

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління  
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)  
(рівень вищої освіти)

Інтелектуальна мобільна платформа з функцією відстеження об'єктів  
(тема)

Виконав:

здобувач 4 року навчання

групи КІУКІ-21-7

Романов Д.М.  
(прізвище, ініціали)

Спеціальність 123 Комп'ютерна інженерія

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

Освітня програма Комп'ютерна інженерія  
(повна назва освітньої програми)

Керівник Мірошник А.М.  
(підпис) (прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

Чумаченко С.В.  
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління

Кафедра Автоматизації проектування обчислювальної техніки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія  
(шифр і назва)

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

Освітня програма Комп'ютерна інженерія  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2025 р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Романову Даніилу Максимовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи (проєкту) Інтелектуальна мобільна платформа з функцією відстеження об'єктів

затверджена наказом по університету від "21 " 05 2025 р. № 403Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 10.06.2025

3. Вихідні дані до роботи (проєкту) \_\_\_\_\_

Технічне завдання на розробку інтелектуальної мобільної платформи

Специфікації мови програмування C++

Специфікації мікроконтролера AVR ATmega328P

Інтегроване середовище розробки програмного забезпечення ArduinoIDE

4. Перелік питань, що потрібно опрацювати у роботі \_\_\_\_\_

Мобільні платформи: теоретичні аспекти та аналіз існуючих рішень

Реалізація апаратної складової системи

Реалізація програмного забезпечення системи

Реалізація макетного зразка та результати дослідної експлуатації

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Презентація (18 слайдів)

---

---

---

---

---

6. Консультанти розділів роботи (проекту)

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

7. Дата видачі завдання 06.05.2025

### КАЛЕНДАРНИЙ ПЛАН


№	Назва етапів роботи (проекту)	Термін виконання етапів проекту (роботи)	Примітка
1	Видача теми проекту, узгодження і затвердження	06.05.2025 - 10.05.2025	
2	Аналіз проблемної галузі, постановка задачі, вибір інструментальних засобів	10.05.2025 - 17.05.2025	
3	Розробка моделі системи	17.05.2025 - 24.05.2025	
4	Розробка апаратної платформи системи	24.05.2025 - 31.05.2025	
5	Розробка програмного забезпечення системи	31.05.2025 - 05.06.2025	
6	Проведення випробування системи	05.06.2025 - 07.06.2025	
7	Оформлення пояснювальної записки	07.06.2025 - 10.06.2025	
8	Перевірка виконаного проекту керівником	10.06.2025 - 13.06.2025	
9	Захист проекту	13.06.2025 - 23.06.2025	

Здобувач

  
(підпис)

Романов Д.М.

Керівник роботи (проекту)

  
(підпис)

старший викладач Мірошник А.М.

(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка до атестаційної роботи містить 61 сторінка, 22 рисунки, 4 таблиці, 3 додатки, 17 джерел за переліком посилань.

ІНТЕЛЕКТУАЛЬНА СИСТЕМА, МОБІЛЬНА ПЛАТФОРМА, РОЗПІЗНАВАННЯ ОБ'ЄКТІВ, НЕЙРОННІ МЕРЕЖІ, ДВИГУН ПОСТІЙНОГО СТРУМУ, ARDUINO, HUSKYLENS.

Об'єкт розробки – інтелектуальна мобільна платформа, здатна здійснювати автоматичне відстеження об'єктів у полі зору.

Предмет розробки – апаратно-програмний комплекс, що забезпечує функціонування мобільної платформи з можливістю аналізу та супроводження об'єктів за допомогою модуля комп'ютерного зору.

Мета розробки – створення інтелектуальної мобільної платформи з функцією відстеження об'єктів на базі мікроконтролера, яка забезпечує точне визначення положення об'єктів, стабільну роботу апаратної частини, інтерактивне управління рухом платформи та надійне супроводження об'єктів у реальному часі.

У першому розділі кваліфікаційної роботи проаналізовано теоретичні засади комп'ютерного зору та функціонування мобільних платформ, а також досліджено переваги та недоліки існуючих технічних рішень.

У другому розділі спроектовано апаратну складову мобільної платформи та наведено перелік конкретних технічних компонентів для її реалізації.

У третьому розділі наведено алгоритм та реалізацію програмного забезпечення мобільної платформи.

В останньому розділі проведено тестування макетного зразка системи та наведено результати, які показали працездатність спроектованої системи.

## ABSTRACT

Bachelor's thesis contains 61 pages, 22 figures, 4 tables, 3 appendices, 17 sources according to the list of references.

INTELLIGENT SYSTEM, MOBILE PLATFORM, OBJECT RECOGNITION, NEURAL NETWORKS, DC MOTOR, ARDUINO, HUSKYLENS.

Object of development – an intelligent mobile platform capable of performing automatic object tracking within its field of view.

Subject of development – a hardware and software complex that ensures the operation of the mobile platform with the ability to analyze and track objects using a computer vision module.

Purpose of development – to create an intelligent mobile platform with object tracking functionality based on a microcontroller, providing accurate object position detection, stable hardware operation, interactive control of platform movement, and reliable real-time object tracking.

The first section of the bachelor's thesis analyzes the theoretical foundations of computer vision and the functioning of mobile platforms, as well as explores the advantages and disadvantages of existing technical solutions.

The second section designs the hardware components of the mobile platform and provides a list of specific technical components for its implementation.

The third section presents the algorithm and implementation of the mobile platform's software.

The final section conducts testing of the prototype system and provides results demonstrating the operability of the designed system.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів ..	8
Вступ .....	9
1 Інтелектуальна мобільна платформа .....	11
1.1 Інтелектуальні мобільні платформи .....	11
1.2 Системи комп'ютерного зору та відстеження об'єктів у реальному часі.....	13
1.3 Аналіз існуючих рішень у сфері інтелектуальних мобільних платформ.....	18
1.4 Проблематика та значущість розробки .....	20
1.5 Технічне завдання на проектування .....	22
2 Розробка апаратної платформи системи .....	23
2.1 Опис структурної схеми мобільної платформи .....	24
2.2 Загальні вимоги до апаратної складової автоматизованої системи	33
2.3 Апаратна реалізація контролера .....	33
2.5 Апаратна реалізація підсистеми керування двигунами .....	35
2.6 Апаратна реалізація підсистеми комп'ютерного зору.....	41
3 Розробка програмного забезпечення .....	44
3.1 Структура програмного забезпечення та вибір технологій для реалізації .....	44
3.2 Реалізація програмного забезпечення для контролера інтелектуальної мобільної платформи .....	46
4 Дослідна експлуатація макетного зразка .....	53

4.1 Опис макетного зразка та методики випробувань .....	53
4.2 Результати моделювання роботи тестового зразка .....	55
Висновки .....	58
Перелік джерел посилання.....	60
Додаток А .....	62
Додаток Б .....	71
Додаток В .....	72

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІМП – інтелектуальна мобільна платформа.

МК – мікроконтролер.

ІЧ – інфрачервоний.

ПЗ – програмне забезпечення.

Інтерфейс UART – універсальний асинхронний прийомопередавач (Universal Asynchronous Receiver-Transmitter).

PWM – широтно-імпульсна модуляція (Pulse Width Modulation).

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

HuskyLens – модуль комп'ютерного зору з підтримкою функції відстеження об'єктів.

Arduino IDE – інтегроване середовище розробки для програмування мікроконтролерів Arduino.

L293D – драйвер двигунів для управління напрямком і швидкістю обертання моторів.

ДПС – двигун постійного струму.

Сенсорний модуль – електронний пристрій для зчитування фізичних параметрів об'єктів або середовища.

Відстеження об'єктів – функція визначення і супроводження об'єктів у полі зору системи.

Тестова експлуатація – процес перевірки працездатності розробленої системи в умовах, максимально наближених до реальних.

Комп'ютерний зір – галузь штучного інтелекту, яка забезпечує аналіз візуальної інформації за допомогою цифрових зображень.

## ВСТУП

У сучасному світі спостерігається стрімкий розвиток робототехнічних систем, зокрема мобільних платформ, що інтегрують інтелектуальні функції для виконання складних завдань. Застосування таких платформ варіюється від побутових умов, як-от домашня автоматизація, до промислових і дослідницьких сфер, включаючи логістику, безпеку, та контроль навколишнього середовища. В умовах зростаючої потреби у автономних рішеннях розробка інтелектуальних мобільних платформ із функцією відстеження об'єктів набуває особливої актуальності.

Одним із ключових завдань для подібних систем є точне та швидке виявлення і супровід об'єктів у реальному часі. Це вимагає інтеграції алгоритмів комп'ютерного зору та штучного інтелекту, здатних обробляти великі обсяги даних з камер або інших сенсорів. Проблема ускладнюється обмеженими обчислювальними ресурсами мобільних платформ, що ставить високі вимоги до оптимізації алгоритмів та апаратної частини.

На сьогодні існує низка платформ, які частково вирішують ці завдання. Проте більшість із них орієнтовані на вузьке коло завдань або вимагають значних фінансових витрат. Відсутність універсальних, доступних і легких у використанні рішень ускладнює інтеграцію таких технологій у повсякденну практику, особливо для малих підприємств, освітніх закладів або індивідуальних користувачів.

Необхідність розробки інтелектуальної мобільної платформи з функцією відстеження об'єктів обумовлена потребою в доступних та адаптивних рішеннях, які можуть застосовуватися в різних сферах. Розробка такої системи дозволить автоматизувати рутинні процеси, підвищити ефективність виконання завдань і мінімізувати людський фактор. Наприклад, у логістиці такі платформи можуть використовуватися для контролю руху вантажів, у

безпеці – для моніторингу зон із обмеженим доступом, а в освітніх цілях – для створення навчальних робототехнічних комплексів.

Ця робота спрямована на створення мобільної платформи, що поєднує апаратні компоненти (камери, мікроконтролери, двигуни) та програмне забезпечення для реалізації інтелектуальних функцій. У процесі розробки увага приділяється вибору та інтеграції ефективних алгоритмів для обробки зображень, розпізнавання та відстеження об'єктів, а також оптимізації їх роботи на ресурсно-обмеженій апаратурі.

Мета кваліфікаційної роботи полягає в створенні універсальної інтелектуальної мобільної платформи, яка здатна забезпечити точне відстеження об'єктів і відповідає сучасним вимогам до автономності, продуктивності та адаптивності.

Предмет дослідження: методи та алгоритми інтелектуального відстеження об'єктів у реальному часі, а також програмно-апаратні засоби для їх реалізації на мобільній платформі.

Об'єкт дослідження: інтелектуальна мобільна платформа, яка використовує алгоритми комп'ютерного зору та штучного інтелекту для виявлення і супроводу об'єктів.

## 1 ІНТЕЛЕКТУАЛЬНА МОБІЛЬНА ПЛАТФОРМА

### 1.1 Інтелектуальні мобільні платформи

Інтелектуальна мобільна платформа — це автоматизований пристрій, здатний автономно пересуватися в просторі, аналізувати навколишнє середовище за допомогою сенсорів і виконувати поставлені завдання завдяки використанню алгоритмів штучного інтелекту. Вона поєднує апаратні компоненти (датчики, приводи, мікроконтролери) з програмним забезпеченням, що дозволяє обробляти дані в реальному часі, ухвалювати рішення та адаптуватися до змін у середовищі [1]. Такі платформи знаходять застосування у робототехніці, автоматизації, логістиці, дослідженнях та розвагах.

Інтелектуальні мобільні платформи займають ключову позицію у сучасних технологіях, оскільки вони забезпечують інтеграцію складних обчислювальних алгоритмів, сенсорних систем та автономного управління. Такі платформи дозволяють автоматизувати широкий спектр завдань, починаючи від логістики й аграрного моніторингу до медичних операцій і рятувальних місій. Їхня мобільність і здатність працювати в реальному часі роблять їх незамінними в умовах, що швидко змінюються, або в середовищах, небезпечних для людини.

Науковий інтерес до інтелектуальних платформ обумовлений можливістю їх адаптації до складних сценаріїв, використанням алгоритмів машинного навчання та комп'ютерного зору. Наприклад, платформи можуть аналізувати оточення за допомогою камер, сенсорів і лідара, розпізнавати об'єкти, ухвалювати рішення на основі зібраних даних і навіть співпрацювати з іншими роботизованими системами через мережеві протоколи. Завдяки цим

можливостям вони є ефективними у сфері точного землеробства, моніторингу інфраструктури та виконанні рутинних виробничих операцій.

Особливу увагу привертає їх значення для розвитку технологій штучного інтелекту. Вони є платформою для тестування та впровадження нових моделей глибокого навчання, алгоритмів оптимізації й автономного управління. Зокрема, мобільні платформи слугують експериментальним середовищем для аналізу поведінки автономних агентів у багатофакторних середовищах, таких як міські вулиці чи виробничі лінії.

У практичному застосуванні вони мають значний вплив на економіку, сприяючи підвищенню ефективності виробництва, зменшенню витрат і підвищенню безпеки. Наприклад, у промислових умовах такі платформи можуть автоматизувати завдання, які раніше вимагали значних людських ресурсів. У медичній сфері мобільні платформи можуть використовуватися для транспортування медикаментів або моніторингу стану пацієнтів у реальному часі.

Приклад інтелектуальної мобільної платформи наведено на рисунку 1.1.



Рисунок 1.1 – Інтелектуальна мобільна платформа TurtleBot

Актуальність теми зумовлена стрімким зростанням попиту на автономні мобільні системи у таких галузях, як логістика, сільське господарство, оборона та охорона здоров'я. Наприклад, мобільні платформи з функцією комп'ютерного зору використовуються для моніторингу складів, збору врожаю, транспортування вантажів, а також для забезпечення безпеки. У медичних застосуваннях робототехнічні системи допомагають у реабілітації, догляді за пацієнтами та навіть у проведенні складних операцій.

Таким чином, розробка інтелектуальної мобільної платформи з функцією відстеження об'єктів є важливим внеском у розвиток робототехніки та автоматизації різних процесів, оскільки поєднує сучасні підходи до комп'ютерного зору, алгоритмів обробки даних та інтеграції апаратної й програмної складових у єдину функціональну систему.

## 1.2 Системи комп'ютерного зору та відстеження об'єктів у реальному часі

Розвиток робототехніки та комп'ютерного зору є ключовими напрямками сучасної науки й техніки, що формують основу інновацій у багатьох сферах, від промисловості до медицини та побуту. Робототехніка, як міждисциплінарна галузь, поєднує досягнення механіки, електроніки, інформатики та штучного інтелекту. У цьому контексті комп'ютерний зір стає критично важливим компонентом, що дозволяє машинам «бачити» і розуміти навколишній світ.

Комп'ютерний зір забезпечує обробку й аналіз візуальної інформації, що надходить від камер або інших сенсорів. Це відкриває нові можливості для автоматизації складних завдань, таких як відстеження об'єктів, розпізнавання образів, навігація у просторі, виявлення перешкод та багато іншого. У поєднанні з інтелектуальними алгоритмами ці можливості стають основою для

створення автономних систем, здатних діяти в реальному часі з високим рівнем точності та адаптивності.

Комп'ютерний зір є галуззю, яка об'єднує методи обробки зображень, машинного навчання та аналізу даних для розпізнавання й інтерпретації візуальної інформації [2]. Основні етапи роботи включають:

- Збір та попередня обробка даних: Вхідними даними для системи є зображення або відео, які проходять етапи фільтрації шумів, корекції освітлення та підвищення якості. Це дозволяє підготувати інформацію для подальшого аналізу.

- Аналіз зображень: Алгоритми комп'ютерного зору виконують сегментацію (поділ зображення на окремі частини), виділення ключових ознак (країв, текстур, кольорових областей) і класифікацію об'єктів. Сучасні системи активно використовують глибокі нейронні мережі, такі як Convolutional Neural Networks (CNN), для розпізнавання складних візуальних патернів.

- Прийняття рішень: На основі отриманих даних система генерує дії, наприклад, визначає маршрут руху, взаємодіє з іншими пристроями або попереджає користувача про потенційні загрози.

Відстеження об'єктів у реальному часі ґрунтується на принципах аналізу послідовності кадрів, отриманих із відеопотоку. Для цього використовуються такі методи, як оптичний потік, кореляційний аналіз та алгоритми глибокого навчання, зокрема нейронні мережі. Ці технології дозволяють системі визначати місцезнаходження об'єктів, прогнозувати їхній рух і коригувати поведінку платформи відповідно до змін у середовищі.

Відстеження об'єктів у реальному часі базується на обробці відеопотоку, де кожен кадр аналізується для виявлення та локалізації об'єктів. Основні принципи роботи включають:

- Алгоритми виявлення: Використовуються методи, такі як YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector) або Faster R-CNN, які дозволяють визначати об'єкти в кадрі з високою точністю та швидкістю.
- Трасування об'єктів: Після виявлення об'єкта система забезпечує його безперервне відстеження, використовуючи методи кореляції, оптичного потоку або Kalman Filter. Це дозволяє прогнозувати положення об'єкта у наступних кадрах навіть за наявності перешкод.
- Інтеграція з апаратною частиною: Високопродуктивні процесори, такі як GPU або спеціалізовані чипи для AI (наприклад, NVIDIA Jetson), забезпечують швидкість обробки та низькі затримки, необхідні для реального часу.
- Адаптація до змін середовища: Системи використовують стратегії самонавчання та адаптації, що дозволяє їм ефективно працювати за умов зміни освітлення, перешкод або непередбачуваних рухів об'єктів.

Реалізація відстеження у реальному часі вимагає високої продуктивності апаратної частини, ефективності алгоритмів і оптимізації затримок при обробці даних. Це досягається за рахунок використання спеціалізованих процесорів, таких як GPU або TPU, і сучасних бібліотек, наприклад, OpenCV чи TensorFlow. Важливим аспектом є інтеграція програмного забезпечення з апаратними сенсорами для забезпечення точного синхронного функціонування системи.

Інтеграція комп'ютерного зору з функцією відстеження об'єктів відкриває широкі можливості у сфері автономної робототехніки, наприклад, для розробки дронів, автомобілів з автоматичним керуванням, мобільних роботів для моніторингу та охорони. Такі системи мають потенціал для застосування в медицині, промисловості, сільському господарстві та побуті.

У сучасних умовах технології комп'ютерного зору активно вдосконалюються завдяки прогресу у глибокому навчанні, збільшенню обчислювальних потужностей та доступності великих масивів даних. Це

створює можливості для розробки інтелектуальних систем, здатних ефективно працювати у складних динамічних середовищах. Таким чином, дослідження у цій сфері не лише мають наукову значущість, а й сприяють вирішенню практичних задач, що стоять перед сучасним суспільством.

Штучний інтелект (ШІ) є одним із ключових інструментів сучасних інформаційних технологій, який дозволяє ефективно обробляти складні дані, включаючи текст, зображення, відео, звук та інші типи інформації. Його застосування базується на методах машинного навчання (ML), глибокого навчання (DL) та теорії ймовірностей.

ШІ для розпізнавання та обробки інформації використовує складні математичні моделі, які дозволяють аналізувати великі обсяги даних. Основними компонентами є:

1. Моделі глибокого навчання (Deep Learning).

Глибокі нейронні мережі (DNN) включають кілька шарів штучних нейронів, які дозволяють виявляти складні залежності в даних [3]. Наприклад, Convolutional Neural Networks (CNN) широко використовуються для аналізу зображень, тоді як Recurrent Neural Networks (RNN) підходять для роботи з послідовностями даних, наприклад тексту чи звуку.

Архітектури типу Transformer (зокрема BERT, GPT) забезпечують високоточну обробку природної мови (NLP), використовуючи механізм уваги для аналізу контексту[3].

2. Теорія ймовірностей та статистика.

Для моделювання невизначеності використовуються методи, такі як Баєсівські мережі, які дозволяють враховувати ймовірності різних подій.

Марковські процеси прийняття рішень (MDP) застосовуються для прогнозування дій у складних системах [3].

3. Обробка великих даних.

ШІ інтегрується з технологіями Big Data для роботи з масивними наборами інформації, що забезпечує навчання моделей на значних обсягах даних, необхідних для високої точності.

Методи розпізнавання та обробки інформації:

1. Розпізнавання зображень та відео:

Використовується сегментація зображень для виділення об'єктів на фоні. Методики, такі як Region-Based CNN (R-CNN) або YOLO (You Only Look Once), дозволяють одночасно розпізнавати та локалізувати об'єкти [2].

Алгоритми для аналізу відео включають методи трасування об'єктів у послідовностях кадрів, наприклад SORT (Simple Online and Realtime Tracking).

2. Обробка текстової інформації:

NLP включає токенізацію, морфологічний аналіз, семантичну сегментацію та синтаксичний аналіз. Наприклад, Word2Vec або GloVe використовуються для представлення слів у вигляді векторів.

Для класифікації тексту застосовуються моделі, такі як Support Vector Machines (SVM) чи нейронні мережі [3].

3. Розпізнавання мовлення та звуку:

Обробка сигналів включає попереднє перетворення даних (наприклад, за допомогою FFT або MFCC для аналізу спектральних ознак).

Гібридні моделі (RNN у поєднанні з CNN) використовуються для високоточного розпізнавання мовлення [2].

ІІІ забезпечує високу точність і швидкість обробки інформації, однак має обмеження, зокрема високу вимогливість до обчислювальних ресурсів, потребу в якісних навчальних даних та складність пояснення рішень моделей (проблема "чорного ящика"). Використання ІІІ для розпізнавання та обробки інформації дозволяє вирішувати складні завдання у багатьох галузях, створюючи нові можливості для автоматизації та підвищення ефективності процесів.

### 1.3 Аналіз існуючих рішень у сфері інтелектуальних мобільних платформ

У сучасному світі технології відстеження об'єктів стали невід'ємною частиною багатьох сфер життя. Від автономних транспортних засобів до систем безпеки – ці технології знаходять все ширше застосування. Розглянемо детально основні технологічні рішення, що формують базис сучасних мобільних платформ з функцією відстеження об'єктів.

Системи комп'ютерного зору є фундаментальним компонентом таких платформ. Вони працюють подібно до людського зору, але з використанням складних алгоритмів обробки зображень. Наприклад, при спостереженні за рухомим автомобілем система не просто фіксує його положення на кожному кадрі, але й аналізує його швидкість, напрямок руху та прогнозує подальшу траєкторію. Сучасні алгоритми комп'ютерного зору, такі як YOLO (You Only Look Once), здатні розпізнавати об'єкти з вражаючою точністю навіть у складних умовах освітлення та при частковому перекритті об'єктів.

Важливу роль відіграють сенсорні технології, які формують комплексну систему сприйняття навколишнього середовища. Розглянемо приклад автономного робота-навантажувача на складі. Такий робот використовує комбінацію різних сенсорів: лідар створює тривимірну карту простору, допомагаючи уникати зіткнень з перешкодами; стереокамери забезпечують точне визначення відстані до об'єктів та їх розмірів; інерційні датчики контролюють стабільність платформи під час руху та перевезення вантажів.

Сучасні системи локалізації та картографування (SLAM) представляють собою складний симбіоз апаратних та програмних рішень. Уявіть собі робота-прибиральника, який вперше потрапив у нове приміщення. Використовуючи технологію SLAM, він одночасно створює карту приміщення та визначає своє положення на ній. При цьому система постійно оновлює інформацію,

враховуючи зміни в навколишньому середовищі, наприклад, переміщення меблів або появу нових перешкод.

Важливим аспектом є обробка та фільтрація даних з різних джерел. Наприклад, у системі безпеки торгового центру відеокамери можуть виявити підозрілий об'єкт, а інфрачервоні датчики підтвердити його наявність навіть при поганому освітленні. Така комплексна система зменшує кількість помилкових спрацьовувань та підвищує загальну надійність системи.

Алгоритми прогнозування руху об'єктів стали значно складнішими та точнішими. Розглянемо приклад системи контролю дорожнього руху. Сучасні алгоритми не просто відстежують поточне положення транспортних засобів, але й аналізують патерни їх руху, враховують взаємодію між різними учасниками руху та навіть можуть передбачати потенційно небезпечні ситуації за кілька секунд до їх виникнення.

Сучасні системи відстеження об'єктів досягли значних успіхів у точності та надійності роботи. Наприклад, у складських умовах автоматизовані системи здатні відстежувати тисячі об'єктів одночасно з точністю позиціонування до кількох сантиметрів. Системи відеоспостереження можуть розпізнавати та відстежувати людей у натовпі, зберігаючи стабільність роботи навіть при частковому перекритті об'єктів.

Проте існують і суттєві обмеження. Наприклад, автономний дрон з функцією відстеження об'єктів може відмінно працювати при денному світлі, але його ефективність значно знижується в умовах поганої видимості або при складних погодних умовах. Енергоспоживання залишається критичним фактором – потужні процесори, необхідні для обробки даних в реальному часі, швидко виснажують батареї, що обмежує час автономної роботи.

Особливої уваги заслуговує проблема калібрування мультисенсорних систем. У випадку з безпілотним автомобілем, наприклад, навіть незначне відхилення в калібруванні датчиків може призвести до неточної оцінки відстані до об'єктів, що є критичним для безпеки руху.

Перспективи розвитку галузі пов'язані з впровадженням нових технологічних рішень. Наприклад, використання нейроморфних процесорів, що імітують роботу людського мозку, може значно знизити енергоспоживання систем при збереженні їх продуктивності. Розвиток технологій квантових обчислень відкриває можливості для створення більш ефективних алгоритмів обробки даних.

У контексті практичного застосування варто розглянути приклад сучасного логістичного центру. Тут мобільні платформи з функцією відстеження об'єктів працюють у складній екосистемі, де взаємодіють автономні навантажувачі, системи конвеєрів та людські оператори. Кожна платформа повинна не тільки відстежувати своє положення та положення вантажів, але й координувати свої дії з іншими учасниками процесу, уникати зіткнень та оптимізувати маршрути переміщення.

Таким чином, аналіз існуючих рішень показує, що галузь мобільних платформ з функцією відстеження об'єктів знаходиться на етапі активного розвитку. Незважаючи на наявні обмеження, постійне вдосконалення технологій та алгоритмів дозволяє створювати все більш ефективні та надійні системи.

#### 1.4 Проблематика та значущість розробки

Сучасний етап розвитку мобільних платформ з функцією відстеження об'єктів характеризується наявністю суттєвих технологічних обмежень, що перешкоджають їх широкому впровадженню. Одним з ключових викликів є проблема енергоефективності. Існуючі системи потребують значних обчислювальних ресурсів для обробки даних у реальному часі, що призводить до високого енергоспоживання та обмеженого часу автономної роботи. Це особливо критично для мобільних платформ, які повинні функціонувати тривалий час без підзарядки.

Суттєвим недоліком є також обмежена адаптивність систем до змінних умов середовища. Більшість існуючих рішень демонструють високу ефективність лише в контрольованих умовах, але їх продуктивність значно знижується при зміні освітлення, погодних умов або в складних динамічних середовищах. Наприклад, системи відстеження, що базуються виключно на комп'ютерному зорі, можуть втрачати точність при недостатньому освітленні або в умовах підвищеної запиленості.

Розвиток промисловості та логістики створює зростаючий попит на інтелектуальні мобільні платформи, здатні ефективно функціонувати в різноманітних умовах експлуатації. Сучасні виробничі процеси вимагають створення систем, які можуть адаптуватися до змінних умов роботи без необхідності постійного перепрограмування або складного налаштування.

Економічна ефективність є критичним фактором при впровадженні таких систем у виробництво. Існуючі рішення часто потребують дорогого обладнання та складного технічного обслуговування, що значно підвищує сукупну вартість володіння. Це створює значний бар'єр для малих та середніх підприємств, які не можуть дозволити собі впровадження високовартісних автоматизованих систем.

Розробка нової інтелектуальної мобільної платформи обумовлена необхідністю подолання існуючих технологічних обмежень та створення більш ефективного рішення. Інтеграція сучасних методів штучного інтелекту та оптимізованих алгоритмів обробки даних дозволить створити систему, здатну працювати в реальних виробничих умовах з високою надійністю та енергоефективністю.

Важливим аспектом є розробка мультисенсорної системи, яка може компенсувати недоліки окремих типів датчиків. Наприклад, комбінування даних від камер, лідарів та інерційних сенсорів дозволить забезпечити стабільне відстеження об'єктів у різних умовах освітлення та при наявності перешкод. При цьому інтелектуальні алгоритми обробки даних повинні

оптимально використовувати доступні обчислювальні ресурси, забезпечуючи необхідну продуктивність при мінімальному енергоспоживанні.

Розробка такої платформи також спрямована на створення модульної та масштабованої архітектури, що дозволить легко адаптувати систему під конкретні потреби користувачів. Це особливо важливо для промислового сектору, де вимоги до функціональності системи можуть значно відрізнятися залежно від специфіки виробництва.

Інтелектуальна складова системи повинна забезпечувати не тільки точне відстеження об'єктів, але й прогнозування їх поведінки, що дозволить оптимізувати планування маршрутів та підвищити загальну ефективність роботи. Наприклад, у складському приміщенні система зможе передбачати потенційні конфлікти маршрутів різних мобільних платформ та заздалегідь коригувати їх траєкторії руху.

Економічна доцільність розробки підтверджується потенціалом зниження експлуатаційних витрат за рахунок використання більш енергоефективних алгоритмів та оптимізованої апаратної платформи. Крім того, модульна архітектура системи дозволить користувачам обирати необхідний функціонал без переплати за надлишкові можливості.

Таким чином, розробка нової інтелектуальної мобільної платформи є актуальним завданням, вирішення якого дозволить подолати існуючі технологічні обмеження та створити більш ефективне рішення для промислового застосування. Успішна реалізація проекту сприятиме подальшій автоматизації виробничих процесів та підвищенню конкурентоспроможності підприємств.

### 1.5 Технічне завдання на проектування

Метою розробки є створення енергоефективної та економічно доцільної мобільної платформи, здатної автономно переміщуватися у просторі та

здійснювати відстеження об'єктів з високою точністю. Система повинна забезпечувати стабільну роботу в різних умовах освітлення та при наявності перешкод, а також мати можливість адаптації до різних промислових застосувань.

Об'єктом розробки є інтелектуальна мобільна платформа, що представляє собою автономну систему, здатну до переміщення у просторі та відстеження об'єктів з використанням комплексу сенсорів та алгоритмів машинного навчання. Платформа призначена для застосування в промислових умовах та повинна забезпечувати надійне функціонування в різних умовах експлуатації.

Предметом розробки є програмно-апаратний комплекс, що включає систему технічного зору, набір сенсорів для визначення положення у просторі, обчислювальний модуль для обробки даних та прийняття рішень, а також програмне забезпечення для керування платформою та обробки інформації.

Апаратна складова проектованої системи складається з наступних компонентів: контролер, підсистема комп'ютерного зору, підсистема керування двигунами та живлення.

Функціонально система повинна аналізувати навколишнє середовище та при ідентифікації об'єкта фіксувати його положення та розміри. Далі проводиться аналіз координат центру об'єкта щодо середньої лінії зображення та обчислення відхилення об'єкта від центральної осі платформи. Наступним іде розрахунок швидкостей для кожного двигуна залежно від положення об'єкта. Якщо об'єкт зсунувся вліво – збільшення швидкості правих двигунів і зменшення швидкості лівих, якщо об'єкт зсунувся вправо – навпаки, якщо об'єкт знаходиться у центрі – підтримання прямолінійного руху. Передача сигналів на драйвер двигунів для реалізації відповідного руху. Окрім цього ще система повинна регулювати дистанцію до об'єкта – збільшення швидкості при наблизенні до об'єкта або її зменшення у разі перевищення оптимальної дистанції.

2Розробка апаратної платформи системи

У попередньому розділі кваліфікаційної роботи було детально розглянуто предметну область, а саме теоретичні засади комп'ютерного зору та відстеження об'єктів у реальному часі та існуючі технології у сфері мобільних платформ.

У цьому розділі буде детально описано процес створення апаратної складової інтелектуальної мобільної платформи та вибір необхідних для цього компонентів.

## 2.1 Опис структурної схеми мобільної платформи

Структурна схема інтелектуальної мобільної платформи наведена на рисунку 2.1.

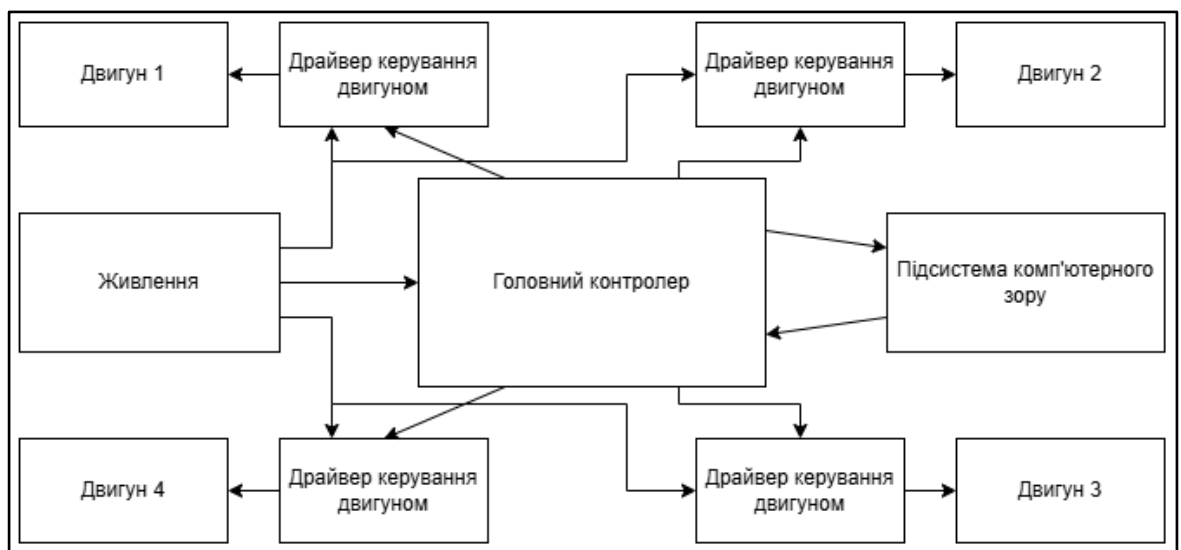


Рисунок 2.1 – Структурна схема інтелектуальної мобільної платформи

Як видно з рисунка 2.1 структурна схема інтелектуальної мобільної платформи складається з декількох частин.

В основі лежить головний контролер, який виконує роль «мозку» та керує усією системою. Його основна задача приймати сигнали від підсистеми комп'ютерного зору про навколишнє середовище та об'єкти, а потім на основі цих даних формувати керуючі сигнали для двигунів.

Враховуючи розвиток сучасних технологій у сфері мікро- та нано-електроніки, у якості контролера системи доцільним буде використання мікроконтролерів, адже їх обчислювальна потужність та невеликий розмір дозволяють інтегрувати їх, майже, у будь-яку систему.

Мікроконтролер (MCU) — це інтегральна мікросхема, призначена для програмного керування електронними пристроями[4]. Він поєднує в одному кристалі процесорне ядро, шини даних і команд, периферійні модулі, оперативну та постійну пам'ять. По суті, це компактний комп'ютер, вбудований у чіп, який функціонує як периферійний процесор, забезпечуючи автономне виконання завдань у складі електронних систем. Приклади мікроконтролерів представлені на рисунку 2.2.

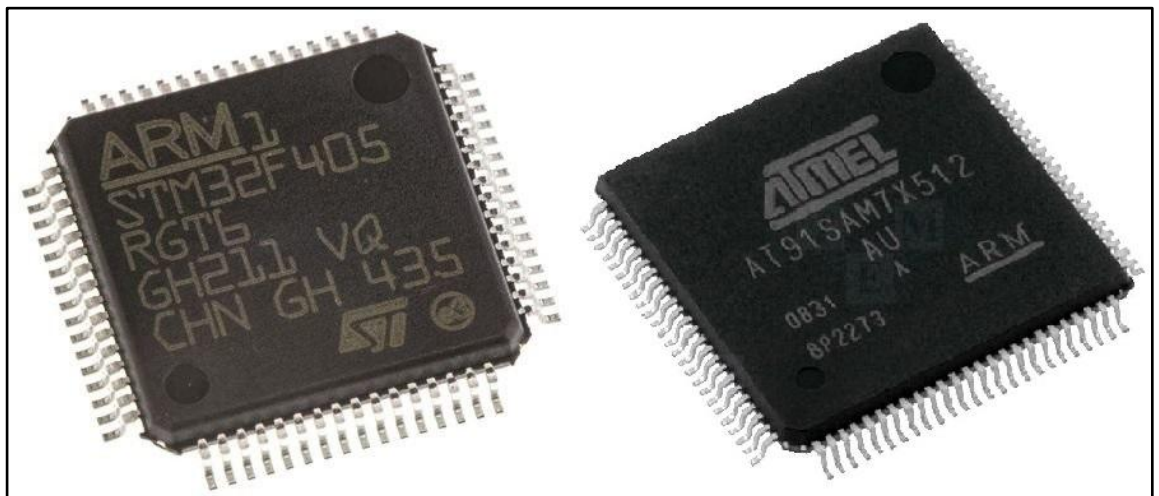


Рисунок 2.2 – Мікроконтролери

При розробці мікроконтролерів необхідно знайти баланс між розмірами і вартістю з одного боку та гнучкістю і продуктивністю з іншого. Оптимальні параметри можуть суттєво змінюватися залежно від застосування, тому існує широкий спектр мікроконтролерів з різними архітектурами, обсягом і типом пам'яті, периферійними модулями та варіантами корпусів.

Неповний список периферійних пристроїв, які можуть використовуватися в мікроконтролерах, включає:

- універсальні цифрові порти, які можна налаштовувати як на введення, так і на виведення;
- різні інтерфейси введення-виводу, такі як UART, I<sup>2</sup>C, SPI, CAN, USB, IEEE 1394, Ethernet;
- аналого-цифрові та цифро-аналогові перетворювачі;
- компаратори;
- широтно-імпульсні модулятори (ШІМ-контролер);
- таймери;
- контролери безколекторних двигунів, у тому числі крокових;
- контролери дисплеїв та клавіатур;
- радіочастотні приймачі та передавачі;
- масиви вбудованої флеш-пам'яті;
- вбудований тактовий генератор і сторожовий таймер.

На відміну від мікропроцесорів, мікроконтролери часто використовують гарвардську архітектуру, розділяючи пам'ять для команд і даних. Вони можуть включати вбудовану енергонезалежну пам'ять для програм і даних, причому дешевші моделі підтримують одноразовий запис. Інші типи дозволяють багаторазове оновлення програм у пам'яті, що розширює можливості їх використання в різних умовах.

Обмеження за ціною та споживанням енергії впливають на тактову частоту роботи мікроконтролерів. Виробники прагнуть забезпечити високу продуктивність, водночас пропонуючи модифікації з різними частотами та

рівнями напруги живлення. Багато моделей використовують статичну пам'ять для оперативних регістрів, що дозволяє зберігати дані навіть при повній зупинці тактового генератора. У таких пристроях часто реалізовані режими енергозбереження, які дозволяють відключати частину периферії або обчислювальні блоки, забезпечуючи гнучкість і економічність у використанні енергії.

Використання в сучасному мікроконтролері достатнього потужного обчислювального пристрою з широкими можливостями, побудованого на одній мікросхемі замість цілого набору значно знижує розміри, енергоспоживання і вартість побудованих на його базі пристроїв.

Так, як платформа мобільна, тобто вона повинна рухатися, необхідно забезпечити таку можливість. Електронні компоненти, що здатні перетворювати електричну енергію у рух, називаються двигунами.

Двигун — це технічний пристрій, що перетворює один вид енергії (хімічну, електричну, теплову тощо) на механічну роботу, яка забезпечує рух об'єктів або виконання інших механічних завдань[5]. Основними характеристиками двигуна є потужність, ефективність, швидкість обертання або лінійного руху, а також тип енергії, яку він використовує. Двигуни класифікуються за джерелом енергії (електричні, теплові, гідравлічні тощо) і принципом дії (ротаційні, поршневі, турбінні тощо). Вони є ключовими елементами багатьох механічних систем, від побутових приладів до промислових машин.

Двигуни поділяються на первинні та вторинні залежно від джерела енергії. Первинні двигуни перетворюють природні види енергії, такі як сила вітру, гравітація або тепловий ефект від спалювання палива чи ядерних реакцій, у механічну енергію. Прикладами є вітрові та водяні колеса, теплові двигуни. Вторинні двигуни використовують вже перетворену енергію, наприклад, електричну чи гідравлічну. Сюди належать електричні (рис.2.3), пневматичні та гідравлічні двигуни, які є ключовими для передачі енергії у сучасних технологічних системах.

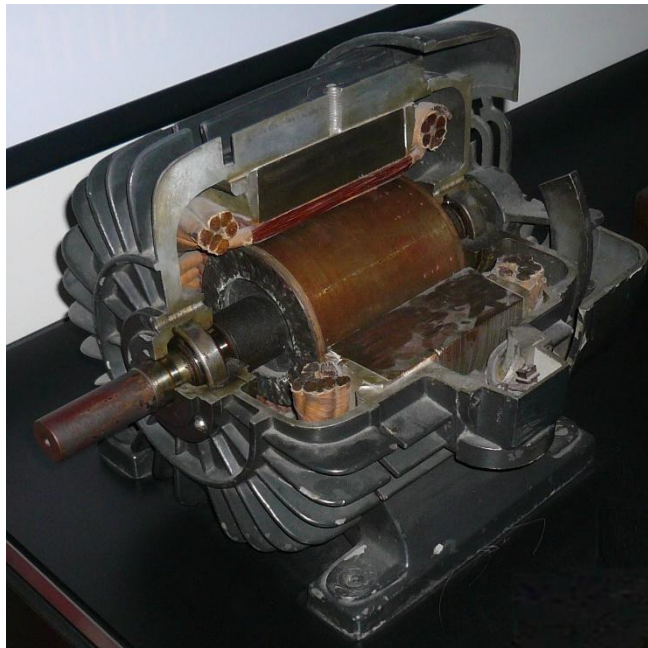


Рисунок 2.3 - Електродвигун

За джерелом енергії двигуни класифікують наступним чином:

- Електричні (постійного струму (електродвигун постійного струму), змінного струму (синхронні та асинхронні));
- електростатичні;
- хімічні;
- ядерні;
- гравітаційні;
- пневматичні;
- гідравлічні;
- лазерні.

За типами руху двигуни класифікують наступним чином:

- обертальний рух твердих тіл;
- поступальний рух твердих тіл;
- зворотно-поступальний рух твердих тіл;
- рух реактивного струменя;
- інші види руху.

Електродвигуни, що забезпечують поступальний та/або зворотно-поступальний рух твердого тіла:

- лінійні;
- індукційні;
- п'єзоелектричні.

Враховуючи те, що проєктована система є електронною, доцільним буде використання саме електродвигунів. Проте, щоб керування двигунами було коректним необхідно між ними та контролером поставити драйвер керування двигуном.

Драйвер керування двигуном — це електронний пристрій або модуль, що забезпечує контроль і управління електричними двигунами [5]. Він виступає посередником між мікроконтролером або іншим керуючим пристроєм і двигуном, забезпечуючи передачу сигналів управління та живлення, адаптованих до вимог конкретного типу двигуна. Зовнішній вигляд драйвера керування двигуном показано на рисунку 2.4.

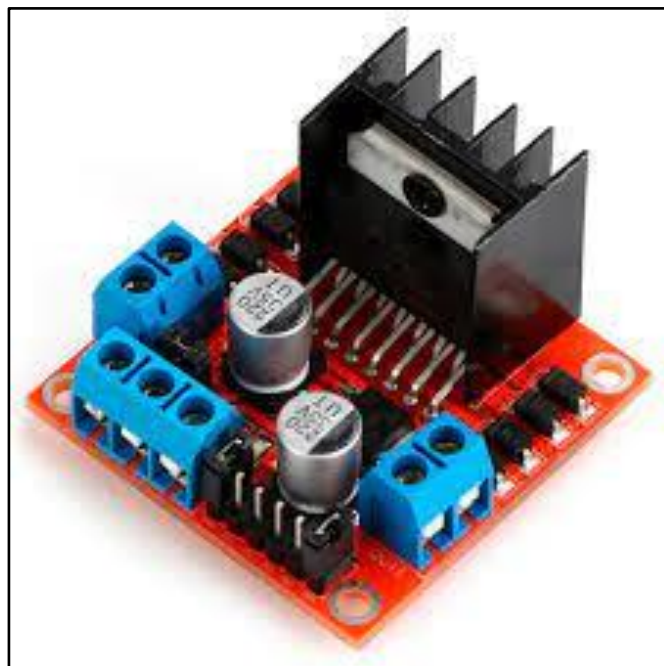


Рисунок 2.4 – Драйвер керування двигуном

Основні функції драйвера:

1. Підсилення сигналу: Сигнали від мікроконтролера мають низьку потужність і не можуть безпосередньо керувати двигуном. Драйвер підсилює ці сигнали до необхідного рівня.

2. Напрямок і швидкість обертання: Драйвери дозволяють змінювати напрямок обертання двигуна шляхом комутації полярності живлення. Крім того, швидкість регулюється шляхом зміни середнього значення напруги (часто за допомогою широтно-імпульсної модуляції — PWM).

3. Захист двигуна: Багато драйверів оснащені механізмами захисту від перевантаження по струму, перегріву, короткого замикання та інших несправностей, що продовжує термін служби двигуна.

4. Сумісність з різними типами двигунів: Драйвери розробляються для роботи з різними видами двигунів: постійного струму, кроковими, безщітковими (BLDC) або асинхронними. Кожен тип вимагає особливого алгоритму управління.

Особливості використання:

- Драйвери повинні відповідати характеристикам двигуна (напруга, струм).
- Важливим аспектом є тепловідведення, оскільки драйвери можуть нагріватися при тривалій роботі.
- Для інтеграції в систему драйвери часто мають стандартні інтерфейси (I2C, SPI, UART).

Таким чином, драйвер керування двигуном є критичним компонентом у системах, що вимагають високоточного управління рухом, таких як роботи, дрони, промислові машини тощо.

Для реалізації функціоналу відстеження об'єктів у реальному часі необхідно передбачити підсистему комп'ютерного зору.

Основою підсистеми комп'ютерного зору є камера (рис.2.5), яка виступає у ролі сенсора «зору», для того, щоб мобільна платформа могла «бачити» та аналізувати те, що відбувається навколо неї.

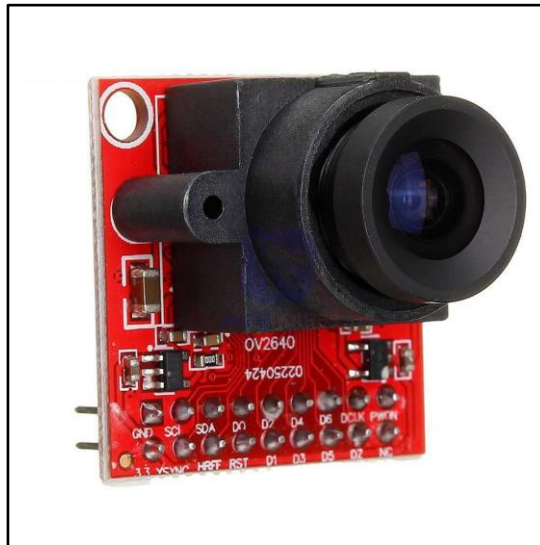


Рисунок 2.5 – Модуль камери

Камера перетворює оптичні зображення у цифрові сигнали, які можуть бути аналізовані алгоритмами машинного зору. Камера працює на основі світлочутливих елементів (наприклад, CMOS або CCD-матриць), які реєструють інтенсивність світла на кожному пікселі [5].

У системах комп'ютерного зору камера відіграє роль джерела даних, забезпечуючи вхідну інформацію для алгоритмів, що виконують розпізнавання об'єктів, аналіз сцени або визначення руху. Якість і точність роботи таких систем значною мірою залежать від характеристик камери: роздільної здатності, швидкості захоплення кадрів, динамічного діапазону та чутливості до світла.

Камера також дозволяє системам працювати в реальному часі, що є важливим для завдань, таких як автономне водіння, роботи-маніпулятори або системи моніторингу. Вибір камери залежить від специфіки завдання: для простих застосувань достатньо стандартної веб-камери, тоді як для складних

завдань можуть використовуватися спеціалізовані модулі, наприклад, стереокамери, тепловізори або камери з високою частотою кадрів.

Таким чином, камера не лише забезпечує зорове сприйняття, але й створює основу для роботи алгоритмів комп'ютерного зору, що перетворюють необроблені дані у структуровану інформацію для прийняття рішень.

Але сама собі камера може лише збирати візуальну інформацію та перетворювати її на цифровий вигляд, за для аналізу цих даних необхідно передбачити у підсистемі комп'ютерного зору обчислювальну потужність та пам'ять, а також можливість інтеграції алгоритмів штучного інтелекту для відстеження об'єктів у реальному часі.

Для автономного функціонування мобільної платформи необхідно реалізувати правильне енергоживлення, щоб усі компоненти системи працювали чітко та без збоїв.

Максимального рівня енергетичної автономності можливо досягти за допомогою акумуляторів, адже вони дозволяють накопичувати електроенергію і потім використовувати її для роботи системи. Проте, на відміну від живлення від мережі, живлення від акумуляторів є «слабшим», тож необхідно точно прорахувати енерговитрати усієї системи, щоб правильно підібрати акумулятори по їх технічних характеристиках. Зовнішній вигляд акумулятора представлено на рисунку 2.6.



Рисунок 2.6 – Акумулятори

## 2.2 Загальні вимоги до апаратної складової автоматизованої системи

Основними критеріями вибору технічних рішень для компонентів структурної схеми є забезпечення мінімально необхідного функціоналу, відповідність вимогам надійності, зниження вартості та енергоспоживання, навіть у випадках живлення від електромережі, а також простота технічного обслуговування і дотримання експлуатаційних та екологічних стандартів. Хоча ці параметри зазвичай розглядаються в порядку пріоритетності, для окремих елементів системи їх значущість може варіюватися залежно від специфіки завдання.

У рамках атестаційної роботи основний акцент зроблено на програмній складовій системи, включаючи розробку алгоритмів та структур даних. Тому апаратна частина базується на готових технічних рішеннях, які за необхідності адаптуються до вимог проекту, що дозволяє уникнути повної розробки апаратного забезпечення з нуля.

Крім того, система повинна відповідати кліматичним нормам середовища експлуатації. Кліматичні умови поділяються на робочі параметри, які забезпечують стабільну роботу пристроїв у межах заданих характеристик, та граничні параметри, що допускають короточасні відхилення від номінальних значень із можливістю повернення до початкових характеристик після припинення дії несприятливих факторів.

## 2.3 Апаратна реалізація контролера

Контролер є найважливішою частиною системи, адже виконує одразу декілька ключових функцій, а саме:

- контроль роботи усієї системи;
- отримання даних з камери та їх аналіз для подальшого керування драйверами двигунів, щоб за необхідності змінювати рух платформи;

- з'єднання між собою усіх компонентів системи.

Враховуючи усе перелічене раніше, необхідно у якості контролера обрати компонент, який буде підходити за наступними характеристиками:

- мати достатню обчислювальну потужність для аналізу даних з камери та управління драйверами двигунів;
- мати достатній обсяг оперативної пам'яті (ОЗП) для роботи програмного забезпечення;
- мати усі необхідні інтерфейси: UART, SPI, I<sup>2</sup>C;
- мати компактний форм-фактор та низький рівень енергоживлення;
- мати відносно невелику вартість.

Беручи до уваги всі вимоги, що висуваються до таких систем, мікроконтролер є оптимальним вибором для реалізації контролера інтелектуальної мобільної платформи. Як уже зазначалося, використання мікропроцесорних технологій значно спрощує архітектуру системи, оскільки один мікропроцесор може виконувати функції кількох аналогових компонентів. З цієї причини було обрано мікропроцесорний підхід для розробки контролера в даній системі.

Сучасні мікропроцесори представлені широким спектром моделей, які відрізняються не лише зовнішнім виконанням, а й технічними характеристиками, функціональними можливостями та умовами експлуатації. При виборі мікропроцесора важливо враховувати такі параметри, як обчислювальна потужність, обсяг оперативної та постійної пам'яті, наявність графічного ядра, підтримка різноманітних інтерфейсів, а також розміри та експлуатаційні характеристики.

Одним із найпоширеніших варіантів, що поєднує доступність і функціональність, є технічне рішення на основі мікроконтролера AVR ATmega328P (додаток Б), реалізоване у вигляді плати Arduino UNO [6]. Зовнішній вигляд плати представлено на рисунку 2.7, а технічні характеристики – у таблиці 2.1.

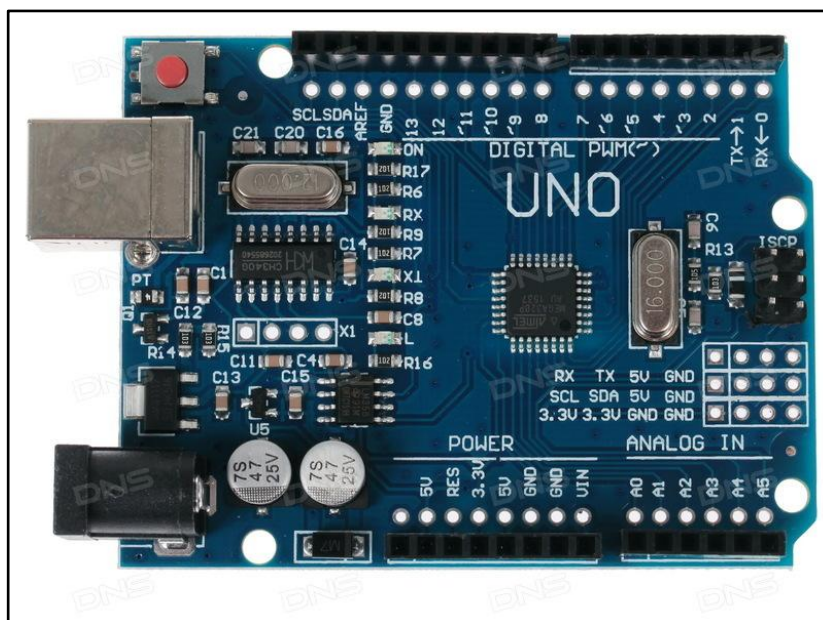


Рисунок 2.7 – Плата Arduino UNO

Таблиця 2.1 – Основні технічні характеристики плати Arduino UNO

Характеристика	Значення
Мікроконтролер	ATmega328
Робоча напруга	5 В
Вхідна напруга (рекомендована)	7-12 В
Вхідна напруга (гранична)	6-20 В
Цифрові входи/виходи	14 (6 – виходи ШІМ)
Аналогові входи	6
Постійний струм через входи/виходи	40 мА
Постійний струм для виходу 3.3 В	50 мА
Флеш-пам'ять	32 Кб (ATmega328)
ОЗП	2 Кб (ATmega328)
EEPROM	1 Кб (ATmega328)
Тактова частота	16 МГц
Підтримка інтерфейсів	SPI, I2C, Serial

## 2.5 Апаратна реалізація підсистеми керування двигунами

Як було описано вище, для реалізації руху платформи необхідне використання двигунів. У сучасному світі дуже велика кількість двигунів, які

відрізняються як характеристиками (потужність, розміри, енергоживлення) так і особливостями використання (двигуни постійного чи змінного струму, синхронні чи асинхронні), проте враховуючи вимоги, які зазвичай висуваються до мобільних платформ, доцільним буде використання двигунів постійного струму (ДПС).

Зовнішній вигляд та основні технічні характеристики ДПС наведено на рисунку 2.8 та у таблиці 2.2 відповідно.



Рисунок 2.8 – Двигун постійного струму

Таблиця 2.2 – Технічні характеристики НС-SR04

Характеристика	Значення
Тип двигуна	DC Motor
Номінальна напруга живлення	6–12 В
Номінальний струм	0,5–2 А
Максимальний струм	3 А
Номінальна швидкість	3000–12000 об/хв
Максимальний обертальний момент	0,5–1 Н·м
Коефіцієнт корисної дії (ККД)	70–85%

Діапазон робочих температур	-20°C до +60°C
Вага двигуна	200–500 г
Тип підшипників	Кулькові або втулкові
Система охолодження	Пасивна
Ресурс роботи	1000–5000 годин

Двигун постійного струму (ДПС) – це електромеханічний пристрій, який перетворює електричну енергію постійного струму на механічну енергію обертального руху [5]. Основу його роботи становить явище електромагнітної індукції, згідно з яким взаємодія магнітного поля і струму в провіднику створює обертальний момент. Завдяки своїй простоті конструкції, широкому діапазону регулювання швидкості та надійності, двигуни постійного струму широко використовуються в системах автоматизації, транспортних засобах, робототехніці та побутовій техніці. Внутрішню конструкцію ДПС наведено на рисунку 2.9.

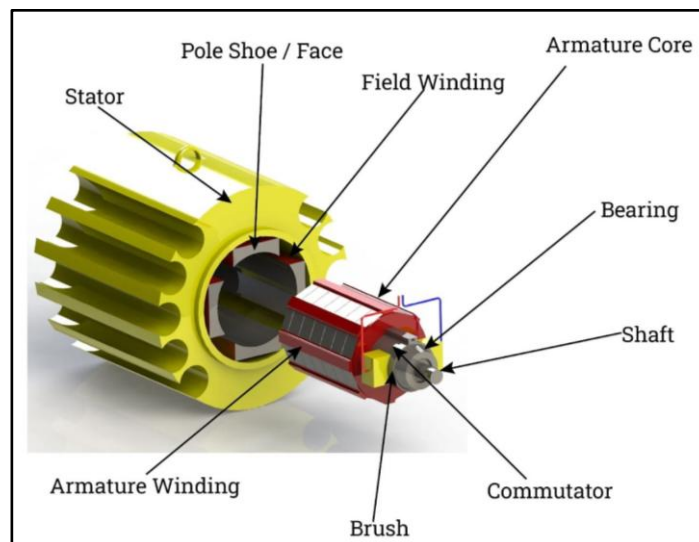


Рисунок 2.9 – Внутрішня конструкція двигуна постійного струму

Принцип роботи двигуна постійного струму базується на дії електромагнітних сил, які виникають у магнітному полі при проходженні

електричного струму через провідник. Двигун складається з таких основних елементів:

- Статор – нерухома частина двигуна, яка забезпечує постійне магнітне поле. У деяких конструкціях статор включає постійні магніти, а в інших – електромагніти.

- Ротор (якір) – рухома частина двигуна, яка обертається під дією електромагнітних сил. На роторі розташовані провідники, через які проходить електричний струм.

- Комутатор – механічний або електронний пристрій, що змінює напрямок струму в обмотках ротора, забезпечуючи постійний напрямок обертального моменту.

- Щітки – елементи, які забезпечують передачу струму від зовнішнього джерела до ротора через комутатор.

Коли електричний струм подається на обмотки ротора через комутатор, у провідниках виникає магнітне поле, яке взаємодіє з магнітним полем статора. Взаємодія цих полів створює силу, яка змушує ротор обертатися. Комутатор забезпечує зміну напрямку струму в обмотках таким чином, щоб обертальний момент завжди діяв у одному напрямку.

Двигуни постійного струму зазвичай застосовуються там, де потрібна висока точність регулювання швидкості, висока динаміка розгону або обмежений простір для встановлення.

Для більш точного керування двигунами необхідно до схеми додати спеціальні компоненти – драйвери керування ДПС, які дозволять більш точно та гнучко керувати роботою двигунів постійного струму.

Драйвер керування двигуном постійного струму (ДПС) — це електронний пристрій, який забезпечує контроль роботи двигуна шляхом регулювання напруги, струму, напрямку обертання, швидкості та режимів роботи [5]. Драйвери використовуються для узгодження параметрів двигуна із системою керування (наприклад, мікроконтролером), забезпечуючи



- Напруга живлення двигунів: 5–46 В.
- Максимальний струм на канал: 2 А.
- Інтегровані захисти від перегріву.
- Простий інтерфейс керування через цифрові входи мікроконтролера.

Для спрощення схеми мобільної платформи доцільним буде використання спеціальної плати-розширення, яка поєднує всередині одразу 4 драйвери ДПС, що дозволяє реалізувати керування 4 двигунами без необхідності на кожен двигун ставити окремий драйвер. Прикладом такої плати є L293D DC Motor Drive Shield [7], зовнішній вигляд та технічні характеристики якої наведено на рисунку 2.11 та у таблиці 2.3 відповідно.

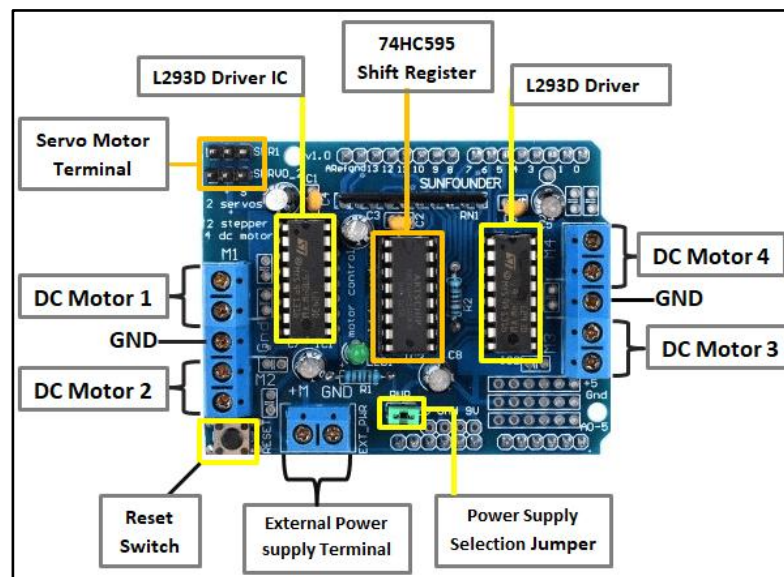


Рисунок 2.11 – L293D DC Motor Drive Shield

Таблиця 2.3 – Технічні характеристики L293D DC Motor Drive Shield

Характеристика	Опис
Кількість керованих двигунів	До 4 двигунів постійного струму або 2 крокових двигунів
Робоча напруга	4.5–36 В для двигунів
Напруга логіки	5 В
Максимальний струм на канал	600 мА (піковий струм — до 1.2 А)

Підтримувані режими роботи	Прямий хід, реверс, гальмування, зупинка
Інтерфейс керування	GPIO пін Arduino
Інтегровані мікросхеми	Дві L293D для керування двигунами
Роз'єм для живлення	Клемна колодка або пін для підключення зовнішнього джерела живлення
Захист	Від перегріву та короткого замикання
Додаткові функції	Можливість підключення сервоприводів (до 2 штук)
Сумісність	Arduino UNO, Mega та інші плати Arduino
Габарити	69 мм × 54 мм
Вага	40 г

## 2.6 Апаратна реалізація підсистеми комп'ютерного зору

Підсистема комп'ютерного зору виконує роль «очей» і дозволяє мобільній платформі фіксувати середовище навколо та реагувати на зміни у цьому середовищі. Технічна реалізація цієї підсистеми потребує сучасних технологій, таких як камера для фіксації та нейронні мережі для аналізу знімків.

Для коректної роботи підсистеми комп'ютерного зору у реальному часі необхідно передбачити достатню обчислювальну потужність та пам'ять, а враховуючи те, що на контролері мобільної платформи вже висить керування 4 двигунами, його обчислювальної потужності та пам'яті може не вистачити, тому при виборі технічного рішення для реалізації підсистеми комп'ютерного зору треба звертати увагу на наявність окремого контролера.

Враховуючи усі вимоги, що висуваються до систем комп'ютерного зору, було обрано найкращий компромісний варіант, який поєднує у собі достатній функціонал, підходящу обчислювальну потужність та форм-фактор, а також відносно невелику вартість – модуль Huskylens Smart Vision Sensor.

HuskyLens — це інтелектуальний модуль камери, розроблений для спрощення інтеграції функцій комп'ютерного зору в різноманітні проекти [8]. Він поєднує в собі апаратне забезпечення та вбудовані алгоритми штучного інтелекту, що дозволяє розпізнавати та відстежувати об'єкти без необхідності складного програмування.

HuskyLens спрощує реалізацію функцій комп'ютерного зору, надаючи користувачам можливість швидко додавати можливості розпізнавання та відстеження об'єктів у свої проекти. Завдяки вбудованим алгоритмам та простому інтерфейсу, він підходить для освітніх цілей, прототипування та розробки робототехнічних систем, що вимагають обробки візуальної інформації в реальному часі.

Зовнішній вигляд та основні технічні характеристики HuskyLens Smart Vision Sensor представлені на рисунку 2.12 та у таблиці 2.4.

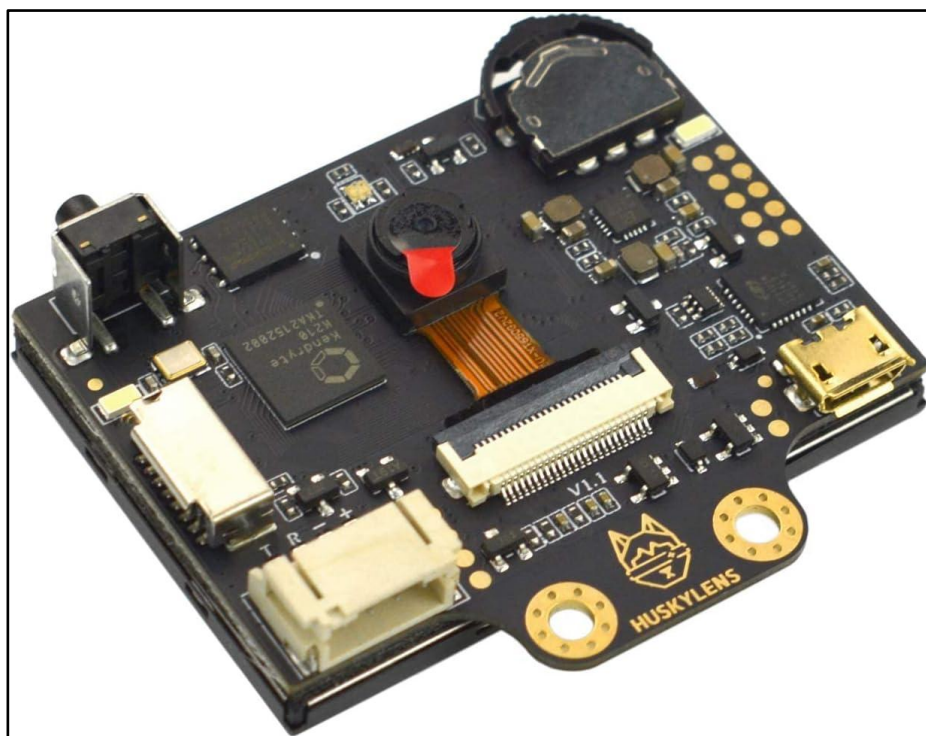


Рисунок 2.12 – HuskyLens Smart Vision Sensor

Таблиця 2.4 – Технічні характеристики Huskylens Smart Vision Sensor

Характеристика	Значення
Процесор	Kendryte K210 (Dual-core 64-bit RISC-V)
Роздільна здатність камери	640 × 480 (VGA)
Дисплей	2-дюймовий TFT LCD, роздільна здатність 320 × 240
Пам'ять	8 МБ SRAM, 16 МБ флеш-пам'яті
Інтерфейси зв'язку	UART, I2C, GPIO
Формат живлення	3.3 В / 5 В (від плати або зовнішнього джерела)
Споживання струму	Близько 250 мА
Підтримувані функції	<ul style="list-style-type: none"> <li>- Розпізнавання обличчя</li> <li>- Розпізнавання об'єктів</li> <li>- Виявлення ліній</li> <li>- Виявлення кольорів</li> <li>- Виявлення етикеток</li> <li>- Навчання об'єктів</li> </ul>
Кількість розпізнаваних об'єктів	До 20 одночасно
Час навчання	Менше 0.5 секунди на об'єкт
Оновлення прошивки	Через USB або UART
Розмір	52 × 44 × 35 мм
Вага	30 г
Сумісність	Arduino, Raspberry Pi, micro:bit, STM32

### 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У попередньому розділі було детально розглянуто усі складові апаратної частини інтелектуальної мобільної платформи та наведено аргументований вибір конкретних технічних рішень для її реалізації. Проте сама по собі мобільна платформа працювати не буде, необхідно реалізувати програмне забезпечення, яке описує повний функціонал системи. У цьому розділі представлено розробку такого програмного забезпечення.

Програмне забезпечення (ПЗ) – сукупність програм системи обробки інформації і програмних документів, необхідних для експлуатації цих програм [4].

#### 3.1 Структура програмного забезпечення та вибір технологій для реалізації

Структуру програмного забезпечення представлено на рисунку 3.1.

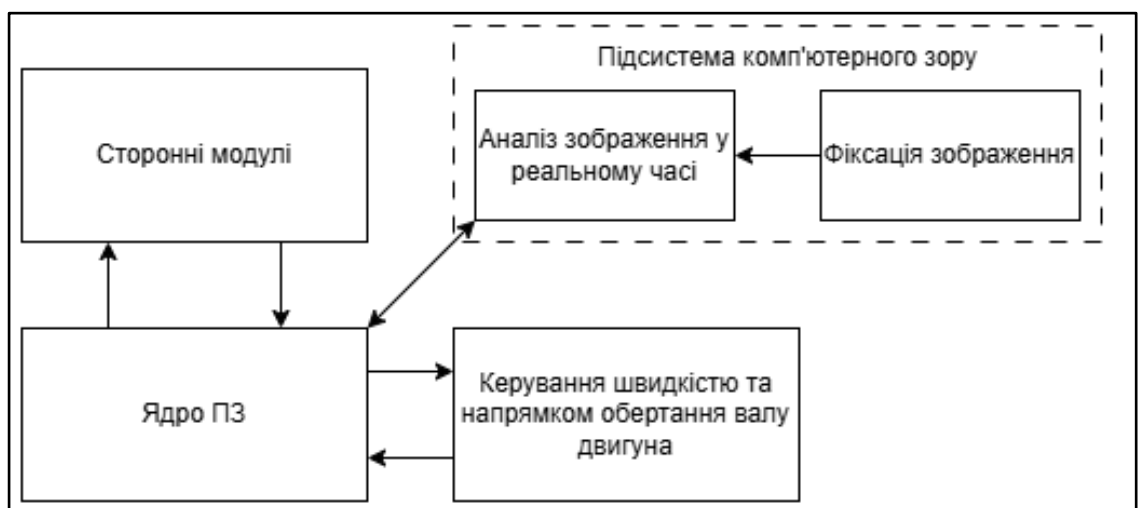


Рисунок 3.1 – Структура ПЗ інтелектуальної мобільної платформи

Як показано на рисунку 3.1, програмне забезпечення інтелектуальної мобільної платформи, так само як і її апаратна складова, складається з кількох підсистем: підсистеми комп'ютерного зору, підсистеми керування двигунами та ядром ПЗ, яке поєднує ці підсистеми та описує загальний функціонал. Для спрощення розробки програмного забезпечення системи доцільним буде використання готових бібліотек, в яких вже реалізований деякий функціонал для роботи з тими чи іншими компонентами.

Оскільки апаратна реалізація контролера мобільної платформи базується на використанні плати Arduino, програмне забезпечення необхідно розробляти саме для цієї платформи, що передбачає застосування мови програмування C++ та інтегрованого середовища розробки Arduino IDE.

C++ є компільованою мовою програмування загального призначення зі статичною (жорсткою) типізацією. Вона підтримує кілька програмних парадигм, включаючи процедурну, об'єктно-орієнтовану та узагальнену (шаблонну) парадигми. C++ надає широкий набір стандартних бібліотек, які охоплюють часто використовувані структури даних, алгоритми, інструменти для введення та виведення даних, обробку регулярних виразів, а також засоби для виконання багатопоточних операцій. Ця мова поєднує риси як високорівневих, так і низькорівневих мов програмування [9].

Інтегроване середовище розробки (IDE) – це набір інструментів, який застосовують розробники для створення програмного забезпечення [10-12].

Для програмування мікроконтролерів, вбудованих у друковану плату Arduino, використовується офіційне інтегроване середовище розробки – Arduino IDE[10], зовнішній вигляд якого наведено на рисунку 3.2.

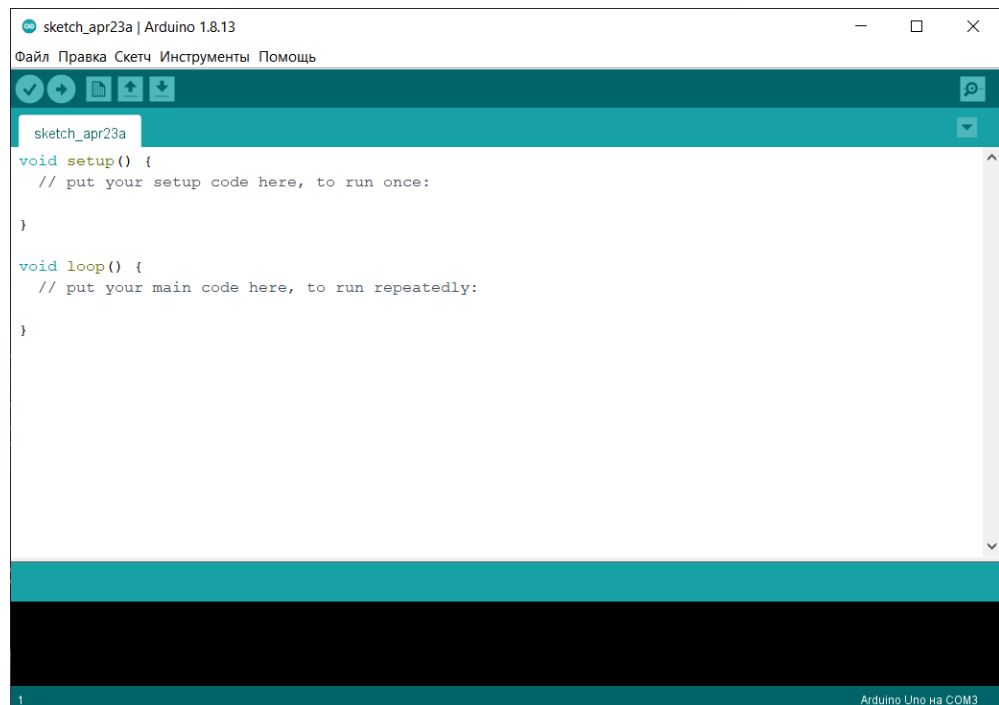


Рисунок 3.2 – ArduinoIDE

### 3.2 Реалізація програмного забезпечення для контролера інтелектуальної мобільної платформи

Контролер є найважливішим компонентом, адже він виконує роль центру керування усією системою, саме тому програмне забезпечення контролера повинно описувати повний функціонал і бути стійким до помилок. Нижче наведено декілька лістингів коду програмного забезпечення для контролеру плати Arduino. Повний код програмного забезпечення наведено у додатку В.

Як було описано у попередньому підрозділі при розробці ПЗ доцільно використовувати готові бібліотеки, враховуючи те, що функціонально контролер мобільної платформи повинен керувати двигунами на основі даних, які надходять з підсистеми комп'ютерного зору, необхідно додати до проекту декілька бібліотек, які вже мають подібний функціонал, а саме:

- HUSKYLENS.h –бібліотека для взаємодії з камерою HUSKYLENS через інтерфейс I2C.
- SoftwareSerial.h – бібліотека для створення додаткових послідовних портів для зв'язку.
- AFMotor.h –бібліотека для управління двигунами через драйвер двигунів Motor Shield.

### Лістинг 3.1 – Декларативна частина ПЗ

```

AF_DCMotor rightBack(1);
AF_DCMotor rightFront(2);
AF_DCMotor leftFront(3);
AF_DCMotor leftBack(4);
byte maxMotorSpeed = 50;
int motorOffset = 25;
int objectWidth = 50;
int turnGain = 12;
int offCentre = 20;
int distGain = 6;
int leftLimit = 160-offCentre;
int rightLimit = 160+offCentre;
int leftSp = 0;
int rightSp = 0;
bool isTurning = false;
bool isTurningLeft = true;
HUSKYLENS huskylens;
int ID1 = 1;

```

У лістингу 3.1 наведено код, у якому спочатку створюються 4 об'єкти двигунів, по одному на кожне колесо (номери в дужках відповідають портам на драйвері ДПС), а також усі необхідні змінні, які задають певні параметри

системи. Об'єкт *huskylens* створюється для взаємодії з камерою, а *IDI* використовується для ідентифікації об'єкта, який відстежується.

### Лістинг 3.2 – Код функції `setup()`

```
void setup()
{
  Serial.begin(115200);
  Wire.begin();
  while (!huskylens.begin(Wire))
  {
    Serial.println(F("Begin failed!"));
    delay(100);
  }
  huskylens.writeAlgorithm(ALGORITHM_OBJECT_TRACKING);
  rightBack.setSpeed(0);
  rightFront.setSpeed(0);
  leftFront.setSpeed(0);
  leftBack.setSpeed(0);
  moveForward();
}
```

У лістингу 3.2 наведено код функції `setup()`, яка виконується один раз при запуску системи. У цій функції спочатку іде налагодження з'єднання з камерою, у разі помилки відображаються відповідні повідомлення; далі камера переходить у режим відстеження об'єктів; потім усі двигуни налаштовуються на нульову швидкість і напрямом «вперед».

### Лістинг 3.3 – Код функції для руху

```
void moveForward()
```

```

{
    rightBack.run(FORWARD);
    rightFront.run(FORWARD);
    leftFront.run(FORWARD);
    leftBack.run(FORWARD);
}

```

#### Лістинг 3.4 – Код функції для зупинки

```

void stopMove()
{
    rightBack.run(RELEASE);
    rightFront.run(RELEASE);
    leftFront.run(RELEASE);
    leftBack.run(RELEASE);
}

```

#### Лістинг 3.5 – Код функції для задання швидкості моторів

```

void setMotorSpeed (int leftTempSp, int rightTempSp)
{
    rightBack.setSpeed(rightTempSp);
    rightFront.setSpeed(rightTempSp);
    leftFront.setSpeed(leftTempSp+motorOffset);
    leftBack.setSpeed(leftTempSp+motorOffset);
}

```

Код, який наведено у лістингах 3.3 – 3.5 описує функції керування рухом мобільної платформи. Функція *moveForward* передає керуючі сигнали через драйвер на двигуни тим самим забезпечує рух платформи вперед, а функція *stopMove* навпаки передає сигнали про зупинку і усі двигни зупиняються. Функція *setMotorSpeed* встановлює швидкість кожному двигуну окремо, що дозволяє більш гнучко керувати рухом мобільної платформи та робити повороти чи розганятися/гальмувати.

Блок-схема алгоритму роботи програмного забезпечення наведена на рисунках 3.3 – 3.5. Вся програма являє собою нескінченний цикл, який реалізовано у функції *loop()*.

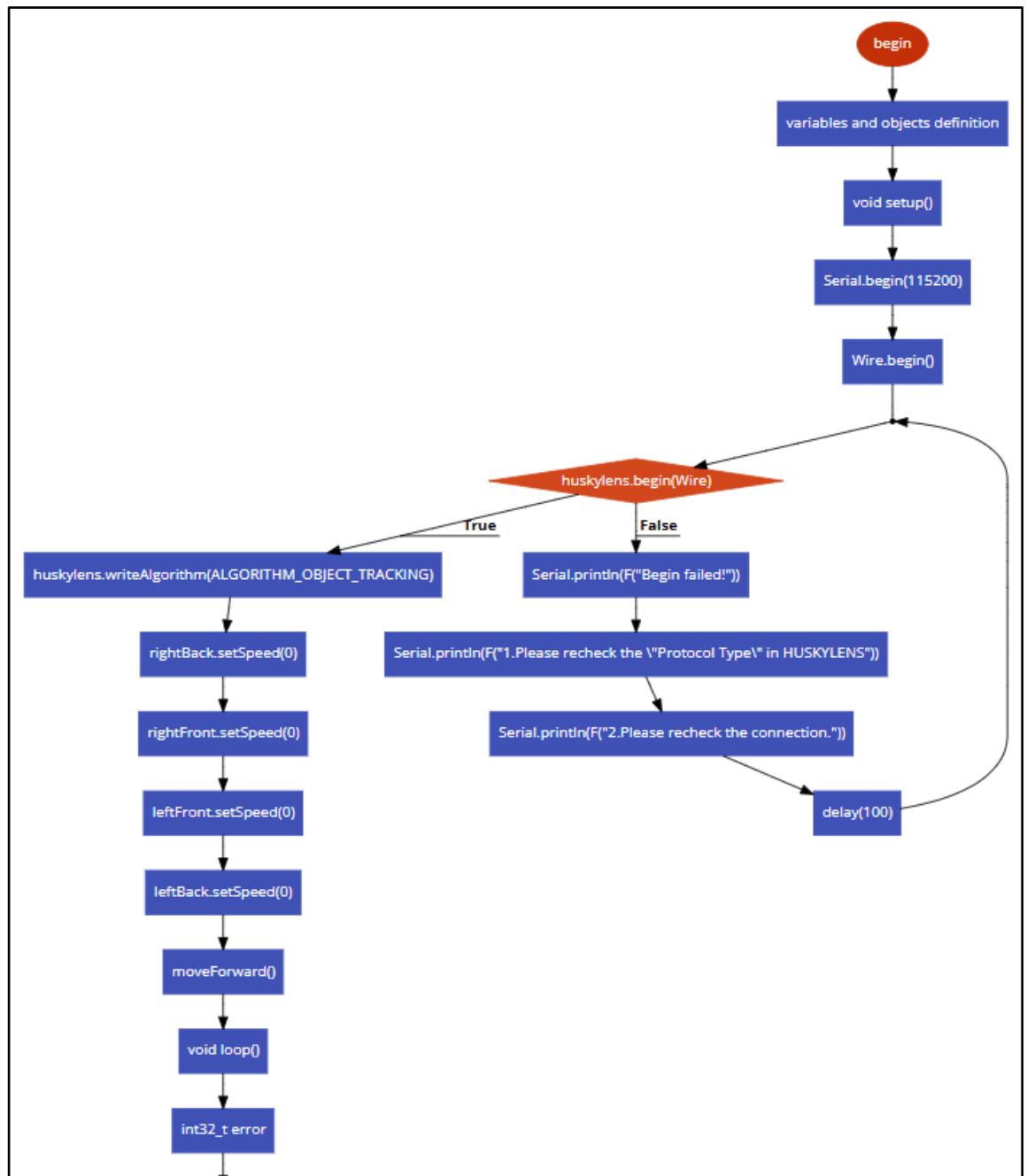


Рисунок 3.3 – Узагальнений алгоритм ПЗ (початок)



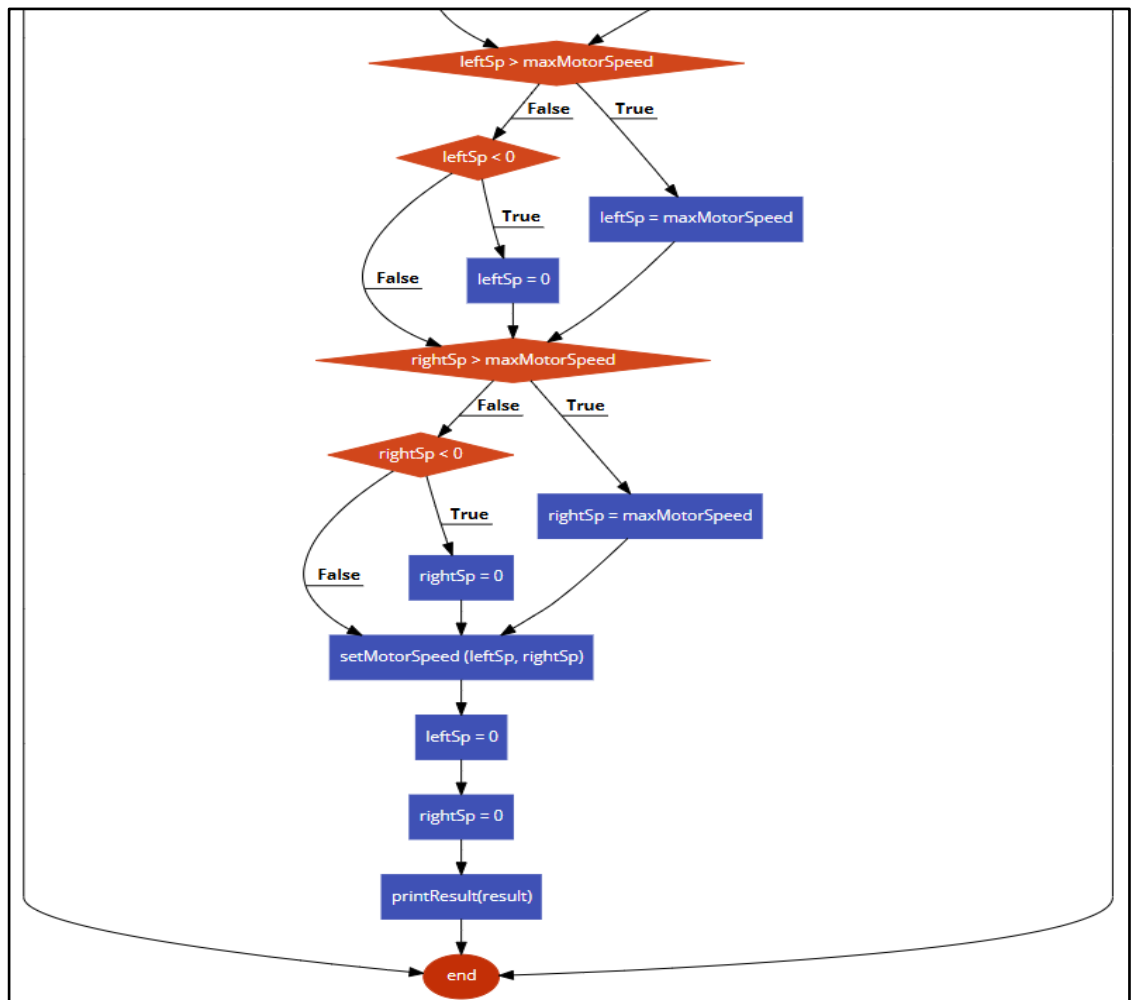


Рисунок 3.5 – Узагальнений алгоритм ПЗ (кінець)

Як видно з рисунків 3.3 – 3.5, принцип роботи програмного забезпечення базується на відстеженні об'єкта камерою Huskylens (програмне забезпечення зашите в саму камеру) та передачі керуючих сигналів на плату-розширення з драйверами двигунів про напрям та швидкість руху мобільної платформи.

У програмному забезпеченні передбачено виникнення помилок і реалізовано спеціальних функціонал реакції системи на ці помилки. У разі якщо система не може ідентифікувати об'єкт, мобільна платформа не рухається. Програма працює у нескінченному циклі доки є електроживлення у системі, або користувач не вимкне її вручну.

## 4 ДОСЛІДНА ЕКСПЛУАТАЦІЯ МАКЕТНОГО ЗРАЗКА

У попередніх розділах здійснено детальний аналіз предметної області та теоретичних аспектів, необхідних для розробки. Проведено глибоке дослідження методів і технологій, що використовуються у галузях створення мобільних платформ та систем комп'ютерного зору. Було реалізовано як апаратну частину інтелектуальної мобільної платформи на основі плати Arduino та модуля комп'ютерного зору Huskylens, так і програмну складову системи. Для перевірки працездатності та оцінки ефективності створеної системи передбачається проведення тестової експлуатації прототипу. Цей етап включає складання апаратної частини та запуск програмного забезпечення в умовах, максимально наближених до реальних умов експлуатації системи користувачем.

### 4.1 Опис макетного зразка та методики випробувань

Для проведення тестування роботи мобільної платформи необхідно реалізувати апаратну та програмну складову системи так, як це наведено у попередніх розділах.

Враховуючи, що основний акцент у кваліфікаційній роботі зроблено на розробку програмного забезпечення, при реалізації апаратної складової було використано готові технічні рішення. Усі компоненти системи підключаються головного контролера – плати Arduino, за допомогою спеціальних дротів, а за для спрощення процесу збирання та запобігання плутанини з проводами, доцільно використовувати макетну плату (breadboard). Зовнішній вигляд зібраної апаратної складової інтелектуальної мобільної платформи наведено на рисунку 4.1.

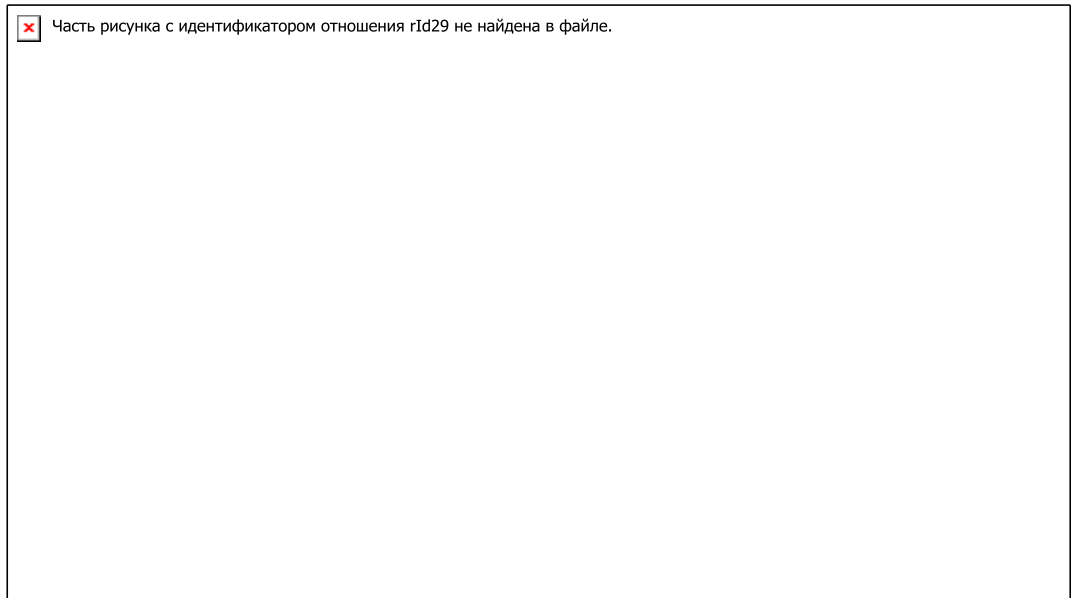


Рисунок 4.1 – Зовнішній вигляд інтелектуальної мобільної платформи

Після того, як апаратна частина системи завершена, потрібно завантажити до мікроконтролера плати Arduino програмне забезпечення, яке забезпечує роботу системи. Далі необхідно провести тестування розробленої системи, яке складається з наступних етапів:

1. Запуск системи;
2. Перевірка працездатності усіх компонентів системи;
3. Перевірка коректності роботи підсистеми комп'ютерного зору;
4. Навчання моделі нейронної мережі на конкретному об'єкті;
5. Перевірка правильності розпізнавання об'єктів нейронною мережею;
6. Комплексна перевірка роботи системи – слідування мобільної платформи за об'єктом;
7. Перевірка часу відгуку системи та таймаутів;
8. Перевірка відпрацювання нестандартних ситуацій, за яких виникають помилки у системі.

Результати тестування макетного зразка проєктованої системи представлені у наступному підрозділі.

## 4.2 Результати моделювання роботи тестового зразка

Спочатку необхідно перевірити запуск системи і працездатність усіх компонентів. У разі, якщо якийсь компонент мобільної платформи не працює чи працює некоректно, система виведе повідомлення про це у послідовний порт і треба буде перевірити в чому проблема та виправити її. Якщо жодних проблем із запуском та працездатністю компонентів немає, наступним етапом необхідно навчити модель нейронної мережі, яка вже «зашита» у підсистему комп'ютерного зору (модуль Huskylens) розпізнаванню об'єкта. В якості об'єкта може бути будь-що, але краще брати щось з чіткими краями та монохромне, щоб нейронна мережа давала менше неточних результатів. Процес навчання нейронної мережі на конкретному об'єкті представлено на рисунку 4.2.



Рисунок 4.2 – Навчання моделі нейронної мережі на конкретному об'єкті

Далі необхідно перевірити чи коректно підсистема комп'ютерного зору ідентифікує об'єкт після навчання нейронної мережі. Для цього необхідно ще раз поставити об'єкт перед камерою та подивитися, чи зреагує система на нього (рис. 4.3). Якщо система не зреагувала на об'єкт, то необхідно зробити перенавчання моделі нейронної мережі, або поміняти об'єкт.

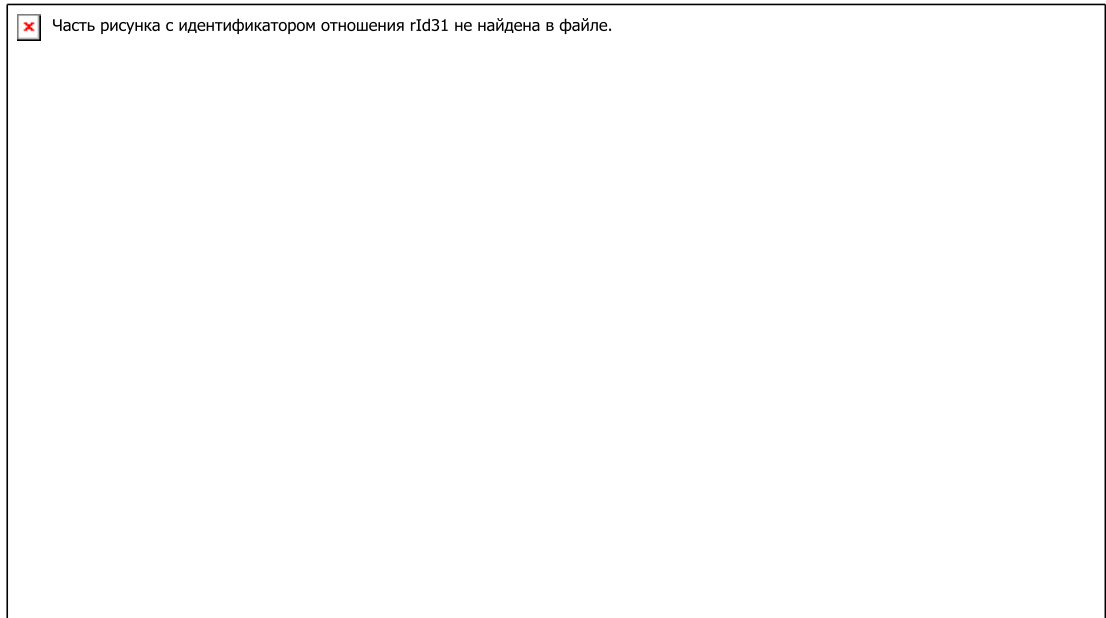


Рисунок 4.3 – Реакція системи на фіксацію камерою об'єкта

Наступним етапом тестування є остаточна перевірка роботи усього програмного забезпечення, а саме слідування мобільної платформи за об'єктом, часових затримок у системі та відпрацювання виключень з функціоналу. Результат наведено на рисунку 4.4.

Тести показали стабільну працездатність системи відповідно до функціоналу програмного забезпечення, часові затримки між моментом фіксації об'єкта та реакцією мобільної платформи в межах норми, враховуючи обчислювальні потужності використаних компонентів. Обробка виключень з функціоналу системи також працює коректно, що дозволяє системі працювати без збоїв.



Рисунок 4.4 – Слідування мабольної платформи за об'єктом

На основі проведених тестувань встановлено, що розроблена система відповідає всім вимогам, передбаченим програмним забезпеченням. Вона успішно пройшла всі етапи перевірки, продемонструвавши стабільну працездатність і високу ефективність. Модель нейронної мережі демонструє швидке навчання та забезпечує коректні результати. Функціональні можливості системи реалізуються без збоїв, а час відгуку на дії користувача відповідає заданим параметрам. Випадки виникнення помилок або несправностей обробляються відповідно до закладених у програмному забезпеченні алгоритмів.

Варто зазначити, що реальна продуктивність системи може дещо відрізнятись в умовах практичної експлуатації, оскільки тестування проводилося на макетному стенді. У процесі перевірки не було виявлено критичних збоїв у роботі системи.

## ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було розроблено інтелектуальну мобільну платформу з функцією відстеження об'єктів, апаратна складова якої базується на мікроконтролері Arduino UNO, модулі комп'ютерного зору Huskylens, драйвері двигунів L293D Motor Driver Shield та чотирьох двигунах постійного струму. Програмне забезпечення системи створено в середовищі розробки Arduino IDE з використанням мови програмування C++.

У першому розділі роботи було проведено дослідження предметної області, зокрема принципів функціонування систем відстеження об'єктів та особливостей мобільних платформ. Розглянуто сучасні технології та апаратні засоби, що застосовуються для таких систем. На основі аналізу зроблено висновки, що створення інтелектуальної мобільної платформи на базі доступних компонентів є доцільним і ефективним для реалізації навчальних і дослідницьких завдань.

У другому розділі детально описано розробку апаратної частини системи. Описано процес інтеграції мікроконтролера Arduino UNO, модуля Huskylens, драйвера двигунів L293D та двигунів постійного струму. Ці компоненти забезпечують рух платформи та можливість відстеження об'єктів у реальному часі.

Третій розділ присвячено розробці програмного забезпечення. Програмне забезпечення написане на мові програмування C++ із використанням стандартних бібліотек Arduino IDE. У програмі реалізовано алгоритми управління рухом платформи та взаємодії з модулем Huskylens для відстеження об'єктів. Забезпечено обробку даних від датчиків та адаптацію руху платформи залежно від положення об'єкта.

У заключному розділі наведено результати тестування системи. У ході дослідної експлуатації макетного зразка підтверджено працездатність та ефективність розробленої системи. Платформа коректно відстежує об'єкти, адаптуючи свій рух до їхнього положення. Система продемонструвала стабільну роботу без збоїв, а час реакції на зміни положення об'єкта відповідає очікуванням.

Таким чином, розроблена інтелектуальна мобільна платформа з функцією відстеження об'єктів на базі Arduino UNO та модуля Huskylens успішно виконує поставлені завдання. Вона може бути використана для навчальних і дослідницьких цілей, а також як основа для подальших розробок у сфері робототехніки та комп'ютерного зору.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Мобільна платформа [Електронний ресурс] // Вікіпедія: вільна енциклопедія. – Електронні данні. – Режим доступу: [https://uk.wikipedia.org/wiki/Мобільна\\_платформа](https://uk.wikipedia.org/wiki/Мобільна_платформа)
2. Комп'ютерний зір [Електронний ресурс] // Вікіпедія: вільна енциклопедія. – Електронні данні. – Режим доступу: [https://uk.wikipedia.org/wiki/Комп'ютерний\\_зір](https://uk.wikipedia.org/wiki/Комп'ютерний_зір)
3. Нейронні мережі [Електронний ресурс] // Вікіпедія: вільна енциклопедія. – Електронні данні. – Режим доступу: [https://uk.wikipedia.org/wiki/Нейронні\\_мережі](https://uk.wikipedia.org/wiki/Нейронні_мережі)
4. Півняк Г. Г., Бусигін Б. С., Дівізіюк М. М. Тлумачний словник з інформатики. — Д.: Національний гірничий університет, 2010. — 392 с. — ISBN 978-966-350-306-0.
5. Великий тлумачний словник сучасної української мови / уклад. і гол. ред. В. Т. Бусел. — 8-ме вид., випр. і доп. — К.: Перун, 2019. — ISBN 978-966-569-615-5.
6. Arduino [Електронний ресурс] // Arduino: офіційний сайт. – Електронні данні. – Режим доступу: <https://www.arduino.cc/>
7. *L293x Quadruple Half-H Drivers datasheet (Rev. D)*. [Електронний ресурс] // Texas Instruments. – Електронні данні. – Режим доступу: <https://www.ti.com/lit/ds/symlink/l293.pdf>
8. *HUSKYLENS AI Machine Vision Sensor*. [Електронний ресурс] // DFRobot. – Електронні данні. – Режим доступу: [https://wiki.dfrobot.com/HUSKYLENS\\_V1.0\\_SKU\\_SEN0305\\_SEN0336](https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336)
9. C++ Programming: Principles and Practice Using C++ / Bjarne Stroustrup. — Addison-Wesley Professional, 2014. — 1312 p. — ISBN 978-0321992789.

10. Elliot Williams. *Make: AVR Programming: Learning to Write Software for Hardware*. — Maker Media, Inc., 2014. — 472 p. — ISBN 978-1449355784.
11. Julien Bayle. *C Programming for Arduino*. — Packt Publishing, 2013. — 512 p. — ISBN 978-1849517584.
12. Simon Monk. *Programming Arduino: Getting Started with Sketches*. — McGraw-Hill Education, 2016. — 192 p. — ISBN 978-1259641633.
13. Gonzalez, R. C., & Woods, R. E. (2017). *Digital Image Processing (4th ed.)*. Upper Saddle River, NJ: Pearson. ISBN 978-0133356724.
14. Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural Networks*, 61, 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>
15. Zhang, Z., Cui, P., & Zhu, W. (2021). *Deep Learning on Graphs*. Cambridge, UK: Cambridge University Press. ISBN 978-1108494475.
16. Russakovsky, O., Deng, J., Su, H., et al. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), 211–252. <https://doi.org/10.1007/s11263-015-0816-y>
17. Ballard, D. H., & Brown, C. M. (1982). *Computer Vision*. Englewood Cliffs, NJ: Prentice Hall. ISBN 978-0131653160.