Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматизації, комп'ютерно-інтегрованих технологій та систем
(повна назва)

Кафедра Комп'ютерно-інегрованих технологій, автоматизації, робототехніки та безпекової інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)
(позначення документа)

КТ 1. Програмно-апаратний комплекс для сортування поштових відправлень. ПТ 2. Розробка програмного забезпечення
(тема)

Виконав:
здобувач 4 року навчання,
групи АКТАКІТ-22-2
Владислав КАЩЕЄВ
(власне ім'я, прізвище)

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Автоматизація та комп'ютерно-інтегровані технології
(повна назва освітньої програми)

Керівник доц. Дмитро НІКІТІН
(посада, власне ім'я, прізвище)

Допускається до захисту
Зав. кафедри _____ Ігор НЕВЛЮДОВ
(підпис) (власне ім'я, прізвище)

2026 р.

Я, Кащеєв Владислав Андрійович, як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«03» липня 2026 р.  Владислав КАЩЕЄВ

Харківський національний університет радіоелектроніки

Факультет Автоматизації, комп'ютерно-інтегрованих технологій та систем
Кафедра Комп'ютерно-інегрованих технологій, автоматизації, робототехніки та безпекової інженерії

Рівень вищої освіти перший (бакалаврський)
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Автоматизація та комп'ютерно-інтегровані технології

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувачеві Кащеєву Владиславу Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи КТ 1. Програмно-апаратний комплекс для сортування поштових відправлень. ПТ 2. Розробка програмного забезпечення
затверджена наказом по університету від «04» 06 2026 р. № 472 Ст
2. Термін подання здобувачем роботи 14.07.2026 р.
3. Вихідні дані до роботи 1. Одноплатний комп'ютер Raspberry Pi 5 під керуванням операційної системи Raspberry Pi OS.
2. Мова програмування Python 3 та середовище розробки програмного забезпечення.
3. Бібліотеки OpenCV, Pyzbar, Flask, Tkinter для реалізації комп'ютерного зору, розпізнавання QR-кодів, мережевої взаємодії та графічного інтерфейсу користувача.
4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ;
4.2 Аналіз сучасних систем автоматизації логістичних процесів, методів навігації мобільних роботів та програмного забезпечення.;
4.3 Проєктування програмно-апаратного комплексу сортування поштових відправлень.
та розробка структури програмного забезпечення оператора і мобільного робота;
4.4 Розробка програмного забезпечення, реалізація QR-навігації, автоматичного калібрування та керування мобільним роботом.

5. Перелік графічного матеріалу із зазначенням обов'язкових креслеників, сх плакатів, комп'ютерних ілюстрацій: Демонстраційний матеріал в форматі *. А4 – 10 сторінок

6. Консультанти розділів роботи (п. 6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п. 1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Приміт-ка
1	Аналіз літературних джерел	27.04.2026 – 10.05.2026	виконано
2	Аналіз існуючих систем сортування поштових відправлень та програмного забезпечення	11.05.2026 – 24.05.2026	виконано
3	Проєктування програмно-апаратного комплексу сортування поштових відправлень	25.05.2026 – 07.06.2026	виконано
4	Реалізація системи QR-навігації мобільного робота	23.06.2026 – 26.06.2026	виконано
5	Тестування програмного забезпечення комплексу сортування поштових відправлень	26.06.2026 – 30.06.2026	виконано
6	Оформлення пояснювальної записки	01.07.2026 – 03.07.2026	
7	Подання роботи на перевірку Інтернет-сервісом StrikePlagiarism		
8	Подання роботи на рецензію		
9	Подання роботи на підпис зав. кафедри		
11	Подання кваліфікаційної роботи в ЕК		

Дата видачі завдання 27.04.2026 р.

Здобувач _____ Владислав КАЩЕЄВ
(підпис) (посада, власне ім'я, прізвище)

Керівник роботи _____ доц. Дмитро НІКІТІН
(підпис) (власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 119 с., 12 табл., 50 рис., 2 дод., 31 джерело.

МОБІЛЬНИЙ РОБОТ, СОРТУВАННЯ ПОШТОВИХ ВІДПРАВЛЕНЬ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, QR-НАВІГАЦІЯ, КОМП'ЮТЕРНИЙ ЗІР, PYTHON, RASPBERRY PI, OPENCV, АВТОМАТИЧНЕ КАЛІБРУВАННЯ.

Об'єкт розробки – процес автоматизації сортування поштових відправлень із використанням мобільної роботизованої платформи.

Предмет розробки – методи, алгоритми та програмне забезпечення керування мобільним роботом, реалізації QR-навігації, автоматичного калібрування складського приміщення та виконання транспортних завдань.

Мета кваліфікаційної роботи – розробити програмне забезпечення програмно-апаратного комплексу сортування поштових відправлень, яке забезпечує автоматичне калібрування складського приміщення, навігацію мобільного робота за допомогою QR-кодів, взаємодію з оператором та виконання транспортних завдань.

У кваліфікаційній роботі проведено аналіз сучасних систем автоматизації логістичних процесів, програмного забезпечення для керування мобільними роботами та методів QR-навігації. Розроблено архітектуру програмно-апаратного комплексу сортування поштових відправлень, реалізовано програмне забезпечення оператора та мобільного робота із застосуванням мови програмування Python, бібліотек OpenCV, Pyzbar, Flask і Tkinter. Реалізовано алгоритми автоматичного калібрування складського приміщення, побудови карти контрольних точок, визначення відстані до QR-кодів, навігації мобільного робота та виконання транспортних завдань.

ABSTRACT

Explanatory note: 119 pages, 12 tables, 50 figures, 2 appendices, 31 references.

MOBILE ROBOT, PARCEL SORTING, SOFTWARE, QR NAVIGATION, COMPUTER VISION, PYTHON, RASPBERRY PI, OPENCV, AUTOMATIC CALIBRATION.

The object of development is a hardware and software complex for automating the parcel sorting process using a mobile robotic platform.

The subject of development is the methods, algorithms, and software for mobile robot control, QR-based navigation, automatic warehouse calibration, and transportation task execution.

The purpose of the qualification work is to develop software for a parcel sorting hardware and software complex that provides automatic warehouse calibration, QR-code-based mobile robot navigation, operator interaction, and transportation task execution.

Research methods include system analysis, object-oriented programming methods, computer vision techniques, automatic control algorithms, and experimental software testing.

The qualification work analyzes modern logistics automation systems, software for mobile robot control, and QR-code navigation methods. The architecture of the hardware and software complex for parcel sorting has been developed. The operator software and mobile robot software were implemented using Python, OpenCV, Pyzbar, Flask, and Tkinter. Algorithms for automatic warehouse calibration, control point mapping, QR-code distance estimation, mobile robot navigation, and transportation task execution have been implemented. Experimental testing confirmed the operability of the developed software and demonstrated the feasibility of its application for automating parcel sorting processes.

ЗМІСТ

Перелік скорочень 9

Вступ 11

1.1 Особливість автоматизації логістичних процесів 13

1.2 Сучасні системи сортування поштових відправлень 14

1.3 Використання мобільних роботів у складських комплексах 16

1.4 Аналіз методів навігації мобільних роботів 18

1.5 Використання QR-кодів для ідентифікації та навігації 20

1.6 Аналіз принципів автоматичного керування мобільними роботами 22

1.7 Аналіз апаратних платформ для мобільних роботів 25

1.8 Аналіз існуючого програмного забезпечення та формування вимог до програмного забезпечення комплексу сортування поштових відправлень 29

1.9 Постановка задачі дослідження 36

2 Проєктування програмно-апаратного комплексу сортування поштових відправлень 38

2.1 Загальна структура програмно-апаратного комплексу 38

2.2 Проєктування архітектури програмного забезпечення 40

2.3 Проєктування системи QR-навігації 42

2.4 Проєктування алгоритму калібрування системи 45

2.5 Розробка алгоритму сортування поштових відправлень 47

2.6 Проєктування взаємодії програмного забезпечення з мобільним роботом 50

2.7 Проєктування інтерфейсу оператора 52

3 Розробка програмного забезпечення комплексу сортування поштових відправлень 55

3.1 Вибір засобів розробки програмного забезпечення 55
3.2 Реалізація структури програмного забезпечення 56
3.3 Розробка модуля взаємодії з мобільним роботом..... 58
3.4 Розробка модуля обробки та розпізнавання QR-кодів..... 61
3.5 Реалізація алгоритму калібрування складського приміщення..... 66
3.6 Реалізація алгоритму сортування поштових відправлень 70
3.7 Розробка графічного інтерфейсу оператора..... 74
4 Дослідження та тестування програмного забезпечення 78
4.1 Методика проведення експериментальних досліджень..... 78
4.2 Тестування модуля розпізнавання QR-кодів..... 82
4.3 Тестування алгоритму калібрування складського приміщення..... 85
4.4 Тестування алгоритму сортування поштових відправлень 88
4.5 Аналіз умов праці під час роботи з програмним забезпеченням..... 91
4.6 Вимоги безпеки під час роботи за персональним компютером..... 92
4.7 Пожежна безпека..... 92
Перелік джерел посилання 96
Додаток а Лістинг коду..... 99
ДОДАТОК Б Демонстраційний матеріал 120

ПЕРЕЛІК СКОРОЧЕНЬ

КІТ – Київський інноваційний термінал.

AGV (Automated Guided Vehicle) – автоматизований транспортний засіб, що рухається за заздалегідь визначеним маршрутом.

AMR (Autonomous Mobile Robot) – автономний мобільний робот, здатний самостійно будувати маршрут та адаптуватися до змін середовища.

CPU (Central Processing Unit) – центральний процесор.

FPS (Frames Per Second) – кількість кадрів за секунду у відеопотоці.

GPIO (General Purpose Input/Output) – універсальні входи/виходи мікроконтролера або одноплатного комп'ютера.

GUI (Graphical User Interface) – графічний інтерфейс користувача.

HTTP (HyperText Transfer Protocol) – протокол передачі гіпертекстових даних.

IP-адреса – унікальна мережева адреса пристрою в комп'ютерній мережі.

LiDAR (Light Detection and Ranging) – технологія визначення відстані за допомогою лазерного випромінювання.

MJPEG (Motion JPEG) – формат потокової передачі відео, що використовується для трансляції зображення з камери мобільного робота.

OpenCV (Open Source Computer Vision Library) – бібліотека алгоритмів комп'ютерного зору та обробки зображень.

QR-код – двовимірний матричний код для зберігання інформації (Quick Response Code).

RFID (Radio Frequency Identification) – технологія автоматичної ідентифікації об'єктів за допомогою радіочастотних міток.

SLAM (Simultaneous Localization and Mapping) – алгоритм одночасного визначення місцезнаходження та побудови карти середовища.

TCP (Transmission Control Protocol) – транспортний протокол передачі даних.

Wi-Fi (Wireless Fidelity) – технологія бездротового передавання даних.

WMS (Warehouse Management System) – система управління складом.

ВСТУП

Сучасний розвиток електронної комерції, поштових та логістичних сервісів супроводжується постійним збільшенням обсягів обробки поштових відправлень, що потребує впровадження сучасних засобів автоматизації складських процесів. Використання автоматизованих систем сортування дозволяє підвищити продуктивність логістичних центрів, скоротити час обробки відправлень, зменшити кількість помилок та знизити вплив людського фактора. Одним із найбільш перспективних напрямів розвитку таких систем є застосування мобільних роботизованих платформ, здатних автономно виконувати транспортні операції в межах складських приміщень.

Ефективність роботи мобільного робота значною мірою визначається можливостями програмного забезпечення, яке забезпечує взаємодію між оператором, роботизованою платформою та засобами навігації. Важливу роль у цьому процесі відіграють алгоритми комп'ютерного зору, автоматичного калібрування робочої зони, побудови карти контрольних точок та навігації за допомогою QR-кодів (Quick Response Code). Використання таких технологій дозволяє швидко адаптувати систему до зміни конфігурації складського приміщення без необхідності складного переналаштування обладнання, що робить розробку спеціалізованого програмного забезпечення актуальним науково-практичним завданням .

Об’єктом розробки є процес автоматизації сортування поштових відправлень із використанням мобільної роботизованої платформи.

Предметом розробки є методи, алгоритми та програмне забезпечення керування мобільним роботом, реалізації QR-навігації, автоматичного калібрування складського приміщення та виконання транспортних завдань.

Метою роботи є розробка програмного забезпечення програмно-апаратного комплексу сортування поштових відправлень, яке забезпечує автоматичне

калібрування складського приміщення, навігацію мобільного робота за допомогою QR-кодів, взаємодію з оператором та виконання транспортних завдань.

Для досягнення поставленої мети у роботі необхідно вирішити такі задачі:

– провести аналіз сучасних систем автоматизації логістичних процесів, програмного забезпечення для керування мобільними роботами та методів QR-навігації;

– проаналізувати існуючі програмні рішення та сформувати вимоги до програмного забезпечення комплексу;

– розробити архітектуру програмного забезпечення та структуру взаємодії його основних модулів; реалізувати програмне забезпечення оператора і мобільного робота із застосуванням сучасних засобів програмування та комп'ютерного зору;

– реалізувати алгоритми автоматичного калібрування складського приміщення, QR-навігації та виконання транспортних завдань;

– провести тестування розробленого програмного забезпечення й оцінити ефективність його роботи в умовах моделювання логістичного процесу.

Розробка програмного забезпечення комплексу сортування поштових відправлень відповідає сучасним тенденціям цифрової трансформації логістичної галузі та сприяє досягненню Цілей сталого розвитку Організації Об'єднаних Націй. Зокрема, результати роботи узгоджуються з Ціллю 9 [1] «Промисловість, інновації та інфраструктура», яка передбачає розвиток інноваційних технологій, автоматизацію виробничих процесів та впровадження сучасних інформаційних систем, а також із Ціллю 11 «Сталий розвиток міст і громад», що спрямована на підвищення ефективності міської логістики та транспортної інфраструктури. Звіт з кваліфікаційної роботи оформлено згідно з ДСТУ 3008:2015 [2] та з використанням навчально-методичних матеріалів кафедри комп'ютерно-інтегрованих технологій, автоматизації, робототехніки та безпекової інженерії [3-4].

1 АНАЛІЗ СУЧАСНИХ СИСТЕМ СОРТУВАННЯ ПОШТОВИХ ВІДПРАВЛЕНЬ

1.1 Особливість автоматизації логістичних процесів

У сучасних умовах розвитку поштових сервісів кількість відправлень постійно зростає. Через це логістичні компанії змушені шукати нові способи підвищення ефективності своєї роботи. Такими способами є автоматизація процесів сортування, обліку та переміщення вантажів. Її використання дозволяє зменшити час обробки замовлень і підвищити точність виконання операцій. Автоматизація логістичних процесів [5,6] передбачає застосування програмних і технічних засобів для управління матеріальними потоками. Завдяки цьому можна скоротити кількість помилок під час сортування відправлень. Крім того, автоматизовані системи допомагають швидше реагувати на зміни в навантаженні та забезпечувати стабільну якість обслуговування клієнтів. На багатьох сучасних складах використовуються системи управління складом [8]. Такі системи дають можливість контролювати переміщення товарів, вести облік залишків і відстежувати виконання складських операцій у режимі реального часу. Це значно спрощує роботу працівників та дозволяє підвищити загальну ефективність складського комплексу. Останніми роками все більшого поширення набувають роботизовані транспортні платформи. Вони можуть автоматично переміщувати вантажі між різними зонами складу, доставляти необхідні матеріали до робочих місць та брати участь у процесах сортування. Важливою складовою автоматизації є також використання технологій ідентифікації об’єктів. До них належать штрих-коди, QR-коди, RFID-мітки [13] та системи комп’ютерного зору. Завдяки цим технологіям можна швидко визначати місцезнаходження товарів і контролювати їх переміщення на всіх етапах логістичного процесу. Таким чином, автоматизація логістики є важливим напрямом розвитку сучасних складських та поштових

комплексів. Вона сприяє підвищенню швидкості та точності виконання операцій, а також створює основу для подальшого впровадження роботизованих систем і новітніх технологій у сфері логістики.

Практичним прикладом успішного впровадження автоматизації логістичних процесів в Україні є компанія «Нова пошта» [5,6]. Для обробки зростаючого обсягу відправлень компанія створює інноваційні сортувальні термінали, оснащені автоматизованими конвеєрними системами та обладнанням для сканування посилок. Крім того, у Дніпрі було відкрито автоматизований термінал, який може обробляти до 26 тис. посилок на годину з перспективою збільшення потужності до 36 тис. посилок на годину. Використання таких автоматизованих рішень дозволяє значно прискорити сортування відправлень, підвищити точність логістичних операцій та зменшити навантаження на персонал.

Одним із перспективних напрямів розвитку автоматизованих логістичних систем є використання мобільних роботів, які здатні самостійно переміщуватися між зонами складу та виконувати завдання з транспортування відправлень. Для навігації таких роботів можуть використовуватися різні технології, серед яких особливий інтерес становлять QR-коди завдяки простоті реалізації та невисокій вартості впровадження.

1.2 Сучасні системи сортування поштових відправлень

У зв’язку зі збільшенням популярності електронної комерції значно зросли обсяги поштових відправлень. Але логістичним компаніям необхідно обробляти велику кількість посилок за короткий проміжок часу. Саме тому все більшого попиту набувають автоматизовані системи сортування, які дозволяють прискорити обробку відправлень, зменшити кількість помилок та підвищити ефективність роботи логістичних центрів.

Одним із найбільш технологічно розвинених логістичних об’єктів в Україні є Київський інноваційний термінал [6] (КІТ) компанії «Нова пошта» (рисунок 1.1). Перша черга терміналу була введена в експлуатацію у 2018 році в місті Києві.

Основним призначенням комплексу є автоматизоване сортування поштових відправлень та забезпечення високої пропускної здатності логістичної мережі компанії.



Рисунок 1.1 – Київський інноваційний термінал компанії «Нова пошта»

Загальна площа Київського інноваційного терміналу становить близько 23,9 тис. м², а потужність першої черги дозволяє обробляти понад 350 тис. відправлень на добу. Для автоматизації процесів сортування використовується обладнання компанії Vanderlande, яке забезпечує автоматичне сканування та розподіл відправлень між напрямками доставки. Система здатна виконувати сканування п'яти сторін посилки зі швидкістю до 2,3 відправлення [6] за секунду, що значно скорочує час обробки поштових відправлень. Крім того, термінал обслуговує понад 400 вантажних автомобілів на добу та забезпечує безперервну роботу логістичного центру в декілька змін. Використання автоматизованих сортувальних комплексів дозволяє суттєво підвищити швидкість обробки посилок, зменшити вплив людського фактора та забезпечити стабільну якість логістичних послуг. Досвід компанії «Нова пошта» демонструє ефективність впровадження сучасних автоматизованих технологій та підтверджує доцільність використання

роботизованих і програмно-апаратних систем у процесах сортування поштових відправлень.

1.3 Використання мобільних роботів у складських комплексах

Розвиток електронної комерції та збільшення кількості вантажних перевезень сприяли активному впровадженню автоматизованих технологій у логістичній сфері. Одним із сучасних рішень є використання мобільних роботів [8,9], які можуть виконувати транспортування вантажів, брати участь у комплектуванні замовлень і забезпечувати переміщення товарів між різними зонами складу. Використання таких систем дозволяє підвищити ефективність роботи логістичних центрів та скоротити час виконання складських операцій.

Сьогодні мобільні роботи широко застосовуються на автоматизованих складах і в логістичних комплексах. Їх основне призначення полягає у транспортуванні вантажів та взаємодії з іншими елементами складської інфраструктури. Для орієнтації в просторі роботи використовують датчики, камери, сканери та спеціальні алгоритми навігації, що дозволяє їм визначати своє місцезнаходження та самостійно рухатися за заданим маршрутом.

Залежно від способу навігації мобільні роботи поділяються на автоматизовані транспортні засоби AGV (Automated Guided Vehicle) та автономні мобільні роботи AMR (Autonomous Mobile Robot). Роботи типу AGV рухаються за попередньо визначеними маршрутами, використовуючи розмітку, магнітні стрічки або QR-мітки. У свою чергу, роботи AMR [8,9] мають більш гнучкі можливості, оскільки можуть самостійно будувати маршрут, виявляти перешкоди та адаптуватися до змін навколишнього середовища. Для цього застосовуються технології комп’ютерного зору, побудови карт приміщення та алгоритми штучного інтелекту.

Одним із найбільш відомих прикладів використання мобільних роботів є складські комплекси компанії Amazon. Після придбання Kiva Systems у 2012 році компанія почала активно впроваджувати роботизовані рішення у власних

логістичних центрах. На сьогодні тисячі роботів забезпечують транспортування товарів, сортування відправлень та виконання допоміжних складських операцій. Для цього використовуються роботи серій Hercules (рисунок 1.2), Titan та Proteus [7].



Рисунок 1.2 – Мобільний робот Hercules компанії Amazon

Роботи Hercules здатні переміщувати стелажі масою до 560 кг та доставляти їх до зон комплектації замовлень. Завдяки цьому працівникам не потрібно витрачати час на пошук необхідних товарів у складських рядах, оскільки стелажі автоматично подаються до робочих станцій. Такий підхід дозволяє підвищити продуктивність складу та скоротити час обробки замовлень.

Для задач автоматизованого сортування особливий інтерес становлять мобільні платформи, обладнані камерами та системами комп'ютерного зору. Такі роботи можуть використовувати QR-коди [13], штрих-коди та інші візуальні мітки для визначення свого положення та навігації в приміщенні. Це дозволяє спростити систему позиціонування та знизити вартість реалізації автоматизованого комплексу.

У межах даної кваліфікаційної роботи розглядається мобільна роботизована платформа на базі Raspberry Pi 5, яка здійснює переміщення між контрольними точками за допомогою QR-навігації. Такий підхід дозволяє реалізувати автоматизоване сортування та транспортування поштових відправлень без використання дорогих систем позиціонування.

Отже, мобільні роботизовані системи є перспективним напрямом розвитку сучасної логістики. Їх застосування дозволяє підвищити швидкість виконання складських операцій, зменшити навантаження на персонал та створює передумови для подальшої автоматизації логістичних комплексів.

1.4 Аналіз методів навігації мобільних роботів

Одним із важливих етапів під час розроблення мобільних роботизованих систем є організація навігації в робочому середовищі. Саме навігаційна система забезпечує визначення поточного положення робота, побудову маршруту та переміщення між заданими точками. Від її ефективності залежить точність виконання поставлених завдань, швидкість роботи та загальна продуктивність роботизованого комплексу. На сьогодні існує декілька способів навігації мобільних роботів (таблиця 1.1) [8, 9]. Одним із найпростіших рішень є використання фізичної розмітки маршруту [10]. У такому випадку робот рухається за лінією, магнітною стрічкою або іншими направляючими елементами, розташованими на підлозі. Перевагою цього методу є простота реалізації та невисока вартість, однак зміна маршруту потребує внесення змін до інфраструктури приміщення, що знижує гнучкість системи.

Ще одним методом є застосування RFID-міток. У цьому випадку в різних точках приміщення встановлюються спеціальні радіочастотні мітки, інформація з яких зчитується роботом під час руху. Такий підхід дозволяє визначати місцезнаходження робота та організовувати його переміщення між окремими зонами. Водночас використання RFID потребує додаткового обладнання, а точність позиціонування може бути недостатньою для окремих задач [10].

У більш складних робототехнічних системах використовуються технології LiDAR (Light Detection and Ranging) та алгоритми SLAM (Simultaneous Localization and Mapping). LiDAR дає змогу отримувати інформацію про навколишнє середовище шляхом вимірювання відстані до об'єктів за допомогою лазерного випромінювання. Алгоритми SLAM дозволяють одночасно будувати карту приміщення та визначати положення робота на цій карті. Такі рішення забезпечують високу точність навігації та добре підходять для роботи в динамічних умовах. Проте їх використання пов’язане зі збільшенням вартості системи та підвищеними вимогами до обчислювальних ресурсів.

Окрему увагу привертають системи навігації, що базуються на використанні візуальних маркерів. До них належать QR-коди, штрих-коди та інші графічні позначки. За допомогою камери та алгоритмів комп'ютерного зору робот може розпізнавати такі мітки, визначати своє положення та переміщуватися між необхідними точками маршруту.

Таблиця 1.1 – Порівняння методів навігації мобільних роботів

Метод навігації	Переваги	Недоліки
Лінійна навігація (розмітка, магнітна стрічка)	Простота реалізації, низька вартість	Низька гнучкість, складність зміни маршруту
RFID-навігація	Автоматична ідентифікація контрольних точок	Необхідність встановлення RFID-міток та зчитувачів
LiDAR + SLAM	Висока точність, адаптація до змін середовища	Висока вартість обладнання, значні обчислювальні ресурси
QR-навігація	Низька вартість, простота реалізації, швидка зміна маршруту	Потребує наявності та видимості QR-міток

Для програмно-апаратного комплексу сортування поштових відправлень, який розробляється в межах даної кваліфікаційної роботи, було обрано навігацію на основі QR-кодів. Перевагами даного підходу є невисока вартість реалізації, простота налаштування та можливість швидкої зміни конфігурації системи без додаткових апаратних модифікацій. Порівняно з RFID-навігацією та системами на основі LiDAR, використання QR-кодів не потребує спеціалізованих датчиків і може бути реалізоване за допомогою стандартної камери, що особливо важливо для малобюджетних роботизованих платформ. Крім того, QR-коди можуть містити інформацію про конкретні складські зони, що спрощує реалізацію алгоритму сортування та ідентифікацію пунктів призначення.

1.5 Використання QR-кодів для ідентифікації та навігації

Одним із поширених засобів автоматичної ідентифікації об’єктів є QR-коди. QR-код являє собою двовимірний матричний код, який використовується для зберігання текстової та цифрової інформації у графічному вигляді (рисунок 1.3). Технологія була розроблена компанією DENSO WAVE та сьогодні широко застосовується в логістиці, складських комплексах, торгівлі та робототехнічних системах. Основною перевагою QR-кодів є можливість зберігання більшого обсягу інформації порівняно зі звичайними штрих-кодами. Крім того, вони можуть зчитуватися під різними кутами та підтримують механізми корекції помилок, що дозволяє успішно розпізнавати код навіть у разі його часткового пошкодження.

У логістичних системах QR-коди використовуються для маркування товарів, контейнерів, складських зон та транспортних платформ. Після зчитування коду система отримує доступ до інформації про об’єкт, його місцезнаходження або пункт призначення.

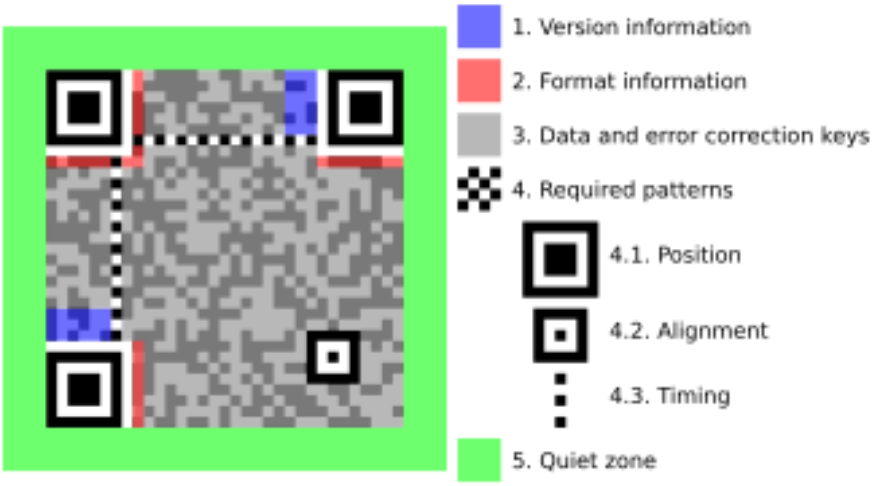


Рисунок 1.3 – Приклад структури QR-коду

Використання такого підходу дозволяє спростити облік вантажів та підвищити швидкість виконання логістичних операцій. Окрім ідентифікації об’єктів, QR-коди можуть використовуватися як елементи навігаційної інфраструктури мобільних роботів. У цьому випадку коди розміщуються в контрольних точках маршруту, а робот за допомогою камери виконує їх пошук та розпізнавання (рисунок 1.4). Після декодування QR-коду система отримує інформацію про поточне місцезнаходження робота або його подальший маршрут руху.

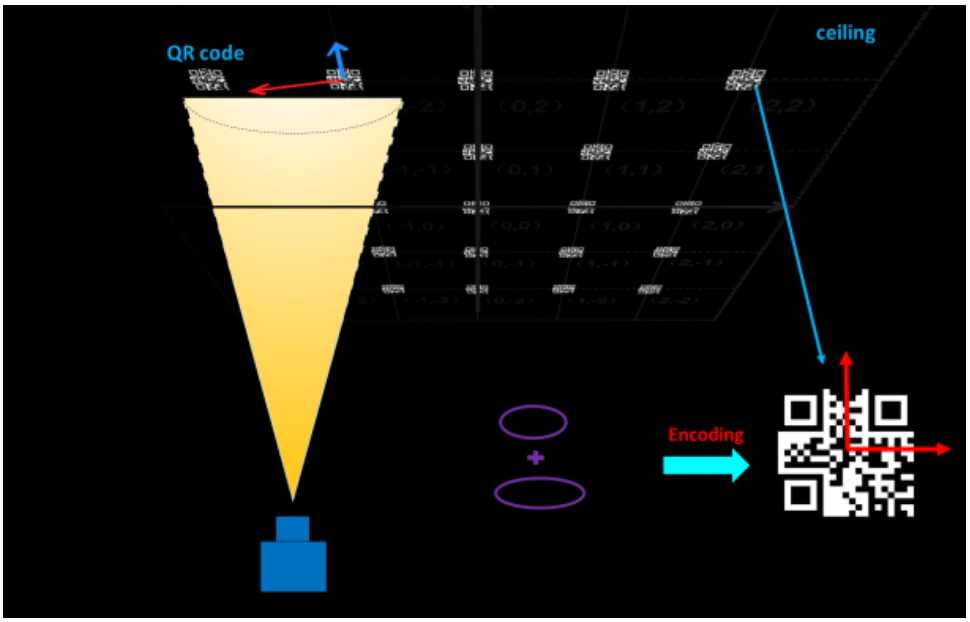


Рисунок 1.4 – Використання QR-кодів для навігації мобільного робота

Для реалізації навігації за QR-кодами достатньо використовувати камеру та програмні алгоритми комп’ютерного зору. Зображення з камери аналізується в режимі реального часу, після чого виконується пошук і декодування QR-коду. Отримані дані використовуються для визначення координат контрольної точки та формування подальших команд керування роботом. Порівняно з навігаційними системами на основі LiDAR або RFID, QR-навігація характеризується нижчою вартістю реалізації та меншою складністю впровадження. Для її роботи не потрібні спеціалізовані датчики або складна інфраструктура, що особливо важливо для малобюджетних роботизованих платформ.

У межах даної кваліфікаційної роботи QR-коди використовуються як контрольні точки маршруту мобільної роботизованої платформи. Кожен код містить інформацію про відповідну складську зону або пункт призначення. Після зчитування QR-коду робот визначає своє поточне положення та отримує дані для подальшого переміщення між об'єктами складського комплексу. Для розпізнавання QR-кодів у розробленій системі використовуються бібліотеки OpenCV та Pyzbar. OpenCV забезпечує отримання й попередню обробку відеозображення, а бібліотека Pyzbar виконує пошук і декодування QR-кодів [18,19,28]. Її засоби дозволяють виконувати пошук, локалізацію та декодування QR-кодів у режимі реального часу, що забезпечує можливість побудови системи навігації мобільного робота та реалізації функцій сортування поштових відправлень.

1.6 Аналіз принципів автоматичного керування мобільними роботами

Однією з важливих складових сучасних робототехнічних систем є автоматичне керування. Його основне призначення полягає у забезпеченні виконання поставлених завдань без постійного втручання людини. Системи автоматичного керування широко застосовуються у промисловості, транспорті, логістиці та робототехніці, забезпечуючи підвищення точності, швидкості та надійності виконання різноманітних операцій.

Система автоматичного керування являє собою сукупність технічних і програмних засобів, які формують керуючі впливи на об’єкт керування відповідно до заданої мети. У випадку мобільного робота об'єктом керування виступає рухома платформа, а керуючими впливами є команди, що визначають швидкість та напрямок її руху [22, 23].

Залежно від принципу роботи системи автоматичного керування поділяються на розімкнені та замкнені (рисунок 1.5). У розімкнених системах керуючий сигнал формується без урахування фактичного стану об'єкта керування. У замкнених системах використовується зворотний зв'язок, завдяки якому система отримує інформацію про результати своєї роботи та може коригувати подальші дії.

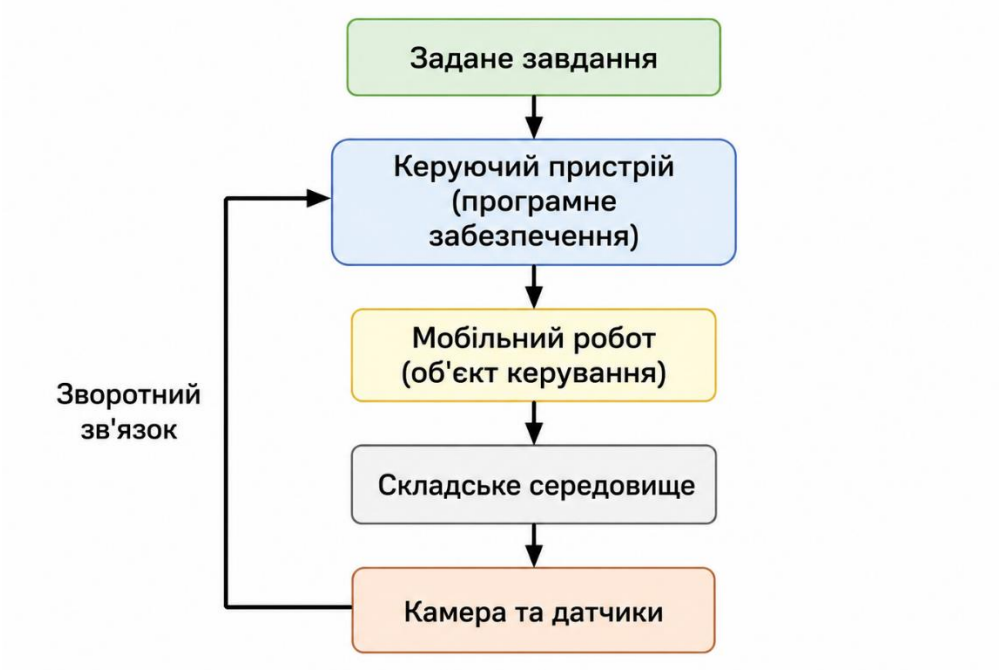


Рисунок 1.5 – Структурна схема системи автоматичного керування мобільним роботом

Основною особливістю замкнених систем є використання зворотного зв'язку. Принцип його роботи полягає у постійному отриманні інформації про поточний стан об'єкта керування та порівнянні отриманих даних із заданими параметрами. У разі виникнення відхилень система формує нові керуючі сигнали, що дозволяє підтримувати необхідний режим роботи [10, 23].

Важливою характеристикою систем автоматичного керування є їхня стійкість. Стійкою вважається система, яка після впливу зовнішніх збурень здатна повернутися до нормального режиму роботи. Для мобільних роботів такими збуреннями можуть бути зміна освітлення, тимчасова втрата навігаційної мітки або поява перешкод на маршруті руху (рисунок 1.6).



Рисунок 1.6 – Принцип роботи замкненої системи керування зі зворотним зв’язком

У сучасних робототехнічних системах джерелами інформації для зворотного зв'язку можуть бути датчики відстані, енкодери, гіроскопи, камери та інші вимірювальні пристрої. Отримані дані використовуються для визначення поточного положення робота та прийняття рішень щодо його подальшого переміщення.

У межах даної кваліфікаційної роботи функції зворотного зв’язку реалізуються за допомогою камери та системи розпізнавання QR-кодів. Під час руху робот аналізує зображення з камери та визначає наявність контрольних точок маршруту. Отримана інформація використовується для підтвердження досягнення необхідної складської зони та коригування подальших дій. Таким чином, застосування принципів теорії автоматичного керування дозволяє забезпечити

автономне виконання транспортних завдань мобільним роботом та підвищити ефективність роботи комплексу сортування поштових відправлень [22,23].

1.7 Аналіз апаратних платформ для мобільних роботів

Під час розроблення мобільних робототехнічних систем одним із важливих етапів є вибір апаратної платформи. Саме вона відповідає за виконання алгоритмів керування, обробку інформації від датчиків та взаємодію з периферійними пристроями. Для створення мобільних роботів сьогодні використовуються як мікроконтролери, так і одноплатні комп’ютери, які відрізняються між собою продуктивністю, функціональними можливостями та вартістю [14–17, 24].

Однією з найпоширеніших платформ для навчальних і експериментальних проєктів є Arduino (рисунок 1.7). Вона набула популярності завдяки простоті програмування, доступній вартості та великій кількості додаткових модулів. Arduino часто застосовується для керування двигунами, роботи з датчиками та реалізації нескладних алгоритмів [14]. Разом із тим обчислювальні можливості мікроконтролера є обмеженими, тому виконання задач комп’ютерного зору або обробки відео на такій платформі є проблематичним [14].

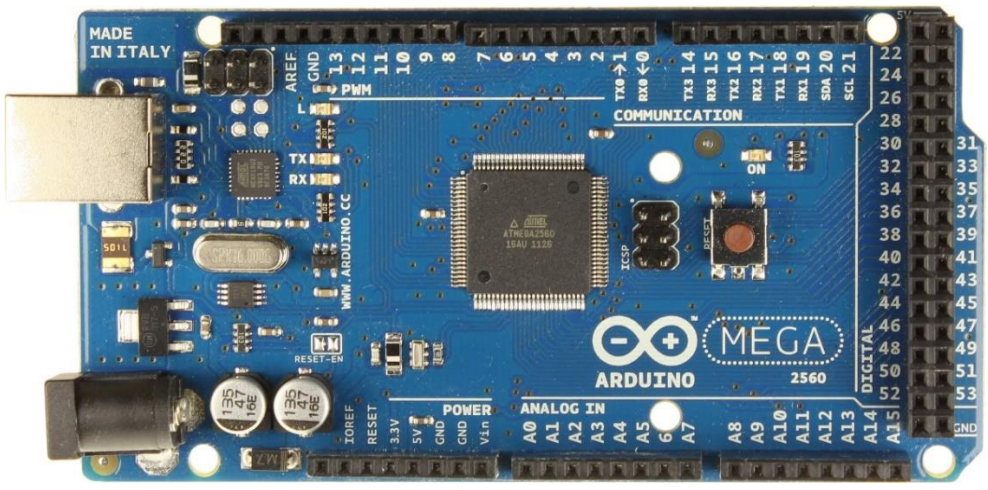


Рисунок 1.7 – Плата Arduino Mega 2560

Ще однією популярною платформою є ESP32 (рисунок 1.8). Особливістю цього мікроконтролера є наявність вбудованих модулів Wi-Fi та Bluetooth [15], що дає змогу використовувати його в системах дистанційного керування та проєктах Інтернету речей. ESP32 має низьке енергоспоживання та достатню продуктивність для реалізації простих робототехнічних рішень. Однак для виконання складних алгоритмів обробки зображень його можливостей недостатньо [15].

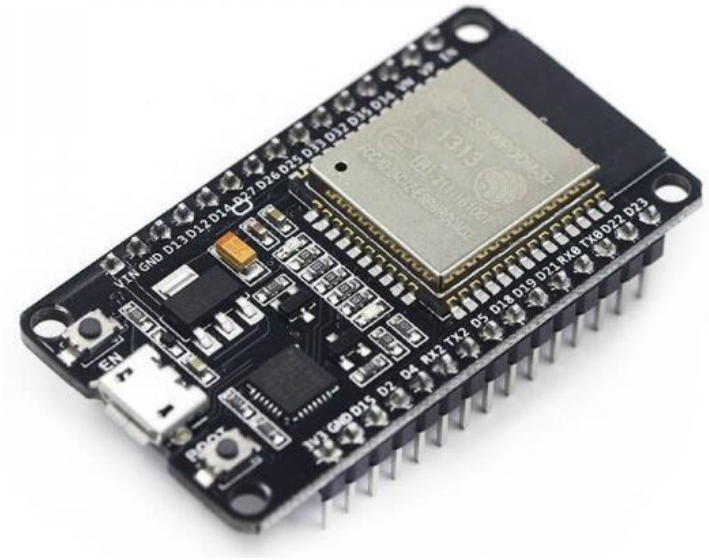


Рисунок 1.8 – Мікроконтролер ESP32

Для більш складних систем часто використовуються одноплатні комп’ютери сімейства Raspberry Pi (рисунок 1.9). Вони працюють під керуванням операційної системи Linux [16,24] та підтримують велику кількість мов програмування і програмних бібліотек. Завдяки цьому Raspberry Pi широко застосовується в робототехніці, системах комп’ютерного зору та проєктах автоматизації. Однією з найсучасніших моделей є Raspberry Pi 5. Вона оснащена чотириядерним процесором ARM Cortex-A76, має до 8 Гб оперативної пам’яті та підтримує підключення камер, дисплеїв і різноманітних периферійних пристроїв через інтерфейс GPIO. Продуктивності цієї платформи достатньо для обробки відеопотоку, розпізнавання QR-кодів і реалізації систем навігації мобільних роботів [16,24].



Рисунок 1.9 – Одноплатний комп’ютер Raspberry Pi 5

Окремо варто відзначити платформу NVIDIA Jetson Nano, яка орієнтована на задачі штучного інтелекту [17] та комп’ютерного зору (рисунок 1.10). Завдяки наявності графічного процесора вона дозволяє прискорювати обчислення під час роботи з нейронними мережами. Проте вартість Jetson Nano є вищою порівняно з Raspberry Pi, тому її використання не завжди є доцільним у навчальних та експериментальних проєктах.

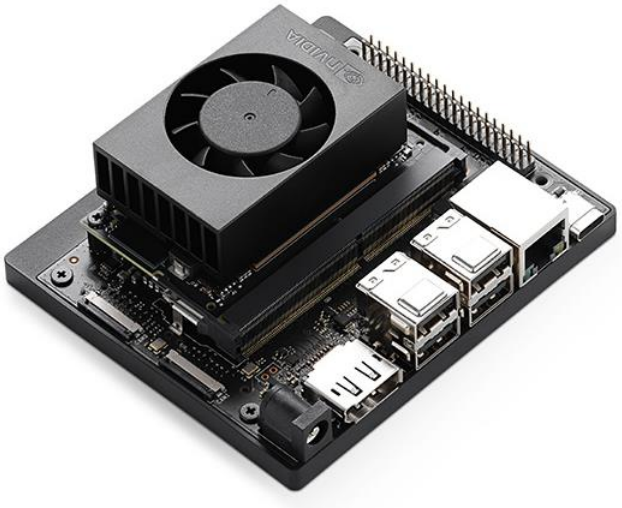


Рисунок 1.10 – Одноплатний комп’ютер NVIDIA Jetson Nano

Для більш наочного порівняння розглянутих рішень складено таблицю 1.2.

Таблиця 1.2 – Порівняння апаратних платформ для мобільних роботів

Характеристика	Arduino Mega	ESP32	Raspberry Pi 5	Jetson Nano
Тип пристрою	Мікроконтролер	Мікроконтролер	Одноплатний ПК	Одноплатний ПК
Операційна система	Ні	Ні	Linux	Linux
Підключення камери	Обмежене	Обмежене	Так	Так
Підтримка комп'ютерного зору	Ні	Частково	Так	Так
Wi-Fi	Ні	Так	Так	Так
Вартість	Низька	Низька	Середня	Висока

Порівняння розглянутих платформ показало, що для реалізації програмно-апаратного комплексу сортування поштових відправлень найбільш доцільним є використання Raspberry Pi 5 [17]. Дана платформа має достатню продуктивність для роботи з камерою, розпізнавання QR-кодів та керування мобільною платформою. Крім того, важливою перевагою є широкий вибір програмних бібліотек і відносно невисока вартість.

Важливою перевагою Raspberry Pi 5 є підтримка широкого спектра сучасних мов програмування та бібліотек, що використовуються під час розробки робототехнічних систем. Зокрема, платформа підтримує роботу з бібліотеками OpenCV, Pyzbar, Flask і Tkinter, які були використані під час створення програмного забезпечення комплексу. Крім того, Raspberry Pi 5 має розвинені мережеві можливості, що дозволяє організувати обмін даними між мобільною платформою та програмою оператора через локальну комп'ютерну мережу.

Таким чином, у межах даної кваліфікаційної роботи Raspberry Pi 5 обрано як основну апаратну платформу для створення мобільного робота та реалізації системи автоматизованого сортування поштових відправлень.

1.8 Аналіз існуючого програмного забезпечення та формування вимог до програмного забезпечення комплексу сортування поштових відправлень

Перед початком розробки програмного забезпечення доцільно провести аналіз існуючих програмних рішень, що використовуються для керування мобільними роботами та автоматизації логістичних процесів. Проведення такого аналізу дозволяє визначити основні функціональні можливості сучасних програмних продуктів, оцінити їх переваги та недоліки, а також сформувати перелік вимог до програмного забезпечення, що розробляється. Крім того, аналіз існуючих рішень дає можливість уникнути дублювання вже реалізованих функцій та використати сучасні підходи до побудови архітектури програмних систем [25–29].

Однією з найбільш поширених платформ для розробки програмного забезпечення мобільних роботів є Robot Operating System (ROS) (рисунок 1.11). Дана програмна платформа являє собою набір програмних бібліотек та інструментів, призначених для створення робототехнічних систем різного рівня складності. ROS підтримує модульний принцип побудови програмного забезпечення, що дозволяє розділити функціональні можливості системи на окремі незалежні компоненти. Серед основних можливостей платформи можна виділити підтримку роботи з датчиками, камерами, навігаційними алгоритмами, системами позиціонування та мережевою взаємодією між окремими програмними модулями [25,26].

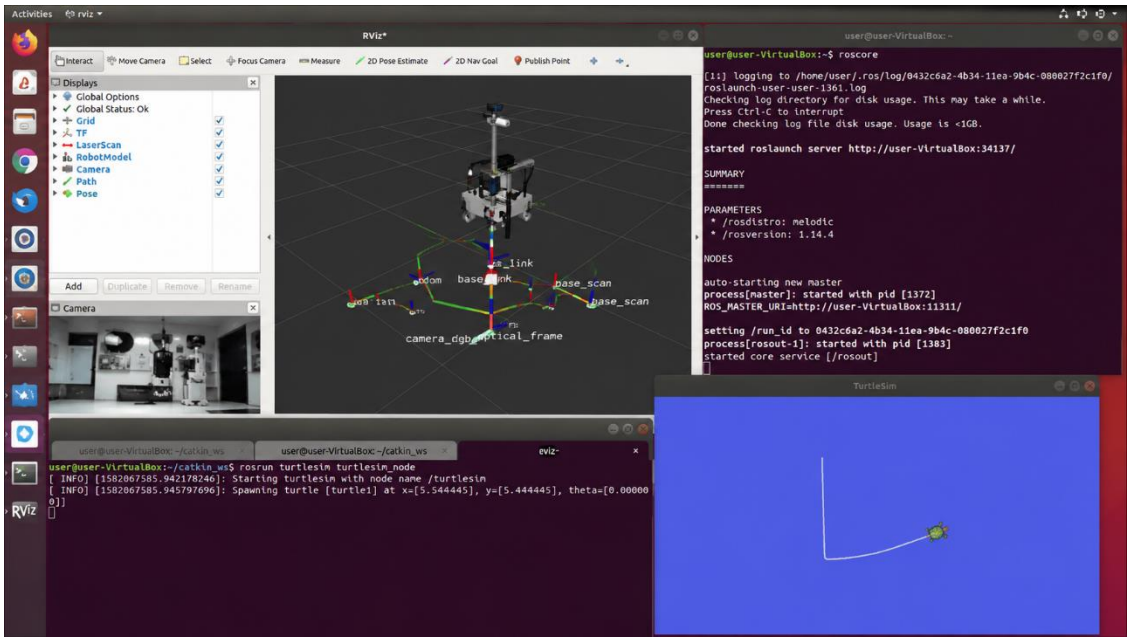


Рисунок 1.11 – Інтерфейс програмного середовища Robot Operating System (ROS)

Незважаючи на широкі функціональні можливості, використання ROS потребує значної кількості обчислювальних ресурсів, має складний процес налаштування та вимагає від розробника глибоких знань архітектури системи. Для невеликих мобільних роботів, які виконують вузькоспеціалізовані задачі, використання повної платформи ROS не завжди є доцільним.

Іншим сучасним програмним рішенням є RoboDK (рисунок 1.12), яке використовується для моделювання, програмування та тестування роботизованих систем. Програмне забезпечення дозволяє створювати цифрові моделі роботів, перевіряти траєкторії руху, програмувати виробничі операції та виконувати візуальне моделювання роботи роботизованих комплексів. Основною перевагою RoboDK є зручний графічний інтерфейс користувача та можливість інтеграції з великою кількістю промислових роботів [27].

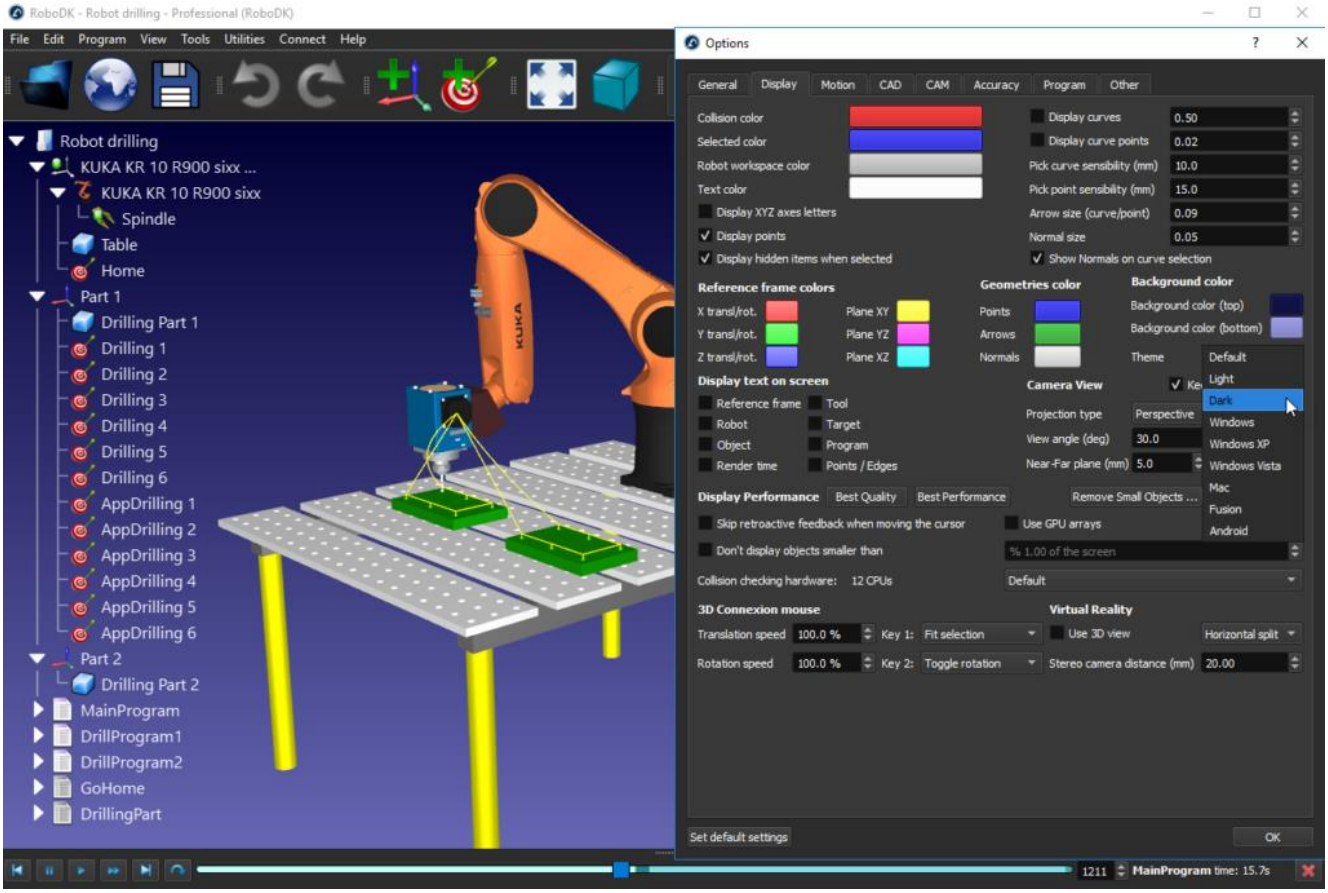


Рисунок 1.12 – Інтерфейс програмного забезпечення RoboDK

Разом із тим RoboDK орієнтований переважно на промислову робототехніку та автоматизацію виробничих процесів. Засоби роботи із системами комп’ютерного зору, QR-навігацією та мобільними платформами реалізовані обмежено, що ускладнює його використання для задач автоматизованого сортування поштових відправлень.

Для реалізації функцій комп’ютерного зору в сучасних робототехнічних системах широко використовується бібліотека OpenCV. Вона містить велику кількість алгоритмів обробки зображень, роботи з відеопотоком, пошуку об’єктів, визначення геометричних параметрів та розпізнавання маркерів [18,19,28].

Саме OpenCV використовується як основний інструмент обробки відео у розробленому програмному забезпеченні. Спільно з бібліотекою Pyzbar вона забезпечує розпізнавання QR-кодів, визначення їх координат, оцінку відстані до

контрольної точки та розрахунок кутового відхилення мобільного робота [18,19,21,28,29].

На основі проведеного аналізу можна зробити висновок, що жодне із розглянутих програмних рішень повністю не задовольняє вимоги, поставлені до комплексу сортування поштових відправлень. Універсальні програмні платформи мають широкий функціонал, проте характеризуються високою складністю налаштування та надлишковими можливостями для реалізації поставленої задачі. Саме тому було прийнято рішення щодо розробки власного спеціалізованого програмного забезпечення, орієнтованого на виконання конкретних логістичних операцій.

Проведений аналіз існуючих програмних рішень дозволив сформувати основні вимоги до програмного забезпечення комплексу сортування поштових відправлень. Оскільки розроблювана система призначена для керування мобільною роботизованою платформою в умовах складського приміщення, програмне забезпечення повинно забезпечувати не лише взаємодію з апаратною частиною комплексу, а й реалізовувати алгоритми навігації, обробки відеоданих та підтримувати зручний графічний інтерфейс користувача.

Порівняльний аналіз розглянутих програмних засобів наведено в таблиці 1.3.

Таблиця 1.3 – Порівняльний аналіз існуючого програмного забезпечення для керування мобільними роботами

Програмне забезпечення	Основні функції	Переваги	Недоліки
ROS	Навігація, взаємодія з датчиками, модульна архітектура	Велика кількість готових модулів, відкритий код	Складне налаштування, високі вимоги до ресурсів

«Продовження таблиці 1.3»

Програмне забезпечення	Основні функції	Переваги	Недоліки
RoboDK	Моделювання та програмування роботів	Зручний графічний інтерфейс, підтримка промислових роботів	Обмежена підтримка мобільної навігації та QR-кодів
OpenCV + Pyzbar	Обробка відео та розпізнавання QR-кодів	Висока швидкодія, гнучкість, простота інтеграції	Не забезпечує керування роботом без додаткових модулів
Розроблене програмне забезпечення	QR-навігація, калібрування, керування мобільним роботом	Орієнтація на конкретну задачу, простий інтерфейс, модульна структура	Обмежене використання поза межами поставленої задачі

У процесі проєктування програмного забезпечення були сформовані функціональні та нефункціональні вимоги, які визначають основні характеристики та можливості системи.

До функціональних вимог програмного забезпечення належать:

– авторизація користувача та доступ до основних функцій системи;
– встановлення з’єднання з мобільним роботом через локальну мережу;
– отримання та відображення відеопотоку з камери робота;
– автоматичне розпізнавання QR-кодів;
– визначення відстані та кутового відхилення до навігаційної мітки;

– виконання процедури автоматичного калібрування робочої зони;

– побудова карти контрольних точок;

– створення транспортних завдань оператором;

– передача команд керування мобільному роботу;

– відображення поточного стану роботи системи;

– ведення журналу подій та повідомлень.

Крім функціональних можливостей, програмне забезпечення повинно відповідати ряду нефункціональних вимог, які визначають його якість та ефективність використання.

До основних нефункціональних вимог належать:

– модульна структура програмного забезпечення;

– можливість подальшого розширення функціоналу;

– стабільна робота у режимі реального часу;

– висока швидкість обробки відеопотоку;

– надійний обмін даними між оператором та мобільним роботом;

– сумісність із Raspberry Pi 5;

– простий та інтуїтивно зрозумілий графічний інтерфейс;

– можливість роботи як у режимі симуляції, так і з реальним мобільним роботом.

Однією з найважливіших вимог до програмного забезпечення є забезпечення зручної взаємодії оператора із системою. Інтерфейс користувача повинен надавати можливість швидкого створення транспортних завдань, запуску процедури калібрування, перегляду відеопотоку з камери та контролю процесу виконання маршруту. Усі основні функції повинні бути доступними без переходу між великою кількістю вікон, що дозволяє скоротити час виконання операцій та підвищити ефективність роботи оператора.

Основні вимоги до програмного забезпечення наведено в таблиці 1.4.

Таблиця 1.4 – Основні вимоги до програмного забезпечення комплексу сортування поштових відправлень

Категорія вимог	Характеристика
Авторизація	Доступ користувача до програмного комплексу
Взаємодія з роботом	Підключення та передача команд через мережу
Навігація	Автоматичне розпізнавання QR-кодів та побудова маршруту
Калібрування	Автоматичне формування карти складського приміщення
Інтерфейс	Зручне графічне середовище керування
Швидкодія	Робота в режимі реального часу
Надійність	Стійкий обмін даними та обробка помилок
Масштабованість	Можливість додавання нових маршрутів та контрольних точок

На підставі сформованих вимог було визначено архітектуру розроблюваного програмного забезпечення та перелік його основних модулів. Зокрема, система повинна забезпечувати взаємодію між оператором і мобільним роботом, виконувати обробку відеоданих, розпізнавати QR-коди, здійснювати автоматичне калібрування робочої зони та керувати виконанням транспортних завдань. Саме ці вимоги були покладені в основу подальшого проєктування програмного забезпечення, що розглядається у другому розділі кваліфікаційної роботи [25–29].

1.9 Постановка задачі дослідження

Проведений аналіз сучасних систем автоматизації логістичних процесів показав, що одним із перспективних напрямів розвитку складських комплексів є використання мобільних роботизованих платформ. Застосування роботів дозволяє автоматизувати процеси транспортування вантажів, зменшити навантаження на персонал та підвищити ефективність роботи логістичних центрів.

У результаті дослідження існуючих методів навігації мобільних роботів було встановлено, що для навчальних та експериментальних систем доцільним є використання технології QR-навігації. Даний підхід характеризується простотою реалізації, невисокою вартістю та можливістю швидкої адаптації до зміни конфігурації складського приміщення [5–9].

Проведений аналіз апаратних платформ показав, що одноплатний компютер Raspberry Pi 5 забезпечує достатню продуктивність для реалізації алгоритмів комп’ютерного зору, розпізнавання QR-кодів та керування мобільною роботизованою платформою. Наявність підтримки сучасних програмних бібліотек дозволяє використовувати дану платформу як основу для побудови програмно-апаратного комплексу сортування поштових відправлень [12,13].

Метою кваліфікаційної роботи є розробка програмно-апаратного комплексу сортування поштових відправлень на базі мобільного робота з використанням технології QR-навігації.

Для досягнення поставленої мети необхідно вирішити такі задачі:

– провести аналіз сучасних систем автоматизації логістичних процесів та сортування поштових відправлень;

– дослідити існуючі методи навігації мобільних роботів та обґрунтувати вибір технології QR-навігації;

– розробити структуру програмно-апаратного комплексу сортування поштових відправлень;

– розробити програмне забезпечення для керування мобільним роботом та обробки інформації з QR-кодів;

– реалізувати систему навігації мобільного робота на основі технології комп'ютерного зору;

– провести експериментальні дослідження та оцінити ефективність роботи розробленого комплексу.

Результатом виконання роботи має стати програмно-апаратний комплекс, який забезпечує автоматичне переміщення мобільного робота між складськими зонами, розпізнавання QR-кодів та виконання завдань із сортування поштових відправлень.

2 ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ СОРТУВАННЯ ПОШТОВИХ ВІДПРАВЛЕНЬ

Розділ присвячений проєктуванню програмно-апаратного комплексу сортування поштових відправлень на базі мобільної роботизованої платформи. У розділі розглянуто загальну структуру системи, архітектуру програмного забезпечення, принципи взаємодії між оператором та мобільним роботом, а також особливості реалізації системи навігації з використанням QR-кодів.

Особливу увагу приділено проєктуванню основних програмних модулів, які забезпечують формування та виконання завдань, обробку інформації від камери, розпізнавання QR-кодів, побудову карти контрольних точок та взаємодію з мобільною роботизованою платформою. Крім того, у розділі наведено алгоритми калібрування системи та сортування поштових відправлень, які реалізовані в межах розробленого програмного забезпечення.

2.1 Загальна структура програмно-апаратного комплексу

На основі проведеного аналізу існуючих систем автоматизації логістичних процесів було розроблено структуру програмно-апаратного комплексу сортування поштових відправлень. Основним призначенням комплексу є автоматизація переміщення відправлень між складськими зонами за допомогою мобільної роботизованої платформи та програмного забезпечення оператора.

Розроблена система складається з двох основних компонентів: програмного забезпечення оператора та мобільного робота. Програма оператора використовується для створення та контролю завдань, відображення інформації про стан системи та моніторингу виконання операцій. Мобільна платформа, у свою чергу, відповідає за переміщення між контрольними точками та взаємодію з навігаційними мітками (рисунок 2.1) [21].

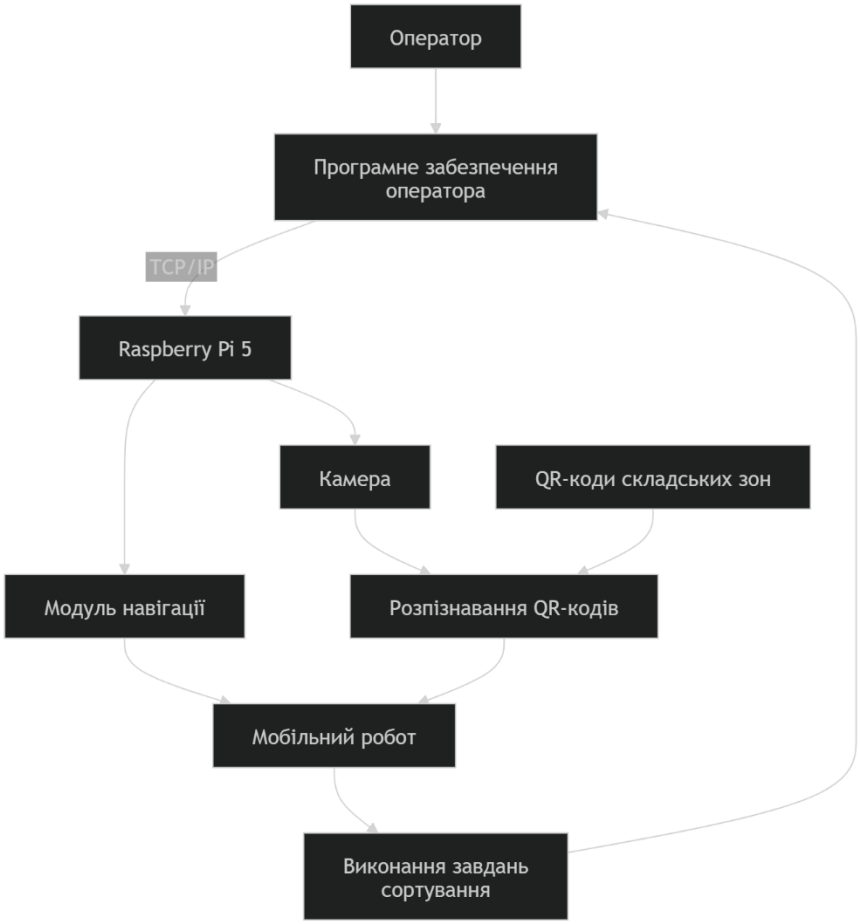


Рисунок 2.1 – Загальна структура програмно-апаратного комплексу сортування поштових відправлень

Обмін даними між програмою та роботом здійснюється через локальну мережу з використанням протоколу TCP/IP. Оператор може передавати команди на виконання певних дій, а робот надсилає інформацію про свій поточний стан і результати виконання поставлених завдань.

Для навігації в системі використовуються QR-коди, розміщені у контрольних точках складського приміщення. Під час початкового налаштування робот виконує сканування QR-кодів і отримує інформацію про розташування окремих зон. Надалі ці дані використовуються для переміщення між заданими пунктами маршруту [13].

Програмна частина комплексу містить модуль керування завданнями, модуль обміну даними з роботом, систему журналювання подій, засоби обробки QR-кодів та графічний інтерфейс користувача. Поділ на окремі модулі спрощує подальше розширення функціональних можливостей системи та її модернізацію.

Під час роботи оператор формує завдання на транспортування відправлення до необхідної складської зони. Після отримання команди робот виконує переміщення між контрольними точками та повідомляє про завершення операції. У разі виникнення помилок або неможливості зчитування навігаційних міток відповідна інформація відображається у програмі оператора.

Таким чином, запропонована структура програмно-апаратного комплексу забезпечує взаємодію між програмною частиною та мобільною роботизованою платформою, що дає можливість автоматизувати процес сортування поштових відправлень.

2.2 Проєктування архітектури програмного забезпечення

Програмне забезпечення є основною складовою розробленого комплексу сортування поштових відправлень. Воно забезпечує взаємодію оператора з мобільною роботизованою платформою, дозволяє створювати та виконувати завдання, контролювати роботу системи та отримувати інформацію про поточний стан робота.

Під час розроблення було обрано модульний принцип побудови програмного забезпечення. Такий підхід дає можливість розділити систему на окремі функціональні частини, що спрощує процес розробки, тестування та подальшого вдосконалення програмного комплексу [21,25].

До складу програмного забезпечення входять модуль інтерфейсу користувача, модуль керування завданнями, модуль обміну даними з роботом, модуль обробки QR-кодів та система журналювання подій.

Модуль інтерфейсу користувача призначений для взаємодії оператора з програмою. За його допомогою можна реєструвати поштові відправлення, створювати завдання для робота, запускати процедуру калібрування, переходити до режиму ручного керування та контролювати стан виконання операцій (рисунок 2.2).

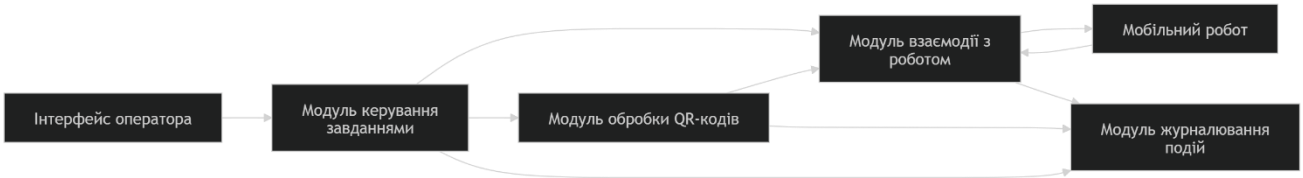


Рисунок 2.2 – Структура програмного забезпечення комплексу сортування поштових відправлень

Модуль керування завданнями відповідає за формування та обробку черги завдань. Після створення нового завдання система визначає необхідні параметри переміщення та передає відповідну інформацію мобільній платформі.

Обмін даними між програмою оператора та роботом здійснюється через локальну мережу з використанням протоколу TCP/IP. Модуль взаємодії з роботом забезпечує передачу команд, отримання службової інформації та контроль стану з'єднання [21].

Для реалізації навігації використовується модуль обробки QR-кодів. Під час роботи система отримує відеопотік із камери та виконує пошук і розпізнавання QR-кодів за допомогою бібліотеки OpenCV. Отримані дані використовуються для визначення поточного положення робота та виконання поставлених завдань [18,19,21,28].

Окремим компонентом програмного забезпечення є система журналювання подій. У журналі відображаються повідомлення про запуск програми, виконання команд, результати роботи робота та можливі помилки. Це спрощує контроль роботи системи та полегшує пошук несправностей.

Застосування модульної архітектури дозволило розділити окремі функції програми та спростити подальшу модернізацію системи. За необхідності структура програмного забезпечення може бути доповнена новими компонентами без суттєвих змін уже реалізованих модулів. Таким чином, розроблена архітектура програмного забезпечення забезпечує взаємодію всіх компонентів комплексу та створює основу для реалізації автоматизованого сортування поштових відправлень (рисунок 2.3).

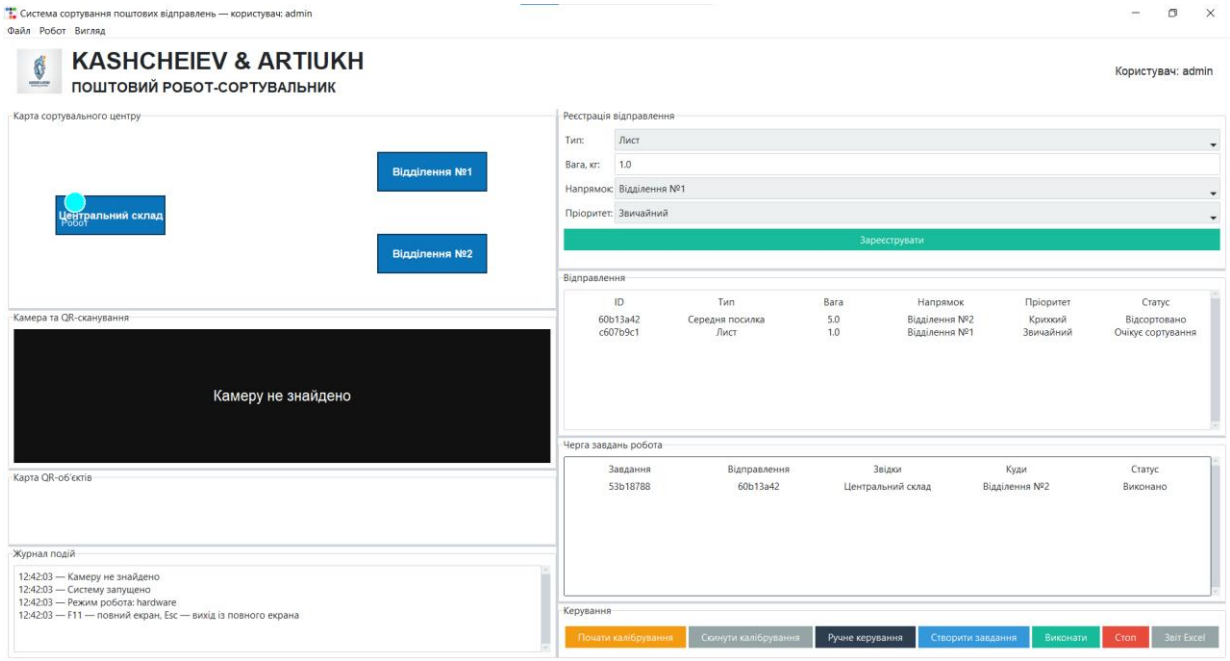


Рисунок 2.3 – Головне вікно програмного забезпечення оператора

2.3 Проєктування системи QR-навігації

Одним із важливих компонентів розробленого програмно-апаратного комплексу є система навігації мобільного робота. Для забезпечення переміщення між складськими зонами було обрано навігацію на основі QR-кодів. Такий підхід не потребує використання дорогих систем позиціонування та дозволяє реалізувати навігацію за допомогою камери й програмних засобів обробки зображень.

Принцип роботи системи полягає у використанні QR-кодів як контрольних точок маршруту. Кожен QR-код містить інформацію про певну складську зону або пункт призначення [13]. Під час руху робот виконує пошук QR-кодів за допомогою камери, після чого система визначає поточне положення платформи та приймає рішення щодо подальшого переміщення (рисунок 2.4) [12,13].

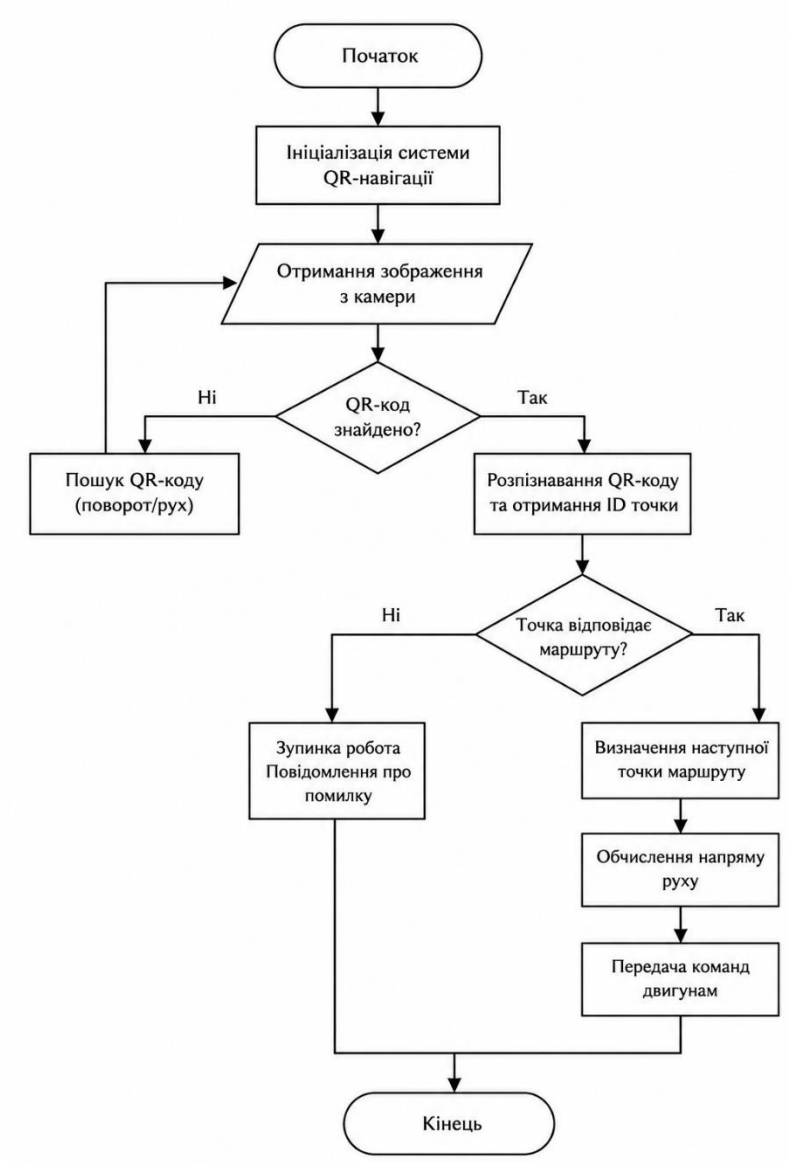


Рисунок 2.4 – Загальний принцип роботи системи QR-навігації

Для реалізації навігації використовується камера, підключена до одноплатного комп’ютера Raspberry Pi 5. Відеопотік надходить до програмного модуля обробки зображень, де за допомогою бібліотеки OpenCV виконується пошук і розпізнавання QR-кодів. Отримані дані використовуються для визначення поточного положення робота та виконання навігаційних команд.

У системі передбачено декілька контрольних точок, які відповідають окремим зонам сортувального центру. Кожна точка має власний QR-код, що містить інформацію про відповідну ділянку маршруту. Це дозволяє організувати

переміщення робота між складськими зонами та виконувати завдання сортування (рисунок 2.5).

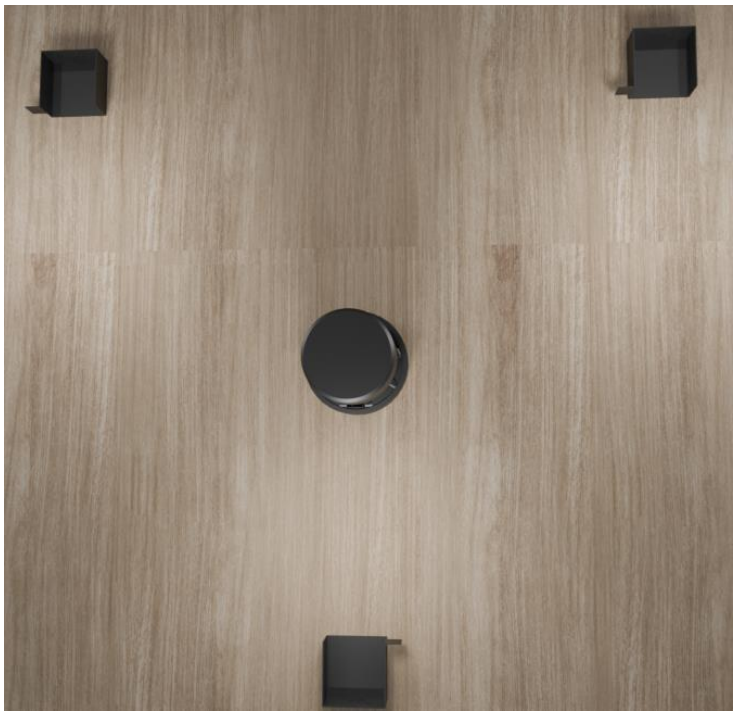


Рисунок 2.5 – Розміщення QR-кодів у межах сортувального центру

Під час калібрування робот виконує пошук QR-кодів та формує карту контрольних точок. Отримана інформація зберігається та надалі використовується під час автоматичного виконання завдань. У разі зміни конфігурації приміщення достатньо повторно виконати процедуру калібрування.

Однією з переваг використання QR-навігації є простота реалізації та можливість швидкого внесення змін до маршруту без модернізації апаратної частини системи. Для зміни конфігурації достатньо змінити розташування QR-кодів або додати нові контрольні точки (рисунок 2.6).

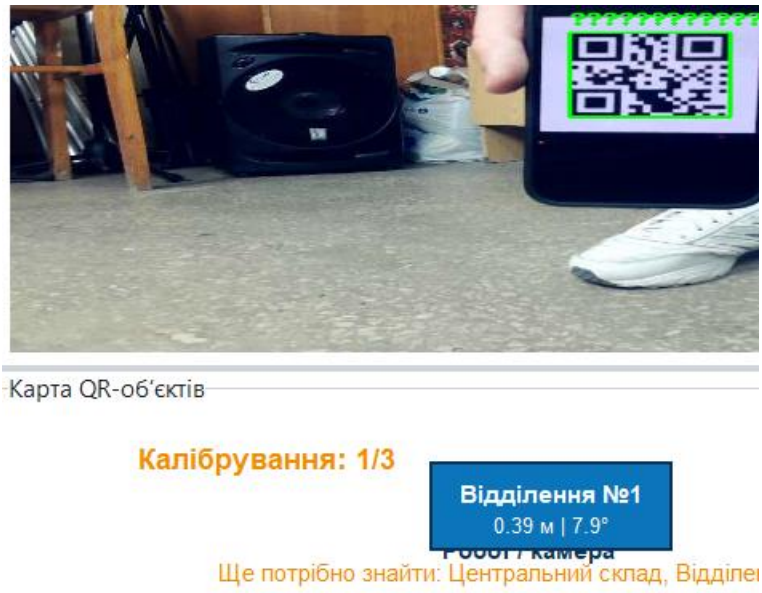


Рисунок 2.6 – Приклад зчитування QR-коду камерою мобільного робота

Таким чином, використання QR-навігації дозволило реалізувати просту та достатньо точну систему позиціонування мобільної платформи, яка забезпечує виконання завдань автоматизованого сортування поштових відправлень.

2.4 Проєктування алгоритму калібрування системи

Перед початком виконання завдань мобільний робот повинен отримати інформацію про розташування контрольних точок у межах сортувального центру. Для цього в програмному забезпеченні реалізовано режим калібрування, який дозволяє сформувати карту складських зон на основі QR-кодів.

Основною метою калібрування є визначення доступних контрольних точок та збереження інформації про них для подальшого використання під час навігації. Завдяки цьому робот може виконувати переміщення між заданими зонами без необхідності ручного налаштування маршруту.

Процедура калібрування запускається оператором через інтерфейс програми за допомогою відповідної команди. Після запуску робот переходить у режим пошуку QR-кодів. Зображення з камери передається до модуля комп'ютерного зору, який виконує пошук і розпізнавання навігаційних міток [18,19].

Після виявлення QR-коду система визначає назву відповідної контрольної точки та зберігає отримані дані. У межах розробленого комплексу використовуються QR-коди, що відповідають зонам «Центральний склад», «Відділення № 1» та «Відділення № 2» (рисунок 2.7).

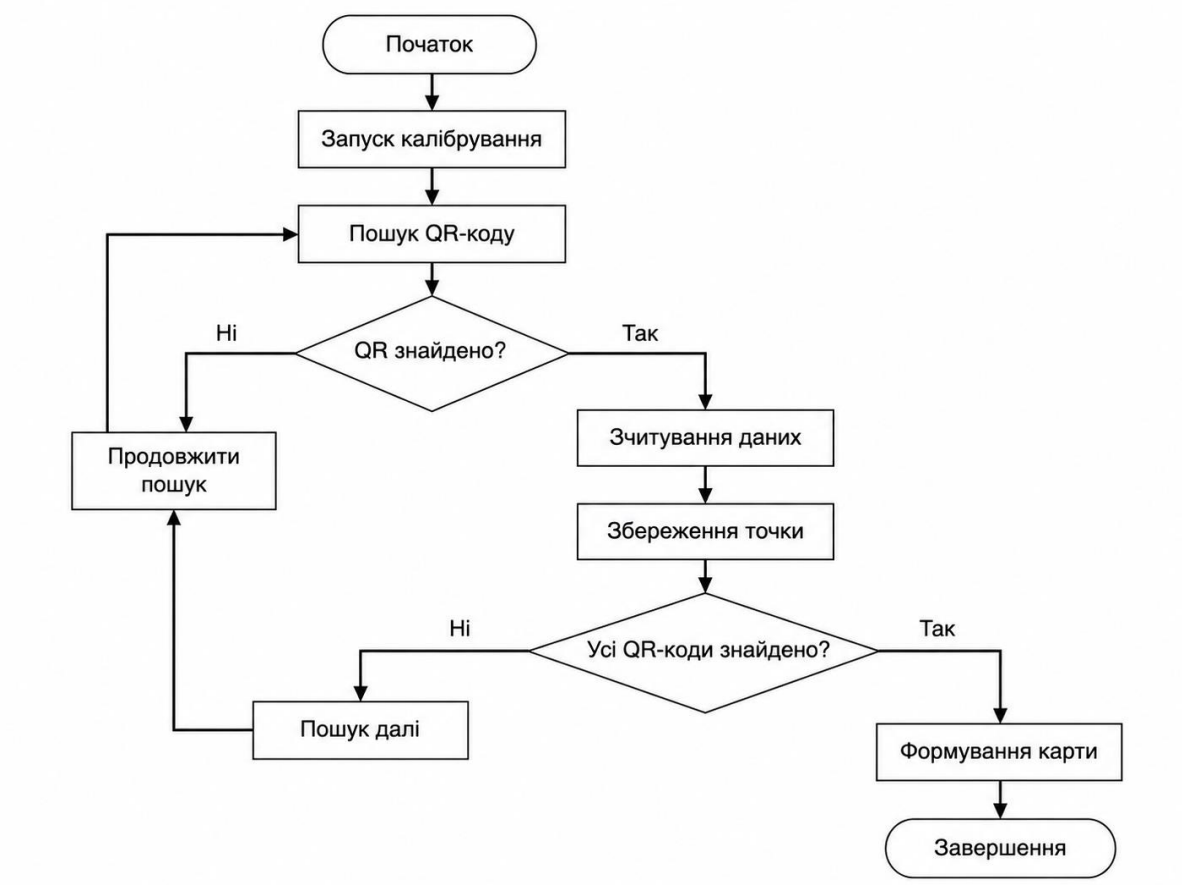


Рисунок 2.7 – Алгоритм калібрування системи

Після завершення процедури калібрування програмне забезпечення формує карту складських зон. Отримана інформація використовується під час виконання транспортних завдань та може бути повторно використана при наступному запуску системи.

Перевагою такого підходу є можливість швидкого переналаштування комплексу. У разі зміни розташування QR-кодів достатньо повторно виконати калібрування, після чого система автоматично оновить карту контрольних точок (рисунок 2.8).

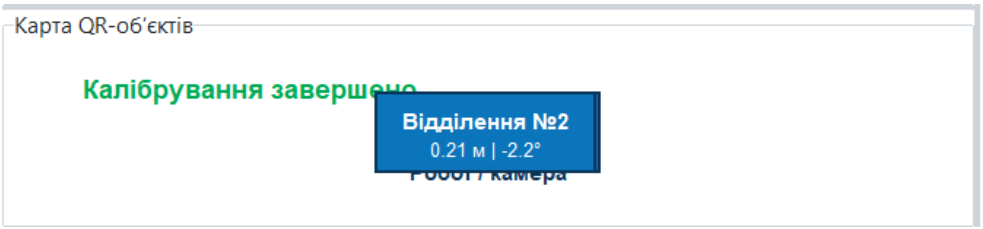


Рисунок 2.8 – Карта складських зон після завершення калібрування

Після успішного завершення калібрування система переходить у режим виконання транспортних завдань та повідомляє оператора про готовність до роботи.

Таким чином, реалізований алгоритм калібрування дозволяє автоматично визначати розташування контрольних точок і забезпечує подальшу роботу системи навігації мобільного робота [13].

2.5 Розробка алгоритму сортування поштових відправлень

Після завершення процедури калібрування та формування карти складських зон комплекс переходить до виконання основної функції – сортування поштових відправлень. Даний процес реалізується за допомогою програмного забезпечення оператора та мобільної роботизованої платформи, які взаємодіють між собою через локальну мережу.

Перед початком роботи оператор створює транспортне завдання через інтерфейс програми. Під час створення завдання вказується пункт призначення, до якого необхідно доставити відправлення. Після підтвердження введені дані передаються до модуля керування, який визначає відповідну складську зону та послідовність переміщення між контрольними точками.

На основі інформації, отриманої під час калібрування, формується маршрут руху мобільного робота. Після цього програмне забезпечення передає необхідні команди, а робот розпочинає виконання транспортного завдання.

Під час переміщення робот постійно аналізує відеопотік із камери та виконує пошук QR-кодів. Після виявлення чергової контрольної точки здійснюється зчитування та перевірка отриманої інформації. Якщо розпізнаний QR-код відповідає очікуваній точці маршруту, робот продовжує рух до наступної зони (рисунок 2.9) [13].

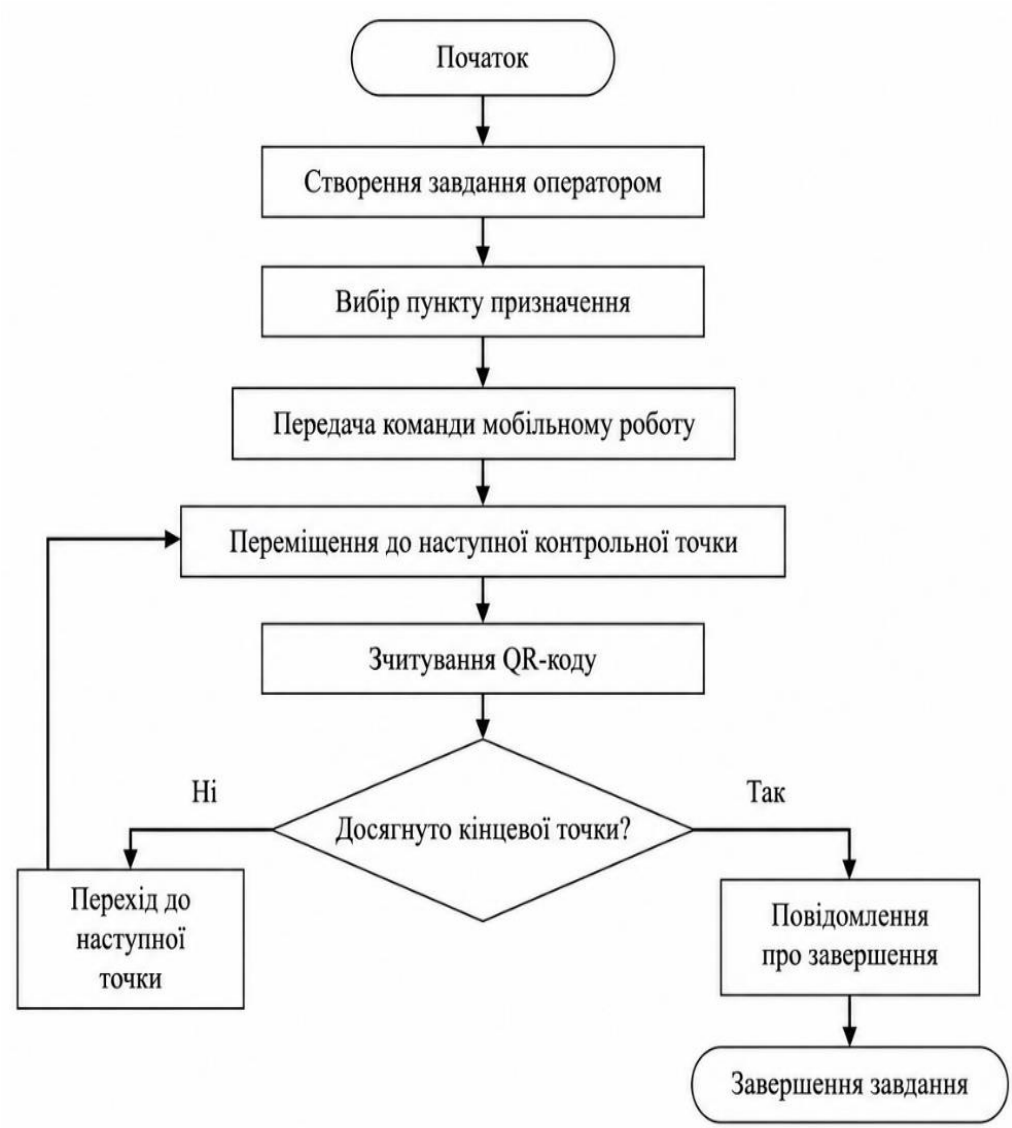


Рисунок 2.9 – Алгоритм сортування поштових відправлень

Якщо під час руху QR-код не було виявлено або виникла помилка навігації, робот переходить у режим повторного пошуку маркера. Це дозволяє підвищити надійність роботи комплексу та забезпечити коректне виконання поставленого завдання.

Після прибуття до кінцевої точки виконується перевірка відповідності пункту призначення створеному завданню. У разі успішного завершення операції відправлення вважається доставленим, а в інтерфейсі програми відображається повідомлення про успішне виконання завдання (рисунок 2.10) [21].

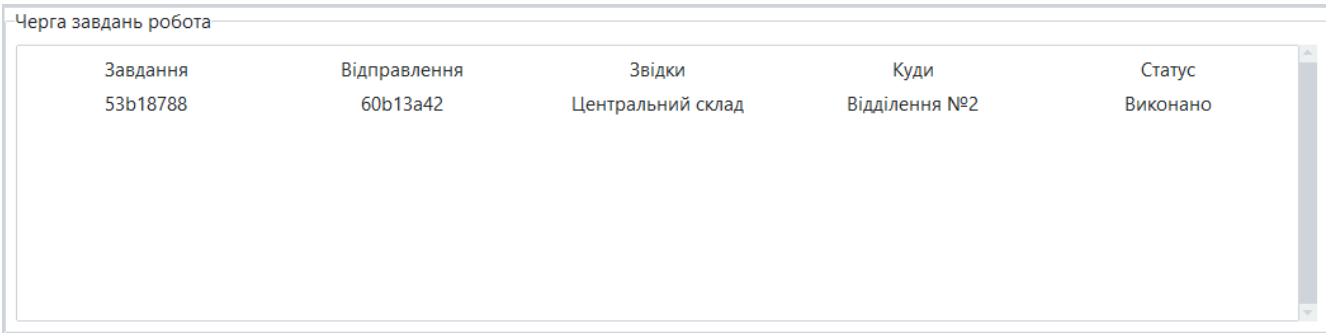


Рисунок 2.10 – Послідовність виконання транспортного завдання

Розроблений алгоритм дозволяє автоматизувати процес переміщення поштових відправлень між складськими зонами без необхідності постійного ручного керування роботом. Використання QR-кодів як контрольних точок спрощує реалізацію системи навігації та забезпечує достатню точність виконання транспортних операцій.

Таким чином, реалізований алгоритм сортування забезпечує виконання основних функцій програмно-апаратного комплексу та використовується під час роботи мобільної платформи для автоматизованого переміщення поштових відправлень.

2.6 Проєктування взаємодії програмного забезпечення з мобільним роботом

Для забезпечення роботи комплексу сортування поштових відправлень необхідно організувати обмін даними між програмою оператора та мобільною роботизованою платформою. Взаємодія між компонентами системи реалізується через локальну мережу з використанням клієнт-серверного принципу обміну даними.

У розробленому комплексі програмне забезпечення оператора відповідає за формування та передачу команд, а програмне забезпечення, встановлене на Raspberry Pi 5, забезпечує їх обробку та керування мобільною платформою. Після встановлення з’єднання оператор отримує можливість контролювати роботу робота та отримувати інформацію про його поточний стан.

Для обміну даними використовується протокол TCP/IP. Даний протокол забезпечує стабільний двосторонній зв’язок між програмою оператора та мобільною платформою, що дозволяє передавати команди керування та службові повідомлення (рисунок 2.11) [21].



Рисунок 2.11 – Схема взаємодії програмного забезпечення з мобільним роботом

У процесі роботи оператор може передавати команди ручного керування, запускати процедуру калібрування та створювати транспортні завдання. Після отримання команди робот виконує відповідну дію та надсилає інформацію про результат її виконання.

Крім команд керування, мобільна платформа передає інформацію про поточний стан системи, результати зчитування QR-кодів та повідомлення про можливі помилки. Отримані дані відображаються в інтерфейсі оператора та записуються до журналу подій.

Для візуального контролю роботи робота використовується відеопотік з камери, встановленої на мобільній платформі. Зображення передається через мережу та відображається у вікні програми в режимі реального часу, що дозволяє оператору контролювати процес навігації та виконання транспортних завдань (рисунок 2.12).



Рисунок 2.12 – Обмін даними між програмою оператора та мобільним роботом

Для підвищення надійності роботи передбачено контроль мережевого з’єднання. У разі втрати зв’язку оператор отримує відповідне повідомлення, а після відновлення з’єднання робота системи може бути продовжена без повторного запуску програмного комплексу.

Таким чином, реалізований механізм взаємодії забезпечує дистанційне керування мобільною платформою, передачу службової інформації та підтримує роботу системи автоматизованого сортування поштових відправлень [20].

2.7 Проєктування інтерфейсу оператора

Ефективність роботи програмно-апаратного комплексу значною мірою залежить від зручності взаємодії користувача із системою. Тому під час розроблення програмного забезпечення особливу увагу було приділено створенню простого та зрозумілого інтерфейсу оператора, який забезпечує керування мобільною платформою, контроль виконання завдань та моніторинг поточного стану системи.

Інтерфейс оператора реалізовано у вигляді графічного застосунку, що запускається на персональному комп’ютері. Усі основні функції комплексу об'єднані в одному робочому вікні, що дозволяє оператору виконувати керування системою без використання додаткового програмного забезпечення.

Головне вікно програми містить елементи підключення до робота, засоби запуску основних режимів роботи, інформацію про стан системи та область відображення поточних даних. За допомогою інтерфейсу оператор може запускати калібрування, виконувати ручне керування та контролювати процес сортування поштових відправлень (рисунок 2.13) [21].

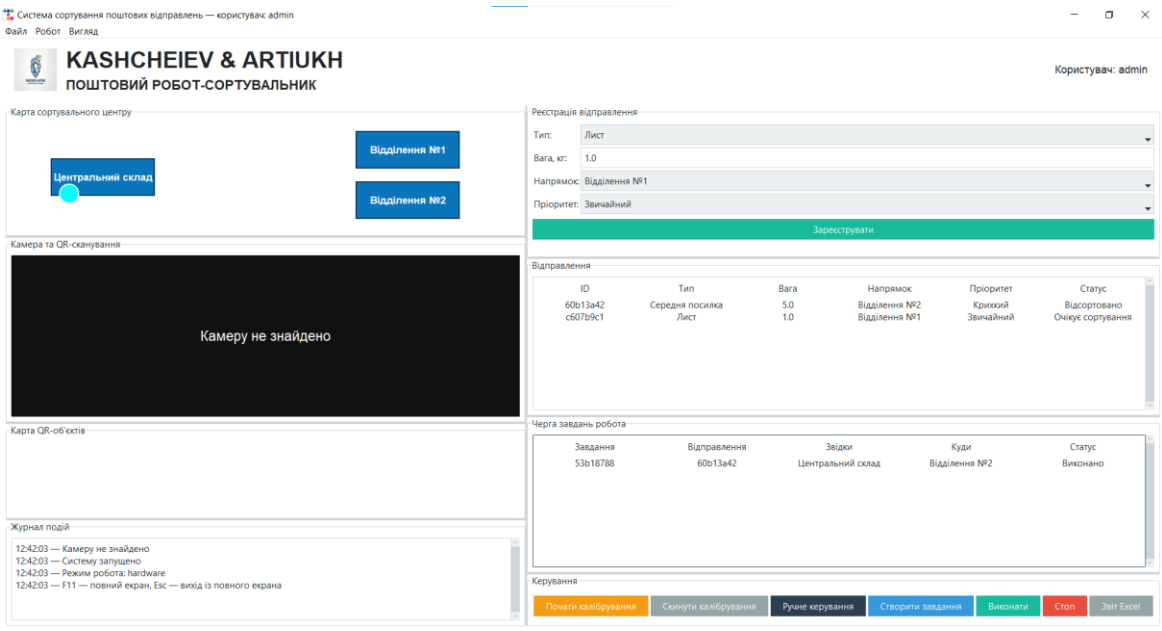


Рисунок 2.13 – Головне вікно програмного забезпечення

Для виконання транспортних операцій у програмі передбачено модуль роботи із завданнями. Користувач може створювати нові завдання, обирати пункт призначення та контролювати процес їх виконання. Інформація про активні завдання відображається в режимі реального часу.

Окремий режим призначений для ручного керування мобільною платформою. Він використовується під час тестування обладнання, перевірки працездатності системи та виконання сервісних операцій. Оператор може надсилати команди руху та контролювати переміщення робота (рисунок 2.14).

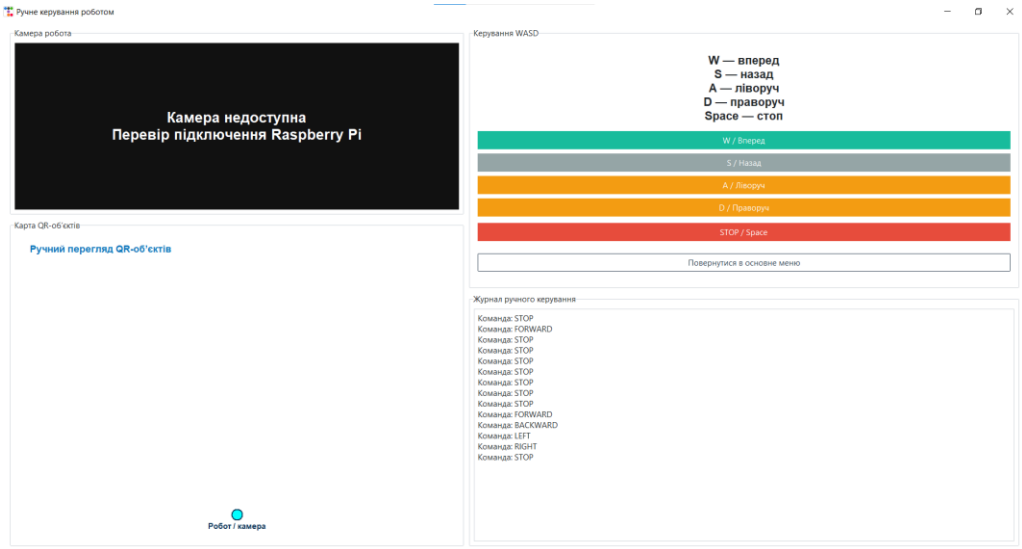


Рисунок 2.14 – Вікно ручного керування мобільним роботом

Для контролю навігації та роботи системи використовується відеопотік з камери, встановленої на роботі. Зображення передається через локальну мережу та відображається у вікні програми, що дозволяє спостерігати за процесом переміщення та пошуку QR-кодів.

Крім засобів керування, програмне забезпечення містить журнал подій, у якому відображаються повідомлення про підключення до робота, запуск калібрування, виконання команд, результати зчитування QR-кодів та можливі помилки. Наявність журналу спрощує контроль роботи комплексу та пошук несправностей (рисунок 2.15) [21].



Рисунок 2.15 – Журнал подій системи

Під час проєктування інтерфейсу основна увага приділялася простоті використання та швидкому доступу до основних функцій системи. Завдяки цьому оператор може оперативно контролювати роботу комплексу та керувати процесом сортування поштових відправлень.

Таким чином, розроблений інтерфейс забезпечує зручну взаємодію користувача з програмно-апаратним комплексом та створює умови для ефективного керування мобільною роботизованою платформою.

У другому розділі було виконано проєктування програмно-апаратного комплексу сортування поштових відправлень на основі мобільної роботизованої платформи. Розроблено загальну структуру системи, архітектуру програмного забезпечення, алгоритми калібрування та сортування, а також систему QR-навігації й інтерфейс оператора.

У результаті проєктування визначено основні принципи взаємодії між програмним забезпеченням і мобільним роботом, що створює основу для практичної реалізації комплексу та подальшого дослідження його роботи.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМПЛЕКСУ СОРТУВАННЯ ПОШТОВИХ ВІДПРАВЛЕНЬ

У даному розділі розглянуто практичну реалізацію програмно-апаратного комплексу сортування поштових відправлень, спроєктованого у попередньому розділі. Описано вибір програмних засобів розробки, реалізацію основних функціональних модулів, організацію взаємодії між програмою оператора та мобільною роботизованою платформою, а також особливості реалізації системи QR-навігації.

Крім того, наведено опис графічного інтерфейсу користувача, засобів обробки відеопотоку та механізмів обміну даними між компонентами комплексу. Отримані результати дозволили реалізувати програмне забезпечення, яке забезпечує автоматизоване сортування поштових відправлень та керування мобільним роботом у межах складського приміщення.

3.1 Вибір засобів розробки програмного забезпечення

Для реалізації програмного забезпечення комплексу сортування поштових відправлень було обрано мову програмування Python. Дана мова широко використовується у робототехніці, системах автоматизації та комп’ютерному зорі. Крім того, Python має велику кількість готових бібліотек для роботи з мережевими протоколами, обробкою зображень і створенням графічних інтерфейсів.

Розробка програмного забезпечення виконувалася у середовищі PyCharm, яке підтримує роботу з Python, містить інструменти налагодження та дозволяє зручно працювати з проєктами різної складності. Для реалізації системи комп'ютерного зору використовується бібліотека OpenCV (рисунок 3.1). Вона застосовується для отримання відеопотоку з камери, обробки зображень та розпізнавання QR-кодів у режимі реального часу.



Рисунок 3.1 – Використання бібліотеки OpenCV для обробки зображень

Для створення графічного інтерфейсу оператора використовуються бібліотеки Tkinter та ttkbootstrap. Бібліотека Tkinter входить до стандартного набору Python, а ttkbootstrap використовується для покращення зовнішнього вигляду елементів інтерфейсу.

Обмін даними між програмою оператора та мобільною платформою реалізовано за допомогою протоколу TCP/IP. Це дозволяє передавати команди керування та отримувати інформацію про поточний стан робота.

Передача відеопотоку з камери реалізована за допомогою вебфреймворку Flask. Сервер Flask запускається на Raspberry Pi 5 та надає доступ до зображення через локальну мережу. Отриманий відеопотік використовується оператором для контролю роботи мобільної платформи.

Як апаратну платформу використовується одноплатний комп’ютер Raspberry Pi 5 під керуванням операційної системи Raspberry Pi OS. Продуктивності даної платформи достатньо для роботи алгоритмів комп'ютерного зору, мережевої взаємодії та керування мобільним роботом.

3.2 Реалізація структури програмного забезпечення

Програмне забезпечення комплексу сортування поштових відправлень реалізовано за модульним принципом. Такий підхід дозволив розділити

функціональність системи на окремі програмні компоненти, кожен із яких відповідає за виконання певних завдань. Це спрощує розроблення, тестування та подальше вдосконалення програмного забезпечення.

Структура програмного забезпечення включає модулі графічного інтерфейсу оператора (рисунок 3.2), керування завданнями, взаємодії з мобільним роботом, навігації, розпізнавання QR-кодів та передачі відеопотоку.

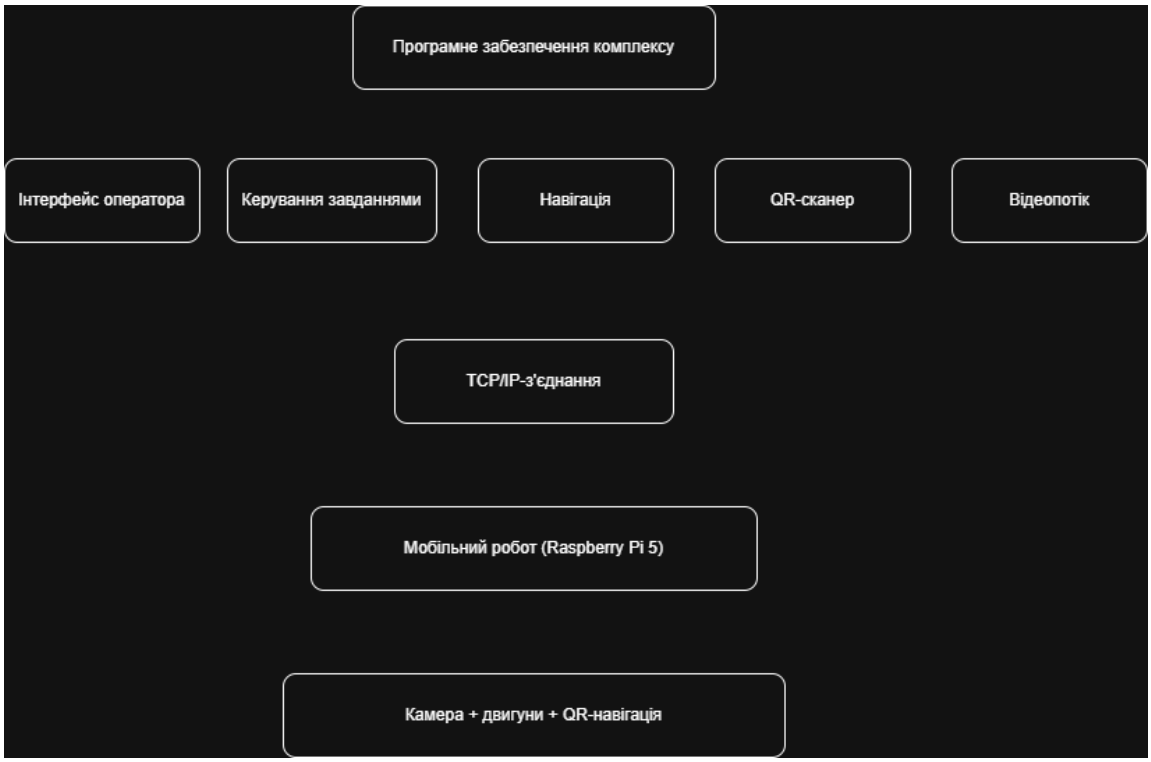


Рисунок 3.2 – Структура програмного забезпечення комплексу сортування поштових відправлень

Запуск програми здійснюється через головний модуль, який виконує ініціалізацію всіх необхідних компонентів системи та створює головне вікно застосунку.

Графічний інтерфейс оператора забезпечує доступ до основних функцій комплексу. Через нього користувач може виконувати підключення до робота, запускати калібрування, переглядати відеопотік, створювати транспортні завдання та контролювати стан системи.

Модуль керування завданнями відповідає за формування та виконання транспортних операцій. Він забезпечує вибір пункту призначення, зберігання інформації про завдання та передачу необхідних даних іншим компонентам системи.

Модуль взаємодії з мобільним роботом реалізує передачу команд через мережеве з’єднання та отримання інформації про стан роботизованої платформи.

Навігаційний модуль використовує інформацію, отриману під час калібрування, та забезпечує визначення послідовності проходження контрольних точок.

Модуль розпізнавання QR-кодів виконує обробку зображення з камери та визначення навігаційних міток. Отримані результати використовуються для контролю правильності переміщення мобільної платформи.

Для спостереження за роботою системи реалізовано модуль передачі відеопотоку, який забезпечує відображення зображення з камери робота в режимі реального часу.

На мобільній платформі функціонує окремий програмний модуль, який приймає команди від програми оператора та забезпечує їх виконання. Обмін даними між компонентами системи здійснюється через локальну мережу з використанням протоколу TCP/IP.

Таким чином, реалізована структура програмного забезпечення забезпечує взаємодію між інтерфейсом оператора, системою навігації, модулем розпізнавання QR-кодів та мобільною роботизованою платформою, що дозволяє реалізувати автоматизоване сортування поштових відправлень.

3.3 Розробка модуля взаємодії з мобільним роботом

Одним із ключових компонентів програмного забезпечення комплексу сортування поштових відправлень є модуль взаємодії з мобільним роботом. Його основним призначенням є забезпечення обміну даними між програмою оператора та роботизованою платформою через локальну мережу.

Для реалізації взаємодії використовується клієнт-серверна архітектура. Програма оператора виконує функції клієнта та формує команди керування, тоді як програмне забезпечення, встановлене на мобільному роботі, виконує функції сервера та обробляє отримані запити.

Передача інформації здійснюється за допомогою протоколу TCP/IP. Використання даного протоколу дозволяє забезпечити надійний обмін даними між пристроями та контроль доставки команд керування. Перед початком роботи програма оператора встановлює мережеве з'єднання з мобільною платформою та перевіряє доступність сервера (рисунок 3.3).

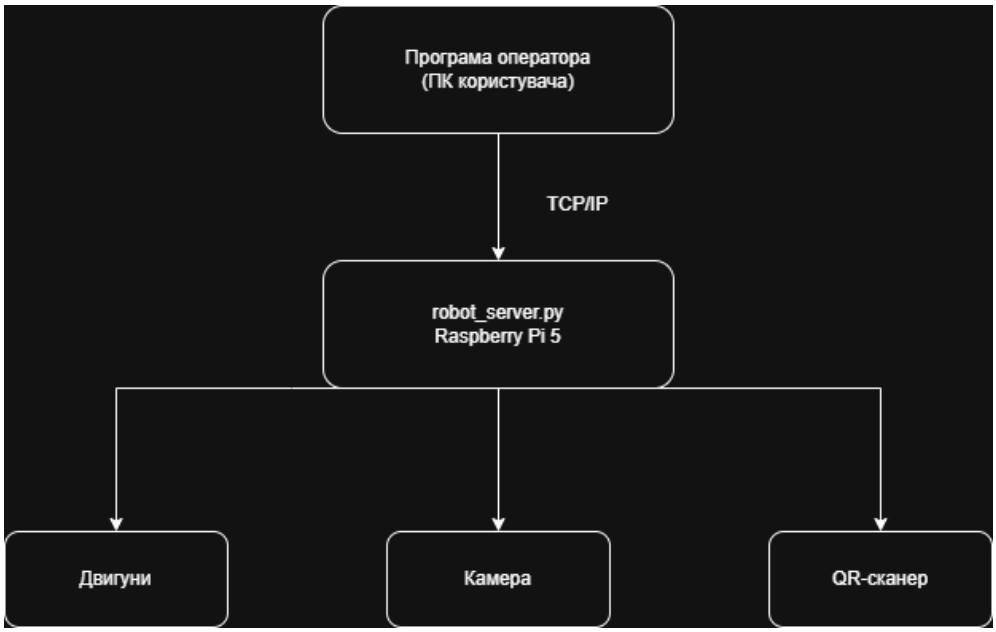


Рисунок 3.3 – Схема взаємодії програмного забезпечення оператора з мобільним роботом

У процесі роботи оператор може передавати роботу команди руху (таблиця 3.1) та виконання окремих функцій. До основних команд належать рух вперед, рух назад, поворот ліворуч, поворот праворуч, зупинка руху, запуск калібрування та запуск автоматичного виконання завдання.

Таблиця 3.1 – Основні команди керування мобільним роботом

Команда	Призначення
FORWARD	Рух вперед
BACKWARD	Рух назад
LEFT	Поворот ліворуч
RIGHT	Поворот праворуч
STOP	Зупинка робота
SCAN	Сканування QR-коду
CALIBRATION	Запуск калібрування
AUTO	Автоматичне виконання завдання

Після отримання команди серверна частина програмного забезпечення виконує її обробку та передає відповідну інформацію системі керування роботом. Після завершення операції формується повідомлення про результат виконання, яке повертається до програми оператора.

Крім команд керування модуль забезпечує передачу службової інформації. До неї належать повідомлення про стан підключення, результати розпізнавання QR-кодів, завершення калібрування, поточний режим роботи та інформація про помилки.

Для підвищення надійності роботи комплексу реалізовано механізм контролю стану мережевого з’єднання. У випадку втрати зв’язку система повідомляє оператора про виникнення помилки та призупиняє виконання поточного завдання до моменту відновлення підключення.

Таким чином, реалізований модуль взаємодії забезпечує стабільний обмін даними між програмою оператора та мобільною роботизованою платформою, що дозволяє здійснювати дистанційне керування, виконання транспортних завдань і моніторинг стану системи в режимі реального часу.

3.4 Розробка модуля обробки та розпізнавання QR-кодів

Для забезпечення навігації мобільного робота було розроблено програмний модуль роботи з камерою та розпізнавання QR-кодів. Даний модуль реалізовано у вигляді класу CameraModule, який відповідає за отримання відеопотоку, пошук QR-кодів, визначення відстані до маркера, розрахунок кута відхилення та передачу отриманих даних до інших частин програмного забезпечення.

Модуль використовує бібліотеки OpenCV, PIL та pyzbar. Бібліотека OpenCV застосовується для отримання кадрів із камери, зміни розміру зображення, нанесення графічних позначень та обробки відеопотоку. Бібліотека pyzbar використовується для декодування QR-кодів, а PIL забезпечує перетворення зображення у формат, придатний для відображення в графічному інтерфейсі оператора.

Основним етапом роботи модуля є запуск камери. Для цього використовується метод start(), у якому створюється об'єкт cv2.VideoCapture. Джерело відеосигналу задається через параметр CAMERA_SOURCE, що дозволяє використовувати як локальну камеру, так і інше джерело відеопотоку. Після успішного відкриття камери запускається окремий потік, у якому безперервно виконується зчитування кадрів.

Фрагмент запуску камери та відеопотоку:

```
def start(self):
    self.cap = cv2.VideoCapture(CAMERA_SOURCE)
    self.cap.set(cv2.CAP_PROP_BUFFERSIZE, 1)

    if not self.cap.isOpened():
        print("Не вдалося відкрити камеру:", CAMERA_SOURCE)
        return False

    self.running = True
```

```
    threading.Thread(
        target=self.capture_loop,
        daemon=True
    ).start()

    return True
```

Зчитування кадрів виконується у методі capture_loop(). У цьому методі камера постійно отримує нові кадри, після чого кожен кадр змінюється до розміру 640 пікселів × 480 пікселів та зберігається у змінній latest_frame. Використання окремого потоку дозволяє не блокувати роботу основного інтерфейсу програми та забезпечує безперервне оновлення зображення.

Пошук QR-кодів виконується у методі get_frame(). Спочатку поточний кадр копіюється, після чого за допомогою функції decode(frame) виконується пошук QR-кодів у зображенні. Для зменшення навантаження на систему розпізнавання виконується не на кожному кадрі, а з інтервалом приблизно 0,25 с. Це дозволяє зберегти плавність роботи відеопотоку та одночасно забезпечити своєчасне виявлення маркерів.

Фрагмент розпізнавання QR-коду:

```
decoded_objects = decode(frame)

self.detected_markers = {}

for obj in decoded_objects:
    qr_code = obj.data.decode("utf-8").strip()

    x, y, w, h = obj.rect

    distance = self.estimate_distance(w)

    angle = self.estimate_angle(
```

```
        x + w / 2,
        frame_width
    )
```

Після виявлення QR-коду система отримує його текстове значення та координати області, у якій він був знайдений. Отриманий код зіставляється зі словником QR_ZONE_MAP, який дозволяє перетворити технічне значення QR-коду на назву складської зони. Наприклад, QR-код може відповідати зоні «Відділення № 1», «Центральний склад» або «Відділення № 2».

Важливою особливістю модуля є визначення орієнтовної відстані до QR-коду. Для цього використовується метод estimate_distance(), який розраховує відстань на основі реального розміру QR-коду, фокусної відстані камери та ширини маркера в пікселях. Такий підхід дозволяє приблизно оцінити, наскільки близько робот знаходиться до контрольної точки.

Розрахунок відстані до QR-коду виконується за формулою:

$$L = \frac{S \cdot F}{P},$$

де L – відстань від камери до QR-коду, м;

S – реальний розмір QR-коду, м;

F – фокусна відстань камери, пікселів;

P – ширина QR-коду на отриманому зображенні, пікселів.

Для перевірки роботи алгоритму використаємо такі вихідні дані:

– реальний розмір QR-коду S=0,10 м;

– фокусна відстань камери F = 720 пікселів;

– ширина QR-коду на зображенні P = 150 пікселів.

Після підстановки значень у формулу отримаємо:

$$L = \frac{0{,}10 \cdot 720}{150} = 0{,}48 \text{ м}.$$

Отриманий результат свідчить про те, що мобільний робот знаходиться приблизно на відстані 0,48 м від контрольної точки. Дане значення використовується програмним забезпеченням для оцінки моменту зупинки мобільного робота перед QR-кодом та забезпечення точного позиціонування.

Розрахунок відстані до QR-коду:

```
def estimate_distance(self, pixel_width):
    if pixel_width <= 0:
        return None

    return (REAL_QR_SIZE_METERS * FOCAL_LENGTH) / pixel_width
```

Крім відстані, модуль визначає кутове відхилення QR-коду відносно центра кадру. Для цього використовується метод estimate_angle(). Якщо QR-код знаходиться лівіше або правіше центра зображення, система може визначити напрямок, у який необхідно скоригувати рух робота. Це є важливим для точнішого позиціонування мобільної платформи перед контрольним маркером.

Розрахунок кута відхилення QR-коду:

```
def estimate_angle(self, qr_center_x, frame_width):
    frame_center_x = frame_width / 2
    offset = qr_center_x - frame_center_x

    return (
        offset / frame_center_x
    ) * (CAMERA_HORIZONTAL_FOV / 2)
```

Після розпізнавання QR-коду на зображення накладається прямокутна рамка та текстова мітка, яка містить назву зони, відстань до маркера та кутове відхилення. Це дозволяє оператору візуально контролювати процес розпізнавання та перевіряти правильність роботи системи навігації.

Інформація про знайдені QR-коди зберігається у словнику detected_markers. Для кожної знайденої зони зберігаються такі параметри: початковий QR-код, назва

зони, відстань до маркера та кут відхилення. Ці дані можуть використовуватися алгоритмом калібрування та модулем навігації для визначення поточного положення робота.

Окремо реалізовано метод get_nearest_marker(), який дозволяє визначити найближчий QR-маркер серед усіх знайдених. Це потрібно для ситуацій, коли в полі зору камери одночасно знаходиться декілька QR-кодів. У такому випадку система обирає той маркер, до якого відстань є найменшою.

Визначення найближчого QR-маркера:

```
def get_nearest_marker(self, ignored_zones=None):
    if ignored_zones is None:
        ignored_zones = []

    available_markers = {
        zone: data
        for zone, data in self.detected_markers.items()
        if zone not in ignored_zones
    }

    if not available_markers:
        return None, None

    nearest_zone = min(
        available_markers,
        key=lambda zone: available_markers[zone]["distance"]
    )

    return nearest_zone, available_markers[nearest_zone]
```

Таким чином, розроблений модуль CameraModule забезпечує повний цикл роботи з QR-кодами: отримання відеопотоку, пошук маркерів, декодування інформації, розрахунок відстані, визначення кута відхилення та передавання даних

до навігаційної системи. Це дозволяє використовувати QR-коди як контрольні точки для калібрування складського приміщення та подальшого автоматичного сортування поштових відправлень.

3.5 Реалізація алгоритму калібрування складського приміщення

Перед початком виконання транспортних завдань система повинна отримати інформацію про розташування контрольних точок складського приміщення. Для цього у програмному забезпеченні реалізовано алгоритм калібрування, який дозволяє автоматично знаходити QR-коди та формувати структуру складських зон.

Процедура калібрування запускається оператором через графічний інтерфейс програми. Після активації даного режиму робот переходить у стан пошуку навігаційних маркерів та починає рух по робочій зоні. Основною метою є знаходження всіх QR-кодів, які використовуються як контрольні точки системи навігації.

Під час руху камера безперервно передає зображення до модуля розпізнавання QR-кодів. Після виявлення маркера система визначає його назву та координати розташування в кадрі. Отримана інформація використовується для ідентифікації контрольної точки та подальшої навігації мобільного робота (рисунок 3.4).

Після запуску процедури калібрування мобільний робот послідовно здійснює пошук контрольних точок, розпізнає QR-коди, зчитує інформацію про них та формує карту складського приміщення. Загальний час виконання калібрування залежить від кількості контрольних точок, швидкості їх пошуку та часу обробки отриманих даних.

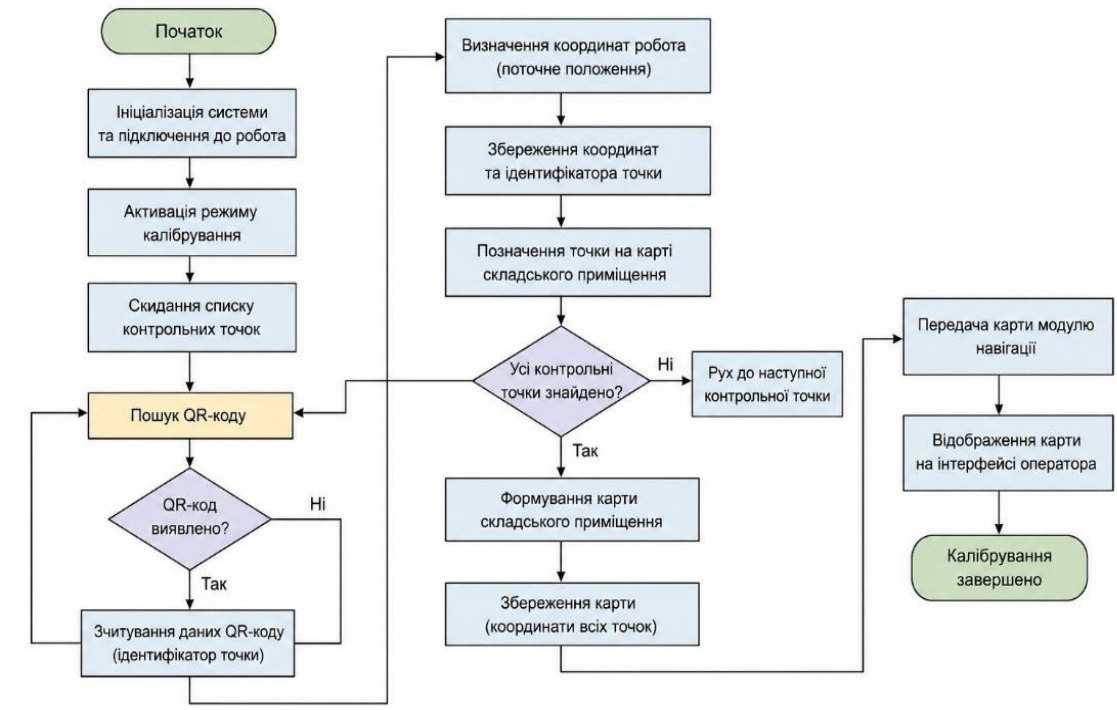


Рисунок 3.4 – Алгоритм калібрування складського приміщення

Для оцінювання часу виконання процедури калібрування приймемо, що в межах експериментальної моделі використовується три контрольні точки, які необхідно послідовно знайти та додати до карти. Під час виконання кожного циклу система виконує три основні операції:

– пошук QR-коду – 4 с;

– розпізнавання та зчитування інформації – 0,5 с;

– збереження координат контрольної точки – 0,3 с.

Загальний час калібрування робота визначається за формулою:

$$T = n \cdot \left(t_{\text{пошук}} + t_{\text{розпізнавання}} + t_{\text{збереження}}\right),$$

де T – загальний час калібрування, с;

n – кількість контрольних точок;

$t_{\text{пошук}}$ – час пошуку QR-коду, с;

$t_{\text{розпізнавання}}$ – час розпізнавання QR-коду, с;

$t_{\text{збереження}}$ – час збереження координат, с.

Для розробленої системи:

– $n = 3$;

$t_{\text{пошук}} = 4$ с;

$t_{\text{розпізнавання}} = 0{,}5$ с;

$t_{\text{збереження}} = 0{,}3$ с.

Після підстановки значень отримуємо:

$$T = 3 \cdot (4 + 0{,}5 + 0{,}3) = 14{,}4 \text{ с}.$$

Отже, процедура автоматичного калібрування займає близько 15 с, що дозволяє оперативно підготувати систему до виконання транспортних завдань навіть після зміни розташування контрольних точок.

Після успішного зчитування QR-коду інформація про відповідну складську зону записується до структури даних системи. У межах експериментальної моделі використовуються контрольні точки, які відповідають окремим складським зонам. Кожна знайдена точка зберігається лише один раз, що виключає дублювання інформації під час повторного виявлення QR-коду.

Для реалізації даного механізму використовується колекція calibrated_points, у якій накопичуються всі знайдені контрольні точки. Після виявлення нового QR-коду виконується перевірка наявності відповідного запису в системі. Якщо точка ще не була додана, інформація про неї зберігається для подальшого використання.

Фрагмент коду збереження контрольних точок під час калібрування:

```
if zone_name not in self.calibrated_points:
    self.calibrated_points[zone_name] = position
```

Після кожного успішного розпізнавання виконується перевірка кількості знайдених контрольних точок. Якщо не всі QR-коди були знайдені, робот продовжує пошук наступного маркера. Після завершення сканування всіх контрольних точок система переходить до формування структури розташування складських зон.

На основі зібраних даних формується структура розташування контрольних точок, яка надалі використовується навігаційним модулем для побудови маршрутів між складськими зонами та виконання транспортних завдань. Завдяки цьому відпадає необхідність ручного налаштування параметрів перед кожним запуском системи.

Однією з переваг реалізованого алгоритму є можливість автоматичної адаптації до зміни конфігурації складського приміщення. Якщо розташування QR-кодів змінюється, оператору достатньо повторно запустити процедуру калібрування. Після цього система автоматично оновить збережені дані та використовуватиме нову конфігурацію контрольних точок.

Після завершення процесу калібрування програмне забезпечення формує повідомлення про успішне завершення операції та переводить систему в режим виконання транспортних завдань . Усі знайдені точки стають доступними для навігаційного модуля та використовуються під час автоматичного сортування поштових відправлень (рисунок 3.5).

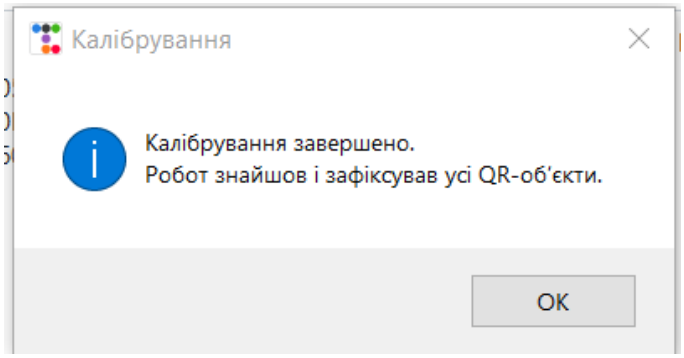


Рисунок 3.5 – Повідомлення про завершення калібрування системи

Таким чином, реалізований алгоритм калібрування дозволяє автоматично визначати розташування контрольних точок, формувати структуру складських зон та забезпечувати підготовку системи до виконання завдань сортування поштових відправлень.

3.6 Реалізація алгоритму сортування поштових відправлень

Основним призначенням розробленого програмного забезпечення є автоматизація процесу сортування поштових відправлень за допомогою мобільного робота. Після завершення калібрування та формування карти складського приміщення система переходить до виконання транспортних завдань, створених оператором.

Перед початком виконання транспортного завдання мобільна роботизована платформа переходить у режим автоматичного керування. У цьому режимі рух робота здійснюється за принципом замкненої системи автоматичного керування, у якій формування керуючих команд відбувається на основі інформації, отриманої від системи комп’ютерного зору.

Об’єктом автоматичного керування є мобільна роботизована платформа. Керуючим пристроєм виступає програмний модуль навігації, реалізований мовою Python та виконуваний на Raspberry Pi 5. Виконавчими механізмами є електродвигуни приводу коліс, а функцію зворотного зв’язку виконує відеокамера, що безперервно передає інформацію про положення QR-кодів у полі зору.

На основі отриманих даних програмне забезпечення формує керуючі команди руху: рух уперед, поворот ліворуч, поворот праворуч або зупинка. Завдяки використанню зворотного зв’язку система автоматично коригує траєкторію руху мобільної платформи та забезпечує її переміщення до необхідної складської зони (рисунок 3.6).

Процес сортування починається зі створення нового завдання через графічний інтерфейс програми. Користувач обирає необхідну складську зону, після чого інформація передається до модуля керування завданнями. На основі отриманих даних визначається пункт призначення та запускається процес переміщення мобільного робота (рисунок 3.7).

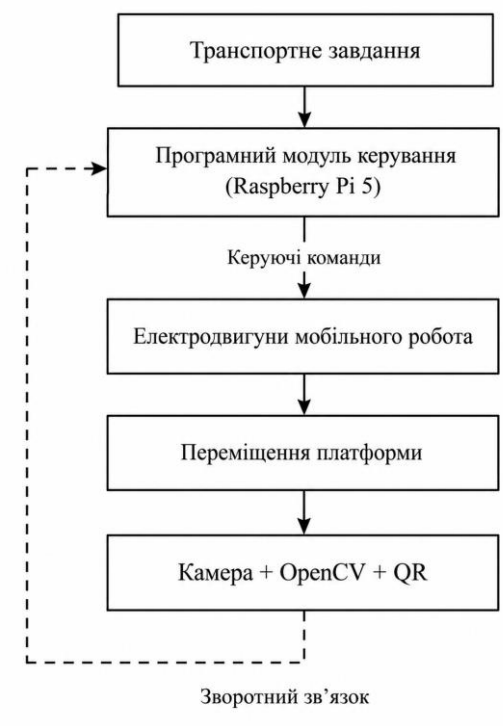


Рисунок 3.6 – Структурна схема автоматичного керування мобільною роботизованою платформою

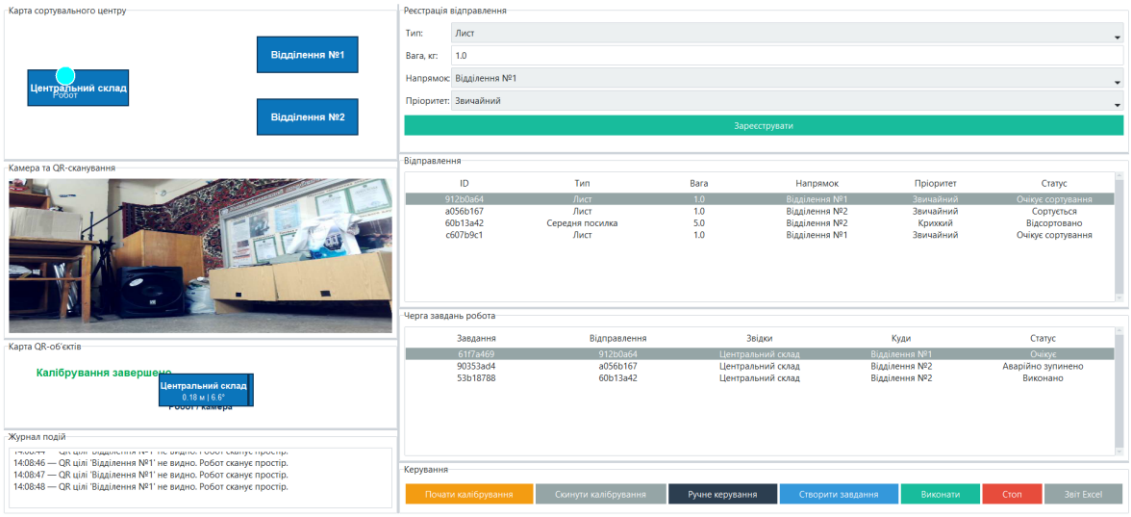


Рисунок 3.7 – Запуск виконання транспортного завдання в програмному забезпеченні

Після отримання команди мобільний робот починає рух до цільової зони. Під час переміщення система безперервно отримує відеопотік із камери та виконує пошук QR-кодів. Кожен успішно розпізнаний QR-код використовується для

підтвердження поточного положення робота та контролю правильності виконання маршруту.

Для забезпечення автоматичного керування під час руху виконується безперервне визначення положення мобільної платформи відносно контрольної точки. На основі координат QR-коду в кадрі обчислюється відстань до маркера та величина кутового відхилення. Отримані значення використовуються для формування коригуючих керуючих впливів.

Похибка позиціонування визначається як:

$$e = L_{\text{зад}} - L_{\text{факт}},$$

де $L_{\text{зад}}$ – необхідна відстань до контрольної точки;

$L_{\text{факт}}$ – відстань, визначена системою комп'ютерного зору.

У разі перевищення допустимого значення похибки програмне забезпечення автоматично формує команди коригування напрямку руху. Після досягнення необхідного положення формується команда зупинки та завершується виконання транспортного завдання.

Якщо знайдений QR-код відповідає очікуваній контрольній точці, система продовжує виконання завдання. У випадку втрати навігаційного маркера або помилки розпізнавання робот переходить у режим повторного пошуку та намагається відновити коректне переміщення.

Для реалізації алгоритму використовується навігаційний модуль, який працює на основі даних, отриманих під час калібрування. Це дозволяє визначати момент досягнення необхідної складської зони та завершення транспортного завдання.

Приклад перевірки досягнення пункту призначення:

```
if zone_name == target_zone:
    self.log("Пункт призначення досягнуто")
```

```
self.run_robot_command("stop")
self.task_completed = True
```

Після прибуття до заданої складської зони робот зупиняється та передає повідомлення про успішне завершення операції. Відповідна інформація відображається у журналі подій програмного забезпечення та стає доступною оператору.

У процесі роботи система також контролює можливі помилки. У разі втрати зв'язку з мобільною платформою або неможливості розпізнавання QR-кодів програмне забезпечення формує відповідне повідомлення та переводить систему в безпечний режим роботи (рисунок 3.8).

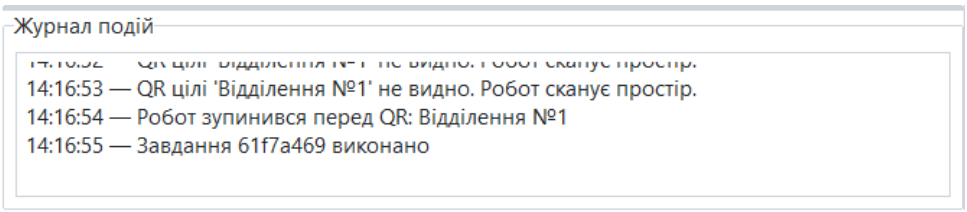


Рисунок 3.8 – Повідомлення про успішне завершення транспортного завдання

Однією з переваг реалізованого алгоритму є можливість багаторазового використання сформованої карти складського приміщення. Після виконання калібрування робот може здійснювати необмежену кількість транспортних операцій без повторного сканування всіх контрольних точок.

Завдяки використанню QR-навігації забезпечується достатня точність позиціонування мобільного робота та автоматизація процесу переміщення поштових відправлень між складськими зонами.

Таким чином, реалізований алгоритм сортування поштових відправлень поєднує програмні засоби керування, алгоритми комп'ютерного зору та принципи автоматичного керування. Використання замкненої системи зі зворотним зв'язком дозволяє автоматично контролювати положення мобільної роботизованої платформи, своєчасно коригувати траєкторію руху та забезпечувати точне виконання транспортних операцій між складськими зонами.

3.7 Розробка графічного інтерфейсу оператора

Для забезпечення зручної взаємодії користувача з комплексом сортування поштових відправлень було розроблено графічний інтерфейс оператора. Основним призначенням інтерфейсу є надання засобів керування мобільним роботом, створення транспортних завдань, запуск процедур калібрування та моніторинг поточного стану системи.

Інтерфейс реалізовано мовою Python із використанням бібліотек Tkinter та ttkbootstrap. Використання даних засобів дозволило створити сучасний графічний застосунок із підтримкою віконного режиму роботи та інтерактивних елементів керування.

Головне вікно програмного забезпечення містить інформацію про стан підключення до мобільного робота, елементи запуску основних режимів роботи та область відображення журналу подій. Через головне меню оператор може запускати калібрування, створювати нові транспортні завдання та переходити до режиму ручного керування (рисунок 3.9).

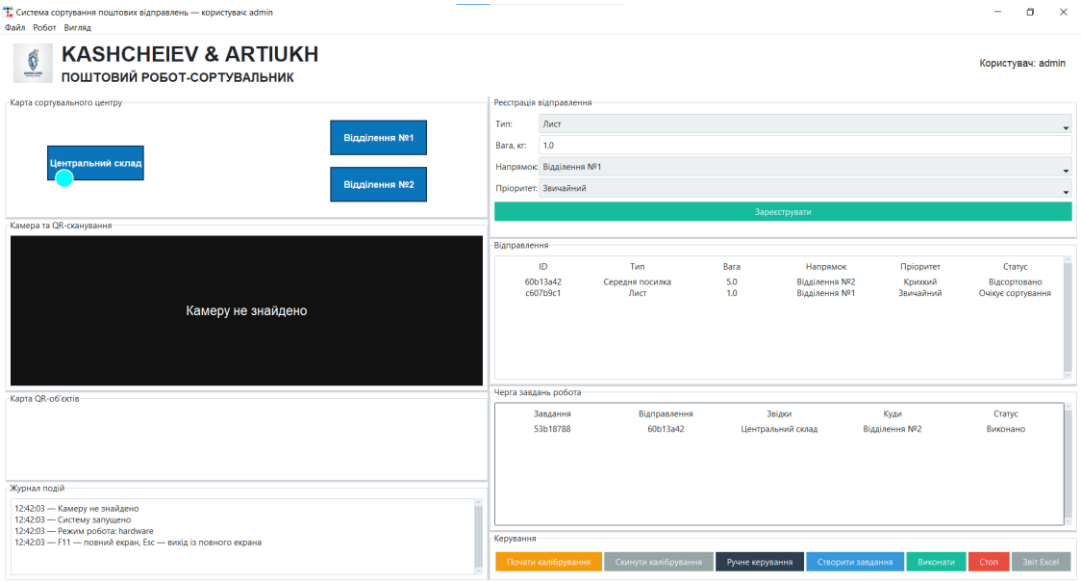


Рисунок 3.9 – Головне вікно програмного забезпечення

Однією з основних функцій інтерфейсу є створення транспортних завдань. Для цього користувач обирає пункт призначення та запускає виконання маршруту. Після створення завдання інформація передається до модуля керування, який забезпечує передачу команд мобільному роботу.

Для перевірки працездатності системи реалізовано режим ручного керування (рисунок 3.10). У даному режимі оператор може самостійно надсилати команди руху, виконувати тестування окремих функцій та контролювати переміщення мобільної платформи.

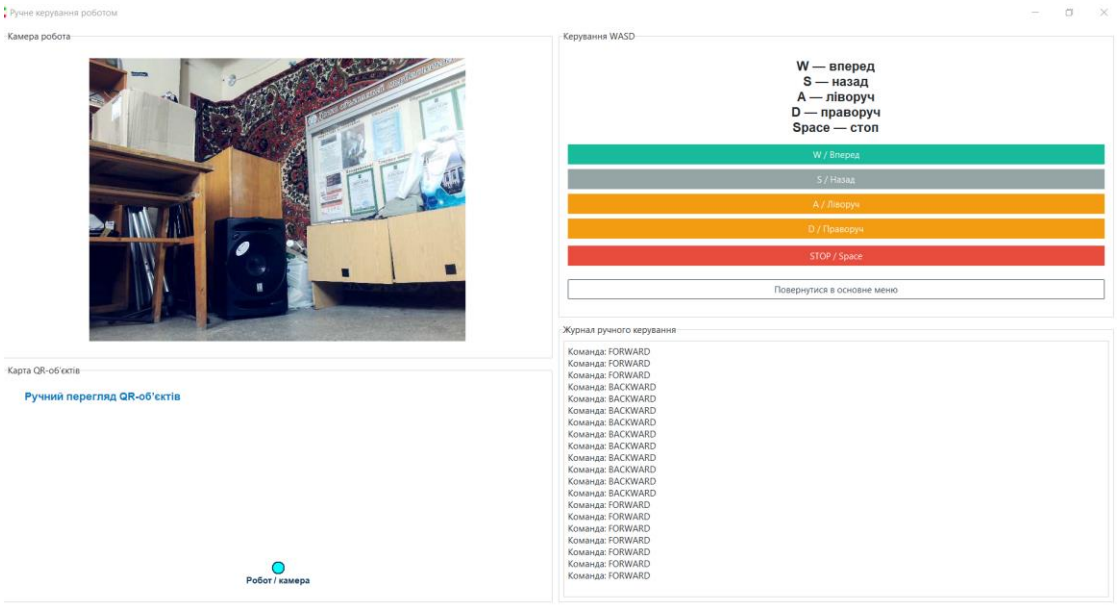


Рисунок 3.10 – Елементи ручного керування мобільним роботом

Важливою складовою інтерфейсу є відображення відеопотоку з камери робота. Отримане зображення використовується для контролю навігації, пошуку QR-кодів та оцінки поточного положення мобільної платформи. Передача відео здійснюється через локальну мережу за допомогою сервісу потокової передачі даних.

Для контролю роботи системи використовується журнал подій (рисунок 3.11). У журналі відображаються повідомлення про підключення до робота, запуск калібрування, результати розпізнавання QR-кодів, створення транспортних завдань та виникнення можливих помилок.

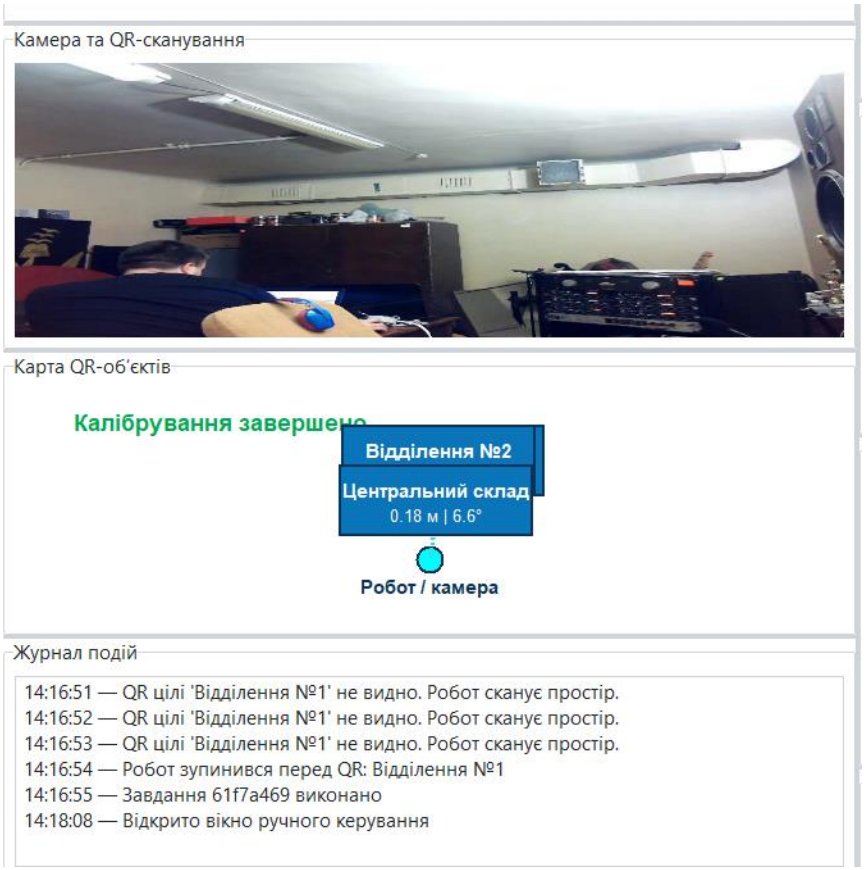


Рисунок 3.11 – Журнал подій та відображення відеопотоку

Під час розробки інтерфейсу особлива увага приділялася простоті використання та швидкому доступу до основних функцій комплексу. Усі елементи керування згруповані відповідно до їхнього призначення, що дозволяє оператору швидко виконувати необхідні дії та контролювати роботу системи.

Розроблений інтерфейс забезпечує ефективну взаємодію між користувачем та програмно-апаратним комплексом, дозволяє контролювати всі етапи роботи мобільного робота та спрощує виконання завдань сортування поштових відправлень.

Таким чином, створений графічний інтерфейс оператора забезпечує зручне керування системою, моніторинг стану мобільного робота та контроль виконання транспортних операцій у режимі реального часу.

У третьому розділі виконано розробку програмного забезпечення комплексу сортування поштових відправлень на базі мобільної роботизованої платформи.

Обґрунтовано вибір програмних засобів розробки та реалізовано модульну структуру програмного забезпечення. Розроблено модуль взаємодії з мобільним роботом, реалізовано систему обробки та розпізнавання QR-кодів, а також алгоритм калібрування складського приміщення для автоматичного формування карти контрольних точок. Реалізовано алгоритм автоматичного сортування поштових відправлень та створено графічний інтерфейс оператора, який забезпечує керування мобільною платформою, моніторинг стану системи та контроль виконання транспортних завдань. Отримані результати підтверджують можливість практичного використання розробленого програмного забезпечення для автоматизації процесу сортування поштових відправлень та створюють основу для проведення експериментальних досліджень, результати яких наведено в наступному розділі.

4 ДОСЛІДЖЕННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Методика проведення експериментальних досліджень

Для перевірки працездатності розробленого програмного забезпечення було проведено серію експериментальних досліджень, спрямованих на оцінку роботи модулів розпізнавання QR-кодів, калібрування складського приміщення та виконання транспортних завдань. Основною метою експериментів було підтвердження можливості використання розробленого комплексу для автоматизації процесів сортування поштових відправлень.

Дослідження проводилися на експериментальному стенді, який моделює роботу складського приміщення. У межах стенда використовувалися контрольні точки, представлені QR-кодами, що відповідають основним зонам логістичного комплексу: «Центральний склад», «Відділення №1» та «Відділення №2». Дані QR-коди використовувалися для навігації мобільного робота та визначення його поточного положення (рисунок 4.1).

Під час проведення експериментів використовувався мобільний робот, оснащений камерою для отримання відеопотоку та модулем бездротового мережевого зв’язку. Обробка інформації виконувалася програмним забезпеченням, розробленим мовою Python. Для розпізнавання QR-кодів використовувалася бібліотека OpenCV, а передача відеопотоку здійснювалася через мережевий сервіс Flask. Робота програмного забезпечення базується на модулі CameraModule, який виконує пошук QR-кодів, визначення відстані до маркерів та розрахунок кутового відхилення.

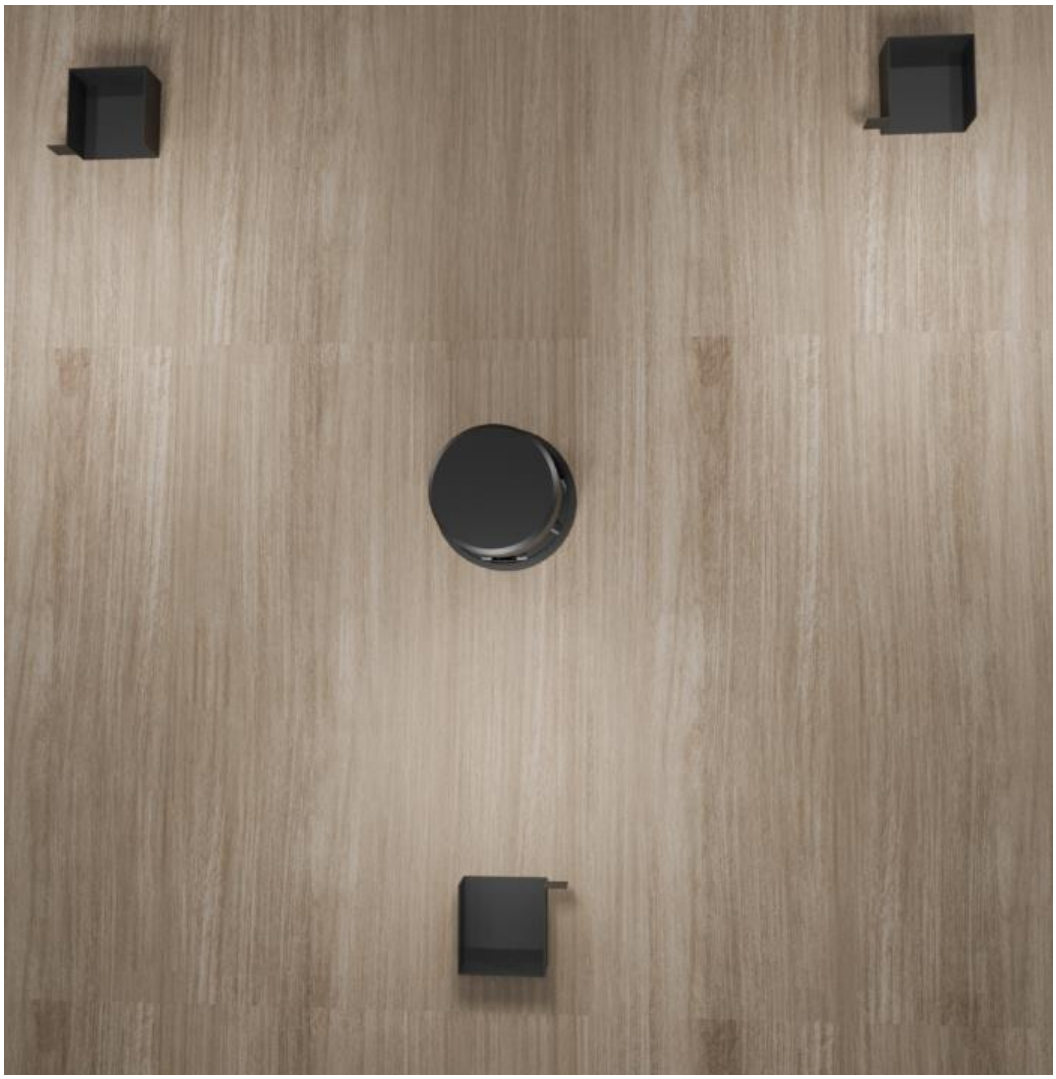


Рисунок 4.1 – Експериментальний стенд для дослідження роботи комплексу сортування поштових відправлень

Конфігурація програмного забезпечення передбачає використання трьох контрольних точок: «Центральний склад», «Відділення № 1» та «Відділення № 2». Для кожної точки використовується окремий QR-код, а результати їх розпізнавання використовуються під час калібрування та побудови маршрутів (рисунки 4.2-4.4).



Рисунок 4.2 – QR-код контрольної точки «Центральний склад»



Рисунок 4.3 – QR-код контрольної точки «Відділення № 1»



Рисунок 4.4 – QR-код контрольної точки «Відділення № 2»

У процесі досліджень перевірялися такі функції системи:

– коректність розпізнавання QR-кодів;

– визначення відстані до контрольних точок;

– визначення кутового відхилення відносно QR-коду;

– робота алгоритму калібрування складського приміщення;

– формування карти складських зон;

– виконання транспортних завдань між контрольними точками;

– взаємодія між програмою оператора та мобільним роботом.

Для забезпечення обєктивності результатів кожен експеримент виконувався не менше трьох разів (таблиця 4.1). Під час проведення випробувань фіксувалися результати розпізнавання QR-кодів, успішність побудови карти складського приміщення та правильність виконання маршрутів між складськими зонами.

Таблиця 4.1 – Основні параметри експериментального стенда

Параметр	Значення
Кількість контрольних точок	3
Тип навігації	QR-навігація
Камера	Raspberry Pi Camera Module
Мова програмування	Python
Бібліотека комп’ютерного зору	OpenCV
Мережевий протокол	TCP/IP
Кількість маршрутів для тестування	4
Кількість повторів кожного експерименту	3

Запропонована методика дозволяє комплексно оцінити роботу розробленого програмного забезпечення та визначити ефективність використання QR-навігації для автоматизації процесів сортування поштових відправлень.

Таким чином, сформована методика експериментальних досліджень забезпечує перевірку всіх основних функцій комплексу та створює основу для проведення подальшого тестування модулів розпізнавання QR-кодів, калібрування та сортування поштових відправлень.

4.2 Тестування модуля розпізнавання QR-кодів

Одним із ключових компонентів розробленого програмного забезпечення є модуль розпізнавання QR-кодів, який використовується для визначення контрольних точок маршруту та позиціонування мобільного робота в межах складського приміщення. Тому важливим етапом експериментальних досліджень стала перевірка коректності роботи даного модуля.

Для проведення випробувань використовувалися три QR-коди, що відповідають контрольним точкам системи навігації: «Центральний склад», «Відділення № 1» та «Відділення № 2». Кожен QR-код розміщувався в полі зору камери мобільного робота, після чого виконувалося його автоматичне розпізнавання засобами програмного забезпечення.

Під час тестування використовувався розроблений модуль CameraModule, який здійснює отримання відеопотоку, пошук QR-кодів, визначення відстані до маркера та розрахунок кута його розташування відносно центра кадру. Після розпізнавання інформація про QR-код записується до структури даних системи та передається навігаційному модулю (рисунки 4.5-4.7).

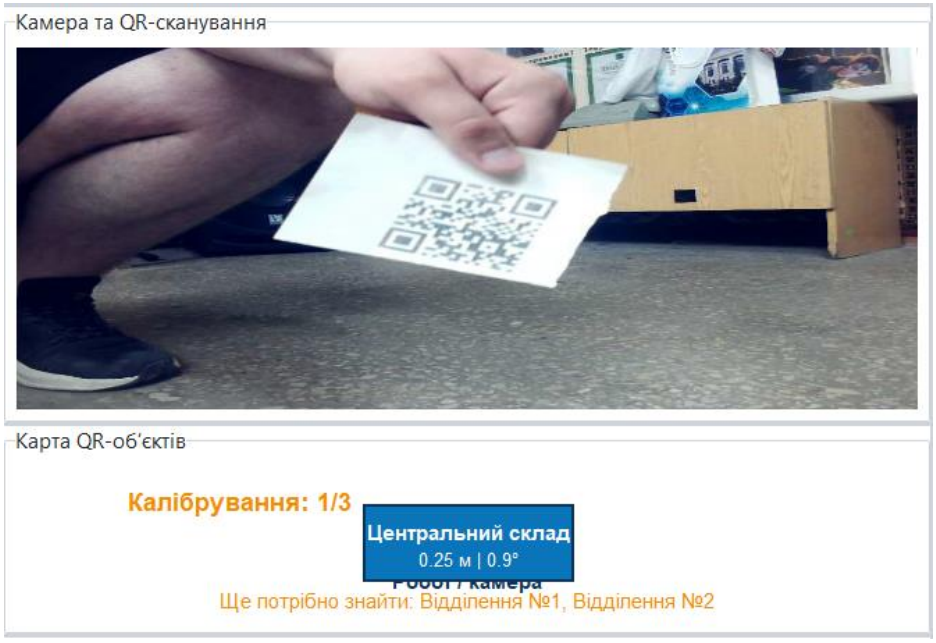


Рисунок 4.5 – Розпізнавання QR-коду «Центральний склад»

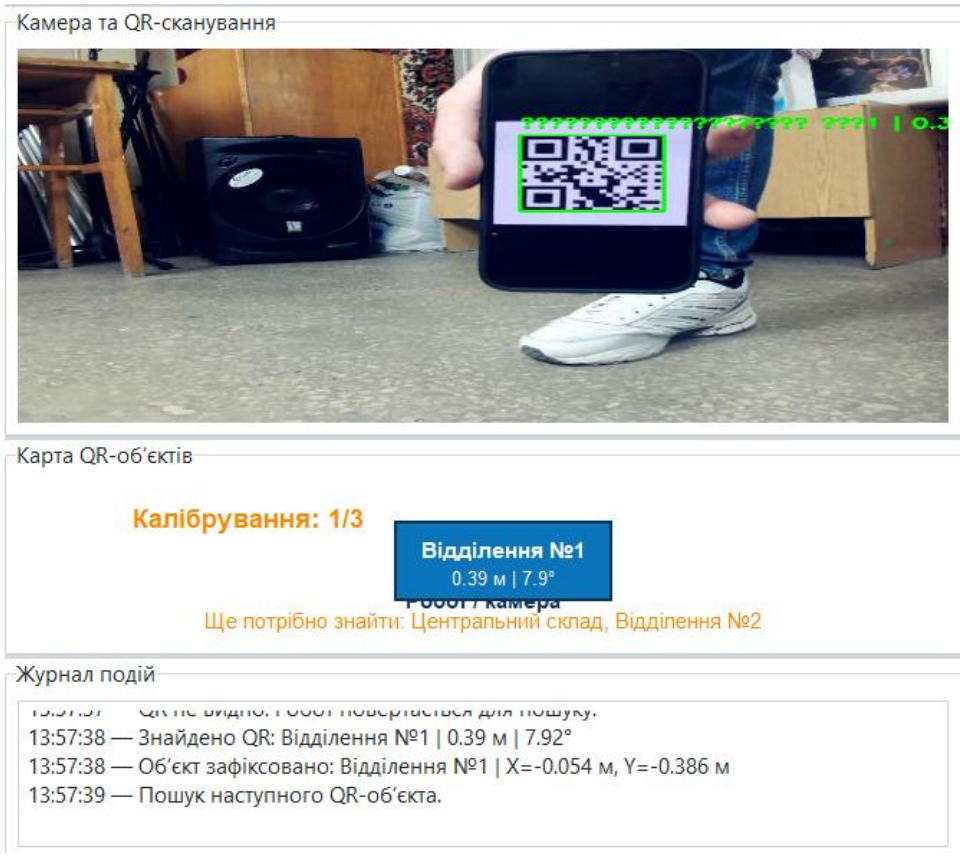


Рисунок 4.6 – Розпізнавання QR-коду «Відділення № 1»



Рисунок 4.7 – Розпізнавання QR-коду «Відділення № 2»

Під час виконання експериментів перевірялися такі параметри:

– коректність розпізнавання QR-коду;

– відповідність назви складської зони;

– визначення відстані до маркера;

– визначення кутового відхилення;

– стабільність роботи при багаторазовому скануванні.

Для кожної контрольної точки було проведено три повторних сканування. У всіх випадках система успішно виконала розпізнавання QR-кодів та правильно визначила відповідні складські зони. Помилкових спрацьовувань або випадків неправильного декодування зафіксовано не було (таблиця 4.2).

Таблиця 4.2 – Результати тестування розпізнавання QR-кодів

№	Контрольна точка	Кількість випробувань	Успішних розпізнавань
1	Центральний склад	3	3
2	Відділення № 1	3	3
3	Відділення № 2	3	3

Крім факту розпізнавання, програмне забезпечення виконувало розрахунок відстані до QR-коду та кута його розташування. Отримані дані використовувалися

навігаційною підсистемою для визначення подальших команд руху робота. Відповідні функції реалізовані у методах estimate_distance() та estimate_angle() модуля CameraModule (рисунок 4.8).

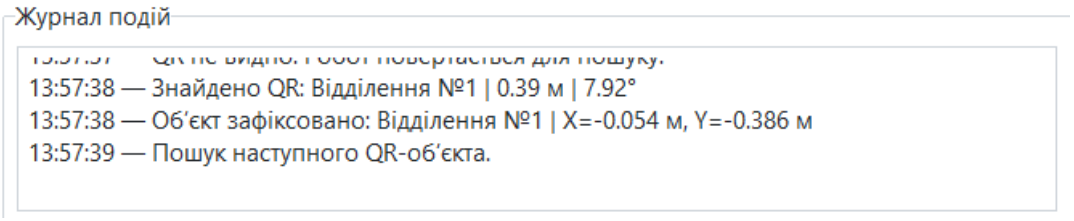


Рисунок 4.8 – Відображення відстані та кутового відхилення до QR-коду в інтерфейсі системи

Для оцінки стабільності роботи модуль тестувався в умовах різного розташування QR-кодів у кадрі. Система успішно виконувала розпізнавання як при розташуванні маркера в центрі зображення, так і при його зміщенні ліворуч або праворуч відносно камери. При цьому коректно обчислювався кут відхилення, який надалі використовується для корекції траєкторії руху робота.

Результати проведених випробувань підтвердили працездатність розробленого модуля розпізнавання QR-кодів та можливість його використання в системі навігації мобільного робота. Отримані дані стали основою для проведення наступного етапу досліджень, пов’язаного з тестуванням алгоритму калібрування складського приміщення.

4.3 Тестування алгоритму калібрування складського приміщення

Після перевірки працездатності модуля розпізнавання QR-кодів було проведено тестування алгоритму калібрування складського приміщення. Основною метою випробувань було підтвердження можливості автоматичного виявлення контрольних точок та побудови карти складських зон без ручного введення координат.

У процесі калібрування мобільний робот послідовно здійснює пошук QR-кодів, виконує їх розпізнавання та зберігає інформацію про знайдені контрольні точки. Після завершення пошуку система формує карту складського приміщення, яка надалі використовується для виконання транспортних завдань.

Під час проведення експерименту в робочій зоні були розміщені три контрольні точки:

– Центральний склад;

– Відділення №1;

– Відділення №2.

Робот запускався в режимі калібрування та виконував автоматичний пошук усіх доступних QR-кодів. Після виявлення кожного маркера система виконувала його розпізнавання та запис інформації до структури даних карти (рисунок 4.9).

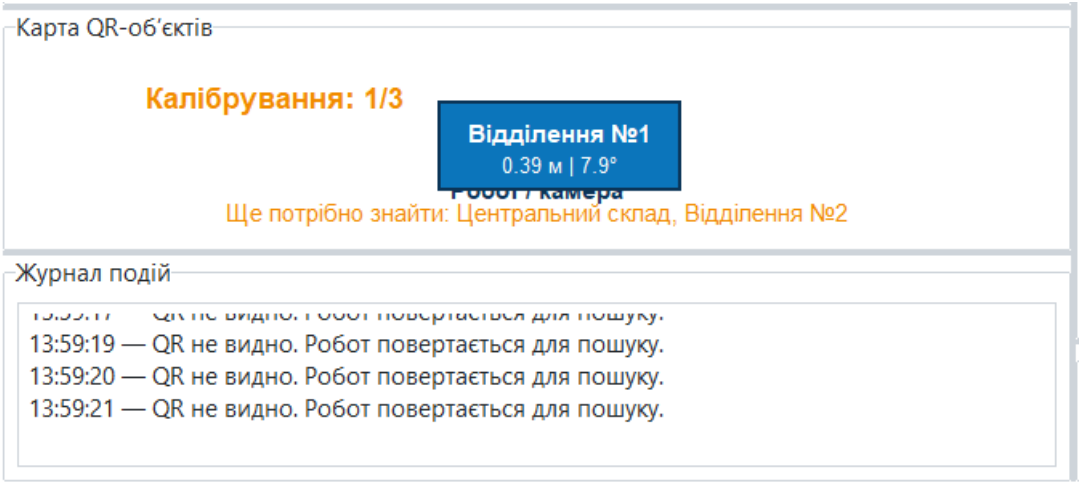


Рисунок 4.9 – Процес пошуку контрольних точок під час калібрування

Після завершення сканування виконувався аналіз кількості знайдених контрольних точок. Калібрування вважалося успішним лише за умови виявлення всіх QR-кодів, передбачених конфігурацією системи. У програмному забезпеченні перелік обов'язкових контрольних точок визначається масивом REQUIRED_CALIBRATION_MARKERS.

Для оцінки стабільності роботи алгоритму процедура калібрування виконувалася три рази. У всіх випадках система успішно виявила всі контрольні точки та виконала формування карти складського приміщення (таблиця 4.3).

Таблиця 4.3 – Результати калібрування системи

Контрольна точка	Результат
Центральний склад	Знайдено
Відділення №1	Знайдено
Відділення №2	Знайдено
Формування карти	Виконано

У процесі випробувань система успішно виявила всі контрольні точки та сформувала карту складського приміщення (рисунок 4.10). Помилок під час побудови карти або втрати даних не зафіксовано.

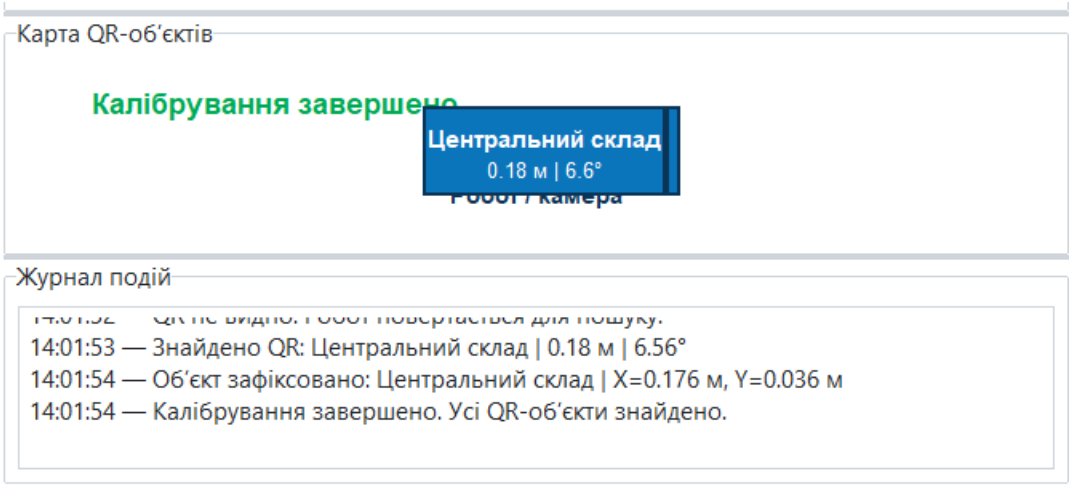


Рисунок 4.10 – Карта складських зон після завершення калібрування

Додатково було перевірено можливість повторного виконання калібрування після зміни розташування QR-кодів. Після повторного запуску процедури система успішно оновила інформацію про контрольні точки та сформувала нову карту без необхідності ручного втручання оператора.

Під час виконання експериментів також перевірялася стабільність роботи алгоритму. У всіх випадках система успішно завершувала побудову карти та переходила до режиму виконання транспортних завдань (рисунок 4.11).

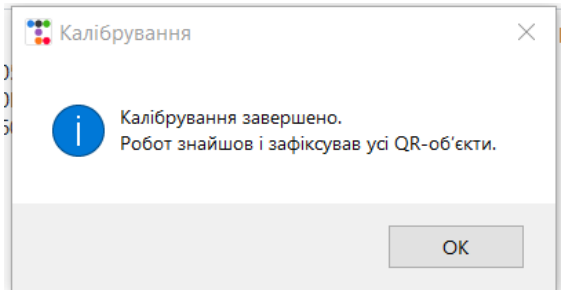


Рисунок 4.11 – Повідомлення про успішне завершення калібрування

Отримані результати підтвердили працездатність алгоритму калібрування та можливість його використання для автоматичного формування карти складського приміщення. Це дозволяє адаптувати систему до зміни конфігурації робочої зони та забезпечує гнучкість використання комплексу сортування поштових відправлень.

4.4 Тестування алгоритму сортування поштових відправлень

Після успішного завершення калібрування системи було проведено тестування алгоритму сортування поштових відправлень. Основною метою даного етапу досліджень була перевірка правильності побудови маршрутів, навігації між контрольними точками та виконання транспортних завдань мобільним роботом.

У розробленій системі процес сортування починається зі створення транспортного завдання оператором. Після вибору пункту призначення інформація передається до навігаційного модуля, який визначає необхідний маршрут руху та здійснює контроль його виконання. Для навігації використовуються дані про контрольні точки, отримані під час калібрування складського приміщення.

Під час руху робот виконує пошук QR-кодів, визначає відстань до контрольної точки, розраховує кут відхилення та здійснює корекцію напрямку

руху. Якщо QR-код цільової зони відсутній у полі зору камери, система переходить у режим пошуку та продовжує навігацію (рисунок 4.12).

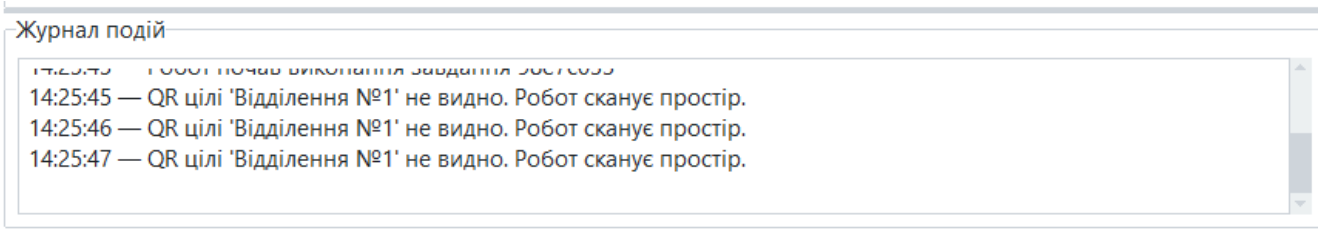


Рисунок 4.12 – Виконання транспортного завдання мобільним роботом

Для перевірки працездатності алгоритму було сформовано декілька маршрутів між контрольними точками складського приміщення. Кожний маршрут виконувався декілька разів, що дозволило оцінити стабільність роботи системи (таблиця 4.4).

Таблиця 4.4 – Результати тестування маршрутів

№	Маршрут	Результат
1	Центральний склад → Відділення №1	Успішно
2	Центральний склад → Відділення №2	Успішно
3	Відділення №1 → Центральний склад	Успішно
4	Відділення №2 → Центральний склад	Успішно

У процесі виконання маршрутів система успішно визначала поточне положення робота за допомогою QR-кодів та виконувала корекцію напрямку руху. При виявленні відхилення від маршруту використовувалися команди повороту, що дозволяло відновити правильну траєкторію переміщення.

Під час тестування також перевірялася робота механізму визначення досягнення цільової точки. Після наближення до необхідного QR-коду система виконувала зупинку робота та формувала повідомлення про завершення транспортного завдання.

Для оцінки надійності роботи було проведено серію повторних випробувань. У ході експериментів не було зафіксовано випадків втрати маршруту або помилкового визначення контрольних точок. Усі сформовані завдання були успішно виконані (таблиця 4.5).

Таблиця 4.5 – Узагальнені результати виконання транспортних завдань

Показники	Значення
Кількість іспитів	3
Успішно виконано	3
Невдалих маршрутів	0
Відсоток успішного виконання	100%

Отримані результати підтвердили працездатність алгоритму сортування поштових відправлень та можливість його використання для автоматизованого переміщення поштових відправлень між складськими зонами. Реалізований підхід забезпечує стабільну роботу системи навігації та дозволяє виконувати транспортні операції без безпосереднього втручання оператора.

У четвертому розділі було проведено експериментальні дослідження та тестування розробленого програмного забезпечення комплексу сортування поштових відправлень. Перевірено працездатність основних функціональних модулів системи, зокрема модуля розпізнавання QR-кодів, алгоритму калібрування складського приміщення та алгоритму сортування поштових відправлень.

Результати тестування показали, що система успішно виконує розпізнавання контрольних QR-кодів та визначає їх положення відносно мобільного робота.

Реалізований модуль комп’ютерного зору забезпечує стабільну роботу навігаційної підсистеми та дозволяє використовувати QR-коди як навігаційні маркери.

Під час перевірки алгоритму калібрування було підтверджено можливість автоматичного виявлення контрольних точок та формування карти складського приміщення без ручного налаштування координат. Система успішно знаходила всі контрольні точки та коректно формувала карту робочої зони.

Результати тестування алгоритму сортування підтвердили можливість автоматичного виконання транспортних завдань між складськими зонами. Усі маршрути були успішно виконані, а мобільний робот коректно визначав цільові точки та завершував рух після досягнення пункту призначення.

Отримані результати підтверджують працездатність розробленого програмного забезпечення та можливість його використання для автоматизації процесів сортування поштових відправлень на базі мобільних роботизованих платформ.

4.5 Аналіз умов праці під час роботи з програмним забезпеченням

Під час виконання кваліфікаційної роботи розробка, налагодження та тестування програмного забезпечення здійснювалися на персональному комп’ютері із використанням інтегрованого середовища розробки Python та засобів комп'ютерного зору OpenCV. Також виконувалося тестування взаємодії програмного забезпечення з мобільною роботизованою платформою на базі Raspberry Pi 5, обробка відеопотоку з камери, розпізнавання QR-кодів та перевірка роботи алгоритмів навігації.

Основними шкідливими виробничими факторами під час виконання роботи є тривале навантаження на органи зору внаслідок постійної роботи за монітором, статичне навантаження на опорно-руховий апарат через тривале перебування у сидячому положенні, а також нервово-емоційне навантаження, пов’язане з розробкою, налагодженням та тестуванням програмного забезпечення.

Для зменшення негативного впливу зазначених факторів необхідно забезпечити правильну організацію робочого місця, дотримуватися вимог ергономіки, підтримувати оптимальний рівень освітленості робочої поверхні та виконувати регламентовані перерви під час роботи за персональним комп'ютером відповідно до чинних нормативних документів [26].

4.6 Вимоги безпеки під час роботи за персональним компютером

Перед початком роботи необхідно перевірити справність персонального комп'ютера, монітора, мережевого обладнання, Raspberry Pi 5, відеокамери та інших пристроїв, що входять до складу програмно-апаратного комплексу. Не допускається експлуатація обладнання з пошкодженими кабелями живлення, несправними блоками живлення або електричними розетками.

Під час роботи оператор повинен знаходитися у зручному положенні, використовуючи регульоване робоче крісло. Монітор рекомендується розташовувати на відстані від 50 см до 70 см від очей, а його верхня межа повинна знаходитися приблизно на рівні очей користувача. Забороняється самостійно виконувати ремонт обладнання, яке перебуває під напругою, а також здійснювати підключення або відключення електронних компонентів під час подачі живлення.

Після завершення роботи необхідно коректно завершити виконання програмного забезпечення, вимкнути персональний комп’ютер, Raspberry Pi 5 та інше обладнання відповідно до вимог експлуатаційної документації [27].

4.7 Пожежна безпека

Під час експлуатації програмно-апаратного комплексу необхідно дотримуватися вимог пожежної безпеки, оскільки до складу системи входять електронні пристрої, які працюють від електричної мережі.

Не допускається перевантаження електричної мережі шляхом підключення великої кількості електроприладів до одного подовжувача або розетки.

Забороняється використання несправних блоків живлення, пошкоджених кабелів або обладнання з ознаками перегрівання.

Робоче місце повинно утримуватися у чистоті, а поблизу електронного обладнання не допускається зберігання легкозаймистих матеріалів. У приміщенні повинні бути передбачені первинні засоби пожежогасіння, а також забезпечений вільний доступ до шляхів евакуації у разі виникнення надзвичайної ситуації [28].

Дотримання вимог охорони праці під час розробки та експлуатації програмного забезпечення мобільного робота дозволяє забезпечити безпечні умови праці оператора, знизити ризик виникнення аварійних ситуацій та підвищити надійність функціонування програмно-апаратного комплексу. Виконання вимог електробезпеки, пожежної безпеки та правил ергономічної організації робочого місця сприяє збереженню працездатності користувача, зменшенню рівня втомлюваності та забезпечує ефективне виконання робіт із розробки, тестування та експлуатації системи автоматизованого сортування поштових відправлень.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи досягнуто поставленої мети, яка полягала у розробці програмного забезпечення програмно-апаратного комплексу сортування поштових відправлень із використанням мобільної роботизованої платформи та технології QR-навігації.

Під час виконання роботи проведено аналіз сучасних систем автоматизації логістичних процесів, методів навігації мобільних роботів, засобів комп'ютерного зору та існуючого програмного забезпечення для керування робототехнічними системами. За результатами аналізу визначено переваги використання QR-кодів як навігаційних маркерів, досліджено принципи автоматичного керування мобільними роботами та сформовано основні вимоги до програмного забезпечення комплексу сортування поштових відправлень.

У ході роботи спроєктовано архітектуру програмно-апаратного комплексу, розроблено структуру програмного забезпечення, алгоритми автоматичного калібрування складського приміщення, QR-навігації та виконання транспортних завдань. Запропонована модульна структура програмного забезпечення забезпечує взаємодію між оператором, мобільною роботизованою платформою та системою комп'ютерного зору, а також дозволяє розширювати функціональні можливості комплексу без зміни його загальної архітектури.

Розроблено програмне забезпечення оператора та мобільного робота із застосуванням мови програмування Python, бібліотек OpenCV, Pyzbar, Flask і Tkinter. Реалізовано функції авторизації користувача, отримання відеопотоку з камери, автоматичного розпізнавання QR-кодів, визначення відстані та кутового відхилення до навігаційних міток, автоматичного калібрування складського приміщення, побудови карти контрольних точок, створення транспортних завдань і дистанційного керування мобільною роботизованою платформою.

У процесі виконання роботи також проведено інженерні розрахунки основних параметрів функціонування системи, зокрема визначення відстані до QR-коду, оцінювання часу виконання процедури калібрування та аналізу роботи алгоритмів навігації. Це дозволило підтвердити коректність використаних математичних моделей та обґрунтувати ефективність реалізованих програмних рішень.

Під час експериментального дослідження та тестування перевірено працездатність усіх основних модулів програмного забезпечення. Результати тестування підтвердили коректну роботу алгоритмів розпізнавання QR-кодів, автоматичного калібрування, побудови карти контрольних точок та виконання транспортних маршрутів між складськими зонами. Розроблене програмне забезпечення забезпечує стабільну взаємодію між оператором і мобільним роботом та дозволяє автоматизувати процес виконання транспортних завдань.

Практична цінність роботи полягає у створенні спеціалізованого програмного забезпечення для мобільної роботизованої платформи, яке може бути використане для автоматизації внутрішньоскладських логістичних процесів, сортування поштових відправлень, навчальних досліджень у галузі робототехніки та автоматизованих систем керування. Запропоновані алгоритми та програмні рішення можуть бути адаптовані для використання в інших роботизованих логістичних комплексах, а також доповнені новими засобами навігації, системами уникнення перешкод та інтеграцією із сучасними складськими інформаційними системами.

Таким чином, поставлену мету кваліфікаційної роботи досягнуто, усі визначені завдання виконано, а отримані результати підтверджують можливість практичного використання розробленого програмного забезпечення для автоматизації процесів сортування поштових відправлень із застосуванням мобільних роботизованих платформ.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Указ Президента України №722/2019 «Про Цілі сталого розвитку України на період до 2030 року». – Режим доступу: https://www.president.gov.ua/documents/7222019-29825.

2. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.

3. Методичні вказівки з підготовки кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти денної і заочної форми навчання спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології освітньої [Електронний ресурс] / упоряд.: І. Ш. Невлюдов, О. І. Филипенко, О. В. Токарєва, С. П. Новоселов, О.В Сичова ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – електрон. вид. – Харків : ХНУРЕ, 2023. – 760 Kb

4. Навчальний посібник з підготовки кваліфікаційної роботи бакалаврадля здобувачів вищої освіти денної і заочної форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» : Навчальний посібник / І. Ш. Невлюдов, О. І. Филипенко, О. В. Токарєва, С. П. Новоселов, О. В. Сичова. Харків : Видавництво Іванченка І. С., 2023. – 151 с.

5. Дебелий Я., Петрушко Л. Термінал «Нової пошти» сортує до 50 000 посилок на годину. Що відбувається з посилкою у надрах найсучаснішого логістичного центру України // Forbes Ukraine. 23 квітня 2021. URL: https://forbes.ua/company/nova-poshta-terminal-16042021-1387 (дата звернення: 08.06.2026).

6. Nova Poshta opens first phase of KIT parcel sorting center in Kyiv // Interfax-Ukraine. 2018. URL: https://en.interfax.com.ua/news/general/489693.html (дата звернення: 08.06.2026).

7. Amazon uses robots that sort, lift, and carry packages—see them in action // Amazon. URL: https://www.aboutamazon.com/news/operations/amazon-robotics-robots-fulfillment-center (дата звернення: 08.06.2026).

8. Keith R., La H. M. Review of Autonomous Mobile Robots for the Warehouse Environment // arXiv. 2024. URL: https://arxiv.org/abs/2406.08333 (дата звернення: 08.06.2026).

9. Robolist. AMR Warehouse Guide: Autonomous Mobile Robots in Warehousing. URL: https://www.robolist.ai/guides/amr-warehouse (дата звернення: 08.06.2026).

10. Бєлов П. О., Стеценко К. В. Навігація та управління мобільним роботом // Automation and Development of Electronic Devices (ADED'2022). Харків : ХНУРЕ, 2022. С. 11–14. URL: https://openarchive.nure.ua/handle/document/20670

11. Незовибатько О. І. Методи навігації мобільних наземних роботів в недетермінованих середовищах // Радіоелектроніка та молодь у XXI столітті. Харків : ХНУРЕ, 2021. Т. 3. С. 75–76. URL: https://openarchive.nure.ua/handle/document/17009

12. Коваленко М. Схема керування транспортними роботами на основі візуальних ознак // Automation and Development of Electronic Devices (ADED'2023). Харків : ХНУРЕ, 2023. Вип. 2. С. 92–96. URL: https://openarchive.nure.ua/handle/document/24966

13. Li T., Hashimoto H. Localization and Navigation Using QR Code for Mobile Robot in Indoor Environment // Proceedings of IEEE International Conference on Robotics and Biomimetics (ROBIO). 2015. URL: https://ieeexplore.ieee.org/ (дата звернення: 08.06.2026).

14. Arduino Documentation. URL: https://docs.arduino.cc/

15. ESP32 Documentation. URL: https://docs.espressif.com/projects/esp-idf/

16. Raspberry Pi Documentation. URL: https://www.raspberrypi.com/documentation/

17. NVIDIA Jetson Nano Developer Kit. URL: https://developer.nvidia.com/embedded/jetson-nano-developer-kit

18. Bradski G., Kaehler A. Learning OpenCV 4 Computer Vision with Python. Sebastopol : O'Reilly Media, 2021. 976 p.

19. OpenCV Documentation. URL: https://docs.opencv.org/ (дата звернення: 08.06.2026).

20. Flask Documentation. URL: https://flask.palletsprojects.com/ (дата звернення: 08.06.2026).

21. Python Software Foundation. Python Documentation. URL: https://docs.python.org/3/ (дата звернення: 08.06.2026).

22. Dorf R. C., Bishop R. H. Modern Control Systems. 13th Edition. Pearson Education, 2017. 1040 p.

23. Ковриго Ю. М. Основи теорії автоматичного керування. Харків : ХНУРЕ, 2018.

24. Upton E., Halfacree G. Raspberry Pi User Guide. 5th Edition. Wiley, 2022. 312 p.

25. Quigley M., Gerkey B., Smart W. D. Programming Robots with ROS: A Practical Introduction to the Robot Operating System. Sebastopol : O'Reilly Media, 2015. 448 p.

26. Open Robotics. Robot Operating System (ROS) Documentation. URL: https://docs.ros.org/ (дата звернення: 02.07.2026).

27. RoboDK Documentation. URL: https://robodk.com/doc/en/ (дата звернення: 02.07.2026).

28. OpenCV Documentation. URL: https://docs.opencv.org/4.x/ (дата звернення: 02.07.2026).

29. Python Software Foundation. Python 3 Documentation. URL: https://docs.python.org/3/

30. Комплекс навчально-методичного забезпечення навчальної дисципліни «Безпека праці в індустрії ІТ-технологій» підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету [http://catalogue.nure.ua/knmz] / ХНУРЕ; розроб.: Т. Є. Стиценко, Г. В. Пронюк, Н. М. Сердюк. – Харків, 2017. – 122 с.;

31. ДСанПіН 3.3.2.007-98 (вимоги до роботи з ПК).