

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»**

**ІНТЕЛЕКТУАЛЬНА КВАДРОПЕДНА РОБОТИЗОВАНА
ПЛАТФОРМА НА БАЗІ RASPBERRY PI З ВИКОРИСТАННЯМ
LLM**

**INTELLIGENT QUADRUPED ROBOTIC PLATFORM BASED ON
RASPBERRY PI USING LLM**

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи KI-42
Поварчук Станіслав Вікторович

(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

Поварчуку Станіславу Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Інтелектуальна квадрокоптерна роботизована платформа на базі Raspberry Pi з використанням LLM

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

Аналіз теоретичних підходів до побудови квадрокоптерних роботизованих систем з елементами штучного інтелекту

Розробка апаратно-програмної структури квадрокоптерної роботизованої платформи

Практична реалізація та тестування інтелектуальної квадрокоптерної платформи

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Загальний вигляд робот-собаки SunFounder PiDog як фізичної моделі квадрокоптерної платформи

Етап складання кінцівок квадрокоптерної платформи

Розміщення сенсорних модулів на фізичній моделі

Послідовність налаштування Raspberry Pi та програмного середовища

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз теоретичних підходів до побудови квадроопедних роботизованих систем з елементами штучного інтелекту</i>	<i>Гринюк С.В., доцент</i>		
<i>Розробка апаратно-програмної структури квадроопедної роботизованої платформи</i>	<i>Гринюк С.В., доцент</i>		
<i>Практична реалізація та тестування інтелектуальної квадроопедної 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>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>Аналіз теоретичних підходів до побудови квадроопедних роботизованих систем з елементами штучного інтелекту</i>	до 02.03.2026 р.	
3.	<i>Розробка апаратно-програмної структури квадроопедної роботизованої платформи</i>	до 02.04.2026 р.	
4.	<i>Практична реалізація та тестування інтелектуальної квадроопедної 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)

Станіслав ПОВАРЧУК

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Сергій ГРИНЮК

(прізвище, ініціали)

АНОТАЦІЯ

Поварчук С. В. Інтелектуальна квадроопедна роботизована платформа на базі Raspberry Pi з використанням LLM. 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 інтелектуальних квадроопедних роботизованих платформ. Проаналізовано місце квадроопедних роботів у розвитку мобільної робототехніки, їхні переваги порівняно з колісними та гусеничними платформами, особливості кінематики, алгоритмів руху й координації сервоприводів.

У другому розділі обґрунтовано вибір апаратних компонентів інтелектуальної квадроопедної платформи. Розглянуто Raspberry Pi як центральний обчислювальний модуль системи, а також описано вибір сервоприводів, плати Robot HAT, камери, мікрофона, динаміка, сенсорних модулів і системи живлення. Розроблено структурну та принципову схеми підключення компонентів, що відображають взаємодію між обчислювальним, сенсорним, виконавчим і комунікаційним блоками платформи.

У третьому розділі описано практичну реалізацію квадроопедної роботизованої платформи на основі робот-собаки SunFounder PiDog. Розглянуто процес складання фізичної моделі, встановлення та налаштування Raspberry Pi, підключення периферійних модулів, калібрування сервоприводів і реалізацію програмного керування рухом. Також проведено тестування роботи платформи.

Ключові слова: система сигналізації, ультразвуковий датчик, Arduino Uno, безпека, охоронна система, мікроконтролер, сенсор, Arduino IDE.

ANNOTATION

Povarchuk S. Intelligent quadruped robotic platform based on Raspberry Pi using LLM. Manuscript.

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

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

The first section considers the theoretical foundations of building intelligent quadruped robotic platforms. The place of quadruped robots in the development of mobile robotics, their advantages compared to wheeled and tracked platforms, features of kinematics, motion algorithms, and servo coordination are analyzed.

The second section justifies the choice of hardware components of the intelligent quadruped platform. Raspberry Pi is considered as the central computing module of the system, and the selection of servos, Robot HAT board, camera, microphone, speaker, sensor modules and power supply system is also described. A structural and schematic diagram of the components is developed, reflecting the interaction between the computing, sensor, executive and communication blocks of the platform.

The third section describes the practical implementation of a quadruped robotic platform based on the SunFounder PiDog robot dog. The process of assembling a physical model, installing and configuring Raspberry Pi, connecting peripheral modules, calibrating servos and implementing software motion control is considered. The platform operation is also tested.

Keywords: alarm system, ultrasonic sensor, Arduino Uno, security, security system, microcontroller, sensor, Arduino IDE.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ТЕОРЕТИЧНИХ ПІДХОДІВ ДО ПОБУДОВИ КВАДРОПЕДНИХ РОБОТИЗОВАНИХ СИСТЕМ З ЕЛЕМЕНТАМИ ШТУЧНОГО ІНТЕЛЕКТУ	10
1.1 Квадропедні роботизовані системи як напрям розвитку мобільної робототехніки.....	10
1.2 Технічні можливості Raspberry Pi у задачах керування мобільними роботами.....	15
1.3 Особливості реалізації кінематики та алгоритмів руху квадропедної платформи	20
1.4 Перспективи інтеграції LLM у роботизовані системи для обробки команд і прийняття рішень.....	27
РОЗДІЛ 2 РОЗРОБКА АПАРАТНО-ПРОГРАМНОЇ СТРУКТУРИ КВАДРОПЕДНОЇ РОБОТИЗОВАНОЇ ПЛАТФОРМИ	34
2.1 Обґрунтування вибору Raspberry Pi як центрального обчислювального модуля системи	34
2.2 Вибір апаратних компонентів квадропедної платформи:	40
2.3 Розробка структурної та принципової схем інтелектуальної квадропедної роботизованої платформи.....	45
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ КВАДРОПЕДНОЇ ПЛАТФОРМИ.....	52
3.1 Складання фізичної моделі квадропедної роботизованої платформи.....	52
3.2 Налаштування Raspberry Pi, встановлення програмного середовища та підключення периферійних модулів.....	59
3.3 Реалізація програмного керування рухом квадропедного робота.....	64
3.4 Тестування роботи платформи	71
ВИСНОВКИ	76
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	79

ВСТУП

Сучасний розвиток комп'ютерної інженерії, робототехніки та штучного інтелекту сприяє активному впровадженню мобільних роботизованих систем у навчальну, дослідницьку, промислову та сервісну діяльність. Особливе місце серед таких систем займають квадрокоптерні роботизовані платформи, які імітують принципи пересування чотириногих живих організмів і можуть рухатися в умовах, де колісні або гусеничні роботи мають обмежену ефективність. Завдяки наявності кількох опорних кінцівок, сервоприводів, сенсорів і керуючих модулів такі платформи здатні виконувати складні рухові дії, адаптуватися до нерівностей поверхні та взаємодіяти з навколишнім середовищем.

Актуальність теми зумовлена необхідністю створення доступних апаратно-програмних рішень для побудови інтелектуальних роботизованих платформ, які можуть використовуватися для навчання, експериментальних досліджень, тестування алгоритмів керування та впровадження елементів штучного інтелекту. Використання Raspberry Pi як центрального обчислювального модуля дає змогу поєднати керування апаратними компонентами, обробку сенсорних даних, роботу з камерою, мережею та програмними сервісами. Така платформа є зручною для реалізації роботизованих систем, оскільки підтримує мову програмування Python, має достатню продуктивність для виконання прикладних алгоритмів і забезпечує можливість інтеграції з хмарними або локальними інтелектуальними сервісами.

Окремої уваги набуває використання великих мовних моделей у складі роботизованих систем. LLM можуть застосовуватися для обробки текстових або голосових команд користувача, формування відповідей, інтерпретації запитів і вибору відповідних сценаріїв поведінки робота. У поєднанні з Raspberry Pi, сенсорними модулями та виконавчими механізмами це дає можливість створити більш гнучку систему взаємодії між людиною і роботизованою платформою. У такому випадку робот не лише виконує заздалегідь запрограмовані команди, а й може реагувати на запити користувача у більш природній формі.

Метою кваліфікаційної роботи є розробка та дослідження інтелектуальної квадропедної роботизованої платформи на базі Raspberry Pi з використанням великої мовної моделі для організації взаємодії з користувачем та керування окремими сценаріями роботи системи.

Для досягнення поставленої мети в роботі необхідно вирішити такі завдання:

- проаналізувати особливості побудови квадропедних роботизованих платформ та напрями їх застосування;
- розглянути можливості використання Raspberry Pi у складі мобільних роботизованих систем;
- дослідити принципи керування рухом квадропедного робота та координації роботи сервоприводів;
- обґрунтувати вибір апаратних компонентів для створення роботизованої платформи;
- розробити структурну та принципову схеми взаємодії основних компонентів системи;
- запропонувати алгоритм роботи платформи з урахуванням сенсорних даних, команд користувача та взаємодії з LLM;
- реалізувати програмне забезпечення для керування рухом, обробки команд і виконання базових сценаріїв роботи;
- провести тестування працездатності платформи та проаналізувати отримані результати;

Об'єктом дослідження є процес побудови та функціонування мобільних квадропедних роботизованих платформ з елементами інтелектуального керування.

Предметом дослідження є апаратно-програмні засоби реалізації квадропедної роботизованої платформи на базі Raspberry Pi з використанням великої мовної моделі для обробки команд користувача та формування сценаріїв поведінки робота.

У роботі використано методи аналізу науково-технічних джерел, порівняння апаратних компонентів, структурного моделювання, алгоритмізації, програмної реалізації, експериментального тестування та узагальнення результатів. Застосування цих методів дає змогу обґрунтувати вибір технічних рішень, сформулювати архітектуру системи та перевірити її працездатність у практичних умовах.

Практичне значення роботи полягає у можливості використання розробленої кваліфікованої роботизованої платформи як навчального та експериментального прототипу для дослідження принципів мобільної робототехніки, керування сервоприводами, взаємодії з сенсорними модулями та інтеграції LLM у роботизовані системи. Запропоноване рішення може бути застосоване під час виконання лабораторних робіт, студентських проєктів, демонстраційних занять і подальших досліджень у сфері інтелектуальних робототехнічних систем.

РОЗДІЛ 1

АНАЛІЗ ТЕОРЕТИЧНИХ ПІДХОДІВ ДО ПОБУДОВИ КВАДРОПЕДНИХ РОБОТИЗОВАНИХ СИСТЕМ З ЕЛЕМЕНТАМИ ШТУЧНОГО ІНТЕЛЕКТУ

1.1 Квадропедні роботизовані системи як напрям розвитку мобільної робототехніки

Квадропедні роботизовані системи є одним із перспективних напрямів розвитку мобільної робототехніки, оскільки вони поєднують можливості автономного переміщення, адаптації до складного рельєфу, сенсорного сприйняття навколишнього середовища та інтелектуального керування. На відміну від колісних або гусеничних платформ, квадропедні роботи використовують чотири опорні кінцівки, що дає змогу реалізовувати більш гнучкі способи руху, зокрема ходьбу, повороти на місці, подолання нерівностей, стабілізацію корпусу та зміну положення центра мас. Саме тому такі системи активно досліджуються в контексті мобільної робототехніки, сервісних роботів, інспекційних платформ, навчальних робототехнічних комплексів і автономних систем взаємодії з людиною [1, 2]. У сучасних оглядах підкреслюється, що розвиток квадропедних роботів пов'язаний не лише з удосконаленням механічної конструкції, а й із розвитком методів керування, сенсорної інтеграції, машинного навчання та роботизованої автономності.

З технічної точки зору квадропедна роботизована платформа являє собою складну мехатронну систему, до складу якої входять несуча рама, чотири кінцівки з кількома ступенями свободи, сервоприводи або електродвигуни, контролери руху, джерело живлення, сенсорні модулі та обчислювальний блок. Кожна кінцівка виконує роль окремого виконавчого механізму, координована робота якого забезпечує переміщення всієї платформи. Для базових навчальних або експериментальних моделей зазвичай використовують сервоприводи, оскільки вони дають змогу керувати кутовим положенням ланок ноги та реалізовувати прості алгоритми ходи. У складніших промислових і

дослідницьких системах застосовуються високоточні електроприводи з датчиками положення, моменту та зворотного зв'язку, що забезпечує динамічну стабілізацію платформи під час руху [1, 3].

Основною перевагою квадропедної роботизованої системи є здатність пересуватися в середовищах, де традиційні мобільні платформи мають обмеження. Колісний робот ефективний на рівній поверхні, однак його можливості знижуються під час руху по сходах, нерівному ґрунту, перешкодах або поверхнях зі змінним нахилом. Гусенична платформа має кращу прохідність, проте поступається у гнучкості маневрування, точності постановки опор і можливості імітувати біомеханічні принципи руху. Квадропедний робот, навпаки, може змінювати положення кожної ноги окремо, утримувати рівновагу за рахунок розподілу опорних точок і адаптувати траєкторію руху залежно від стану поверхні. У промислових зразках, наприклад у роботі Spot, така концепція застосовується для переміщення різними типами місцевості, автономного обходу перешкод, самостійного відновлення положення після падіння та виконання інспекційних завдань [4].

Загальну роль квадропедних роботизованих систем у структурі мобільної робототехніки подано на рисунку 1.1.



Рисунок 1.1 – Місце квадропедних роботизованих систем у розвитку мобільної робототехніки [4]

Особливістю квадроопедних платформ є те, що їхній рух не зводиться лише до переміщення виконавчих механізмів. Для забезпечення стабільної ходи необхідно розв'язувати комплекс задач, пов'язаних із кінематикою, динамікою, балансуванням і синхронізацією приводів. Кінематика визначає взаємозв'язок між кутами повороту суглобів і положенням стопи в просторі, тоді як динамічна модель враховує масу корпусу, інерційні характеристики, контакт із поверхнею та зміну навантаження на опорні кінцівки. У простих навчальних моделях часто використовують наперед задані шаблони ходи, наприклад послідовне переміщення ніг або симетричні рухи пар кінцівок. У більш складних системах застосовуються алгоритми зворотної кінематики, регулятори положення, методи машинного навчання та адаптивні алгоритми керування [1, 2].

Квадроопедна роботизована система також є зручною платформою для інтеграції сенсорних модулів. До таких модулів можуть належати ультразвукові або інфрачервоні датчики відстані, камери, інерціальні вимірювальні модулі, мікрофони, датчики освітленості, температури, нахилу та інші засоби сприйняття середовища. Сенсорна інформація дає змогу роботі не лише виконувати механічне переміщення, а й аналізувати простір навколо себе, виявляти перешкоди, реагувати на команди користувача та формувати сценарії поведінки. Саме перехід від простої механічної платформи до сенсорно насиченої системи є важливим етапом розвитку мобільної робототехніки, оскільки робот починає функціонувати як кіберфізична система, у якій фізичний рух поєднується з обробкою даних і програмним прийняттям рішень [2, 3].

У межах роботи особливий інтерес становить використання Raspberry Pi як центрального обчислювального модуля квадроопедної платформи. Raspberry Pi має достатні обчислювальні можливості для запуску Python-програм, роботи з камерою, мережею, сенсорними модулями та зовнішніми сервісами. Наприклад, Raspberry Pi 4 Model B містить 64-бітний чотириядерний процесор, бездротові інтерфейси Wi-Fi і Bluetooth, Gigabit Ethernet, USB-порти, 40-контактний GPIO-роз'єм та інтерфейс CSI для підключення камери [5]. Новіші моделі Raspberry Pi 5 мають вищу продуктивність і можуть бути використані для складніших

сценаріїв обробки даних, зокрема комп'ютерного зору, локальної обробки команд або взаємодії з хмарними інтелектуальними сервісами [6].

Структурно квадрокоптерний робот можна розглядати як систему, у якій центральний обчислювальний модуль приймає дані від сенсорів, обробляє команди користувача, формує керуючі сигнали та передає їх на виконавчі механізми. Якщо в систему інтегрується велика мовна модель, то вона може виконувати роль інтелектуального програмного рівня, який інтерпретує природномовні команди, формує відповідь користувачу або обирає відповідний сценарій поведінки робота. У сучасних дослідженнях LLM розглядаються як інструмент для покращення взаємодії людини з роботом, планування дій, інтерпретації команд і підтримки автономного прийняття рішень [7, 8]. Водночас у роботизованих системах LLM не повинна безпосередньо замінювати низькорівневі контури керування приводами, оскільки стабілізація руху, безпека та синхронізація сервоприводів потребують детермінованих алгоритмів і

Принципову структуру квадрокоптерної роботизованої платформи з елементами інтелектуального керування подано на рисунку 1.2.

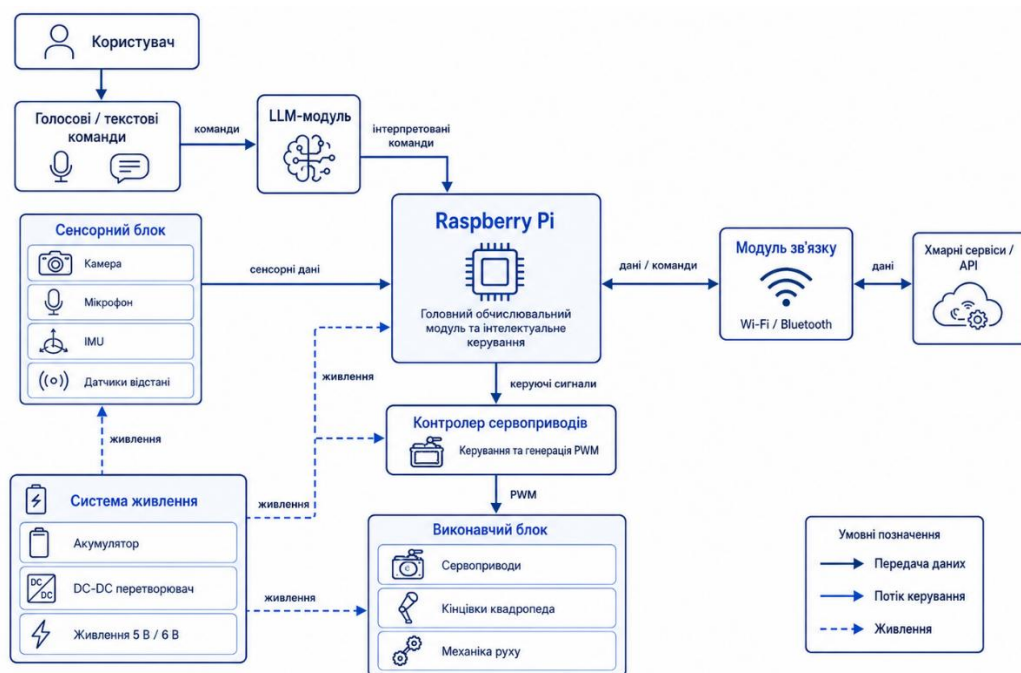


Рисунок 1.2 – Структура квадрокоптерної роботизованої платформи з інтелектуальним керуванням [8]

Розвиток квадроопедних роботизованих систем також пов'язаний із концепцією втіленого штучного інтелекту, коли алгоритми ШІ взаємодіють не лише з текстовими або цифровими даними, а й із фізичним середовищем через сенсори та виконавчі механізми. У цьому контексті квадроопедна платформа є придатним об'єктом для навчальних і дослідницьких завдань, оскільки вона демонструє зв'язок між програмним рішенням і реальною фізичною дією. Наприклад, команда користувача «рухайся вперед», «зупинись», «поверни праворуч» або «оглянь простір перед собою» може бути перетворена програмним рівнем на набір конкретних дій: аналіз команди, перевірку стану сенсорів, вибір алгоритму ходи, формування сигналів для сервоприводів і контроль виконання руху. Такий підхід дає змогу розглядати квадроопедного робота не лише як механічний пристрій, а як інтегровану апаратно-програмну систему з елементами інтелектуальної поведінки [7, 8].

Для бакалаврської кваліфікаційної роботи квадроопедна роботизована платформа має важливу навчальну й інженерну цінність. Вона дає змогу поєднати декілька напрямів комп'ютерної інженерії: мікрокомп'ютерні системи, програмування мовою Python, роботу з GPIO-інтерфейсами, керування сервоприводами, обробку сенсорних даних, комп'ютерний зір, мережеву взаємодію та інтеграцію інтелектуальних сервісів. На відміну від суто програмних проєктів, така система потребує врахування реальних апаратних обмежень: напруги живлення, споживання струму, затримок передавання сигналів, механічного навантаження на приводи, стійкості конструкції та стабільності програмного керування. Саме тому розробка квадроопедної платформи є комплексним завданням, яке відповідає змісту підготовки фахівців із комп'ютерної інженерії.

Квадроопедні роботизовані системи є важливим напрямом розвитку мобільної робототехніки, оскільки вони забезпечують вищу адаптивність до середовища, мають широкі можливості сенсорної інтеграції та можуть використовуватися як основа для впровадження елементів штучного інтелекту. У межах цієї роботи квадроопедна платформа на базі Raspberry Pi розглядається як

апаратно-програмний прототип, який поєднує механічну мобільність, сенсорне сприйняття, програмне керування та взаємодію з LLM. Це створює підґрунтя для подальшого обґрунтування вибору компонентів, побудови структурної схеми, розробки алгоритмів руху та реалізації практичного прототипу роботизованої системи.

1.2 Технічні можливості Raspberry Pi у задачах керування мобільними роботами

Raspberry Pi є одноплатним комп'ютером, який широко використовується у навчальних, експериментальних і прикладних робототехнічних системах завдяки поєднанню компактних розмірів, достатньої обчислювальної продуктивності, підтримки операційних систем Linux, наявності інтерфейсів введення-виведення та можливості роботи з мовою програмування Python. У задачах керування мобільними роботами Raspberry Pi виконує роль центрального обчислювального вузла, який приймає дані від сенсорів, обробляє команди користувача, формує логіку поведінки робота та передає керуючі сигнали на виконавчі модулі. На відміну від простих мікроконтролерів, Raspberry Pi може виконувати більш складні програмні задачі: обробку зображень, мережеву взаємодію, запуск локальних серверів, роботу з API, елементи комп'ютерного зору та інтеграцію з інтелектуальними програмними сервісами [9, 10]. Офіційна документація Raspberry Pi підтверджує наявність апаратних і програмних засобів, необхідних для побудови таких прикладних систем, зокрема GPIO, USB, Ethernet, Wi-Fi, Bluetooth, інтерфейсів камери та дисплея.

У контексті квадропедної роботизованої платформи Raspberry Pi доцільно розглядати не як безпосередній силовий контролер сервоприводів, а як головний керуючий комп'ютер, що координує роботу окремих апаратних модулів. Це важливо, оскільки рух квадропедного робота потребує одночасного керування кількома сервоприводами, обробки даних з інерціального модуля, камери або датчиків відстані, а також формування послідовностей руху для кінцівок.

Raspberry Pi може виконувати високорівневу логіку: вибір режиму руху, обробку команди «йти вперед», «повернути», «зупинитися», «оглянути простір» або «повернутися у вихідне положення». Водночас генерацію стабільних PWM-сигналів для великої кількості сервоприводів доцільно передавати окремому контролеру, наприклад модулю PCA9685 або іншому сервоконтролеру. Такий розподіл функцій підвищує стабільність системи, оскільки Raspberry Pi зосереджується на обчисленнях і логіці керування, а спеціалізований контролер забезпечує точніше формування сигналів для виконавчих механізмів.

Однією з ключових переваг Raspberry Pi є наявність 40-контактного GPIO-роз'єму, який забезпечує підключення зовнішніх електронних компонентів. Через GPIO можна організувати взаємодію з датчиками, кнопками, світлодіодами, релейними модулями, контролерами сервоприводів, I2C- та SPI-пристроями. Для мобільного робота це означає можливість створення модульної апаратної структури, у якій до центрального обчислювального вузла підключаються сенсори орієнтації, датчики відстані, модулі керування живленням, плати розширення та комунікаційні засоби. Наприклад, Raspberry Pi 4 Model B має 40-контактний GPIO-роз'єм, інтерфейс камери MIPI CSI, USB-порти, Gigabit Ethernet, Wi-Fi та Bluetooth, що дозволяє використовувати його як основу для роботизованих систем з сенсорним і мережевим керуванням [10].

Важливим аспектом використання Raspberry Pi у мобільній робототехніці є підтримка камер і засобів комп'ютерного зору. Наявність інтерфейсу CSI дає змогу підключати камеру безпосередньо до плати, а USB-порти можуть використовуватися для зовнішніх вебкамер або додаткових периферійних пристроїв. У квадрокоптерній платформі камера може застосовуватися для виявлення перешкод, розпізнавання об'єктів, спостереження за середовищем, трансляції відео або реалізації простих алгоритмів навігації. У поєднанні з бібліотеками Python, OpenCV та іншими програмними засобами Raspberry Pi може виконувати базову обробку зображень, що розширює функціональність робота порівняно з платформами, які орієнтовані лише на механічне переміщення (рис 1.3).

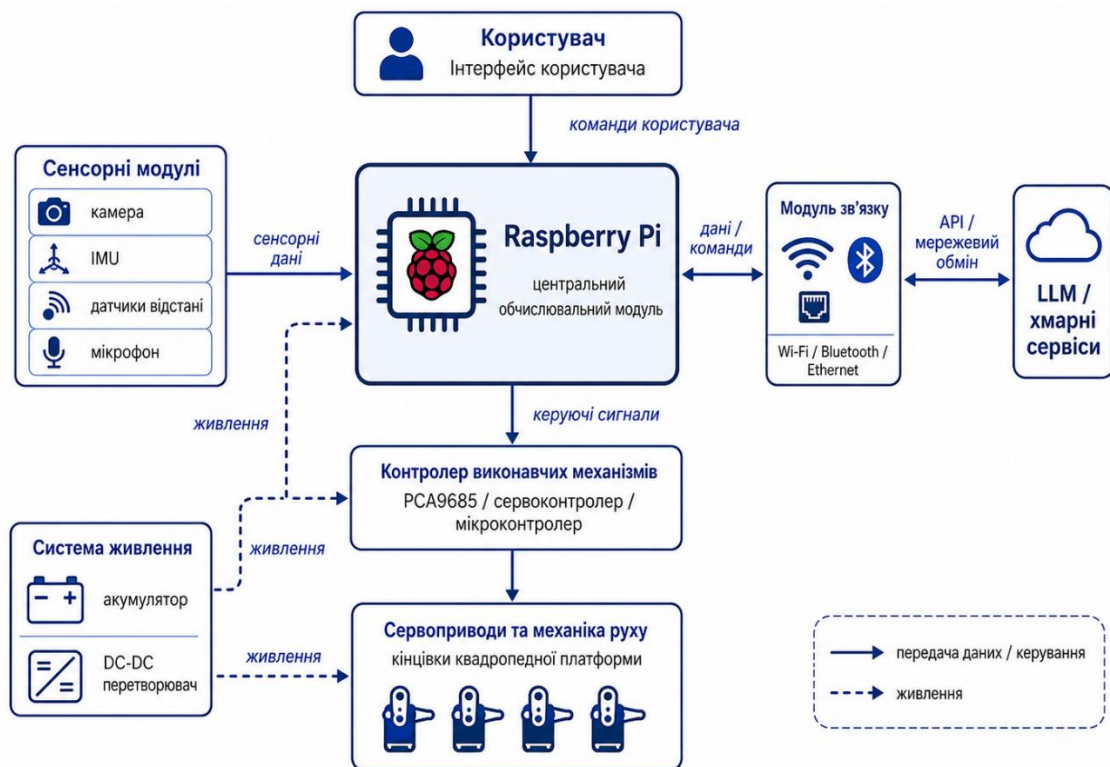


Рисунок 1.3 – Raspberry Pi у структурі керування мобільним роботом [10]

Для задач мобільної робототехніки важливе значення має також мережевий потенціал Raspberry Pi. Наявність Wi-Fi, Bluetooth та Ethernet дає змогу реалізувати дистанційне керування роботом, передавання телеметрії, оновлення програмного забезпечення, взаємодію з мобільним застосунком або вебінтерфейсом. У межах теми кваліфікаційної роботи це особливо важливо, оскільки інтеграція великої мовної моделі може здійснюватися через хмарний API або локальний серверний компонент. У такій архітектурі Raspberry Pi може отримувати команду користувача, передавати її до LLM-модуля, отримувати інтерпретовану відповідь і перетворювати її на конкретну дію робота. Наприклад, природномовна команда «підійди вперед і зупинись перед перешкодою» може бути перетворена на послідовність операцій: аналіз команди, перевірка відстані до об'єкта, вибір алгоритму ходи, запуск руху та зупинка після досягнення заданої умови.

З погляду продуктивності Raspberry Pi 4 і Raspberry Pi 5 мають достатні апаратні можливості для реалізації різних рівнів керування мобільним роботом.

Raspberry Pi 4 Model B оснащений 64-бітним чотириядерним процесором ARM Cortex-A72, підтримує до 8 ГБ оперативної пам'яті залежно від модифікації, має Wi-Fi, Bluetooth, Gigabit Ethernet, USB 3.0, USB 2.0, GPIO, CSI та DSI - інтерфейси [10]. Raspberry Pi 5, згідно з офіційними матеріалами, побудований на 64-бітному чотириядерному процесорі Arm Cortex-A76 з частотою 2,4 ГГц і забезпечує суттєве зростання продуктивності порівняно з Raspberry Pi 4, що робить його більш придатним для задач обробки зображень, роботи з кількома процесами та складніших сценаріїв інтелектуального керування [9].

Однак Raspberry Pi має не лише переваги, а й певні обмеження, які потрібно враховувати під час розробки мобільного робота. По-перше, плата не призначена для прямого живлення потужних сервоприводів, тому силові елементи виконавчого блоку повинні мати окрему стабілізовану лінію живлення. По-друге, Raspberry Pi працює під керуванням операційної системи загального призначення, тому не завжди забезпечує жорсткий режим реального часу. Це означає, що критичні за часом задачі, наприклад точна генерація PWM для великої кількості сервоприводів, краще покладати на окремий контролер. По - третє, під час роботи з камерою, мережею та алгоритмами обробки даних необхідно враховувати тепловиділення, споживання енергії та стабільність живлення. Для квадропедної платформи ці фактори є важливими, оскільки нестабільне живлення або перевантаження процесора може призвести до затримок у виконанні команд і некоректної роботи сервоприводів.

У роботизованих проєктах Raspberry Pi також може використовуватися разом із ROS 2, що дає змогу застосовувати готові програмні підходи до побудови роботизованих систем. ROS 2 є набором програмних бібліотек та інструментів для створення роботизованих застосунків, а офіційна документація ROS зазначає підтримку ARM-процесорів, зокрема використання 64-бітного середовища на Raspberry Pi як рекомендованого варіанта для простішого встановлення пакетів [11]. Для бакалаврської роботи повне використання ROS 2 не є обов'язковим,

однак його наявність демонструє, що Raspberry Pi може бути інтегрований у сучасну екосистему робототехнічного програмного забезпечення (рис. 1.4).

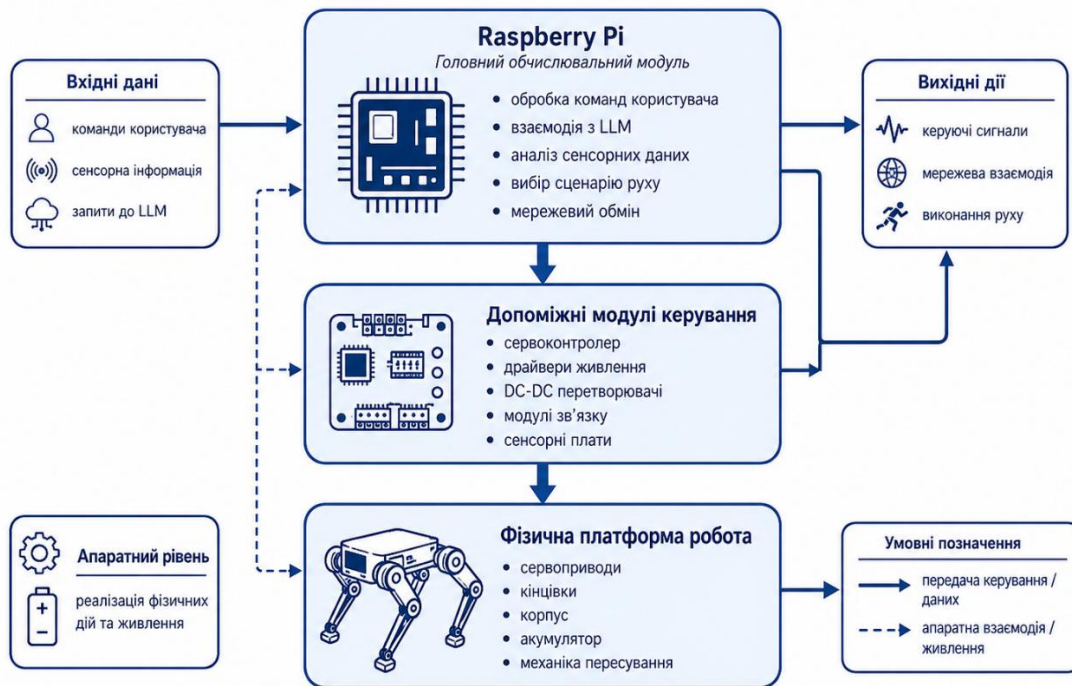


Рисунок 1.4 – Схема розподілу функцій між Raspberry Pi та допоміжними модулями [11]

У задачах інтелектуального керування Raspberry Pi може виконувати роль проміжного вузла між фізичною платформою і програмним рівнем штучного інтелекту. Якщо LLM працює у хмарному середовищі, Raspberry Pi передає текстову команду через API, отримує інтерпретовану відповідь і запускає відповідний локальний алгоритм. Якщо використовується локальна легка модель або окремий AI-прискорювач, частина обробки може виконуватися безпосередньо на пристрої. У будь-якому випадку важливо розмежовувати рівні керування: LLM доцільно використовувати для інтерпретації команд, вибору сценарію та формування відповідей користувачу, тоді як безпосереднє керування рухом має виконуватися за допомогою перевірених алгоритмів, які враховують стан сенсорів, положення кінцівок, обмеження сервоприводів і вимоги безпеки.

Отже, Raspberry Pi має достатні технічні можливості для використання як центрального обчислювального модуля мобільної квадропедної роботизованої

платформи. Його перевагами є підтримка Linux, Python, GPIO, камер, мережевих інтерфейсів, зовнішніх модулів і сучасних програмних бібліотек. У межах розроблюваної системи Raspberry Pi доцільно використовувати для обробки сенсорних даних, взаємодії з користувачем, інтеграції LLM, формування високорівневих команд і координації роботи допоміжних модулів. При цьому задачі точного керування сервоприводами, стабілізації живлення та формування PWM-сигналів потрібно покладати на спеціалізовані апаратні вузли, що забезпечує надійність і передбачуваність роботи квадропедної роботизованої платформи.

1.3 Особливості реалізації кінематики та алгоритмів руху квадропедної платформи

Кінематика квадропедної роботизованої платформи є основою для формування її руху, оскільки саме вона визначає взаємозв'язок між положенням окремих ланок кінцівок, кутами повороту сервоприводів і просторовим положенням опорних точок робота. На відміну від колісних мобільних платформ, де рух переважно описується швидкістю обертання коліс, квадропедний робот потребує узгодженого переміщення чотирьох кінцівок. Кожна кінцівка виконує роль окремого виконавчого механізму, який має певну кількість ступенів свободи та забезпечує постановку опори в задану точку. Саме тому реалізація руху квадропедної платформи передбачає розв'язання задач прямої кінематики, зворотної кінематики, планування траєкторії кінцівки та синхронізації фаз руху між усіма ногами робота [12, 13]. Сучасні огляди з локомоції квадропедних роботів підкреслюють, що планування ходи, кінематичне моделювання та керування рухом залишаються ключовими технічними викликами для таких систем.

У найпростішому випадку кінцівка квадропедного робота може мати два або три ступені свободи. Два ступені свободи дають змогу реалізувати базове піднімання та переміщення ноги вперед або назад, тоді як три ступені свободи

забезпечують гнучкішу зміну положення кінцівки у просторі. Для навчальної або експериментальної платформи на базі Raspberry Pi доцільним є використання сервоприводів, оскільки вони дозволяють задавати кутове положення кожного суглоба й реалізовувати повторювані шаблони руху. Наприклад, одна кінцівка може містити сервопривід для горизонтального відведення, сервопривід для піднімання стегнової частини та сервопривід для зміни положення нижньої частини ноги. У такій структурі кожна нога має власний набір кутів, а загальна кількість керованих приводів для чотиринової платформи може становити 8 або 12 сервоприводів залежно від складності конструкції.

Загальну структуру кінематичної моделі кінцівки квадроопедної платформи доцільно подати на рисунку 1.5.

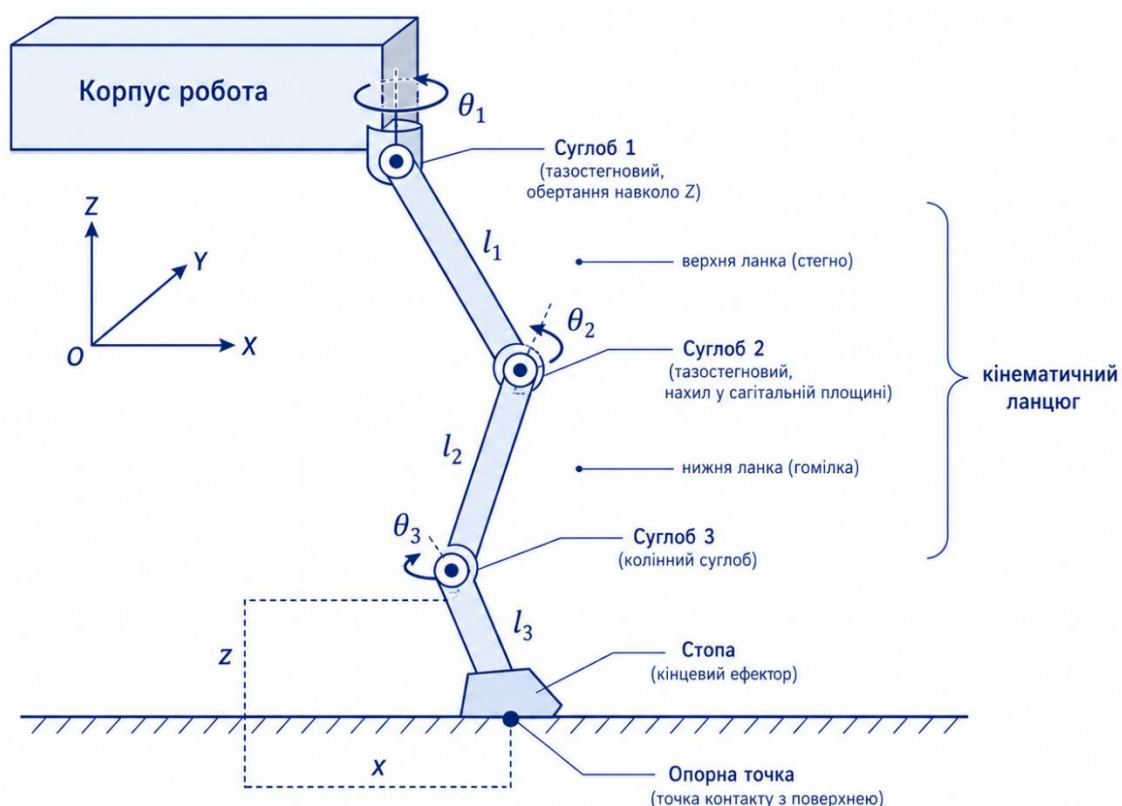


Рисунок 1.5 – Кінематична схема кінцівки квадроопедної платформи [13]

Пряма кінематика використовується для визначення положення кінцевої точки ноги на основі відомих кутів сервоприводів. Тобто, якщо відомі кути θ_1 , θ_2

і θ_3 , а також довжини ланок кінцівки, можна розрахувати координати точки опори відносно корпусу робота. Такий підхід є корисним для аналізу поточного положення ноги та перевірки того, чи не виходить кінцівка за межі допустимої робочої зони. У практичній реалізації це важливо для запобігання механічним перевантаженням, зіткненню ланок між собою або встановленню сервоприводів у недопустимі положення.

Зворотна кінематика має ще більше значення для керування квадропедною платформою, оскільки вона дозволяє визначити, які саме кути сервоприводів потрібно встановити, щоб стопа робота опинилася у заданій точці простору. Наприклад, якщо алгоритм руху визначає, що передня права нога має переміститися вперед на певну відстань і піднятися над поверхнею, система повинна перетворити цю бажану координату стопи на конкретні кути сервоприводів. Саме зворотна кінематика дає змогу перейти від високорівневого опису руху до низькорівневих керуючих сигналів для сервоприводів. У сучасних робототехнічних рішеннях зворотна кінематика часто поєднується з плануванням траєкторій, стабілізацією корпусу та зворотним зв'язком від сенсорів [13, 14]. У відкритих реалізаціях квадропедних роботів на Raspberry Pi також застосовують аналітичну зворотну кінематику та генерацію ходи на рівні Python-програм.

Окремим завданням є планування траєкторії руху кінцівки. Під час ходи кожна нога квадропедного робота проходить дві основні фази: фазу опори та фазу перенесення. У фазі опори кінцівка контактує з поверхнею та бере участь в утриманні маси робота. У фазі перенесення нога відривається від поверхні, переміщується вперед або вбік і готується до наступної постановки. Для плавності руху траєкторія стопи може задаватися не прямолінійно, а за допомогою дугоподібної, циклоїдальної або поліноміальної кривої. Це дозволяє уникати різких рухів, зменшувати навантаження на сервоприводи та робити ходу робота більш стабільною [13, 15]. У дослідженнях з керування квадропедами також використовуються криві Безьє та інші математичні засоби формування плавної траєкторії ноги під час руху.

Основу алгоритмів руху квадропедної платформи становить поняття ходи. Хо́да визначає послідовність переміщення ніг, часові інтервали між фазами руху та кількість кінцівок, які одночасно перебувають в опорі. Для простих роботизованих платформ найчастіше застосовують статичну або квазістатичну ходу, оскільки вона забезпечує вищу стійкість і простіша для програмної реалізації. У такому режимі робот у кожний момент часу зберігає достатню кількість точок опори, а проєкція центра мас залишається всередині опорного багатокутника. Це дає змогу зменшити ризик перекидання платформи, хоча швидкість руху при цьому є нижчою. Динамічні типи ходи, наприклад риссю або бігом, забезпечують вищу швидкість, але потребують точнішої синхронізації приводів, стабілізації корпусу та обробки даних від інерціальних сенсорів [12, 14]. Новіші огляди з локомоції квадропедних роботів також виділяють поєднання класичних методів планування ходи з оптимізаційними та навчальними підходами.

Основні типи ходи квадропедної платформи можна подати у вигляді таблиці 1.1.

Таблиця 1.1 – Основні типи ходи квадропедної роботизованої платформи

Тип ходи	Принцип реалізації	Переваги	Обмеження
Статична хода	Ноги переміщуються послідовно, більшість кінцівок залишаються в опорі	Висока стійкість, проста реалізація	Низька швидкість руху
Квазістатична хода	Центр мас контролюється так, щоб залишатися в межах опорного контуру	Добра стійкість на малих швидкостях	Потребує узгодження положення корпусу
Хо́да риссю	Діагональні пари ніг працюють синхронно	Вища швидкість, природніший рух	Потребує точнішої координації
Динамічна хода	Частина фаз руху виконується з короткочасним зменшенням кількості опор	Швидке переміщення, адаптація до активного руху	Вища складність стабілізації

Для розроблюваної квадропедної платформи на базі Raspberry Pi доцільно використовувати квазістатичний алгоритм ходи як базовий. Такий підхід є оптимальним для бакалаврської роботи, оскільки він дозволяє забезпечити

достатню стійкість, реалізувати керування сервоприводами без складних динамічних моделей і поступово доповнювати систему сенсорним зворотним зв'язком. Алгоритм може працювати за циклічним принципом: вибір напрямку руху, визначення послідовності переміщення ніг, розрахунок координат стоп, обчислення кутів сервоприводів, передавання PWM-сигналів на контролер і перевірка стану сенсорів. Якщо на шляху виявлено перешкоду або отримано команду зупинки, руховий цикл переривається, а платформа переходить у безпечний стан (рис. 1.6).

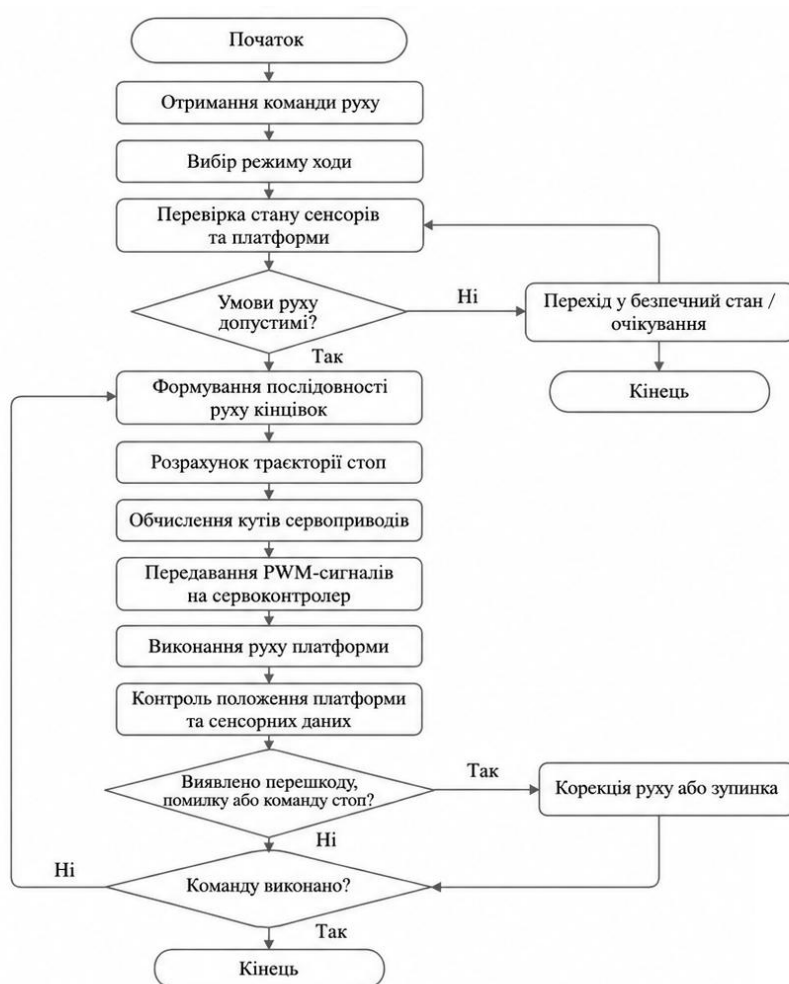


Рисунок 1.6 – Загальний алгоритм формування руху квадропедної платформи

У програмній реалізації алгоритм руху доцільно поділити на кілька функціональних рівнів. Перший рівень відповідає за приймання команд користувача або LLM-модуля, наприклад «рухатися вперед», «повернути

праворуч», «зупинитися» або «виконати демонстраційний режим». Другий рівень перетворює цю команду на конкретний режим ходи. Третій рівень формує координати опорних точок для кожної ноги. Четвертий рівень виконує розрахунок кутів сервоприводів за допомогою зворотної кінематики. П'ятий рівень передає значення на сервоконтролер, який формує PWM-сигнали для виконавчих механізмів. Такий поділ спрощує налагодження системи, оскільки кожен рівень можна тестувати окремо.

Важливу роль у реалізації руху відіграє синхронізація сервоприводів. Якщо рух кожної ноги буде виконуватися незалежно, без урахування положення інших кінцівок, робот може втратити стійкість або виконувати ривки. Тому програмний алгоритм повинен забезпечувати часову узгодженість між фазами руху. Наприклад, під час повільної статичної ходи спочатку може переміщуватися одна нога, потім корпус зміщується для перенесення центра мас, після чого активується наступна кінцівка. Під час ходи риссю одночасно працюють діагональні пари ніг, що потребує точнішої синхронізації та стабільнішого живлення сервоприводів. Для Raspberry Pi така задача може виконуватися на рівні програмного планувальника руху, тоді як формування стабільних PWM - сигналів доцільно покласти на окремий сервоконтролер.

Сенсорний зворотний зв'язок є необхідним для підвищення надійності алгоритмів руху. Інерціальний модуль IMU може використовуватися для оцінювання нахилу корпусу, прискорення та кутової швидкості, датчики відстані – для виявлення перешкод, а камера – для візуального контролю простору. У простішому варіанті сенсорні дані можуть лише блокувати небезпечний рух, наприклад зупиняти платформу перед перешкодою. У складнішому варіанті вони можуть використовуватися для корекції положення корпусу або вибору альтернативної траєкторії. Сучасні дослідження демонструють, що адаптація руху квадропедів до складної місцевості може базуватися як на класичних методах керування, так і на методах навчання з підкріпленням, однак для бакалаврського прототипу доцільніше реалізувати

простий і контрольований алгоритм з поступовим розширенням функціональності [16, 17].

Під час реалізації кінематики квадропедної платформи необхідно враховувати апаратні обмеження. До них належать максимальні кути повороту сервоприводів, допустимий струм навантаження, механічні зазори, вага корпусу, розміщення акумулятора та стійкість конструкції. Якщо алгоритм задає надто різкі або великі переміщення кінцівок, сервоприводи можуть працювати з перевантаженням, що призводить до нестабільного руху або просідання напруги живлення. Тому практична реалізація повинна передбачати обмеження кутів, плавну зміну положення, затримки між фазами руху, калібрування початкового положення кожної ноги та перевірку працездатності сервоприводів перед запуском основного циклу.

У межах даної роботи алгоритм руху квадропедної платформи доцільно реалізувати як поєднання наперед заданих шаблонів ходи та розрахункових процедур зворотної кінематики. Наперед задані шаблони спрощують початковий запуск робота, а зворотна кінематика забезпечує гнучкіше формування положення кінцівок. Raspberry Pi у такій архітектурі виконує високорівневе керування: отримує команду, визначає сценарій руху, обробляє сенсорні дані та передає обчислені значення на сервоконтролер. Допоміжний контролер сервоприводів забезпечує стабільне формування PWM-сигналів і розвантажує Raspberry Pi від низькорівневих часово-критичних задач.

Реалізація кінематики та алгоритмів руху квадропедної платформи є комплексним завданням, яке поєднує механічну модель кінцівок, розрахунок кутів сервоприводів, планування траєкторії стопи, вибір типу ходи, синхронізацію руху та використання сенсорного зворотного зв'язку. Для бакалаврської роботи найбільш доцільним є застосування квазістатичної або статичної ходи, оскільки вона є стабільною, зрозумілою для реалізації та достатньою для демонстрації роботи інтелектуальної квадропедної платформи. Надалі такий підхід може бути розширений за рахунок складніших алгоритмів

стабілізації, комп'ютерного зору, адаптивного керування та використання LLM для вибору сценаріїв поведінки робота.

1.4 Перспективи інтеграції LLM у роботизовані системи для обробки команд і прийняття рішень

Інтеграція великих мовних моделей у роботизовані системи є одним із сучасних напрямів розвитку інтелектуальної робототехніки. Якщо традиційні роботизовані платформи переважно виконують наперед задані команди або працюють за жорстко визначеним алгоритмом, то використання LLM дає змогу перейти до більш гнучкої взаємодії між людиною і роботом. У такому підході користувач може формулювати команди природною мовою, а система повинна інтерпретувати їх, визначити намір, зіставити його з доступними можливостями робота та сформулювати відповідний сценарій дії. У сучасних наукових оглядах LLM розглядаються як перспективний засіб для покращення взаємодії людини з роботом, планування дій, прийняття рішень, обробки команд і підвищення автономності роботизованих систем [18].

У контексті квадропедної роботизованої платформи LLM доцільно розглядати не як безпосередній контролер сервоприводів, а як високорівневий інтелектуальний модуль. Це означає, що модель не повинна напряду формувати PWM-сигнали або керувати окремими приводами в реальному часі. Її основне завдання полягає в інтерпретації команди користувача, виборі логічного сценарію поведінки та передаванні узагальненого завдання до локального програмного модуля Raspberry Pi. Наприклад, команда «підійди вперед і зупинись перед перешкодою» може бути перетворена на послідовність дій: визначити режим руху вперед, активувати сенсорний контроль відстані, запустити алгоритм ходи, перевіряти наявність перешкоди та зупинити платформу при досягненні заданої умови.

Важливою перевагою LLM є здатність працювати з природномовними запитами, які можуть бути нечіткими або неповними. Для звичайного

програмного алгоритму команда повинна мати формалізований вигляд, наприклад «move_forward(10)» або «turn_left(30)». Натомість LLM може опрацьовувати команди на зразок «відійди трохи назад», «повернись до мене», «оглянь простір перед собою» або «зупинись, якщо побачиш перешкоду». Після інтерпретації така команда перетворюється на одну або кілька формалізованих інструкцій, які вже можуть бути виконані локальним програмним забезпеченням робота. Саме така проміжна роль LLM є найбільш доцільною для бакалаврського прототипу, оскільки вона дозволяє поєднати інтелектуальну обробку команд із надійними класичними алгоритмами руху.

Одним із відомих підходів до поєднання мовних моделей і робототехніки є концепція «заземлення» мовних команд у реальних можливостях робота. Наприклад, у підході SayCan мовна модель використовується для визначення того, яка дія є доречною для виконання інструкції, а окрема оцінка можливостей робота визначає, чи може робот фізично виконати цю дію в поточному середовищі [19]. Такий підхід є важливим, оскільки мовна модель може запропонувати логічно правильну дію, але вона не завжди враховує фізичні обмеження конкретної платформи: тип приводів, заряд акумулятора, наявність перешкод, стійкість корпусу або доступні режими ходи.

Для квадропедної платформи це означає, що команда, отримана від LLM, повинна проходити через додатковий рівень перевірки. Наприклад, якщо користувач просить робота «перестрибнути перешкоду», система має перевірити, чи підтримує прототип такий режим руху. Якщо відповідного алгоритму немає, Raspberry Pi повинен замінити цю дію на безпечнішу альтернативу, наприклад «зупинитися перед перешкодою» або «обійти перешкоду», якщо така функція реалізована. LLM виконує функцію інтерпретатора й планувальника, але остаточне рішення про виконання руху має приймати локальний контролер із урахуванням стану апаратної частини.

У сучасних дослідженнях також розвивається напрям мультимодальних моделей, які поєднують текст, зображення, сенсорні дані та дії робота. Прикладом є PaLM-E, де мовна модель доповнюється візуальною та сенсорною

інформацією для виконання задач втіленого міркування, планування і взаємодії з фізичним середовищем [20]. Для мобільної роботизованої платформи це має важливе значення, оскільки робот працює не лише з текстовими командами, а й із даними від камери, IMU-модуля, датчиків відстані та інших сенсорів.

Ще одним перспективним напрямом є використання моделей типу Vision - Language-Action. У таких системах модель не лише аналізує текст і зображення, а й формує дії, які можуть бути виконані роботизованою системою. У роботі RT-2 досліджено підхід, за якого знання, отримані з великих наборів вебданих і візуально-мовного навчання, використовуються для узагальнення роботизованих дій [21]. Для бакалаврської роботи така модель може розглядатися не як обов'язковий елемент реалізації, а як перспективний напрям розвитку системи в майбутньому, оскільки повноцінне керування рухом через Vision-Language-Action моделі потребує значних обчислювальних ресурсів, навчальних даних і додаткових механізмів безпеки.

У 2025 році розвиток цього напрямку посилився завдяки появі спеціалізованих моделей для робототехніки, орієнтованих на сприйняття, планування, використання інструментів і виконання складних багатокрокових задач у фізичному середовищі. Наприклад, Google DeepMind описує Gemini Robotics 1.5 як модель для фізичних агентів, яка дає змогу роботам сприймати, планувати, міркувати, використовувати інструменти та діяти в складніших сценаріях [22]. Для цієї роботи такі рішення є важливими як приклад загального напрямку розвитку галузі, однак у практичному прототипі доцільно реалізовувати спрощену архітектуру: LLM – для обробки команд, Raspberry Pi – для логіки керування, сервоконтролер – для формування сигналів руху.

Архітектура інтеграції LLM у квадрокоптерну платформу може мати кілька варіантів. Перший варіант передбачає використання хмарного API, коли Raspberry Pi передає текстовий запит до зовнішнього LLM-сервісу та отримує відповідь у вигляді структурованої команди. Такий підхід простіший у реалізації, оскільки не потребує запуску великої моделі безпосередньо на Raspberry Pi. Його недоліками є залежність від інтернет-з'єднання, затримки передавання даних і

потреба враховувати питання конфіденційності. Другий варіант передбачає використання локальної полегшеної моделі або окремого AI-прискорювача. Це зменшує залежність від мережі, але підвищує вимоги до апаратного забезпечення. Третій варіант є комбінованим: базові команди виконуються локально, а складніші запити передаються до хмарного сервісу [23].

Таблиця 1.2 – Можливі способи інтеграції LLM у кваліфіковану роботизовану платформу

Спосіб інтеграції	Сутність підходу	Переваги	Обмеження
Хмарна LLM через API	Raspberry Pi передає команду до зовнішнього сервісу та отримує відповідь	Висока якість обробки мови, простіша реалізація	Залежність від інтернету, затримки, питання приватності
Локальна LLM	Модель запускається безпосередньо на пристрої або локальному сервері	Автономність, кращий контроль даних	Високі вимоги до ресурсів
Гібридна архітектура	Прості команди виконуються локально, складні передаються до хмари	Баланс автономності та функціональності	Складніша логіка перемикавання режимів
LLM як планувальник дій	Модель формує сценарій, а рух виконується класичними алгоритмами	Безпечніше керування, зрозуміла структура	Потрібен модуль перевірки команд
Мультимодальна LLM	Модель працює з текстом, зображеннями та сенсорними даними	Краща взаємодія із середовищем	Потребує камери, сенсорів і потужнішої обробки

Практично для розроблюваної платформи найбільш доцільною є архітектура, у якій LLM виконує функцію високорівневого інтерпретатора команд, а не безпосереднього контролера руху. У такій архітектурі користувач надсилає команду текстом або голосом, Raspberry Pi передає її до LLM-модуля, модель повертає структуровану відповідь, а локальне програмне забезпечення перевіряє її на відповідність допустимим сценаріям. Наприклад, відповідь LLM може бути перетворена на одну з команд: «MOVE_FORWARD», «TURN_LEFT», «STOP», «SCAN_ENVIRONMENT», «RETURN_TO_IDLE». Після цього

Raspberry Pi запускає відповідний алгоритм руху, а сервоконтролер забезпечує керування виконавчими механізмами.

Перспективним є також застосування LLM для діалогової взаємодії з користувачем. Робот може не лише виконувати команди, а й уточнювати їх. Наприклад, якщо користувач дає команду «підійди ближче», система може уточнити, на яку відстань потрібно переміститися, або виконати обмежений рух із сенсорним контролем. Якщо команда є потенційно некоректною або не підтримується поточною версією платформи, LLM може сформулювати зрозуміле пояснення: «цей режим руху не реалізовано», «перешкода виявлена попереду», «заряд акумулятора недостатній для продовження руху». Це підвищує зручність використання платформи та робить її більш придатною для навчальних і демонстраційних цілей [24].

Окреме значення має використання LLM для прийняття рішень на рівні сценаріїв поведінки. У простій реалізації робот може мати набір попередньо визначених режимів: очікування, рух уперед, поворот, огляд простору, зупинка, обхід перешкоди. LLM може вибирати один із цих режимів залежно від команди користувача та поточного стану системи. У складнішому варіанті модель може формувати послідовність дій, наприклад: «перевірити простір перед собою», «якщо перешкод немає – рухатися вперед», «якщо перешкода є – зупинитися і повідомити користувача». Такий підхід наближає роботизовану платформу до концепції автономного агента, однак потребує суворого обмеження доступних дій і перевірки кожного кроку.

Разом із перевагами інтеграція LLM у роботизовані системи має низку обмежень. До них належать можливість помилкової інтерпретації команди, затримки під час обробки запитів, залежність від мережі у хмарному варіанті, складність перевірки правильності відповіді моделі та ризик формування дій, які не відповідають реальним можливостям робота. Тому для технічної реалізації важливо використовувати проміжний рівень валідації команд. Він має перевіряти, чи належить команда до дозволеного набору дій, чи не суперечить

вона даним сенсорів, чи не перевищує механічні обмеження платформи та чи може бути виконана без втрати стійкості.

Для бакалаврського прототипу доцільно реалізувати безпечну спрощену модель інтеграції LLM. Вона може містити такі етапи: отримання текстової або голосової команди, передавання запиту до LLM, перетворення відповіді у формалізовану команду, перевірка команди локальним модулем безпеки, запуск відповідного алгоритму руху та контроль виконання за сенсорними даними. Такий підхід дозволяє продемонструвати переваги інтелектуального керування, але не перекладає критичні задачі стабілізації та керування приводами на мовну модель.

Загальну логіку інтеграції LLM у роботизовану платформу доцільно подати на рисунку 1.7.

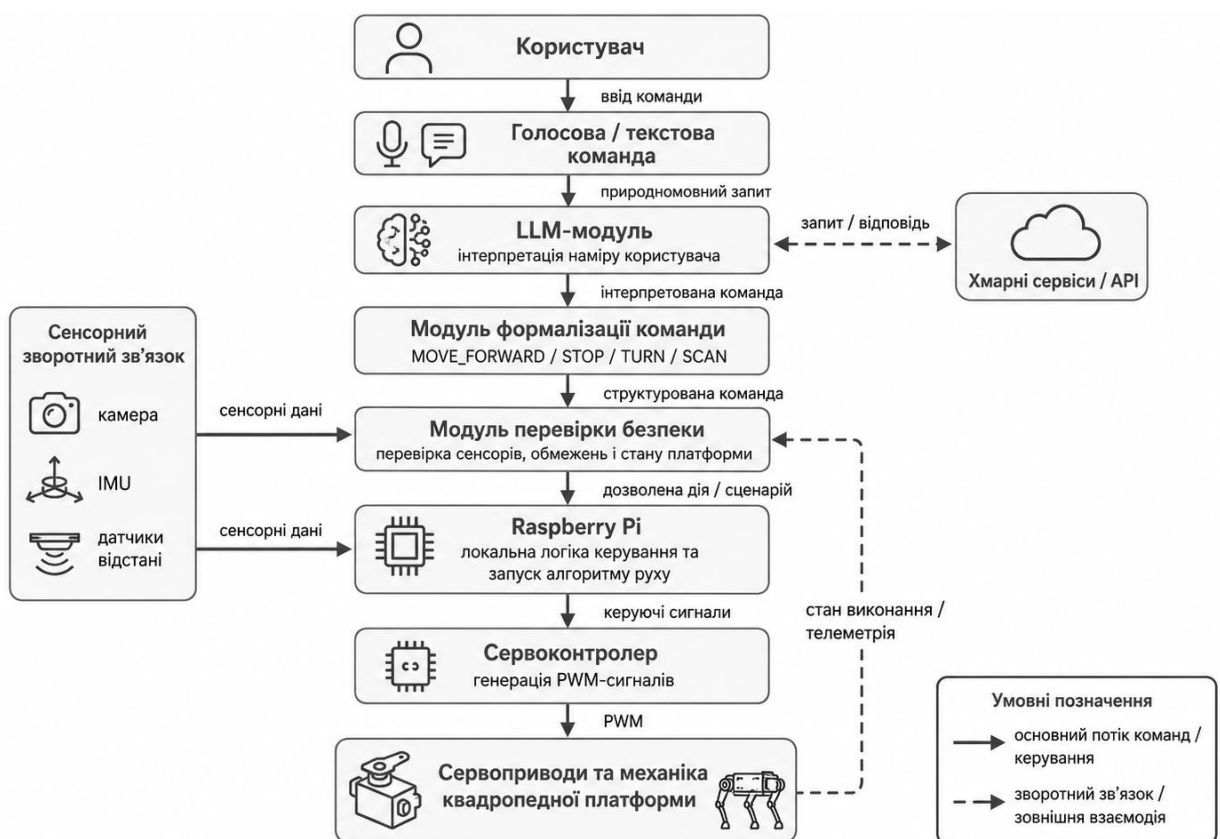


Рисунок 1.7 – Узагальнена схема інтеграції LLM у систему керування квадроцепдною платформою

Перспективи інтеграції LLM у роботизовані системи пов'язані з переходом від жорсткого командного керування до більш природної та адаптивної взаємодії між людиною і роботом. Для квадропедної платформи на базі Raspberry Pi така інтеграція дає змогу реалізувати інтерпретацію команд природною мовою, вибір сценаріїв руху, формування відповідей користувачу та часткове планування дій. Водночас LLM повинна використовуватися лише на високому рівні керування, тоді як безпосереднє формування руху, контроль сервоприводів, перевірка сенсорів і забезпечення безпечної поведінки мають виконуватися локальними алгоритмами. Саме такий розподіл функцій дозволяє поєднати переваги сучасних мовних моделей із надійністю класичних апаратно-програмних засобів керування роботизованою платформою.

РОЗДІЛ 2

РОЗРОБКА АПАРАТНО-ПРОГРАМНОЇ СТРУКТУРИ КВАДРОПЕДНОЇ РОБОТИЗОВАНОЇ ПЛАТФОРМИ

2.1 Обґрунтування вибору Raspberry Pi як центрального обчислювального модуля системи

Під час розробки інтелектуальної квадропедної роботизованої платформи одним із ключових етапів є вибір центрального обчислювального модуля, який забезпечуватиме обробку команд користувача, взаємодію із сенсорними компонентами, запуск алгоритмів руху, мережевий обмін даними та інтеграцію з LLM-модулем. Для реалізації такої системи було обрано Raspberry Pi, оскільки цей одноплатний комп'ютер поєднує достатню продуктивність, підтримку операційної системи Linux, наявність GPIO-інтерфейсів, можливість підключення камери, роботу з Python-бібліотеками та підтримку бездротових мережевих інтерфейсів [25, 26]. Офіційні специфікації Raspberry Pi 4 Model B підтверджують наявність 64-бітного чотириядерного процесора, модифікацій оперативної пам'яті до 8 ГБ, Wi-Fi, Bluetooth, Gigabit Ethernet, USB-портів, 40 - контактного GPIO-роз'єму та CSI-інтерфейсу для камери.

Raspberry Pi розглядається як головний вузол високорівневого керування, а не як безпосередній силовий контролер сервоприводів. Це означає, що на нього покладаються задачі обробки інформації, прийняття логічних рішень, вибору сценаріїв руху, взаємодії з користувачем і передавання команд до допоміжних модулів. Безпосереднє формування PWM-сигналів для сервоприводів доцільно реалізовувати через окремий сервоконтролер, наприклад PCA9685, оскільки квадропедна платформа потребує одночасного керування значною кількістю виконавчих механізмів. Такий розподіл функцій знижує навантаження на Raspberry Pi та підвищує стабільність роботи системи.

Для квадропедної платформи особливо важливо, щоб центральний модуль міг працювати з кількома потоками даних одночасно. Raspberry Pi здатний приймати інформацію від камери, IMU-модуля, датчиків відстані, мікрофона,

модуля зв'язку та сервоконтролера. На програмному рівні це дає змогу реалізувати багаторівневу логіку: отримання команди, її обробку, перевірку стану сенсорів, вибір алгоритму ходи, формування керуючих команд і контроль виконання. Для навчального або дослідницького прототипу така функціональність є важливою, оскільки дозволяє не обмежуватися простим переміщенням робота, а створити повноцінну апаратно-програмну систему з елементами інтелектуального керування.

Під час вибору Raspberry Pi враховувалися такі технічні критерії: обчислювальна продуктивність, кількість доступної оперативної пам'яті, підтримка Python, можливість роботи з камерою, наявність мережевих інтерфейсів, сумісність із сенсорними модулями, підтримка GPIO, енергоспоживання, компактність і доступність навчальних матеріалів. У порівнянні з Arduino Uno, Raspberry Pi має значно ширші можливості для обробки даних, роботи з операційною системою, запуску вебсервісів, взаємодії з API та інтеграції з LLM. Водночас Arduino або інші мікроконтролери можуть ефективно виконувати допоміжні задачі низькорівневого керування, тому їх доцільно розглядати не як повну заміну Raspberry Pi, а як додаткові виконавчі модулі.

Для обґрунтування вибору центрального обчислювального модуля доцільно порівняти можливості Raspberry Pi та мікроконтролерних платформ у контексті керування квадроцепним роботом. Основні відмінності між цими апаратними рішеннями подано в таблиці 2.1.

Таблиця 2.1 – Порівняння можливостей Raspberry Pi та мікроконтролерних платформ у задачах керування квадроцепним роботом

Критерій порівняння	Raspberry Pi	Arduino Uno / подібні мікроконтролери	Значення для квадроцепної платформи
Тип пристрою	Одноплатний комп'ютер	Мікроконтролерна плата	Raspberry Pi краще підходить для високорівневої логіки
Операційна система	Підтримка Linux / Raspberry Pi OS	Переважно виконання одного скетчу	Дає змогу запускати складніші програми та сервіси

Продовження таблиці 2.1

Критерій порівняння	Raspberry Pi	Arduino Uno / подібні мікроконтролери	Значення для квадропедної платформи
Обробка зображень	Підтримка камер і бібліотек комп'ютерного зору	Обмежена або відсутня	Важливо для орієнтації та спостереження
Мережеві можливості	Wi-Fi, Bluetooth, Ethernet	Потребує додаткових модулів	Дає змогу працювати з API, LLM і вебінтерфейсом
GPIO та інтерфейси	GPIO, I2C, SPI, UART, USB	GPIO, I2C, SPI, UART	Обидві платформи можуть взаємодіяти з периферією
Керування сервоприводами	Доцільно через сервоконтролер	Може формувати PWM напругу	Для великої кількості сервоприводів потрібен окремий контролер
Інтеграція LLM	Можлива через API або локальні сервіси	Практично обмежена	Raspberry Pi є придатним для інтелектуального рівня керування

З таблиці 2.1 видно, що Raspberry Pi доцільно використовувати саме як центральний обчислювальний модуль, оскільки він забезпечує ширші можливості для програмної обробки, мережевої взаємодії та інтеграції інтелектуальних сервісів. Для квадропедної платформи це має принципове значення, оскільки система повинна не лише подавати сигнали на сервоприводи, а й обробляти команди користувача, аналізувати сенсорні дані, взаємодіяти з LLM-модулем і контролювати логіку виконання руху.

Окремим аргументом на користь Raspberry Pi є підтримка камер і засобів комп'ютерного зору. Офіційна документація Raspberry Pi описує декілька поколінь камерних модулів, зокрема Camera Module 2, Camera Module 3, Global Shutter Camera та High Quality Camera, що свідчить про сформовану екосистему апаратних засобів для візуального сприйняття середовища [27]. Для квадропедного робота камера може використовуватися для спостереження за простором, виявлення перешкод, передавання відео, реалізації елементів машинного зору або формування додаткових даних для майбутнього розширення системи.

Ще однією перевагою Raspberry Pi є підтримка мережевих інтерфейсів. Для інтелектуальної квадропедної платформи це особливо важливо, оскільки взаємодія з LLM найчастіше реалізується через мережевий API або локальний сервер. Raspberry Pi може приймати команду користувача через вебінтерфейс, мобільний застосунок, SSH-з'єднання або голосовий модуль, після чого передавати запит до LLM-модуля й отримувати формалізовану відповідь. Наявність Wi-Fi, Bluetooth і Ethernet у Raspberry Pi 4 Model B створює основу для гнучкої організації зв'язку між користувачем, роботом і зовнішніми сервісами [26].

У разі потреби підвищення продуктивності доцільно використовувати Raspberry Pi 5. В офіційному продуктному описі зазначено, що Raspberry Pi 5 оснащено 64-бітним чотириядерним процесором Arm Cortex-A76 з частотою 2,4 ГГц, а продуктивність процесора зросла приблизно у 2-3 рази порівняно з Raspberry Pi 4 [25]. Також Raspberry Pi 5 доступний у модифікаціях оперативної пам'яті до 16 ГБ, що робить його більш придатним для складніших сценаріїв, пов'язаних з обробкою зображень, паралельною роботою кількох програмних модулів і взаємодією з інтелектуальними сервісами.

Разом з тим вибір Raspberry Pi 5 не є єдино можливим. Для базової реалізації квадропедної платформи може бути достатньо Raspberry Pi 4 Model B з 4 або 8 ГБ оперативної пам'яті. Такий варіант є доцільним, якщо основні задачі системи обмежуються обробкою команд, роботою з камерою, передаванням даних через мережу, взаємодією з хмарним LLM-сервісом і запуском базових алгоритмів ходи. Raspberry Pi 5 доцільніше обирати тоді, коли планується розширення системи, наприклад додавання складніших алгоритмів комп'ютерного зору, локальних AI-модулів, ROS 2 або кількох паралельних сервісів.

У межах розроблюваної системи Raspberry Pi виконує такі основні функції:

- 1) Приймання команд користувача через текстовий, голосовий або мережевий інтерфейс.

2) Передавання команди до LLM-модуля та отримання інтерпретованого результату.

3) Формалізація відповіді LLM у вигляді дозволених команд руху.

4) Перевірка стану сенсорів і обмежень безпеки перед виконанням дії.

5) Запуск алгоритмів руху квадропедної платформи.

6) Передавання керуючих параметрів на сервоконтролер.

7) Обробка даних від камери, IMU-модуля та датчиків відстані.

8) Організація мережевого обміну, журналювання й налагодження системи.

Зазначені функції підтверджують, що Raspberry Pi є не лише засобом підключення периферії, а центральним логічним ядром системи. Його роль полягає в координації всіх підсистем: сенсорної, виконавчої, комунікаційної та інтелектуальної. У цьому полягає відмінність від мікроконтролерів, які ефективні в задачах низькорівневого керування, але мають обмеження під час роботи з операційними системами, мережею, API та складною обробкою даних.

Важливо враховувати й апаратні обмеження Raspberry Pi. По-перше, GPIO-лінії не призначені для живлення сервоприводів або потужних виконавчих механізмів. По-друге, для сервоприводів квадропедної платформи необхідна окрема стабілізована лінія живлення, оскільки одночасна робота кількох приводів може створювати значне струмове навантаження. По-третє, для підключення компонентів із логікою 5 В необхідно враховувати рівні сигналів і за потреби застосовувати узгоджувальні схеми або перетворювачі рівнів. Офіційна апаратна документація Raspberry Pi підтверджує наявність 40-контактного GPIO-роз'єму та описує типові інтерфейси плат, але силові навантаження мають виноситися на окремі драйвери й модулі живлення [28].

Для підключення сервоприводів доцільно використати окремий сервоконтролер, який прийматиме команди від Raspberry Pi через I2C або інший інтерфейс і формуватиме PWM-сигнали для виконавчих механізмів. Такий підхід є технічно обґрунтованим, оскільки квадропедна платформа потребує узгодженої роботи багатьох сервоприводів. Raspberry Pi в цьому випадку не витрачає ресурси на точне формування імпульсів для кожного каналу, а виконує

обчислювальні задачі: розрахунок положення кінцівок, вибір шаблону ходи, контроль стану платформи та взаємодію з LLM-модулем.

Доцільність вибору Raspberry Pi також пояснюється його сумісністю з робототехнічними програмними екосистемами. Зокрема, ROS 2 підтримує ARM-процесори, а офіційна документація ROS 2 зазначає, що arm64 має вищий рівень підтримки порівняно з arm32, оскільки для Tier 1 доступні пакети та бінарні архіви [29]. Для цієї бакалаврської роботи повне використання ROS 2 може бути необов'язковим, однак така сумісність створює можливість подальшого розвитку платформи в напрямі складнішої робототехнічної архітектури.

Узагальнену роль Raspberry Pi у структурі розроблюваної системи (рис. 2.1) можна описати як поєднання чотирьох функціональних рівнів. Перший рівень – комунікаційний, який забезпечує приймання команд користувача та взаємодію з LLM або хмарними сервісами. Другий рівень – аналітичний, пов'язаний з обробкою сенсорних даних і перевіркою умов руху. Третій рівень – керуючий, який відповідає за вибір сценарію поведінки та запуск алгоритму ходи. Четвертий рівень – інтерфейсний, який забезпечує передавання команд до сервоконтролера, модулів зв'язку та інших допоміжних компонентів.

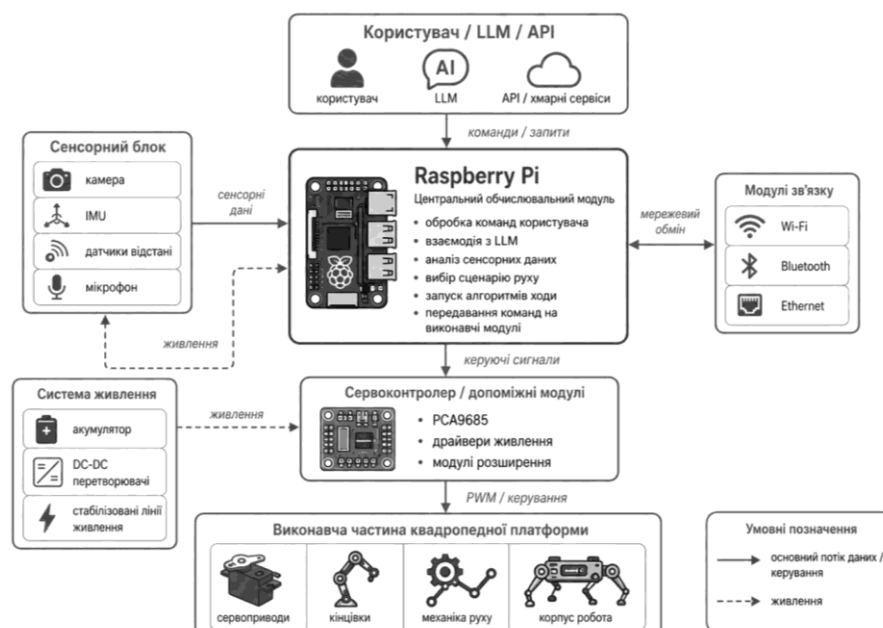


Рисунок 2.1 – Raspberry Pi як центральний обчислювальний модуль квадропедної платформи [29]

Raspberry Pi було обрано як центральний обчислювальний модуль системи через поєднання продуктивності, компактності, підтримки Linux, наявності GPIO, сумісності з камерними модулями, мережевими можливостями та придатності до інтеграції з LLM. Для розроблюваної квадропедної роботизованої платформи він забезпечує високорівневу логіку керування, обробку команд користувача, взаємодію із сенсорним блоком і передавання команд до виконавчих модулів. Водночас для забезпечення стабільності роботи сервоприводів і безпечного живлення доцільно використовувати окремі допоміжні модулі: сервоконтролер, драйвери живлення, DC-DC перетворювачі та сенсорні плати. Такий розподіл функцій дозволяє створити надійну, масштабовану та придатну до подальшого розвитку інтелектуальну квадропедну платформу.

2.2 Вибір апаратних компонентів квадропедної платформи:

Вибір апаратних компонентів квадропедної роботизованої платформи є важливим етапом практичної реалізації системи, оскільки саме від нього залежать стійкість руху, точність позиціонування кінцівок, якість взаємодії з користувачем, можливість обробки сенсорних даних і подальша інтеграція LLM-модуля. На відміну від простої колісної платформи, квадропедний робот потребує значно складнішої апаратної структури, оскільки для формування руху необхідно узгоджено керувати кількома сервоприводами, контролювати положення корпусу, отримувати дані про навколишнє середовище та підтримувати стабільне живлення виконавчих механізмів.

У межах розроблюваної системи апаратні компоненти доцільно поділити на кілька функціональних груп: виконавчі механізми, модулі керування сервоприводами, сенсорний блок, аудіомодулі, засоби візуального сприйняття, модулі зв'язку та систему живлення. Такий поділ спрощує проєктування, оскільки кожен компонент виконує визначену роль у загальній архітектурі платформи. Raspberry Pi при цьому залишається центральним обчислювальним

модулем, а допоміжні компоненти забезпечують виконання конкретних апаратних функцій.

Основним виконавчим елементом квадропедної платформи є сервопривід. Для формування руху кінцівок робот повинен мати можливість задавати кутове положення кожного суглоба. У базовій конструкції одна кінцівка може мати два або три ступені свободи, тому для чотирьох кінцівок потрібно від 8 до 12 сервоприводів. Для навчального прототипу доцільно використовувати мікросервоприводи типу SG90 або їх міцніші аналоги з металевими шестернями, якщо конструкція має більшу масу. Згідно з технічними даними TowerPro SG90, такий сервопривід має масу 9 г, робочу напругу 4,8 В, крутний момент 1,8 кг·см за 4,8 В і швидкість 0,1 с/60° за 4,8 В, що робить його придатним для легких навчальних роботизованих конструкцій [30].

Під час вибору сервоприводів потрібно враховувати не лише номінальний кут повороту, а й реальне навантаження на кінцівки. Якщо маса корпусу, акумулятора, Raspberry Pi та сенсорів є значною, слабкі сервоприводи можуть працювати з перевантаженням, що призведе до ривків, нагрівання або нестабільного руху. Тому для демонстраційної моделі можна використати SG90, а для більш надійного прототипу доцільно розглянути сервоприводи з металевим редуктором, наприклад MG90S або подібні моделі. Важливо також передбачити окреме живлення сервоприводів, оскільки їх не можна живити безпосередньо від GPIO Raspberry Pi.

Для керування великою кількістю сервоприводів доцільно використати окремий сервоконтролер. Найбільш поширеним варіантом для таких задач є модуль на базі PCA9685, який дає змогу формувати PWM-сигнали для кількох каналів і підключається до Raspberry Pi через інтерфейс I2C. За даними Adafruit, 16-канальний 12-бітний PWM/Servo Driver на PCA9685 може керувати до 16 сервоприводів через I2C, використовуючи лише дві лінії керування, а вбудований PWM-контролер формує сигнали для всіх 16 каналів без додаткового навантаження на основний процесор [31].

Використання PCA9685 є технічно обґрунтованим для квадропедної платформи, оскільки 12 сервоприводів можуть бути підключені до одного модуля, а Raspberry Pi при цьому виконує лише високорівневі обчислення: розрахунок кутів сервоприводів, вибір шаблону ходи, обробку сенсорних даних і передавання команд через I2C. Такий підхід підвищує стабільність системи, оскільки генерація PWM-сигналів не залежить від поточного навантаження операційної системи Raspberry Pi. Крім того, наявність запасних каналів PCA9685 дозволяє в майбутньому підключити додаткові сервоприводи, наприклад для повороту камери, руху головного модуля або керування допоміжними механізмами.

Важливою складовою квадропедної платформи є камера. Вона може використовуватися для відеоспостереження, виявлення перешкод, фіксації навколишнього середовища, реалізації елементів комп'ютерного зору та подальшого розширення системи в напрямі мультимодальної взаємодії з LLM. Для Raspberry Pi доцільно використовувати офіційні камерні модулі, оскільки вони мають сумісність із CSI-інтерфейсом і підтримуються програмним стеком Raspberry Pi. Офіційна документація зазначає, що Raspberry Pi Camera Module 3 є 12-мегапіксельним модулем на базі Sony IMX708 із роздільною здатністю 4608×2592 пікселів, має автофокус і випускається у варіантах зі стандартним або широким полем зору [32].

Для цієї роботи оптимальним варіантом є Camera Module 3 або сумісна USB-камера. Camera Module 3 доцільна тоді, коли потрібна компактність, низька затримка та коректна інтеграція з Raspberry Pi OS. USB-камера може бути простішою для підключення, однак вона займає USB-порт і може мати більші затримки під час передавання відео. Офіційна документація Raspberry Pi також зазначає, що всі камери Raspberry Pi можуть записувати фотографії високої роздільної здатності та відео Full HD 1080p або краще за допомогою підтримуваних програмних засобів [33].

Мікрофон у квадропедній платформі необхідний для приймання голосових команд користувача. Якщо в системі передбачається використання LLM, то

голосовий інтерфейс може бути важливим елементом природної взаємодії з роботом. Для цього доцільно використовувати цифровий MEMS-мікрофон з інтерфейсом I2S, наприклад INMP441. За технічним описом INMP441 є всеспрямованим цифровим MEMS-мікрофоном з нижнім акустичним портом, має 24-бітний I2S-інтерфейс і може підключатися безпосередньо до цифрових процесорів без окремого аудіокодека [34].

Використання I2S-мікрофона є кращим порівняно з простими аналоговими мікрофонними модулями, оскільки Raspberry Pi не має вбудованого аналогового входу. У разі застосування аналогового мікрофона довелося б додавати окремий аналого-цифровий перетворювач, що ускладнює схему. Цифровий мікрофон дозволяє отримувати аудіосигнал у цифровій формі, що спрощує подальшу обробку голосу, розпізнавання команд і передавання текстового запиту до LLM - модуля.

Динамік або аудіомодуль потрібен для формування голосових або звукових відповідей робота. У простішому варіанті можна використати активний мінідинамік, підключений через USB-аудіокарту або аудіовихід відповідного модуля. У більш компактній реалізації доцільно застосувати I2S-підсилювач із динаміком, що дозволить Raspberry Pi генерувати звукові повідомлення без додаткових громіздких пристроїв. У межах цієї роботи динамік може використовуватися для озвучування стану системи: підтвердження прийняття команди, повідомлення про перешкоду, низький заряд акумулятора або завершення виконання сценарію.

Сенсорний блок квадропедної платформи повинен забезпечувати базове сприйняття навколишнього середовища та стану самого робота. До його складу доцільно включити IMU-модуль, датчики відстані та за потреби додаткові сенсори освітленості або стану живлення. IMU-модуль, наприклад MPU6050 або аналогічний сенсор, використовується для визначення нахилу корпусу, прискорення та кутової швидкості. Це важливо для оцінювання стабільності платформи, особливо під час руху, поворотів або переміщення нерівною поверхнею.

Датчики відстані необхідні для виявлення перешкод перед роботом або з боків платформи. У простій реалізації можна використати ультразвуковий датчик HC-SR04 або інфрачервоні датчики відстані. Ультразвуковий датчик є доступним і простим у використанні, однак потребує правильного узгодження рівнів сигналів із Raspberry Pi, оскільки логічні входи Raspberry Pi працюють із рівнем 3,3 В. Для підвищення надійності можна застосувати ToF-датчики відстані, які мають компактні розміри, кращу точність на малих дистанціях і зручну взаємодію через I2C.

Узагальнений склад апаратних компонентів квадропедної платформи наведено в таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору апаратних компонентів квадропедної платформи

Компонент	Орієнтовний варіант реалізації	Функція у системі	Причина вибору
Центральний модуль	Raspberry Pi 4/ Raspberry Pi 5	Обробка команд, сенсорних даних, взаємодія з LLM	Підтримка Linux, Python, GPIO, камер і мережевих інтерфейсів
Сервоприводи	SG90/MG90S або аналоги	Формування руху кінцівок	Доступність, керування кутовим положенням, придатність для прототипування
Сервоконтролер	PCA9685	Генерація PWM-сигналів	16 каналів, I2C, розвантаження Raspberry Pi
Камера	Raspberry Pi Camera Module 3/USB-камера	Візуальне сприйняття середовища	Підтримка з боку Raspberry Pi OS, можливість комп'ютерного зору
Мікрофон	INMP441 I2S MEMS	Приймання голосових команд	Цифровий інтерфейс I2S, відсутність потреби в окремому АЦП
Динамік	I2S-підсилювач із мінідинаміком/USB-аудіо	Звукові відповіді системи	Можливість реалізації діалогової взаємодії
IMU-модуль	MPU6050/MPU9250 або аналог	Контроль нахилу й руху корпусу	Оцінювання стабільності платформи
Датчики відстані	HC-SR04/ToF-датчики	Виявлення перешкод	Підвищення безпеки руху
Система живлення	Акумулятор, DC-DC перетворювачі	Живлення Raspberry Pi, сервоприводів і сенсорів	Стабілізація напруги та розділення силових ліній

Як показано в таблиці 2.2, апаратні компоненти мають бути підібрані не ізолювано, а з урахуванням їхньої сумісності між собою. Наприклад, сервоприводи потребують окремої лінії живлення на 5-6 В, Raspberry Pi потребує стабільного живлення 5 В, а сенсорні модулі можуть працювати з рівнями 3,3 В або 5 В. Тому в системі потрібно передбачити DC-DC перетворювачі, спільну «землю» між модулями та захист від просідання напруги під час одночасної роботи сервоприводів (рис. 2.2).

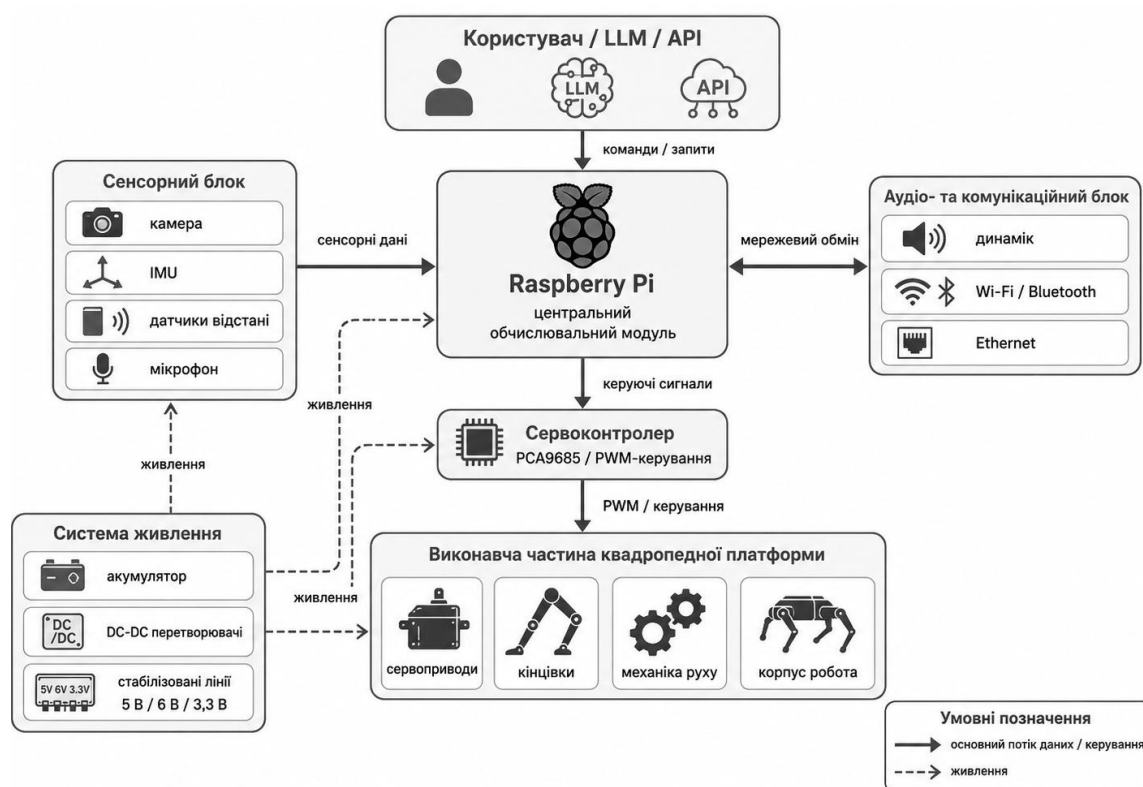


Рисунок 2.2 – Загальна структура апаратних компонентів квадронодної роботизованої платформи [34]

2.3 Розробка структурної та принципової схем інтелектуальної квадронодної роботизованої платформи

Розробка структурної та принципової схем є важливим етапом створення інтелектуальної квадронодної роботизованої платформи, оскільки саме на цьому етапі визначається логіка взаємодії між центральним обчислювальним модулем,

сенсорним блоком, сервоконтролером, виконавчими механізмами, аудіомодулями, системою живлення та програмним рівнем інтелектуального керування. Для розроблюваної системи центральним вузлом обрано Raspberry Pi, який виконує функції обробки команд користувача, взаємодії з LLM, аналізу сенсорних даних, вибору сценарію руху та передавання керуючих параметрів до сервоконтролера.

Структурна схема відображає загальну архітектуру платформи на рівні функціональних блоків. Вона дає змогу показати, які підсистеми входять до складу робота, які дані між ними передаються та який модуль відповідає за прийняття рішень. У цій роботі структурну схему доцільно будувати за ієрархічним принципом: верхній рівень - взаємодія з користувачем та LLM, середній рівень – Raspberry Pi як центральний обчислювальний модуль, нижній рівень – сервоконтролер, сервоприводи та механіка руху квадропедної платформи.

Структурна схема інтелектуальної квадропедної роботизованої платформи подана на рисунку 2.3.

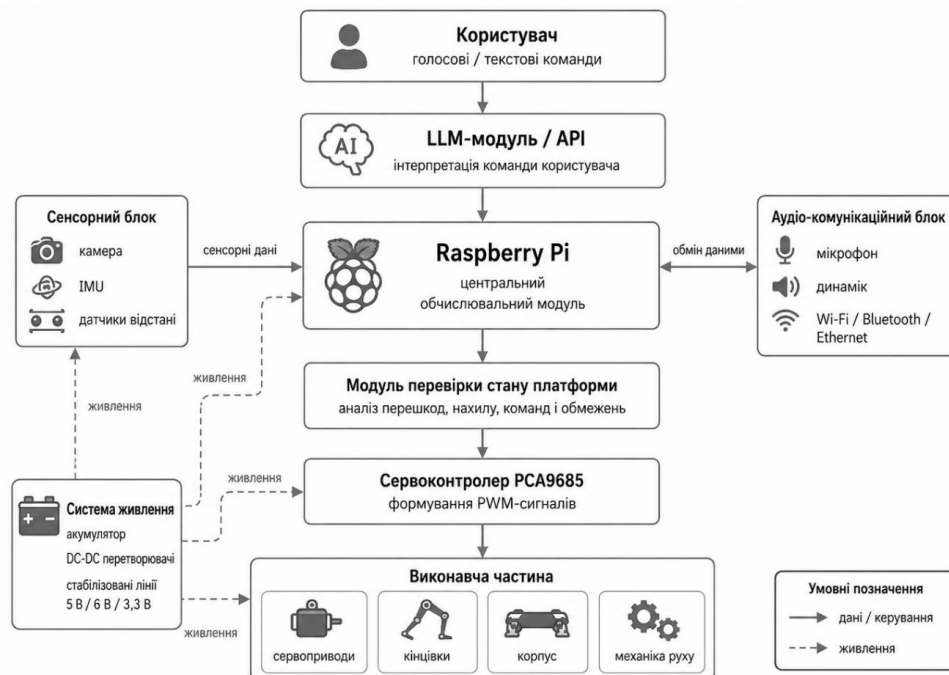


Рисунок 2.3 – Структурна схема інтелектуальної квадропедної роботизованої платформи

На рисунку 2.3 Raspberry Pi розміщується у центральній частині схеми, оскільки він координує роботу всіх апаратних та програмних компонентів. Команди користувача можуть надходити у текстовому або голосовому вигляді. У разі голосового керування сигнал спочатку приймається мікрофоном, після чого програмний модуль розпізнавання мовлення перетворює його на текстову команду. Далі команда передається до LLM-модуля або API, де виконується інтерпретація наміру користувача. Результатом роботи LLM є формалізована команда, наприклад «рух вперед», «зупинка», «поворот», «огляд простору» або «перехід у режим очікування».

Raspberry Pi отримує формалізовану команду та перевіряє її на відповідність доступним режимам роботи платформи. Після цього система аналізує сенсорні дані: показники IMU-модуля, інформацію від датчиків відстані, зображення з камери та стан окремих апаратних модулів. Якщо умови руху є безпечними, Raspberry Pi запускає відповідний алгоритм ходи та передає параметри положення кінцівок до сервоконтролера PCA9685. Сервоконтролер формує PWM-сигнали для сервоприводів, які змінюють положення кінцівок квадропедної платформи.

Окрему роль у структурній схемі виконує система живлення. Вона має забезпечити стабільне живлення Raspberry Pi, сервоприводів, сенсорів, камери, мікрофона та допоміжних модулів. Для цього доцільно використовувати акумуляторний блок і DC-DC перетворювачі, які формують окремі стабілізовані лінії живлення. Raspberry Pi потребує стабільної лінії 5 В, сервоприводи - окремої силової лінії 5-6 В, а окремі сенсорні модулі можуть працювати від 3,3 В або 5 В. При цьому всі модулі повинні мати спільну землю GND, щоб забезпечити коректну передачу керуючих сигналів.

Розподіл основних функцій між блоками структурної схеми наведено в таблиці 2.3.

Таблиця 2.3 – Функціональне призначення основних блоків інтелектуальної квадропедної платформи

Блок системи	Основне призначення	Дані або сигнали, що передаються
Користувач	Формування команд для робота	Текстові або голосові команди
LLM-модуль	Інтерпретація природномовних команд	Формалізовані сценарії дій
Raspberry Pi	Центральна обробка, логіка керування, запуск алгоритмів руху	Команди, сенсорні дані, керуючі параметри
Сенсорний блок	Збирання даних про середовище і стан платформи	Дані камери, IMU, датчиків відстані
Аудіо-комунікаційний блок	Голосова взаємодія та мережевий обмін	Аудіосигнали, мережеві запити, відповіді API
Сервоконтролер PCA9685	Генерація PWM-сигналів для сервоприводів	PWM-сигнали керування
Виконавча частина	Реалізація руху платформи	Механічне переміщення кінцівок
Система живлення	Забезпечення стабільної напруги для всіх модулів	5 В, 6 В, 3,3 В, GND

Принципова схема, на відміну від структурної, деталізує електричні з'єднання між компонентами. Вона повинна показувати, яким чином Raspberry Pi підключається до сервоконтролера, сенсорних модулів, мікрофона, камери, динаміка, системи живлення та виконавчих пристроїв. При розробці принципової схеми необхідно враховувати рівні логічних сигналів, струмове навантаження, потребу в окремому живленні сервоприводів і спільне з'єднання «землі» між усіма модулями.

У розроблюваній системі сервоконтролер PCA9685 підключається до Raspberry Pi через інтерфейс I2C. Типово для Raspberry Pi використовуються лінії SDA та SCL, а також живлення логічної частини від 3,3 В. Силова лінія живлення сервоприводів підводиться окремо до входу V+ сервоконтролера, оскільки сервоприводи можуть споживати значний струм під час старту, утримання положення або одночасного переміщення кількох кінцівок. Такий підхід дозволяє уникнути перевантаження Raspberry Pi та зменшити ризик перезавантаження плати через просідання напруги.

Принципова схема підключення компонентів інтелектуальної квадропедної платформи подана на рисунку 2.4.

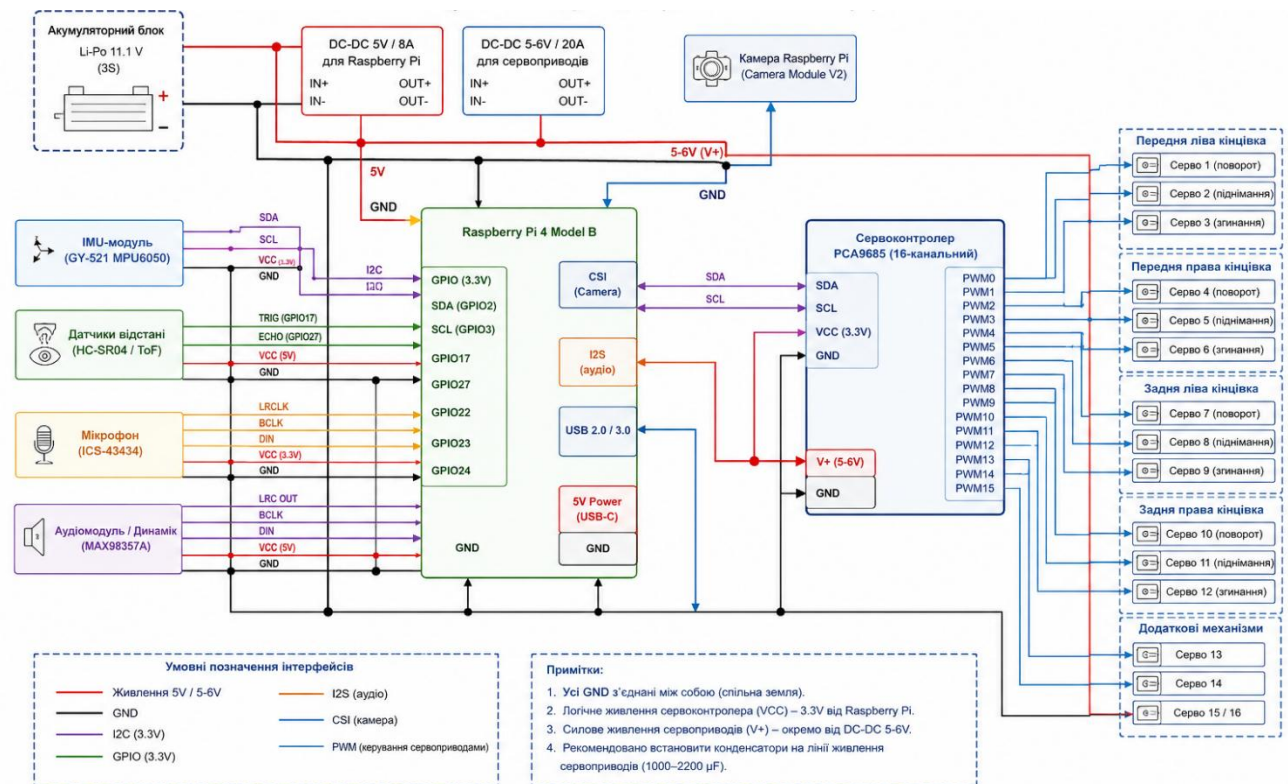


Рисунок 2.4 – Принципова схема підключення компонентів інтелектуальної квадропедної роботизованої платформи

На рисунку 2.4 показано, що живлення системи має бути розділене на логічну та силову частини. Логічна частина забезпечує Raspberry Pi, сенсорні модулі, мікрофон і комунікаційні пристрої. Силова частина живить сервоприводи, які є найбільш енергоємними компонентами системи. Для підвищення стабільності роботи доцільно використовувати окремий DC-DC перетворювач для Raspberry Pi та окремий перетворювач для сервоприводів. При цьому з'єднання GND між усіма частинами схеми є обов'язковим, оскільки без спільної землі керуючі сигнали між Raspberry Pi, PCA9685 і сервоприводами можуть передаватися некоректно.

Для зручності практичної реалізації орієнтовний розподіл підключень компонентів подано в таблиці 2.4.

Таблиця 2.4 – Орієнтовне підключення компонентів квадропедної платформи до Raspberry Pi та допоміжних модулів

Компонент	Інтерфейс підключення	Призначення підключення	Особливості реалізації
PCA9685	I2C	Керування сервоприводами	Логічне живлення від 3,3 В, силове живлення сервоприводів окремо
Сервоприводи	PWM-канали PCA9685	Рух кінцівок платформи	Потрібна окрема лінія 5-6 В і спільний GND
Камера	CSI або USB	Отримання відеоданих	CSI-камера компактніша, USB - камера простіша у заміні
IMU-модуль	I2C	Визначення нахилу та руху корпусу	Потрібне калібрування початкового положення
Датчики відстані	GPIO або I2C	Виявлення перешкод	Для 5-вольтових датчиків потрібне узгодження рівнів сигналів
Мікрофон	I2S або USB	Приймання голосових команд	I2S-мікрофон доцільний через відсутність аналогового входу Raspberry Pi
Динамік	I2S, USB або аудіомодуль	Звуковий зворотний зв'язок	Може використовуватися для повідомлень про стан системи
DC-DC перетворювач 5 В	Лінія живлення	Живлення Raspberry Pi	Потрібна стабільна напруга без просідань
DC-DC перетворювач 5-6 В	Лінія живлення	Живлення сервоприводів	Має витримувати сумарний струм навантаження

Особливу увагу в принциповій схемі потрібно приділити підключенню сервоприводів. Якщо квадропедна платформа має 12 сервоприводів, то вони можуть бути підключені до каналів CH0-CH11 сервоконтролера PCA9685. Наприклад, канали CH0-CH2 можуть відповідати передній лівій кінцівці, CH3 - CH5 - передній правій, CH6-CH8 - задній лівій, CH9-CH11 - задній правій. Такий розподіл спрощує програмну реалізацію, оскільки кожна кінцівка має власну групу каналів, а програмний код може оперувати не окремими номерами каналів, а логічними позначеннями суглобів.

У процесі розробки принципової схеми також необхідно врахувати питання електричної сумісності. Raspberry Pi працює з логічним рівнем 3,3 В, тому підключення модулів із 5-вольтовими виходами потребує обережності. Наприклад, для деяких ультразвукових датчиків відстані вихідний сигнал Echo

може мати рівень 5 В, тому його потрібно подавати на Raspberry Pi через подільник напруги або перетворювач рівнів. Неврахування цієї вимоги може призвести до пошкодження GPIO-входів Raspberry Pi.

Крім того, доцільно передбачити елементи захисту і стабілізації живлення: вимикач живлення, запобіжник або захист від перевантаження, конденсатори на лінії живлення сервоприводів, індикатори наявності напруги та розділення силової і логічної частин. Такі елементи підвищують надійність роботи платформи, особливо під час одночасного руху кількох кінцівок, коли сервоприводи створюють короточасні пікові навантаження.

Алгоритмічно структурна та принципова схеми пов'язані між собою. Структурна схема визначає, які функціональні блоки мають бути в системі, а принципова схема уточнює, якими електричними лініями ці блоки підключаються. Наприклад, у структурній схемі сенсорний блок позначається як єдиний вузол, а в принциповій схемі він деталізується на камеру, IMU, датчики відстані та мікрофон. Аналогічно виконавча частина у структурній схемі подається як блок «сервоприводи та механіка руху», а в принциповій схемі вона деталізується на канали сервоконтролера, лінії PWM, силове живлення та спільну землю.

Таким чином, розроблена структурна схема показує загальну логіку функціонування інтелектуальної квадропедної роботизованої платформи, а принципова схема визначає практичний порядок підключення апаратних компонентів. Запропонована архітектура передбачає використання Raspberry Pi як центрального обчислювального модуля, PCA9685 як сервоконтролера, окремої системи живлення для логічної та силової частин, сенсорного блоку для сприйняття середовища та аудіо-комунікаційного блоку для взаємодії з користувачем. Такий підхід забезпечує модульність, масштабованість і технічну надійність розроблюваної платформи, а також створює основу для подальшої інтеграції LLM у процес обробки команд і прийняття рішень.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ КВАДРОПЕДНОЇ ПЛАТФОРМИ

3.1 Складання фізичної моделі квадропедної роботизованої платформи

Практична реалізація інтелектуальної квадропедної роботизованої платформи в межах роботи виконувалася на основі робототехнічного комплекту SunFounder PiDog. Цей набір є готовою апаратною основою для створення робота-собаки на базі Raspberry Pi та дає змогу дослідити принципи складання, підключення, калібрування й програмного керування квадропедною платформою. Використання саме готового комплекту є доцільним для бакалаврської роботи, оскільки він містить основні механічні, електронні та сенсорні компоненти, необхідні для побудови фізичної моделі робота: корпусні елементи, сервоприводи, плату розширення Robot HAT, камеру, сенсорні модулі, акумуляторний блок і виконавчі механізми. В офіційній документації SunFounder PiDog зазначено, що комплект має стандартну версію та версію V2; при цьому V2 отримала оновлену плату Robot HAT і сервоконфігурацію, що забезпечує сумісність із Raspberry Pi 3/4/5 та Zero 2W, тоді як стандартна версія не призначена для Raspberry Pi 5.

На початковому етапі було виконано підготовку компонентів до складання. До складу фізичної моделі входять механічна рама, чотири кінцівки, сервоприводи для формування руху, Raspberry Pi як центральний обчислювальний модуль, плата Robot HAT як проміжний апаратний вузол керування, сенсорні модулі, камера, динамік, мікрофонний або звуковий модуль, акумулятор і з'єднувальні кабелі. Офіційна документація PiDog окремо подає апаратні модулі комплекту, серед яких Robot HAT, Camera Module, Sound Direction Sensor, 6-DOF IMU, Dual Touch Sensor, 11-channel Light Board, Ultrasonic Module і 3-pin Battery. Це підтверджує, що платформа є не лише механічною моделлю для ходьби, а повноцінним сенсорно-виконавчим комплексом для реалізації взаємодії з користувачем і середовищем.

Перед безпосереднім монтажем було доцільно перевірити комплектність деталей, стан сервоприводів, цілісність з'єднувальних кабелів, наявність кріпильних елементів і відповідність плати Robot HAT обраній моделі Raspberry Pi. Особливо важливим є правильний вибір версії комплекту. Якщо в системі використовується Raspberry Pi 4, може застосовуватися стандартна версія PiDog, а для Raspberry Pi 5 доцільною є саме версія PiDog V2, оскільки вона враховує вимоги до GPIO та живлення новішої плати. В офіційному описі зазначено, що V2 має перероблені сервоприводи, HAT-схему та оптимізоване живлення для Raspberry Pi 5.

Загальний вигляд апаратної основи SunFounder PiDog доцільно подати на рисунку 3.1.



Рисунок 3.1 – Загальний вигляд робот-собаки SunFounder PiDog як фізичної моделі квадропедної платформи

Після підготовки компонентів виконувалося складання механічної основи робота. Спочатку формувалася центральна частина корпусу, до якої надалі

кріпилися Raspberry Pi, Robot HAT, акумуляторний блок і допоміжні модулі. Корпус виконує не лише несучу функцію, а й забезпечує просторове розміщення основних електронних компонентів. Важливо, щоб центр мас платформи був розташований якомога ближче до геометричного центру корпусу, оскільки це позитивно впливає на стійкість робота під час ходьби, поворотів і переходу між положеннями.

Наступним етапом було складання кінцівок квадропедної платформи. Кожна кінцівка складається з окремих механічних ланок і сервоприводів, які виконують функції умовних суглобів. У конструкції PiDog використовується багатосервоприводна схема, що дає змогу імітувати рухи робот-собаки: піднімання, опускання, згинання й переміщення кінцівок. За даними описів комплекту, PiDog використовує 12 сервоприводів, що дозволяє реалізовувати ходьбу, зміну положення корпусу та інші дії, характерні для чотиринової платформи.

Під час монтажу сервоприводів важливим є дотримання їхнього початкового положення. Якщо сервопривід буде закріплений без попереднього вирівнювання, кінцівка може мати неправильну геометрію, що призведе до перекосу корпусу, нестабільної ходи або перевантаження окремих приводів. В офіційній інструкції зі складання SunFounder зазначено, що перед фіксацією сервоприводу його потрібно встановити в нульове положення; для цього спочатку встановлюють Raspberry Pi OS, необхідні модулі, запускають відповідний скрипт і переводять усі PWM-канали в нульове положення. Для версії PiDog V2 процедура може відрізнятись: на форумі SunFounder зазначено, що для V2 вже не потрібно запускати код для скидання положення сервоприводів перед складанням, а монтаж виконується відповідно до друкованої інструкції.

Після встановлення сервоприводів виконувалося складання чотирьох кінцівок і їх кріплення до корпусу. На цьому етапі потрібно перевірити симетричність лівої та правої сторін, правильність орієнтації сервоприводів, надійність фіксації гвинтів і відсутність механічних перешкод під час руху ланок. Якщо окрема ланка закріплена надто щільно або має перекис, це може створювати

додаткове навантаження на сервопривід. Тому після складання кожної кінцівки доцільно вручну перевірити діапазон її руху без подавання живлення на виконавчі механізми (рис. 3.2)

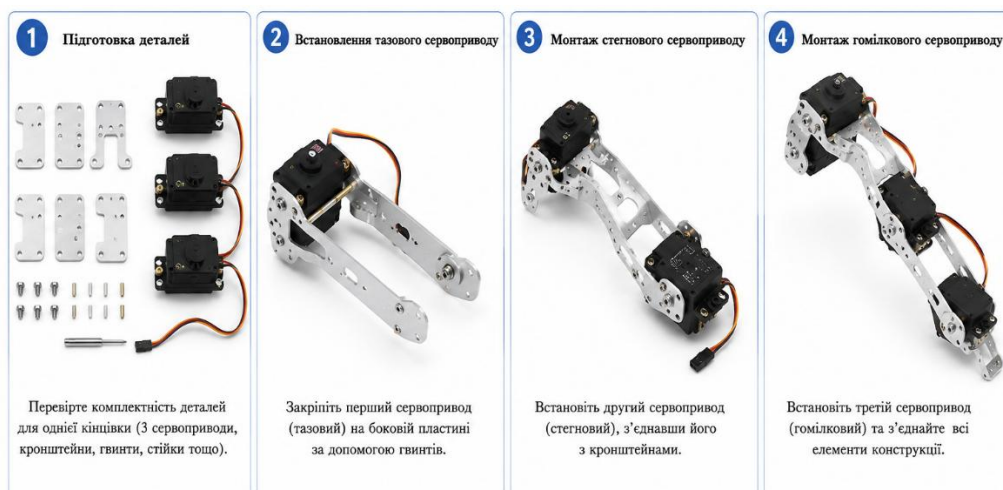


Рисунок 3.2 – Етап складання кінцівок квадропедної платформи

Після завершення механічного складання було встановлено центральний обчислювальний модуль Raspberry Pi та плату Robot HAT. Robot HAT у цій системі виконує роль апаратного посередника між Raspberry Pi, сервоприводами, сенсорними модулями, динаміком і системою живлення. В офіційній документації SunFounder зазначено, що Robot HAT V4 є багатофункціональною платою розширення для Raspberry Pi; вона містить мікроконтролер для розширення PWM-виходів і ADC-входів, драйвер двигунів, I2S-аудіомодуль, монофонічний динамік, виведені GPIO, індикатори живлення, кнопку та підтримує вхідне живлення 7-12 В. Для версії V2 використовується Robot HAT V5, де інтегровано мікрофон і передбачено можливості аудіозапису та розпізнавання голосу.

Встановлення Raspberry Pi виконувалося таким чином, щоб зберігався доступ до основних роз'ємів: живлення, USB, CSI-роз'єму камери, GPIO та мережевих інтерфейсів. Такий підхід спрощує подальше налагодження, оновлення програмного забезпечення, підключення периферії та перевірку роботи системи. Після монтажу Raspberry Pi до Robot HAT підключаються

шлейфи сервоприводів, сенсорні модулі та камера. Особливу увагу потрібно звернути на правильність полярності живлення та орієнтацію сигнальних роз'ємів, оскільки помилкове підключення сервоприводів або сенсорів може призвести до некоректної роботи системи.

Сенсорна частина фізичної моделі формується з кількох модулів. Камера використовується для візуального сприйняття середовища, ультразвуковий модуль – для визначення відстані до перешкод, IMU-модуль – для контролю положення та руху корпусу, а звуковий або мікрофонний модуль – для взаємодії з користувачем. У документації SunFounder зазначено, що 6-DOF IMU PiDog базується на SH3001, який поєднує тривісний гіроскоп і тривісний акселерометр, працює від 3,3 В та використовує інтерфейс ІІС. Ультразвуковий модуль HC-SR04 у комплекті забезпечує безконтактне вимірювання відстані в діапазоні від 2 до 400 см із точністю до 3 мм та має виводи TRIG, ECHO, GND і VCC.

Розміщення сенсорів під час складання має забезпечувати їхню практичну придатність. Камера повинна бути встановлена у передній частині платформи або в зоні, що імітує «голову» робота, щоб забезпечити огляд простору перед роботом. Ультразвуковий датчик доцільно орієнтувати вперед, оскільки основним завданням на базовому рівні є виявлення перешкод на траєкторії руху. IMU-модуль потрібно закріплювати на стабільній частині корпусу, оскільки його показники використовуються для оцінювання нахилу та руху всієї платформи. Звукові модулі та динамік розміщуються так, щоб вони не заважали руху кінцівок і не створювали механічних перешкод.

Окремим етапом є організація живлення. У квадрокоптерній платформі живлення має забезпечувати одночасну роботу Raspberry Pi, Robot HAT, сервоприводів і сенсорних модулів. Найбільше навантаження створюють саме сервоприводи, тому під час перевірки системи необхідно контролювати стабільність живлення, заряд акумулятора та відсутність просідання напруги під час одночасного руху кількох кінцівок. Robot HAT спрощує підключення живлення, оскільки передбачає вхід 7-12 В, індикатори заряду та індикатори живлення. Проте під час практичного використання потрібно дотримуватися

вимог виробника до підключення акумулятора й не допускати механічного пошкодження кабелів живлення під час руху робота (рис. 3.3).

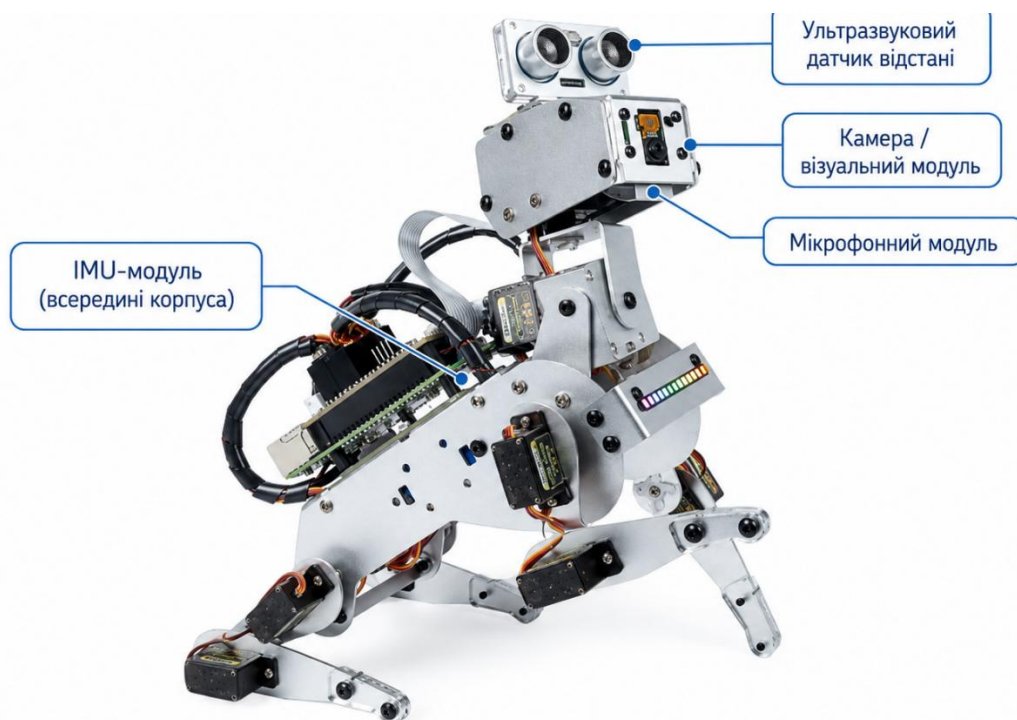


Рисунок 3.3 – Розміщення сенсорних модулів на фізичній моделі

Після механічного та електричного складання виконувалося первинне налагодження платформи. Воно включало перевірку живлення, запуск Raspberry Pi OS, встановлення необхідного програмного середовища, підключення до плати Robot HAT, перевірку роботи сервоприводів, камери, IMU-модуля, ультразвукового датчика та аудіомодулів. Важливо, що SunFounder PiDog має окремий розділ документації для роботи з Python, калібрування, прикладів руху та AI-взаємодії, що дає змогу використовувати платформу не лише як готовий робот, а як навчальний і дослідницький інструмент.

Після складання особливу увагу було приділено калібруванню сервоприводів. Калібрування потрібне для того, щоб усі кінцівки займали правильні початкові положення, а рух робота був симетричним і плавним. В офіційній документації PiDog описано процедуру калібрування з використанням калібрувальної лінійки, під час якої користувач обирає тип шаблону 90° або 60°,

налаштовує положення окремих сервоприводів і повторює процедуру для всіх чотирьох кінцівок. Після завершення калібрування рекомендовано повторно запустити приклад ходьби або пози робота, щоб перевірити плавність рухів.

Фізична модель PiDog після складання забезпечує виконання базових рухових дій: стояння, сидіння, переміщення вперед, повороти, зміну положення корпусу та реакцію на зовнішні сигнали. Крім того, наявність камери, IMU, ультразвукового датчика, динаміка та звукових модулів створює основу для реалізації інтелектуальної поведінки. У документації SunFounder наведено приклад «AI Voice Assistant Dog», у якому PiDog може очікувати ключове слово, слухати користувача, передавати текст до LLM, відповідати голосом через TTS і виражати реакцію за допомогою рухів, жестів та LED-ефектів. Це відповідає темі кваліфікаційної роботи, оскільки демонструє можливість поєднання фізичної квадроопедної платформи з мовною моделлю для обробки команд і формування поведінки робота.

Узагальнену послідовність складання фізичної моделі подано в таблиці 3.1.

Таблиця 3.1 – Основні етапи складання фізичної моделі квадроопедної платформи SunFounder PiDog

Етап складання	Зміст виконаних дій	Практичне значення
Перевірка комплектності	Огляд корпусних деталей, сервоприводів, Robot HAT, сенсорів, камери, акумулятора та кабелів	Дає змогу уникнути помилок перед монтажем
Підготовка сервоприводів	Вирівнювання або встановлення початкового положення сервоприводів	Забезпечує правильну геометрію кінцівок
Складання корпусу	Монтаж центральної рами та місць кріплення електроніки	Формує основу для розміщення Raspberry Pi й модулів
Складання кінцівок	Кріплення сервоприводів і механічних ланок	Забезпечує можливість реалізації ходи
Встановлення Raspberry Pi та Robot HAT	Монтаж центрального обчислювального модуля і плати розширення	Забезпечує апаратну основу керування
Підключення сенсорів	Монтаж камери, IMU, ультразвукового та звукових модулів	Забезпечує сприйняття середовища
Підключення живлення	Підключення акумулятора та перевірка індикаторів живлення	Забезпечує автономну роботу системи
Калібрування	Налаштування положення сервоприводів і перевірка симетрії руху	Підвищує стабільність і плавність ходи

Складання фізичної моделі квадропедної роботизованої платформи на основі SunFounder PiDog передбачає послідовне виконання механічного монтажу, встановлення Raspberry Pi та Robot HAT, підключення сервоприводів, сенсорних модулів, камери, аудіокомпонентів і системи живлення, а також калібрування виконавчих механізмів. Використання PiDog є доцільним для цієї кваліфікаційної роботи, оскільки платформа поєднує готову механічну основу, багатосервоприводну схему руху, сенсорний блок і підтримку програмного керування на Raspberry Pi. Це дає змогу перейти від теоретичної моделі квадропедного робота до практичного прототипу, придатного для тестування алгоритмів руху, взаємодії з користувачем і подальшої інтеграції LLM у систему керування.

3.2 Налаштування Raspberry Pi, встановлення програмного середовища та підключення периферійних модулів

Після складання фізичної моделі квадропедної роботизованої платформи SunFounder PiDog наступним етапом є налаштування Raspberry Pi, встановлення необхідного програмного середовища та перевірка роботи периферійних модулів. Цей етап є важливим, оскільки саме Raspberry Pi виконує роль центрального обчислювального модуля, через який здійснюється запуск програм керування, взаємодія з платою Robot HAT, приймання даних від сенсорів, обробка команд користувача та подальша інтеграція LLM-модуля.

Налаштування Raspberry Pi доцільно розпочинати з підготовки носія з операційною системою. Для цього використовується Raspberry Pi Imager, який дає змогу встановити Raspberry Pi OS на microSD-карту та одразу задати базові параметри: ім'я пристрою, обліковий запис користувача, локалізацію, підключення до Wi-Fi та параметри віддаленого доступу. В офіційній документації Raspberry Pi зазначено, що Raspberry Pi Imager використовується для запису операційної системи на завантажувальний носій і первинного налаштування основних параметрів системи.

Для роботи з PiDog бажано використовувати актуальну 64-бітну версію Raspberry Pi OS, оскільки вона забезпечує кращу сумісність із сучасними бібліотеками Python, мережевими сервісами та програмними компонентами, які застосовуються для роботи з робототехнічною платформою. Після запису образу операційної системи microSD-карта встановлюється у Raspberry Pi, після чого плата підключається до живлення, монітора або мережі. На цьому етапі перевіряється успішне завантаження системи, робота Wi-Fi або Ethernet-з'єднання, доступ до терміналу та коректність системного часу.

Після першого запуску Raspberry Pi необхідно виконати оновлення системних пакетів. Це дозволяє отримати актуальні версії бібліотек, драйверів і служб, які потрібні для стабільної роботи апаратних модулів. У практичній реалізації оновлення системи доцільно виконувати перед встановленням бібліотек SunFounder, оскільки застарілі пакети можуть спричинити помилки під час інсталяції Python-модулів або роботи з інтерфейсами GPIO, I2C, I2S та камерою.

Після оновлення системи виконується встановлення програмного забезпечення SunFounder PiDog. Офіційна документація SunFounder передбачає послідовне встановлення кількох програмних компонентів: бібліотеки robot-hat для взаємодії з платою Robot HAT, бібліотеки vilib для роботи з камерою та візуальними функціями, модуля pidog для керування робот-собакою, а також скрипта i2samp.sh для налаштування I2S-аудіопідсилювача. У документації окремо зазначено, що без виконання скрипта i2samp.sh робот може не відтворювати звук.

Програмне середовище Raspberry Pi включає операційну систему Raspberry Pi OS, Python 3, бібліотеки SunFounder, модулі для роботи з Robot HAT, засоби керування сервоприводами, бібліотеки для обробки зображень, аудіомодулі та мережеві компоненти для взаємодії з LLM. Такий склад програмного забезпечення забезпечує можливість запуску базових рухових сценаріїв PiDog, перевірки сенсорів, роботи з камерою, відтворення звуку та обробки команд користувача.

Для коректної роботи апаратної частини необхідно активувати відповідні інтерфейси Raspberry Pi. Насамперед використовується I2C, через який може здійснюватися взаємодія з окремими сенсорними модулями та платою розширення. Для камери застосовується CSI-інтерфейс або USB-з'єднання залежно від конкретного модуля. Для звукових компонентів використовується I2S, оскільки Robot HAT підтримує аудіомодулі та динамік. GPIO-лінії використовуються для взаємодії з окремими цифровими сенсорами, кнопками або допоміжними сигналами. У документації SunFounder для PiDog зазначено, що Robot HAT є платою розширення для Raspberry Pi, яка поєднує апаратні можливості керування, аудіо, живлення та підключення периферійних пристроїв.

Особливу роль у системі відіграє плата Robot HAT. Вона є проміжним апаратним вузлом між Raspberry Pi та периферійними модулями PiDog. Через неї здійснюється підключення сервоприводів, сенсорів, звукових модулів і живлення. У версії Robot HAT V4 передбачено розширення PWM-виходів, ADC-виходів, драйвер двигунів, I2S-аудіомодуль, динамік, GPIO-роз'єми, індикатори живлення та підтримку вхідного живлення 7-12 В. Для PiDog V2 використовується оновлена плата Robot HAT V5, яка краще узгоджується з новішими моделями Raspberry Pi та має додаткові можливості для аудіо й голосової взаємодії.

Після встановлення програмних модулів виконується перевірка сервоприводів. Це один із найважливіших етапів, оскільки неправильне початкове положення або помилкове підключення сервоприводів може призвести до некоректного руху кінцівок. Для PiDog передбачено процедури калібрування, під час яких користувач налаштовує положення окремих сервоприводів, перевіряє симетричність кінцівок і контролює плавність руху. Офіційна документація SunFounder описує калібрування PiDog із використанням спеціальної калібрувальної лінійки та налаштуванням положення сервоприводів для забезпечення коректної геометрії платформи.

Підключення периферійних модулів здійснюється відповідно до їхнього функціонального призначення. Камера встановлюється у передній частині

робота та підключається до Raspberry Pi через CSI або USB. Вона використовується для візуального сприйняття середовища, спостереження за простором перед роботом і подальшої реалізації алгоритмів комп'ютерного зору. Ультразвуковий датчик відстані встановлюється на передній частині моделі та застосовується для визначення наявності перешкод. IMU-модуль розміщується всередині або на жорсткій частині корпусу для оцінювання нахилу, прискорення та руху платформи. Мікрофон і динамік забезпечують аудіо взаємодію з користувачем: приймання голосових команд, озвучення відповідей і повідомлення про стан системи.

Як показано в таблиці 3.2, кожен периферійний модуль має бути перевірений окремо перед запуском повного сценарію роботи робота

Таблиця 3.2 – Призначення периферійних модулів

Периферійний модуль	Спосіб підключення	Призначення в системі	Результат перевірки
Сервоприводи кінцівок	Через Robot HAT	Формування руху квадропедної платформи	Перевірка початкового положення, калібрування, тест ходи
Камера	CSI або USB	Отримання зображення та відеоданих	Перевірка запуску камери та отримання кадру
Ультразвуковий датчик	Через роз'єм Robot HAT / GPIO	Виявлення перешкод перед роботом	Перевірка вимірювання відстані
IMU-модуль	I2C	Контроль нахилу, прискорення та орієнтації корпусу	Перевірка зміни показників під час нахилу моделі
Мікрофон	I2S або вбудований модуль Robot HAT	Приймання голосових команд	Перевірка запису або розпізнавання звуку
Динамік	I2S-аудіомодуль / Robot HAT	Відтворення звукових повідомлень	Перевірка відтворення тестового аудіо
Мережевий інтерфейс	Wi-Fi або Ethernet	Підключення до локальної мережі та LLM/API	Перевірка доступу до мережі та віддаленого керування

Такий підхід зменшує ймовірність помилок, оскільки дозволяє локалізувати несправність у конкретному модулі. Наприклад, якщо робот не виконує команду руху, потрібно окремо перевірити живлення, підключення сервоприводів, роботу Robot HAT, правильність калібрування та програмний

виклик відповідної дії. Якщо не працює аудіо вихід, необхідно перевірити встановлення I2S – компонентів і виконання скрипта налаштування аудіо модуля.

Окремий етап налаштування пов'язаний із перевіркою мережевої взаємодії. Для інтеграції LLM Raspberry Pi повинен мати доступ до локальної мережі або інтернету. У хмарному варіанті Raspberry Pi передає текстову команду до API, отримує відповідь і перетворює її на допустиму команду для роботи. У локальному варіанті може використовуватися окремий сервер або полегшена модель, розміщена в локальній мережі. У будь-якому випадку перед інтеграцією LLM потрібно переконатися, що Raspberry Pi має стабільне мережеве з'єднання, коректно працює DNS, доступні потрібні Python-бібліотеки, а обробка команд не блокує виконання базових рухових функцій.

Узагальнену послідовність налаштування Raspberry Pi та програмного середовища для SunFounder PiDog подано на рисунку 3.4.



Рисунок 3.4 – Послідовність налаштування Raspberry Pi та програмного середовища для SunFounder PiDog

Після завершення налаштування Raspberry Pi і встановлення програмного середовища платформа SunFounder PiDog стає готовою до запуску базових програм керування. На цьому етапі важливо перевірити не лише факт виконання команд, а й стабільність роботи системи: чи не перезавантажується Raspberry Pi під час руху, чи немає ривків сервоприводів, чи коректно працює камера, чи змінюються показники IMU під час нахилу, чи спрацьовує ультразвуковий датчик перед перешкодою, чи відтворюється звук через динамік.

Налаштування Raspberry Pi для роботи з SunFounder PiDog передбачає підготовку операційної системи, встановлення бібліотек robot-hat, vilib і pidog, налаштування I2S-аудіо, підключення периферійних модулів, перевірку сенсорів, калібрування сервоприводів і запуск тестових прикладів. Виконання цих дій забезпечує перехід від зібраної фізичної моделі до працездатної апаратно-програмної платформи, готової до реалізації алгоритмів руху, обробки команд користувача та подальшої інтеграції великої мовної моделі в систему керування.

3.3 Реалізація програмного керування рухом квадроцепного робота

Реалізація програмного керування рухом квадроцепного є одним із ключових етапів практичної частини роботи, оскільки саме програмний рівень забезпечує перетворення команд користувача або LLM-модуля у конкретні рухові дії платформи. На відміну від колісних роботів, де рух формується переважно через зміну швидкості обертання двигунів, квадроцепна платформа потребує синхронізованого керування кількома сервоприводами, які відповідають за положення окремих кінцівок. Тому програмне забезпечення має враховувати початкове положення сервоприводів, послідовність переміщення кінцівок, обмеження кутів, часові затримки між фазами руху та стан сенсорних модулів.

У дагій системі Raspberry Pi виконує роль центрального обчислювального модуля, який запускає програмні сценарії руху, приймає команди користувача, взаємодіє з бібліотеками керування SunFounder PiDog, аналізує дані від сенсорів і передає керуючі сигнали до виконавчої частини. Безпосереднє керування

сервоприводами здійснюється через плату Robot HAT, яка забезпечує апаратний зв'язок між Raspberry Pi та сервоприводами кінцівок. Такий підхід дає змогу розділити високорівневу логіку руху і низькорівневе формування сигналів для виконавчих механізмів.

Програмне керування рухом квадропедної платформи реалізується за модульним принципом. Основними програмними модулями є модуль ініціалізації робота, модуль калібрування сервоприводів, модуль виконання базових рухів, модуль обробки команд, модуль перевірки сенсорних даних і модуль безпечної зупинки. Така структура спрощує налагодження системи, оскільки кожен функціональний блок можна перевіряти окремо. Наприклад, перед запуском складного сценарію руху спочатку перевіряється правильність положення сервоприводів, потім - реакція окремих кінцівок, після цього - виконання базової ходи, і лише після цього - інтеграція з командами користувача або LLM.

Загальна структура програмного керування рухом квадропедного робота подана на рисунку 3.5.

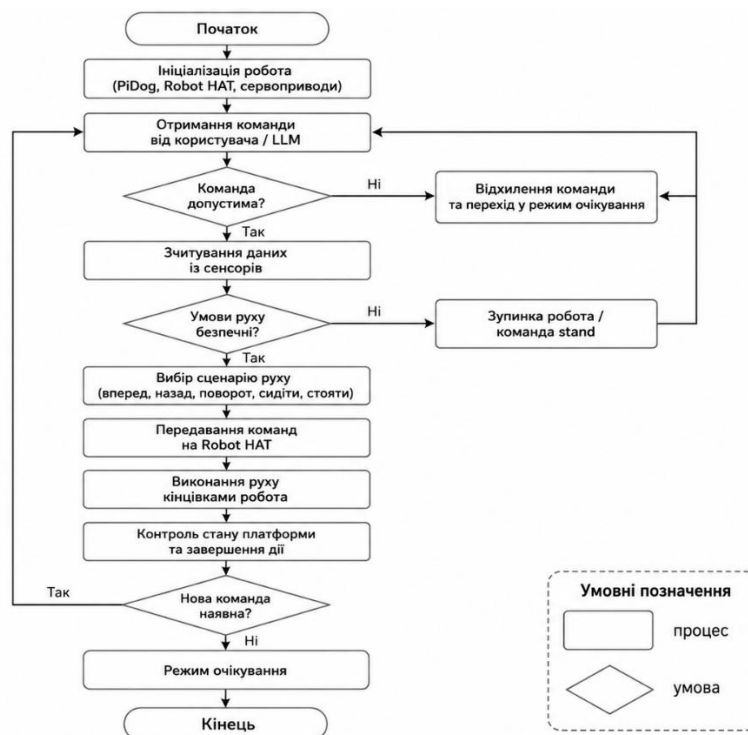


Рисунок 3.5 –Блок-схема програмного керування рухом квадропедного робота

На початку роботи програми виконується імпорт необхідних бібліотек, створення об'єкта робота та переведення платформи у початковий стан. Початковий стан є важливим, оскільки всі подальші рухи виконуються відносно базової позиції. Для квадроцепного робота такою позицією може бути положення стояння, у якому всі чотири кінцівки розташовані симетрично, корпус перебуває у стабільному положенні, а центр мас знаходиться в межах опорного контуру. Якщо початкове положення визначено некоректно, подальша хода може супроводжуватися перекосом корпуса або нерівномірним навантаженням на сервоприводи.

Фрагмент базової логіки ініціалізації програмного керування можна подати як лістинг 3.1.

Лістинг 3.1 – Код базової логіки ініціалізації програмного керування

```
from pidog import Pidog
import time

robot = Pidog()

def init_robot():
    robot.do_action("stand", speed=70)
    time.sleep(1)
    robot.wait_all_done()

init_robot()
```

кінець лістингу 3.1

У наведеному фрагменті створюється об'єкт `Pidog`, після чого робот переводиться у положення стояння. Команда `wait_all_done()` використовується для очікування завершення попередньої дії перед переходом до наступного етапу. Це важливо для уникнення ситуацій, коли нова команда надходить до того, як попередній рух повністю завершився.

Базові рухові дії квадроцепного робота реалізуються через набір попередньо визначених сценаріїв. До таких дій належать рух уперед, рух назад, поворот ліворуч, поворот праворуч, зупинка, перехід у положення сидіння, стояння або демонстраційний режим. У межах бакалаврської роботи доцільно

реалізувати саме ці базові сценарії, оскільки вони дають змогу перевірити працездатність механічної частини, правильність підключення сервоприводів і стабільність програмного керування.

Таблиця 3.3 – Основні програмні команди керування рухом

Команда	Призначення	Очікуваний результат
stand	Переведення робота у положення стояння	Робот займає стабільну початкову позицію
sit	Перехід у положення сидіння	Корпус опускається, кінцівки змінюють положення
forward	Рух уперед	Робот виконує послідовну ходу вперед
backward	Рух назад	Робот переміщується назад
turn_left	Поворот ліворуч	Платформа змінює напрямок руху вліво
turn_right	Поворот праворуч	Платформа змінює напрямок руху вправо
stop	Зупинка руху	Припинення виконання поточного сценарію
bark або speak	Звукова реакція	Відтворення звукового повідомлення або сигналу

Як видно з таблиці 3.3, програмні команди мають бути простими, зрозумілими та прив'язаними до конкретних дій робота. Це особливо важливо для подальшої інтеграції з LLM, оскільки мовна модель не повинна напямку керувати сервоприводами, а має лише перетворювати природномовний запит користувача у одну з дозволених формалізованих команд.

Приклад реалізації функції виконання базових рухових команд наведено на лістингу 3.2.

Лістинг 3.2 – Код реалізації функції виконання базових рухових команд

```
def execute_motion(command):
    if command == "MOVE_FORWARD":
        robot.do_action("forward", speed=60)
    elif command == "MOVE_BACKWARD":
        robot.do_action("backward", speed=60)
    elif command == "TURN_LEFT":
        robot.do_action("turn_left", speed=55)
    elif command == "TURN_RIGHT":
        robot.do_action("turn_right", speed=55)
    elif command == "SIT":
        robot.do_action("sit", speed=60)
    elif command == "STAND":
        robot.do_action("stand", speed=60)
```

```

elif command == "STOP":
    robot.stop()
else:
    robot.do_action("stand", speed=50)

robot.wait_all_done()
import time

robot = Pidog()

def init_robot():
    robot.do_action("stand", speed=70)
    time.sleep(1)
    robot.wait_all_done()

init_robot()

```

кінець лістингу 3.2

У цьому фрагменті реалізовано перетворення формалізованої команди у відповідну дію робота. Якщо команда не належить до дозволеного набору, робот не виконує небезпечних рухів, а переходить у стабільне положення стояння. Такий підхід є важливим з погляду безпеки, оскільки дозволяє уникнути непередбачуваної поведінки у випадку помилкової команди або некоректної відповіді LLM.

Для підвищення надійності руху в програмному забезпеченні доцільно реалізувати перевірку сенсорних даних перед виконанням команди. Наприклад, перед рухом уперед система повинна перевірити відстань до перешкоди за допомогою ультразвукового датчика. Якщо перешкода розташована надто близько, команда руху не виконується, а робот переходить у режим очікування або повідомляє користувача про неможливість руху. Такий механізм дає змогу поєднати програмне керування сервоприводами з елементами автономної поведінки.

Приклад логіки перевірки перешкоди перед рухом подано в лістингу 3.3.

Лістинг 3.3 – Код реалізації функції виконання базових рухових команд

```
SAFE_DISTANCE_CM = 25
```

```
def can_move_forward(distance_cm):
```

```

if distance_cm is None:
    return False
return distance_cm > SAFE_DISTANCE_CM

def move_forward_safe(distance_cm):
    if can_move_forward(distance_cm):
        robot.do_action("forward", speed=60)
        robot.wait_all_done()
    else:
        robot.stop()
        robot.do_action("stand", speed=50)
        robot.wait_all_done()

```

кінець лістингу 3.3

У наведеному прикладі рух уперед дозволяється лише тоді, коли відстань до перешкоди перевищує заданий безпечний поріг. Якщо сенсор не повертає коректного значення або перешкода розміщена близько, робот не починає рух. У практичній реалізації цей підхід може бути доповнений голосовим повідомленням через динамік або виведенням повідомлення у консоль.

Окрім перевірки відстані, важливим є контроль положення корпусу. Для цього може використовуватися IMU-модуль, який дає змогу оцінювати нахил платформи. Якщо нахил перевищує допустиме значення, програма повинна зупинити рух або перейти у стабілізаційний режим. Для бакалаврського прототипу достатньо реалізувати просту перевірку граничного кута нахилу, без складного динамічного керування рівновагою.

Узагальнений алгоритм програмного керування рухом квадроцепдного робота можна описати так: система отримує команду, перевіряє її належність до дозволеного набору, аналізує дані сенсорів, визначає можливість виконання дії, запускає відповідний руховий сценарій, контролює завершення дії та повертає платформу до стабільного стану. Такий алгоритм забезпечує передбачувану й безпечну поведінку робота.

У програмній реалізації важливо уникати одночасного запуску кількох рухових команд. Наприклад, якщо робот виконує поворот, а в цей момент надходить нова команда руху вперед, система повинна або дочекатися завершення поточної дії, або коректно її зупинити. Для цього в програмі доцільно

використовувати станову модель, у якій робот може перебувати в одному з режимів: «очікування», «рух», «поворот», «зупинка», «помилка», «безпечний режим». Станова модель дозволяє краще контролювати логіку виконання команд і уникати конфліктів між діями.

Окрему увагу потрібно приділити параметру швидкості. Надто висока швидкість руху може спричинити ривки, втрату стійкості або підвищене навантаження на сервоприводи. Тому на етапі тестування доцільно використовувати помірні значення швидкості, поступово збільшуючи їх лише після перевірки стабільності конструкції. Для навчального прототипу оптимальним є плавний режим руху, який дає змогу спостерігати за роботою кінцівок і своєчасно виявляти помилки в калібруванні.

Програмне керування рухом квадроцепного робота реалізується як поєднання готових бібліотечних функцій SunFounder PiDog і власної логіки обробки команд. Готові функції використовуються для виконання базових рухів, а власний програмний рівень відповідає за приймання команд, перевірку безпеки, вибір сценарію та підготовку до інтеграції з LLM. Такий підхід є доцільним, оскільки дозволяє не дублювати низькорівневі алгоритми керування сервоприводами, а зосередитися на створенні інтелектуального рівня взаємодії з користувачем.

Загалом програмне керування рухом квадроцепного робота має забезпечити три основні вимоги: передбачуваність, безпечність і розширюваність. Передбачуваність означає, що кожна команда повинна мати визначений результат. Безпечність передбачає перевірку команд і сенсорних умов перед виконанням руху. Розширюваність означає можливість додавання нових сценаріїв, наприклад обходу перешкоди, патрулювання, реакції на голосову команду або взаємодії з LLM.

3.4 Тестування роботи платформи

Після складання фізичної моделі квадропедної роботизованої платформи SunFounder PiDog, налаштування Raspberry Pi, встановлення програмного середовища та реалізації базових алгоритмів керування було виконано тестування працездатності системи. Метою тестування було визначення коректності роботи основних апаратно-програмних компонентів, перевірка виконання рухових команд, оцінювання стабільності положення робота, реакції на команди користувача та автономності роботи від акумуляторного живлення.

Тестування виконувалося поетапно, оскільки квадропедна платформа складається з кількох взаємопов'язаних підсистем: механічної конструкції, сервоприводів, плати Robot HAT, Raspberry Pi, сенсорних модулів, аудіокомпонентів і програмного забезпечення. Такий підхід дав змогу окремо перевірити кожен функціональний блок і лише після цього перейти до комплексного випробування всієї системи. Основна увага приділялася тому, щоб робот коректно переходив у початкове положення, виконував команди руху, не втрачав стійкість під час переміщення, реагував на текстові або голосові команди та міг працювати автономно протягом визначеного часу.

На початковому етапі було перевірено правильність запуску Raspberry Pi та програмного середовища. Після подавання живлення система завантажувала Raspberry Pi OS, після чого запускалося програмне забезпечення для керування PiDog. Перевірялося підключення до Robot HAT, доступність сервоприводів, коректність роботи встановлених бібліотек, а також можливість виконання базових команд керування. У разі успішного запуску робот переходив у положення «stand», що використовувалося як початкова стабільна позиція для подальших випробувань.

Для перевірки руху було протестовано виконання базових команд: стояння, сидіння, рух уперед, рух назад, поворот ліворуч, поворот праворуч і зупинка. Кожна команда запускалася декілька разів, щоб оцінити повторюваність руху та відсутність випадкових збоїв. Особлива увага приділялася симетричності роботи

кінцівок, плавності переміщення сервоприводів, відсутності ривків і правильності завершення рухової дії. Якщо після виконання команди робот повертався у стабільне положення без перекосу корпусу, дія вважалася виконаною коректно.

Як показано в таблиці 3.4, для кожної команди було визначено очікувану поведінку платформи та критерій успішного виконання. Такий підхід дав змогу не лише перевірити факт запуску руху, а й оцінити якість його виконання. Наприклад, команда руху вперед вважалася успішною не тільки тоді, коли робот переміщувався, а й тоді, коли він зберігав стійкість, не просідав на одну зі сторін і не демонстрував неконтрольованих коливань корпусу.

Таблиця 3.4 – Перевірка виконання базових рухових команд квадроопедної платформи

Команда	Очікувана дія платформи	Критерій успішного виконання
stand	Перехід у положення стояння	Корпус вирівняний, усі кінцівки мають симетричне положення
sit	Перехід у положення сидіння	Робот плавно змінює положення без ривків
forward	Рух уперед	Платформа переміщується вперед із збереженням стійкості
backward	Рух назад	Робот виконує контрольований рух назад
turn_left	Поворот ліворуч	Корпус змінює напрямок без втрати рівноваги
turn_right	Поворот праворуч	Робот повертає праворуч із коректною роботою кінцівок
stop	Зупинка руху	Поточна дія припиняється, робот переходить у безпечний стан

Окремо перевірялася стабільність платформи під час стояння та руху. Стабільність оцінювалася за положенням корпусу, рівномірністю навантаження на кінцівки, відсутністю ковзання опорних елементів і здатністю робота завершити рухову команду без падіння або значного перекосу. Для цього робот встановлювався на рівну поверхню, після чого виконувались короткі цикли руху вперед, поворотів і повернення у початкове положення. У процесі тестування було встановлено, що стабільність значною мірою залежить від правильності калібрування сервоприводів, рівня заряду акумулятора та швидкості виконання команд.

На рисунку 3.6 доцільно подати загальну схему тестування роботи квадропедної платформи.

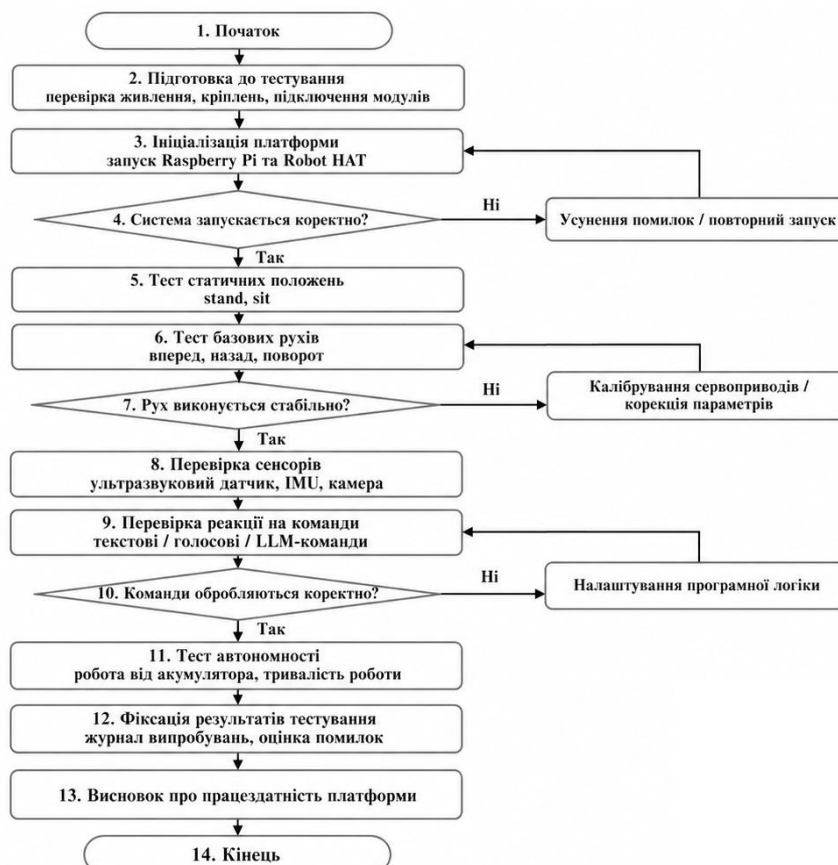


Рисунок 3.6 – Схема тестування роботи квадропедної роботизованої платформи

До основних чинників, які впливали на стабільність руху, належали точність початкового положення кінцівок, синхронність роботи сервоприводів, рівномірність розміщення маси компонентів на корпусі та якість поверхні, на якій проводилося тестування. Якщо сервоприводи були відкалібровані неправильно, робот міг нахилитися в один бік або нерівномірно переносити вагу під час ходи. Тому перед основними тестами виконувалося повторне калібрування положення кінцівок і перевірка роботи кожного сервоприводу.

Реакція на команди перевірялася шляхом подавання керуючих інструкцій через програмний інтерфейс. На першому етапі використовувалися прості текстові команди, які вводилися вручну або передавалися через консольний

інтерфейс. Далі тестувалася можливість перетворення команди користувача у формалізовану дію, наприклад «вперед» – MOVE_FORWARD, «назад» – MOVE_BACKWARD, «зупинись» – STOP. Така перевірка є важливою для подальшої інтеграції LLM, оскільки мовна модель повинна формувати не довільні інструкції, а лише ті команди, які підтримуються програмним забезпеченням роботи.

З таблиці 3.6 видно, що кожна природномовна команда повинна бути приведена до службового формату. Це зменшує ризик некоректного виконання дій і дозволяє відокремити рівень мовної взаємодії від рівня безпосереднього керування сервоприводами. Якщо команда не входила до переліку дозволених дій, система не виконувала рух, а переводила робота в режим очікування або положення «stand». Такий механізм є необхідним для безпечної роботи платформи, особливо в разі використання LLM, яка може інтерпретувати запити користувача по-різному.

Таблиця 3.6 – Перевірка реакції робота на команди користувача

Вхідна команда користувача	Формалізована команда	Очікувана реакція робота
«Рухайся вперед»	MOVE_FORWARD	Запуск руху вперед
«Поверни ліворуч»	TURN_LEFT	Виконання повороту ліворуч
«Поверни праворуч»	TURN_RIGHT	Виконання повороту праворуч
«Сядь»	SIT	Перехід у положення сидіння
«Стань»	STAND	Перехід у положення стояння
«Зупинись»	STOP	Припинення поточного руху
«Оглянь простір»	SCAN	Перевірка сенсорів або запуск огляду

Під час тестування сенсорної частини перевірялася робота ультразвукового датчика відстані, IMU-модуля та камери. Ультразвуковий датчик використовувався для виявлення перешкод перед роботом. Якщо відстань до об'єкта була меншою за встановлений безпечний поріг, команда руху вперед блокувалася або виконувалася з подальшою зупинкою. IMU-модуль застосовувався для контролю нахилу корпусу. Якщо під час руху фіксувався значний нахил, система могла перейти в режим зупинки або очікування. Камера

перевірялася шляхом отримання зображення та підтвердження її готовності до використання в подальших сценаріях комп'ютерного зору.

Окремим напрямом тестування була перевірка автономності роботи. Оскільки квадрокоптерний робот є мобільною платформою, він повинен працювати від акумуляторного живлення без постійного підключення до зовнішнього джерела. Під час перевірки автономності оцінювався час роботи системи в режимі очікування, час роботи під час періодичного виконання рухових команд і стабільність живлення під час одночасної роботи сервоприводів. Було враховано, що найбільше енергоспоживання виникає під час активного руху, коли кілька сервоприводів одночасно змінюють положення та утримують масу конструкції.

Загалом результати тестування підтвердили працездатність квадрокоптерної роботизованої платформи SunFounder PiDog як апаратно-програмної системи для реалізації базових рухів, сенсорного контролю та взаємодії з користувачем. Платформа коректно виконує основні рухові команди, переходить у стабільні положення, реагує на команди користувача, підтримує роботу сенсорних модулів і може функціонувати в автономному режимі від акумулятора. При цьому якість роботи системи значною мірою залежить від правильності складання, калібрування сервоприводів, стабільності живлення та коректності програмної логіки керування.

ВИСНОВКИ

У кваліфікаційній роботі було розглянуто особливості побудови інтелектуальної квадропедної роботизованої платформи на базі Raspberry Pi з використанням великої мовної моделі. У процесі виконання роботи встановлено, що квадропедні роботизовані системи є перспективним напрямом розвитку мобільної робототехніки, оскільки вони мають кращу адаптивність до складного рельєфу порівняно з колісними платформами, забезпечують гнучке переміщення за рахунок опорних кінцівок і можуть використовуватися як основа для реалізації інтелектуальної взаємодії з користувачем.

У роботі проаналізовано можливості використання Raspberry Pi у складі мобільних роботизованих систем. Визначено, що Raspberry Pi доцільно застосовувати як центральний обчислювальний модуль, оскільки він підтримує операційну систему Linux, мову програмування Python, роботу з камерою, мережевими інтерфейсами, сенсорними модулями та зовнішніми програмними сервісами. Це дає змогу використовувати платформу не лише для керування рухом, а й для обробки команд, аналізу сенсорних даних і взаємодії з LLM - модулем.

Досліджено принципи керування рухом квадропедного робота та особливості координації сервоприводів. Встановлено, що для забезпечення стабільної ходи необхідно враховувати початкове положення кінцівок, синхронізацію рухів, послідовність фаз переміщення, обмеження кутів повороту сервоприводів і стійкість корпусу. Для практичної реалізації найбільш доцільним є використання готових бібліотечних функцій SunFounder PiDog у поєднанні з власною логікою обробки команд і перевірки безпечності руху.

У ході роботи обґрунтовано вибір апаратних компонентів для створення роботизованої платформи. Як фізичну основу використано робот-собаку SunFounder PiDog, що містить механічну конструкцію квадропедної платформи, сервоприводи, плату Robot HAT, сенсорні модулі, камеру, аудіокомпоненти та систему живлення. Такий вибір є доцільним для бакалаврської роботи, оскільки

комплект забезпечує готову апаратну базу для дослідження руху, калібрування сервоприводів, підключення периферійних модулів і реалізації інтелектуальної взаємодії.

Розроблено структурну та принципову схеми взаємодії основних компонентів системи. У структурній схемі визначено функціональні блоки платформи: користувацький інтерфейс, LLM-модуль, Raspberry Pi, Robot HAT, сенсорний блок, аудіокомунікаційний блок, сервоприводи та систему живлення. У принциповій схемі відображено порядок підключення компонентів, зокрема лінії живлення, сигнальні інтерфейси, підключення сенсорів, камери, сервоприводів і допоміжних модулів. Це дало змогу сформувати цілісну апаратно-програмну архітектуру платформи.

Запропоновано алгоритм роботи платформи з урахуванням сенсорних даних, команд користувача та взаємодії з LLM. Передбачено, що велика мовна модель використовується на високому рівні керування для інтерпретації природномовних команд, тоді як безпосереднє керування рухом виконується локальними алгоритмами Raspberry Pi та Robot HAT. Такий розподіл функцій підвищує безпечність системи, оскільки LLM не керує сервоприводами напряму, а лише формує або уточнює сценарій дії.

Реалізовано програмне забезпечення для керування рухом, обробки команд і виконання базових сценаріїв роботи квадропедної платформи. Програмна частина передбачає ініціалізацію робота, переведення його у початковий стан, виконання команд руху вперед, назад, повороту, зупинки, переходу в положення стояння або сидіння, а також можливість перевірки допустимості команд. Така логіка створює основу для подальшого розширення системи за рахунок голосового керування, обробки команд через LLM та використання складніших сценаріїв поведінки.

Проведено тестування працездатності платформи, у межах якого перевірено запуск Raspberry Pi, роботу Robot HAT, виконання базових рухових команд, стабільність положення корпусу, реакцію на команди користувача, функціонування сенсорних модулів і автономну роботу від акумулятора.

Результати тестування підтвердили, що платформа здатна виконувати основні рухові дії, переходити у стабільні положення, реагувати на команди та функціонувати як єдина апаратно-програмна система.

У результаті виконання кваліфікаційної роботи було розроблено та досліджено інтелектуальну квадроцепдну роботизовану платформу на базі Raspberry Pi з можливістю інтеграції LLM для обробки команд користувача. Запропоноване рішення має практичне значення як навчальний і дослідницький прототип, що може використовуватися для вивчення принципів мобільної робототехніки, керування сервоприводами, роботи з сенсорними модулями, програмування на Python та впровадження елементів штучного інтелекту в роботизовані системи.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A survey of the development of quadruped robots: joint configuration, dynamic locomotion control method and mobile manipulation approach. *Biomimetic Intelligence and Robotics*. URL: <https://www.sciencedirect.com/science/article/pii/S2667379721000292> (date of access: 15.01.2026).
2. Grounding Language in Robotic Affordances. *arXiv*. URL: <https://arxiv.org/abs/2204.01691> (date of access: 16.01.2026).
3. An Embodied Multimodal Language Model. *arXiv*. URL: <https://arxiv.org/abs/2303.03378> (date of access: 17.01.2026).
4. Large Language Models for Robotics: A Survey. *arXiv*. URL: <https://arxiv.org/abs/2311.07226> (date of access: 18.01.2026).
5. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. *Proceedings of The 7th Conference on Robot Learning. Proceedings of Machine Learning Research*. URL: <https://proceedings.mlr.press/v229/zitkovich23a.html> (date of access: 19.01.2026).
6. A survey on legged robots: advances, technologies and applications. *Engineering Applications of Artificial Intelligence*. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0952197624015768> (date of access: 20.01.2026).
7. Review of Quadruped Robots: Structure, Control, and Autonomous Motion. *Advanced Intelligent Systems*. URL: <https://advanced.onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202300783> (date of access: 22.01.2026).
8. Quadruped Robot's Gait Planning and Kinematic Analysis. *Applied and Computational Engineering*. URL: <https://www.ewadirect.com/proceedings/ace/article/view/17591> (date of access: 25.01.2026).
9. Quadruped Robot Control: An Approach Using Body Planar Motion Control, Legs Impedance Control, and Bezier Curves. *Sensors*. URL: <https://www.mdpi.com/1424-8220/24/12/3825> (date of access: 28.01.2026).

10. Gait Planning, and Motion Control Methods for Quadruped Robots: Achieving High Environmental Adaptability: A Review. *Computer Modeling in Engineering & Sciences*. URL: <https://www.techscience.com/CMES/v143n1/60463> (date of access: 01.02.2026).
11. Enhancing Quadruped Robot Walking on Unstructured Terrains: A Combination of Stable Blind Gait and Deep Reinforcement Learning. *Electronics*. URL: <https://www.mdpi.com/2079-9292/14/7/1431> (date of access: 04.02.2026).
12. PALo: Learning Posture-Aware Locomotion for Quadruped Robots. *arXiv*. URL: <https://arxiv.org/abs/2503.04462> (date of access: 07.02.2026).
13. Quadruped Robots: Bridging Mechanical Design, Control, and Autonomous Motion. *Robotics*. URL: <https://www.mdpi.com/2218-6581/14/5/57> (date of access: 10.02.2026).
14. Boston Dynamics. *Spot – The Agile Mobile Robot*. URL: <https://bostondynamics.com/products/spot/> (date of access: 12.02.2026).
15. Raspberry Pi Ltd. *Raspberry Pi 4 Model B specifications*. URL: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/> (date of access: 15.02.2026).
16. Raspberry Pi Ltd. *Raspberry Pi 4 Model B Datasheet*. URL: <https://pip.raspberrypi.com/documents/RP-008341-DS-raspberry-pi-4-datasheet.pdf> (date of access: 18.02.2026).
17. Raspberry Pi Ltd. *Raspberry Pi 5 Product Brief*. URL: <https://pip.raspberrypi.com/documents/RP-008348-DS-raspberry-pi-5-product-brief.pdf> (date of access: 20.02.2026).
18. Raspberry Pi Ltd. *Raspberry Pi 5*. URL: <https://www.raspberrypi.com/products/raspberry-pi-5/> (date of access: 23.02.2026).
19. Raspberry Pi Ltd. *Raspberry Pi Camera Module 3 Product Brief*. URL: <https://pip-assets.raspberrypi.com/categories/786-raspberry-pi-camera-module-3/documents/RP-008151-DS-1-camera-module-3-product-brief.pdf> (date of access: 25.02.2026).

20. Raspberry Pi Ltd. Camera. *Raspberry Pi Documentation*. URL: <https://www.raspberrypi.com/documentation/accessories/camera.html> (date of access: 28.02.2026).

21. Raspberry Pi Ltd. Camera software. *Raspberry Pi Documentation*. URL: https://www.raspberrypi.com/documentation/computers/camera_software.html (date of access: 03.03.2026).

22. Raspberry Pi Ltd. *Raspberry Pi computer hardware documentation*. URL: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html> (date of access: 06.03.2026).

23. OpenCV. *OpenCV documentation*. URL: <https://docs.opencv.org/> (date of access: 09.03.2026).

24. Open Robotics. ROS 2 on Raspberry Pi. *ROS 2 Documentation*. URL: <https://docs.ros.org/en/foxy/How-To-Guides/Installing-on-Raspberry-Pi.html> (date of access: 12.03.2026).

25. SunFounder Raspberry Pi Robot – PiDog. *SunFounder*. URL: <https://docs.sunfounder.com/projects/pidog/en/latest/> (date of access: 15.03.2026).

26. PiDog Hardware Components. *SunFounder*. URL: https://docs.sunfounder.com/projects/pidog/en/latest/hardware/cpn_hardware.html (date of access: 18.03.2026).

27. Robot HAT. PiDog Documentation. *SunFounder*. URL: https://docs.sunfounder.com/projects/pidog/en/latest/hardware/cpn_robot_hat.html (date of access: 21.03.2026).

28. Install All Modules. PiDog Documentation. *SunFounder*. URL: https://docs.sunfounder.com/projects/pidog/en/latest/python/python_start/install_all_modules.html (date of access: 24.03.2026).

29. Calibration. PiDog Documentation. *SunFounder*. URL: https://docs.sunfounder.com/projects/pidog/en/latest/python/py_0_calibrate.html (date of access: 27.03.2026).

30. PiDog: Raspberry Pi Robot Dog Python Library. *GitHub*. URL: <https://github.com/sunfounder/pidog> (date of access: 30.03.2026).

31. Adafruit Industries. Adafruit PCA9685 16-Channel Servo Driver: *Overview*. URL: <https://learn.adafruit.com/16-channel-pwm-servo-driver/overview> (date of access: 02.04.2026).
32. Adafruit 16-Channel 12-bit PWM/Servo Driver. *Adafruit Industries*. URL: <https://cdn-learn.adafruit.com/downloads/pdf/16-channel-pwm-servo-driver.pdf> (date of access: 05.04.2026).
33. Adafruit MAX98357 I2S Class-D Mono Amp: Overview. *Adafruit Industries*. URL: <https://learn.adafruit.com/adafruit-max98357-i2s-class-d-mono-amp/overview> (date of access: 10.04.2026).
34. Робот-собака SunFounder PiDog з інтегруванням ШІ ChatGPT-4o. *Arduino.ua*. URL: <https://arduino.ua/prod7538-robot-sobaka-sunfounder-pidog-z-integrovanim-shi-chatgpt-4o> (date of access: 20.04.2026).

