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 безпеки
(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

РОБОТИЗОВАНА МОБІЛЬНА ПЛАТФОРМА З ПРОГРАМНОЮ НАВІГАЦІЄЮ НА БАЗІ RASPBERRY PI

ROBOTIC MOBILE PLATFORM WITH AUTONOMOUS NAVIGATION BASED ON RASPBERRY PI

спеціальність 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія
(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІс-21
Митчик Максим Петрович

(підпис)

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

(підпис)

Кваліфікаційну роботу
допущено до захисту
« 04 » червня 2025 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2025 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
« 10 » 01 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Митчику Максиму Петровичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Роботизована мобільна платформа з автономною навігацією на базі Raspberry Pi*

Керівник роботи *к.т.н, доц. Гринюк Сергій Васильович*

затверджені наказом закладу вищої освіти від «01» квітня 2025 року № 244/01-07
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 10.06.2025р.

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

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Огляд літератури з дослідження автономної навігації роботизованих мобільних платформ
Розробка апаратної частини мобільної платформи
Програмна реалізація
Висновки

5. Перелік графічного (ілюстративного) матеріалу:

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Огляд літератури з дослідження автономної навігації роботизованих мобільних платформ	*Гринюк С.В., доцент*		
Розробка апаратної частини мобільної платформи	*Гринюк С.В., доцент*		
Програмна реалізація	*Гринюк С.В., доцент*		
Нормоконтроль	*Багнюк Н.В., доцент*		
Гарант ОП	*Лавренчук С.В., доцент*		
Показник запозичень тексту	***___%***		
Академічна доброчесність	*Міскевич О.І.,ст.викладач*		

7. Дата видачі завдання 08.04.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	*Огляд літератури з дослідження автономної навігації роботизованих мобільних платформ*	до 10.04.2025 р.	Виконано
2.	*Розробка апаратної частини мобільної платформи*	до 10.05.2025 р.	Виконано
3.	*Програмна реалізація*	до 20.05.2025 р.	Виконано
4.	*Висновки та пропозиції*	до 20.05.2025 р.	Виконано
5.	*Формування списку використаних джерел*	до 23.05.2025 р.	Виконано
6.	*Оформлення ілюстративного матеріалу*	до 10.05.2025 р.	Виконано
7.	*Представлення остаточного варіанту кваліфікаційної роботи керівнику*	до 15.05.2025 р.	Виконано
8.	*Нормоконтроль*	до 30.05.2025 р.	Виконано
9.	*Інструментальна перевірка на академічний плагіат*	до 03.06.2025 р.	Виконано
10.	*Представлення кваліфікаційної та всіх супровідних документів на кафедру*	до 10.06.2025 р.	Виконано

Здобувач вищої освіти _____ (підпис) Митчик М.П. (прізвище, ініціали)

Керівник кваліфікаційної роботи _____ (підпис) Гринюк С.В. (прізвище, ініціали)

АНОТАЦІЯ

Митчик М. П. Роботизована мобільна платформа з автономною навігацією на базі Raspberry Pi. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2025.

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

Перший розділ містить аналіз ринку сучасних роботизованих платформ, детальний огляд їхніх технічних характеристик і можливостей, а також дослідження існуючих алгоритмів автономної навігації та уникнення перешкод. Особлива увага приділяється порівнянню підходів до планування руху, включаючи глобальні й локальні алгоритми, з урахуванням їхньої придатності для роботи в умовах реального часу.

Другий розділ присвячено розробці апаратної частини платформи. Тут здійснюється обґрунтований вибір мікрокомп'ютера, сенсорів, сервоприводів, модулів живлення та додаткових компонентів. Докладно описано процес збирання прототипу, наведено особливості інтеграції компонентів, а також враховано питання стабільності роботи системи та можливості масштабування.

Третій розділ фокусується на програмній частині реалізації. Описано етапи встановлення та налаштування операційної системи на Raspberry Pi, підготовку програмного середовища Python, а також розробку й тестування програмного модуля, який забезпечує автономне ухилення мобільної платформи від перешкод. Наведено приклади алгоритмів, що використовуються, обговорено результати тестування, а також зроблено висновки щодо ефективності роботи платформи в умовах експериментального середовища.

Ключові слова: робототехніка, мобільна платформа, автономна навігація; Raspberry Pi, сенсорні системи, мікроконтролер, Python.

ANNOTATION

Mytchyk M. Robotic mobile platform with autonomous navigation based on Raspberry Pi. Manuscript.

Qualification work of the bachelor of the specialty «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

Qualification work consists of an introduction, three chapters, conclusions, a list of sources used.

The first chapter contains an analysis of the market of modern robotic platforms, a detailed review of their technical characteristics and capabilities, as well as a study of existing algorithms for autonomous navigation and obstacle avoidance. Special attention is paid to the comparison of approaches to motion planning, including global and local algorithms, taking into account their suitability for operation in real-time conditions.

The second chapter is devoted to the development of the platform hardware. Here, a reasoned choice of a microcomputer, sensors, servo drives, power modules and additional components is made. The process of assembling the prototype is described in detail, the features of component integration are given, and the issues of system stability and scalability are also taken into account.

The third section focuses on the software part of the implementation. The stages of installing and configuring the operating system on the Raspberry Pi, preparing the Python software environment, and developing and testing a software module that provides autonomous obstacle avoidance of the mobile platform are described. Examples of the algorithms used are given, the test results are discussed, and conclusions are drawn regarding the platform’s efficiency in the experimental environment.

Keywords: robotics, mobile platform, autonomous navigation; Raspberry Pi, sensor systems, microcontroller, Python

ЗМІСТ

ВСТУП

Робототехніка стрімко розвивається, відкриваючи нові можливості для автоматизації різноманітних сфер діяльності. Серед найцікавіших напрямків цього розвитку – створення мобільних платформ, здатних автономно орієнтуватися в середовищі, ухилятися від перешкод, аналізувати навколишній простір та приймати рішення в режимі реального часу. Така техніка вже застосовується у логістиці, на складах, у сільському господарстві, у військових розробках, а також у побуті (наприклад, роботи-прибиральники).

Основою для створення подібних систем є надійне обчислювальне ядро, яке може обробляти великі обсяги даних від сенсорів та камер, управляти рухом та забезпечувати стабільність роботи. Raspberry Pi, що представляє собою компактний мікрокомп’ютер, здатний виконувати завдання, характерні для повноцінних комп’ютерних систем, стає одним із найпопулярніших рішень для побудови роботизованих платформ. Його переваги — невисока ціна, широкий вибір аксесуарів, відкритий доступ до програмного забезпечення та активна спільнота розробників.

Зважаючи на те, що в багатьох сферах людської діяльності необхідно використовувати автономні системи (наприклад, для роботи у важкодоступних або небезпечних для людини умовах), тема розробки мобільних платформ з автономною навігацією є надзвичайно актуальною. Такі системи можуть застосовуватися для патрулювання, моніторингу територій, автоматизованої доставки вантажів, картографування території, пошуково-рятувальних операцій та навіть у розважальній індустрії.

Мета кваліфікаційної роботи полягає у створенні функціональної роботизованої мобільної платформи з автономною навігацією на базі Raspberry Pi, яка здатна пересуватися в просторі, виявляти та уникати перешкод, слідувати заданим маршрутам або формувати їх самостійно.

Об’єктом дослідження є роботизовані мобільні платформи.

Предметом дослідження є процеси розробки, програмування та тестування автономної навігації на базі Raspberry Pi.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– провести аналіз існуючих роботизованих платформ з автономною навігацією;

– обґрунтувати вибір апаратного та програмного забезпечення;

– розробити та зібрати апаратну частину системи;

– реалізувати алгоритми автономної навігації.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ АВТОНОМНОЇ НАВІГАЦІЇ РОБОТИЗОВАНИХ МОБІЛЬНИХ ПЛАТФОРМ

1.1 Аналіз ринку сучасних роботизованих платформ

Роботизовані мобільні платформи з автономною навігацією (AMR, англ. Autonomous Mobile Robots) – це самохідні системи, здатні орієнтуватися в просторі без прямого керування людиною. Вони використовують сенсори, камери, LiDAR, GPS та алгоритми штучного інтелекту для побудови карт, локалізації та планування маршрутів у реальному часі.

Згідно з дослідженням Grand View Research (рис. 1.1), світовий ринок AMR у 2024 році оцінювався в $4,07 млрд і очікується, що до 2030 року він зросте до $9,56 млрд зі щорічним приростом 15,1 % [1].

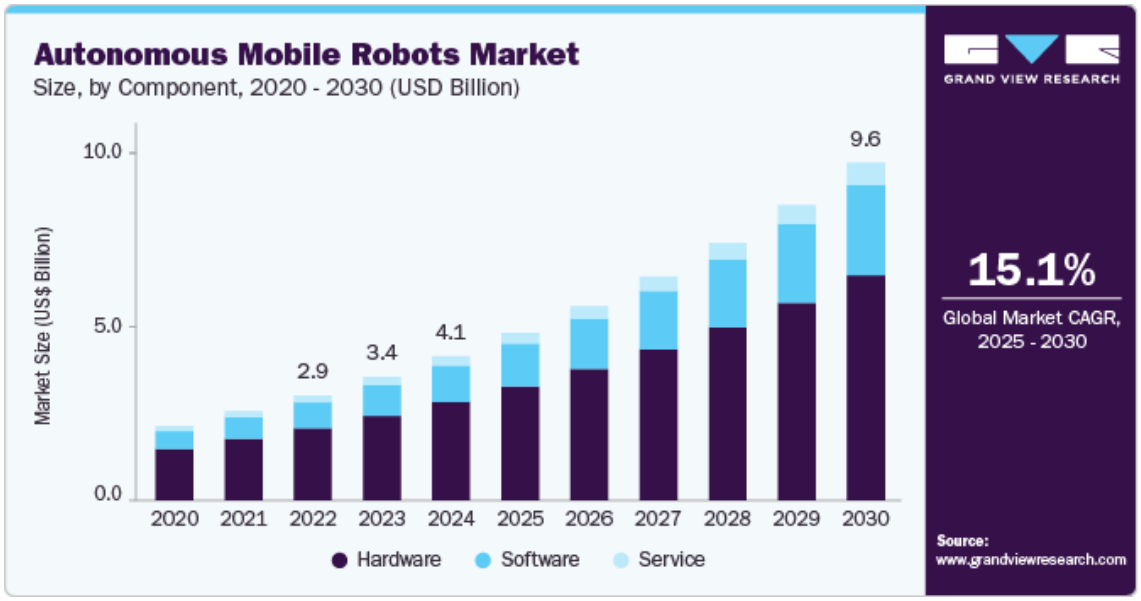

Рисунок 1.1 – Тенденція ринку автономних мобільних роботів [1]

Зростаюче впровадження роботів-ромбів (AMR) у таких галузях, як логістика, виробництво, охорона здоров’я та роздрібна торгівля, завдяки їхній ефективності в автоматизації завдань, таких як управління запасами, обробка матеріалів та транспортування товарів, стимулює зростання ринку. Зростання

активності електронної комерції та потреба в ефективному виконанні замовлень зробили AMR важливими для складських операцій та операцій ланцюга поставок. Передові технології, такі як штучний інтелект, машинне навчання та комп'ютерний зір, розширили можливості AMR, дозволяючи їм автономно орієнтуватися в складних середовищах. Ці роботи все частіше використовуються для оптимізації операцій, скорочення часу простою та підвищення загальної продуктивності.

Крім того, інтеграція систем автоматизованого моделювання (AMR) з практиками Індустрії 4.0 та мережами Інтернету речей (IoT) дозволила обмін даними та аналітику в режимі реального часу, що підвищило операційну ефективність. Зростаюча увага до оптимізації витрат та мінімізації людських помилок ще більше посилила впровадження AMR у всіх галузях промисловості. Зростаюча залежність від робототехніки для управління запасами, транспортування матеріалів та обробки товарів також прискорила попит. Зростаюча увага до оптимізації витрат та мінімізації людських помилок ще більше посилила впровадження AMR у всіх галузях промисловості [1].

Північноамериканська індустрія автономних мобільних роботів домінувала у світі з часткою доходу понад 22 % у 2024 році. Потужна технологічна екосистема регіону, включаючи основних виробників автономних мобільних роботів (AMR) та інноваційні центри, підтримує швидкий розвиток та впровадження автономних систем. Висока активність електронної комерції стимулює попит на автоматизацію складів, де AMR виконують сортування, транспортування та управління запасами. Крім того, урядові ініціативи та інвестиції в дослідження штучного інтелекту та робототехніки підвищують конкурентну перевагу регіону.

Очікується, що у 2024 році індустрія автономних мобільних роботів у США зростатиме. Це можна пояснити постійним розвитком технологій штучного інтелекту, машинного навчання та сенсорних систем, які покращують можливості автономних мобільних роботів (AMR). Великі центри виконання замовлень та логістичні центри країни дедалі більше покладаються на AMR для

задоволення зростаючого попиту електронної комерції та омніканальної роздрібної торгівлі. Інвестиції як з приватного, так і з державного секторів прискорюють інновації та впровадження AMR у виробництві, охороні здоров'я та інших галузях промисловості. Зусилля щодо вирішення проблеми нестачі робочої сили та підвищення безпеки експлуатації сприяють розширенню ринку.

Очікується, що індустрія автономних мобільних роботів у європейському регіоні зазнає значного зростання протягом прогнозованого періоду, зумовленого зростанням автоматизації у виробництві, логістиці та охороні здоров'я, де витрати на робочу силу та нормативні вимоги стимулюють роботизовані рішення. Такі країни, як Німеччина та Франція, значно інвестують в ініціативи «Промисловість 4.0», сприяючи цифровій трансформації та впровадженню робототехніки (рис. 1.2). Проблеми міської логістики в густонаселених районах збільшують попит на компактні та ефективні автономні мобільні робототехніки (AMR) для оптимізації простору та робочого процесу. Державна підтримка через субсидії та програми фінансування сприяє інноваціям та впровадженню AMR [1].

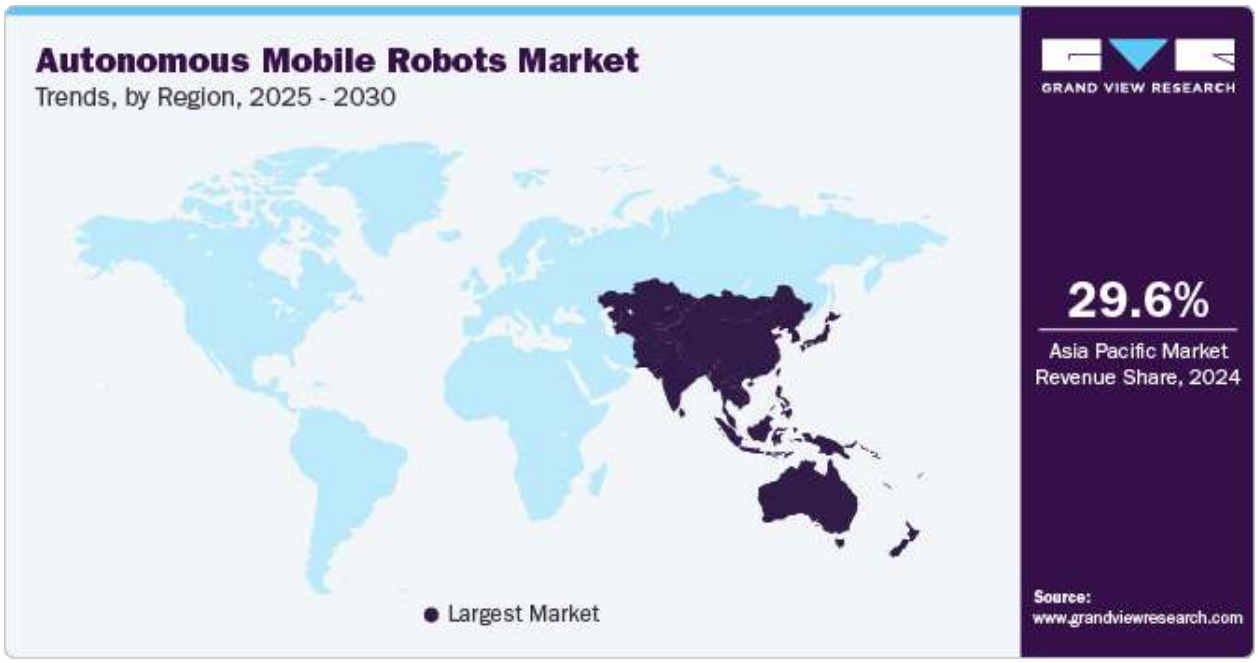

Рисунок 1.2 – Регіональні аналітичні дані [1]

Очікується, що індустрія автономних мобільних роботів в Азіатсько-Тихоокеанському регіоні зареєструє найшвидший середньорічний темп

зростання (CAGR) протягом прогнозованого періоду. Такі країни, як Китай, Індія та Японія, інвестують у розумну інфраструктуру та автоматизацію для підвищення продуктивності та зменшення залежності від робочої сили. Бурхливий ринок електронної комерції в регіоні зумовлює потребу в ефективних рішеннях для автоматизації складів. Урядова політика, що підтримує Індустрію 4.0 та цифрову трансформацію, заохочує впровадження передових технологій робототехніки. Крім того, зростання місцевих виробників робототехніки та іноземні інвестиції сприяють прискореному зростанню ринку в Азіатсько-Тихоокеанському регіоні.

Деякі ключові гравці в індустрії автономних мобільних роботів, такі як BALYO, Bastian Solutions, LLC та Daifuku Co., Ltd., активно працюють над розширенням своєї клієнтської бази та отриманням конкурентної переваги. Вони впроваджують різні стратегічні ініціативи для досягнення цієї мети, включаючи партнерства, злиття та поглинання, співпрацю та розробку нових продуктів і технологій. Такий проактивний підхід дозволяє їм розширювати свою присутність на ринку та впроваджувати інновації у відповідь на мінливі потреби в безпеці.

1.2 Огляд сучасних роботизованих платформ

Мікроконтролери відіграють ключову роль у створенні сучасних роботизованих платформ завдяки своїй компактності, енергоефективності та доступності. Їх використовують як обчислювальні ядра у невеликих автономних роботах, роботах-маніпуляторах, квадрокоптерах, роботах для освітніх цілей, а також у промислових рішеннях. Популярні платформи на основі Arduino, STM32, ESP32, Teensy, Micro:bit використовуються як в освітніх, так і в комерційних проєктах.

Arduino є однією з найпопулярніших платформ для створення роботизованих систем завдяки своїй відкритості, простоті використання та широкій спільноті розробників. Ця платформа базується на мікроконтролерах

AVR, серед яких найпоширенішим є ATmega328P, що встановлюється на плату Arduino Uno. Її ключовою особливістю є легкість інтеграції з різними сенсорами, модулями зв’язку, виконавчими пристроями та драйверами, що робить її доступною як для новачків, так і для досвідчених інженерів.

Роботизовані платформи на базі Arduino найчастіше будуються на простих двоколісних (2WD) (рис. 1.3) або чотириколісних (4WD) шасі, оснащених мотор-редукторами, якими керує мікроконтролер через драйвер двигунів (наприклад, L298N або L293D). Такі системи можуть оснащуватися ультразвуковими сенсорами HC-SR04 для вимірювання відстані до перешкод, що дозволяє реалізувати алгоритми автономного пересування та уникнення зіткнень. Важливою особливістю цих роботів є використання простих логічних схем і PID-регуляторів для точного керування траєкторією руху.

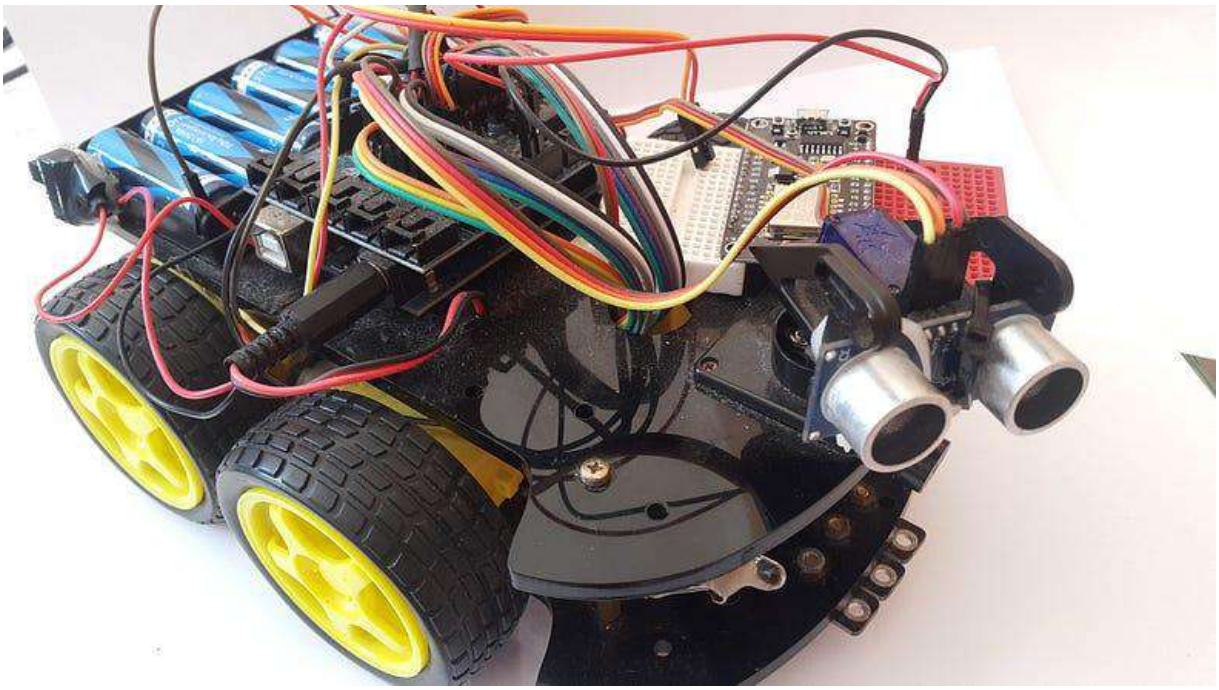

Рисунок 1.3 – Роботизована платформа на базі Arduino [2]

Крім колісних платформ, Arduino широко застосовується для створення роботизованих маніпуляторів (рис. 1.4), які виконують завдання захоплення, транспортування чи складання об’єктів. Такі маніпулятори часто використовують сервоприводи (наприклад, SG90 або MG996R), а алгоритми

керування можуть базуватися як на простих командних послідовностях, так і на складних сценаріях зворотного зв’язку, наприклад, з використанням сенсорів тиску або позиціонування.

Рисунок 1.4 – Arduino Robotic Arm [3]

Окремий клас Arduino-роботів – це роботи зі слідування за лінією (рис. 1.5), які працюють на основі інфрачервоних датчиків.

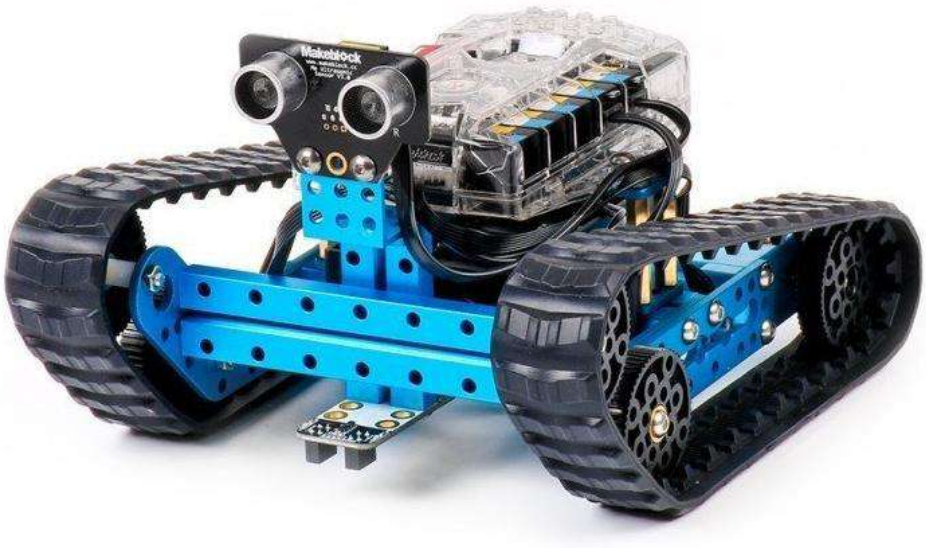

Рисунок 1.5 – Лінійний Arduino-робот [4]

Вони орієнтуються на контраст між поверхнями (наприклад, чорна лінія на білому фоні) та регулюють швидкість окремих моторів для підтримання правильного напрямку. Такі роботи є класичним прикладом впровадження алгоритмів управління у навчальних лабораторіях з робототехніки.

Інша поширена категорія – роботи з дистанційним керуванням через Bluetooth або Wi-Fi (рис. 1.6). У таких системах Arduino інтегрується з модулями HC-05 (Bluetooth) або ESP8266 (Wi-Fi), що дає змогу створювати інтерфейси для керування з мобільних додатків або веб-браузера. Це відкриває можливості для побудови складніших проектів, таких як прототипи розумних автомобілів чи телеметричних систем.

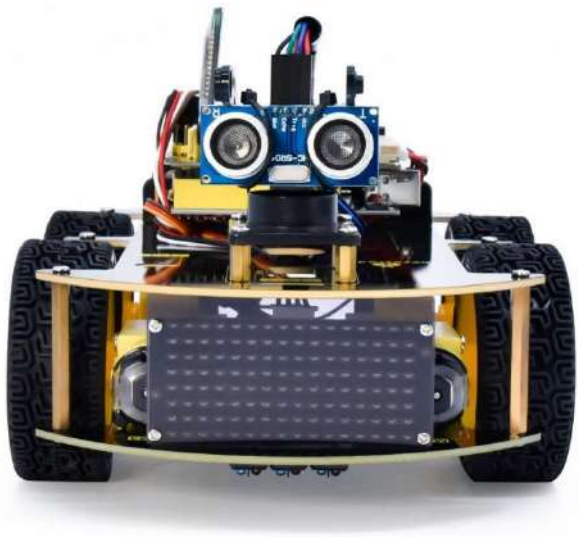

Рисунок 1.6 – Розумна повнопривідна робо-платформа V2.0 з Bluetooth [5]

Наукові дослідження демонструють, що Arduino-based роботи мають велике значення для розвитку освітніх та дослідницьких проектів. Наприклад, у роботі [6] описується низьковартісний автономний робот на Arduino з ультразвуковими сенсорами для уникнення перешкод, а дослідження [7] аналізує розробку та контроль Arduino-робота, що слідує за лінією. Крім того, представили досвід створення Arduino-маніпулятора для освітнього середовища, зосереджуючись на практичних аспектах програмування та механіки.

STM32-based роботи є прикладом високопродуктивних роботизованих систем, які базуються на мікроконтролерах серії STM32 з ядрами ARM Cortex-M. Ці мікроконтролери вирізняються значно більшими обчислювальними можливостями, ніж популярні AVR-мікроконтролери (наприклад, Arduino), а також ширшими можливостями енергозбереження, мультизадачності й роботи з апаратними інтерфейсами. Саме тому STM32 часто обирають для задач, де потрібна обробка великого обсягу даних у реальному часі, наприклад, у балансуючих роботах, безпілотних літальних апаратах, розпізнаванні жестів або обробці даних із камер.

Однією з найпоширеніших категорій STM32-роботів є роботи з функцією стабілізації – наприклад, двоколісні балансуючі платформи (self-balancing robots) (рис. 1.7), які використовують датчики IMU (інерційні вимірювальні одиниці), зокрема акселерометри та гіроскопи, для утримання рівноваги. Завдяки високій швидкості обчислень STM32 такі роботи можуть ефективно обробляти дані сенсорів і реалізовувати складні алгоритми стабілізації, зокрема PID-регулювання, з частотою оновлення, що забезпечує плавний і точний рух.

Рисунок 1.7 – Розумна повнопривідна робо-платформа V2.0 з Bluetooth [6]

Інша важлива категорія – квадрокоптери та дрони, побудовані на STM32, де необхідна робота з численними сенсорами (барометри, магнітометри, GPS), моторними драйверами та модулями зв’язку. У таких застосуваннях STM32 дозволяє ефективно реалізовувати алгоритми польотного контролю, враховуючи змінні умови середовища й реакції платформи, що критично для безпеки польоту (рис. 1.8).

Рисунок 1.8 – Квадрокоптер STM Kargu [9]

STM32 також активно використовуються у мобільних платформах із автономною навігацією, де потрібна обробка даних від кількох ультразвукових, інфрачервоних сенсорів чи камер, а також реалізація алгоритмів планування маршруту й уникнення перешкод. У таких платформах важливою є підтримка апаратних протоколів SPI, I2C, CAN, що дозволяють ефективно обмінюватися даними між мікроконтролером та підключеними модулями.

Особливу роль STM32-платформи відіграють у дослідженнях і розробках, де необхідна робота з low-level програмуванням, використанням операційних систем реального часу (RTOS), а також написанням оптимізованого коду для забезпечення високої продуктивності.

STM32-based роботи демонструють потужність сучасних вбудованих систем, які поєднують апаратну ефективність із гнучкими можливостями програмування. Такі системи є важливим кроком між простими роботами на

Arduino й високорівневими рішеннями на базі Raspberry Pi, оскільки дозволяють досягати промислової надійності та точності без значного збільшення енергоспоживання. У результаті STM32 є платформою вибору для багатьох інженерних і наукових проєктів, де потрібне поєднання компактності, швидкодії та масштабованості.

ESP32-based платформи належать до новітнього покоління роботизованих систем, що поєднують високу обчислювальну потужність, вбудовані модулі бездротового зв'язку та низьке енергоспоживання. Мікроконтролер ESP32, розроблений компанією Espressif Systems, має подвійне ядро Xtensa LX6, працює на частоті до 240 МГц і підтримує Wi-Fi та Bluetooth, що робить його особливо привабливим для проєктів у сфері Інтернету речей (IoT), мобільної робототехніки та бездротових сенсорних мереж.

Завдяки інтеграції бездротових технологій ESP32 широко використовується у розробці дистанційно керованих роботів, роботів зі стрімінгом відео, мобільних роботизованих платформ, що передають телеметрію, а також розумних автоплатформ, які можна контролювати через Wi-Fi або Bluetooth із мобільних додатків. Такі роботи, як правило, комплектуються камерами (наприклад, ESP32-CAM), модулями управління моторами (L298N, DRV8833), ультразвуковими сенсорами HC-SR04, а також датчиками для збору навколишніх даних (температура, вологість, освітленість) (рис. 1.9).

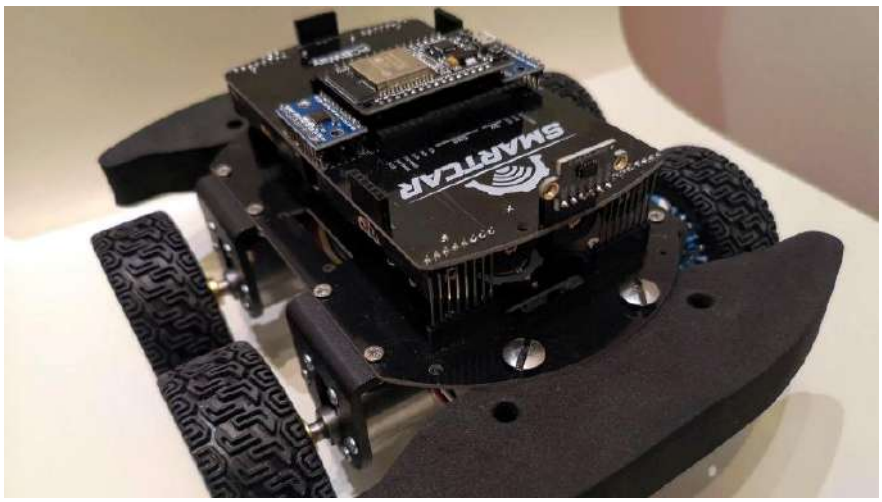

Рисунок 1.9 – ESP32 Rover [10]

Одним із яскравих прикладів є роботи на базі ESP32-CAM, де відео з камери передається на мобільний телефон або браузер у режимі реального часу. Це дозволяє створювати телеметричні системи для моніторингу віддалених зон або прості системи відеоспостереження. Завдяки низькому енергоспоживанню такі системи можуть живитися від акумуляторів, що робить їх придатними для використання в мобільних або переносних рішеннях.

ESP32 також активно застосовується в освітніх роботах, де потрібна інтеграція Bluetooth-керування. Наприклад, роботи можуть з’єднуватися з мобільним додатком і виконувати команди користувача, а також одночасно передавати дані назад (наприклад, рівень заряду батареї чи статус сенсорів). Це відкриває широкі можливості для вивчення принципів зворотного зв’язку, віддаленого управління та побудови IoT-екосистем (рис. 1.10).

Рисунок 1.10 – Освітній ESP32-робот із Bluetooth-керуванням [11]

Завдяки потужності ESP32 стає можливим впровадження алгоритмів машинного навчання безпосередньо на борту робота. Наприклад, розпізнавання голосових команд або жестів може виконуватися без підключення до хмари, що значно підвищує швидкодію системи та її автономність. Розробники активно використовують фреймворки, такі як TensorFlow Lite for Microcontrollers, для впровадження нейромережевих моделей на ESP32.

Наукові дослідження демонструють перспективність ESP32 для побудови автономних роботизованих систем..

Платформи на основі ESP32 є чудовим вибором для інженерних, дослідницьких і освітніх проєктів, де потрібна гнучкість у програмуванні, розширені можливості бездротового зв’язку, висока інтеграція з IoT та відносно невисока ціна. Вони забезпечують плавний перехід від простих Arduino-систем до складніших роботизованих рішень, наближених до промислового рівня.

1.3 Аналіз існуючих алгоритмів уникнення перешкод

Однією з ключових задач автономної мобільної платформи є можливість ефективного планування маршруту та уникнення перешкод у реальному середовищі. Ці функції забезпечують системі здатність переміщатися між заданими точками, адаптуватися до змін навколо та зберігати стабільність руху. Алгоритми, що використовуються для реалізації таких функцій, поділяються на кілька основних категорій залежно від їх принципів роботи, складності та сфери застосування.

Один із найпоширеніших підходів – це алгоритм A* (рис. 1.11), який є евристичним методом пошуку найкоротшого шляху на графі. Він використовує функцію оцінки, що поєднує фактичну вартість маршруту від початкової точки до поточної (g) та оцінку відстані до цілі (h). Завдяки використанню евристики, A* забезпечує високу ефективність пошуку оптимального шляху.

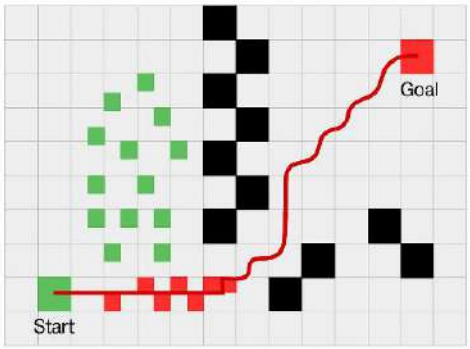

Рисунок 1.11 – Принцип роботи алгоритму A* [12]

Недоліком є те, що його ефективність сильно залежить від якості евристичної функції, а також обчислювальні витрати можуть різко зростати при збільшенні розміру карти [12].

Ще одним важливим підходом є алгоритм Dijkstra (рис. 1.12), який знаходить найкоротший шлях у графі без використання евристики. Цей метод гарантує знаходження глобально оптимального маршруту, проте для великих графів він менш ефективний, ніж A*, оскільки розглядає всі можливі варіанти без пріоритету. Dijkstra часто застосовується у ситуаціях, коли потрібно точно враховувати змінні вартості пересування, наприклад, для енергетично оптимальних маршрутів [13].

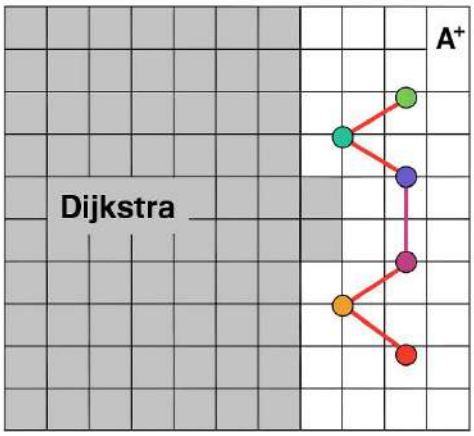

Рисунок 1.12 – Принцип роботи алгоритму Dijkstra [13]

Для сценаріїв з непередбачуваними змінами середовища особливо важливими є реактивні алгоритми, зокрема алгоритм Bug. Він працює за принципом: рухатися в напрямку цілі, доки не буде виявлено перешкоду, після чого робот обходить перешкоду по її контуру, доки знову не виявиться пряма траєкторія до цілі. Такий підхід простий у реалізації, проте має обмеження – наприклад, може застрягати у локальних мінімумах.

У складних динамічних середовищах використовуються алгоритми на основі швидкісних векторних полів (VFH, Vector Field Histogram). Вони дозволяють враховувати рухомі перешкоди та адаптувати траєкторію у реальному часі, формуючи швидкісні вектори руху. Такі методи вимагають

високих обчислювальних ресурсів та часто використовуються у поєднанні з потужними мікроконтролерами або бортовими комп'ютерами (наприклад, Raspberry Pi).

Особливе місце займають алгоритми, що використовують SLAM (Simultaneous Localization and Mapping) (рис. 1.13). Вони дозволяють будувати карту середовища під час навігації та одночасно локалізувати платформу на цій карті. SLAM інтегрує дані від різних сенсорів (LiDAR, ультразвукові датчики, камери) та застосовує фільтри Калмана або часткові фільтри для прогнозування положення та корекції за спостереженнями. Цей підхід є надзвичайно потужним, проте потребує високої обчислювальної потужності, тому найчастіше використовується в комбінації з обчислювальними платформами типу NVIDIA Jetson або Raspberry Pi [14].

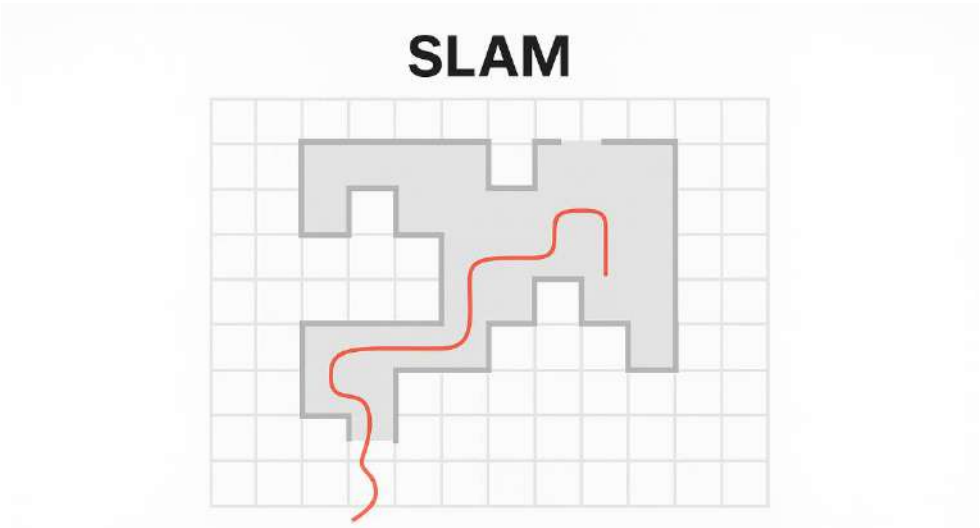

Рисунок 1.13 – Побудова карти за допомогою SLAM [14]

Наукові публікації останніх років демонструють активний розвиток гібридних методів, що поєднують планування маршруту та навчання на основі штучних нейронних мереж.

Таким чином, аналіз сучасних алгоритмів планування маршруту та уникнення перешкод демонструє, що вибір конкретного підходу має ґрунтуватися на врахуванні особливостей середовища, обчислювальних можливостей платформи та поставлених завдань. У багатьох випадках

оптимальним є використання комбінованих або адаптивних підходів, що дозволяють максимально використовувати переваги кожного окремого алгоритму.

РОЗДІЛ 2
РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ МОБІЛЬНОЇ ПЛАТФОРМИ

2.1 Вибір та обґрунтування апаратних компонентів

Проєктування роботизованої мобільної платформи з автономною навігацією потребує ретельного підбору апаратних компонентів, які забезпечать високу функціональність, адаптивність до середовища, легкість інтеграції й подальшого розширення системи. В кваліфікаційній роботі як центральний керуючий модуль використовується Raspberry Pi 3 Model B.

Raspberry Pi 3 Model B (рис. 2.1) є однією з найбільш популярних версій одноплатного мікрокомп'ютера, випущеної Raspberry Pi Foundation у лютому 2016 року. Це компактний комп'ютер розміром з банківську картку, який поєднує в собі високу обчислювальну потужність, багатий набір периферійних інтерфейсів та доступну ціну, що робить його ідеальним для використання в освітніх проєктах, прототипуванні та вбудованих системах, зокрема робототехніці.

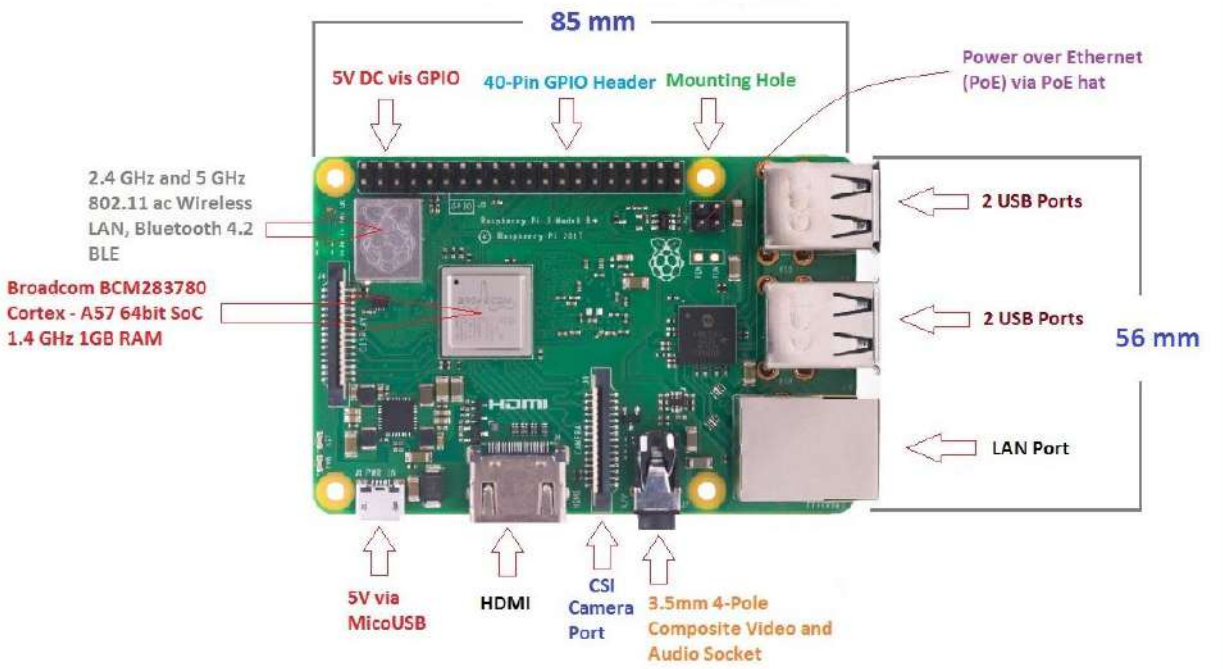

Рисунок 2.1 – Мікрокомп'ютер Raspberry Pi 3 Model B [15]

Raspberry Pi 3 Model B має такі основні інтерфейси:

– 4 порти USB 2.0 для підключення периферії (камер, клавіатури, миші, флеш-накопичувачів);

– Ethernet-порт (10/100 Мбіт/с) для дротового з’єднання з мережею;

– HDMI-вихід для підключення до монітора або телевізора;

– комбінований аудіовихід/відеовихід;

– роз’єм CSI для камери та DSI для дисплея;

– 40-піновий GPIO-роз’єм для підключення сенсорів, моторів, світлодіодів, реле тощо.

На рисунку 2.2 показано розпіновку Raspberry Pi 3 B+.

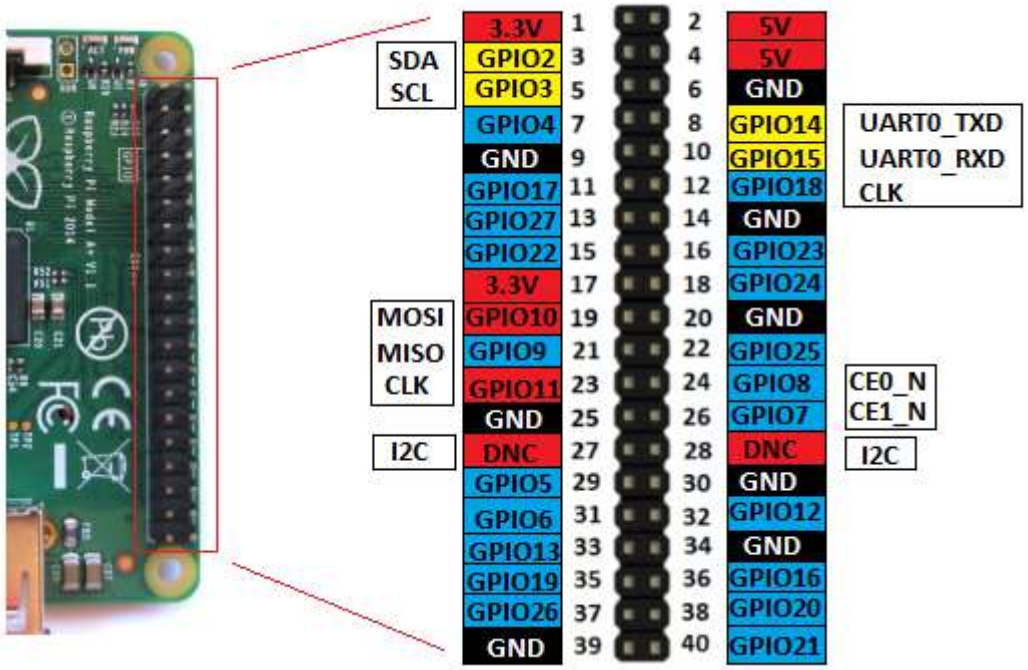

Рисунок 2.2 – Розпіновка Raspberry Pi 3 B+ [16]

Raspberry Pi 3 Model B оснащено чотириядерним процесором ARM Cortex-A53 із частотою 1,2 ГГц (Broadcom BCM2837 SoC), що забезпечує близько 10 разів вищу продуктивність порівняно з першими версіями Raspberry Pi. Оперативна пам’ять об’ємом 1 ГБ LPDDR2 дозволяє виконувати багатозадачні

процеси, включаючи обробку даних сенсорів, управління виконавчими пристроями та обчислення простих алгоритмів комп’ютерного зору.

Важливою особливістю цієї моделі є вбудовані модулі Wi-Fi (802.11 b/g/n) та Bluetooth 4.1, що значно розширює можливості бездротової комунікації. Це дає змогу реалізовувати дистанційне керування платформою, обмін даними з іншими пристроями або інтеграцію в локальні мережі та хмарні сервіси без потреби у зовнішніх адаптерах.

Живлення Raspberry Pi 3 Model B здійснюється через micro-USB-роз’єм, при рекомендованому блоці живлення 5 В/2,5 А. Споживана потужність залежить від навантаження, але зазвичай складає 2-5 Вт, що робить платформу енергоефективною.

Серед особливостей програмного забезпечення слід зазначити підтримку операційних систем Raspberry Pi OS (раніше Raspbian), Ubuntu Mate, Windows 10 IoT Core, а також можливість встановлення спеціалізованого ПЗ для робототехніки, наприклад, ROS (Robot Operating System), OpenCV, TensorFlow Lite.

Raspberry Pi 3 Model B добре підходить для побудови автономних роботизованих систем середньої складності, наприклад:

– мобільних роботів з ультразвуковою навігацією;

– роботів із комп’ютерним зором (наприклад, для розпізнавання міток);

– IoT-платформ із передачею даних у хмару;

– систем моніторингу на базі камери.

Завдяки поєднанню продуктивності, компактності, енергоефективності та розширюваності, Raspberry Pi 3 Model B став еталонною платформою для багатьох проєктів, що потребують гнучкості та універсальності.

Важливою частиною апаратної системи є ультразвуковий сенсор HC-SR04, розташовані на передній панелі робота. Він виконує функцію «очей» – виявляє перешкоди на шляху та дозволяє будувати карту навколишнього простору, що є критичним для реалізації алгоритмів уникнення зіткнень.

Ультразвуковий сенсор HC-SR04 (рис. 2.3) є популярним модулем для вимірювання відстані, який широко використовується у робототехніці, системах автоматизації та освітніх проєктах завдяки своїй доступності, простоті інтеграції та надійності. Він дозволяє визначати відстань до об’єктів шляхом аналізу часу затримки між випромінюванням та отриманням ультразвукового імпульсу.

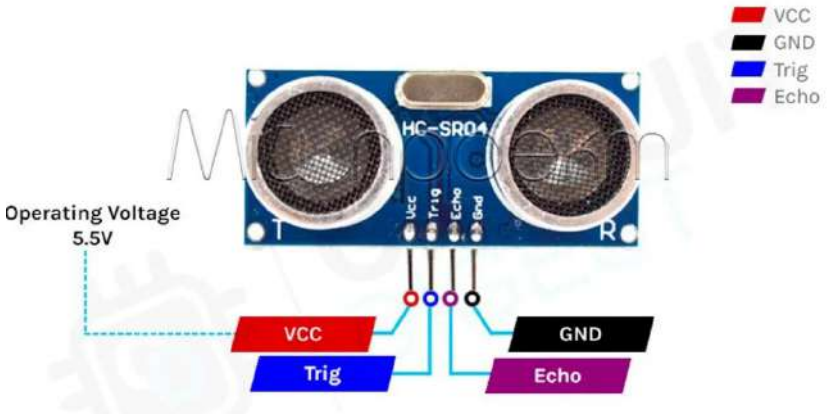

Рисунок 2.3 – Ультразвуковий сенсор HC-SR04 [17]

Технічні характеристики [17]:

– робоча напруга: 3-5В DC;

– струм споживання: 15mA (під час роботи), 2mA (режим очікування);

– частота ультразвукових імпульсів: 40 кГц;

– діапазон вимірювання: 2 см – 400 см;

– роздільна здатність: 0.3 см;

– кут вимірювання: 15° (ефективний кут спрямованості);

– вихідний сигнал: PWM (тривалість імпульсу пропорційна відстані);

– точність вимірювання: ±0.5 – 1 см;

– тригерний вхідний сигнал: 10μs високий рівень (TTL);

– габаритні розміри: 45 × 20 × 15 мм;

– вага: 10 г;

– робоча температура: -40°C до +85°C;

– контакти: VCC (живлення +5 В), Trig (тригерний вхід для запуску вимірювання) Echo (вихід з тривалістю імпульсу пропорційною до виміряної відстані) та GND (загальний провід (земля).

HC-SR04 працює на основі принципу ехолокації – подібно до того, як кажани чи дельфіни орієнтуються у просторі. Сенсор використовує ультразвукові хвилі (з частотою 40 кГц) для вимірювання часу, за який сигнал доходить до перешкоди та повертається назад.

Основні етапи роботи:

1) генерація імпульсу (контролер (наприклад, Raspberry Pi або Arduino) подає на пін TRIG короткий цифровий імпульс (тривалістю 10 мкс). Це активує передавач (трансмітер) модуля, який випромінює серію ультразвукових імпульсів);

2) поширення сигналу (ультразвукові хвилі поширюються крізь повітря зі швидкістю приблизно 343 м/с (при температурі 20 °C). Якщо на шляху сигналу є перешкода, хвилі відбиваються назад у напрямку сенсора) (рис. 2.4);

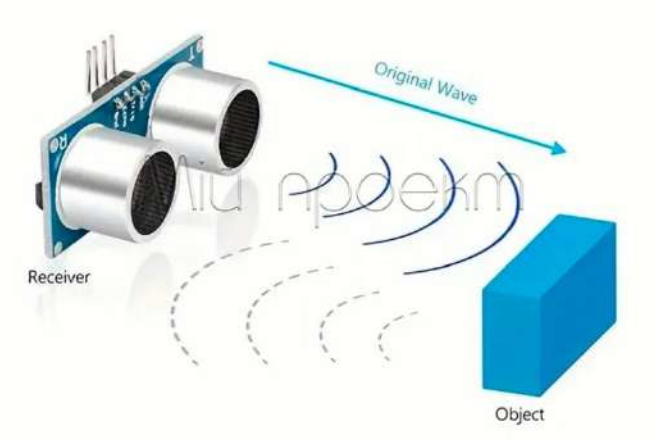

Рисунок 2.4 – Ультразвуковий сенсор HC-SR04 [17]

3) прийом відбитого сигналу: ресивер (пін ECHO) фіксує момент повернення сигналу та формує на виході високий рівень напруги. Тривалість цього високого рівня відповідає часу проходження сигналу в обидві сторони;

4) розрахунок відстані (мікроконтролер вимірює тривалість імпульсу на піні ECHO за формулою (1.1):

$$\text{Відстань (см)} = \frac{\text{Час(мкс)} \times 0{,}0343}{2} \quad (1.1)$$

Оскільки HC-SR04 працює на логічному рівні 5 В, а GPIO Raspberry Pi – на 3,3 В, важливо врахувати захист лінії ECHO. Для цього використовують резистивний дільник напруги або логічний перетворювач. Пін TRIG Raspberry Pi може безпосередньо подавати 3,3 В сигнал, бо HC-SR04 сприймає це як високий рівень. Схема підключення HC-SR04 до Raspberry Pi наступна:

– HC-SR04 VCC – Raspberry Pi 5 В (пін 2 або 4);

– HC-SR04 GND – Raspberry Pi GND (пін 6);

– HC-SR04 TRIG – Raspberry Pi GPIO 23 (пін 16);

– HC-SR04 ECHO – резистивний дільник (1 кОм + 2 кОм) → Raspberry Pi GPIO 24 (пін 18).

Для руху обрано двоногу конструкцію з серводвигунами. Такий тип механічної архітектури дозволяє роботу здійснювати крокуючі рухи, зберігаючи стійкість навіть на нерівних поверхнях. Кожна нога обладнана серводвигунами SG90, які забезпечують обертання у кількох ступенях свободи. Сервоприводи підключені до Raspberry Pi через додатковий контролер PWM-сигналів (наприклад, PCA9685), який дозволяє управляти багатьма приводами одночасно без перевантаження обчислювальних ресурсів основного мікрокомп’ютера.

Мікросервопривод SG90 – це популярний мікросервопривод, який широко використовується в робототехніці, освітніх проєктах і DIY-конструкціях завдяки своїй компактності, легкості та доступності. Виробником SG90 є компанія Tower Pro, але на ринку також доступні численні сумісні моделі.

Сервопривод SG90 (рис. 2.5) є електромеханічним пристроєм із замкнутим контуром керування, який забезпечує точне позиціонування вала у заданому діапазоні (зазвичай від 0° до 180°). На відміну від звичайного DC-мотора, сервопривод має вбудований контролер і потенціометр, які дозволяють контролювати кут повороту замість швидкості обертання.

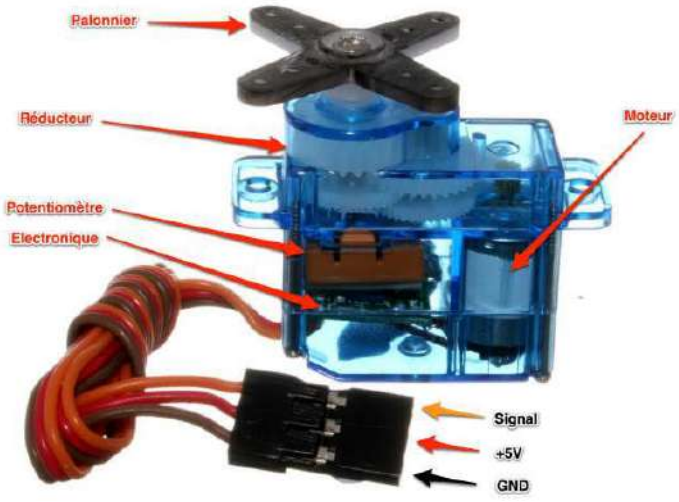

Рисунок 2.5 – Сервопривод SG90 [18]

Основою керування SG90 є широтно-імпульсна модуляція (ШІМ, PWM). Контролер (наприклад, Raspberry Pi або Arduino) подає на вхід SG90 серію імпульсів тривалістю від 1 мс до 2 мс з частотою близько 50 Гц. Тривалість імпульсу визначає, який кут повороту має зайняти вихідний вал сервопривода (рис. 2.6):

– імпульс 1 мс – положення ~0°;

– імпульс 1,5 мс – положення ~90° (середина);

– імпульс 4 мс – положення ~180°.

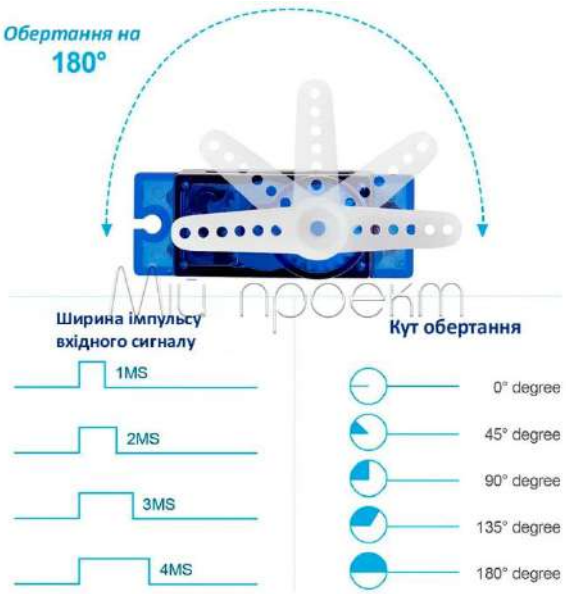

Рисунок 2.6 – Сервопривод SG90 [19]

Після отримання сигналу внутрішній контролер сервопривода порівнює поточне положення вала (визначене через потенціометр) з командним значенням та активує DC-мотор для досягнення потрібного кута. Коли вал досягає заданого положення, двигун автоматично вимикається, щоб утримати позицію. Цей механізм дозволяє сервоприводу залишатися у заданому стані без необхідності подавати постійний сигнал або силу.

В таблиці 2.1 представлено основні технічні характеристики SG90.

Таблиця 2.2 – Основні технічні характеристики SG90

Параметр	Значення
Робоча напруга	4,8-6 В
Максимальний кут повороту	180°
Швидкість обертання	0,1 с/60° (при 4,8 В)
Крутний момент	1,8 кг·см (при 4,8 В)
Тип сигналу	ШІМ (широтно-імпульсна модуляція)
Вага	~9 г
Габарити	23 × 12,2 × 29 мм

Живлення Raspberry Pi 3 Model B здійснюється через micro-USB-роз’єм або двох акумуляторів, при рекомендованому блоці живлення 5 В/2,5 А. Споживана потужність залежить від навантаження, але зазвичай складає 2–5 Вт, що робить платформу енергоефективною.

Серед особливостей програмного забезпечення слід зазначити підтримку операційних систем Raspberry Pi OS (раніше Raspbian), Ubuntu Mate, Windows 10 IoT Core, а також можливість встановлення спеціалізованого ПЗ для робототехніки, наприклад, ROS (Robot Operating System), OpenCV, TensorFlow Lite.

Raspberry Pi 3 Model B добре підходить для побудови автономних роботизованих систем середньої складності, наприклад:

– мобільних роботів з ультразвуковою навігацією;

– роботів із комп’ютерним зором (наприклад, для розпізнавання міток);

– IoT-платформ із передачею даних у хмару;

– систем моніторингу на базі камери.

Завдяки поєднанню продуктивності, компактності, енергоефективності та розширюваності, Raspberry Pi 3 Model B став еталонною платформою для багатьох проєктів, що потребують гнучкості й універсальності.

Robot HAT (Hardware Attached on Top) (рис. 2.7) – це спеціалізована плата-розширення для Raspberry Pi, призначена для побудови роботизованих систем. Вона встановлюється поверх Raspberry Pi через стандартний 40-піновий GPIO-роз'єм і забезпечує додаткові апаратні можливості для керування моторами, серводвигунами, сенсорами, а також для підключення зовнішніх модулів.

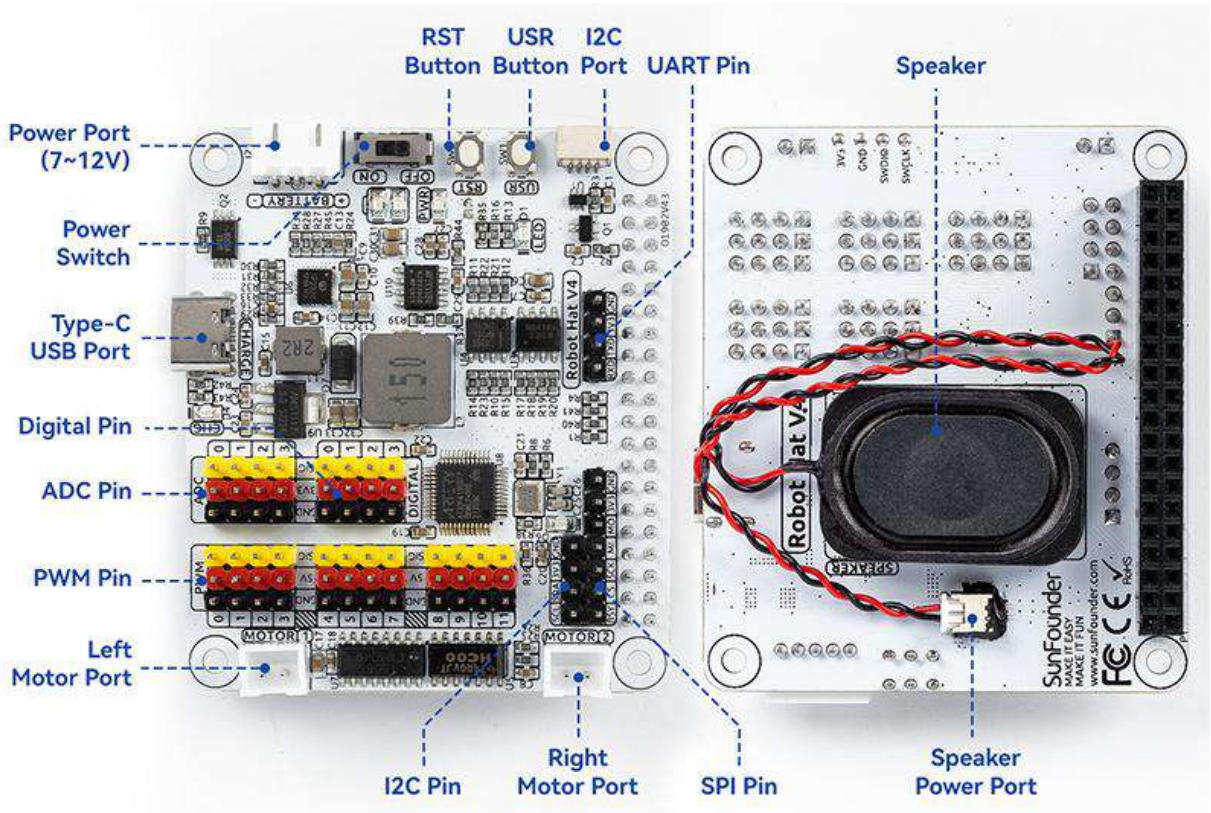

Рисунок 2.7 – Плата розширення Robot HAT [20]

Основне призначення Robot HAT полягає в тому, щоб спростити підключення різних виконавчих пристроїв (DC-моторів, сервоприводів, крокових двигунів) до Raspberry Pi без потреби в окремих драйверах або складних схемах. Плата зазвичай містить вбудовані драйвери моторів (наприклад, на базі чипів

L298P або DRV8833), контролери PWM-сигналів, а також логічні перетворювачі рівнів для безпечної роботи з різними модулями.

Основні характеристики Robot HAT наведені нижче [20]:

– вхід живлення: USB Type-C, 5 В/2 А;

– потужність заряджання: 5 В/2 А 10 Вт;

– вихідна потужність: 5 В/3 А;

– батарейки в комплекті: 2 x 3.7V 18650 літій-іонні батареї, інтерфейс XH2.0 3P;

– захист батареї: захист від зворотної полярності;

– захист від заряджання: захист від зниженої напруги на вході, захист від перенапруги на вході, балансування зарядки, захист від перегріву;

– вбудовані індикатори заряджання/живлення;

– вбудовані індикатори рівня заряду батареї;

– драйвер двигуна: 5 В/1,8 А x 2;

– 4-канальний 12-бітний АЦП;

– 12-канальний ШІМ;

– 4-канальні цифрові сигнали;

– вбудований інтерфейс SPI, інтерфейс UART, інтерфейс I2C;

– монодинамік: 8Ω1W.

2.2 Збірка прототипу мобільної платформи

Модель мобільної платформи включає такі основні компоненти:

1) мікроконтролер на базі Raspberry Pi;

2) плату розширення Sensor Robot HAT для підключення сенсорів та виконавчих пристроїв;

3) сервоприводи для забезпечення рухливості механічних частин;

4) два акумулятори номінальною напругою 3,7 В кожен, що забезпечують автономне живлення;

5) ультразвуковий сенсор відстані HC-SR04 для виявлення перешкод та вимірювання дистанції.

На рисунку 2.8 показано структурну схему всієї системи.

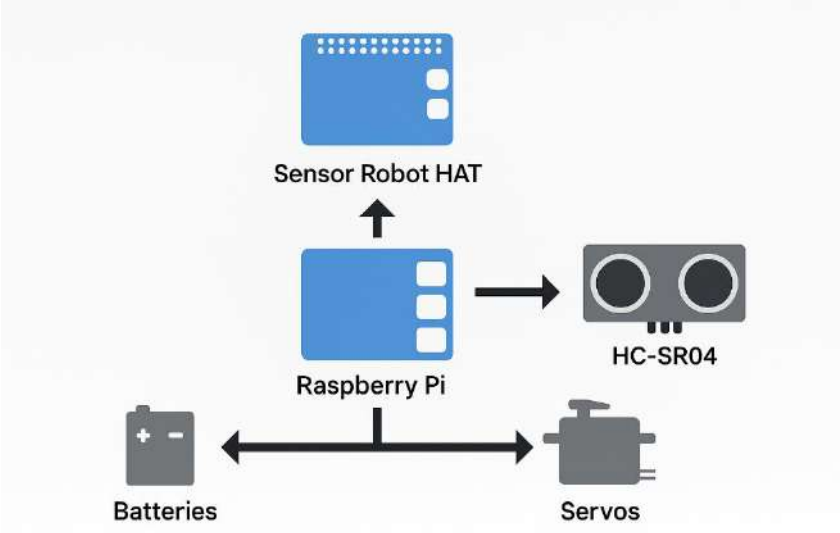

Рисунок 2.8 – Структурна схема мобільної платформи

Етапи збору мобільної платформи [21]:

1) закріпити тримач батареї (рис. 2.9);

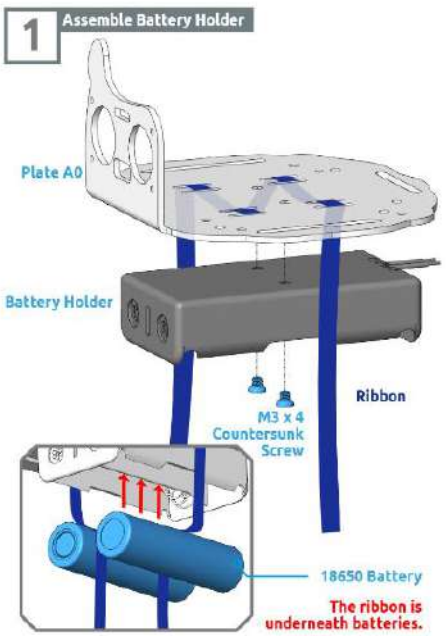

Рисунок 2.9 – Тримач батареї [21]

2) встановити ультразвуковий датчик та сервоприводи (рис. 2.10);

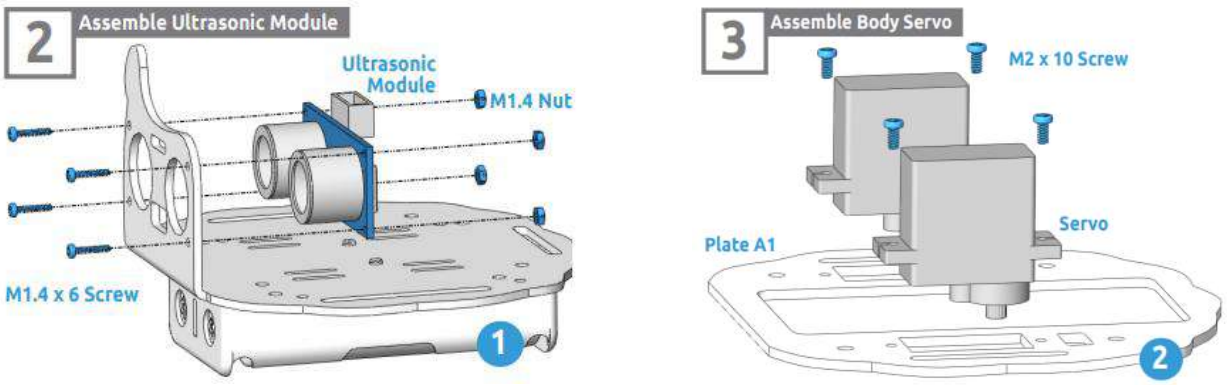

Рисунок 2.10 – Встановлення ультразвукового датчика та сервоприводів [21]

3) закріпити дві частини робота (рис. 2.10);

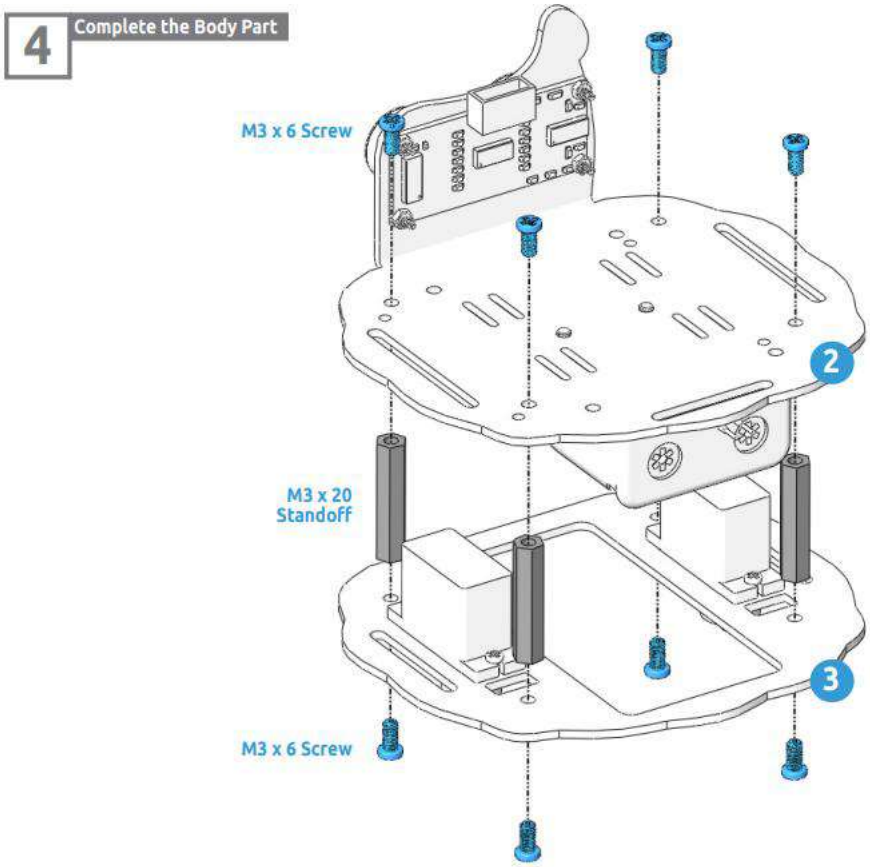

Рисунок 2.10 – З’єднання двох частин робота [21]

4) встановити Raspberry Pi та Robot HAT (рис. 2.11);

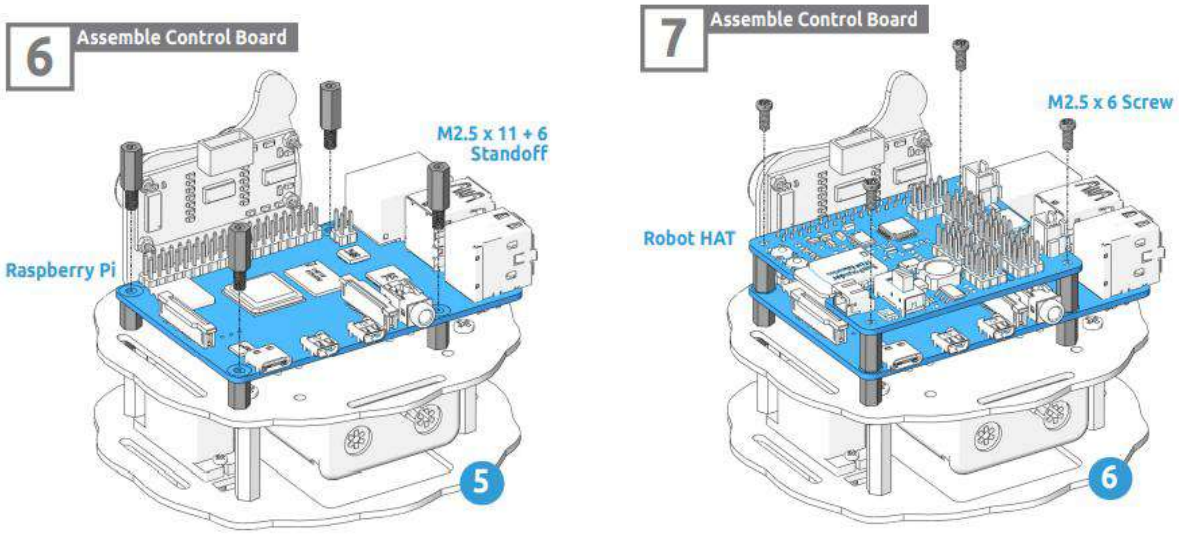

Рисунок 2.11 – Встановлення плат [21]

5) підключити ультразвуковий сенсор (рис. 2.12);

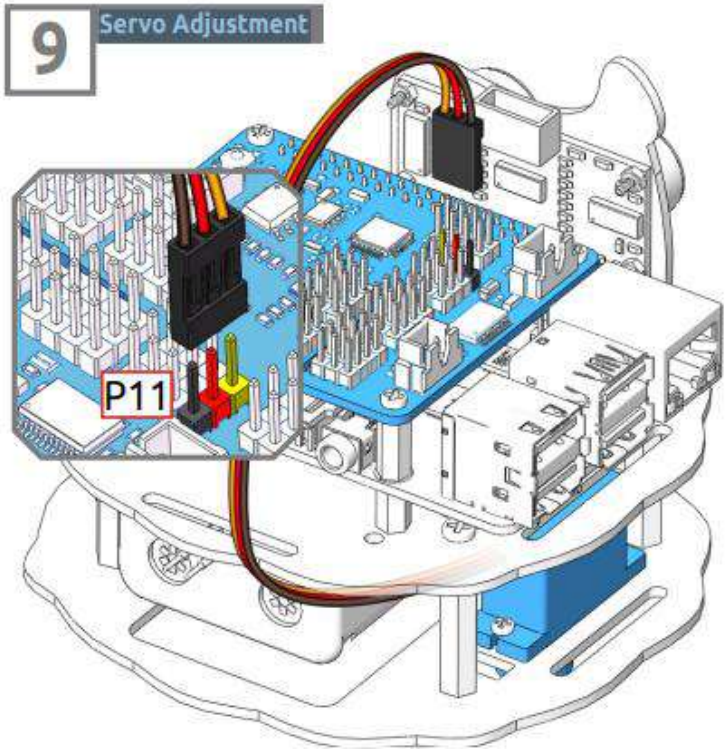

Рисунок 2.12 – Підключення датчика [21]

6) встановити та підключити сервоприводи (рис. 2.13).

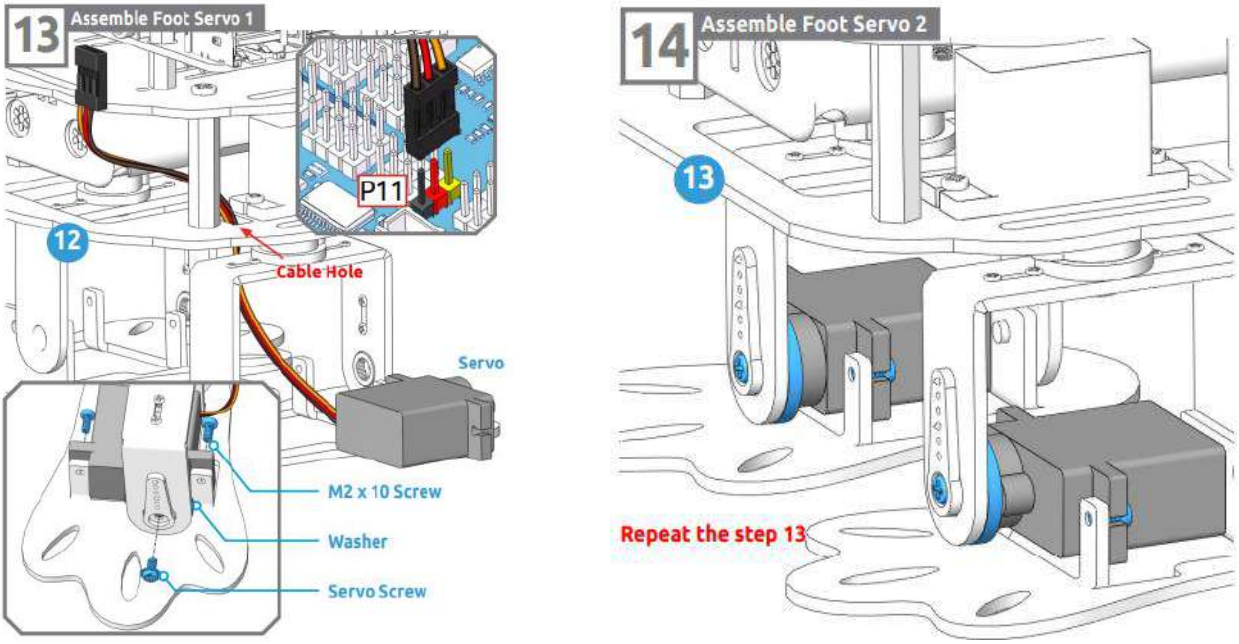

Рисунок 2.13 – Встановлення та підключення сервоприводів [21]

Розроблена мобільна платформа (рис. 2.14) забезпечує оптимальне поєднання компактності, енергоефективності та функціональної гнучкості, що дозволяє реалізовувати широкий спектр робототехнічних задач: від уникнення перешкод і побудови карт до керування сервоприводами й збору сенсорних даних.

Рисунок 2.14 – Мобільний робот

Отже, на основі проведеного аналізу було створено надійну апаратну основу для побудови автономної мобільної платформи, яка відповідає сучасним вимогам до робототехнічних систем як навчального, так і дослідницького призначення. Здійснений вибір апаратних компонентів — мікрокомп’ютера Raspberry Pi, сенсорів, сервоприводів, плати розширення, акумуляторів та інших елементів – обґрунтований як з точки зору функціональної придатності, так і з огляду на доступність, енергоефективність, гнучкість та можливість масштабування системи.

Розроблена платформа забезпечує інтеграцію сенсорної інформації та алгоритмів прийняття рішень, що дозволяє їй адаптивно взаємодіяти з навколишнім середовищем, будувати маршрути руху, уникати перешкод, а також стабільно працювати в автономному режимі.

Таким чином, підготовлена апаратна частина створює міцну базу для подальшої розробки програмного забезпечення, впровадження вдосконалених алгоритмів, а також розширення функціональності мобільної платформи, що забезпечує її відповідність сучасним трендам розвитку робототехніки.

РОЗДІЛ 3
ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Встановлення ОС на Raspberry Pi

Для забезпечення повноцінного функціонування мобільної роботизованої платформи на базі Raspberry Pi необхідно встановити операційну систему, яка забезпечить керування апаратними ресурсами, запуск програмного забезпечення, роботу з периферійними пристроями та інтеграцію з мережевими службами. Найпоширенішою ОС для Raspberry Pi є Raspberry Pi OS (раніше відома як Raspbian) – офіційний дистрибутив на базі Debian, оптимізований для апаратних особливостей Raspberry Pi.

Основні етапи встановлення ОС:

1) Raspberry Pi розробила графічний інструмент для запису на SD-карту, який працює на Mac OS, Ubuntu 18.04 та Windows, і є найпростішим варіантом для більшості користувачів, оскільки він завантажує образ та автоматично встановлює його на SD-карту. Перейти на сторінку завантаження: https://www.raspberrypi.org/software/. Натиснути на посилання для Raspberry Pi Imager, яке відповідає операційній системі, а після завершення завантаження натисніути на нього, щоб запустити інсталятор;

2) під час запуску інсталятора операційна система може спробувати заблокувати його запуск. Наприклад, у Windows я отримую таке повідомлення: Якщо це з’явиться, натиснути «Додаткова інформація» , а потім «Усе одно запустити», а потім дотримуйтесь інструкцій для встановлення Raspberry Pi Imager;

3) вставити SD-карту у слот для SD-карт комп’ютера або ноутбука;

4) на Raspberry Pi Imager натиснути ВИБРАТИ ОС – Raspberry Pi OS (інша) (рис. 3.1). Прокрутити униз до кінця щойно відкритої сторінки, і побачите Raspberry Pi OS (Legacy) та Raspberry Pi OS Lite (Legacy) (рис. 3.2);

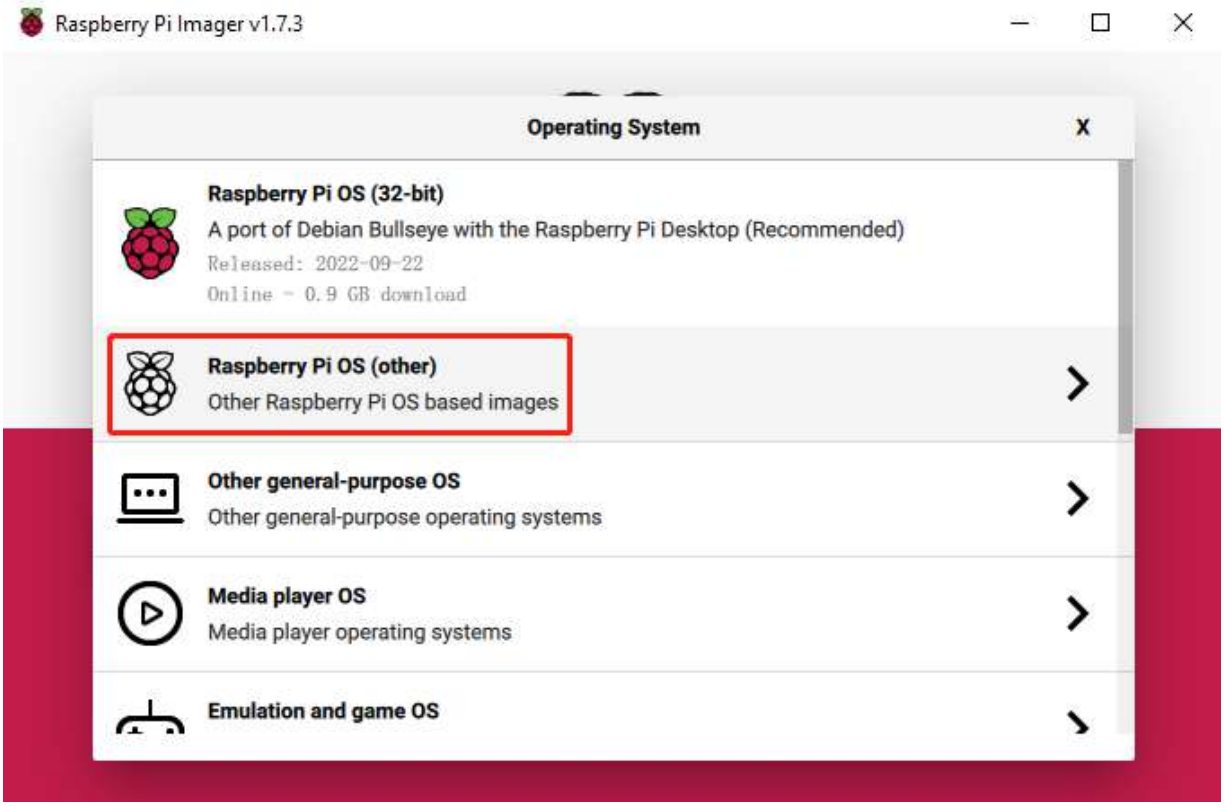

Рисунок 3.1 – Вибір ОС

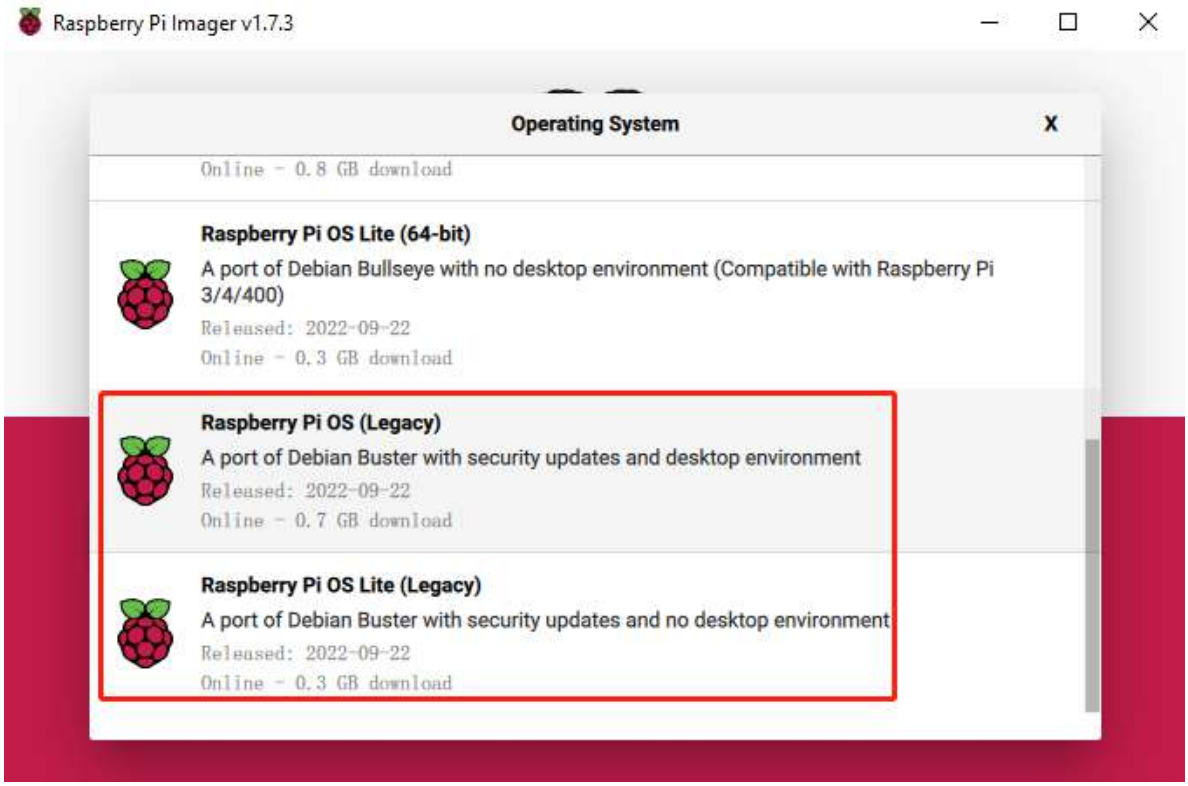

Рисунок 3.2 – Перелік доступних ОС

5) вибрати SD-карту, яку ви використовуєте (рис. 3.3);

Рисунок 3.3 – Вибір SD-карти

6) для того, щоб відкрити сторінку розширених параметрів (рис. 3.3), натисніть кнопку налаштувань (з'являється після вибору операційної системи) або натисніть Ctrl+Shift+X. Увімкніть ssh та встановіть ім'я користувача та ім'я користувача. Ви можете вибрати, щоб ці параметри налаштування завжди використовувалися;

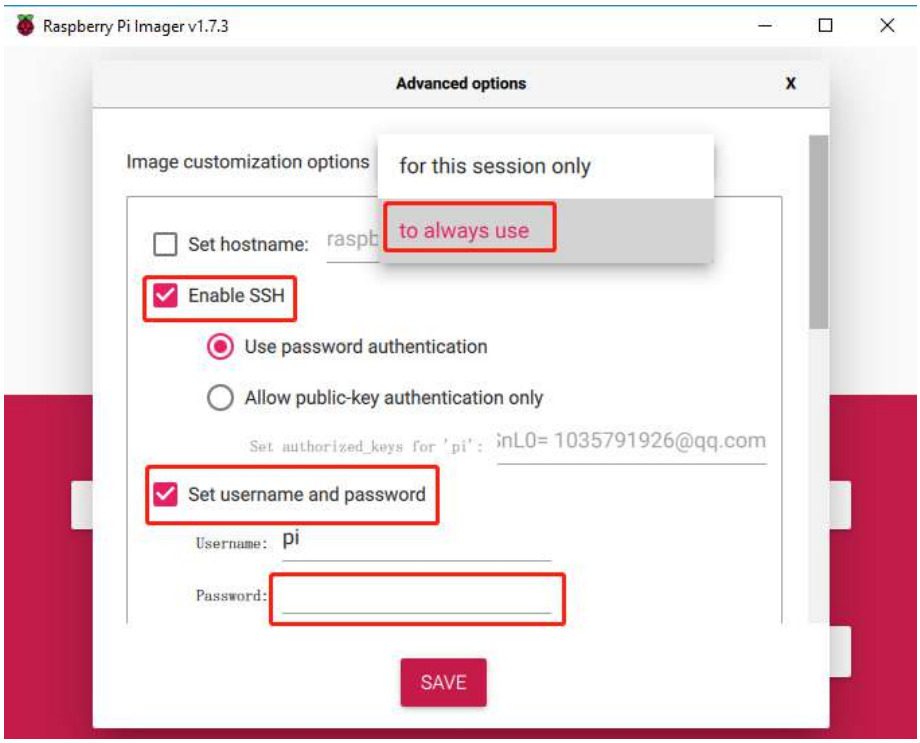

Рисунок 3.4 – Налаштування параметрів ОС

7) натисніть кнопку WRITE (рис. 3.5);

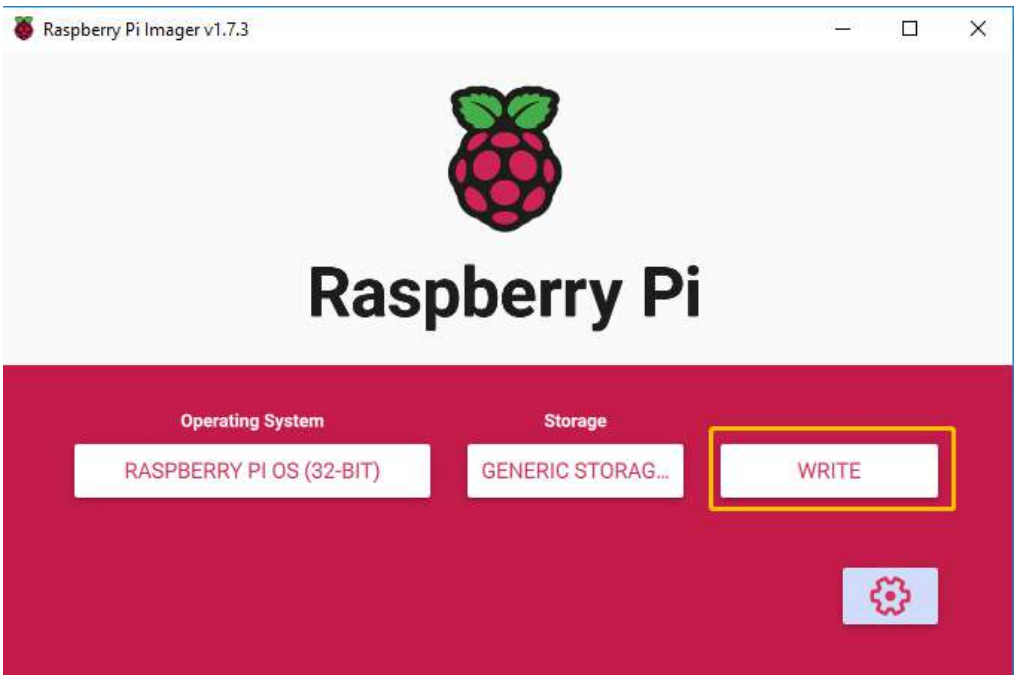

Рисунок 3.5 – Вікно запису ОС

8) після певного часу очікування з’явиться наступне вікно, яке сигналізує про завершення запису (рис. 3.6).

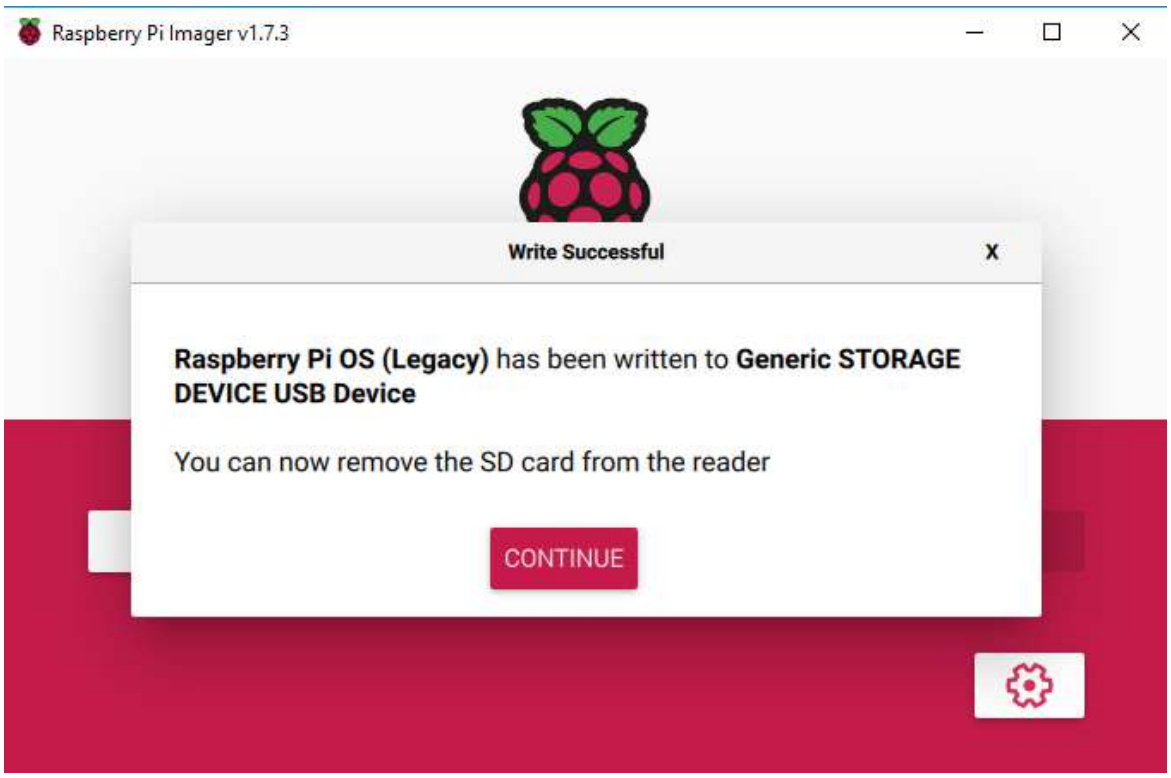

Рисунок 3.6 – Повідомлення про запис ОС

3.2 Налаштування Raspberry Pi

Після встановлення операційної системи наступним кроком є налаштування Raspberry Pi для коректної роботи в обраному середовищі. Цей етап передбачає як базову конфігурацію, так і спеціалізовані налаштування, необхідні для робототехнічних застосувань.

Якщо немає монітора, можна віддалено увійти до свого Raspberry Pi, використовуючи програмне забезпечення VNC Viewer.

Для цього необхідно виконати вхід до VNC:

1) завантажити та встановити VNC Viewer на персональний комп'ютер;

2) відкрити його після завершення встановлення. Потім ввести ім'я хоста або IP-адресу та натиснути Enter (рис. 3.7);

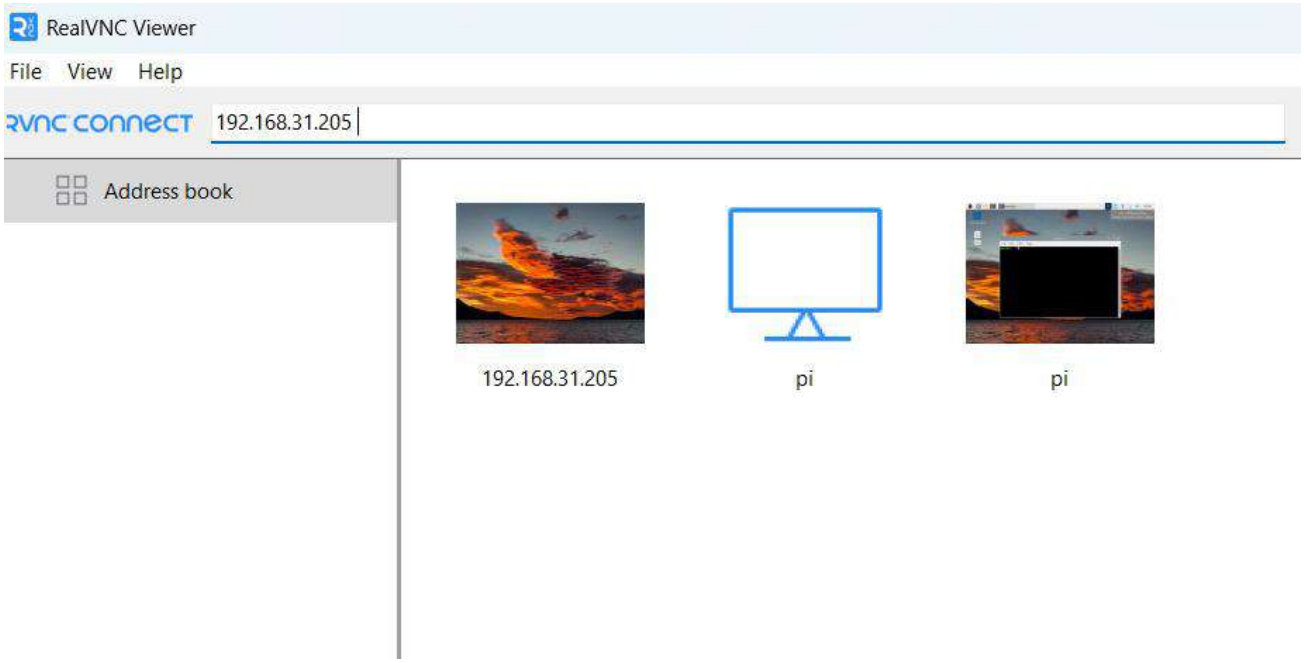

Рисунок 3.7 – Вікно VNC Viewer

3) тепер можна побачити робочий стіл Raspberry Pi (рис. 3.8).

Після першого запуску Raspberry Pi (рис. 3.8) користувачеві необхідно провести базову конфігурацію системи, щоб забезпечити її стабільну роботу та підготовити середовище для подальшої розробки.

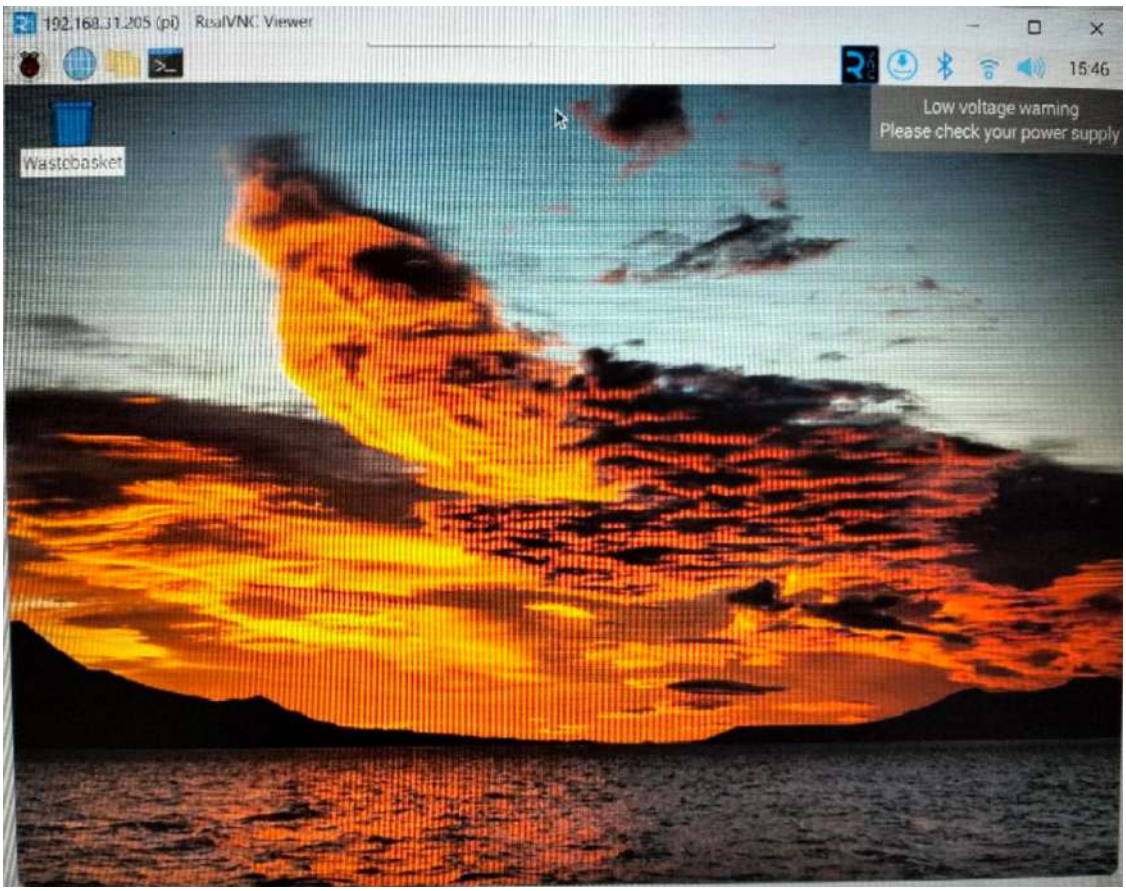

Рисунок 3.8 – Робочий стіл ОС Raspberry PI

Одним із ключових кроків є зміна пароля адміністратора, оскільки за замовчуванням у системі встановлений стандартний пароль, що створює потенційну загрозу безпеці. Налаштування мови, часової зони та розкладки клавіатури є важливим для коректного функціонування системи, правильного відображення дати й часу, а також забезпечення зручності введення команд користувачем.

Підключення Raspberry Pi до мережі здійснюється через Ethernet-кабель або Wi-Fi, що дозволяє інтегрувати пристрій у локальну інфраструктуру та забезпечити доступ до мережевих ресурсів, включаючи оновлення та віддалене адміністрування. Якщо передбачається робота в headless-режимі (тобто без підключення монітора), слід активувати SSH-доступ, який дозволяє керувати Raspberry Pi через командний рядок із віддаленого комп’ютера.

Важливим аспектом налаштування є оновлення операційної системи та встановленого програмного забезпечення. Це здійснюється через системні

менеджери пакунків, що дозволяє усунути відомі вразливості, підвищити стабільність та забезпечити сумісність з актуальними версіями бібліотек. Також при використанні великої microSD-карти може бути необхідним розширення файлової системи, щоб повністю задіяти доступний обсяг пам’яті.

Таким чином, базове налаштування Raspberry Pi формує основу для стабільної та безпечної роботи пристрою в межах робототехнічної системи, дозволяючи надалі зосередитися на спеціалізованих налаштуваннях, таких як активація апаратних інтерфейсів, розгортання бібліотек, налаштування драйверів та запуск прикладного програмного забезпечення.

3.3 Програмна реалізація алгоритму уникнення перешкод мобільною платформою

Для забезпечення автономної навігації мобільної роботизованої платформи критично важливою є реалізація ефективних алгоритмів уникнення перешкод, що дозволяють роботу самостійно визначати оптимальні траєкторії та адаптуватися до наявності статичних або динамічних перешкод.

Реалізація алгоритмів уникнення перешкод на Raspberry Pi відбувається з використанням мови програмування Python, що дає змогу поєднувати доступ до сенсорних даних (наприклад, з ультразвукового сенсора HC-SR04) із побудовою плану дій у реальному часі.

На Raspberry Pi Python попередньо встановлено в стандартному дистрибутиві Raspberry Pi OS. Для розробки програмного забезпечення можна використовувати різні середовища розробки, зокрема вбудований інтерпретатор Python, редактор Thonny IDE (рис. 3.9), а також більш потужні середовища, такі як Visual Studio Code, що забезпечує підтримку розширень, відлагодження коду та інтеграцію з Git.

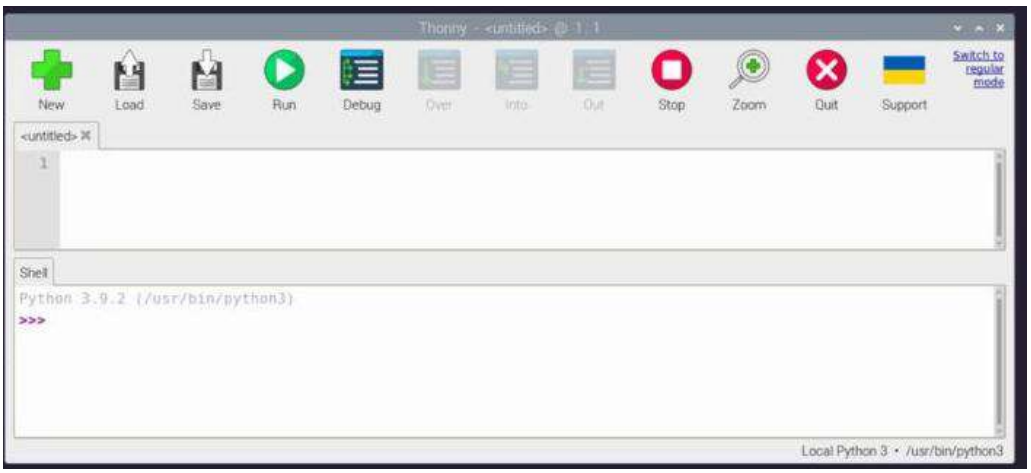

Рисунок 3.9 – Середовище Thonny IDE

Python дозволяє безпосередньо працювати з GPIO-інтерфейсами Raspberry Pi, використовуючи бібліотеки RPi.GPIO або gpiozero, що надають зручні високорівневі інтерфейси для керування пинами вводу/виводу, ШІМ-сигналами та зчитування даних із сенсорів. Крім того, для реалізації алгоритмів комп'ютерного зору активно застосовуються бібліотеки OpenCV і NumPy, а для побудови алгоритмів навігації та роботи з картою – бібліотеки networkx або scipy.

Розгортання середовища Python на Raspberry Pi передбачає встановлення необхідних залежностей через менеджер пакунків pip або систему пакетного менеджменту Debian (apt). Для автоматизації налаштування середовища рекомендується використовувати віртуальні середовища (venv), які дозволяють ізолювати залежності проєкту та спростити управління версіями бібліотек.

Алгоритм аналізує вхідні дані про відстань до перешкод, розраховує допустимі варіанти руху та генерує команди для керування сервоприводами й моторами через GPIO-інтерфейс або спеціалізовані драйвери на платі Robot HAT.

Зокрема, система у базовій конфігурації може працювати за принципом потенційних полів, де цільова точка створює притягуючий потенціал, а перешкоди – відштовхувальний. Поєднання цих сил дозволяє роботу плавно змінювати траєкторію, наближаючись до мети та уникаючи зіткнень. У складніших випадках можливе використання SLAM (Simultaneous Localization and Mapping), що дозволяє роботу одночасно будувати карту навколишнього

середовища й локалізувати власне положення на ній, коригуючи маршрут з урахуванням щойно виявлених об’єктів

У цій роботі передбачено використання ультразвукового модуля для виявлення перешкод на маршруті. Якщо система фіксує об’єкт попереду, вона генерує сигнал і переходить до пошуку іншого напрямку, щоб продовжити рух уперед.

На рисунку 3.10 показано блок-схему алгоритму уникнення перешкоди роботизованою платформою.

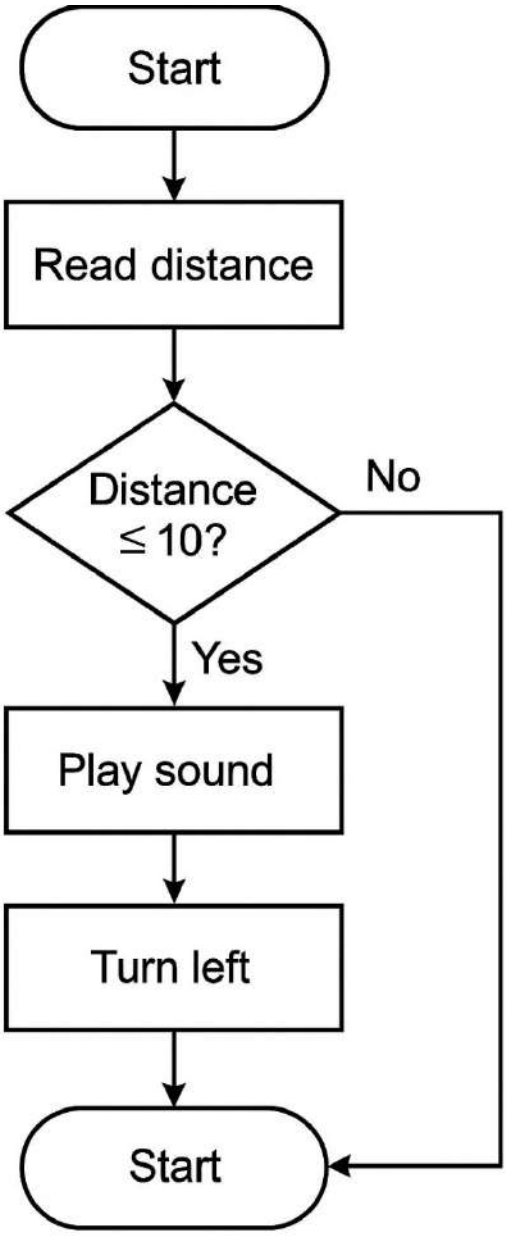

Рисунок 3.10 – Блок-схема алгоритму

На рисунку 3.11 наведено програмний код, що реалізує алгоритм уникнення від перешкоди об’єктом.

```
from pisloth import Sloth
from robot_hat import TTS, Music, Ultrasonic, Pin
import time
import os
import random
import logging

# Ініціалізація логування
logging.basicConfig(filename='robot_log.txt', level=logging.INFO, format='%(asctime)s - %(message)s')

# Ініціалізація модулів
tts = TTS()
music = Music()
sloth = Sloth([1, 2, 3, 4])
sloth.set_offset([0, 0, 0, 0])

# Ініціалізація ультразвукового сенсора
sonar = Ultrasonic(Pin("D2"), Pin("D3"))
ALERT_DISTANCE = 10
avoid_count = 0  # Лічильник уникнень

def avoid_obstacle():
    global avoid_count
    avoid_count += 1
    logging.info(f"Уникнення #{avoid_count}")
    try:
        music.sound_effect_threading('./sounds/sign.wav')
    except Exception as e:
        logging.error(f"Помилка відтворення звуку: {e}")
    sloth.do_action('hook', 1, 95)
    time.sleep(0.5)
    sloth.do_action('stand', 1, 95)
    time.sleep(0.5)

    # Випадковий вибір напрямку
    direction = random.choice(['left', 'right'])
    sloth.do_action(f'turn {direction}', 7, 90)
    logging.info(f"Роблю поворот {direction}")
    sloth.do_action('stand', 1, 95)
    time.sleep(0.2)

def move_forward():
    sloth.do_action('forward', 1, 90)
    logging.info("Рух вперед")

def main():
    distance = sonar.read()
    logging.info(f"Виміряна відстань: {distance} см")
    if distance < 0:
        logging.warning("Некоректні дані сенсора")
        return
    if distance <= ALERT_DISTANCE:
        avoid_obstacle()
    else:
        move_forward()

if __name__ == "__main__":
    try:
        while True:
            main()
            time.sleep(0.1)  # Запобігаємо надто частим вимірюванням
    except KeyboardInterrupt:
        print("\nЗупинка програми користувачем")
        logging.info("Програму зупинено користувачем")
    except Exception as e:
        print(f"Виникла помилка: {e}")
        logging.error(f"Неочікувана помилка: {e}")
```

Рисунок 3.11 – Програмний код

Поданий програмний код реалізує керування роботизованою платформою на базі модуля PiSloth з використанням мови Python та бібліотек pisloth і robot_hat. Основною метою алгоритму є реалізація автономного руху платформи з можливістю виявлення перешкод за допомогою ультразвукового сенсора та відповідного реагування.

На початку програми ініціалізуються основні об’єкти: голосовий синтезатор TTS, музичний модуль Music, об’єкт Sloth (що представляє робота з чотирма сервоприводами) та ультразвуковий сенсор Ultrasonic, підключений до пінів D2 (тригер) та D3 (exo). Параметр alert_distance, заданий на рівні 10 см, визначає поріг спрацьовування алгоритму уникнення перешкод.

Основна функція main() виконується у нескінченному циклі. На кожній ітерації вона зчитує відстань до об’єкта перед роботом за допомогою ультразвукового сенсора. Якщо значення відстані некоректне (негативне), програма переходить до наступної ітерації без дій. Якщо відстань менша або дорівнює пороговому значенню (alert_distance), виконується блок обробки перешкоди: програється звуковий сигнал (sign.wav), платформа переходить у стан «hook» (підняття корпусу), робить паузу 0,5 секунди, а потім повертається у положення «stand» (стояча поза) ще на 0,5 секунди. Після цього робот здійснює поворот ліворуч (команда ‘turn left’ із параметрами швидкості 7 та кута 90°), стає у позицію «stand» і чекає 0,2 секунди перед наступною командою.

Якщо ж перешкод не виявлено (відстань більша за поріг), робот продовжує рухатися вперед (команда ‘forward’ із параметрами швидкості 1 та кута 90°). Цикл повторюється нескінченно, що забезпечує безперервну перевірку середовища та реагування на зміну умов у реальному часі.

Загалом, алгоритм демонструє простий приклад поведінкового керування мобільною платформою, де рішення приймаються на основі даних сенсорів, а керуючі сигнали передаються на сервоприводи й мультимедійні модулі, що дозволяє досягти інтерактивної та автономної роботи системи.

ВИСНОВКИ

В кваліфікаційні роботі розглянутий актуальний напрямок, а саме робототехніка, яка є сьогодні однією з найдинамічніших сфер науково-технічного розвитку. Особливий інтерес викликають автономні мобільні платформи, які завдяки поєднанню сенсорних систем, алгоритмів навігації, комп'ютерного зору та адаптивного керування здатні виконувати складні завдання без безпосередньої участі людини. Вони знаходять застосування у сфері логістики, сільського господарства, військових розробок, медицини, побуту й навіть у розважальній індустрії, що підтверджує актуальність обраної теми.

В результаті аналізу були вивчені сучасні роботизовані платформи, що реалізують автономну навігацію, включаючи їх архітектуру, типи сенсорів, використовувані алгоритми й особливості апаратної реалізації. Отримані дані дозволили визначити ключові характеристики, які забезпечують ефективну роботу систем: гнучкість апаратної бази, енергетична ефективність, масштабованість та доступність програмного забезпечення. Це сформувало підґрунтя для прийняття обґрунтованих рішень у власному проєкті.

Проведений детальний огляд доступних апаратних рішень, серед яких як основний обчислювальний модуль було обрано Raspberry Pi завдяки його продуктивності, доступності й широкій підтримці розробників. У якості платформи програмування обрана мова Python, яка забезпечує простоту роботи з апаратним рівнем через бібліотеки GPIO та має численні інструменти для розробки алгоритмів навігації. Крім того, вибір ультразвукових сенсорів, сервоприводів і плати розширення був обґрунтований з точки зору сумісності та ефективності для поставлених завдань.

Здійснено проєктування та збір апаратної частини, що включає підключення мікрокомп'ютера, модулів сенсорів, виконавчих пристроїв та живлення. Всі компоненти інтегровані в єдину функціональну систему, яка забезпечує стабільну роботу в автономному режимі. В процесі розробки були

враховані фактори компактності, зручності монтажу та можливості розширення системи в майбутньому.

Реалізовано програмні модулі, що дозволяють платформі в реальному часі отримувати інформацію про навколишнє середовище, аналізувати наявність перешкод та будувати траєкторію руху. Алгоритми, які були впроваджені, забезпечують можливість уникнення зіткнень, слідування заданому маршруту або його коригування залежно від обставин. Ефективність роботи була підтверджена тестовими випробуваннями платформи у реальних умовах.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Тенденції ринку автономних мобільних роботів. *Grandviewresearch.* URL: https://www.grandviewresearch.com/industry-analysis/autonomous-mobile-robots-market (дата звернення: 02.04.2025).

2. Guide to Building a 2WD Robot with Arduino. *Medium.* URL: https://medium.com/@rafaellevissa/guide-to-building-a-2wd-robot-with-arduino-5f37dbbf4e2a (дата звернення: 08.04.2025).

3. Arduino Robotic Arm. *Cults3d.* URL: https://cults3d.com/ru/3d-model/gadzhet/arduino-robotic-arm (дата звернення: 08.02.2025).

4. Лінійний Arduino-робот. *Moyo.* URL: https://www.moyo.ua/ua/robot-konstruktor_makeblock_mbot_rangerransformable_stem_educational_kit/276744.html (дата звернення: 08.04.2025).

5. Розумна повнопривідна робо-платформа V2.0 з Bluetooth. *Arduino*. URL: https://arduino.ua/prod7227-sunfounder-picar-x-ai-video-robot-car-kit-for-raspberry-pi-chatgpt-4o-enabled-with-voice-command-video-recognition-python-scratch-camera-mic-rechargeable-batterry-rpi-not-included (дата звернення: 08.04.2025).

6. Autonomous robot for mapping using ultrasonic sensors. *Researchgate*. URL: https://www.researchgate.net/publication/323056345_Autonomous_robot_for_mapping_using_ultrasonic_sensors (дата звернення: 08.04.2025).

7. Design and Control of an Arduino-based Line Following Robo. *Diva-portal* URL: https://www.diva-portal.org/smash/get/diva2:1908071/fulltext01.pdf (дата звернення: 08.04.2025).

8. XiaoR STM32 Self Balancing Programmable Robot Robotics Bangladesh. *Google.* URL: https://www.google.com/url?sa=i&url=https%3A%2F%2Fstore.roboticsbd. 2Frobotics-parts%2F1254-xiaor-stm32-self-balancing-programmable-robot-robotics (дата звернення: 08.04.2025).

9. STM Kargu. *Wikipedia* URL: https://uk.wikipedia.org/wiki/STM_Kargu (дата звернення: 18.04.2025).

10. ESP32 Rover. *Hackaday.* URL: https://hackaday.com/wpcontent/uploads/2020/02/esp32car_feat.jpg (дата звернення: 18.04.2025).

11. Освітній ESP32-робот із Bluetooth-керуванням. *Ua.sz-kuongshun*. URL: https://ua.sz-kuongshun.com/uploads/10680/kuongshun-esp32-cam-smart-robot-car-esp32ff1ee.jpg (дата звернення: 18.04.2025).

12. Combined improved A* and greedy algorithm for path planning of multi-objective mobile robot. *Nature.* URL: https://www.nature.com/articles/s41598-022-17684-0 (дата звернення: 28.04.2025).

13. A coverage optimization algorithm for underwater acoustic sensor networks based on Dijkstra method. *Ieeexplore.* URL: https://ieeexplore.ieee.org/document/10194237 (дата звернення: 28.04.2025).

14. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. *Arxiv*. URL: https://arxiv.org/abs/2007.11898 (дата звернення: 28.04.2025).

15. Raspberry Pi 3 Model B+ Specifications. *Raspberrypi.* URL: https://raspberrypi.in.ua/wpcontent/ uploads/2018/03/datasheet_raspberry _pi_3_ model_b. pdf. (дата звернення: 30.04.2025).

16. Introduction to Raspberry Pi 3 B+. *Theengineeringprojects.* URL: https://www.theengineeringprojects.com/2018/07/introduction-to-raspberry-pi-3-b-plus.html (дата звернення: 30.04.2025).

17. Ультразвуковий сенсор HC-SR04. *Myproject*. URL: https://myproject.com.ua/hc-sr04-ultrazvukovij-datchik-vidstani-ua.html (дата звернення: 08.05.2025).

18. Getting started with the SG90 Servo. *Upesy.* URL: https://www.upesy.com/blogs/tutorials/esp32-servo-motor-with-arduino-code-sg90 (дата звернення: 08.05.2025).

19. SG90 9g сервопривід. *Myproject*. URL: https://myproject.com.ua/sg90-9g-servoprivid-ua.html (дата звернення: 08.05.2025).

20. Robot HAT. *Peppe8o* URL: https://peppe8o.com/raspberry-pi-robot-hat-sunfounder/ (дата звернення: 08.05.2025).

21. PiSloth User Manual. *Github*. URL: https://github.com/sunfounder/pisloth/blob/master/docs/PiSloth%20User%20Manual.pdf (дата звернення: 08.05.2025).