

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та охоронних систем

(повне найменування кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»
РОБОТИЗОВАНА ПЛАТФОРМА З ІНТЕЛЕКТУАЛЬНИМ
РОЗПІЗНАВАННЯМ ОБ'ЄКТІВ НА ОСНОВІ ТЕХНОЛОГІЙ
КОМП'ЮТЕРНОГО ЗОРУ ТА МАШИННОГО НАВЧАННЯ
ROBOTIC PLATFORM WITH INTELLIGENT OBJECT
RECOGNITION BASED ON COMPUTER VISION AND MACHINE
LEARNING TECHNOLOGIES**

спеціальність **123 Комп'ютерна інженерія**

(шифр і назва спеціальності)

освітня програма **Комп'ютерна інженерія**

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи **КІ-42**

Махно Олександр Сергійович

(підпис)

Керівник:

к.т.н., доцент

Гринюк Сергій Васильович

(підпис)

Кваліфікаційну роботу
допущено до захисту

« » червня 2026 р.

Гарант освітньої програми:

к.т.н., доцент

Лавренчук Світлана Василівна

(підпис)

Луцьк – 2026 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та безпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц. Т. Терлецький

« 23 » 12 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Махну Олександр Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Роботизована платформа з інтелектуальним розпізнаванням об'єктів на основі технологій комп'ютерного зору та машинного навчання

Керівник роботи к.т.н., доцент Гринюк Сергій Васильович

затверджені наказом закладу вищої освіти від «20» грудня 2025 року № 536/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 28.05.2026 р.

3. Вихідні дані до роботи Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області, різні інтернет-ресурси технічного спрямування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

Вступ

Теоретичні основи побудови роботизованих платформ з інтелектуальним розпізнаванням об'єктів

Проектування роботизованої платформи з комп'ютерним зором та машинним навчанням

Практична реалізація роботизованої платформи

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Завершений вигляд мобільного робота після встановлення камери, сенсорів і плати керування

Послідовність налаштування модулів мобільного робота

Принципова схема підключення компонентів роботизованої платформи

Структурна схема підключення компонентів роботизованої платформи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Теоретичні основи побудови роботизованих платформ з інтелектуальним розпізнаванням об'єктів</i>	<i>Гринюк С.В., доцент</i>		
<i>Проектування роботизованої платформи з комп'ютерним зором та машинним навчанням</i>	<i>Гринюк С.В., доцент</i>		
<i>Практична реалізація роботизованої платформи</i>	<i>Гринюк С.В., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>	<i>%</i>		
<i>Академічна доброчесність</i>	<i>Міскевич О.І., ст. викладач</i>		

7. Дата видачі завдання 23.12.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Огляд літератури із досліджуваної проблеми, аналіз предметної області та наявних рішень</i>	до 10.02.2026 р.	
2.	<i>Теоретичні основи побудови роботизованих платформ з інтелектуальним розпізнаванням об'єктів</i>	до 02.03.2026 р.	
3.	<i>Проектування роботизованої платформи з комп'ютерним зором та машинним навчанням</i>	до 02.04.2026 р.	
4.	<i>Практична реалізація роботизованої платформи</i>	до 10.04.2026 р.	
5.	<i>Представлення остаточного варіанту кваліфікаційної роботи керівникові</i>	до 10.05.2026 р.	
6.	<i>Нормоконтроль</i>	до 23.05.2026 р.	
7.	<i>Інструментальна перевірка на академічний плагіат</i>	до 25.05.2026 р.	
8.	<i>Здача кваліфікаційної роботи та всіх супровідних документів на кафедру</i>	до 28.05.2026 р.	

Здобувач вищої освіти

(підпис)

Олександр МАХНО

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Сергій ГРИНЮК

(прізвище, ініціали)

АНОТАЦІЯ

Махно О. С. Роботизована платформа з інтелектуальним розпізнаванням об'єктів на основі технологій комп'ютерного зору та машинного навчання. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2026.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, переліку використаних джерел, додатків.

У першому розділі розглянуто теоретичні основи побудови роботизованих платформ з інтелектуальним розпізнаванням об'єктів. Проаналізовано роль мобільних роботизованих систем у сучасній комп'ютерній інженерії, визначено їхню структуру, функціональні можливості та сфери застосування.

У другому розділі виконано проектування роботизованої платформи з комп'ютерним зором та машинним навчанням. Сформовано основні вимоги до апаратної та програмної частини системи, обґрунтовано вибір апаратної платформи для реалізації мобільного робота, а також визначено критерії її використання.

У третьому розділі описано практичну реалізацію роботизованої платформи на базі ELEGOO Smart Robot Car Kit V4.0. Подано загальну характеристику набору, описано етапи складання апаратної частини мобільного робота, зокрема монтаж шасі, двигунів, коліс, плати керування, сенсорних модулів, камери, сервоприводу та системи живлення.

Ключові слова: роботизована платформа, мобільний робот, Arduino, комп'ютерний зір, машинне навчання, розпізнавання об'єктів, мікроконтролер, датчики, керування рухом.

ANNOTATION

Makhno O. Robotic platform with intelligent object recognition based on computer vision and machine learning technologies. Manuscript.

Bachelor's qualification work EP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2026.

The qualification work consists of an introduction, three sections, conclusions, a list of sources used, and appendices.

The first section considers the theoretical foundations of building robotic platforms with intelligent object recognition. The role of mobile robotic systems in modern computer engineering is analyzed, their structure, functionality, and areas of application are determined.

In the second section, the design of a robotic platform with computer vision and machine learning is performed. The main requirements for the hardware and software parts of the system are formed, the choice of a hardware platform for implementing a mobile robot is justified, and the criteria for its use are determined.

The third section describes the practical implementation of a robotic platform based on the ELEGOO Smart Robot Car Kit V4.0. A general description of the kit is given, and the stages of assembling the hardware of the mobile robot are described, including the installation of the chassis, motors, wheels, control board, sensor modules, camera, servo drive, and power supply system.

Keywords: robotic platform, mobile robot, Arduino, computer vision, machine learning, object recognition, microcontroller, sensors, motion control.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ РОБОТИЗОВАНИХ ПЛАТФОРМ З ІНТЕЛЕКТУАЛЬНИМ РОЗПІЗНАВАННЯМ ОБ’ЄКТІВ	11
1.1 Основні принципи комп’ютерного зору в задачах розпізнавання об’єктів	11
1.2 Методи машинного навчання для класифікації та виявлення об’єктів на зображеннях	15
1.3 Особливості використання мікроконтролерних платформ у мобільній робототехніці.....	19
1.4 Аналіз існуючих рішень роботизованих платформ з елементами інтелектуального керування	24
РОЗДІЛ 2 ПРОЄКТУВАННЯ РОБОТИЗОВАНОЇ ПЛАТФОРМИ З КОМП’ЮТЕРНИМ ЗОРОМ ТА МАШИННИМ НАВЧАННЯМ.....	29
2.1 Обґрунтування вибору апаратної платформи для реалізації мобільного робота.....	29
2.2 Розробка структурної та принципової схем роботизованої платформи.....	35
2.3 Вибір програмних засобів для реалізації комп’ютерного зору та машинного навчання.....	41
2.4 Розробка алгоритму взаємодії модуля розпізнавання об’єктів із системою керування рухом.....	46
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ РОБОТИЗОВАНОЇ ПЛАТФОРМИ	51
3.1 Складання апаратної частини мобільного робота на базі Arduino	51
3.2 Налаштування модулів руху, датчиків перешкод, камери та бездротового керування.....	58
3.3 Реалізація програмного керування рухом роботизованої платформи в середовищі Arduino IDE	63
3.4 Реалізація модуля комп’ютерного зору для розпізнавання об’єктів	68
3.5 Інтеграція результатів розпізнавання об’єктів із командами керування рухом робота	71
ВИСНОВКИ	77

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	81
-----------------------------------	----

ВСТУП

Сучасний розвиток робототехніки, комп'ютерного зору та машинного навчання створює передумови для розроблення мобільних роботизованих платформ, здатних не лише виконувати запрограмовані команди руху, а й аналізувати навколишнє середовище, розпізнавати об'єкти та приймати рішення на основі отриманих даних. Такі системи є важливим напрямом комп'ютерної інженерії, оскільки поєднують апаратні засоби керування, сенсорні модулі, програмні алгоритми, засоби обробки зображень та елементи інтелектуального аналізу інформації.

Актуальність теми зумовлена зростанням потреби у створенні доступних роботизованих систем, які можуть використовуватися для навчання, досліджень, прототипування та виконання прикладних завдань. Традиційні мобільні роботи на основі мікроконтролерів здатні реалізовувати базові функції, зокрема рух заданою траєкторією, об'їзд перешкод, слідування за лінією або дистанційне керування. Однак для підвищення рівня автономності таких платформ необхідно доповнювати їх засобами комп'ютерного зору та машинного навчання, які забезпечують можливість аналізу відеопотоку, виявлення об'єктів і формування відповідної реакції системи.

Особливого значення набуває поєднання мікроконтролерної платформи з окремим програмним або апаратним модулем розпізнавання об'єктів. У такій архітектурі мікроконтролер виконує функції керування двигунами, датчиками та виконавчими механізмами, а модуль комп'ютерного зору відповідає за обробку зображень, класифікацію об'єктів і передавання результатів аналізу до системи керування. Такий підхід дає змогу раціонально розподілити функції між апаратними та програмними компонентами, враховуючи обмежені обчислювальні ресурси мікроконтролера та потребу в більш продуктивному середовищі для виконання алгоритмів машинного навчання.

Метою кваліфікаційної роботи є розробка роботизованої платформи з інтелектуальним розпізнаванням об'єктів на основі технологій комп'ютерного зору та машинного навчання.

Об'єктом дослідження є процес функціонування мобільної роботизованої платформи з елементами інтелектуального керування.

Предметом дослідження є апаратні засоби, програмні алгоритми та методи комп'ютерного зору і машинного навчання, що використовуються для розпізнавання об'єктів і керування рухом роботизованої платформи.

Для досягнення поставленої мети в роботі передбачено виконати такі завдання:

- проаналізувати теоретичні основи побудови мобільних роботизованих платформ;
- дослідити принципи комп'ютерного зору в задачах розпізнавання об'єктів;
- розглянути методи машинного навчання для класифікації та виявлення об'єктів на зображеннях;
- визначити особливості використання мікроконтролерних платформ у мобільній робототехніці;
- проаналізувати існуючі роботизовані платформи з елементами інтелектуального керування;
- сформулювати вимоги до апаратної та програмної частини роботизованої платформи;
- розробити структурну схему взаємодії основних компонентів системи;
- реалізувати програмне керування рухом мобільного робота;
- налаштувати модуль комп'ютерного зору для розпізнавання об'єктів;
- інтегрувати результати розпізнавання об'єктів із логікою керування роботизованою платформою;
- провести тестування розробленої системи та оцінити її працездатність.

Методи дослідження включають аналіз науково-технічних джерел, порівняльний аналіз апаратних і програмних рішень, структурне моделювання,

алгоритмізацію процесів керування, методи комп'ютерного зору, методи машинного навчання, а також експериментальне тестування роботизованої платформи.

Практичне значення роботи полягає у створенні прототипу мобільної роботизованої платформи, здатної отримувати візуальні дані з камери, розпізнавати об'єкти та виконувати дії відповідно до результатів аналізу. Результати роботи можуть бути використані як навчальний приклад для вивчення мобільної робототехніки, мікроконтролерного керування, комп'ютерного зору, машинного навчання та принципів інтеграції апаратних і програмних компонентів у кіберфізичних системах.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ РОБОТИЗОВАНИХ ПЛАТФОРМ З ІНТЕЛЕКТУАЛЬНИМ РОЗПІЗНАВАННЯМ ОБ'ЄКТІВ

1.1 Основні принципи комп'ютерного зору в задачах розпізнавання об'єктів

Комп'ютерний зір є одним із провідних напрямів розвитку штучного інтелекту, що забезпечує автоматизоване отримання, оброблення, аналіз та інтерпретацію зображень або відеопотоків. Його основне завдання полягає в тому, щоб перетворити цифрове зображення, яке складається з пікселів, у змістову інформацію про об'єкти, їхні властивості, просторове розміщення та взаємозв'язки у сцені [1]. У роботизованих системах комп'ютерний зір виконує особливо важливу роль, оскільки дає змогу мобільній платформі не лише рухатися за наперед заданою траєкторією, а й аналізувати навколишнє середовище, виявляти об'єкти, визначати перешкоди та формувати реакцію на зміну умов руху.

У задачах розпізнавання об'єктів доцільно розрізняти кілька основних процесів: класифікацію зображення, локалізацію об'єкта, детектування об'єктів, сегментацію та відстеження. Класифікація передбачає визначення того, який саме об'єкт або клас об'єктів присутній на зображенні. Локалізація доповнює класифікацію визначенням місця розташування об'єкта в кадрі. Детектування об'єктів є складнішим завданням, оскільки система повинна одночасно знайти один або кілька об'єктів, визначити їхні координати та встановити належність до відповідного класу [2]. Для мобільної роботизованої платформи найбільш практичне значення має саме детектування, оскільки воно дозволяє визначити не лише тип об'єкта, а й його положення відносно камери.

Загальний процес комп'ютерного зору в задачах розпізнавання об'єктів складається з кількох послідовних етапів. На першому етапі здійснюється отримання зображення або відеопотоку з камери. Якість вхідних даних залежить від роздільної здатності камери, частоти кадрів, кута огляду, освітлення,

фокусування та стабільності передавання даних. Для мобільного робота ці чинники є особливо важливими, оскільки під час руху можуть виникати розмиття зображення, тіні, відблиски, зміна фону або часткове перекриття об'єктів. Тому перед безпосереднім розпізнаванням зображення зазвичай проходить етап попередньої обробки [3].

Попередня обробка зображення полягає у приведенні кадру до вигляду, зручного для подальшого аналізу. До основних операцій цього етапу належать зміна розміру зображення, перетворення колірного простору, нормалізація яскравості та контрастності, фільтрація шумів, виділення контурів і формування області інтересу. Наприклад, для зменшення обчислювального навантаження зображення може бути перетворене у відтінки сірого, а для аналізу кольорових об'єктів - у колірний простір HSV. Такі операції широко реалізуються за допомогою бібліотеки OpenCV, яка містить інструменти для базової та розширеної обробки зображень, а також засоби для роботи з алгоритмами виявлення об'єктів [2, 3].

Наступним етапом є виділення ознак об'єкта. Ознаками можуть бути контури, краї, кути, форма, колір, текстура або характерні точки зображення. У класичних методах комп'ютерного зору такі ознаки задавалися вручну за допомогою спеціальних алгоритмів. Цей підхід є ефективним у порівняно простих умовах, коли об'єкт має сталі візуальні характеристики, чітко відокремлений від фону та не змінює своєї форми. Проте в реальних умовах роботи мобільної платформи зображення часто змінюється через освітлення, рух камери, складний фон або часткове перекриття об'єктів, що знижує ефективність класичних методів [1].

Сучасні системи розпізнавання об'єктів переважно базуються на методах машинного навчання та глибоких нейронних мережах. На відміну від класичних алгоритмів, де ознаки задаються розробником, нейронна мережа здатна самостійно навчатися виділяти важливі візуальні характеристики на основі навчальних прикладів. Особливо поширеними є згорткові нейронні мережі, які ефективно працюють із зображеннями, оскільки враховують просторову

структуру пікселів та локальні залежності між ними [4]. Завдяки цьому модель може виявляти не лише прості ознаки, наприклад краї або контури, а й складніші візуальні шаблони, характерні для певних класів об'єктів.

У задачах детектування об'єктів використовуються моделі, які одночасно виконують класифікацію та локалізацію. До таких підходів належать YOLO, SSD, Faster R-CNN, а також сучасні детектори на основі Transformer-архітектур. Результатом роботи таких моделей є набір прямокутних областей, у яких виявлено об'єкти, назви відповідних класів та значення впевненості моделі. Для роботизованої платформи важливе значення має не лише точність розпізнавання, а й швидкість обробки кадрів, оскільки затримка між отриманням зображення та реакцією робота може впливати на якість руху та безпечність взаємодії з перешкодами [5].

Принципову послідовність роботи комп'ютерного зору в задачі розпізнавання об'єктів доцільно подати на рисунку 1.1.



Рисунок 1.1 – Узагальнена послідовність розпізнавання об'єктів засобами комп'ютерного зору [5]

У контексті мобільної роботизованої платформи результати комп'ютерного зору повинні бути пов'язані з логікою керування рухом. Наприклад, якщо система розпізнає перешкоду перед роботом, платформа може зупинитися,

змінити напрямок руху або обрати іншу траєкторію. Якщо камера виявляє заданий об'єкт, робот може наближатися до нього, супроводжувати його або виконувати іншу запрограмовану дію. Отже, комп'ютерний зір у складі роботизованої платформи є не окремим програмним модулем, а частиною загальної системи прийняття рішень [1, 5].

Для забезпечення якісного розпізнавання важливе значення має навчальна вибірка. Вона повинна містити достатню кількість зображень об'єктів, які система має розпізнавати, бажано в різних умовах освітлення, на різному фоні, під різними кутами та на різних відстанях. Зазвичай набір даних поділяють на навчальну, валідаційну та тестову частини. Навчальна частина використовується для налаштування параметрів моделі, валідаційна – для контролю процесу навчання, а тестова – для перевірки здатності моделі працювати з новими зображеннями [6].

Оцінювання якості розпізнавання об'єктів здійснюється за допомогою таких показників, як точність класифікації, повнота виявлення, кількість хибних спрацювань, середня точність детектування, швидкість обробки кадрів та затримка реакції системи. Для роботизованої платформи ці показники мають не лише програмне, а й практичне значення. Важливо, щоб система встигала обробити зображення до того, як робот наблизиться до перешкоди або об'єкта, а результати розпізнавання були достатньо стабільними під час руху [5, 6].

Основними принципами комп'ютерного зору в задачах розпізнавання об'єктів є отримання якісного зображення, його попередня обробка, виділення інформативних ознак, класифікація та локалізація об'єктів, оцінювання достовірності результату й передавання отриманих даних до системи керування. У роботизованій платформі ці принципи мають прикладне значення, оскільки результат розпізнавання безпосередньо впливає на поведінку робота, його здатність орієнтуватися в середовищі та виконувати задані дії.

1.2 Методи машинного навчання для класифікації та виявлення об'єктів на зображеннях

Машинне навчання є одним із ключових підходів до реалізації інтелектуального розпізнавання об'єктів у системах комп'ютерного зору. Його основна ідея полягає в тому, що програмна модель не отримує жорстко задані правила для кожного можливого випадку, а навчається знаходити закономірності на основі прикладів. У задачах розпізнавання зображень такими прикладами є набори фотографій або відеокадрів, для яких заздалегідь відомо, які об'єкти на них зображено. Завдяки цьому модель може поступово формувати здатність розпізнавати схожі об'єкти на нових зображеннях, які не використовувалися під час навчання [7].

У межах комп'ютерного зору найчастіше використовують навчання з учителем. У цьому випадку для кожного зображення або його окремої області вказується правильна відповідь, тобто клас об'єкта, його координати або маска сегментації. Наприклад, якщо модель навчається розпізнавати автомобіль, людину або перешкоду, то в навчальному наборі мають бути зображення з відповідними мітками. Для класифікації достатньо вказати клас усього зображення, а для детектування потрібно додатково позначити розташування об'єкта за допомогою рамки. Саме наявність якісно підготовлених і розмічених даних значною мірою визначає точність майбутньої системи розпізнавання [8].

Одним із базових завдань машинного навчання в комп'ютерному зорі є класифікація зображень. Вона передбачає віднесення зображення до певного класу, наприклад «людина», «машина», «предмет», «перешкода» тощо. У такому випадку модель аналізує все зображення загалом і формує відповідь про найбільш імовірний клас. Для роботизованої платформи класифікація може бути корисною тоді, коли потрібно визначити, який об'єкт перебуває в полі зору камери. Проте сама класифікація не дає інформації про точне положення об'єкта в кадрі, тому для керування рухом робота її можливості є обмеженими.

Більш практичним для мобільної роботизованої платформи є детектування об'єктів. На відміну від класифікації, детектування дозволяє не лише визначити клас об'єкта, а й знайти його розташування на зображенні. Результатом роботи моделі зазвичай є прямокутна рамка навколо об'єкта, назва класу та значення впевненості. Для робота це має важливе значення, оскільки координати об'єкта в кадрі можуть бути використані для прийняття рішень: зупинитися, повернути, об'їхати перешкоду або рухатися в напрямку виявленого об'єкта. Саме тому в системах мобільної робототехніки детектування часто є більш цінним, ніж проста класифікація [9].

Основою сучасних методів розпізнавання об'єктів є нейронні мережі, зокрема згорткові нейронні мережі. Їх особливість полягає в тому, що вони здатні автоматично виділяти ознаки зображення на різних рівнях складності. На початкових шарах мережа може виявляти прості елементи, наприклад краї, кути або кольорові переходи. На глибших шарах формуються складніші ознаки, що відповідають частинам об'єктів або їхнім цілісним візуальним образам. Такий підхід дає змогу уникнути ручного опису ознак і підвищити якість розпізнавання в умовах змінного освітлення, різних фонів та різних положень об'єкта [10].

Для задач реального часу часто використовують моделі сімейства YOLO. Їхня перевага полягає в тому, що зображення аналізується за один прохід нейронної мережі, а результат формується у вигляді набору знайдених об'єктів, їхніх координат і рівнів упевненості. Такі моделі орієнтовані на швидке виявлення об'єктів і тому можуть застосовуватися в робототехніці, відеоспостереженні, автономній навігації та інших системах, де важлива оперативна реакція. У сучасних дослідженнях, зокрема в YOLOv10, увага приділяється зменшенню затримки обробки, оптимізації архітектури та підвищенню балансу між точністю і швидкодією [9].

Окрім YOLO-подібних моделей, у задачах виявлення об'єктів активно розвиваються підходи на основі Transformer-архітектур. Одним із прикладів є RT-DETR, який орієнтований на детектування об'єктів у режимі реального часу. Такі моделі демонструють, що сучасні методи машинного навчання поступово

переходять від класичних згорткових архітектур до комбінованих підходів, де враховується не лише локальна структура зображення, а й ширші зв'язки між його частинами [11]. Для роботизованої платформи це відкриває можливість використання більш точних моделей, однак водночас підвищує вимоги до обчислювальних ресурсів.

Важливим методом практичного застосування машинного навчання є перенесення навчання. Його суть полягає в тому, що для розпізнавання об'єктів використовується вже попередньо навчена модель, яка адаптується до конкретного набору даних. Наприклад, модель, навчена на великому наборі зображень, може бути додатково навчена розпізнавати об'єкти, які важливі для конкретної роботизованої платформи. Це зменшує потребу у великій кількості власних зображень і скорочує час підготовки моделі. Такий підхід підтримується, зокрема, інструментами TensorFlow Lite Model Maker, які спрощують навчання моделей для подальшого використання на мобільних та вбудованих пристроях [12].

Під час створення моделі розпізнавання важливо правильно підготувати набір даних. Він має містити зображення об'єктів у різних умовах: при різному освітленні, на різному фоні, з різної відстані, під різними кутами та з можливими частковими перекриттями. Якщо навчальні дані є надто однотипними, модель може добре працювати лише в обмежених умовах і помилятися під час реального тестування. Для роботизованої платформи це особливо важливо, оскільки під час руху камера може отримувати нестабільні зображення, а об'єкти можуть змінювати положення в кадрі.

Процес навчання моделі зазвичай передбачає поділ даних на навчальну, валідаційну та тестову вибірки. Навчальна вибірка використовується для налаштування параметрів моделі, валідаційна – для контролю якості навчання, а тестова - для перевірки здатності моделі працювати з новими даними. Такий поділ дозволяє зменшити ризик перенавчання, коли модель добре запам'ятовує навчальні приклади, але погано розпізнає нові зображення. У практичній

реалізації системи для мобільного робота потрібно прагнути не лише високої точності на навчальних даних, а й стабільної роботи в реальних умовах.

Для оцінювання якості моделей машинного навчання використовують різні метрики. У задачах класифікації поширеними є точність, повнота та загальна частка правильних відповідей. У задачах детектування додатково враховують правильність локалізації об'єкта, кількість хибних спрацювань, кількість пропущених об'єктів і середню точність виявлення. Для роботизованої платформи важливо також оцінювати швидкість обробки кадрів, затримку реакції системи та стабільність роботи під час руху. Модель, яка має високу точність, але працює занадто повільно, може бути непридатною для керування мобільним роботом у реальному часі.

Під час вибору методу машинного навчання для роботизованої платформи необхідно враховувати апаратні обмеження. Arduino-сумісний контролер, доцільно застосовувати для керування двигунами, датчиками та виконавчими механізмами. Водночас повноцінну обробку зображень і запуск моделей машинного навчання доцільно виконувати на зовнішньому пристрої, наприклад комп'ютері, смартфоні, мінікомп'ютері або edge-модулі. Для таких пристроїв можуть використовуватися полегшені моделі, зокрема MobileNet, які орієнтовані на роботу в умовах обмежених ресурсів [10].

Методи машинного навчання дають змогу реалізувати інтелектуальне розпізнавання об'єктів на зображеннях і відеокадрах. Для роботизованої платформи найбільш доцільними є моделі детектування об'єктів, оскільки вони дозволяють визначати не лише клас об'єкта, а й його положення в кадрі. У межах розроблюваної системи машинне навчання може використовуватися як окремий програмний модуль, який аналізує відеопотік з камери, формує результати розпізнавання та передає їх до системи керування рухом. Такий підхід забезпечує поєднання апаратної частини мобільного робота з інтелектуальними алгоритмами комп'ютерного зору (рис. 1.2).

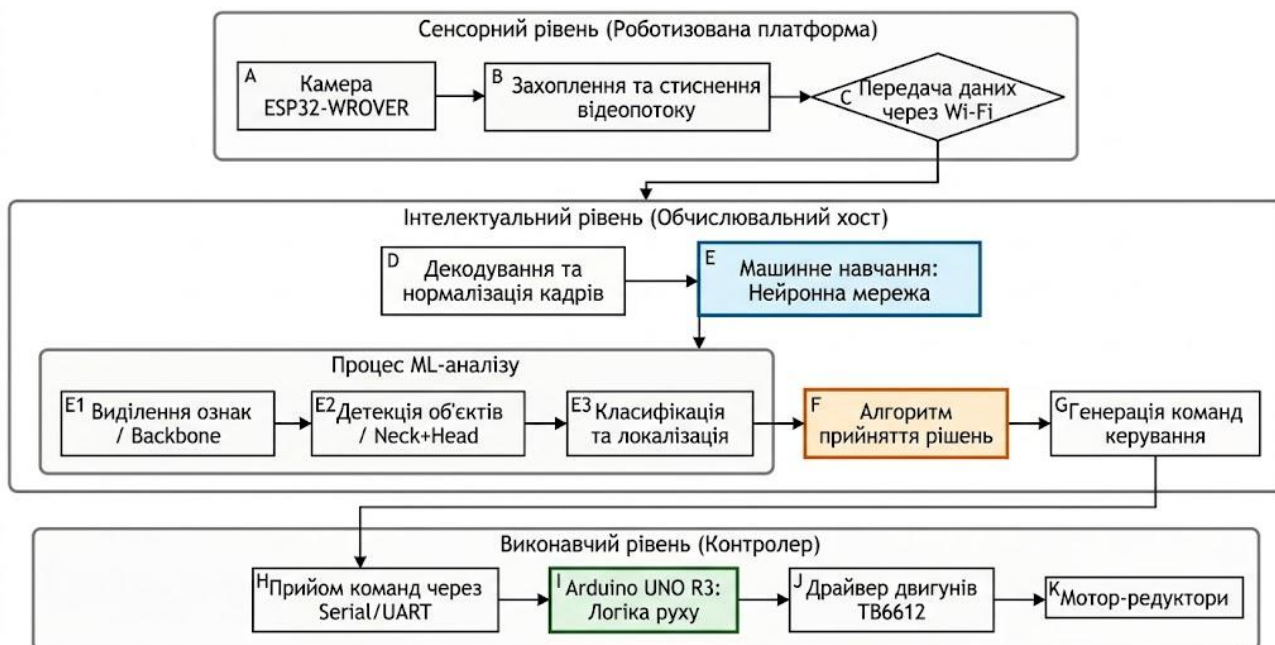


Рисунок 1.2 – Блок-схема застосування машинного навчання для розпізнавання об'єктів роботизованою платформою [12]

1.3 Особливості використання мікроконтролерних платформ у мобільній робототехніці

Мікроконтролерні платформи є основою багатьох навчальних, експериментальних і прикладних мобільних роботизованих систем. Їх використання пояснюється тим, що мікроконтролер поєднує в одному пристрої обчислювальне ядро, пам'ять, цифрові та аналогові інтерфейси введення-виведення, засоби керування периферією та можливість програмування алгоритмів роботи пристрою. У мобільній робототехніці мікроконтролер виконує роль центрального вузла керування, який отримує сигнали від датчиків, обробляє їх відповідно до заданої програми та формує керуючі сигнали для двигунів, сервоприводів, світлової або звукової індикації.

Однією з найпоширеніших мікроконтролерних платформ у навчальній робототехніці є Arduino-сумісна платформа. Її популярність зумовлена відкритістю екосистеми, простотою програмування, наявністю великої кількості бібліотек, модулів розширення та прикладів реалізації роботизованих пристроїв. Arduino Uno Rev3 побудована на базі мікроконтролера ATmega328P, має 14

цифрових входів-виходів, 6 аналогових входів, 6 PWM-виходів, 32 КБ Flash-пам'яті, 2 КБ SRAM, 1 КБ EEPROM і тактову частоту 16 МГц [13]. Такі характеристики є достатніми для реалізації базових алгоритмів керування рухом, зчитування даних із датчиків, формування ШІМ-сигналів для двигунів і організації обміну даними з додатковими модулями.

У мобільному роботі мікроконтролерна платформа найчастіше відповідає за низькорівневе керування. До таких функцій належать запуск і зупинка двигунів, зміна напрямку руху, керування швидкістю, обробка сигналів від ультразвукових або інфрачервоних датчиків, визначення наявності перешкод, керування сервоприводом, обмін даними через послідовний порт, Bluetooth або Wi-Fi-модуль. Наприклад, в офіційній документації Arduino Robot окремо зазначено функціональний поділ між платою керування двигунами та платою керування логікою робота, що демонструє доцільність розділення задач руху та обробки даних у роботизованій системі [14].

Особливістю мікроконтролерних платформ є робота в умовах обмежених обчислювальних ресурсів. Порівняно з персональними комп'ютерами, смартфонами або одноплатними комп'ютерами типу Raspberry Pi, мікроконтролери мають значно менший обсяг оперативної пам'яті, нижчу тактову частоту та обмежені можливості для обробки великих масивів даних. Тому їх доцільно використовувати для задач реального часу, які потребують швидкої реакції на сигнали датчиків, але не вимагають складних обчислень. Саме тому Arduino-сумісний контролер у роботизованій платформі доцільно розглядати як виконавчий та керуючий вузол, а не як основний пристрій для повноцінного розпізнавання об'єктів засобами машинного навчання.

У контексті роботи з комп'ютерним зором це обмеження має принципове значення. Повноцінна обробка відеопотоку, аналіз зображень і запуск моделей машинного навчання потребують значно більших ресурсів, ніж ті, які доступні класичним Arduino-сумісним платам. Сучасний напрям TinyML передбачає запуск спрощених моделей машинного навчання на мікроконтролерах, однак такі моделі потребують оптимізації, квантування, зменшення розміру та адаптації до

конкретної апаратної платформи [15]. Доцільним є комбінований підхід: мікроконтролер відповідає за керування рухом, а розпізнавання об'єктів виконується на окремому пристрої або модулі з більшими обчислювальними можливостями.

Такий підхід відповідає логіці побудови роботизованих систем, у яких функції розподіляються між різними рівнями. Нижній рівень забезпечує взаємодію з апаратними компонентами: двигунами, датчиками, драйверами моторів, індикаторами та модулями живлення. Верхній рівень виконує обробку складніших даних, зокрема зображень, відеопотоку або результатів роботи моделі машинного навчання. Після цього результат аналізу передається до мікроконтролера у вигляді простої команди: рухатися вперед, зупинитися, повернути ліворуч, повернути праворуч, об'їхати перешкоду або слідувати за об'єктом.

Для практичної реалізації такої системи може бути використаний набір ELEGOO Smart Robot Car Kit V4.0. Він є навчальним роботизованим набором на базі Arduino-сумісної платформи, який орієнтований на вивчення програмування, електроніки та робототехніки [16]. Виробник надає до набору інструкції, програмний код, застосунок і даташити, що спрощує процес складання, налагодження та тестування роботизованої платформи [17]. Наявність камери у версії V4.0 дозволяє розглядати цей набір не лише як платформу для руху та уникнення перешкод, а й як основу для експериментів із візуальним сприйняттям середовища.

Водночас потрібно враховувати, що камера в такій системі не означає автоматичну можливість виконання складного машинного навчання безпосередньо на Arduino-контролері. Наприклад, ELEGOO окремо описує процедуру завантаження коду до модуля камери, що свідчить про функціональну відокремленість камери від основного контролера роботи [18]. Це підтверджує доцільність архітектури, у якій відеомодуль або зовнішній пристрій виконує обробку зображень, а Arduino-сумісний контролер приймає лише підсумкові команди для керування рухом.

Важливою перевагою мікроконтролерних платформ у мобільній робототехніці є їхня здатність працювати з різними типами датчиків. До роботизованої платформи можуть підключатися ультразвукові датчики відстані, інфрачервоні датчики лінії, датчики освітлення, енкодери двигунів, гіроскопи, акселерометри, модулі Bluetooth, Wi-Fi або інші периферійні пристрої. Завдяки цьому мікроконтролер може поєднувати дані з кількох джерел і формувати більш надійну поведінку робота. Наприклад, якщо комп'ютерний зір визначає наявність об'єкта, а ультразвуковий датчик підтверджує малу відстань до нього, система може прийняти рішення про зупинку або зміну траєкторії.

Ще однією особливістю використання мікроконтролерів є необхідність врахування живлення та енергоспоживання. Мобільний робот працює від акумуляторного джерела, тому всі його компоненти мають споживати енергію раціонально. Двигуни, камера, бездротові модулі та сервоприводи створюють додаткове навантаження на систему живлення. Мікроконтролер у такій системі повинен не лише виконувати програмний алгоритм, а й забезпечувати стабільну взаємодію компонентів, не допускаючи перевантаження портів введення-виведення. Для підключення двигунів зазвичай застосовують окремі драйвери моторів, оскільки порти мікроконтролера не призначені для безпосереднього живлення силових навантажень.

У процесі проєктування роботизованої платформи важливе значення має вибір інтерфейсів обміну даними. Arduino-сумісні плати підтримують поширені інтерфейси UART, I2C, SPI, а також цифрові й аналогові входи-виходи. UART може використовуватися для обміну з Bluetooth-модулем, Wi-Fi-модулем або зовнішнім пристроєм; I2C - для підключення сенсорів і модулів розширення; SPI – для швидкого обміну з окремими периферійними пристроями. Така гнучкість дозволяє поступово розширювати функціональність мобільного робота, додаючи нові сенсори або модулі без повної зміни апаратної архітектури.

З позиції програмної реалізації перевагою Arduino-сумісних платформ є використання середовища Arduino IDE та мови програмування, наближеної до C/C++. Це дозволяє реалізовувати алгоритми керування у вигляді зрозумілої

структури програми, яка складається з функції налаштування `setup()` та основного циклу `loop()`. У функції `setup()` задаються режими роботи пінів, ініціалізуються бібліотеки та інтерфейси зв'язку. У циклі `loop()` постійно виконується зчитування датчиків, аналіз умов і формування керуючих сигналів. Така модель програмування добре підходить для мобільного робота, оскільки його робота є циклічною та залежить від постійного аналізу поточного стану середовища.

У межах розробленої роботизованої платформи мікроконтролер доцільно використовувати для виконання стабільних і повторюваних задач: керування напрямком руху, регулювання швидкості, обробки простих сенсорних сигналів, отримання команд від модуля розпізнавання та виконання відповідної дії. Складніші задачі, пов'язані з комп'ютерним зором і машинним навчанням, мають бути винесені на окремий рівень обробки. Такий розподіл підвищує надійність системи, зменшує навантаження на мікроконтролер і дозволяє поєднати переваги простої апаратної платформи з можливостями сучасних інтелектуальних алгоритмів.

Загальну роль мікроконтролерної платформи у складі мобільного робота доцільно подати на рисунку 1.4.

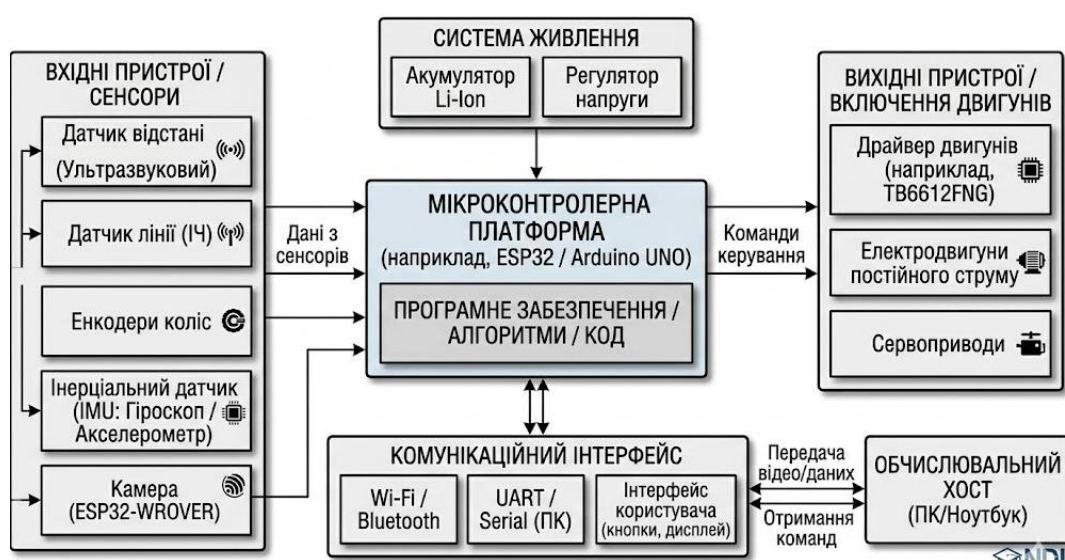


Рисунок 1.3 – Роль мікроконтролерної платформи у структурі мобільного робота [18]

Мікроконтролерні платформи є важливим елементом мобільної робототехніки, оскільки забезпечують безпосереднє керування апаратними компонентами робота. Їхніми перевагами є доступність, простота програмування, сумісність із великою кількістю датчиків і модулів, а також придатність для реалізації алгоритмів руху в реальному часі. Водночас обмежені обчислювальні ресурси не дозволяють ефективно виконувати складні задачі комп'ютерного зору без додаткових пристроїв. Тому для роботизованої платформи з інтелектуальним розпізнаванням об'єктів найбільш обґрунтованим є комбінований підхід, за якого Arduino-сумісний контролер відповідає за рух і взаємодію з датчиками, а розпізнавання об'єктів реалізується на окремому програмному або апаратному рівні.

1.4 Аналіз існуючих рішень роботизованих платформ з елементами інтелектуального керування

Розвиток мобільної робототехніки сприяв появі значної кількості навчальних, дослідницьких і прикладних роботизованих платформ, які поєднують механічну основу, електронні модулі керування, сенсори, засоби зв'язку та програмне забезпечення для автономного або напівавтономного руху. Такі платформи використовуються для вивчення основ робототехніки, комп'ютерної інженерії, програмування мікроконтролерів, комп'ютерного зору, машинного навчання та алгоритмів навігації. Їх аналіз є важливим для обґрунтування вибору апаратної бази майбутньої розробки, оскільки дає змогу порівняти можливості різних типів роботів, визначити їхні переваги, обмеження та доцільність використання в межах кваліфікаційної роботи.

Існуючі роботизовані платформи умовно можна поділити на кілька груп. До першої групи належать прості Arduino-сумісні роботи, орієнтовані на вивчення базових принципів руху, роботи з датчиками та керування виконавчими механізмами. До другої групи можна віднести роботи на базі одноплатних комп'ютерів, зокрема Raspberry Pi або NVIDIA Jetson, які мають більші

обчислювальні можливості та можуть виконувати складніші задачі комп'ютерного зору. До третьої групи належать ROS-сумісні дослідницькі платформи, які використовуються для вивчення автономної навігації, побудови карт, SLAM-алгоритмів, планування траєкторій і багаторівневого керування. Для бакалаврської роботи важливо обрати таку платформу, яка буде доступною, достатньо функціональною та придатною для практичного складання і тестування.

Одним із найбільш відомих рішень у сфері навчальної та дослідницької робототехніки є TurtleBot3. Це малогабаритна, модульна та налаштовувана мобільна роботизована платформа, орієнтована на використання з ROS у навчанні, дослідженнях, прототипуванні та аматорських проєктах [19]. Її перевагою є підтримка екосистеми ROS, можливість підключення різних сенсорів, зокрема лазерного сканера, а також використання для задач автономної навігації, картографування та керування рухом. TurtleBot3 є більш професійним рішенням порівняно з простими Arduino-наборами, однак його використання потребує глибшого розуміння ROS, Linux-середовища, мережевої взаємодії та складнішої програмної архітектури.

Іншим прикладом є NVIDIA JetBot - навчальна роботизована платформа на основі мінікомп'ютера NVIDIA Jetson Nano. Її особливістю є орієнтація саме на задачі штучного інтелекту, комп'ютерного зору та нейронних мереж. NVIDIA зазначає, що JetBot працює на базі Jetson Nano AI computer і може підтримувати роботу з кількома сенсорами та нейронними мережами для розпізнавання об'єктів [20]. Така платформа є потужнішою з погляду машинного навчання, оскільки має графічні обчислювальні ресурси, придатні для запуску моделей комп'ютерного зору. Водночас для бакалаврської роботи на базі Arduino вона може бути надмірно складною та дорожчою, а також потребувати інших підходів до програмування.

До поширених рішень також належать роботи на базі Raspberry Pi, наприклад навчальні набори типу PiCar-X або інші smart robot car kits. Такі платформи зазвичай поєднують колісне шасі, камеру, серводвигуни, модулі

керування рухом та програмне середовище Python. Їх перевагою є можливість використання бібліотек OpenCV, TensorFlow Lite та інших засобів для обробки зображень і реалізації простих моделей розпізнавання. У порівнянні з Arduino-сумісними роботами Raspberry Pi має значно більші обчислювальні можливості, що дозволяє виконувати частину задач комп'ютерного зору безпосередньо на борту робота. Проте така система є складнішою з погляду налаштування операційної системи, роботи з живленням, залежностями програмних бібліотек і стабільністю виконання програм у реальному часі.

Окрему групу становлять навчальні набори на базі Arduino, до яких належить ELEGOO Smart Robot Car Kit V4.0. Цей набір орієнтований на вивчення електроніки, програмування та базових принципів робототехніки. Виробник надає до нього навчальні матеріали, інструкції, програмний код, застосунок і даташити, що значно спрощує процес складання та налагодження платформи [21]. Важливою перевагою ELEGOO Smart Robot Car Kit V4.0 є те, що він має готову апаратну основу для реалізації мобільного робота: шасі, двигуни, контролер, модулі керування, датчики перешкод, засоби бездротового керування та камеру. Це дозволяє зосередитися не лише на механічному складанні, а й на програмній логіці руху та інтеграції з модулем розпізнавання об'єктів.

Порівняно з TurtleBot3 або JetBot, набір ELEGOO має нижчий поріг входження. Для його використання достатньо базових знань Arduino IDE, схем підключення електронних компонентів і принципів роботи датчиків. Це робить платформу придатною для бакалаврської кваліфікаційної роботи, де важливо не лише описати теоретичні принципи, а й створити фізичний прототип. Водночас ELEGOO Smart Robot Car Kit V4.0 має певні обмеження. Основний Arduino-сумісний контролер не призначений для повноцінного запуску складних моделей машинного навчання, тому розпізнавання об'єктів доцільно реалізовувати на зовнішньому пристрої або окремому модулі обробки відеопотоку. Саме такий підхід дозволяє поєднати доступність Arduino-платформи з можливостями сучасного комп'ютерного зору.

Для наочності порівняння основних рішень доцільно подати в таблиці 1.1.

Таблиця 1.1 – Порівняння існуючих роботизованих платформ з елементами інтелектуального керування

Платформа	Апаратна основа	Основні можливості	Переваги	Обмеження
TurtleBot3	ROS-сумісна мобільна платформа	Навігація, SLAM, картографування, дослідження автономного руху	Підтримка ROS, модульність, дослідницьке призначення	Вища складність налаштування та програмування
NVIDIA JetBot	NVIDIA Jetson Nano	Комп'ютерний зір, нейронні мережі, AI-експерименти	Висока продуктивність для машинного навчання	Вища вартість і складність реалізації
Raspberry Pi robot car	Raspberry Pi, камера, Python	Обробка зображень, OpenCV, базове AI-керування	Добрі можливості для комп'ютерного зору	Потребує налаштування ОС і програмних залежностей
ELEGOO Smart Robot Car Kit V4.0	Arduino-сумісний контролер, датчики, камера	Рух, уникнення перешкод, керування, експерименти з камерою	Доступність, проста збірка, навчальні матеріали	Обмежені ресурси для запуску складних ML-моделей

Як видно з таблиці 1.1, кожна платформа має власну сферу доцільного використання. TurtleBot3 краще підходить для складніших дослідницьких задач, пов'язаних із ROS, навігацією та побудовою карт. NVIDIA JetBot є ефективним рішенням для задач штучного інтелекту та комп'ютерного зору, однак потребує потужнішої апаратної бази та складнішого програмного середовища. Raspberry Pi-роботи є компромісом між доступністю та можливістю запуску алгоритмів обробки зображень. Натомість ELEGOO Smart Robot Car Kit V4.0 є оптимальним рішенням для навчального та бакалаврського проєкту, оскільки дозволяє реалізувати фізичну мобільну платформу, дослідити роботу сенсорів, реалізувати алгоритми руху та доповнити систему зовнішнім модулем розпізнавання об'єктів.

У контексті роботи найбільш доцільним є використання комбінованої архітектури. Її суть полягає в тому, що Arduino-сумісний контролер ELEGOO Smart Robot Car Kit V4.0 виконує функції керування рухом, обробки сигналів від датчиків і взаємодії з виконавчими механізмами, а модуль комп'ютерного зору виконує аналіз зображень і формує результати розпізнавання. Такий підхід

відповідає типовій логіці побудови інтелектуальних роботизованих систем, у яких низькорівневе керування відокремлюється від високорівневої обробки даних. У результаті система залишається доступною для практичної реалізації, але водночас отримує ознаки інтелектуальної поведінки.

Особливу увагу під час аналізу існуючих рішень потрібно приділяти не лише апаратним характеристикам, а й можливості практичного тестування. Для бакалаврської роботи важливо, щоб обрана платформа дозволяла виконати всі основні етапи: складання робота, підключення компонентів, реалізацію програмного керування, отримання відеоданих, розпізнавання об'єктів і перевірку реакції робота на результати розпізнавання. ELEGOO Smart Robot Car Kit V4.0 відповідає цим вимогам, оскільки має готову конструктивну базу та документацію, а також може бути розширений за рахунок зовнішнього програмного модуля комп'ютерного зору.

Аналіз існуючих рішень показує, що сучасні роботизовані платформи з елементами інтелектуального керування відрізняються рівнем складності, продуктивністю, вартістю та сферою застосування. Для складних дослідницьких задач доцільними є ROS-сумісні або Jetson-платформи, тоді як для навчального прототипування та бакалаврської роботи більш раціональним є використання Arduino-сумісного набору. ELEGOO Smart Robot Car Kit V4.0 може бути використаний як практична апаратна основа для створення мобільного робота, а технології комп'ютерного зору та машинного навчання можуть бути інтегровані на рівні зовнішнього модуля обробки зображень. Це забезпечує баланс між доступністю, функціональністю та можливістю практичної реалізації проєкту.

РОЗДІЛ 2

ПРОЄКТУВАННЯ РОБОТИЗОВАНОЇ ПЛАТФОРМИ З КОМП'ЮТЕРНИМ ЗОРОМ ТА МАШИННИМ НАВЧАННЯМ

2.1 Обґрунтування вибору апаратної платформи для реалізації мобільного робота

Вибір апаратної платформи є одним із ключових етапів проєктування мобільного робота, оскільки саме апаратна база визначає функціональні можливості системи, стабільність її руху, сумісність із сенсорними модулями, можливість підключення камери, підтримку засобів зв'язку та зручність програмної реалізації. Для роботизованої платформи з інтелектуальним розпізнаванням об'єктів апаратна частина повинна забезпечувати не лише пересування в просторі, а й взаємодію з датчиками, виконавчими механізмами, модулями керування та програмним блоком комп'ютерного зору.

Під час вибору платформи для реалізації мобільного робота враховувалися кілька основних критеріїв: доступність компонентів, простота складання, підтримка Arduino IDE, наявність готових модулів керування рухом, можливість підключення датчиків, підтримка бездротової взаємодії, наявність або можливість інтеграції камери, автономне живлення та придатність до експериментального тестування.

На рисунку 2.1 доцільно показати, що вибір апаратної платформи ґрунтується на сукупності технічних і практичних вимог. Центральним елементом схеми є апаратна платформа мобільного робота, навколо якої розміщуються основні критерії: мікроконтролерне керування, датчики та сенсори, двигуни й драйвери, камера або модуль комп'ютерного зору, засоби зв'язку, система живлення, програмна підтримка та можливість тестування.

Доцільним є використання Arduino-сумісної мобільної платформи. Такий вибір пояснюється тим, що Arduino-сумісні рішення мають відкриту екосистему, просте середовище програмування, значну кількість сумісних модулів і достатні можливості для керування базовими апаратними компонентами мобільного

робота. Arduino Uno Rev3, яка часто використовується в навчальних роботизованих системах, побудована на базі мікроконтролера ATmega328P, має 14 цифрових входів-виходів, 6 аналогових входів, 6 PWM-виходів, USB - інтерфейс і тактову частоту 16 МГц [22]. Такі характеристики є достатніми для керування двигунами, сервоприводами, датчиками перешкод, індикаторами та модулями зв'язку.

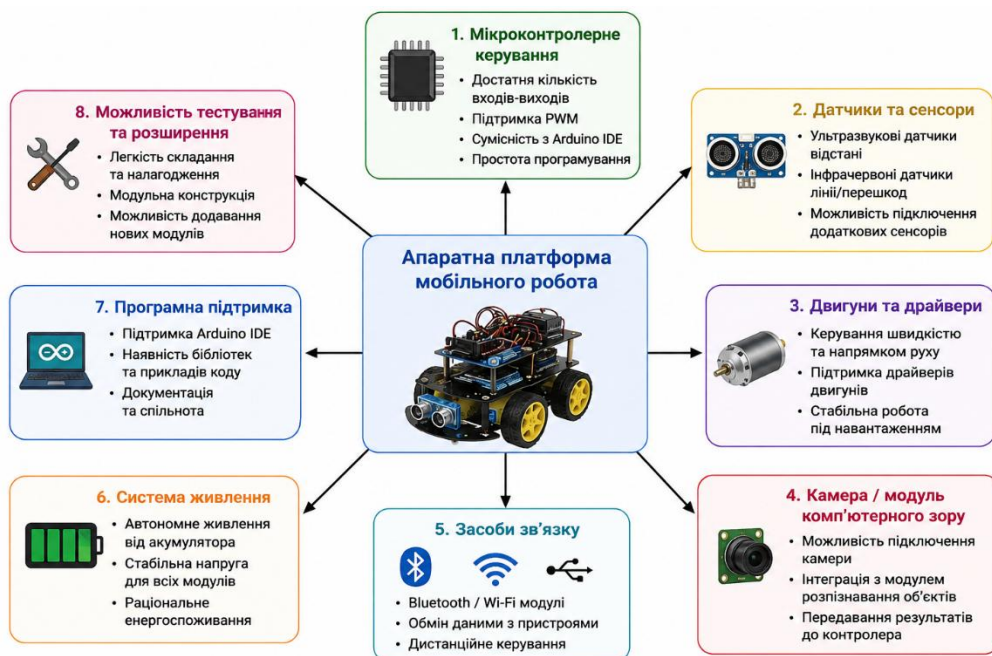


Рисунок 2.1 – Основні критерії вибору апаратної платформи для реалізації мобільного робота [22]

Доцільним є використання Arduino-сумісної мобільної платформи. Такий вибір пояснюється тим, що Arduino-сумісні рішення мають відкриту екосистему, просте середовище програмування, значну кількість сумісних модулів і достатні можливості для керування базовими апаратними компонентами мобільного робота. Arduino Uno Rev3, яка часто використовується в навчальних роботизованих системах, побудована на базі мікроконтролера ATmega328P, має 14 цифрових входів-виходів, 6 аналогових входів, 6 PWM-виходів, USB - інтерфейс і тактову частоту 16 МГц [22]. Такі характеристики є достатніми

для керування двигунами, сервоприводами, датчиками перешкод, індикаторами та модулями зв'язку.

Водночас Arduino-сумісний контролер не доцільно розглядати як основний пристрій для повноцінного виконання алгоритмів комп'ютерного зору та машинного навчання. Його роль у структурі мобільного робота полягає насамперед у виконанні низькорівневих задач керування: зчитуванні сигналів від датчиків, формуванні керуючих імпульсів для двигунів, реалізації базової логіки руху та прийманні команд від зовнішнього або вбудованого модуля розпізнавання. Такий підхід дозволяє раціонально розподілити функції між компонентами системи та уникнути перевантаження мікроконтролера складними обчисленнями.

Для практичної реалізації мобільного робота обрано готову навчальну роботизовану платформу на базі Arduino-сумісного контролера. Одним із таких рішень є ELEGOO Smart Robot Car Kit V4.0, який призначений для вивчення програмування, електроніки, основ робототехніки та роботи з сенсорними модулями [23]. Набір містить апаратну основу для створення мобільного робота, зокрема шасі, двигуни, колеса, плату керування, датчики, модулі зв'язку та елементи для реалізації базових режимів руху. Виробник також надає навчальні матеріали, інструкції, приклади програмного коду, застосунок і даташити, що спрощує процес складання, налаштування та тестування платформи [24].

Перевагою вибору готової апаратної платформи є те, що вона дозволяє зменшити час на добір окремих компонентів і зосередитися на реалізації логіки роботи системи. У такому випадку основна увага приділяється не лише механічному складанню, а й програмному керуванню рухом, інтеграції сенсорів, передаванню команд і взаємодії з модулем комп'ютерного зору.

На рисунку 2.2 відображено взаємодію основних апаратних компонентів. Центральним елементом є Arduino-сумісний мікроконтролер, до якого підключаються датчики перешкод, інфрачервоні датчики лінії, модулі зв'язку, система живлення та драйвери двигунів. Через драйвери здійснюється керування двигунами й колесами, що забезпечує рух мобільної платформи. Окремим

блоком доцільно показати камеру або модуль комп'ютерного зору, який аналізує візуальні дані та передає до мікроконтролера результати розпізнавання об'єктів або сформовані команди.

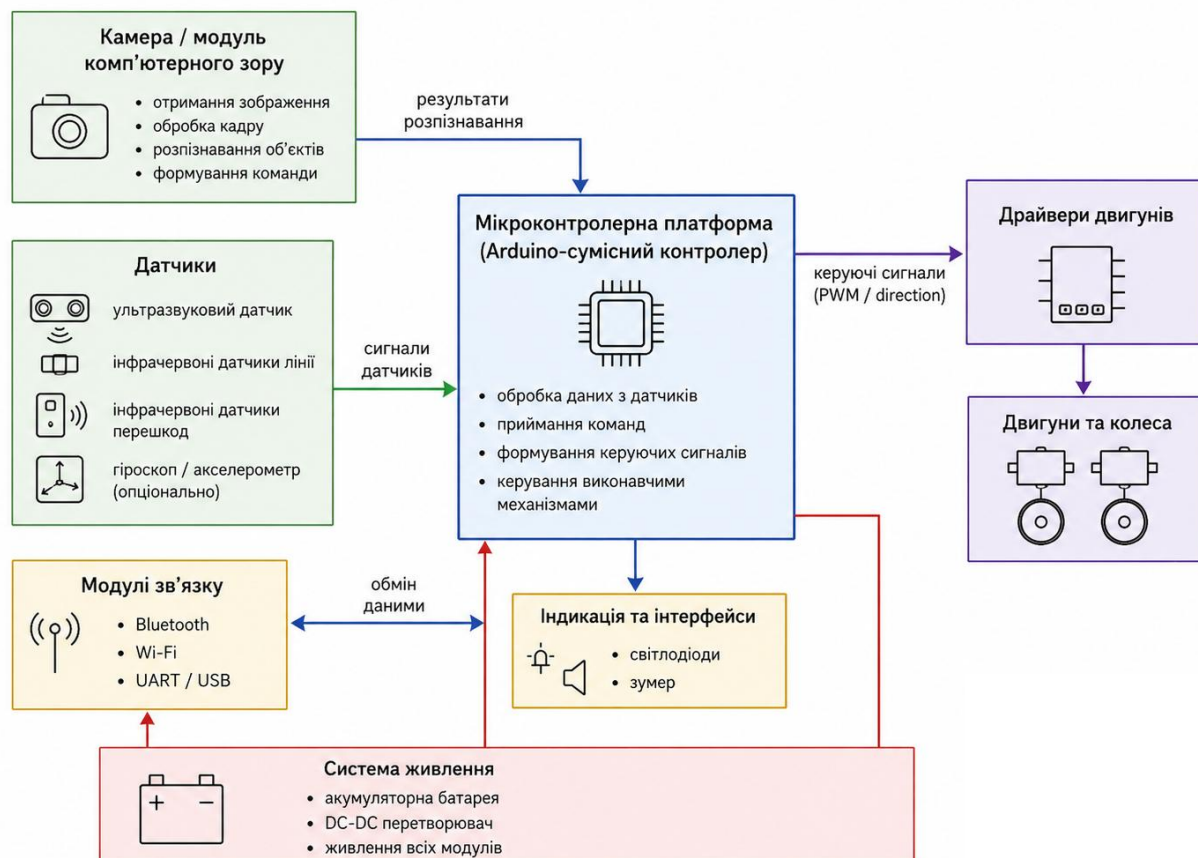


Рисунок 2.2 – Загальна структура апаратної платформи мобільного робота

Обрана апаратна платформа може бути використана як основа для поєднання класичного мікроконтролерного керування з елементами інтелектуального аналізу зображень. Камера або модуль комп'ютерного зору отримує візуальні дані з навколишнього середовища, виконує аналіз кадру, визначає наявність об'єкта, його клас, приблизне положення та рівень упевненості. Після цього результат передається до мікроконтролера у вигляді спрощеної команди: зупинитися, повернути, продовжити рух, об'їхати перешкоду або наблизитися до заданого об'єкта.

Обґрунтування вибору апаратної платформи наведено в таблиці 2.1.

Таблиця 2.1– Обґрунтування вибору апаратної платформи для реалізації мобільного робота

Критерій вибору	Вимоги до платформи	Відповідність обраної апаратної бази
Мікроконтролерне керування	Можливість керування двигунами, датчиками та виконавчими механізмами	Arduino-сумісна плата забезпечує цифрові, аналогові та PWM-виходи
Мобільність	Наявність шасі, коліс, двигунів і автономного живлення	Платформа має готову конструкцію мобільного робота
Сенсорна взаємодія	Підключення датчиків перешкод, лінії та інших модулів	Передбачено використання датчиків для руху, уникнення перешкод і навігації
Програмування	Просте середовище розробки та наявність прикладів коду	Підтримується Arduino IDE, доступні навчальні матеріали та приклади
Інтеграція комп'ютерного зору	Можливість використання камери або зовнішнього модуля аналізу зображень	Камера або зовнішній модуль можуть передавати результати розпізнавання до контролера
Експериментальне тестування	Можливість перевірити рух, реакцію на об'єкти та роботу датчиків	Платформа придатна для практичного складання, налагодження та тестування

Як видно з таблиці 2.1, обрана апаратна платформа відповідає основним вимогам до мобільного робота, який повинен виконувати рух, отримувати дані із сенсорів і реагувати на результати розпізнавання об'єктів. Її використання є доцільним не лише з технічного, а й з навчально-практичного погляду, оскільки вона дозволяє реалізувати повний цикл створення роботизованої системи: від складання апаратної частини до програмування, тестування та аналізу результатів.

Окремої уваги потребує розподіл функцій між мікроконтролером і модулем комп'ютерного зору. Мікроконтролерна платформа виконує функції керування рухом, зчитування даних із датчиків, взаємодії з драйверами двигунів і формування сигналів для виконавчих механізмів. Натомість модуль комп'ютерного зору відповідає за отримання зображення, попередню обробку кадру, розпізнавання об'єктів, визначення їхнього класу та передавання результатів аналізу. Такий розподіл функцій подано на рисунку 2.3.

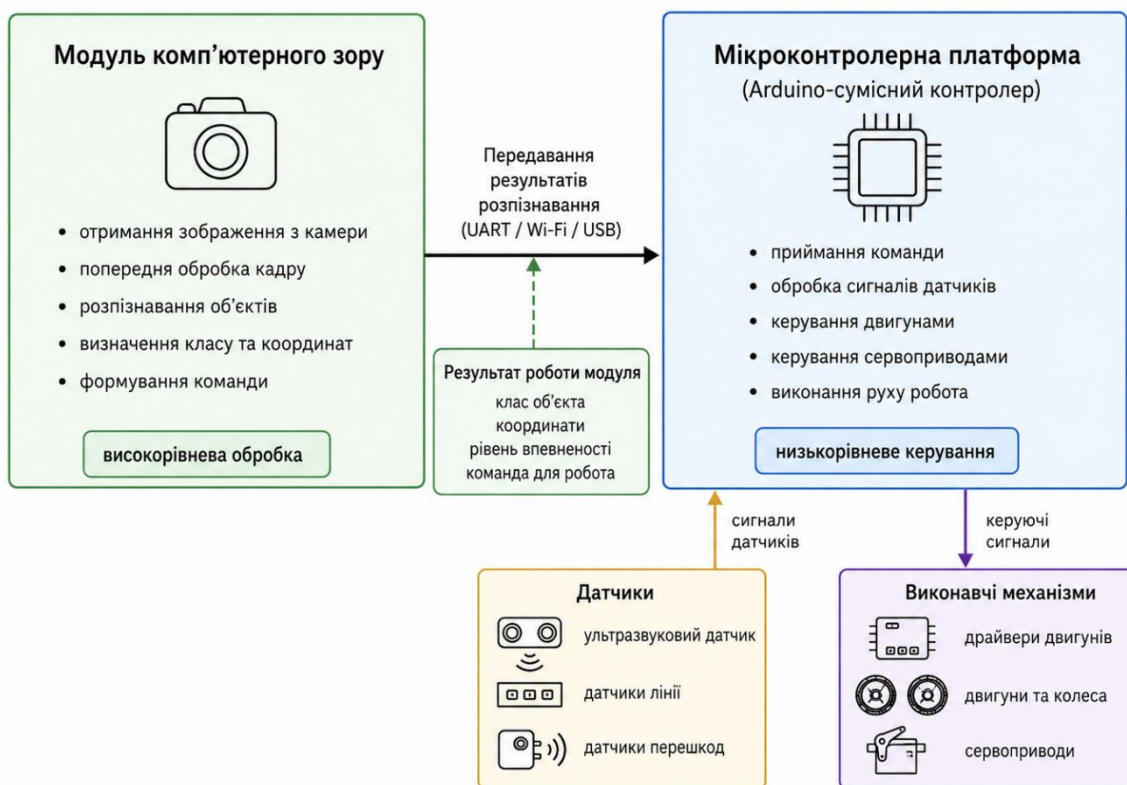


Рисунок 2.3 – Розподіл функцій між мікроконтролером і модулем комп'ютерного зору

На рисунку 2.3 показано два основні функціональні блоки. У першому блоці – модуль комп'ютерного зору відображаються процеси отримання зображення з камери, обробки кадру, розпізнавання об'єкта, визначення класу й координат та формування команди. У другому блоці – мікроконтролерна платформа подаються функції приймання команди, обробки сигналів датчиків, керування двигунами, керування сервоприводами та виконання руху робота. Між цими блоками доцільно показати канал передавання результатів розпізнавання через UART, Wi-Fi або USB.

Застосування такого функціонального поділу є технічно обґрунтованим, оскільки дозволяє не перевантажувати Arduino-сумісний контролер складними обчисленнями, пов'язаними з аналізом зображень. У результаті мікроконтролер стабільно виконує задачі реального часу, а розпізнавання об'єктів може здійснюватися на пристрої з вищою продуктивністю. Такий підхід забезпечує

поєднання доступної апаратної платформи з можливостями сучасного комп'ютерного зору та машинного навчання.

Загальна логіка роботи апаратної платформи полягає в тому, що камера або модуль комп'ютерного зору формує вхідні візуальні дані, програмний блок розпізнавання визначає наявність і тип об'єкта, а мікроконтролер отримує відповідну команду для керування виконавчими механізмами. Наприклад, у разі виявлення перешкоди робот може зупинитися або виконати об'їзд, а в разі виявлення цільового об'єкта – змінити напрям руху або перейти в режим супроводу.

Вибір Arduino-сумісної мобільної роботизованої платформи є обґрунтованим для реалізації бакалаврського проєкту, оскільки вона поєднує доступність, простоту програмування, наявність необхідних апаратних модулів і можливість інтеграції з технологіями комп'ютерного зору. У межах роботи така платформа використовується як основа для створення мобільного робота, здатного виконувати рух, взаємодіяти з датчиками та реагувати на результати інтелектуального розпізнавання об'єктів.

2.2 Розробка структурної та принципової схем роботизованої платформи

Розробка структурної та принципової схем є важливим етапом проєктування роботизованої платформи, оскільки саме на цьому етапі визначається склад системи, взаємозв'язок між її основними модулями, напрям передавання даних, логіка керування виконавчими механізмами та спосіб інтеграції модуля комп'ютерного зору. Структурна схема дає змогу подати роботизовану платформу на рівні функціональних блоків, а принципова схема деталізує підключення окремих апаратних компонентів до мікроконтролера, системи живлення та драйверів двигунів.

Роботизована платформа розглядається як апаратно-програмна система, що складається з мікроконтролерного блоку, сенсорного блоку, модуля

комп'ютерного зору, блоку керування двигунами, виконавчих механізмів, модуля зв'язку та системи живлення. Центральним елементом системи є Arduino-сумісний мікроконтролер, який виконує функції приймання сигналів від датчиків, обробки команд, формування керуючих сигналів для драйверів двигунів і взаємодії з модулем розпізнавання об'єктів.

Структурна схема роботизованої платформи повинна відображати не лише наявність окремих компонентів, а й логіку їх взаємодії. Модуль комп'ютерного зору отримує зображення з камери, виконує обробку кадру, розпізнає об'єкти та формує результат у вигляді класу об'єкта, координат або команди керування. Цей результат передається до мікроконтролера через відповідний канал обміну даними, наприклад UART, USB або Wi-Fi. Мікроконтролер, у свою чергу, аналізує отриману команду разом із даними від датчиків перешкод і формує сигнали для драйверів двигунів. Драйвери забезпечують подавання необхідної напруги та струму на двигуни, унаслідок чого мобільна платформа виконує рух, зупинку, поворот або об'їзд перешкоди.

Загальну структурну схему роботизованої платформи подано на рисунку 2.4.

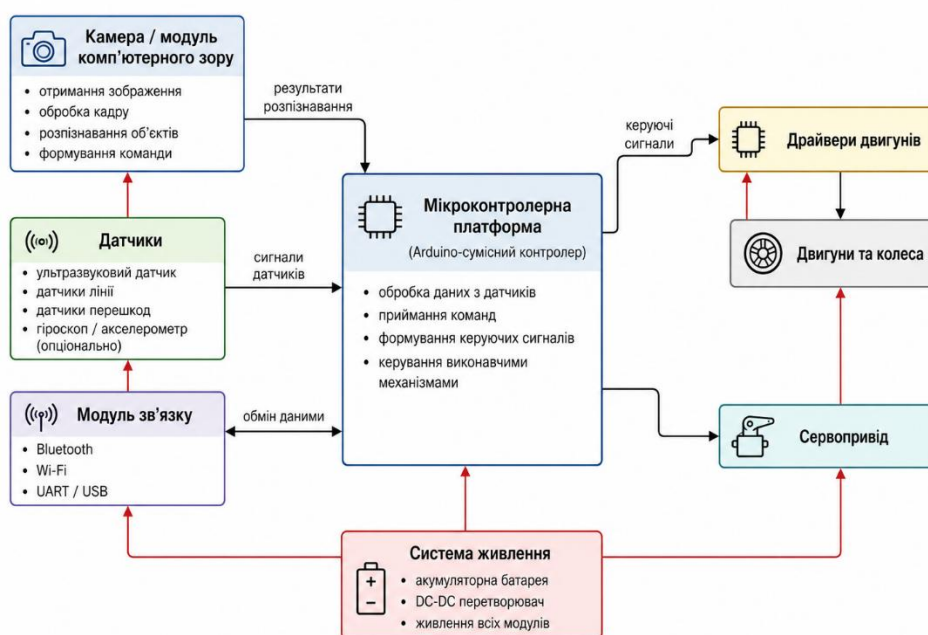


Рисунок 2.4 – Структурна схема роботизованої платформи з модулем комп'ютерного зору

У запропонованій структурі сенсорний блок виконує допоміжну роль, оскільки забезпечує платформу даними про безпосереднє оточення. Ультразвуковий датчик може використовуватися для визначення відстані до перешкод, інфрачервоні датчики - для виявлення лінії або близьких об'єктів, а додаткові датчики можуть застосовуватися для розширення функціональності системи. Дані сенсорів дають змогу підвищити надійність роботи роботи, оскільки рішення про рух приймається не лише на основі відеопотоку, а й на основі простих фізичних вимірювань.

Модуль комп'ютерного зору в структурній схемі виконує функцію високорівневого аналізу середовища. Його завдання полягає в отриманні зображення, попередній обробці кадру, виявленні об'єктів, визначенні їхнього класу та формуванні результату розпізнавання. Наприклад, якщо система розпізнає перешкоду або заданий об'єкт, до мікроконтролера може передаватися команда «зупинка», «поворот ліворуч», «поворот праворуч» або «рух вперед». Завдяки цьому мікроконтролер не виконує складних обчислень, пов'язаних з аналізом зображень, а працює з уже підготовленими командами.

Принципова схема роботизованої платформи деталізує підключення основних компонентів. Вона повинна враховувати, що мікроконтролер не може безпосередньо живити двигуни, оскільки вихідні порти Arduino-сумісної плати розраховані на керуючі сигнали, а не на силове навантаження. Тому між мікроконтролером і двигунами використовується драйвер двигунів, який приймає логічні сигнали керування та забезпечує подавання живлення на електродвигуни. Такий підхід дозволяє окремо організувати логічну частину керування та силову частину руху.

Принципову схему підключення компонентів роботизованої платформи подано на рисунку 2.5.

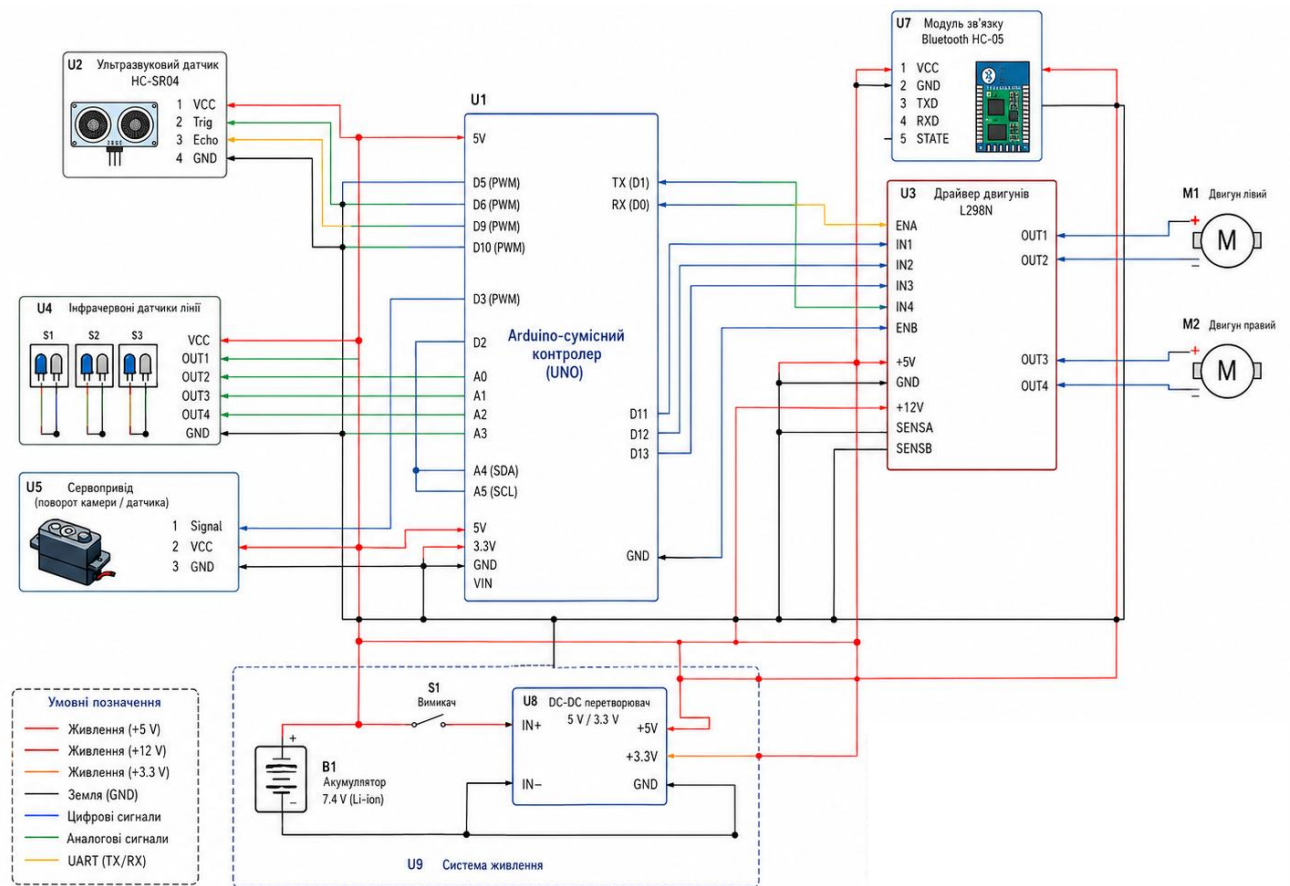


Рисунок 2.5 – Принципова схема підключення компонентів роботизованої платформи

У принциповій схемі доцільно розділити три типи з'єднань: сигнальні, силові та інформаційні. Сигнальні з'єднання використовуються для передавання логічних сигналів між мікроконтролером, датчиками та драйверами двигунів. Силові з'єднання забезпечують живлення двигунів, плати керування, датчиків і модулів зв'язку. Інформаційні з'єднання використовуються для передавання результатів розпізнавання від модуля комп'ютерного зору до мікроконтролера. Такий поділ дозволяє уникнути плутанини між лініями керування та лініями живлення.

Приклад розподілу підключень основних компонентів роботизованої платформи наведено в таблиці 2.2.

Таблиця 2.2 – Орієнтовне підключення компонентів роботизованої платформи до Arduino-сумісного контролера

Компонент	Призначення	Тип підключення	Приклад ліній підключення
Ультразвуковий датчик	Визначення відстані до перешкоди	Цифрові сигнали	Trig, Echo, 5 V, GND
Інфрачервоні датчики лінії	Виявлення лінії або контрастної поверхні	Цифрові або аналогові сигнали	OUT, 5 V, GND
Драйвер двигунів	Керування швидкістю та напрямком руху	PWM і цифрові сигнали	PWM, IN1, IN2, IN3, IN4
Двигуни постійного струму	Забезпечення руху платформи	Силове підключення через драйвер	Виходи драйвера до двигунів
Сервопривід	Поворот камери або датчика	PWM-сигнал	Signal, 5 V, GND
Модуль зв'язку	Обмін даними зі смартфоном або зовнішнім модулем	UART, USB або Wi-Fi	TX, RX, 5 V, GND
Камера/модуль комп'ютерного зору	Отримання зображення та розпізнавання об'єктів	USB, UART або Wi-Fi	Передавання результатів розпізнавання
Система живлення	Живлення всіх компонентів	Силове підключення	Акумулятор, стабілізатор, 5 V, GND

Як видно з таблиці 2.2, кожен компонент роботизованої платформи виконує окрему функцію та має власний тип підключення. Особливо важливим є правильне підключення системи живлення, оскільки двигуни створюють значне навантаження і можуть спричинити просідання напруги. Тому живлення силової частини та логічної частини доцільно організовувати з урахуванням стабілізації напруги та спільної лінії GND. Спільна «земля» необхідна для коректного передавання керуючих сигналів між мікроконтролером, драйверами двигунів і сенсорними модулями.

Під час розробки принципової схеми також потрібно враховувати особливості підключення сервоприводу. Якщо сервопривід використовується для повороту камери або ультразвукового датчика, він отримує керуючий PWM - сигнал від мікроконтролера, але його живлення має бути достатньо стабільним. У разі нестабільного живлення сервопривід може працювати ривками або спричинити перезавантаження мікроконтролера. Тому в практичній

реалізації доцільно передбачити окреме стабілізоване живлення для виконавчих механізмів або використовувати плату розширення, яка вже містить необхідні ланцюги живлення.

Для інтеграції модуля комп'ютерного зору з мікроконтролером важливо визначити формат обміну даними. Найпростішим варіантом є передавання не повного зображення, а лише результату розпізнавання. Наприклад, модуль комп'ютерного зору може надсилати короткі текстові або числові команди: «STOP», «LEFT», «RIGHT», «FORWARD», «OBJECT_FOUND». Такий підхід значно зменшує обсяг передаваних даних і спрощує програмну обробку на мікроконтролері. Arduino-сумісний контролер у цьому випадку лише приймає команду та перетворює її на відповідну дію виконавчих механізмів.

У процесі розробки структурної та принципової схем було враховано, що роботизована платформа повинна працювати в умовах автономного живлення, обмежених обчислювальних ресурсів і змінного середовища. Тому схема побудована за модульним принципом. Кожен функціональний блок може бути перевірений окремо: спочатку перевіряється робота двигунів і драйвера, потім підключаються датчики, після цього тестується модуль зв'язку, а на завершальному етапі виконується інтеграція з модулем комп'ютерного зору. Такий підхід спрощує налагодження системи та дає змогу швидко виявляти помилки підключення або програмної логіки.

Запропонована структура роботизованої платформи забезпечує можливість реалізації кількох режимів роботи. Перший режим – базове керування рухом, за якого робот виконує команди вперед, назад, ліворуч, праворуч і зупинка. Другий режим – рух із використанням датчиків перешкод, коли платформа може зупинятися або змінювати напрям у разі наближення до об'єкта. Третій режим – інтелектуальна реакція на результати комп'ютерного зору, коли поведінка робота залежить від того, який об'єкт було розпізнано в кадрі. Наприклад, у разі розпізнавання цільового об'єкта робот може рухатися в його напрямку, а в разі виявлення перешкоди – виконувати об'їзд.

Структурна схема роботизованої платформи відображає загальну логіку взаємодії її функціональних блоків, а принципова схема деталізує підключення основних компонентів до мікроконтролера, драйверів двигунів і системи живлення. Запропонована архітектура забезпечує розподіл задач між модулем комп'ютерного зору та Arduino-сумісним контролером, що дозволяє поєднати інтелектуальне розпізнавання об'єктів із надійним керуванням рухом мобільного робота. Такий підхід є доцільним для практичної реалізації бакалаврського проєкту, оскільки забезпечує зрозумілу, модульну та розширювану структуру роботизованої системи.

2.3 Вибір програмних засобів для реалізації комп'ютерного зору та машинного навчання

Вибір програмних засобів для реалізації комп'ютерного зору та машинного навчання є важливим етапом проєктування роботизованої платформи, оскільки саме програмне забезпечення забезпечує отримання зображення з камери, попередню обробку кадрів, розпізнавання об'єктів, формування результатів аналізу та передавання команд до мікроконтролерної системи керування. У межах цієї роботи програмні засоби повинні відповідати кільком основним вимогам: бути доступними, підтримувати роботу із зображеннями та відеопотоком, мати засоби для реалізації або використання моделей машинного навчання, забезпечувати можливість обміну даними з Arduino-сумісним контролером і бути придатними для експериментального тестування.

Для реалізації програмної частини доцільно використати мову програмування Python, оскільки вона широко застосовується в задачах комп'ютерного зору, машинного навчання, обробки даних і прототипування інтелектуальних систем. Офіційна документація Python містить опис мови, стандартної бібліотеки, засобів встановлення та налаштування середовища, що робить її зручною основою для створення програмного модуля розпізнавання об'єктів [25]. Використання Python у цій роботі є доцільним через велику

кількість сумісних бібліотек, простий синтаксис і можливість швидкого створення експериментальних програм для роботи з відеопотоком.

Основним програмним засобом для обробки зображень доцільно обрати бібліотеку OpenCV. Вона містить інструменти для читання, відображення та збереження зображень, роботи з відеопотоком, перетворення колірних просторів, зміни розміру кадру, фільтрації шумів, виділення контурів і реалізації базових алгоритмів комп'ютерного зору [26]. У межах роботизованої платформи OpenCV може використовуватися для отримання кадрів із камери, попередньої обробки зображень, підготовки кадру до передавання в модель розпізнавання та візуалізації результатів детектування.

Для реалізації машинного навчання та розпізнавання об'єктів доцільно використати TensorFlow Lite або інші полегшені моделі, орієнтовані на роботу з периферійними та мобільними пристроями. TensorFlow Lite Model Maker підтримує сценарій навчання власної моделі детектування об'єктів на основі підготовленого набору даних, що є важливим для практичної реалізації системи розпізнавання [27]. У роботі такий підхід може застосовуватися для створення моделі, яка розпізнає визначені класи об'єктів, після чого результат розпізнавання передається до мікроконтролера у вигляді спрощеної команди.

Альтернативним програмним засобом для детектування об'єктів можуть бути моделі сімейства YOLO, які орієнтовані на швидке виявлення об'єктів у зображеннях і відеопотоці. Документація Ultralytics зазначає, що YOLO використовується для задач детектування, сегментації, класифікації, оцінювання пози та відстеження об'єктів [28]. Для роботизованої платформи такі моделі є доцільними тоді, коли потрібно отримувати результат розпізнавання в режимі, наближеному до реального часу. Водночас для бакалаврського проєкту важливо обирати полегшені варіанти моделей, оскільки надмірно складні нейронні мережі можуть вимагати значних обчислювальних ресурсів.

Для підготовки власного набору даних доцільно використовувати інструменти розмітки зображень, зокрема CVAT. CVAT є інструментом для анотування зображень і відео в задачах комп'ютерного зору, а його документація

охоплює базові та розширені можливості розмітки даних [29]. У межах цієї роботи такий засіб може бути використаний для позначення об'єктів на зображеннях за допомогою рамок, формування навчальної вибірки та підготовки даних для подальшого навчання або донавчання моделі.

Для програмування мікроконтролерної частини доцільно використати Arduino IDE. Це середовище дає змогу створювати, перевіряти та завантажувати скетчі на Arduino-сумісні плати, встановлювати бібліотеки, працювати з послідовним монітором і виконувати налагодження базових програмних рішень [30]. У межах розроблюваної системи Arduino IDE використовується для реалізації програмного керування двигунами, обробки сигналів від датчиків, приймання команд від модуля комп'ютерного зору та формування керуючих сигналів для виконавчих механізмів.

Для обміну даними між модулем комп'ютерного зору та мікроконтролером може використовуватися послідовний інтерфейс. У програмній частині на Python доцільним є застосування бібліотеки pySerial, яка забезпечує доступ до послідовного порту та підтримує роботу на різних операційних системах [31]. Такий підхід дозволяє передавати до Arduino-сумісного контролера не зображення, а вже підготовлені результати розпізнавання, наприклад команди «STOP», «LEFT», «RIGHT», «FORWARD» або числові коди дій. Це зменшує обсяг передавання даних і спрощує програмну логіку на стороні мікроконтролера.

Загальну логіку добору програмних засобів можна подати так: Python використовується як основна мова реалізації програмного модуля комп'ютерного зору; OpenCV відповідає за роботу із зображеннями та відеопотоком; TensorFlow Lite або YOLO-модель виконує розпізнавання об'єктів; CVAT застосовується для розмітки навчальних даних; pySerial забезпечує обмін результатами розпізнавання з Arduino-сумісним контролером; Arduino IDE використовується для програмування низькорівневої логіки керування мобільним роботом.

На рисунку 2.6 показано взаємодію таких блоків: «Камера», «Python», «OpenCV», «Модель розпізнавання об'єктів», «Модуль передавання команд», «Arduino IDE/скетч мікроконтролера», «Двигуни та виконавчі механізми».

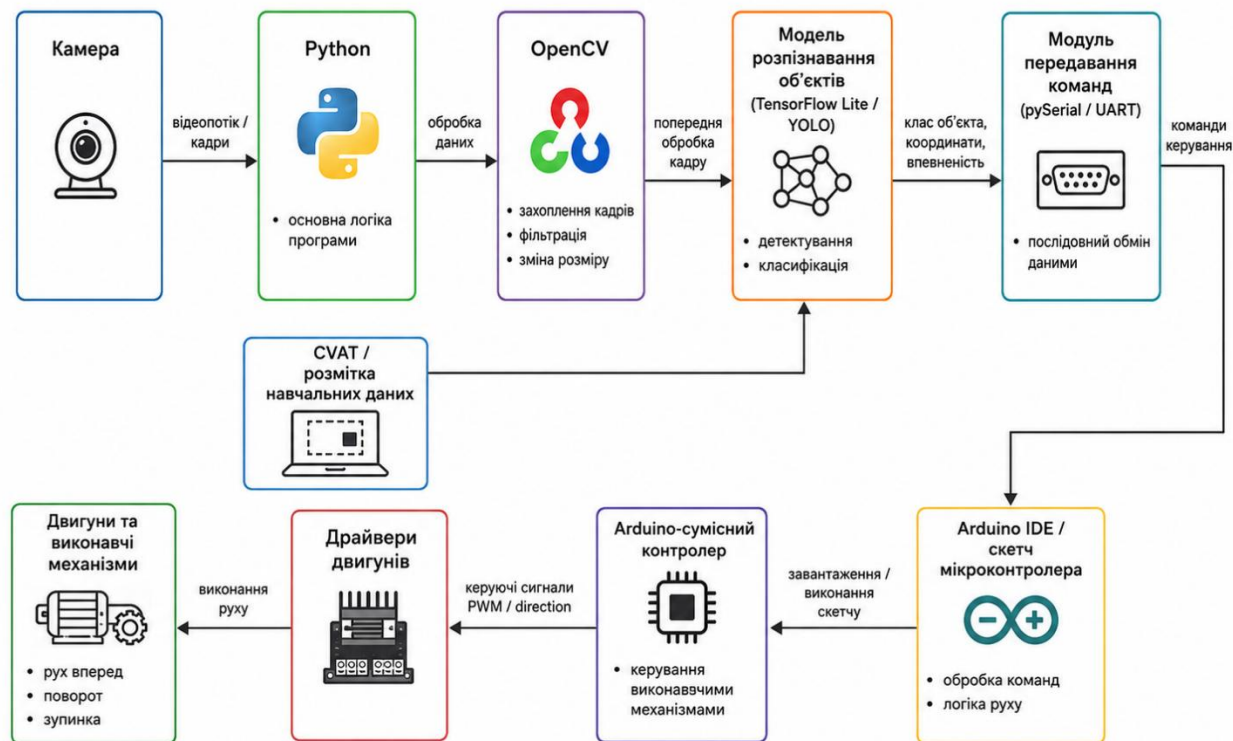


Рисунок 2.6 – Структура програмних засобів для реалізації комп'ютерного зору та керування роботизованою платформою

Для систематизації вибору програмних засобів доцільно подати їх у таблиці 2.3.

Таблиця 2.3 – Програмні засоби для реалізації комп'ютерного зору та машинного навчання

Програмний засіб	Призначення у роботі	Обґрунтування вибору
Python	Основна мова програмування модуля комп'ютерного зору	Простота реалізації, велика кількість бібліотек для обробки зображень і машинного навчання
OpenCV	Отримання та попередня обробка зображень	Підтримка роботи з камерами, відеопотоками, фільтрацією, перетворенням і аналізом кадрів

Продовження таблиці 2.3

Програмний засіб	Призначення у роботі	Обґрунтування вибору
TensorFlow Lite	Запуск або підготовка полегшеної моделі розпізнавання	Орієнтація на мобільні, вбудовані та edge-пристрої
YOLO-модель	Детектування об'єктів у відеопотоці	Висока швидкість виявлення об'єктів і придатність для задач реального часу
CVAT	Розмітка зображень для навчальної вибірки	Підтримка анотування даних для задач класифікації, детектування та сегментації
pySerial	Обмін даними між Python-модулем і Arduino	Передавання команд через послідовний порт
Arduino IDE	Програмування мікроконтролера	Завантаження скетчів, робота з бібліотеками та Serial Monitor

У процесі створення роботизованої платформи передбачено використання програмної архітектури з чітким розподілом функцій між окремими модулями. На рівні комп'ютерного зору Python-програма отримує кадр із камери, виконує попередню обробку за допомогою OpenCV, передає кадр до моделі розпізнавання та отримує результат у вигляді класу об'єкта, координат рамки та рівня впевненості. Після цього програмний модуль перетворює результат розпізнавання на просту команду для роботи.

На рівні мікроконтролера програма, створена в Arduino IDE, приймає команду через послідовний інтерфейс або інший канал зв'язку, аналізує її та виконує відповідну дію. Наприклад, якщо отримано команду «FORWARD», мікроконтролер формує сигнали для руху вперед; якщо отримано «STOP» – зупиняє двигуни; якщо отримано «LEFT» або «RIGHT» – змінює напрямок руху. Такий спосіб організації програмної частини дозволяє зменшити навантаження на мікроконтролер і забезпечити зрозумілу взаємодію між інтелектуальним модулем та апаратною платформою.

Вибір програмних засобів для реалізації комп'ютерного зору та машинного навчання ґрунтується на необхідності поєднання зручного середовища розробки, бібліотек обробки зображень, інструментів розпізнавання об'єктів і засобів зв'язку з мікроконтролером. Найбільш доцільним є використання Python, OpenCV, TensorFlow Lite або YOLO-моделі, CVAT, pySerial та Arduino IDE. Такий

набір програмних засобів забезпечує повний цикл роботи системи: від отримання зображення та розпізнавання об'єкта до формування команди й виконання руху роботизованою платформою.

2.4 Розробка алгоритму взаємодії модуля розпізнавання об'єктів із системою керування рухом

Розробка алгоритму взаємодії модуля розпізнавання об'єктів із системою керування рухом є важливим етапом проєктування роботизованої платформи, оскільки саме цей алгоритм забезпечує перехід від результатів комп'ютерного зору до конкретних дій мобільного робота. Якщо модуль комп'ютерного зору виконує аналіз зображення та визначає наявність об'єкта в кадрі, то система керування рухом повинна перетворити отриману інформацію на команди для двигунів, сервоприводів або інших виконавчих механізмів. Таким чином, алгоритм взаємодії поєднує високорівневу обробку візуальних даних із низькорівневим мікроконтролерним керуванням.

У межах проєктування роботизованої платформи доцільно передбачити програмну архітектуру, у якій завдання комп'ютерного зору, обробки результатів розпізнавання та керування рухом розподіляються між відповідними функціональними компонентами системи. Модуль комп'ютерного зору отримує зображення з камери, виконує попередню обробку кадру, передає його до моделі розпізнавання об'єктів і формує результат у вигляді класу об'єкта, координат рамки та рівня впевненості. Після цього результат аналізу перетворюється на спрощену команду, яка передається до Arduino-сумісного мікроконтролера через послідовний інтерфейс, Wi-Fi або інший канал зв'язку.

Основна ідея алгоритму полягає в тому, що мікроконтролер не обробляє саме зображення, а отримує вже готовий результат розпізнавання. Це дає змогу уникнути перевантаження мікроконтролера складними обчисленнями та забезпечити стабільне виконання задач реального часу. Наприклад, якщо модель комп'ютерного зору розпізнала об'єкт у центральній частині кадру, до

мікроконтролера може бути передана команда «FORWARD» або «STOP» залежно від логіки роботи системи. Якщо об'єкт розташований ліворуч або праворуч від центра кадру, система може сформувати команду повороту у відповідний бік.

Алгоритм взаємодії модуля розпізнавання об'єктів із системою керування рухом складається з таких основних етапів: отримання відеокадру з камери; попередня обробка зображення; розпізнавання об'єкта за допомогою моделі комп'ютерного зору; аналіз результатів розпізнавання; формування керуючої команди; передавання команди до мікроконтролера; виконання дії виконавчими механізмами; перевірка результату та перехід до наступного циклу роботи (рис. 2.7). Така послідовність забезпечує циклічну роботу системи та дозволяє роботу реагувати на зміну ситуації в полі зору камери.

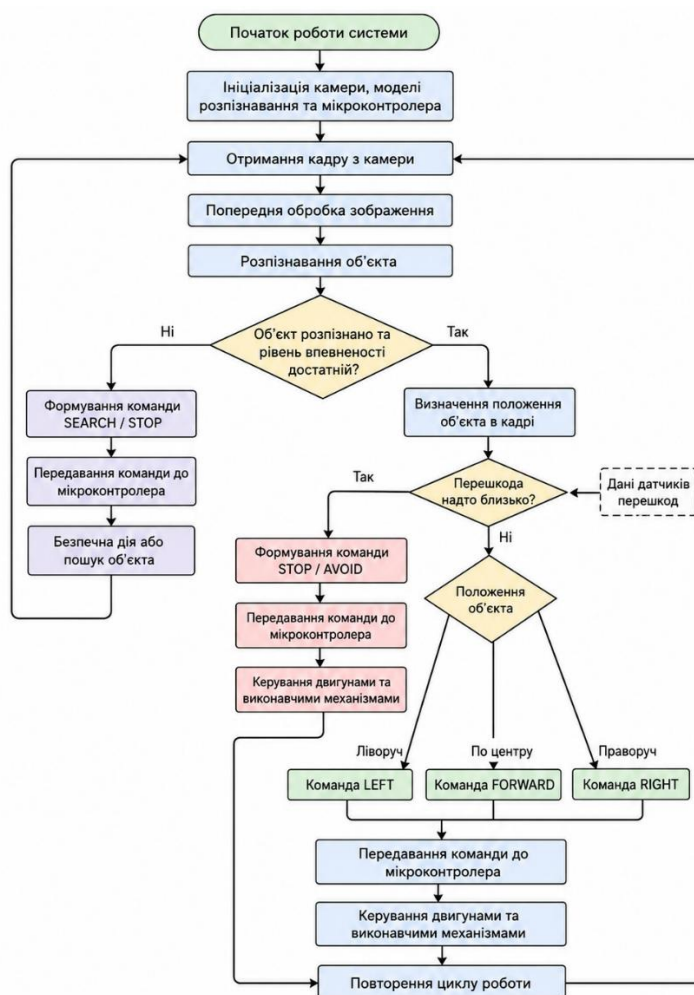


Рисунок 2.7 – Алгоритм взаємодії модуля розпізнавання об'єктів із системою керування рухом

Важливим елементом алгоритму є перевірка рівня впевненості розпізнавання. Модель комп'ютерного зору може формувати кілька результатів для одного кадру, тому система повинна обирати лише ті результати, які перевищують задане порогове значення. Наприклад, якщо рівень впевненості становить менше 0,5 або 50 %, розпізнавання можна вважати ненадійним і не використовувати його для прийняття рішення. Якщо ж рівень впевненості є достатнім, система переходить до аналізу положення об'єкта в кадрі та формування відповідної команди.

Для визначення команди руху може використовуватися положення об'єкта відносно центра зображення. Кадр умовно поділяється на три області: ліву, центральну та праву. Якщо об'єкт виявлено в лівій частині кадру, робот може виконати поворот ліворуч. Якщо об'єкт розташований у правій частині кадру, формується команда повороту праворуч. Якщо об'єкт знаходиться в центральній зоні, робот може рухатися вперед або зупинитися, залежно від призначення системи. Такий підхід є простим для реалізації та добре підходить для навчального прототипу мобільної роботизованої платформи.

Приклад логіки формування команд руху наведено в таблиці 2.4.

Таблиця 2.4 – Формування команд руху за результатами розпізнавання об'єктів

Умова розпізнавання	Аналіз ситуації	Команда для мікроконтролера	Дія роботизованої платформи
Об'єкт не виявлено	У кадрі відсутній цільовий об'єкт	SEARCH або STOP	Пошук об'єкта або зупинка
Об'єкт виявлено ліворуч	Центр рамки об'єкта зміщений у ліву частину кадру	LEFT	Поворот ліворуч
Об'єкт виявлено праворуч	Центр рамки об'єкта зміщений у праву частину кадру	RIGHT	Поворот праворуч
Об'єкт виявлено по центру	Об'єкт розташований у центральній частині кадру	FORWARD	Рух уперед
Об'єкт надто близько	Об'єкт займає значну частину кадру або датчик фіксує малу відстань	STOP	Зупинка
Виявлено перешкоду	Дані датчика підтверджують небезпечне наближення	AVOID	Об'їзд перешкоди

Як видно з таблиці 2.4, алгоритм керування може враховувати не лише результат комп'ютерного зору, а й додаткові дані від сенсорів. Це підвищує

надійність роботи платформи, оскільки візуальне розпізнавання може бути нестабільним за умов слабкого освітлення, розмиття кадру або часткового перекриття об'єкта. Наприклад, якщо камера розпізнала об'єкт у центральній частині кадру, але ультразвуковий датчик показує малу відстань до перешкоди, пріоритет повинна мати команда зупинки. Такий принцип дозволяє уникнути зіткнення та забезпечити безпечнішу поведінку робота.

У програмній реалізації модуль комп'ютерного зору може формувати команду у вигляді текстового рядка або числового коду. Наприклад, команда «F» може означати рух уперед, «L» – поворот ліворуч, «R» – поворот праворуч, «S» – зупинку, «A» – об'їзд перешкоди. Такий формат є зручним для передавання через послідовний порт і простим для обробки в Arduino-скетчі. Мікроконтролер приймає команду, порівнює її з відомими значеннями та викликає відповідну функцію керування двигунами.

На стороні мікроконтролера алгоритм має бути максимально простим і стабільним. Arduino-сумісний контролер повинен постійно перевіряти наявність нових команд, отриманих від модуля комп'ютерного зору. Після приймання команди мікроконтролер виконує відповідну функцію: рух уперед, поворот, зупинку або об'їзд перешкоди. Якщо команда не розпізнана або канал зв'язку тимчасово неактивний, доцільно передбачити безпечний режим, наприклад зупинку двигунів.

Особливу увагу потрібно приділити пріоритетності команд. У системі мобільного робота команди безпеки мають мати найвищий пріоритет. Наприклад, якщо датчик перешкод фіксує небезпечне наближення до об'єкта, команда зупинки повинна виконуватися навіть тоді, коли модуль комп'ютерного зору передає команду руху вперед. Це пояснюється тим, що сенсор відстані може швидше зафіксувати загрозу зіткнення, тоді як модуль комп'ютерного зору може мати затримку обробки кадру.

Пріоритети команд можна подати так: найвищий пріоритет має команда аварійної зупинки; далі – команда уникнення перешкоди; після цього – команди повороту за результатами розпізнавання; найнижчий пріоритет має команда руху

вперед. Така логіка дозволяє забезпечити передбачувану поведінку платформи та зменшити ризик помилкових дій.

У запропонованому алгоритмі важливо врахувати циклічність роботи системи. Мобільна платформа повинна постійно отримувати нові кадри, аналізувати зміну положення об'єкта та коригувати рух. Якщо об'єкт переміщується в кадрі, робот повинен змінювати напрямок руху відповідно до його нового положення. Якщо об'єкт зникає з поля зору, система може перейти в режим пошуку або зупинки. Якщо з'являється перешкода, пріоритет надається безпечному припиненню руху.

Практична реалізація такого алгоритму може бути виконана за допомогою Python-модуля комп'ютерного зору та Arduino-скетчу. Python-програма відповідає за отримання кадрів, розпізнавання об'єктів і передавання команд через pySerial або інший канал зв'язку [31]. Arduino-скетч, створений в Arduino IDE, приймає ці команди та керує двигунами через драйвери, використовуючи цифрові та PWM-сигнали [30]. Для обробки кадрів може застосовуватися OpenCV, а для розпізнавання – TensorFlow Lite або YOLO-модель [26 - 28].

Розроблений алгоритм взаємодії модуля розпізнавання об'єктів із системою керування рухом забезпечує узгоджену роботу програмного модуля комп'ютерного зору та мікроконтролерної платформи. Його основою є перетворення результатів розпізнавання на прості команди руху, які можуть бути швидко оброблені мікроконтролером. Запропонований підхід дозволяє поєднати інтелектуальне виявлення об'єктів із надійним керуванням виконавчими механізмами мобільного робота, а також забезпечує можливість подальшого розширення системи за рахунок додаткових сенсорів, режимів руху та складніших моделей розпізнавання.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ РОБОТИЗОВАНОЇ ПЛАТФОРМИ

3.1 Складання апаратної частини мобільного робота на базі Arduino

Практична реалізація роботизованої платформи передбачає складання апаратної частини мобільного робота на основі навчального набору ELEGOO Smart Robot Car Kit V4.0. Такий набір є зручним, оскільки містить основні компоненти, необхідні для створення колісної роботизованої платформи: шасі, двигуни постійного струму, колеса, Arduino-сумісну плату керування, модулі датчиків, сервопривід, ультразвуковий датчик, камеру, модулі з'єднання, кріпильні елементи та джерело живлення. Наявність готової компонентної бази дозволяє зосередитися не лише на механічному складанні, а й на подальшій інтеграції програмного керування, комп'ютерного зору та алгоритмів розпізнавання об'єктів.

Перед початком складання було виконано перевірку комплектації набору. На цьому етапі всі компоненти були розкладені на робочій поверхні та згруповані за функціональним призначенням: елементи шасі, двигуни й колеса, плата керування, сенсорні модулі, модуль камери, елементи живлення, кабелі та кріплення. Така підготовка є важливою, оскільки дає змогу уникнути помилок під час монтажу та спростити послідовність складання мобільного робота.

На рисунку 3.1 наведено основні елементи комплекту ELEGOO Smart Robot Car Kit V4.0, які використовуються для складання мобільного робота на базі Arduino. У складі набору представлено Arduino-сумісну плату ELEGOO UNO R3, плату розширення, модуль камери, ультразвуковий датчик HC-SR04, інфрачервоний пульт керування, модуль зв'язку, сервопривід, двигуни постійного струму, колеса, батарейний відсік, акрилові пластини шасі, монтажні стійки, гвинти, гайки та інші кріпильні елементи.

Зображені компоненти можна умовно поділити на кілька функціональних груп. До механічної частини належать пластини шасі, колеса, двигуни, монтажні стійки та кріплення, які формують основу рухомої платформи. До електронної

частини входять мікроконтролерна плата, плата розширення, сенсорні модулі, камера, сервопривід і модуль зв'язку. Окремо можна виділити елементи живлення та комутації, зокрема батарейний блок і з'єднувальні проводи.

Наявність повного набору механічних, електронних і монтажних компонентів дає змогу виконати поетапне складання роботизованої платформи без додаткового добору базових деталей. Попередній огляд комплектації є важливим підготовчим етапом, оскільки дозволяє перевірити наявність усіх необхідних елементів, визначити їхнє призначення та підготувати компоненти до подальшого монтажу ходової частини, плати керування, сенсорного блоку й модуля комп'ютерного зору.

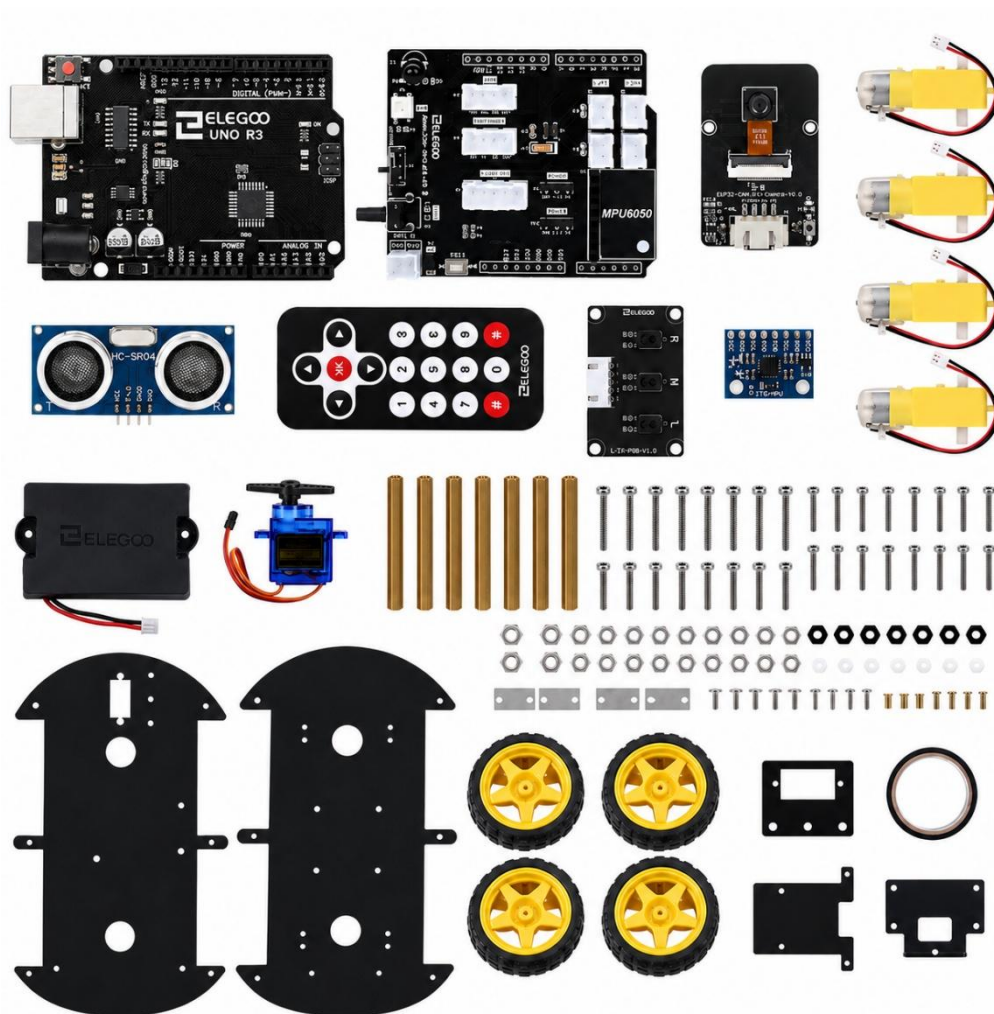


Рисунок 3.1 – Комплектація ELEGOO Smart Robot Car Kit V4.0 перед складанням мобільного робота

Першим етапом складання стало формування механічної основи роботизованої платформи. Для цього було використано нижню акрилову пластину шасі, до якої закріплювалися двигуни постійного струму. Двигуни встановлювалися у відповідні посадкові місця та фіксувалися за допомогою гвинтів і кріпильних елементів. Під час монтажу важливо було зберегти симетричне розташування двигунів, оскільки від цього залежить рівномірність руху платформи, коректність поворотів і стабільність переміщення робота по поверхні (рис. 3.2).

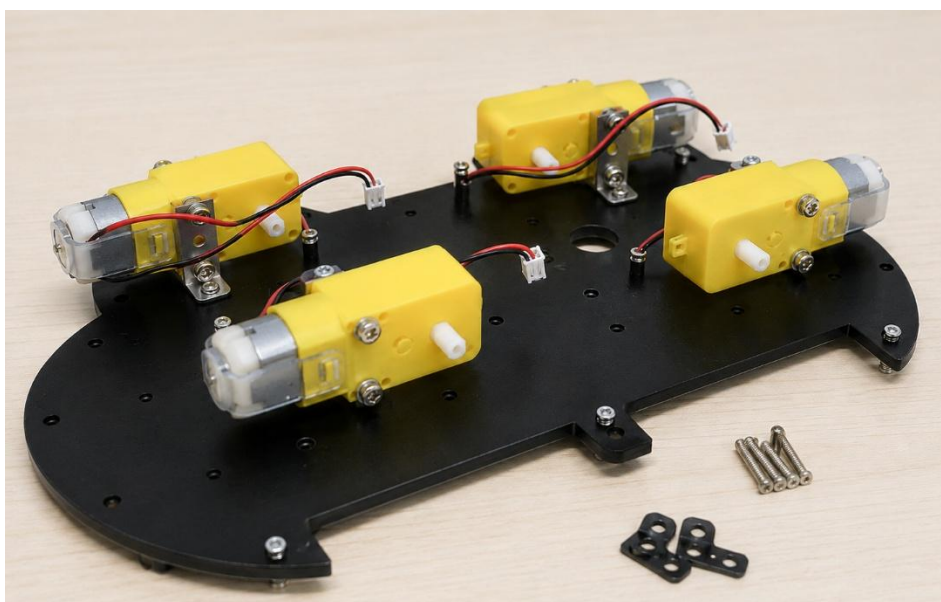


Рисунок 3.2 – Встановлення моторів

Після встановлення двигунів було виконано монтаж коліс. Колеса закріплювалися на валах двигунів таким чином, щоб забезпечити щільне з'єднання без перекосу. На цьому етапі перевірялося вільне обертання кожного колеса, відсутність механічних перешкод і правильність розташування двигунів відносно напрямку руху. У результаті було сформовано нижній ходовий вузол мобільного робота, який є основою для подальшого встановлення електронних компонентів (рис. 3.3)..



Рисунок 3.3 – Формування ходової частини мобільного робота

Наступним етапом стало встановлення верхньої частини шасі та монтаж вертикальних стійок. Стійки використовувалися для з'єднання нижньої та верхньої пластин, а також для створення простору між рівнями конструкції. Такий підхід дозволяє зручно розмістити проводи, двигуни, модулі живлення та плату керування. Верхня пластина виконує роль монтажної основи для розміщення Arduino-сумісного контролера, плати розширення, батарейного відсіку та допоміжних модулів.

Після завершення механічного складання було встановлено батарейний відсік. Його розміщення обиралося так, щоб не порушувати баланс мобільної платформи. Оскільки джерело живлення має помітну масу, його доцільно встановлювати ближче до центральної або задньої частини шасі. Це дозволяє зменшити зміщення центра ваги та підвищити стійкість робота під час руху, поворотів і зупинок.

На наступному етапі було встановлено Arduino-сумісну плату керування та плату розширення. Ці елементи є центральною частиною апаратної системи, оскільки саме через них здійснюється підключення двигунів, датчиків, сервоприводу, модуля камери та інших периферійних пристроїв. Плата закріплювалася на верхньому рівні шасі з використанням гвинтів і дистанційних

елементів. Таке кріплення забезпечує надійність монтажу та зменшує ризик механічного пошкодження електронних компонентів під час експлуатації робота.

Окрему увагу було приділено підключенню двигунів до плати керування. Проводи від двигунів під'єднувалися до відповідних роз'ємів, що забезпечують керування напрямком і швидкістю обертання коліс (рис. 3.4). Оскільки двигуни є силовими елементами системи, їх живлення та керування повинні бути організовані через драйвери, а не безпосередньо через виводи мікроконтролера. Це дозволяє захистити Arduino-сумісний контролер від перевантаження та забезпечити стабільну роботу ходової частини.

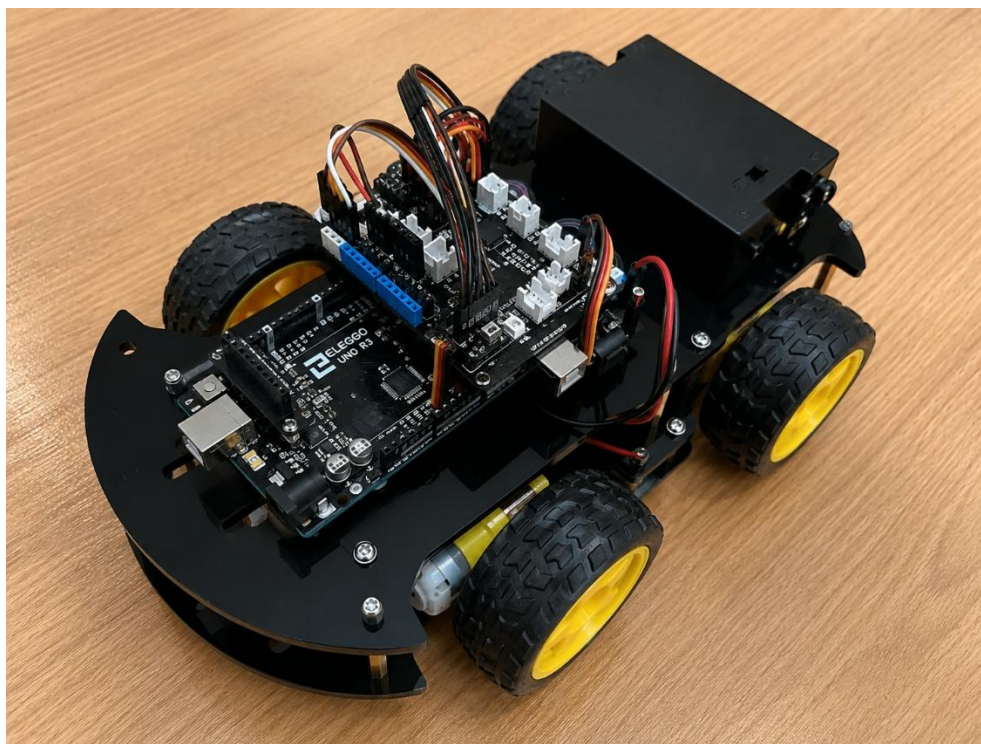


Рисунок 3.4 – Встановлення плати керування та підключення основних електронних модулів

Важливим етапом складання стало встановлення сенсорного блоку. До передньої частини роботизованої платформи було змонтовано ультразвуковий датчик відстані, який використовується для виявлення перешкод перед роботом. Ультразвуковий датчик доцільно розміщувати у фронтальній частині платформи, оскільки саме цей напрям є основним під час руху вперед. Його положення має

забезпечувати вільне поширення ультразвукового сигналу без перекриття корпусом, колесами або іншими елементами конструкції.

Для забезпечення можливості огляду простору перед роботом ультразвуковий датчик було встановлено разом із сервоприводом. Сервопривід дає змогу змінювати напрям вимірювання відстані, що може використовуватися для сканування простору ліворуч і праворуч від основної траєкторії руху. Такий підхід є доцільним для режиму уникнення перешкод, оскільки робот може не лише зупинятися перед об'єктом, а й оцінювати вільні напрями для подальшого руху.

Окрім ультразвукового датчика, у конструкції можуть використовуватися інфрачервоні датчики лінії або перешкод. Їх розміщення зазвичай передбачається в нижній частині шасі, ближче до поверхні, по якій рухається робот. Такі датчики дозволяють реалізувати режим слідування за лінією або додатково контролювати наявність об'єктів у безпосередній близькості до платформи. У межах цієї роботи вони можуть використовуватися як допоміжний сенсорний канал, що доповнює результати комп'ютерного зору.

Наступним етапом стало встановлення модуля камери. Камера розміщувалася у передній частині платформи над сенсорним блоком, що забезпечує огляд простору перед роботом. Таке розташування є доцільним для задач розпізнавання об'єктів, оскільки камера фіксує саме ту частину середовища, в напрямку якої рухається мобільна платформа. Під час монтажу камери важливо забезпечити її стійке положення, правильний кут нахилу та відсутність перекриття поля зору іншими елементами конструкції.

Після встановлення основних компонентів було виконано підключення кабелів. Проводи від двигунів, датчиків, сервоприводу та модуля камери під'єднувалися до відповідних роз'ємів плати керування. Для зменшення ризику помилок під час підключення враховувалося маркування роз'ємів, полярність живлення та призначення сигнальних ліній. Особливу увагу було приділено підключенню ліній живлення та GND, оскільки неправильне з'єднання може

призвести до нестабільної роботи модулів або пошкодження компонентів (рис. 3.5).

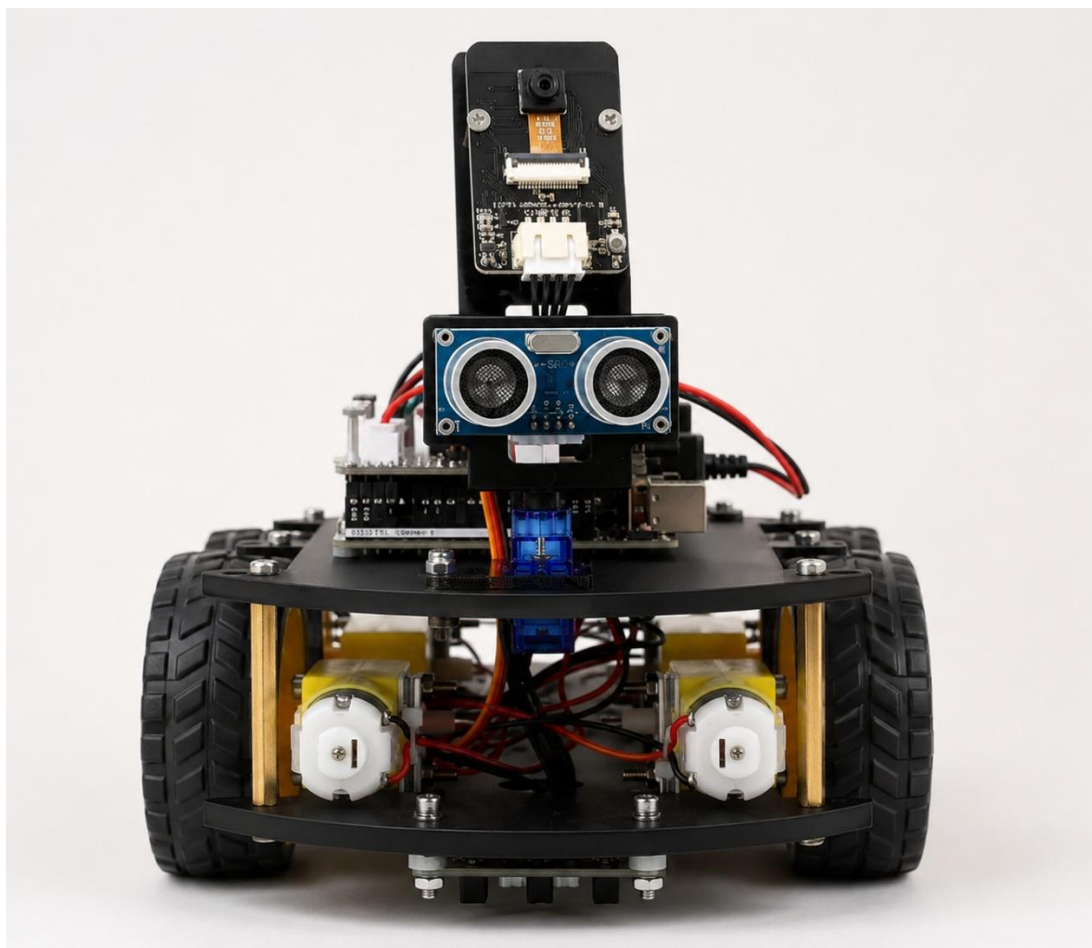


Рисунок 3.5 – Завершений вигляд мобільного робота після встановлення камери, сенсорів і плати керування

Після завершення монтажу було проведено первинну перевірку апаратної частини. На цьому етапі візуально контролювалася якість кріплення компонентів, правильність підключення проводів, надійність фіксації коліс, відсутність зайвого натягу кабелів і можливість вільного обертання рухомих елементів. Також перевірялося, чи не торкаються проводи коліс або двигунів, оскільки це може спричинити механічне пошкодження під час руху.

Основні етапи складання апаратної частини мобільного робота наведено в таблиці 3.1.

Таблиця 3.1 – Етапи складання апаратної частини мобільного робота на базі ELEGOO Smart Robot Car Kit V4.0

Етап складання	Зміст виконаних робіт	Очікуваний результат
Підготовка компонентів	Перевірка комплектації, групування деталей, підготовка інструментів	Компоненти готові до монтажу
Формування шасі	Встановлення двигунів на нижню пластину, монтаж коліс	Сформована ходова частина робота
Монтаж верхнього рівня	Встановлення стійок, верхньої пластини та батарейного відсіку	Отримано двоярусну механічну конструкцію
Встановлення плати керування	Закріплення Arduino-сумісного контролера та плати розширення	Підготовлено центральний вузол керування
Підключення двигунів	З'єднання двигунів із відповідними роз'ємами плати керування	Забезпечено можливість керування рухом
Монтаж сенсорного блоку	Встановлення ультразвукового датчика, сервоприводу та ІЧ-датчиків	Платформа отримує дані про навколишнє середовище
Встановлення камери	Монтаж модуля камери у передній частині робота	Забезпечено отримання візуальних даних
Підключення живлення	Під'єднання батарейного блоку та перевірка полярності	Платформа готова до автономної роботи
Первинна перевірка	Огляд кріплень, кабелів і рухомих частин	Підтверджено готовність до програмного тестування

Після складання апаратної частини роботизована платформа набуває завершеного конструктивного вигляду та може бути використана для наступних етапів роботи: завантаження програмного коду, тестування руху, перевірки датчиків, налаштування камери та інтеграції модуля розпізнавання об'єктів. На цьому етапі важливо не переходити одразу до складних режимів роботи, а виконати послідовну перевірку кожного апаратного вузла окремо. Спочатку перевіряється живлення, потім робота двигунів, після цього – сенсорні модулі, сервопривід і камера.

3.2 Налаштування модулів руху, датчиків перешкод, камери та бездротового керування

Після завершення складання апаратної частини мобільного робота на базі ELEGOO Smart Robot Car Kit V4.0 було виконано налаштування основних функціональних модулів платформи. До таких модулів належать система руху,

датчики виявлення перешкод, камера для отримання візуальної інформації та засоби бездротового керування. Налаштування цих компонентів є необхідним етапом практичної реалізації, оскільки саме від правильності їх підключення, програмної ініціалізації та перевірки залежить стабільність роботи всієї роботизованої системи.

Першим етапом було налаштування модулів руху. У конструкції мобільного робота використовуються двигуни постійного струму з редукторами, які забезпечують обертання коліс і переміщення платформи по поверхні. Керування двигунами здійснюється через плату керування та драйвери, які приймають керуючі сигнали від Arduino-сумісного мікроконтролера. Для перевірки роботи рухової частини було послідовно протестовано обертання кожного колеса, напрям руху платформи, виконання поворотів і зупинку двигунів.

На цьому етапі важливо було перевірити відповідність напрямку обертання коліс заданим командам. Якщо під час подавання команди руху вперед одне з коліс оберталося у протилежному напрямку, це свідчило про необхідність зміни логіки керування або коригування підключення двигуна. Після перевірки напрямку руху було протестовано основні команди: рух уперед, рух назад, поворот ліворуч, поворот праворуч і повна зупинка. Таке тестування дозволило переконатися, що ходова частина платформи працює стабільно та може виконувати базові маневри.

Наступним етапом стало налаштування датчиків перешкод. У складі ELEGOO Smart Robot Car Kit V4.0 для цього використовується ультразвуковий датчик HC-SR04, який дає змогу визначати відстань до об'єктів перед роботом. Принцип його роботи полягає у випромінюванні ультразвукового сигналу та вимірюванні часу його повернення після відбиття від перешкоди. На основі цього часу мікроконтролер обчислює відстань до об'єкта та може прийняти рішення про зупинку, поворот або об'їзд перешкоди.

Під час налаштування ультразвукового датчика було перевірено правильність підключення виводів живлення, «землі», Trig та Echo. Після цього

в Arduino IDE було завантажено тестову програму для вимірювання відстані. Отримані значення контролювалися через послідовний монітор. Для перевірки коректності роботи датчика перед роботом розміщувалися об'єкти на різній відстані, після чого аналізувалася зміна показників. Якщо значення відстані змінювалися відповідно до фактичного розташування перешкоди, датчик вважався налаштованим правильно.

Оскільки ультразвуковий датчик установлено на сервоприводі, було також виконано перевірку повороту сенсорного блоку. Сервопривід дозволяє змінювати напрям вимірювання відстані, що є важливим для реалізації алгоритму уникнення перешкод. У процесі тестування перевірялися положення ліворуч, по центру та праворуч. Такий підхід дає змогу мобільному роботу не лише виявляти

Окремим етапом було налаштування камери. Камера використовується для отримання зображення перед роботом і подальшого застосування алгоритмів комп'ютерного зору. Під час монтажу було перевірено її механічне закріплення, правильність орієнтації, кут нахилу та відсутність перекриття поля зору іншими елементами конструкції. Камера має бути спрямована вперед, оскільки саме в цьому напрямку робот виконує основний рух і має виявляти об'єкти.

Після встановлення камери було перевірено отримання зображення. Для цього використовувався програмний модуль, який дає змогу переглядати відеопотік або окремі кадри з камери. На цьому етапі оцінювалася чіткість зображення, правильність передавання кадру, стабільність роботи камери та достатність кута огляду. Якщо зображення було затемненим, розмитим або мало неправильний нахил, положення камери коригувалося механічно. Правильне налаштування камери є важливим, оскільки якість вхідного зображення безпосередньо впливає на точність подальшого розпізнавання об'єктів.

Для інтеграції камери з модулем комп'ютерного зору передбачено використання програмної обробки кадрів. Камера формує відеопотік, який передається до програмного середовища, де виконується попередня обробка зображення, виявлення об'єктів і формування результатів розпізнавання. Отриманий результат надалі може бути переданий до мікроконтролера у вигляді

простої команди руху. Таким чином, камера є вхідним пристроєм для інтелектуальної частини роботизованої платформи.

Наступним етапом було налаштування бездротового керування. У наборі ELEGOO Smart Robot Car Kit V4.0 передбачена можливість дистанційного керування роботом за допомогою бездротових засобів. Таке керування може використовуватися для ручного тестування платформи, перевірки команд руху, зміни режимів роботи та первинної діагностики. Бездротовий модуль забезпечує обмін даними між мобільним роботом і зовнішнім пристроєм, наприклад смартфоном, планшетом або комп'ютером.

Під час налаштування бездротового керування було перевірено підключення модуля зв'язку до плати керування, правильність живлення та стабільність передавання команд. Після цього виконувалася перевірка реакції робота на команди користувача. Зокрема, перевірялися команди руху вперед, назад, повороту ліворуч, повороту праворуч і зупинки. Така перевірка є важливою, оскільки дає змогу переконатися, що мікроконтролер правильно приймає керуючі сигнали та перетворює їх у дії виконавчих механізмів.

Для зручності перевірки основних модулів було сформовано послідовність налагодження, наведену в таблиці 3.2.

Таблиця 3.2 – Послідовність налаштування основних модулів мобільного робота

Модуль	Дії з налаштування	Очікуваний результат
Модулі руху	Перевірка напрямку обертання двигунів, руху вперед і назад, поворотів та зупинки	Робот стабільно виконує базові команди руху
Ультразвуковий датчик	Перевірка підключення Trig, Echo, 5 V, GND і вимірювання відстані	Датчик коректно визначає відстань до перешкоди
Сервопривід	Перевірка повороту вліво, вправо та в центральне положення	Сенсорний блок змінює напрям огляду
Камера	Перевірка механічного положення, кута огляду та отримання відеопотоку	Камера передає придатне для обробки зображення
Бездротовий модуль	Перевірка передавання команд із зовнішнього пристрою	Робот реагує на команди дистанційного керування
Плата керування	Перевірка приймання сигналів і формування керуючих команд	Мікроконтролер координує роботу всіх модулів

Після окремого налаштування кожного модуля було виконано комплексну перевірку їхньої взаємодії. На цьому етапі робот повинен був приймати команду керування, запускати двигуни, контролювати наявність перешкод за допомогою ультразвукового датчика та забезпечувати отримання зображення з камери. Така перевірка дозволила виявити можливі проблеми, пов'язані з неправильним підключенням проводів, нестабільним живленням або конфліктом між окремими модулями.

Особливу увагу було приділено стабільності живлення. Під час одночасної роботи двигунів, сервоприводу, камери та модуля зв'язку споживання струму зростає, що може призвести до просідання напруги. Унаслідок цього можливі збої в роботі мікроконтролера, перезавантаження плати або нестабільна робота сенсорів. Тому перед тестуванням складніших режимів було перевірено надійність підключення батарейного блоку, правильність полярності та стан з'єднувальних кабелів.

Під час налаштування також враховувалося, що команди від модуля комп'ютерного зору або бездротового керування мають бути простими для обробки мікроконтролером. Наприклад, для керування рухом можуть використовуватися короткі командні символи або текстові команди: «F» – рух вперед, «B» – рух назад, «L» – поворот ліворуч, «R» – поворот праворуч, «S» – зупинка. Такий формат спрощує програмну реалізацію та дозволяє швидко перевірити реакцію робота на отримані команди.

Загальну логіку налаштування модулів мобільного робота можна подати на рисунку 3.6.



Рисунок 3.6 – Послідовність налаштування модулів мобільного робота

Налаштування модулів руху, датчиків перешкод, камери та бездротового керування є необхідним етапом підготовки роботизованої платформи до роботи. У результаті було перевірено працездатність ходової частини, коректність вимірювання відстані, можливість повороту сенсорного блоку, отримання зображення з камери та приймання команд дистанційного керування. Виконане налаштування забезпечує готовність платформи до наступного етапу - реалізації програмного керування рухом і подальшої інтеграції з модулем розпізнавання об'єктів.

3.3 Реалізація програмного керування рухом роботизованої платформи в середовищі Arduino IDE

Реалізація програмного керування рухом роботизованої платформи є одним із ключових етапів практичної розробки мобільного робота на базі ELEGOO Smart Robot Car Kit V4.0. Після складання апаратної частини та підключення основних модулів необхідно забезпечити програмну взаємодію між Arduino-сумісним контролером, двигунами постійного струму, драйвером двигунів, сервоприводом, ультразвуковим датчиком і модулями керування. Саме програмний код визначає, як робот реагує на команди, змінює напрям руху, зупиняється перед перешкодою та виконує базові маневри.

Для програмування мікроконтролерної частини було використано середовище Arduino IDE, яке дозволяє створювати, редагувати, компілювати та завантажувати скетчі до Arduino-сумісної плати. Це середовище є зручним для реалізації навчальних і практичних робототехнічних проєктів, оскільки підтримує роботу з бібліотеками, послідовним монітором, цифровими та аналоговими входами-виходами, PWM-сигналами та зовнішніми модулями [30]. У межах цієї роботи Arduino IDE використано для створення програми, яка забезпечує керування рухом мобільної платформи, обробку даних із датчика перешкод і виконання команд, отриманих від користувача або модуля розпізнавання об'єктів.

На першому етапі програмної реалізації було визначено основні функції руху роботизованої платформи. До них належать рух уперед, рух назад, поворот ліворуч, поворот праворуч і повна зупинка. Кожна з цих дій реалізується через подавання відповідних логічних сигналів на входи драйвера двигунів. Для регулювання швидкості використовується PWM-сигнал, який дозволяє змінювати інтенсивність обертання двигунів. Такий підхід забезпечує гнучке керування платформою та дає змогу надалі інтегрувати програму з модулем комп'ютерного зору.

Для забезпечення зрозумілої структури програми програмний код було поділено на кілька логічних частин. У першій частині визначаються піни, до яких підключено драйвери двигунів, сервопривід і датчик перешкод. У другій частині виконується початкове налаштування режимів роботи пінів у функції `setup()`. У третій частині, яка реалізується у функції `loop()`, відбувається основний цикл роботи: зчитування команд, перевірка стану датчиків і виклик відповідних функцій руху.

У програмі передбачено окремі функції для кожного режиму руху. Це спрощує структуру коду та полегшує його подальше редагування. Наприклад, функція `moveForward()` відповідає за рух уперед, `moveBackward()` - за рух назад, `turnLeft()` - за поворот ліворуч, `turnRight()` - за поворот праворуч, а `stopMotors()` - за зупинку платформи. Такий підхід дає змогу уникнути дублювання коду та забезпечити зручне налагодження кожної дії окремо.

Приклад фрагмента програмного коду для керування рухом роботизованої платформи наведено на лістингу 3.1.

Лістинг 3.1 – Фрагмент програми керування рухом мобільного робота

```
#define ENA 5
#define IN1 6
#define IN2 7
#define IN3 8
#define IN4 9
#define ENB 10

char command;
```

```

void setup() {
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);

  Serial.begin(9600);
  stopMotors();
}

void loop() {
  if (Serial.available() > 0) {
    command = Serial.read();

    if (command == 'F') {
      moveForward();
    }
    else if (command == 'B') {
      moveBackward();
    }
    else if (command == 'L') {
      turnLeft();
    }
    else if (command == 'R') {
      turnRight();
    }
    else if (command == 'S') {
      stopMotors();
    }
  }
}

void moveForward() {
  analogWrite(ENA, 180);
  analogWrite(ENB, 180);

  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

void moveBackward() {
  analogWrite(ENA, 180);
  analogWrite(ENB, 180);

  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
}

```

```

    digitalWrite(IN4, HIGH);
}

void turnLeft() {
    analogWrite(ENA, 120);
    analogWrite(ENB, 180);

    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}

void turnRight() {
    analogWrite(ENA, 180);
    analogWrite(ENB, 120);

    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
}

void stopMotors() {
    analogWrite(ENA, 0);
    analogWrite(ENB, 0);

    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
}

```

кінець лістингу 3.1

У наведеному фрагменті програми піни ENA та ENB використовуються для керування швидкістю двигунів за допомогою PWM-сигналу, а піни IN1, IN2, IN3, IN4 - для визначення напрямку обертання двигунів. Команди надходять через послідовний порт і обробляються у функції loop(). Після отримання відповідного символу викликається функція, яка формує потрібну комбінацію сигналів на входах драйвера двигунів.

Окремо в програмній реалізації враховано можливість роботи з датчиком перешкод. Якщо ультразвуковий датчик фіксує об'єкт на небезпечній відстані, система повинна зупинити двигуни незалежно від отриманої команди руху.

Такий підхід підвищує безпечність роботи мобільної платформи та зменшує ризик зіткнення з перешкодами. У цьому випадку команда зупинки має вищий пріоритет, ніж команда руху вперед.

Для реалізації контролю відстані в програмі може використовуватися бібліотека NewPing або стандартне вимірювання часу проходження імпульсу через виводи Trig та Echo. Принцип роботи полягає в тому, що мікроконтролер надсилає короткий імпульс на вхід Trig, після чого вимірює тривалість сигналу на виході Echo. На основі цього часу обчислюється відстань до перешкоди. Якщо відстань менша за задане порогове значення, викликається функція `stopMotors()` або запускається алгоритм об'їзду.

Приклад логіки врахування перешкоди можна подати так: якщо відстань до об'єкта менша ніж 15-20 см, робот зупиняється; якщо перешкода відсутня, робот виконує команду, отриману від користувача або модуля розпізнавання. Такий підхід є достатнім для базового рівня автономності та може бути розширений у наступних етапах роботи.

Після написання програмного коду було виконано його компіляцію в Arduino IDE. На цьому етапі перевірялася відсутність синтаксичних помилок, правильність оголошення змінних, відповідність назв функцій і коректність підключення бібліотек. Після успішної компіляції скетч було завантажено до Arduino-сумісного контролера через USB-кабель. Для перевірки реакції робота використовувався Serial Monitor, через який надсилалися тестові команди руху.

Під час тестування було послідовно перевірено всі основні режими роботи. Спочатку подавалася команда S для перевірки зупинки двигунів. Після цього перевірялися команди F і B, які відповідають за рух уперед і назад. Далі перевірялися повороти ліворуч і праворуч. Якщо напрям руху не відповідав заданій команді, коригувалася логіка сигналів на входах драйвера або порядок підключення двигунів. Такий підхід дозволив забезпечити правильну відповідність між програмними командами та фактичною поведінкою платформи.

Реалізоване програмне керування рухом є базовим рівнем функціонування роботизованої платформи. Воно забезпечує виконання основних команд переміщення, зупинку, зміну напрямку руху та можливість реагування на перешкоди. Надалі цей програмний модуль може бути інтегрований із системою комп'ютерного зору, яка передаватиме до мікроконтролера команди залежно від результатів розпізнавання об'єктів.

3.4 Реалізація модуля комп'ютерного зору для розпізнавання об'єктів

Реалізація модуля комп'ютерного зору є важливим етапом створення роботизованої платформи, оскільки саме цей модуль забезпечує отримання візуальної інформації з камери, обробку кадрів, виявлення об'єктів у полі зору та формування результатів, які надалі можуть бути використані системою керування рухом. На відміну від звичайного сенсорного керування, де робот реагує лише на значення датчиків відстані або лінії, комп'ютерний зір дає змогу аналізувати складнішу інформацію про навколишнє середовище: форму, положення, розмір і тип об'єкта.

У межах роботи модуль комп'ютерного зору реалізується як окремий програмний компонент, який взаємодіє з камерою та мікроконтролерною платформою. Камера виконує роль джерела візуальних даних, програмний модуль здійснює обробку зображення та розпізнавання об'єктів, а Arduino-сумісний контролер отримує вже готовий результат у вигляді команди руху. Такий підхід є доцільним, оскільки мікроконтролер не має достатніх обчислювальних ресурсів для повноцінної обробки відеопотоку, але добре підходить для виконання керуючих команд, сформованих зовнішнім модулем.

Для реалізації модуля комп'ютерного зору доцільно використати мову програмування Python, бібліотеку OpenCV та модель розпізнавання об'єктів, побудовану на основі TensorFlow Lite або YOLO. Python використовується як основне середовище реалізації логіки програми, OpenCV забезпечує отримання кадрів із камери та попередню обробку зображень, а модель машинного навчання

виконує безпосереднє виявлення об'єктів у кадрі [26-28]. Така структура дозволяє розділити процес на окремі логічні етапи: отримання кадру, підготовка зображення, передавання його до моделі, аналіз результатів і формування команди для роботи.

Першим етапом роботи модуля є отримання зображення з камери. Камера, встановлена у передній частині роботизованої платформи, фіксує простір перед роботом і передає кадри до програмного модуля. Для стабільної роботи системи важливо, щоб камера була правильно зорієнтована, не мала перекриття поля зору іншими елементами конструкції та забезпечувала достатню якість зображення. Якщо кадр є надто темним, розмитим або має неправильний кут нахилу, якість розпізнавання об'єктів може суттєво знизитися.

Другим етапом є попередня обробка зображення. Вона передбачає приведення кадру до формату, зручного для подальшого аналізу. До таких дій належать зміна розміру зображення, нормалізація яскравості, перетворення колірного простору, фільтрація шумів і підготовка кадру до передавання в модель розпізнавання. Наприклад, для зменшення обчислювального навантаження кадр може бути зменшений до фіксованого розміру, який підтримує обрана модель. Це дозволяє прискорити обробку та забезпечити стабільнішу роботу в режимі реального часу.

Третім етапом є безпосереднє розпізнавання об'єктів. На цьому етапі попередньо підготовлений кадр передається до моделі машинного навчання. Модель аналізує зображення та формує результат, який може містити назву класу об'єкта, координати рамки навколо нього та рівень упевненості. Рівень упевненості показує, наскільки модель «впевнена» в тому, що виявлений об'єкт належить до певного класу. Для практичної роботи доцільно встановити порогове значення, наприклад 0,5 % або 50 %, нижче якого результат не використовується для прийняття рішень.

Приклад логіки перетворення результатів розпізнавання на команди керування наведено в таблиці 3.3.

Таблиця 3.3 – Перетворення результатів розпізнавання об'єктів на команди руху

Результат розпізнавання	Умова аналізу кадру	Команда для мікроконтролера	Дія робота
Об'єкт не виявлено	Відсутній об'єкт із достатнім рівнем упевненості	S або SEARCH	Зупинка або пошук об'єкта
Об'єкт ліворуч	Центр рамки об'єкта зміщений у ліву частину кадру	L	Поворот ліворуч
Об'єкт праворуч	Центр рамки об'єкта зміщений у праву частину кадру	R	Поворот праворуч
Об'єкт по центру	Об'єкт розташований у центральній частині кадру	F	Рух уперед
Об'єкт надто близько	Об'єкт займає значну частину кадру або підтверджено перешкоду	S	Зупинка

Як видно з таблиці 3.3, модуль комп'ютерного зору не передає до мікроконтролера складні дані у вигляді повного зображення. Натомість він формує коротку команду, яку Arduino-сумісний контролер може швидко обробити. Такий підхід спрощує взаємодію між програмним модулем розпізнавання та апаратною системою керування рухом.

Для визначення положення об'єкта в кадрі використовується координата центра рамки, яка формується моделлю розпізнавання. Кадр умовно поділяється на три зони: ліву, центральну та праву. Якщо центр рамки потрапляє в ліву зону, робот отримує команду повернути ліворуч. Якщо центр рамки знаходиться в правій зоні, робот повертає праворуч. Якщо об'єкт перебуває в центральній зоні, робот може продовжувати рух уперед. Такий спосіб керування є простим, зрозумілим і придатним для реалізації в навчальному прототипі мобільної платформи.

Алгоритм роботи модуля комп'ютерного зору можна подати у вигляді послідовності операцій. Спочатку виконується ініціалізація камери та завантаження моделі розпізнавання. Після цього система отримує кадр, виконує його попередню обробку та передає до моделі. Якщо об'єкт розпізнано з достатнім рівнем упевненості, визначається його положення в кадрі. На основі

цього положення формується команда руху, яка передається до Arduino-сумісного контролера. Якщо об'єкт не розпізнано або якість результату є недостатньою, система формує команду зупинки або переходить до наступного кадру.

Реалізація модуля комп'ютерного зору для розпізнавання об'єктів передбачає створення програмного блоку, який отримує кадри з камери, виконує їх попередню обробку, застосовує модель машинного навчання, визначає клас і положення об'єкта та формує команду для системи керування рухом. Запропонований підхід дозволяє поєднати можливості комп'ютерного зору з апаратною платформою на базі Arduino, не перевантажуючи мікроконтролер складними обчисленнями. У результаті роботизована платформа отримує можливість реагувати на об'єкти в полі зору камери та виконувати дії відповідно до результатів розпізнавання.

3.5 Інтеграція результатів розпізнавання об'єктів із командами керування рухом робота

Інтеграція результатів розпізнавання об'єктів із командами керування рухом є важливим етапом практичної реалізації роботизованої платформи, оскільки саме вона забезпечує перехід від аналізу зображення до фізичної дії мобільного робота. Модуль комп'ютерного зору визначає наявність об'єкта в кадрі, його положення та рівень упевненості розпізнавання, а система керування рухом перетворює ці результати на команди для двигунів. У результаті роботизована платформа отримує можливість не лише реагувати на команди користувача, а й змінювати свою поведінку відповідно до візуальної інформації про навколишнє середовище.

Модуль комп'ютерного зору виконує високорівневу обробку зображення. Він отримує кадр із камери, виконує попередню обробку, визначає об'єкт у полі зору та формує результат розпізнавання. Такий результат може містити назву класу об'єкта, координати рамки, у якій він виявлений, центральну точку об'єкта та значення впевненості моделі. Однак Arduino-сумісний контролер не потребує

передавання всіх цих даних у повному обсязі. Для керування рухом достатньо передавати спрощену команду, яка однозначно визначає подальшу дію робота.

Основним принципом інтеграції є перетворення результатів розпізнавання на короткі командні символи. Наприклад, команда F означає рух уперед, L - поворот ліворуч, R - поворот праворуч, S - зупинку, A - об'їзд перешкоди або перехід до безпечного режиму. Такий формат є зручним для передавання через послідовний порт, Bluetooth або Wi-Fi, оскільки не потребує значного обсягу даних і легко обробляється в Arduino-скетчі. У цьому випадку модуль комп'ютерного зору виконує складну частину аналізу, а мікроконтролер реалізує просту та стабільну логіку виконання команд.

Загальна логіка інтеграції полягає в тому, що зображення з камери аналізується програмним модулем, після чого система визначає положення об'єкта відносно центра кадру. Якщо об'єкт розташований ліворуч, робот повинен повернути ліворуч для вирівнювання напрямку руху. Якщо об'єкт розташований праворуч, формується команда повороту праворуч. Якщо об'єкт знаходиться в центральній частині кадру, робот може рухатися вперед. Якщо об'єкт не виявлено або рівень упевненості є недостатнім, платформа може зупинитися або перейти в режим очікування.

Для практичної реалізації інтеграції кадр умовно поділяється на три зони: ліву, центральну та праву. Такий поділ дає змогу спростити логіку керування рухом. Якщо центр рамки розпізнаного об'єкта знаходиться в лівій зоні кадру, формується команда L. Якщо центр рамки знаходиться в правій зоні, формується команда R. Якщо центр об'єкта потрапляє в центральну зону, формується команда F. У разі втрати об'єкта або низької впевненості розпізнавання формується команда S, що переводить робота в безпечний стан.

Для передавання команд до Arduino-сумісного контролера може використовуватися послідовний інтерфейс. У такому випадку програмний модуль комп'ютерного зору, реалізований мовою Python, надсилає до мікроконтролера один символ або короткий рядок. Arduino-скетч постійно перевіряє наявність нових даних у послідовному порту, зчитує команду та

викликає відповідну функцію керування двигунами. Такий підхід забезпечує просту взаємодію між програмним модулем розпізнавання та апаратною частиною робота.

Приклад фрагмента логіки передавання команди з Python-модуля до Arduino можна подати як лістинг 3.2.

Лістинг 3.2 – Фрагмент логіки формування та передавання команди до Arduino

```

if confidence < 0.5:
    command = "S"
else:
    object_center = (x1 + x2) / 2

    if object_center < frame_width / 3:
        command = "L"
    elif object_center > 2 * frame_width / 3:
        command = "R"
    else:
        command = "F"

serial_port.write(command.encode())

```

кінець лістингу 3.2

У наведеному фрагменті програмної логіки спочатку перевіряється рівень упевненості розпізнавання. Якщо він є нижчим за заданий поріг, формується команда зупинки. Якщо об'єкт розпізнано достатньо надійно, визначається координата його центра. Далі ця координата порівнюється з межами умовних зон кадру, після чого формується команда повороту ліворуч, повороту праворуч або руху вперед. Після цього команда передається до Arduino-сумісного контролера.

На стороні мікроконтролера команда обробляється в основному циклі програми. Arduino-сумісний контролер приймає символ, порівнює його з допустимими командами та викликає відповідну функцію руху. Якщо отримано команду F, робот рухається вперед; якщо L - повертає ліворуч; якщо R - повертає праворуч; якщо S - зупиняється. У разі отримання невідомої команди доцільно передбачити зупинку платформи як безпечну дію за замовчуванням.

Лістинг 3.3 – Фрагмент обробки команд руху в Arduino IDE

```
void loop() {  
  if (Serial.available() > 0) {  
    char command = Serial.read();  
  
    switch (command) {  
      case 'F':  
        moveForward();  
        break;  
  
      case 'L':  
        turnLeft();  
        break;  
  
      case 'R':  
        turnRight();  
        break;  
  
      case 'S':  
        stopMotors();  
        break;  
  
      case 'A':  
        avoidObstacle();  
        break;  
  
      default:  
        stopMotors();  
        break;  
    }  
  }  
}
```

кінець лістингу 3.3

У цьому фрагменті Arduino-скетчу реалізовано обробку команд, отриманих від модуля комп'ютерного зору. Конструкція switch дозволяє зручно розподілити дії залежно від отриманого символу. Функції moveForward(), turnLeft(), turnRight(), stopMotors() та avoidObstacle() відповідають за формування керуючих сигналів для двигунів і виконавчих механізмів. Така структура програми є зручною для подальшого розширення, оскільки до неї можна додати нові команди або режими роботи.

Важливою умовою інтеграції є пріоритетність команд. У роботизованій платформі команди безпеки повинні мати вищий пріоритет, ніж команди руху,

сформовані на основі розпізнавання об'єктів. Наприклад, якщо модуль комп'ютерного зору визначає, що об'єкт знаходиться по центру кадру та надсилає команду F, але ультразвуковий датчик фіксує малу відстань до перешкоди, система повинна виконати команду S або A. Це дозволяє уникнути зіткнення та підвищити надійність роботи мобільного робота.

З метою зменшення хибних реакцій доцільно використовувати фільтрацію команд. Наприклад, команда може вважатися дійсною лише тоді, коли однаковий результат розпізнавання повторюється протягом кількох послідовних кадрів. Якщо в одному кадрі об'єкт розпізнано ліворуч, а в наступному - праворуч, робот не повинен миттєво змінювати напрям руху. У такому випадку доцільно використовувати коротку затримку або перевірку стабільності результату. Це робить поведінку платформи більш плавною та передбачуваною.

Загальний алгоритм інтеграції результатів розпізнавання з керуванням рухом передбачає такі етапи: отримання кадру з камери, розпізнавання об'єкта, перевірка рівня впевненості, визначення положення об'єкта в кадрі, формування команди, перевірка даних датчика перешкод, передавання команди до мікроконтролера та виконання руху. Така послідовність дозволяє поєднати можливості комп'ютерного зору з базовими алгоритмами безпечного руху.

Під час практичного тестування інтеграції спочатку перевірялася передача команд без фактичного руху платформи. Для цього команди, сформовані Python-модулем, виводилися в консоль і паралельно надсилалися до Arduino. На наступному етапі перевірялася реакція двигунів на отримані команди. Після цього виконувалося тестування з реальним рухом робота, під час якого оцінювалася відповідність між положенням об'єкта в кадрі та поведінкою платформи.

Результатом інтеграції є система, у якій модуль комп'ютерного зору не лише розпізнає об'єкти, а й впливає на поведінку робота. Якщо об'єкт зміщується в кадрі, робот коригує напрям руху. Якщо об'єкт знаходиться перед роботом, платформа рухається вперед або зупиняється залежно від заданої логіки. Якщо

об'єкт не виявлено, система може перейти в режим очікування або пошуку. Така поведінка свідчить про реалізацію елементів інтелектуального керування.

ВИСНОВКИ

У кваліфікаційній роботі досягнуто поставленої мети, яка полягала в розробці роботизованої платформи з інтелектуальним розпізнаванням об'єктів на основі технологій комп'ютерного зору та машинного навчання. У процесі виконання роботи було поєднано апаратні засоби мобільної робототехніки, мікроконтролерне керування, сенсорні модулі, камеру та програмні алгоритми обробки візуальної інформації. Це дало змогу сформувати прототип системи, здатної виконувати рух, отримувати дані з навколишнього середовища, розпізнавати об'єкти та реагувати на результати аналізу.

У роботі проаналізовано теоретичні основи побудови мобільних роботизованих платформ. Встановлено, що сучасна мобільна роботизована платформа є комплексною апаратно-програмною системою, до складу якої входять механічна основа, мікроконтролерний блок, сенсорні модулі, виконавчі механізми, система живлення, модулі зв'язку та програмне забезпечення. З'ясовано, що ефективність такої системи залежить від узгодженої взаємодії всіх компонентів, а також від правильного розподілу функцій між апаратною та програмною частинами.

Дослідження принципів комп'ютерного зору показало, що ця технологія є важливим засобом підвищення автономності мобільних роботів. Комп'ютерний зір забезпечує отримання, попередню обробку та аналіз зображень або відеопотоків. У роботі розглянуто основні етапи розпізнавання об'єктів, зокрема отримання кадру з камери, попередню обробку зображення, виявлення об'єкта, визначення його класу, координат і рівня впевненості. Показано, що результати такого аналізу можуть бути використані для формування команд керування рухом мобільного робота.

У процесі виконання роботи розглянуто методи машинного навчання для класифікації та виявлення об'єктів на зображеннях. Встановлено, що для роботизованої платформи найбільш доцільними є моделі детектування об'єктів, які дають змогу не лише визначати клас об'єкта, а й встановлювати його

положення в кадрі. Особливу увагу приділено підходам на основі згорткових нейронних мереж, YOLO-моделей, TensorFlow Lite та інших засобів, які можуть використовуватися для створення модуля розпізнавання об'єктів у роботизованих системах.

Було визначено особливості використання мікроконтролерних платформ у мобільній робототехніці. Обґрунтовано, що Arduino-сумісний контролер доцільно застосовувати для керування двигунами, датчиками, сервоприводами та іншими виконавчими механізмами. Водночас встановлено, що класичний мікроконтролер має обмежені обчислювальні ресурси, тому не повинен виконувати повноцінну обробку відеопотоку та складні алгоритми машинного навчання. З огляду на це у роботі запропоновано розподілену архітектуру, у якій мікроконтролер відповідає за рух і взаємодію з апаратними компонентами, а модуль комп'ютерного зору – за аналіз зображень.

Аналіз існуючих роботизованих платформ з елементами інтелектуального керування дав змогу визначити переваги та обмеження різних апаратних рішень. Було встановлено, що сучасні платформи відрізняються рівнем складності, апаратною продуктивністю, вартістю, можливостями розширення та сферою застосування. Для виконання бакалаврської роботи найбільш доцільною визначено Arduino-сумісну навчальну мобільну платформу, оскільки вона є доступною, придатною для практичного складання, має зрозумілу структуру та дозволяє реалізувати базові режими руху з подальшою інтеграцією модуля комп'ютерного зору.

У роботі сформовано вимоги до апаратної та програмної частини роботизованої платформи. Апаратна частина повинна забезпечувати рух, підключення сенсорів, камери, системи живлення та модулів зв'язку. Програмна частина має забезпечувати керування двигунами, обробку команд, отримання даних із датчиків, аналіз відеопотоку, розпізнавання об'єктів і передавання результатів до мікроконтролера. Сформовані вимоги стали основою для подальшого проектування структури системи.

Розроблено структурну схему взаємодії основних компонентів системи. У структурі роботизованої платформи виділено мікроконтролерний блок, сенсорний блок, модуль комп'ютерного зору, драйвери двигунів, виконавчі механізми, модулі зв'язку та систему живлення. Також було розроблено принципову схему підключення компонентів, що дозволило визначити логіку електричних з'єднань між платою керування, двигунами, датчиками, сервоприводом, камерою та джерелом живлення.

Практична частина роботи передбачала реалізацію програмного керування рухом мобільного робота. У середовищі Arduino IDE було створено програмну логіку, яка забезпечує виконання базових команд руху: вперед, назад, ліворуч, праворуч і зупинка. Для керування двигунами використано сигнали, що подаються на драйвери двигунів, а для регулювання швидкості передбачено використання PWM-сигналів. Реалізоване програмне керування стало основою для подальшої інтеграції з результатами розпізнавання об'єктів.

У роботі налаштовано модуль комп'ютерного зору для розпізнавання об'єктів. Описано використання камери як джерела візуальних даних, програмного середовища для обробки зображень та моделі машинного навчання для визначення об'єктів у кадрі. Модуль комп'ютерного зору дає змогу отримувати кадри, виконувати їх попередню обробку, визначати об'єкт, його положення та рівень упевненості розпізнавання.

Особливу увагу приділено інтеграції результатів розпізнавання об'єктів із логікою керування роботизованою платформою. Результати аналізу зображення перетворюються на прості команди для мікроконтролера, зокрема рух уперед, поворот ліворуч, поворот праворуч або зупинку. Такий підхід дозволяє уникнути передавання великих обсягів даних до мікроконтролера та забезпечує стабільну взаємодію між модулем комп'ютерного зору і системою керування рухом.

Працездатність розробленої системи передбачено оцінювати через перевірку роботи ходової частини, правильність виконання команд руху, функціонування датчиків перешкод, роботу камери, стабільність передавання команд і реакцію платформи на результати розпізнавання об'єктів. Такий підхід

дозволяє комплексно оцінити як апаратну, так і програмну частину роботизованої платформи.

Проаналізовано теоретичні основи мобільної робототехніки, комп'ютерного зору та машинного навчання, обґрунтовано вибір апаратної платформи, розроблено структурні й принципові рішення, реалізовано програмне керування рухом і запропоновано спосіб інтеграції результатів розпізнавання об'єктів із командами керування роботом.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Computer Vision: Algorithms and Applications. 2nd ed. Cham : Springer, 2022. 925 p. URL: <https://link.springer.com/book/10.1007/978-3-030-34372-9> (date of access: 15.01.2026).
2. Computer Vision: A Modern Approach. 2nd ed. *Boston : Pearson*. URL: <https://eclass.teicrete.gr/modules/document/file.php/TP326/Forsyth%20D.%20Ponce%20J.%20Computer%20Vision%20A%20Modern%20Approach.pdf> (date of access: 17.01.2026).
3. Learning OpenCV: Computer Vision with the OpenCV Library. *Sebastopol : O'Reilly Media*. URL: <https://www.oreilly.com/library/view/learning-opencv/9780596516130/> (date of access: 19.01.2026).
4. OpenCV. *OpenCV Documentation*. URL: <https://docs.opencv.org/> (date of access: 21.01.2026).
5. OpenCV. Object Detection. *OpenCV Documentation*. URL: https://docs.opencv.org/3.4/d5/d54/group__objdetect.html (date of access: 23.01.2026).
6. OpenCV. Camera Calibration and 3D Reconstruction. *OpenCV Documentation*. URL: https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html (date of access: 25.01.2026).
7. Deep Learning. *Cambridge: MIT Press*, URL: <https://www.deeplearningbook.org/> (date of access: 27.01.2026).
8. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. *3rd ed. Sebastopol : O'Reilly Media*. URL: <https://www.oreilly.com/library/view/hands-on-machine-learning/9781098125967/> (date of access: 29.01.2026).
9. Deep learning. *Nature*. URL: <https://www.nature.com/articles/nature14539> (date of access: 31.01.2026).
10. ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*. URL: <https://dl.acm.org/doi/10.1145/3065386> (date of access: 02.02.2026).

11. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. URL: <https://arxiv.org/abs/1506.01497> (date of access: 04.02.2026).
12. You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. URL: <https://arxiv.org/abs/1506.02640> (date of access: 06.02.2026).
13. YOLO9000: Better, Faster, Stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. URL: <https://arxiv.org/abs/1612.08242> (date of access: 09.02.2026).
14. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv*. URL: <https://arxiv.org/abs/2004.10934> (date of access: 12.02.2026).
15. YOLOv10: Real-Time End-to-End Object Detection. *arXiv*. URL: <https://arxiv.org/abs/2405.14458> (date of access: 15.02.2026).
16. DETRs Beat YOLOs on Real-time Object Detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. URL: https://openaccess.thecvf.com/content/CVPR2024/html/Zhao_DETRs_Beat_YOLOs_on_Real-time_Object_Detection_CVPR_2024_paper.html (date of access: 18.02.2026).
17. Searching for MobileNetV3. *Proceedings of the IEEE/CVF International Conference on Computer Vision*. URL: <https://arxiv.org/abs/1905.02244> (date of access: 21.02.2026).
18. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition..* URL: <https://arxiv.org/abs/1801.04381> (date of access: 24.02.2026).
19. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the 36th International Conference on Machine Learning*. URL: <https://arxiv.org/abs/1905.11946> (date of access: 27.02.2026).
20. TensorFlow. *TensorFlow Documentation*. URL: <https://www.tensorflow.org/> (date of access: 02.03.2026).

21. Google AI Edge. *TensorFlow Lite*. URL: <https://ai.google.dev/edge/litert> (date of access: 05.03.2026).
22. Google AI Edge. *Object Detection with TensorFlow Lite Model Maker*. URL: https://ai.google.dev/edge/litert/libraries/modify/object_detection (date of access: 08.03.2026).
23. Ultralytics. *Ultralytics YOLO Documentation*. URL: <https://docs.ultralytics.com/> (date of access: 11.03.2026).
24. Python Software Foundation. *Python Documentation*. URL: <https://docs.python.org/3/> (date of access: 14.03.2026).
25. PySerial. *pySerial Documentation*. URL: <https://pyserial.readthedocs.io/> (date of access: 17.03.2026).
26. CVAT. *CVAT Documentation*. URL: <https://docs.cvat.ai/> (date of access: 20.03.2026).
27. Arduino. UNO R3. *Arduino Documentation*. URL: <https://docs.arduino.cc/hardware/uno-rev3> (date of access: 24.03.2026).
28. Arduino. *Arduino IDE Documentation*. URL: <https://docs.arduino.cc/software/ide/> (date of access: 28.03.2026).
29. ELEGOO. *ELEGOO Smart Robot Car Kit V4.0 Tutorial*. URL: <https://www.elegoo.com/blogs/arduino-projects/elegoo-smart-robot-car-kit-v4-0-tutorial> (date of access: 02.04.2026).
30. ROBOTIS. *TurtleBot3 Features. ROBOTIS e-Manual*. URL: <https://emanual.robotis.com/docs/en/platform/turtlebot3/features/> (date of access: 10.04.2026).
31. NVIDIA. *AI Robot Kits from NVIDIA JetBot Partners*. URL: <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetbot-ai-robot-kit/> (date of access: 20.04.2026).

