованої системи.

Четвертий розділ описує безпеку життєдіяльності та основи охорони праці.

ANNOTATION

A computerized system for orientation and parking of unmanned vehicles // Bachelor's thesis // Yakobchuk Roman Anatoliiovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information System and Software Engineering, Department of Computer Systems and Networks, group CIs-42 // Ternopil, 2024 // p. - 75, fig. - 16, bibliog. - 20.

Key words: computerised orientation system and car park, stm32f407vgt6, ultrasonic distance sensor hc-sr04, lcd display, pixhawk, mavlink.

In the bachelor's thesis, a computerised system for orientation and parking of unmanned vehicles based on the STM32F407VGT6 microcontroller was developed.

The qualification work consists of four sections.

The first section analyses the terms of reference, namely the requirements for a computerised system and analyses possible solutions.

The second section describes the process of designing and implementing the project as an embedded system.

The third section describes the development of software for the device. It also includes testing of the computerised system.

The fourth section describes life safety and the basics of occupational health and safety.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	10
1.1 Огляд аналогів та застосування системи	10
1.2 Аналіз вимог до комп'ютерної системи	11
1.2.1 Призначення системи та функціональні вимоги	11
1.2.2 Апаратні та програмні вимоги.....	12
1.2.3 Інтерфейси системи та інтеграція з існуючими системами	14
1.2.4 Вимоги до продуктивності, енергоспоживання, надійності і безпеки .	15
РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА	17
2.1 Розробка узагальненої структури комп'ютерної системи	17
2.2 Обґрунтування вибору апаратного забезпечення проектованої КС	19
2.2.1 Вибір датчиків відстані, технічні характеристики і переваги.....	21
2.2.2 Вибір дисплея, технічні характеристики і порівняння варіантів.....	22
2.2.3 Вибір блоку керування, технічні характеристики і обґрунтування.....	22
2.3 Розробка схеми з'єднань та електричної принципової схеми комп'ютеризованої системи орієнтування і паркування БТЗ.....	24
2.3.1 Розробка схеми з'єднань.....	24
2.3.2 Розробка електричної принципової схеми	26
2.4 Розробка програмного забезпечення.....	27
2.4.1 Програмна архітектура системи.....	27
2.4.2 Розробка ПЗ для збору і обробки даних з датчиків.....	28
2.4.3 Розробка ПЗ для відображення даних на дисплеї	28
2.4.4 Розробка ПЗ для передачі даних на блок керування за допомогою протоколу Mavlink	29

					КС КРБ 123.329.00.00 ПЗ				
					Зміст				
Змн.	Арк.	№ докум.	Підпис	Дата		Літ.	Арк..	Акрушів	
Розроб.		Якобчук Р.А.					5		
Перевір.		Лецишин Ю.З.							
Рецензент									
Н. контр.		Луцик Н.С.							
Зав. каф.		Осухівська Г.М.							ТНТУ, каф. КС, гр. СІс-42

2.5 Алгоритми прийняття рішень на основі даних з датчиків.....	30
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА	34
3.1 Розробка випробувального стенда	34
3.1.1 Ініціалізація системи.....	34
3.1.2 Налаштування периферійних модулів.....	35
3.1.3 Принцип передачі даних за допомогою протоколу Mavlink.....	38
3.2 Тестування алгоритмів комп'ютеризованої системи орієнтування і паркування для БТЗ.....	41
3.2.1 Розробка тестових сценаріїв для апаратних компонентів	41
3.2.2 Розробка тестових сценаріїв для програмного забезпечення.....	43
3.2.3 Тестування роботи системи в умовах реальних задач	43
3.2.4 Аналіз результатів тестування.....	44
3.3 Практична імплементація розроблюваної системи в БТЗ	45
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	47
4.1 Питання безпеки при експлуатації електроустановок високих, надвисоких та ультрависоких частот	47
4.2 Заходи, що покращують умови праці оператора	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	54
Додаток А.....	58
Додаток Б.....	62
Додаток В.....	64

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

ПЕРЕЛІК СКОРОЧЕНЬ

КС – комп’ютеризована система

ПЗ – програмне забезпечення

I2C – Inter-Integrated Circuit

БПТЗ – безпілотні транспортні засоби

БТЗ – безпілотний транспортний засіб

LCD – Liquid-Crystal Display

GND – ground

SCL – Serial Clock Line

SDA – Serial Data Line

GPIO – General-Purpose Input/Output

UART - Universal Asynchronous Receiver-Transmitter

ЕМП – електромагнітне поле

НЧ – низькі частоти

СЧ – середні частоти

ВЧ – високі частоти

ДВЧ – дуже високі частоти

УВЧ – ультрависокі частоти

НВЧ – надвисокі частоти

НЗВЧ – надзвичайно високі частоти

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

У сучасному світі технологічний прогрес безперервно змінює всі сфери нашого життя, і транспорт не є винятком. Безпілотні транспортні засоби (БТЗ), які ще нещодавно здавалися елементом наукової фантастики, сьогодні стали реальністю. Ці транспортні засоби відкривають нові горизонти для безпеки, ефективності та комфорту перевезень. Проте, для їх повноцінного функціонування необхідні складні системи орієнтування і паркування, здатні забезпечити автономність та надійність у різних умовах.

Системи орієнтування і паркування, що використовуються у безпілотних транспортних засобах, повинні відповідати високим вимогам щодо точності, швидкодії та надійності. Основними компонентами таких систем є датчики, контролери і алгоритми обробки даних. Ультразвукові датчики відстані, зокрема HC-SR04, набули великої популярності завдяки своїй простоті та ефективності. Ці датчики забезпечують точне вимірювання відстані до перешкод, що є критично важливим для безпечного орієнтування та паркування.

Мікроконтролери STM32, зокрема модель STM32F407VGT6, широко використовуються у вбудованих системах завдяки своїм високим обчислювальним можливостям, енергоефективності та наявності різноманітних периферійних інтерфейсів. Вони дозволяють реалізовувати складні алгоритми обробки даних з датчиків та керування виконавчими механізмами.

Однією з ключових складових сучасних безпілотних транспортних засобів є блок керування, такий як Pixhawk, який забезпечує збір, обробку та обмін даними між різними компонентами системи. Використання протоколу Mavlink дозволяє інтегрувати різноманітні сенсори та контролери, створюючи єдину, синхронізовану систему.

В умовах стрімкого розвитку безпілотних технологій виникає потреба у створенні нових, більш ефективних систем орієнтування і паркування. Комплексне використання мікроконтролерів STM32, ультразвукових датчиків

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

НС-SR04 та блоків керування Pixhawk дозволяє розробляти системи, які можуть забезпечити високу точність та надійність у різних умовах експлуатації.

Таким чином, вивчення та розробка систем орієнтування і паркування для безпілотних транспортних засобів є надзвичайно актуальною темою. Це дослідження спрямоване на підвищення безпеки та ефективності БТЗ, що має важливе значення для подальшого розвитку транспортних технологій і їх інтеграції в наше повсякденне життя.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Огляд аналогів та застосування системи

Системи орієнтування та паркування для безпілотних транспортних засобів (БПТЗ) є ключовими компонентами, що визначають їх ефективність та безпеку. Розглянемо існуючі аналоги, які вже успішно застосовуються в різних сферах.

Одним з найбільш відомих прикладів таких систем є автопілот компанії Tesla. Tesla використовує комбінацію камер, радарів і ультразвукових датчиків для збору даних про навколишнє середовище. Ці дані обробляються за допомогою потужних алгоритмів машинного навчання та штучного інтелекту, що дозволяє автопілоту не тільки розпізнавати перешкоди та інші транспортні засоби, але й прогнозувати їхню поведінку. Tesla автопілот здатен виконувати складні маневри, такі як зміну смуги руху та автоматичне паркування, що значно підвищує комфорт і безпеку водія.

Система автоматичного паркування Park Assist, яка використовується в багатьох сучасних автомобілях, також заслуговує на увагу. Ця система використовує ультразвукові датчики для визначення відстані до перешкод і допомагає водію виконувати маневри паркування. Park Assist може автоматично керувати рульовим управлінням, тоді як водій контролює швидкість руху. Це значно знижує ризик пошкоджень автомобіля під час паркування та полегшує процес для водіїв, особливо в умовах обмеженого простору.

У галузі логістики та складського господарства широко застосовуються роботизовані навантажувачі і складські роботи, які використовують схожі технології. Ці роботи обладнані лідарами, камерами та інфрачервоними датчиками, що дозволяє їм точно орієнтуватися у складських приміщеннях,

					КС КРБ 123.329.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Якобчук Р.А.			Аналіз технічного завдання	Літ.	Арк.	Акрушів
Перевір.		Лецишин Ю.З.					10	
Рецензент						ТНТУ, каф. КС, гр. СІс-42		
Н. контр.		Луцик Н.С.						
Зав. каф.		Осухівська Г.М.						

уникати перешкод і виконувати складні маневри з вантажами. Використання таких систем дозволяє підвищити ефективність складських операцій, знизити ймовірність помилок і мінімізувати ризик пошкодження товарів.

У сільському господарстві безпілотні трактори та інші машини використовують системи орієнтування для автоматичного керування під час обробки полів. Ці системи базуються на GPS-навігації, камерах та інших сенсорах, що дозволяє точно визначати положення трактора і виконувати завдання з високою точністю. Це сприяє підвищенню продуктивності та зниженню витрат на сільськогосподарські роботи, а також забезпечує оптимальне використання ресурсів.

Ще одним прикладом застосування подібних технологій є роботи-пилососи, які використовують камери та інфрачервоні сенсори для навігації по приміщенню і прибирання. Такі роботи здатні ефективно виявляти перешкоди, планувати оптимальні маршрути для прибирання і автоматично повертатися до зарядної станції після завершення роботи або при низькому рівні заряду батареї.

Застосування систем орієнтування та паркування не обмежується лише автомобільною промисловістю. У побутовій техніці, промислових роботах, сільському господарстві та інших сферах ці технології сприяють підвищенню ефективності, безпеки та зручності використання різних пристроїв і машин.

Таким чином, існуючі аналоги демонструють високу ефективність і надійність систем орієнтування та паркування, що використовуються в безпілотних транспортних засобах і інших сферах. Це підтверджує важливість розробки таких систем для подальшого вдосконалення і підвищення безпеки автоматизованих процесів.

1.2 Аналіз вимог до комп'ютерної системи

1.2.1 Призначення системи та функціональні вимоги

Система призначена для забезпечення безпеки та точності руху автономних або напівавтономних транспортних засобів та роботів. Вона буде

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

використовуватись для збору та обробки даних про відстань до об'єктів, що дозволить запобігати зіткненням та забезпечувати оптимальні траєкторії руху. Основними сферами застосування системи є промисловість, де автоматизовані машини повинні точно рухатись у обмежених просторах, сільське господарство, де роботи повинні обробляти поля з високою точністю, а також військова техніка, де безпілотні апарати виконують завдання у складних умовах.

Система повинна забезпечувати:

- безперервний та точний збір даних з датчиків відстані, що встановлені на транспортному засобі або роботі;
- обробку зібраних даних у реальному часі для швидкого виявлення перешкод та прийняття рішень щодо їх обходу;
- відображення оброблених даних на LCD дисплеї, щоб оператор міг контролювати стан системи та приймати відповідні заходи;
- передачу зібраних та оброблених даних на блок керування Pixhawk, що дозволяє здійснювати комплексний аналіз та управління системою;
- інтуїтивно зрозумілий інтерфейс для налаштування параметрів системи та перегляду зібраних даних;
- підтримку енергозберігаючих режимів роботи, що дозволить знизити загальне енергоспоживання та продовжити час автономної роботи системи.

Додатково, система повинна функціонувати у різних умовах освітлення та бути стійкою до зовнішніх впливів, таких як пил, волога та механічні навантаження

1.2.2 Апаратні та програмні вимоги

Розробка комп'ютеризованої системи орієнтування і паркування для безпілотних транспортних засобів вимагає ретельного аналізу вимог, що стосуються різних аспектів цієї системи. Вимоги до системи охоплюють апаратне забезпечення, програмне забезпечення, інтерфейси, продуктивність, енергоспоживання, надійність і безпеку.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

У центрі системи знаходиться мікроконтролер STM32F407VGT6, який виконує роль основного обчислювального вузла. Цей мікроконтролер має 32-розрядний процесор Cortex-M4 з частотою до 168 МГц, що забезпечує достатню обчислювальну потужність для обробки даних з декількох датчиків одночасно. Важливим є також великий обсяг флеш-пам'яті (1 МБ) і оперативної пам'яті (192 КБ), що дозволяє зберігати та швидко обробляти необхідні дані.

Для визначення відстані до перешкод використовуються ультразвукові датчики HC-SR04. Ці датчики здатні вимірювати відстань у діапазоні від 2 см до 400 см з точністю до 3 мм. Вони працюють на принципі ехолокації: датчик випромінює ультразвуковий імпульс і вимірює час, за який відбитий сигнал повертається. Такий підхід забезпечує високу точність і надійність вимірювань, що є критично важливим для безпечного паркування.

Для візуалізації даних з датчиків використовується дисплей, який може бути або текстовим (LCD 16x4), або графічним (OLED або TFT).

Дисплей виконує функцію інтерфейсу для користувача, показуючи інформацію про відстань до перешкод у реальному часі. Важливими параметрами є розмір і роздільна здатність дисплея, а також зручність його використання у реальних умовах експлуатації.

Блок керування Pixhawk виконує роль інтегратора, який отримує дані з мікроконтролера STM32 і використовує їх для прийняття рішень щодо руху безпілотного транспортного засобу. Pixhawk підтримує протокол Mavlink, який забезпечує стандартизовану передачу даних між компонентами системи.

Програмне забезпечення мікроконтролера повинно забезпечувати безперервне опитування датчиків відстані, збір даних і їх обробку в реальному часі. Для цього необхідно розробити алгоритми, які дозволять коректно зчитувати сигнали з датчиків, обчислювати відстань до перешкод і фільтрувати отримані дані для усунення шумів і викидів. Важливо забезпечити мінімальні затримки при обробці даних, щоб система могла оперативно реагувати на зміну обстановки.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Для виведення даних на дисплей необхідно розробити програмне забезпечення, яке буде формувати дані у зручному для користувача вигляді. Це може включати як текстове відображення відстаней до перешкод, так і графічне представлення даних у вигляді заповнюваних поділок, що імітує роботу звичайного парктроніка. Програмне забезпечення повинно забезпечувати регулярне оновлення інформації на дисплеї в реальному часі.

Однією з ключових функцій системи є передача даних на блок керування Pixhawk через протокол Mavlink. Це вимагає розробки програмного забезпечення, яке забезпечить формування повідомлень у форматі Mavlink і їх передачу через інтерфейс UART. Важливо забезпечити надійність і безпомилковість передачі даних, оскільки від цього залежить безпека руху безпілотного транспортного засобу.

1.2.3 Інтерфейси системи та інтеграція з існуючими системами

Датчики відстані HC-SR04 підключаються до мікроконтролера через GPIO-піни. Для коректної роботи необхідно забезпечити налаштування відповідних пінів мікроконтролера як вхідних або вихідних залежно від режиму роботи датчика (тригера або еха). Програмне забезпечення повинно забезпечувати правильну послідовність операцій для запуску вимірювань і зчитування результатів.

Дисплей може бути підключений до мікроконтролера через інтерфейс SPI або I2C, залежно від моделі дисплея. Необхідно забезпечити налаштування відповідних периферійних інтерфейсів мікроконтролера і розробити драйвери для роботи з дисплеєм. Це включає ініціалізацію дисплея, передачу даних і команд, а також обробку зворотного зв'язку від дисплея.

Для передачі даних на блок керування Pixhawk використовується інтерфейс UART. Протокол Mavlink визначає формат повідомлень і послідовність передачі даних. Необхідно забезпечити коректну роботу інтерфейсу UART, включаючи налаштування швидкості передачі даних,

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

контроль чіткої послідовності передачі і перевірку цілісності переданих повідомлень.

Блок керування Pixhawk використовує протокол Mavlink для обміну даними з іншими компонентами системи. Необхідно забезпечити повну сумісність з цим протоколом, включаючи підтримку всіх необхідних повідомлень і команд.

Система орієнтування і паркування повинна бути інтегрована з іншими системами безпілотного транспорту, такими як системи навігації, керування рухом і забезпечення безпеки. Це вимагає розробки універсальних інтерфейсів і протоколів для обміну даними.

Система повинна мати можливість масштабування для підтримки додаткових датчиків або функцій у майбутньому. Це включає використання модульної архітектури і підтримку розширюваних інтерфейсів.

Аналіз вимог до комп'ютеризованої системи орієнтування і паркування для безпілотних транспортних засобів дозволяє визначити основні параметри і характеристики, які необхідно врахувати при її розробці. Це включає вибір апаратних і програмних компонентів, забезпечення надійності і безпеки, а також інтеграцію з існуючими системами. Врахування всіх цих аспектів дозволить створити ефективну і надійну систему, яка забезпечить безпечне і точне паркування безпілотних транспортних засобів.

1.2.4 Вимоги до продуктивності, енергоспоживання, надійності і безпеки

Система повинна забезпечувати високу частоту опитування датчиків для отримання актуальної інформації про обстановку навколо безпілотного транспортного засобу. Це важливо для швидкого реагування на зміну умов і забезпечення безпеки руху. Оптимальна частота опитування повинна бути не менше 10-20 Гц.

Час обробки даних з датчиків і їх підготовки для передачі на блок керування повинен бути мінімальним. Це дозволить забезпечити оперативне

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

реагування системи на зміну обстановки. Важливо оптимізувати алгоритми обробки даних для зменшення затримок.

Інтерфейси між компонентами системи повинні забезпечувати достатню пропускну здатність для передачі даних у режимі реального часу. Це включає як інтерфейс між датчиками і мікроконтролером, так і інтерфейси для передачі даних на дисплей і блок керування. Пропускна здатність повинна враховувати можливі пікові навантаження і забезпечувати стабільну роботу системи.

Датчики відстані повинні мати низьке енергоспоживання, щоб не знижувати загальну енергоефективність системи. Використання ультразвукових датчиків HC-SR04 забезпечує відносно низьке енергоспоживання, що є важливим для автономних систем.

Мікроконтролер STM32F407VGT6 підтримує режими низького енергоспоживання, що дозволяє зменшити споживання енергії у стані очікування. Це важливо для забезпечення тривалого часу роботи системи від автономних джерел живлення.

Система повинна бути спроектована таким чином, щоб мінімізувати енергоспоживання під час роботи. Це включає використання режимів енергозбереження для компонентів системи і динамічне керування енергоспоживанням в залежності від поточних умов роботи.

Всі апаратні компоненти системи повинні мати високу надійність і стійкість до зовнішніх впливів. Це включає вибір компонентів з високими технічними характеристиками і тестування їх на відповідність вимогам до експлуатації у реальних умовах.

Програмне забезпечення повинно бути написане з урахуванням найкращих практик розробки, включаючи модульне тестування, відладку і оптимізацію. Необхідно забезпечити відмовостійкість системи і здатність до автоматичного відновлення після збоїв.

Передача даних між компонентами системи повинна бути захищена від можливих загроз, таких як перехоплення або модифікація даних.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА

2.1 Розробка узагальненої структури комп'ютерної системи

Розробка узагальненої структури комп'ютеризованої системи є критичним етапом у процесі створення системи орієнтування та паркування для безпілотних транспортних засобів. На цьому етапі визначаються основні компоненти системи, їх взаємозв'язки та загальна логіка функціонування. Це дозволяє забезпечити злагоджену роботу всіх елементів та високу ефективність системи в цілому.

Основним компонентом системи є мікроконтролер STM32, який виконує роль центрального процесора. Мікроконтролер відповідає за збір даних з датчиків, їх обробку та передачу результатів на дисплей та блок керування Pixhawk. Для реалізації цих функцій використовуються різні периферійні модулі мікроконтролера, такі як інтерфейси I2C, UART та GPIO.

Система включає декілька датчиків відстані, розміщених на різних частинах транспортного засобу. Датчики відстані забезпечують вимірювання відстані до об'єктів навколо автомобіля, що є критично важливим для точного орієнтування та безпечного паркування. Дані з датчиків зчитуються мікроконтролером через цифрові входи, а потім обробляються для визначення точної відстані до найближчих перешкод.

Отримані дані обробляються мікроконтролером з використанням спеціальних алгоритмів. Дані передаються на блок керування Pixhawk за допомогою протоколу Mavlink. Блок керування Pixhawk використовує ці дані для прийняття рішень щодо маневрування та контролю руху транспортного засобу. Передача даних здійснюється через інтерфейс UART, що забезпечує надійну і швидку комунікацію між мікроконтролером та блоком керування.

					КС КРБ 123.329.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Якобчук Р.А.			Літ.	Арк..	Акрушів
Перевір.		Лещинин Ю.З.				17	
Рецензент					Проектна частина		
Н. контр.		Луцик Н.С.			ТНТУ, каф. КС, гр. СІс-42		
Зав. каф.		Осухівська Г.М.					

Розробка узагальненої структури також включає визначення енергоспоживання кожного компонента та заходи з енергозбереження. Це важливо для забезпечення тривалої автономної роботи системи. Мікроконтролер працює в режимах зниженої енергоспоживання під час простою, а датчики відключаються, коли не потрібні.

На рис 2.1 представлена структурна схема комп'ютеризованої системи, що включає мікроконтролер, датчики відстані та блок керування Pixhawk.

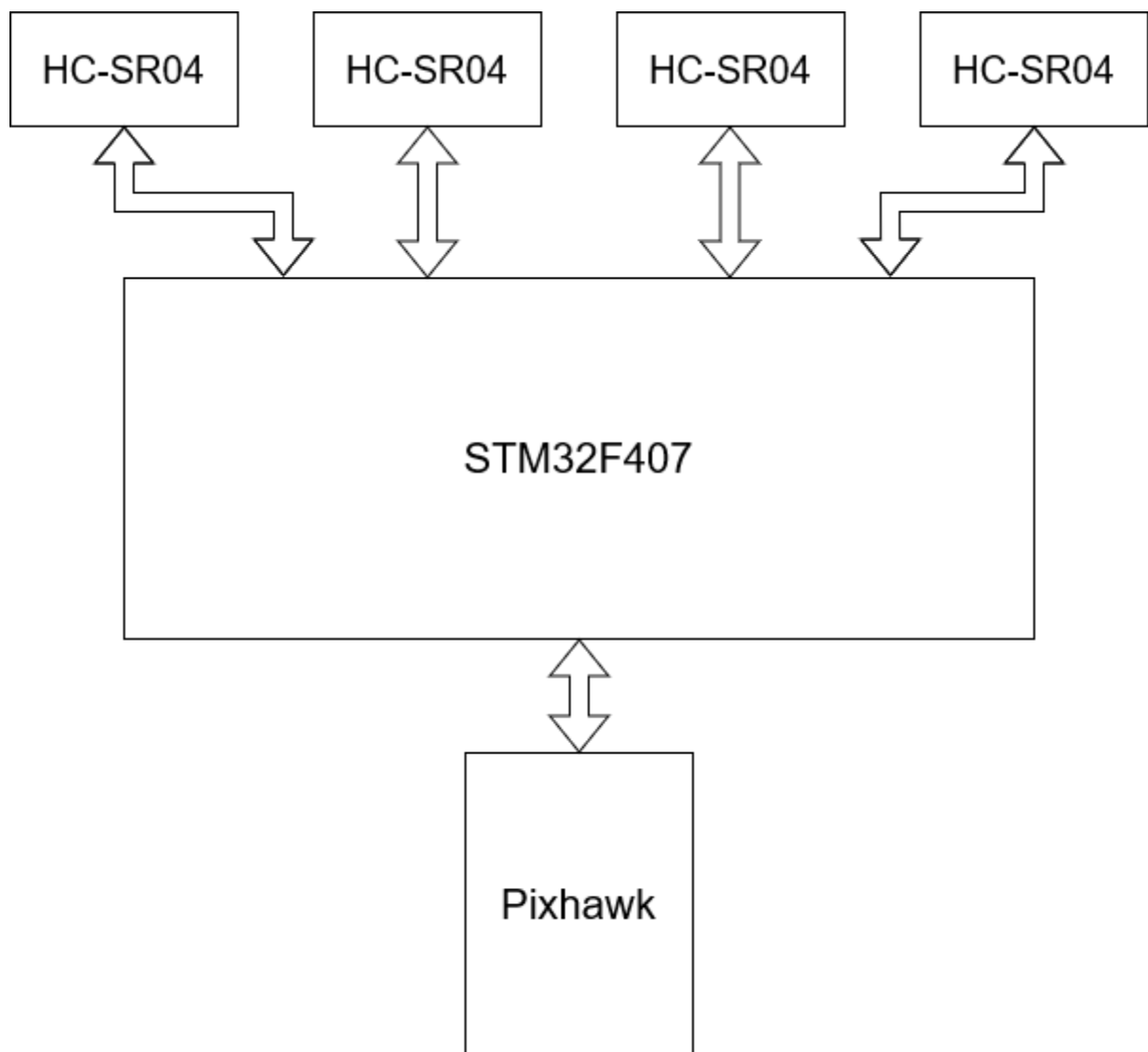


Рисунок 2.1 – Структурна схема

Розробка узагальненої структури також враховує можливі сценарії використання системи. Це дозволяє визначити вимоги до надійності та безпеки

системи. Наприклад, під час паркування в умовах обмеженого простору система повинна забезпечувати високу точність вимірювань та швидку обробку даних для запобігання зіткненням.

Таким чином, узагальнена структура комп'ютеризованої системи орієнтування та паркування включає всі необхідні компоненти та визначає їх взаємодію, що забезпечує ефективну і надійну роботу системи в реальних умовах.

2.2 Обґрунтування вибору апаратного забезпечення проекрованої КС

У цьому розділі розглядається вибір апаратних компонентів для реалізації системи орієнтування і паркування безпілотних транспортних засобів. Підбір компонентів здійснювався з урахуванням їх технічних характеристик, надійності, сумісності та доступності на ринку.

Мікроконтролери серії STM32 від компанії STMicroelectronics зарекомендували себе як потужні та надійні рішення для різноманітних завдань. Серед багатьох варіантів, таких як серії STM32F0, STM32F1, STM32F2, STM32F3, STM32F4, STM32F7, STM32H7 та STM32L, особливу увагу привертає серія STM32F4. Вона відзначається високою продуктивністю, розширеними можливостями периферії та оптимальним співвідношенням ціни і продуктивності.

Мікроконтролер STM32F407VGT6, зображений на рис. 2.2, обраний для проекту через його високі технічні характеристики. Він базується на ядрі ARM Cortex-M4 з робочою частотою до 168 МГц, що забезпечує високу обчислювальну потужність для обробки даних з декількох датчиків та виконання складних алгоритмів. Мікроконтролер має 1 МБ флеш-пам'яті та 192 КБ оперативної пам'яті, що дозволяє зберігати та швидко опрацьовувати великі обсяги даних.

					КС КРБ 123.329.00.00 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.2 – Мікроконтролер STM32F407VGT6

Однією з ключових переваг STM32F407VGT6 є велика кількість периферійних інтерфейсів, таких як USART, SPI, I2C, CAN, USB OTG, Ethernet, та інші. Це забезпечує гнучкість у підключенні різноманітних сенсорів, дисплеїв та інших пристроїв. Вбудовані 12-бітні АЦП (ADC) та 12-бітні ЦАП (DAC) дозволяють ефективно працювати з аналоговими сигналами, що є важливим для нашого проекту.

Мікроконтролер STM32F407VGT6 також має розширені можливості в області таймерів та PWM, що дозволяє точно керувати різними пристроями та забезпечувати високу точність вимірювань. Вбудовані контролери DMA забезпечують швидкий та ефективний обмін даними між периферійними пристроями та пам'яттю мікроконтролера, що зменшує навантаження на центральне ядро.

Загалом, вибір STM32F407VGT6 обумовлений його високою продуктивністю, багатими периферійними можливостями та надійністю. Крім того, STMicroelectronics надає широкий спектр інструментів для розробки та підтримки, що значно спрощує процес розробки та налагодження.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

2.2.1 Вибір датчиків відстані, технічні характеристики і переваги

Датчики відстані HC-SR04, зображені на рис. 2.3, є одними з найпоширеніших ультразвукових датчиків, що використовуються в робототехніці та автоматизації. Їх вибір для проекту обумовлений високою точністю вимірювань, простотою використання та низькою вартістю.

Вигляд датчика відстані HC-SR04 зображено на рисунку 2.2.



Рисунок 2.3 – Датчик відстані HC-SR04

Датчик працює на частоті 40 кГц і здатен вимірювати відстані в діапазоні від 2 см до 400 см з точністю ± 3 мм. Це дозволяє ефективно використовувати його для задач орієнтування та паркування, де необхідно точно визначати відстань до перешкод.

Підключення HC-SR04 до мікроконтролера є досить простим завдяки стандартному інтерфейсу. Датчик має чотири контакти: VCC, GND, Trig та Echo. Для вимірювання відстані необхідно подати короткий імпульс на контакт Trig, після чого датчик відправляє ультразвуковий сигнал і чекає на його відбиття. Контакт Echo видає імпульс, тривалість якого пропорційна відстані до перешкоди. Цей сигнал можна легко зчитувати та обробляти за допомогою мікроконтролера.

Основні переваги HC-SR04 включають високу точність вимірювань, надійність роботи в різних умовах та простоту інтеграції в систему. Крім того,

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

низька вартість робить їх доступними для широкого кола застосувань, що дозволяє створювати економічно ефективні рішення.

2.2.2 Вибір дисплея, технічні характеристики і порівняння варіантів

Для відображення інформації з датчиків та стану системи необхідний дисплей. Вибір типу дисплея залежить від вимог до якості зображення, енергоспоживання та вартості. Серед найбільш поширених варіантів є LCD, OLED та TFT дисплеї.

LCD дисплеї мають низьке енергоспоживання та вартість, але обмежені за якістю зображення та кутом огляду. OLED дисплеї забезпечують високу контрастність та насичені кольори, але мають вищу вартість та короткий термін служби. TFT дисплеї поєднують високу якість зображення та широкий кут огляду, що робить їх оптимальним вибором для нашої системи.

Для цього проекту був обраний LCD дисплей 16x4, оскільки він має ряд переваг, які роблять його оптимальним варіантом для нашої задачі. Дисплей 16x4 означає, що він має 16 символів у кожному з чотирьох рядків. Це дозволяє відображати достатню кількість інформації для КС, включаючи дані з датчиків відстані та індикатори наближення до перешкод. Він забезпечує необхідну функціональність, енергоефективність і надійність, що є критично важливими факторами для безпілотних транспортних засобів.

2.2.3 Вибір блоку керування, технічні характеристики і обґрунтування

Pixhawk є потужним блоком керування, який широко використовується в безпілотних літальних апаратах та наземних роботах. Його вибір для системи обумовлений високою надійністю, гнучкістю та підтримкою відкритого протоколу Mavlink.

Контролер базується на процесорі ARM Cortex-M4 з частотою 168 МГц, що забезпечує високу продуктивність для обробки даних з численних датчиків та виконання складних алгоритмів керування. Він оснащений 256 КБ

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

оперативної пам'яті (SRAM) та 2 МБ флеш-пам'яті для зберігання програмного забезпечення та даних.

Однією з ключових особливостей Pixhawk є вбудовані датчики, такі як гіроскопи, акселерометри, магнітометри та барометри, що дозволяє точно визначати положення та рух апарату. Це особливо важливо для задач орієнтування та стабілізації.

Pixhawk підтримує широкий спектр периферійних інтерфейсів (рис. 2.4), включаючи USART, SPI, I2C, CAN, USB та PWM. Це дозволяє легко інтегрувати його з різноманітними сенсорами, модулями зв'язку та іншими пристроями. Протокол Mavlink забезпечує стандартизований обмін даними між різними компонентами системи, що спрощує процес інтеграції та налаштування.



Рисунок 2.4 – Контролер Pixhawk

Крім того, контролер має розширені можливості для керування двигунами та сервоприводами, що дозволяє створювати складні системи керування рухом. Він підтримує різні типи живлення, що забезпечує гнучкість у виборі джерела живлення та надійність роботи системи.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

Загалом, вибір контролера Pixhawk обумовлений його високою продуктивністю, надійністю та широкими можливостями інтеграції. Його використання дозволяє створити ефективну та надійну систему.

2.3 Розробка схеми з'єднань та електричної принципової схеми комп'ютеризованої системи орієнтування і паркування БТЗ

2.3.1 Розробка схеми з'єднань

Для забезпечення правильного функціонування системи орієнтування і паркування необхідно коректно підключити датчики відстані HC-SR04 до мікроконтролера STM32F407VGT6. Датчики HC-SR04 мають чотири контакти:

- VCC та GND: ці контакти підключаються до відповідних ліній живлення мікроконтролера. VCC підключається до виходу 5V або 3.3V (в залежності від того, яке живлення підтримує датчик), а GND - до землі;

- Trig: цей контакт використовується для генерування ультразвукового імпульсу. Для кожного датчика цей контакт підключається до одного з цифрових виходів мікроконтролера. Вибір конкретного виходу залежить від конфігурації проекту;

- Echo: контакт Echo видає сигнал, тривалість якого пропорційна відстані до перешкоди. Цей контакт підключається до одного з цифрових входів мікроконтролера, який здатен зчитувати імпульсну тривалість.

Мікроконтролер STM32F407VGT6 має достатньо цифрових входів/виходів для підключення чотирьох датчиків HC-SR04. Для оптимізації роботи система може використовувати апаратні таймери мікроконтролера для точного вимірювання тривалості імпульсів Echo.

LCD дисплей 16x4 підключається до мікроконтролера STM32F407VGT6 за допомогою шини I2C. Цей спосіб підключення є зручним і ефективним, оскільки використовує лише чотири дроти:

- GND (земля): Цей роз'єм дисплея підключається до загальної землі (GND) мікроконтролера;

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

- VCC (живлення): Цей роз'єм дисплея підключається до виходу живлення мікроконтролера (звичайно 5V або 3.3V, залежно від моделі дисплея);
- SDA (Serial Data Line): Ця лінія передає дані між дисплеєм і мікроконтролером. Вона підключається до відповідного піну SDA на мікроконтролері;
- SCL (Serial Clock Line): Ця лінія передає сигнали тактування, синхронізуючи передачу даних. Вона підключається до відповідного піну SCL на мікроконтролері.

Блок керування Pixhawk використовується для прийому даних про відстань від мікроконтролера STM32F407VGT6 та для керування безпілотним транспортним засобом. Зв'язок між мікроконтролером і Pixhawk здійснюється через інтерфейс UART з використанням протоколу Mavlink.

а) для живлення мікроконтролера і блоку Pixhawk використовуються відповідні лінії живлення (VCC та GND);

б) для передачі даних використовується інтерфейс UART. Мікроконтролер має декілька UART портів, що дозволяє легко підключити Pixhawk:

- TX (Transmit): підключається до RX (Receive) Pixhawk;
- RX (Receive): підключається до TX (Transmit) Pixhawk.

Програмне забезпечення мікроконтролера забезпечує формування та відправку повідомлень за протоколом Mavlink, що дозволяє Pixhawk отримувати необхідні дані для керування рухом безпілотного транспортного засобу.

На рис. 2.5 зображено схему з'єднань комп'ютеризованої системи орієнтування і паркування безпілотних транспортних засобів.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

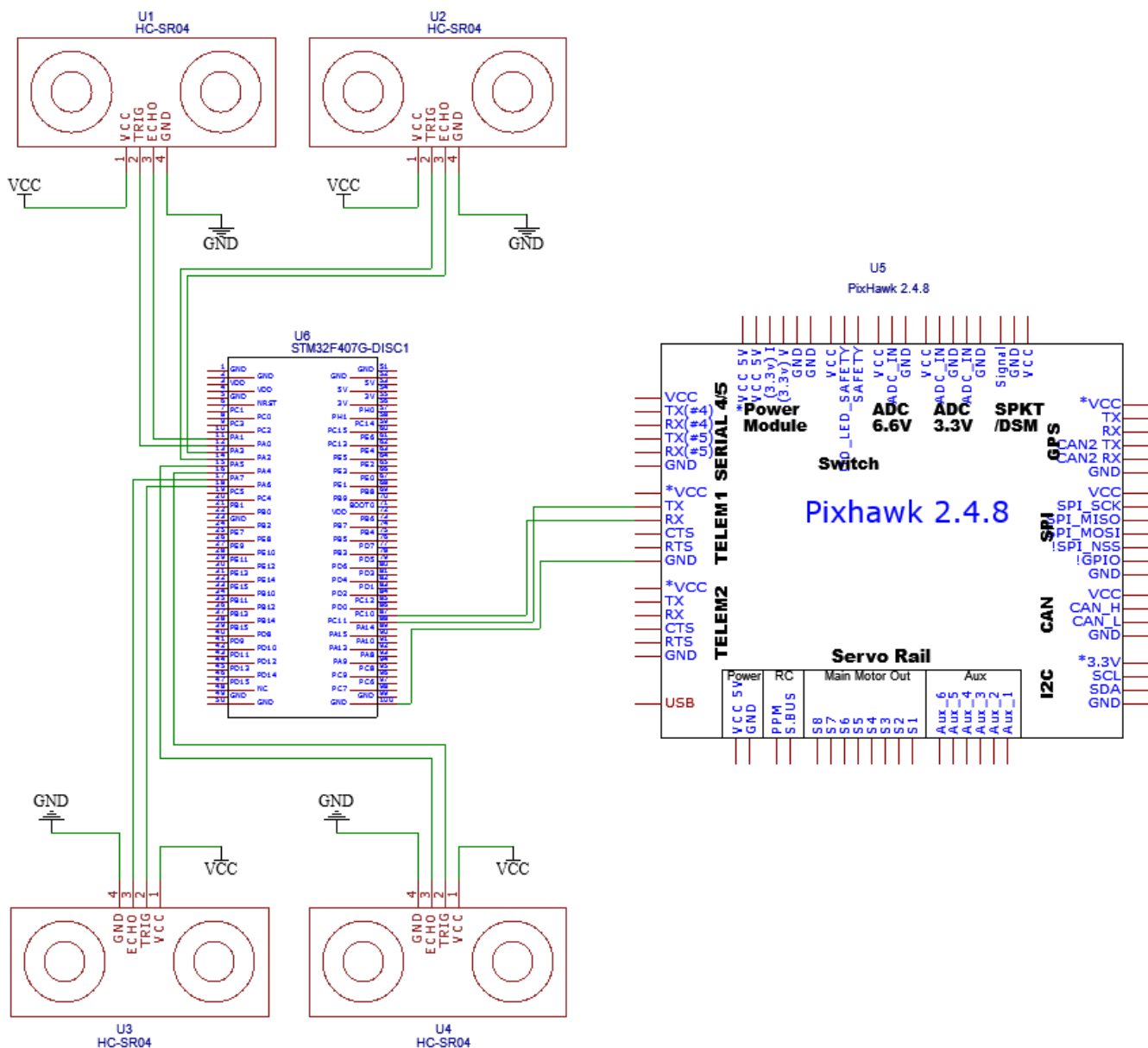


Рисунок 2.5 – Електрична схема з'єднань

2.3.2 Розробка електричної принципової схеми

Розробка електричної принципової схеми є ключовим етапом у створенні комп'ютеризованої системи орієнтування і паркування для безпілотних транспортних засобів. Цей етап включає детальне проектування всіх електричних з'єднань між компонентами системи, що забезпечує їх коректну взаємодію та функціонування.

На рис. 2.6 зображена електрична принципова схема комп'ютеризованої системи орієнтування і паркування безпілотних транспортних засобів.

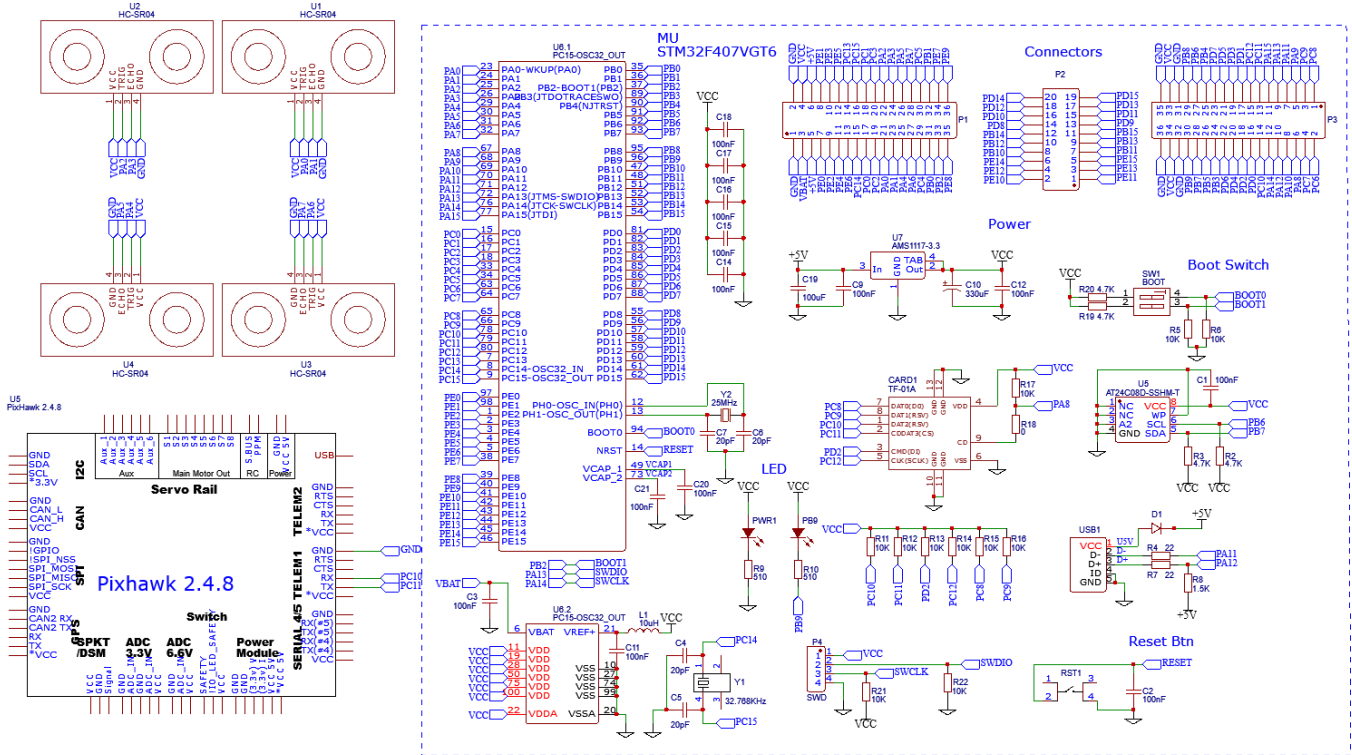


Рисунок 2.6 – Електрична принципова схема

2.4 Розробка програмного забезпечення

2.4.1 Програмна архітектура системи

Програмна архітектура системи розробляється для забезпечення інтеграції різних модулів та забезпечення їх ефективною взаємодією. Вибір правильної архітектури є критичним для забезпечення надійності, масштабованості та високої продуктивності системи.

Архітектура складається з декількох шарів, кожен з яких виконує специфічні функції. На найнижчому рівні знаходяться драйвери апаратних пристроїв, які забезпечують взаємодію з датчиками HC-SR04, дисплеєм та блоком керування Pixhawk. Ці драйвери дозволяють програмному забезпеченню взаємодіяти з апаратними компонентами через стандартизовані інтерфейси.

На середньому рівні розташовані модулі обробки даних, які відповідають за збирання, фільтрацію та обробку даних з датчиків. Ці модулі також виконують функції інтеграції даних з кількох датчиків для створення цілісної картини навколишнього середовища.

На найвищому рівні знаходиться модуль управління та інтерфейс користувача. Цей модуль забезпечує відображення оброблених даних на дисплеї у зрозумілому вигляді. Інтерфейс включає графічні елементи, такі як шкали та індикатори, які відображають відстань до перешкод. Крім того, модуль управління забезпечує передачу даних до блоку керування Pixhawk через протокол Mavlink, що дозволяє системі приймати та виконувати команди від центрального контролера.

2.4.2 Розробка ПЗ для збору і обробки даних з датчиків

Розробка програмного забезпечення для збору і обробки даних з датчиків є критично важливою частиною системи. Це ПЗ забезпечує точне зчитування даних з ультразвукових датчиків HC-SR04 та їх подальшу обробку для отримання надійних результатів.

На початковому етапі важливо правильно налаштувати апаратні інтерфейси для взаємодії з датчиками. Мікроконтролер STM32F4 використовує GPIO порти для генерування тригерних імпульсів та вимірювання тривалості відгуків імпульсів. Точне вимірювання часу між тригерним і відгуким імпульсами дозволяє обчислити відстань до об'єкта.

Після збору даних з датчиків, ПЗ повинно виконати їх обробку для усунення шумів та аномальних значень. Оброблені дані потім передаються на наступні модулі для подальшої обробки або відображення.

Однією з ключових задач цього ПЗ є забезпечення точності та надійності даних, що особливо важливо у складних умовах навколишнього середовища.

2.4.3 Розробка ПЗ для відображення даних на дисплеї

Програмне забезпечення для відображення даних на дисплей є важливим компонентом системи, оскільки воно забезпечує зручний інтерфейс для користувача. Основна задача цього програмного забезпечення полягає у візуалізації оброблених даних з датчиків на LCD дисплей.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

Вибір дисплея і методів його підключення має значний вплив на ефективність роботи системи. Для забезпечення швидкого та надійного обміну даними між мікроконтролером та дисплеєм часто використовують інтерфейс I2C, який дозволяє передавати дані з високою швидкістю.

Програмне забезпечення для відображення даних включає ініціалізацію дисплея, налаштування графічних елементів та постійне оновлення інформації на екрані. Важливо забезпечити швидке та ефективне оновлення графічних елементів, щоб користувач міг отримувати актуальну інформацію в режимі реального часу.

Графічний інтерфейс може включати різні елементи, такі як шкали, індикатори та текстові повідомлення. Ці елементи повинні бути розроблені таким чином, щоб забезпечити зручність та інтуїтивну зрозумілість для користувача. Використання кольорових індикаторів та звукових сигналів може додатково підвищити ефективність інтерфейсу.

2.4.4 Розробка ПЗ для передачі даних на блок керування за допомогою протоколу Mavlink

Розробка програмного забезпечення для передачі даних на блок керування Pixhawk з використанням протоколу Mavlink є ключовим етапом у забезпеченні інтеграції системи орієнтування і паркування з іншими компонентами безпілотного транспортного засобу. Протокол Mavlink є популярним і широко використовуваним у системах безпілотних апаратів, що забезпечує надійну та ефективну передачу даних.

Програмне забезпечення для цього модуля включає формування та передавання повідомлень у форматі Mavlink. Для цього необхідно правильно налаштувати UART інтерфейс на мікроконтролері та реалізувати алгоритми кодування даних у повідомлення Mavlink. Повідомлення можуть містити різноманітні дані, такі як відстані до перешкод, статус сенсорів, діагностичну інформацію.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

Особлива увага приділяється обробці даних перед передаванням, щоб забезпечити їх коректність і актуальність. Використання буферів та механізмів контролю помилок дозволяє мінімізувати ймовірність втрати або спотворення даних під час передачі.

Крім передачі даних, ПЗ повинно також обробляти відповіді від блоку керування Pixhawk. Це забезпечує двосторонній зв'язок і дозволяє системі реагувати на команди та запити від центрального контролера. Реалізація цього модуля включає розробку алгоритмів для парсингу вхідних повідомлень Mavlink та виконання відповідних дій.

Таким чином, розробка програмного забезпечення для передачі даних на блок керування за допомогою протоколу Mavlink є важливим етапом у створенні комплексної системи орієнтування і паркування, що забезпечує надійну і ефективну взаємодію між різними компонентами безпілотного транспортного засобу.

2.5 Алгоритми прийняття рішень на основі даних з датчиків

Алгоритми прийняття рішень на основі даних з датчиків відстані є центральним елементом системи, оскільки саме вони визначають дії безпілотного транспортного засобу у відповідь на виявлені перешкоди.

На основі оброблених даних з датчиків, алгоритми аналізують наявність перешкод. Якщо відстань до об'єкта менша за встановлений поріг, система визначає об'єкт як перешкоду.

Алгоритми визначають не лише наявність перешкод, але й оцінюють їхню відстань та напрямок. Це дозволяє транспортному засобу зрозуміти, де саме знаходиться перешкода та як вона може вплинути на рух.

Залежно від отриманих даних, система приймає рішення щодо дій, які необхідно виконати. Наприклад, якщо перешкода виявлена на шляху руху, система може прийняти рішення про зміну траєкторії або уповільнення.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

Прийняті рішення передаються блоку керування Pixhawk, який здійснює необхідні дії. Це забезпечує плавну інтеграцію між збором даних, їх обробкою та фізичним виконанням команд. На рис. 2.7 зображено блок схему алгоритму роботи системи.

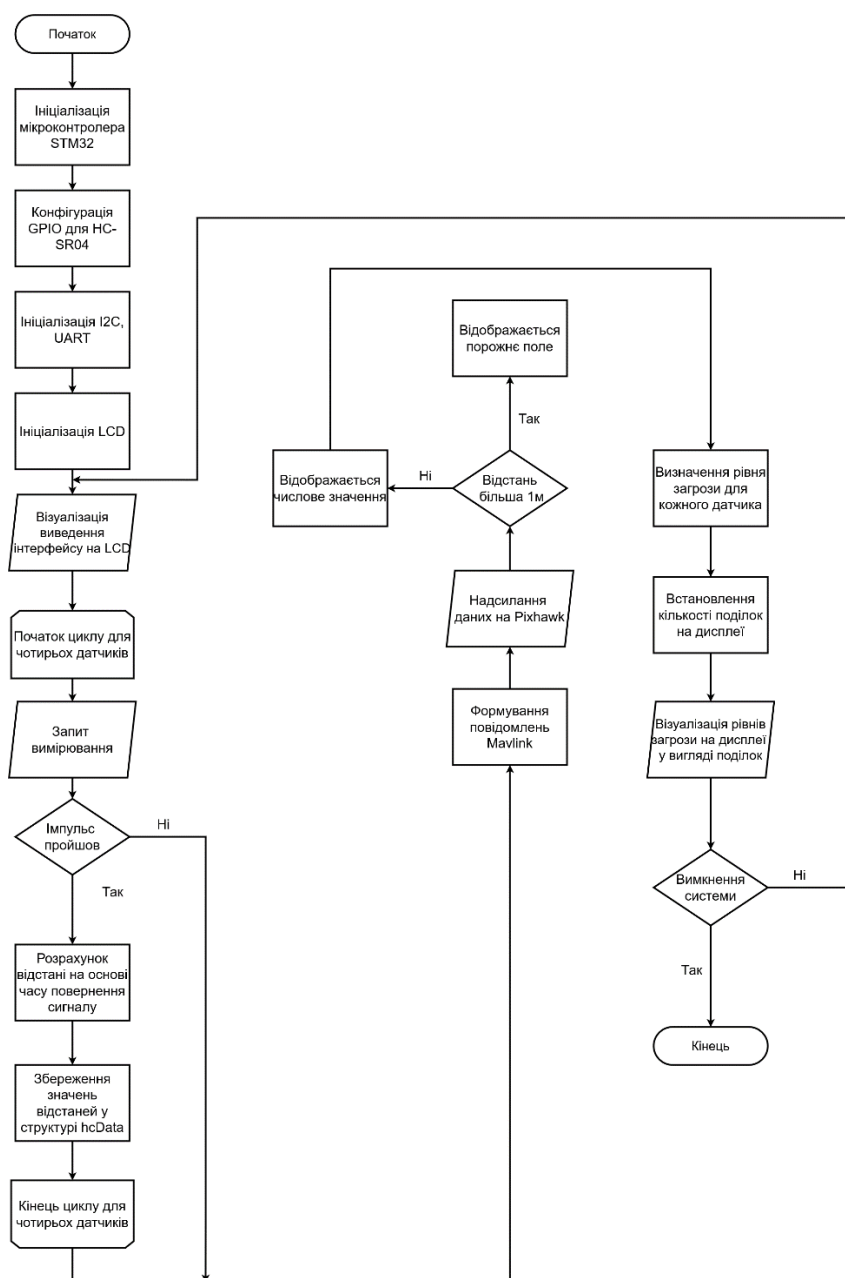


Рисунок 2.7 – Блок схема алгоритму роботи системи

Система запускається, виконується початкова ідентифікація компонентів і підготовка до роботи. На цьому етапі встановлюються необхідні параметри та конфігурації для подальшого виконання завдань.

Змн.	Арк.	№ докум.	Підпис	Дата

Мікроконтролер STM32 налаштовується для роботи із заданими параметрами. Встановлюються тактові частоти, ініціалізуються внутрішні периферійні модулі та системні таймери.

Налаштовуються GPIO порти для роботи з ультразвуковими датчиками HC-SR04. Конфігуруються пін для тригерного сигналу (Trig) та прийому ехо-сигналу (Echo).

Налаштовуються інтерфейси I2C та UART для передачі даних. I2C використовується для зв'язку з LCD дисплеєм, а UART — для передачі даних на блок керування Pixhawk.

LCD дисплей налаштовується для відображення інформації. Встановлюються початкові параметри роботи дисплея, такі як адресація та формат відображення символів.

На дисплеї виводиться початковий інтерфейс користувача. Це може включати схематичне зображення автомобіля та відстаней до перешкод.

Починається цикл зчитування даних з чотирьох датчиків HC-SR04. Цей цикл буде повторюватись для кожного датчика.

Для кожного датчика запускається вимірювання відстані. Надсилається імпульс на тригерний пін (Trig) датчика.

Перевіряється, чи був отриманий ехо-сигнал. Якщо ехо-сигнал прийшов, то вимірюється час його повернення. Якщо ні, цикл переходить до наступного датчика.

На основі виміряного часу повернення ехо-сигналу розраховується відстань до об'єкта. Формула розрахунку враховує швидкість звуку в повітрі.

Отримані значення відстаней зберігаються у структурі hcData для подальшої обробки та візуалізації.

Завершення циклу для поточного датчика та перехід до наступного. Змінна індексу i збільшується на одиницю.

На основі зібраних даних формуються повідомлення за допомогою протоколу Mavlink. Це дозволяє передавати дані на блок керування Pixhawk.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

Формовані повідомлення передаються через інтерфейс UART на блок керування Pixhawk для подальшої обробки та прийняття рішень.

Перевіряється, чи перевищує відстань задане максимальне значення. Якщо так, на дисплеї відображається порожнє поле, інакше — числове значення відстані.

На основі зчитаних відстаней визначається рівень загрози. Чим ближче знаходиться перешкода, тим вищий рівень загрози.

Встановлюється кількість поділок для відображення на дисплеї. Ця кількість залежить від дистанції до перешкоди: чим ближче перешкода, тим більше поділок відображається.

На дисплеї візуалізуються рівні загрози у вигляді поділок. Це дозволяє швидко оцінити ситуацію навколо безпілотного транспортного засобу.

Перевіряється стан системи. Якщо система вимкнена, процес завершується. Якщо система продовжує працювати, цикл повертається до кроку 6 для оновлення даних.

Завершення роботи системи. Всі процеси зупиняються, і система переходить у стан очікування.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

3.1 Розробка випробувального стенда

3.1.1 Ініціалізація системи

Ініціалізація системи починається з конфігурації годинника, налаштування GPIO, таймерів та I2C інтерфейсу. Це забезпечує правильну роботу всіх периферійних пристроїв, підключених до мікроконтролера. Функція `SystemClock_Config` налаштовує системний годинник, використовуючи зовнішній кварцовий резонатор. Вона також налаштовує множник і дільник для отримання необхідної частоти системного годинника, що є критичним для точного таймінгу та синхронізації всіх периферійних пристроїв. Спочатку ініціалізуються структури `RCC` для налаштування осцилятора та конфігурації частот. Встановлюються значення множника PLL для отримання потрібної частоти годинника. У разі помилки викликається функція обробки помилок `Error_Handler`. Налаштування годинника мікроконтролера в `CubeMX` зображене на рис. 3.1.

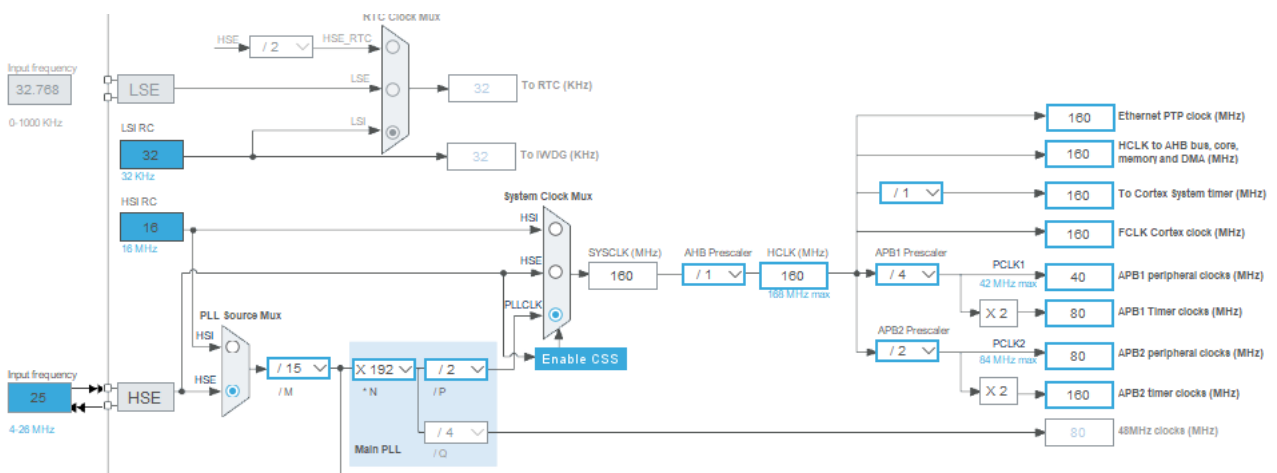


Рисунок 3.1 – Налаштування годинника мікроконтролера в `CubeMX`

						КС КРБ 123.329.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Якобчук Р.А.			Практична частина	Літ.	Арк..	Акрушів
Перевір.		Лецишин Ю.З.					34	
Рецензент						ТНТУ, каф. КС, гр. СІс-42		
Н. контр.		Луцик Н.С.						
Зат. каф.		Осухівська Г.М.						

3.1.2 Налаштування периферійних модулів

Після ініціалізації системного годинника відбувається налаштування периферійних модулів, таких як GPIO (General Purpose Input/Output), таймери та інтерфейс I2C. GPIO порти конфігуруються для роботи з тригерними та ехосигналами ультразвукових датчиків. Це включає в себе налаштування режиму роботи пінів (вхідний або вихідний), швидкість перемикання сигналу та стан підтягування. Таймери налаштовуються для генерації точних часових інтервалів, які необхідні для вимірювання відстаней за допомогою ультразвукових датчиків. Інтерфейс I2C використовується для зв'язку з LCD дисплеєм, що дозволяє виводити інформацію про виміряні відстані. Лістинг функції налаштування GPIO портів зображений на рис. 3.2.

```
static void MX_GPIO_Init(void) {
    GPIO_InitTypeDef GPIO_InitStructure = {0};

    __HAL_RCC_GPIOH_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();

    HAL_GPIO_WritePin(GPIOA,
        Trig1_Pin|Trig2_Pin|Trig3_Pin|Trig4_Pin, GPIO_PIN_RESET);

    GPIO_InitStructure.Pin=
        Trig1_Pin|Trig2_Pin|Trig3_Pin|Trig4_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);

    GPIO_InitStructure.Pin = Eho1_Pin|Eho2_Pin|Eho3_Pin|Eho4_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);
}
```

Рисунок 3.2 – Лістинг функції налаштування GPIO портів

У цьому коді налаштовуються GPIO порти для тригерних і ехосигналів ультразвукових датчиків. Функція HAL_GPIO_WritePin встановлює початковий

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

стан для тригерних сигналів, а структура GPIO_InitTypeDef визначає параметри пінів, такі як режим, швидкість та стан підтягування.

Таймери є ключовими компонентами для забезпечення точних вимірювань часу в системі. Вони використовуються для обчислення часу польоту ультразвукового сигналу, що дозволяє визначити відстань до об'єкта. Таймери налаштовуються на використання внутрішнього годинникового джерела і працюють у режимі лічильника з визначеним періодом. Функція обробки події переповнення таймера викликається кожен раз, коли таймер досягає кінця свого періоду. Ця функція використовується для фіксації часу, коли сигнал від ультразвукового датчика надходить на пін мікроконтролера. Лістинг функції налаштування таймеру TIM8 зображений на рис. 3.3.

```
static void MX_TIM8_Init(void) {
    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};

    htim8.Instance = TIM8;
    htim8.Init.Prescaler = 72-1;
    htim8.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim8.Init.Period = 0xffff-1;
    htim8.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim8.Init.RepetitionCounter = 0;
    htim8.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_Base_Init(&htim8) != HAL_OK) {
        Error_Handler();
    }
    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim8, &sClockSourceConfig)
    != HAL_OK) {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode=
    TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim8,
    &sMasterConfig) != HAL_OK) {
        Error_Handler();
    }
}
```

Рисунок 3.3 – Лістинг функції налаштування таймеру TIM8

Цей код налаштовує таймер TIM8 для використання його в якості джерела часових інтервалів. Налаштовуються параметри переддільника, періоду та

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

режиму роботи таймера. Функція HAL_TIM_Base_Init ініціалізує базовий таймер з вказаними параметрами.

Ультразвукові датчики використовуються для вимірювання відстані до об'єктів. Алгоритм роботи включає в себе ініціалізацію тригерного сигналу, який запускає ультразвуковий імпульс. Після цього відбувається очікування відгуку сигналу, який повертається від об'єкта. Час польоту сигналу вимірюється за допомогою таймерів і використовується для обчислення відстані. Цей алгоритм дозволяє отримати точні вимірювання відстані, які можуть бути використані для навігації та уникнення перешкод. Лістинг алгоритму вимірювання відстані за допомогою ультразвукового датчика зображено на рис. 3.4.

```
int HC_SR04_ReadDistance(GPIO_TypeDef* TRIG_PORT, uint16_t
TRIG_PIN, GPIO_TypeDef* ECHO_PORT, uint16_t ECHO_PIN) {
    HAL_GPIO_WritePin(TRIG_PORT, TRIG_PIN, GPIO_PIN_SET);
    HAL_TIM_SET_COUNTER(&htim8, 0);
    while (HAL_TIM_GET_COUNTER(&htim8) < 10);
    HAL_GPIO_WritePin(TRIG_PORT, TRIG_PIN, GPIO_PIN_RESET);

    uint32_t pMillis = HAL_GetTick();
    while (!(HAL_GPIO_ReadPin(ECHO_PORT, ECHO_PIN)) && pMillis
+ 10 > HAL_GetTick());
    uint32_t Value1 = __HAL_TIM_GET_COUNTER(&htim8);

    pMillis = HAL_GetTick();
    while ((HAL_GPIO_ReadPin(ECHO_PORT, ECHO_PIN)) && pMillis
+ 50 > HAL_GetTick());
    uint32_t Value2 = __HAL_TIM_GET_COUNTER(&htim8);

    return (Value2 - Value1) * 0.034 / 2;
}
```

Рисунок 3.4 – Лістинг алгоритму вимірювання відстані за допомогою ультразвукового датчика

У цьому коді реалізовано алгоритм вимірювання відстані за допомогою ультразвукового датчика. Спочатку встановлюється тригерний сигнал, потім очікується відгук від датчика. Після отримання відгуку обчислюється час польоту сигналу, який потім використовується для обчислення відстані.

LCD дисплей використовується для виведення інформації про виміряні відстані. Ініціалізація включає в себе відправку команд для налаштування

режиму роботи дисплея, таких як включення дисплея, встановлення курсора та очищення екрана. Функція відправки рядка тексту на дисплей використовує цикл для відправки кожного символу окремо. Це дозволяє виводити інформацію в реальному часі, що є важливим для моніторингу та управління системою. Лістинг функції ініціалізації дисплею зображено на рис 3.5.

```
void lcd_init(void) {  
    // Ініціалізація дисплею  
    HAL_Delay(50);  
    lcd_send_cmd(0x30);  
    HAL_Delay(5);  
    lcd_send_cmd(0x30);  
    HAL_Delay(1);  
    lcd_send_cmd(0x30);  
    lcd_send_cmd(0x20);  
    HAL_Delay(10);  
  
    lcd_send_cmd(0x28);  
    lcd_send_cmd(0x08);  
    lcd_send_cmd(0x01);  
    HAL_Delay(10);  
    lcd_send_cmd(0x06);  
    lcd_send_cmd(0x0C);  
}
```

Рисунок 3.5 – Лістинг функції ініціалізації дисплею

Функція `lcd_init` відповідає за початкову ініціалізацію дисплею. Вона відправляє необхідні команди для налаштування режиму роботи дисплея.

3.1.3 Принцип передачі даних за допомогою протоколу Mavlink

Протокол Mavlink (Micro Air Vehicle Link) є легким і ефективним комунікаційним протоколом, який широко використовується в системах безпілотних літальних апаратів (БПЛА) для обміну даними між різними компонентами системи. Його основною метою є забезпечення надійної передачі даних з мінімальною затримкою і високою стійкістю до помилок.

Mavlink є двійковим протоколом, що дозволяє ефективно передавати дані з мінімальним навантаженням на пропускну здатність каналу зв'язку. Він

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

підтримує різні типи повідомлень, що дозволяє передавати команди керування, телеметричні дані, інформацію про стан системи та інші типи даних.

Основою протоколу Mavlink є пакет, що складається з кількох полів, кожне з яких має своє призначення. Ці поля включають префікс синхронізації, довжину корисного навантаження, ідентифікатор повідомлення, дані корисного навантаження, контрольну суму та інші службові поля. Така структура пакета дозволяє легко обробляти дані і забезпечує високу надійність передачі.

Процес передачі даних за допомогою Mavlink включає кілька ключових етапів. Спочатку дані, які необхідно передати, формуються у вигляді повідомлення. Це повідомлення потім інкапсулюється у пакет протоколу Mavlink, який включає всі необхідні службові поля для забезпечення коректної передачі. Після формування пакета він передається через комунікаційний канал, який може бути як дротовим, так і бездротовим.

Однією з основних переваг протоколу Mavlink є його здатність забезпечувати передачу даних з високою стійкістю до помилок. Це досягається за рахунок використання контрольної суми, яка дозволяє виявляти помилки, що виникають під час передачі даних. Якщо контрольна сума не співпадає з розрахованою, пакет вважається пошкодженим і може бути повторно переданий.

Mavlink підтримує різні типи повідомлень, що дозволяє передавати різноманітну інформацію між компонентами системи. Наприклад, для передачі телеметричних даних використовуються спеціалізовані повідомлення, що включають інформацію про поточний стан безпілотного апарату, його положення, швидкість, висоту та інші параметри. Для передачі команд керування використовуються інші типи повідомлень, що включають інформацію про необхідні дії, які повинні виконати компоненти системи.

У контексті дипломного проекту, де мікроконтролер STM32 взаємодіє з блоком керування Pixhawk, протокол Mavlink використовується для передачі оброблених даних з мікроконтролера на блок керування. Це дозволяє забезпечити синхронізацію дій між різними компонентами системи і передавати необхідні команди для виконання маневрів паркування. Наприклад,

					КС КРБ 123.329.00.00 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

мікроконтролер може передавати дані про відстань до перешкод, які зібрані з датчиків відстані, на блок керування Pixhawk для прийняття рішень про корекцію траєкторії руху.

Реалізація протоколу Mavlink на мікроконтролері STM32 включає розробку програмного забезпечення, яке формує і відправляє пакети протоколу Mavlink, а також обробляє отримані пакети. Це забезпечує ефективну і надійну передачу даних між компонентами системи, що є критично важливим для коректної роботи системи орієнтування і паркування безпілотних транспортних засобів.

Лістинг функції для формування та відправки повідомлення протоколу Mavlink зображено на рис. 3.6.

```
void send_mavlink_data(int distance) {
    mavlink_message_t msg;
    uint8_t buf[MAVLINK_MAX_PACKET_LEN];

    mavlink_msg_distance_sensor_pack(1, 200, &msg,
    HAL_GetTick(), 0, 0, distance, 0, 500, 10,
    MAV_DISTANCE_SENSOR_ULTRASOUND, 50, MAV_SENSOR_ROTATION_NONE, 0);
    uint16_t len = mavlink_msg_to_send_buffer(buf, &msg);

    HAL_UART_Transmit(&huart1, buf, len, 100);
}
```

Рисунок 3.6 – Лістинг функції для формування та відправки повідомлення протоколу Mavlink

У цьому коді реалізовано функцію `send_mavlink_data`, яка формує та відправляє повідомлення протоколу Mavlink з даними про відстань. Функція `mavlink_msg_distance_sensor_pack` запаковує дані у формат Mavlink, а `HAL_UART_Transmit` відправляє їх через UART.

Лістинг коду для налаштування UART зображено на рисунку 3.7.

```

static void MX_USART1_UART_Init(void) {
    huart1.Instance = USART1;
    huart1.Init.BaudRate = 57600;
    huart1.Init.WordLength = UART_WORDLENGTH_8B;
    huart1.Init.StopBits = UART_STOPBITS_1;
    huart1.Init.Parity = UART_PARITY_NONE;
    huart1.Init.Mode = UART_MODE_TX_RX;
    huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart1.Init.OverSampling = UART_OVERSAMPLING_16;
    if (HAL_UART_Init(&huart1) != HAL_OK) {
        Error_Handler();
    }
}

```

Рисунок 3.7 – Лістинг коду для налаштування UART

Функція `MX_USART1_UART_Init` налаштовує UART для передачі даних на блок керування Pixhawk. Встановлюються параметри, такі як швидкість передачі даних, довжина слова, кількість стоп-бітів і режим роботи.

Таким чином, протокол Mavlink є потужним інструментом для забезпечення надійної і ефективної передачі даних в системах безпілотних транспортних засобів. Його використання дозволяє забезпечити високу швидкість передачі, стійкість до помилок і гнучкість у передачі різних типів даних, що робить його ідеальним вибором для нашого проекту.

3.2 Тестування алгоритмів комп'ютеризованої системи орієнтування і паркування для БТЗ

3.2.1 Розробка тестових сценаріїв для апаратних компонентів

Тестування апаратних компонентів системи є критично важливим етапом для забезпечення їхньої коректної роботи та інтеграції в загальну систему. Перш за все, необхідно розробити тестові сценарії для кожного з компонентів, таких як датчики HC-SR04, дисплей і блок керування Pixhawk.

Тестування дисплея полягає у перевірці коректного відображення даних з мікроконтролера. Сценарії можуть включати перевірку всіх можливих комбінацій відображення інформації, таких як відстань до перешкоди та рівень

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

заповнення індикатора. Важливо також перевірити роботу дисплея при різних освітлених умовах і кутах огляду.

На рис. 3.8 зображено тестування дисплея.



Рисунок 3.8 – Тестування дисплея

Для датчиків тестові сценарії включають перевірку точності вимірювання відстані в різних умовах. Наприклад, можна розмістити перешкоди на різних відстанях і виміряти час відгуку датчика, а також перевірити стабільність вимірювань. Це дозволить виявити можливі похибки та налаштувати параметри системи для їх компенсації. Також в прототипі було використано з кафдери компютерних системи та мереж

На рис. 3.8 зображено тестування датчика HC-SR04.

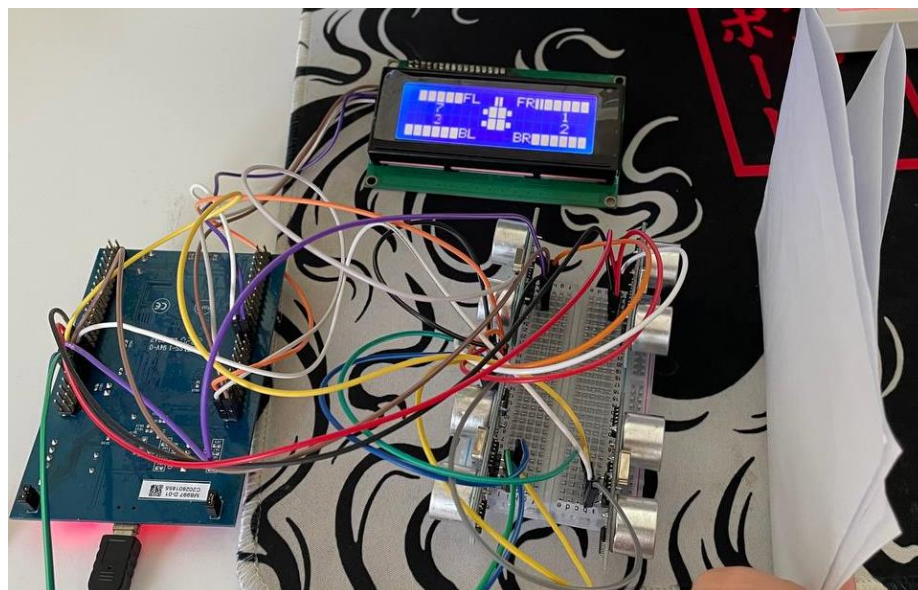


Рисунок 3.9 – Тестування датчиків відстані

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

Для блоку керування Pixhawk тестові сценарії включають перевірку коректності прийому даних через протокол Mavlink і відповідну реакцію на ці дані. Це дозволяє впевнитися, що дані про відстань до перешкоди надходять до контролера без затримок і помилок, і що блок керування правильно їх обробляє для подальшого прийняття рішень.

3.2.2 Розробка тестових сценаріїв для програмного забезпечення

Тестування програмного забезпечення включає перевірку всіх модулів, що забезпечують роботу системи. Це важливо для гарантії того, що програми працюють коректно, навіть у випадку нестандартних ситуацій. Тестові сценарії для ПЗ охоплюють кілька основних напрямків.

Перш за все, необхідно перевірити програму збору і обробки даних з датчиків. Це включає тестування коректності обчислення відстані, обробку помилок датчиків, а також швидкість і стабільність роботи програми.

Далі, потрібно протестувати програму відображення даних на дисплеї. Сценарії включають перевірку коректності виведення різних значень, реакцію на зміну вхідних даних і перевірку швидкості оновлення інформації.

Тестування передачі даних через протокол Mavlink включає перевірку правильності формування пакетів даних, стабільність передачі і коректність обробки даних блоком керування Pixhawk.

3.2.3 Тестування роботи системи в умовах реальних задач

Тестування роботи системи в умовах реальних задач є важливим етапом для перевірки її ефективності та надійності. Це включає створення симуляцій, які відтворюють реальні сценарії використання безпілотного апарату.

Наприклад, можна змодельовати ситуацію, коли безпілотний апарат знаходиться в обмеженому просторі з перешкодами. Моделювання включає перевірку коректності збору даних з датчиків, їх обробку і передачу на блок керування Pixhawk. Важливо впевнитися, що система правильно розпізнає перешкоди і приймає відповідні рішення щодо руху апарату.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

Інший сценарій моделювання може включати тестування системи в умовах змінної навколишньої обстановки, наприклад, на відкритій місцевості з раптово виникаючими перешкодами. Це дозволяє перевірити, як система адаптується до нових умов і наскільки швидко реагує на зміни.

3.2.4 Аналіз результатів тестування

Після проведення тестування і моделювання необхідно провести детальний аналіз отриманих результатів. Це дозволяє виявити слабкі місця в системі і внести необхідні корективи для підвищення її надійності та ефективності.

Аналіз результатів тестування апаратних компонентів включає оцінку точності і стабільності вимірювань датчиків, коректність роботи дисплея та взаємодії між мікроконтролером і блоком керування Pixhawk. Важливо визначити, чи відповідають отримані результати очікуваним і чи немає значних відхилень.

Аналіз результатів тестування програмного забезпечення включає перевірку коректності виконання всіх функцій, стабільності роботи програм і обробки помилок. Важливо також оцінити швидкість роботи програм і їх здатність обробляти дані в режимі реального часу.

Аналіз результатів моделювання дозволяє оцінити ефективність системи в умовах реальних задач. Це включає оцінку коректності розпізнавання перешкод, швидкість реакції системи і прийняття рішень щодо руху апарату. Важливо впевнитися, що система працює стабільно і надійно навіть у складних умовах.

Ретельний аналіз результатів дозволяє виявити потенційні проблеми на ранніх етапах і провести необхідні покращення перед остаточним впровадженням системи.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

3.3 Практична імплементація розроблюваної системи в БТЗ

Практична імплементація розробленої системи орієнтування та паркування в безпілотних транспортних засобах є важливим етапом, що вимагає ретельної інтеграції всіх компонентів та налаштування системи для роботи в реальних умовах. Цей процес передбачає встановлення та налаштування апаратного і програмного забезпечення, тестування системи в різних сценаріях та оптимізацію для забезпечення надійної роботи.

Розпочинається імплементація з встановлення мікроконтролера STM32, датчиків відстані HC-SR04 та блоку керування Pixhawk у транспортний засіб. Кожен компонент закріплюється на відповідних місцях, що забезпечує оптимальну роботу системи. Наприклад, датчики відстані встановлюються на передній та задній частинах автомобіля для точного вимірювання відстані до об'єктів під час паркування.

Після фізичної установки компонентів, виконується підключення електричних і комунікаційних з'єднань відповідно до розробленої схеми. Датчики відстані з'єднуються з відповідними входами мікроконтролера, що дозволяє системі отримувати необхідні дані для обробки.

Наступним етапом є налаштування програмного забезпечення. На мікроконтролер завантажується розроблений код, що забезпечує збір, обробку і відображення даних з датчиків, а також передачу цих даних на блок керування Pixhawk за допомогою протоколу Mavlink. Особливу увагу приділяють налаштуванню параметрів обробки даних, таких як фільтрація та усереднення, для забезпечення високої точності вимірювань і надійності системи.

Після цього проводяться випробування системи в реальних умовах. Транспортний засіб тестується в різних сценаріях, включаючи парковку в обмежених просторах, маневрування в тісних умовах і об'їзд перешкод. Під час тестування фіксуються всі можливі проблеми, такі як некоректні вимірювання або затримки в обробці даних. Ці проблеми аналізуються і виправляються шляхом налаштування апаратного і програмного забезпечення.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

Одним з найважливіших аспектів є оптимізація енергоспоживання системи. Оскільки безпілотний транспортний засіб залежить від автономного живлення, важливо мінімізувати споживання енергії кожним компонентом системи. Для цього використовуються різні методи, такі як зниження тактової частоти мікроконтролера у режимі очікування та вимикання непотрібних компонентів під час простою.

Завершальним етапом є документування процесу імплементації та результатів тестування. Цей етап є надзвичайно важливим, оскільки дозволяє зберегти всю інформацію, зібрану під час розробки і тестування, для подальшого використання і аналізу. Документування включає створення детальних інструкцій для налаштування і використання системи, що забезпечує можливість легкого та безпомилкового впровадження системи новими користувачами або технічним персоналом. Ці інструкції повинні містити крок за кроком опис всіх процедур налаштування, з вказівками щодо вирішення можливих проблем, що можуть виникнути під час експлуатації.

Крім того, необхідно підготувати звіти про виявлені проблеми та їх вирішення. Ці звіти повинні містити детальну інформацію про всі виявлені несправності, їхній вплив на роботу системи, а також опис вжитих заходів для їх усунення. Така документація є важливою для забезпечення надійності і стабільності системи в майбутньому, оскільки дозволяє уникнути повторення тих самих помилок і знижує ризик виникнення нових проблем.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Питання безпеки при експлуатації електроустановок високих, надвисоких та ультрависоких частот

Електромагнітне поле (ЕМП) – особлива форма матерії, за допомогою якої здійснюється взаємодія між електрично зарядженими частинками. Воно складається з двох окремих полів – електричного та магнітного. Силові лінії цих полів взаємно перпендикулярні. Через електромагнітне поле передаються всі види електромагнітного випромінювання – від низькочастотного (радіохвилі) до високочастотного (рентгенівське та гамма-випромінювання).

Електромагнітне поле у просторі поширюється у вигляді електромагнітної хвилі, яка переносить енергію, замкнену в електричному та магнітному полях. Електричні та магнітні поля змінюються одночасно одне з одним. При цьому співвідношення між їх миттєвими значеннями завжди залишаються сталими. Лише на близьких від джерела відстанях, у так званій зоні несформованого поля, ця закономірність порушується.

Основними фізичними параметрами електромагнітного поля є швидкість поширення електромагнітної хвилі, довжина хвилі та частота коливань, які зв'язані між собою співвідношенням. Спектр електромагнітних коливань радіочастот за частотою коливань та довжиною хвилі умовно поділяють на діапазони. За частотою коливань електромагнітні хвилі мають діапазони низьких (НЧ), середніх (СЧ), високих (ВЧ), дуже високих (ДВЧ), ультрависоких (УВЧ), надвисоких (НВЧ) та надзвичайно високих частот (НЗВЧ). Розрізняють кілометрові, гектометрові, декаметрові, метрові, дециметрові діапазони хвиль.

Джерелами електромагнітного випромінювання у виробничому приміщенні бувають неекрановані робочі елементи високочастотних установок

					КС КРБ 123.329.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Якобчук Р.А.			Безпека життєдіяльності, основи охорони праці	Літ.	Арк..	Акрушів
Перевір.		Лецишин Ю.З.					47	
Консульт.		Пилипець М. І.				ТНТУ, каф. КС, гр. СІс-42		
Н. контр.		Луцик Н.С.						
Зав. каф.		Осухівська Г.М.						

(індуктори, конденсатори, ВЧ-трансформатори, фідерні лінії, батареї конденсаторів, котушки коливальних контурів). Під час експлуатації ВЧ-, ДВЧ-, УВЧ-передавачів на радіо- та телецентрах джерелами електромагнітного випромінювання є високочастотні генератори, антенні комутатори, пристрої складання потужностей електромагнітного поля, комунікації (від генератора до антенного пристрою), антени.

Ступінь опромінення працівників залежить від кількості передатчиків (у деяких зонах, радіо- та телецентрах їх може бути до 20), їх потужності, екранування, розміщення окремих їх блоків усередині та поза приміщенням.

Електромагнітні поля особливо негативно впливають на організм людини, яка безпосередньо працює з джерелом випромінювання. В діапазоні промислових частот більше негативний вплив на біологічний об'єкт має електрична складова поля.

Найчутливішими до ЕМП є нейродинамічні процеси, які прямо чи побічно перемикають хронобіологічні процеси організму на патологічний або стресовий режим функціонування.

При дії ЕМП на людину можливі гострі та хронічні форми порушення фізіологічних функцій організму. Такі порушення виникають в результаті дії електричної складової ЕМП на нервову систему, а також на структуру кори головного та спинного мозку, серцево-судинної системи.

У більшості випадків такі зміни в діяльності нервової та серцево-судинної системи мають зворотній характер, але в результаті тривалої дії вони накопичуються, підсилюються з плином часу, але, як правило, зменшуються та зникають при виключенні впливу та поліпшенні умов праці. Тривалий та інтенсивний вплив ЕМП призводить до стійких порушень в організмі людини та захворювань.

Сумісна дія випромінювань широкого діапазону може викликати окрему радіохвильову хворобу.

Тяжкість її наслідків прямо залежить від напруженості ЕМП, фізичних особливостей різних діапазонів частот, тривалості впливу, умов навколишнього

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

середовища, а також від функціонального стану та стійкості організму до впливу різних чинників, можливостей адаптації. Збільшується ризик виникнення загальних захворювань, захворювань органів дихання, травлення тощо. Це може відбуватися також і за дуже невеликої інтенсивності ЕМП, яка незначно перевищує гігієнічні нормативи.

Виникає ряд симптомів, які свідчать про порушення роботи окремих органів – шлунку, печінки, підшлункової залози. Погіршуються харчові та статеві рефлекси, діяльність серцево-судинної системи, фіксуються зміни показників білкового та вуглеводного обміну, змінюється склад крові, зафіксовані зміни на рівні клітин. Систематична дія ЕМП високої та надвисокої частоти на організм людини викликає підвищення кров'яного тиску, трофічні явища (випадіння волосся, ламкість нігтів).

Оператори безпілотних транспортних засобів повинні бути ознайомленими з правилами експлуатації електроустановок високих, надвисоких та ультрависоких частот.

4.2 Заходи, що покращують умови праці оператора

Людина значну частину свого життя витрачає на працю і тому від умов праці, рівня її безпеки залежить працездатність і продуктивність праці, якість роботи, стан здоров'я. Оператори безпілотних транспортних засобів повинні знати певні заходи, що покращують умови праці.

Сприятливі умови забезпечують як соціальну гармонію особи людини, так і ставлення її до праці та задоволення працею. Актуальність питання поліпшення умов праці обумовлюється і тим, що рівень освіти працівників висуває на перше місце необхідність задоволення потреб у змістовній праці в небезпечних умовах. Тому створення сприятливих умов праці має бути одним із головних завдань суспільства, невід'ємною частиною державної соціальної і економічної політики, важливою складовою менеджменту персоналу.

					КС КРБ 123.329.00.00 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

На підприємствах трудовим колективам надані великі повноваження щодо поліпшення умов праці. Вони мають право брати участь в обговоренні та затвердженні комплексних планів інженерно-технічних заходів з досягнення встановлених нормативів безпеки, гігієни праці та виробничого середовища; контролювати використання коштів на виконання планів, ставити питання про притягнення до відповідальності за порушення норм з охорони праці.

В умовах розвитку ринкових відносин поліпшенню умов праці сприяє і необхідність використовувати у конкурентній боротьбі новітні технології, які базуються на досягненнях науково-технічного прогресу, прагнення досягти зниження витрат на виробництво продукції та відповідного зростання прибутку на підприємствах будь-якої форми власності.

Умови праці — це обстановка, в якій здійснюється трудова діяльність, тобто характер устаткування та організації робочих місць, рівень дотримання санітарно-гігієнічних і естетичних норм, психологічний клімат, це сукупність факторів виробничого середовища, що впливають на здоров'я та працездатність людини в процесі праці.

Умови праці поділяються на соціально економічні, які розглядаються у широкому контексті і характеризують відношення до них суспільства, та виробничі, тобто умови праці безпосередньо на робочих місцях.

На формування умов праці впливають фактори, які поділяються на три групи:

Перша група — соціально-економічні фактори, дія яких обумовлює характер умов праці. Серед них виділяються підгрупи:

- нормативно-правові (закони, норми, стандарти а також форми адміністративного та громадського контролю за їх виконанням);
- економічні (матеріальне та економічне стимулювання, моральне заохочення, система пільг та компенсацій за несприятливі умови праці);
- соціально-психологічні (відношення працівників до праці, психологічний клімат);

					КС КРБ 123.329.00.00 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

– суспільно-політичні (форми руху працівників за створення сприятливих умов, винахідництво і раціоналізація).

Друга група — техніко-організаційні фактори. Вони впливають на формування умов праці на робочих місцях, дільницях, цехах. Серед них виділяються такі підгрупи:

- предмети праці та їх продукти (сировина, матеріали, готові вироби);
- технологічні процеси;
- засоби праці;
- організаційні форми виробництва, праці та управління.

Третя група — природні фактори, дія яких не лише обумовлюється особливостями природного середовища, а й висуває додаткові вимоги до устаткування, технології, організації виробництва та праці. Серед них виділяються такі підгрупи:

- географічні (кліматичні зони);
- біологічні (особливості рослинного та тваринного світу);
- геологічні (характер добування корисних копалин).

Усі ці фактори впливають на формування умов праці одночасно і у нерозривній єдності, обумовлюючи поряд з іншими параметрами виробниче середовище. Класифікація факторів допомагає на рівні галузі, об'єднання, окремого виробництва:

- формувати та поліпшувати умови праці, аналізувати їх стан;
- планувати заходи щодо поліпшення умов праці;
- розробляти проекти устаткування, споруджень, технологічних процесів, спрямованих на поліпшення умов праці;
- зосереджувати ресурси (фінансові, матеріальні, трудові) на поліпшення умов праці;
- прогнозувати зміни в умовах праці у зв'язку зі змінами технології, устаткування, впровадження нових матеріалів і технологій.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

Свій вплив на людину система факторів спричиняє опосередковано через сукупність системи елементів, які безпосередньо визначають умови праці на робочих місцях.

Виділяються такі елементи умов праці:

а) санітарно-гігієнічні, що характеризують виробниче середовище, на яке впливають предмети та засоби праці, а також технологічні процеси (промисловий шум, вібрація, токсичні речовини, промисловий пил, температура повітря);

б) психофізіологічні елементи обумовлені змістом праці та її організацією (фізичне навантаження, нервово-психологічна напруга, монотонність трудового процесу). Елементи цієї групи, за винятком фізичних зусиль і монотонності, не мають затверджених нормативів;

в) естетичні елементи сприяють формуванню позитивних емоцій у працівника (художньо-конструктивне рішення робочого місця, освітлення, функціональна музика). Кількісних оцінок елементи цієї групи не мають. Визначення естетичного рівня умов праці здійснюється за допомогою методів експертного оцінювання;

г) соціально-психологічні елементи характеризують взаємовідносини у трудовому колективі, створюючи відповідний психологічний настрій працюючих (соціальний клімат). Вони не мають одиниць виміру, норм і стандартів. Але соціологічні дослідження у вигляді усного опитування, анкетування сприяють їх об'єктивному оцінюванню;

г) технічні елементи визначаються рівнем механізації праці. Праця, а відповідно умови й охорона праці, є реаліями різних систем, з одного боку, «людина — машина (технологічний процес)», «людина — виробниче середовище», «людина — машина (технологічний процес) — виробниче середовище», а з іншого «людина — колектив — суспільство», — «людина — суспільство — природа».

Для перших трьох систем умови праці розглядаються у межах робочого місця, а для останніх двох — у межах підприємства, галузі, регіону.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено комп'ютеризовану систему орієнтування і паркування безпілотних транспортних засобів з передачею інформації на блок керування Pixhawk та виводом інформації на дисплей.

У першому розділі було проведено детальний аналіз аналогів існуючих систем. Встановлено основні вимоги до системи, визначено її функціональні можливості та технічні характеристики. Проведено огляд можливих рішень для кожного з компонентів системи, що дозволило вибрати найбільш оптимальні варіанти для реалізації проекту. Аналіз застосування аналогічних систем підтвердив важливість і актуальність даної тематики у сучасних умовах розвитку безпілотних технологій.

Другий розділ був присвячений розробці узагальненої структури комп'ютеризованої системи. Було створено структурну схему блок схему алгоритму системи, яка включає збір даних з ультразвукових датчиків відстані, обробку цих даних на мікроконтролері STM32, виведення даних на LCD дисплей, а також передачу інформації на блок керування Pixhawk. Розробка схеми з'єднань і електричної принципової схеми забезпечила надійність та ефективність взаємодії між усіма компонентами системи.

У третьому розділі описано процес реалізації та моделювання проектних рішень. Детально розглянуто алгоритм роботи системи, принцип передачі даних за допомогою протоколу Mavlink, а також практичну імплементацію розробленої системи. Тестування показало, що система здатна точно вимірювати відстані до перешкод та своєчасно реагувати на зміни у навколишньому середовищі, забезпечуючи безпечне та ефективне паркування.

У четвертому розділі розглянуті питання з основ безпеки життєдіяльності та охорони праці.

Таким чином, дипломний проект показав можливість створення ефективної та надійної системи, що відповідає сучасним вимогам та стандартам.

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Протокол Mavlink: основи та застосування в БПЛА URL: <https://drone.ua/mavlink-protocol-overview/> (дата звернення 10.05.2024).
2. Інтеграція систем управління БПЛА на базі Pixhawk URL: <https://avia.com.ua/integrating-pixhawk-uas/> (дата звернення 15.05.2024).
3. STM32: Огляд та можливості для розробки БПЛА. URL: <https://embedded-ukraine.com/stm32-for-uas/> (дата звернення 17.05.2024).
4. HC-SR04 ультразвуковий датчик відстані URL: <https://myproject.com.ua/hc-sr04-ultrazvukovij-datchik-vidstani.html> (дата звернення 15.05.2024).
5. Вибір мікроконтролера для безпілотних систем URL: <https://mcu.org.ua/uas-microcontroller> (дата звернення 10.05.2024).
6. Переваги використання STM32 в безпілотних системах URL: https://www.nai.au.kiev.ua/files/metod_bpla.pdf (дата звернення 18.05.2024).
7. Використання I2C для підключення дисплеїв до мікроконтролерів URL: <https://qazf.com.ua/lcd1602-i2c-stm32/> (дата звернення 24.05.2024).
8. Виявлення руху за допомогою ультразвукового датчика присутності URL: <http://mikrotik.kpi.ua/index.php/courses-list/> (дата звернення 05.05.2024).
9. Автоматизована система паркування URL: https://en.wikipedia.org/wiki/parking_system (дата звернення 05.05.2024).
10. Використання технологій безпеки на робочому місці URL: <https://www.bezpeka-shop.com/ua/blog/poleznyye-sovety/> (дата звернення 11.06.2024).
11. Негативний вплив електромагнітних полів на людину URL: <https://oppb.com.ua/articles/negatyvnyy-vplyv-elektromagnitnyh-poliv-na-lyudynu> (дата звернення 11.06.2024).
12. Енергозбереження в безпілотних транспортних засобах URL: <https://nubip.edu.ua/node/84635> (дата звернення 15.05.2024).
13. MAVLink "Micro Air Vehicle Communication Protocol" URL: <https://en.wikipedia.org/wiki/MAVLink> (дата звернення 09.06.2024).

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

14. Using LCD Displays with Arduino URL: <https://forum.dronebotworkshop.com/2018/using-lcd-displays-with-arduino/> (дата звернення 10.05.2024).
15. Лецишин Ю.З. / Лецишин Юрій Зіновійович , 2016 URL: <http://elartu.tntu.edu.ua/handle/lib/21415> (дата звернення 11.06.2024).
16. Лецишин Ю.З. Розробка системи зв'язку як інтегрованого елементу роботизованих систем / Лецишин Юрій Зіновійович, Романишин Нестор Романович, Наконечний Володимир Васильович, Паламарчук Антон Олегович , 2016 URL: <http://elartu.tntu.edu.ua/handle/123456789/17307> (дата звернення 18.05.2024).
17. Лецишин Ю. З. Створення вбудованих систем на базі структурнопараметричних моделей цифрових каналів зв'язку : Лецишин Ю.З., Назаревич Т.О., Міська І.В. VIII Науково-технічна конференція «Інформаційні моделі, системи та технології». Тернопіль, 2020. С. 127.
18. Лецишин Ю. З. Розробка системи зв'язку як інтегрованого елементу роботизованих систем. Проблеми створення, розвитку та застосування високотехнологічних систем спеціального призначення з урахуванням досвіду антитерористичної операції. Збірник тез доповідей XXI Всеукраїнської науковопрактичної конференції. Житомир, 2016. С. 102.
19. Осухівська Г.М., Тиш Є.В., Луцик Н.С., Паламар А.М. Методичні вказівки до виконання кваліфікаційних робіт здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль: ТНТУ, 2022. 28 с.
20. Лецишин Ю. З. Побудова і моделювання цифрового узгодженого фільтру для виявлення складних сигналів / Ю. З. Лецишин, В. О. Ворощак // Збірник тез доповідей VI Міжнародної науково-технічної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“, 16-17 листопада 2017 року. — Т. : ТНТУ, 2017. — Том 2. — С. 203. — (Комп'ютерно-інформаційні технології та системи зв'язку).

					КС КРБ 123.329.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

Додаток А.
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

“Затверджую”

Завідувач кафедри КС _____ Осухівська Г.М.

“ ____ ” _____ 2024 р.

КОМП'ЮТЕРИЗОВАНА СИСТЕМА ОРІЄНТУВАННЯ І ПАРКУВАННЯ
БЕЗПЛОТНИХ ТРАНСПОРТНИХ ЗАСОБІВ

ТЕХНІЧНЕ ЗАВДАННЯ на __ листках

На здобуття освітньо-кваліфікаційного рівня бакалавр

Напрямок 123 Комп'ютерна інженерія

Спеціальність 123 Комп'ютерна інженерія

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

_____ к.т.н., доцент

Лецишин Ю.З.

“ ____ ” _____ 2024 р.

«ВИКОНАВЕЦЬ»

Студент групи СІс-42

_____ Якобчук Р.А.

“ ____ ” _____ 2024 р.

Тернопіль 2024

1. Назва та підстава для виконання роботи.

1.1. Комп'ютеризована система орієнтування і паркування безпілотних транспортних засобів.

1.2. Підставою для виконання кваліфікаційної роботи бакалавра (КРБ) є Наказ по Університету (№ 4/7-468 від 26.04.2024 р.).

2. Виконавець.

2.1. Студент групи СІс-42 кафедри КС Тернопільського національного технічного університету ім. І. Пулюя, Якобчук Роман Анатолійович.

3. Мета роботи.

3.1. Метою роботи є розробити структуру та апаратне забезпечення комп'ютеризованої системи дистанційного керування кухнею.

4. Склад виробу.

4.1. До складу виробу повинні входити:

- мікроконтролер STM32F407VGT6;
- чотири датчики відстані HC-SR04;
- LCD дисплей 16x4;
- блок керування Pi4hawk.

5. Технічні вимоги.

5.1. Вимоги по призначенню.

5.1.1. Вбудована система повинна мати наступні параметри:

- охоплення відстані 1м;
- точність вимірювання 1см;
- висока частота сигналу;

5.1.2. Система повинна живитись напругою постійного струму 5 В через USB порт та/або 12 В.

5.2. Вимоги до умов експлуатації.

5.2.1. По умовам експлуатації виріб повинен відповідати наступним вимогам:

- робочий температурний діапазон: від -20°C до +85°C;
- відносна вологість: до 95% без конденсації;
- витримка впливу електромагнітних перешкод.

5.3. Конструктивні вимоги.

5.3.1. Конструювання корпусу приладу в КРБ не передбачено:

- розміри плати: не більше 2000 мм x 1000 мм;
- надійне кріплення всіх компонентів;
- зручний доступ до роз'ємів USB та кнопки управління.

5.3.2. Для побудови системи має бути використані сучасні компоненти.

5.3.3. При побудові системи необхідно передбачити наявність роз'ємів живлення і обміну даними.

5.4. Вимоги до надійності.

5.4.1. Система повинна відповідати вимогам ДСТУ 2862-94.

5.4.2. Наробка на відмову, не менше 30000 год.

5.5. Вимоги метрології.

5.5.1. Вимірювання параметрів системи при моделюванні повинно виконуватись на універсальних вимірювальних приладах.

6. Економічні показники.

6.1. Собівартість системи повинна бути не більше 2500 грн.

7. Вимоги до документації.

7.1. Конструкторська документація повинна відповідати вимогам ЄСКД та ДСТУ.

7.2. До складу документації повинно входити:

- ПЗ;
- функціональна схема;

- електрично принципова схема;
- структурна схема;
- блок схема алгоритму роботи КС.

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

8. Стадії та етапи розробки КРБ

8.1 Стадії та етапи виконання КРБ наведенні в таблиці 1.

Таблиця 1

№ з/п	Назва етапів роботи	Термін виконання етапів роботи
1	<i>Розробка і затвердження технічного завдання</i>	
2	<i>Аналіз технічного завдання</i>	
3	<i>Аналіз вимог та принципів організації комп'ютеризованої системи орієнтування і паркування безпілотних транспортних засобів</i>	
4	<i>Проектування комп'ютеризованої системи орієнтування і паркування безпілотних транспортних засобів</i>	
5	<i>Розробка схем і програмного забезпечення комп'ютеризованої системи орієнтування і паркування безпілотних транспортних засобів</i>	
6	<i>Розробка інструкцій з використання системи</i>	
7	<i>Безпека життєдіяльності, основи охорони праці</i>	
8	<i>Оформлення кваліфікаційної роботи</i>	
9	<i>Попередній захист кваліфікаційної роботи</i>	
10	<i>Захист кваліфікаційної роботи</i>	

9. У дане ТЗ можуть вноситись зміни по узгодженню сторін.

Додаток Б.

Перелік елементів електрично-принципової схеми

<i>Позн.</i>	<i>Найменування</i>	<i>К-сть</i>	<i>Примітка</i>
	<i>Мікросхеми</i>		
<i>U1-U4</i>	<i>HC-SR04</i>	<i>4</i>	
<i>U5</i>	<i>Pixhawk</i>	<i>1</i>	
<i>U6</i>	<i>STM32F407VGT6</i>	<i>1</i>	
<i>U7</i>	<i>I2C converter</i>	<i>1</i>	
<i>U8</i>	<i>LCD display 16x4</i>	<i>1</i>	

					КС КРБ 123.329.00.00 ПЕ						
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Компютеризована система орієнтування і паркування безпілотної транспортних засобів <i>Перелік елементів</i>			<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>	
<i>Розроб.</i>		<i>Якобчук Р.А.</i>								<i>62</i>	<i>1</i>
<i>Перевір.</i>		<i>Лецишин Ю.З.</i>									
<i>Реценз.</i>											
<i>Н. контр.</i>		<i>Луцик Н.С.</i>									
<i>Зав. каф.</i>		<i>Осухівська Г.М.</i>			ТНТУ, каф. КС, гр. СІс-42						

Додаток В.
Лістинг


```

#include "main.h"

#include <stdbool.h>
#include <stdio.h>
#include <LCD1602_config.h>
#include <LCD1602.h>

#define DIST1 5
#define DIST2 10
#define DIST3 15
#define DIST4 25
#define DIST5 30
#define DIST6 50

#define MAX_DIST 99
I2C_HandleTypeDef hi2c1;

TIM_HandleTypeDef htim8;

struct hcData_t hcData = {{0}, {0}, {0}};

uint32_t pMillis = 0;
uint32_t Value1 = 0;
uint32_t Value2 = 0;
int distance = 0;

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_TIM8_Init(void);
static void MX_I2C1_Init(void);
void timerOS(uint16_t time);
void printDef(void);
uint8_t I2C_Scan(I2C_HandleTypeDef *hi2c);
void displayPoints(uint8_t numHc);
void displayPoint(uint8_t numHc, uint8_t state);
void setPoints(void);
void displayDistances(void);
void setDistances(void);

int HC_SR04_ReadDistance(GPIO_TypeDef* TRIG_PORT, uint16_t TRIG_PIN,
GPIO_TypeDef* ECHO_PORT, uint16_t ECHO_PIN) {
    HAL_GPIO_WritePin(TRIG_PORT, TRIG_PIN, GPIO_PIN_SET); // pull
the TRIG pin HIGH
    __HAL_TIM_SET_COUNTER(&htim8, 0);
    while (__HAL_TIM_GET_COUNTER (&htim8) < 10); // wait for 10 us
    HAL_GPIO_WritePin(TRIG_PORT, TRIG_PIN, GPIO_PIN_RESET); //
pull the TRIG pin low

    pMillis = HAL_GetTick(); // used this to avoid infinite while
loop (for timeout)
    // wait for the echo pin to go high
    while (!(HAL_GPIO_ReadPin (ECHO_PORT, ECHO_PIN)) && pMillis +
10 > HAL_GetTick());
    Value1 = __HAL_TIM_GET_COUNTER (&htim8);

```

```

    pMillis = HAL_GetTick(); // used this to avoid infinite while
loop (for timeout)

    while ((HAL_GPIO_ReadPin (ECHO_PORT, ECHO_PIN)) && pMillis + 50
> HAL_GetTick());
    Value2 = __HAL_TIM_GET_COUNTER (&tim8);

    return (int) (1.35*(Value2-Value1)*0.034/2);
}

uint32_t    getStableDistance(GPIO_TypeDef*    TRIG_PORT,    uint16_t
TRIG_PIN, GPIO_TypeDef* ECHO_PORT, uint16_t ECHO_PIN) {
    for (int i = 0; i < 5; i++) {
        distance    +=    HC_SR04_ReadDistance(TRIG_PORT,    TRIG_PIN,
ECHO_PORT, ECHO_PIN);
    }
    return distance / 5;
}

void setDistances(void){
    hcData.dist[HC_FL]    =    HC_SR04_ReadDistance(GPIOA,    Trig1_Pin,
GPIOA, Eho1_Pin);
    hcData.dist[HC_FR]    =    HC_SR04_ReadDistance(GPIOA,    Trig2_Pin,
GPIOA, Eho2_Pin);
    hcData.dist[HC_BL]    =    HC_SR04_ReadDistance(GPIOA,    Trig3_Pin,
GPIOA, Eho3_Pin);
    hcData.dist[HC_BR]    =    HC_SR04_ReadDistance(GPIOA,    Trig4_Pin,
GPIOA, Eho4_Pin);
}

void displayDistances(void) {
    char buffer[6];

    LCD_SetCursor(0,1); // Print HC_FL dist
    if (hcData.dist[HC_FL] > MAX_DIST) {
        snprintf(buffer, sizeof(buffer), "    ");
        LCD_PrintString(buffer);
    } else {
        snprintf(buffer, sizeof(buffer), " %3d ", hcData.dist[HC_FL]);
        LCD_PrintString(buffer);
    }

    LCD_SetCursor(14,1); // Print HC_FR dist
    if (hcData.dist[HC_FR] > MAX_DIST){
        snprintf(buffer, sizeof(buffer), "    ");
        LCD_PrintString(buffer);
    }
    else {
        snprintf(buffer, sizeof(buffer), " %3d ", hcData.dist[HC_FR]);
        LCD_PrintString(buffer);
    }
}

```

```

LCD_SetCursor(0,2); // Print HC_BL dist
if (hcData.dist[HC_BL] > MAX_DIST){
    snprintf(buffer, sizeof(buffer), "      ");
    LCD_PrintString(buffer);
}
else {
    snprintf(buffer, sizeof(buffer), " %3d ", hcData.dist[HC_BL]);
    LCD_PrintString(buffer);
}

LCD_SetCursor(14,2); // Print HC_BR dist
if (hcData.dist[HC_BR] > MAX_DIST){
    snprintf(buffer, sizeof(buffer), "      ");
    LCD_PrintString(buffer);
}
else {
    snprintf(buffer, sizeof(buffer), " %3d ", hcData.dist[HC_BR]);
    LCD_PrintString(buffer);
}

    setPoints();
}

int main(void)
{
9  HAL_Init();

    SystemClock_Config();

    MX_GPIO_Init();
    MX_TIM8_Init();
    MX_I2C1_Init();
    /* USER CODE BEGIN 2 */
    HAL_TIM_Base_Start(&htim8);

    HAL_Delay(100);

    uint8_t myChar_0[SIZE_USER_CHAR] = { 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00 }; // clear
    uint8_t myChar_1[SIZE_USER_CHAR] = { 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff }; // fill
    uint8_t myChar_2[SIZE_USER_CHAR] = { 0x00, 0x00, 0x07, 0x07, 0x07,
0x07, 0x00, 0x00 }; // wheel R
    uint8_t myChar_3[SIZE_USER_CHAR] = { 0x00, 0x00, 0x1c, 0x1c, 0x1c,
0x1c, 0x00, 0x00 }; // wheel L
    uint8_t myChar_4[SIZE_USER_CHAR] = { 0x00, 0x03, 0x03, 0x03, 0x03,
0x03, 0x03, 0x03 }; // wheel R
    uint8_t myChar_5[SIZE_USER_CHAR] = { 0x00, 0x18, 0x18, 0x18, 0x18,
0x18, 0x18, 0x18 }; // wheel L
    LCD_Init();

    LCD_CreateChar(myChar_0, LCD_CLEAR);
    LCD_CreateChar(myChar_1, LCD_FILL);
    LCD_CreateChar(myChar_2, LCD_WHEEL_R);

```

```

LCD_CreateChar(myChar_3, LCD_WEEL_L);
LCD_CreateChar(myChar_4, LCD_MUZ_R);
LCD_CreateChar(myChar_5, LCD_MUZ_L);

printDef();

HAL_Delay(500);

    setDistances();
    displayDistances();
}
}

void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    __HAL_RCC_PWR_CLK_ENABLE();
    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitStruct.PLL.PLLM = 15;
    RCC_OscInitStruct.PLL.PLLN = 192;
    RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
    RCC_OscInitStruct.PLL.PLLQ = 4;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }

    RCC_ClkInitStruct.ClockType =
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;

    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) !=
HAL_OK)
    {
        Error_Handler();
    }
}

static void MX_I2C1_Init(void)
{
    hi2c1.Instance = I2C1;

```

```

hi2c1.Init.ClockSpeed = 100000;
hi2c1.Init.DutyCycle = I2C_DUTYCYCLE_2;
hi2c1.Init.OwnAddress1 = 0;
hi2c1.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
hi2c1.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
hi2c1.Init.OwnAddress2 = 0;
hi2c1.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
hi2c1.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
if (HAL_I2C_Init(&hi2c1) != HAL_OK)
{
    Error_Handler();
}

}

static void MX_TIM8_Init(void)
{

    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};
    htim8.Instance = TIM8;
    htim8.Init.Prescaler = 72-1;
    htim8.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim8.Init.Period = 0xffff-1;
    htim8.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim8.Init.RepetitionCounter = 0;
    htim8.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_Base_Init(&htim8) != HAL_OK)
    {
        Error_Handler();
    }
    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim8, &sClockSourceConfig) !=
HAL_OK)
    {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim8, &sMasterConfig)
!= HAL_OK)
    {
        Error_Handler();
    }

}

static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure = {0};
    __HAL_RCC_GPIOH_CLK_ENABLE();

```

```

__HAL_RCC_GPIOA_CLK_ENABLE();
__HAL_RCC_GPIOB_CLK_ENABLE();

HAL_GPIO_WritePin(GPIOA, Trig1_Pin|Trig2_Pin|Trig3_Pin|Trig4_Pin,
GPIO_PIN_RESET);

GPIO_InitStruct.Pin = Trig1_Pin|Trig2_Pin|Trig3_Pin|Trig4_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

GPIO_InitStruct.Pin = Eho1_Pin|Eho2_Pin|Eho3_Pin|Eho4_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

}
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
    if (htim == &htim8) {
        //timeMSec++;
    }
}

uint8_t I2C_Scan(I2C_HandleTypeDef *hi2c) {
    for (uint8_t address = 1; address < 128; address++) {
        if (HAL_I2C_IsDeviceReady(hi2c, address << 1, 1,
HAL_MAX_DELAY) == HAL_OK) {
            return address;
        }
    }
    return 0;
}

void printDef(void) {
    // FL
    LCD_SetCursor(6,0);
    LCD_PrintString("FL");
    // FR
    LCD_SetCursor(12,0);
    LCD_PrintString("FR");
    // DL
    LCD_SetCursor(9,0);
    LCD_PrintMyChar(LCD_MUZ_R);
    // DR
    LCD_SetCursor(10,0);
    LCD_PrintMyChar(LCD_MUZ_L);
    // wheel R up
    LCD_SetCursor(8,1);
    LCD_PrintMyChar(LCD_WHEEL_R);
    // wheel L up
    LCD_SetCursor(11,1);
    LCD_PrintMyChar(LCD_WHEEL_L);
}

```

```

    // car
    LCD_SetCursor(9,1);
    LCD_PrintMyChar(LCD_FILL);
    LCD_SetCursor(10,1);
    LCD_PrintMyChar(LCD_FILL);
    LCD_SetCursor(9,2);
    LCD_PrintMyChar(LCD_FILL);
    LCD_SetCursor(10,2);
    LCD_PrintMyChar(LCD_FILL);
    // BL
    LCD_SetCursor(6,3);
    LCD_PrintString("BL");
    // wheel L down
    LCD_SetCursor(8,2);
    LCD_PrintMyChar(LCD_WHEEL_R);
    // wheel R down
    LCD_SetCursor(11,2);
    LCD_PrintMyChar(LCD_WHEEL_L);
    // BR
    LCD_SetCursor(12,3);
    LCD_PrintString("BR");
}

void setPoints(void) {
    for (uint8_t i = 0; i < NUM_HC; i++){
        if ( hcData.dist[i] < DIST1)
        {
            hcData.numPoint[i] = 6;
        }
        else if ( hcData.dist[i] < DIST2)
        {
            hcData.numPoint[i] = 5;
        }
        else if ( hcData.dist[i] < DIST3)
        {
            hcData.numPoint[i] = 4;
        }
        else if ( hcData.dist[i] < DIST4)
        {
            hcData.numPoint[i] = 3;
        }
        else if ( hcData.dist[i] < DIST5)
        {
            hcData.numPoint[i] = 2;
        }
        else if ( hcData.dist[i] < DIST6)
        {
            hcData.numPoint[i] = 1;
        }
        else {
            hcData.numPoint[i] = 0;
        }
        displayPoints(i);
    }
}

```

```

}

void displayPoints(uint8_t numHc) {
    switch(numHc) {
        case HC_FL:{
            if (hcData.numPoint[HC_FL] < hcData.oldNumPoint[HC_FL]){
                while (hcData.numPoint[HC_FL] !=
hcData.oldNumPoint[HC_FL]){
                    displayPoint(numHc, 0);
                    hcData.oldNumPoint[HC_FL]--;
                }
            }
            else if (hcData.numPoint[HC_FL] >
hcData.oldNumPoint[HC_FL]){
                while (hcData.numPoint[HC_FL] !=
hcData.oldNumPoint[HC_FL]){
                    hcData.oldNumPoint[HC_FL]++;
                    displayPoint(numHc, 1);
                }
            }
            break;
        }
        case HC_FR:{
            if (hcData.numPoint[HC_FR] < hcData.oldNumPoint[HC_FR]){
                while(hcData.numPoint[HC_FR] !=
hcData.oldNumPoint[HC_FR]){
                    displayPoint(numHc, 0);
                    hcData.oldNumPoint[HC_FR]--;
                }
            }
            else if (hcData.numPoint[HC_FR] >
hcData.oldNumPoint[HC_FR]){
                while (hcData.numPoint[HC_FR] !=
hcData.oldNumPoint[HC_FR]){
                    hcData.oldNumPoint[HC_FR]++;
                    displayPoint(numHc, 1);
                }
            }
            break;
        }
        case HC_BL:{
            if (hcData.numPoint[HC_BL] < hcData.oldNumPoint[HC_BL]){
                while(hcData.numPoint[HC_BL] !=
hcData.oldNumPoint[HC_BL]){
                    displayPoint(numHc, 0);
                    hcData.oldNumPoint[HC_BL]--;
                }
            }
            else if (hcData.numPoint[HC_BL] >
hcData.oldNumPoint[HC_BL]){
                while(hcData.numPoint[HC_BL] !=
hcData.oldNumPoint[HC_BL]){
                    hcData.oldNumPoint[HC_BL]++;
                    displayPoint(numHc, 1);
                }
            }
        }
    }
}

```



```

        }
    }
    break;
}
case HC_BR:{
    if (hcData.numPoint[HC_BR] < hcData.oldNumPoint[HC_BR]){
        while (hcData.numPoint[HC_BR] !=
hcData.oldNumPoint[HC_BR]){
            displayPoint(numHc, 0);
            hcData.oldNumPoint[HC_BR]--;
        }
    }
    else if (hcData.numPoint[HC_BR] >
hcData.oldNumPoint[HC_BR]){
        while (hcData.numPoint[HC_BR] !=
hcData.oldNumPoint[HC_BR]){
            hcData.oldNumPoint[HC_BR]++;
            displayPoint(numHc, 1);
        }
    }
    break;
}
default:
    break;
}
}

```

```

void displayPoint(uint8_t numHc, uint8_t state){
    switch(numHc) {
    case HC_FL:{
        switch(hcData.oldNumPoint[HC_FL]) {
        case 6: {
            LCD_SetCursor(0,0);
            LCD_PrintMyChar(state);
            break;
        }
        case 5: {
            LCD_SetCursor(1,0);
            LCD_PrintMyChar(state);
            break;
        }
        case 4: {
            LCD_SetCursor(2,0);
            LCD_PrintMyChar(state);
            break;
        }
        case 3: {
            LCD_SetCursor(3,0);
            LCD_PrintMyChar(state);
            break;
        }
        case 2: {
            LCD_SetCursor(4,0);
            LCD_PrintMyChar(state);

```

```

        break;
    }
    case 1: {
        LCD_SetCursor(5,0);
LCD_PrintMyChar(state);
        break;
    }
    case 0: {
        break;
    }
    default:break;
    }
    break;
}
case HC_FR:{
    switch(hcData.oldNumPoint[HC_FR]){
        case 6: {
            LCD_SetCursor(19,0);
LCD_PrintMyChar(state);
            break;
        }
        case 5: {
            LCD_SetCursor(18,0);
LCD_PrintMyChar(state);
            break;
        }
        case 4: {
            LCD_SetCursor(17,0);
LCD_PrintMyChar(state);
            break;
        }
        case 3: {
            LCD_SetCursor(16,0);
LCD_PrintMyChar(state);
            break;
        }
        case 2: {
            LCD_SetCursor(15,0);
LCD_PrintMyChar(state);
            break;
        }
        case 1: {
            LCD_SetCursor(14,0);
LCD_PrintMyChar(state);
            break;
        }
        case 0: {
            break;
        }
        default:break;
    }
    break;
}
case HC_BL:{

```

```

switch (hcData.oldNumPoint[HC_BL]) {
case 6: {
    LCD_SetCursor(0,3);
LCD_PrintMyChar(state);
    break;
}
case 5: {
    LCD_SetCursor(1,3);
LCD_PrintMyChar(state);
    break;
}
case 4: {
    LCD_SetCursor(2,3);
LCD_PrintMyChar(state);
    break;
}
case 3: {
    LCD_SetCursor(3,3);
LCD_PrintMyChar(state);
    break;
}
case 2: {
    LCD_SetCursor(4,3);
LCD_PrintMyChar(state);
    break;
}
case 1: {
    LCD_SetCursor(5,3);
LCD_PrintMyChar(state);
    break;
}
case 0: {
    break;
}
default:break;
}
break;
}
case HC_BR:{
switch (hcData.oldNumPoint[HC_BR]) {
case 6: {
    LCD_SetCursor(19,3);
LCD_PrintMyChar(state);
    break;
}
case 5: {
    LCD_SetCursor(18,3);
LCD_PrintMyChar(state);
    break;
}
case 4: {
    LCD_SetCursor(17,3);
LCD_PrintMyChar(state);
    break;
}
}
}
}

```

```

        }
        case 3: {
            LCD_SetCursor(16,3);
LCD_PrintMyChar(state);
            break;
        }
        case 2: {
            LCD_SetCursor(15,3);
LCD_PrintMyChar(state);
            break;
        }
        case 1: {
            LCD_SetCursor(14,3);
LCD_PrintMyChar(state);
            break;
        }
        case 0: {
            break;
        }
        default:break;
    }
    break;
}
default:
    break;
}
}
void Error_Handler(void)
{
return state */
    __disable_irq();
    while (1)
    {
    }
}

#ifdef USE_FULL_ASSERT
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and
line number,
    ex: printf("Wrong parameters value: file %s on line %d\r\n",
file, line) */
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

```