

**Міністерство освіти і науки України  
Луцький національний технічний університет  
Факультет комп'ютерних та інформаційних технологій  
Кафедра інженерії програмного забезпечення**

**КВАЛІФІКАЦІЙНА РОБОТА  
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КЕРУВАННЯ  
КІНЕМАТИКОЮ БАГАТОНОГО РОБОТА НА ПЛАТФОРМІ  
ARDUINO**

**DEVELOPMENT OF SOFTWARE FOR CONTROLLING THE  
KINEMATICS OF A MULTI-LEGGED ROBOT ON THE ARDUINO  
PLATFORM**

спеціальність 121 «Інженерія програмного забезпечення»  
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти  
групи ІПЗс-21

**Петленко Владислав Ігорович**

Керівник:

**Потейчук Михайло Іванович**

Кваліфікаційну роботу  
допущено до захисту

«10» 06 2025 р.

Гарант освітньої програми:

к.т.н., доцент

Ліщина Наталія Миколаївна



Луцьк – 2025 року

# ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри



«20» 12 2024р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Петленку Владиславу Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Розробка програмного забезпечення для керування кінематикою багатоналогого робота на платформі Arduino»

Керівник роботи: асистент, Потейчук М. І.

затверджені наказом закладу вищої освіти від «19» грудня 2024 р. № 474/01-02

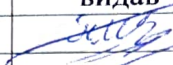
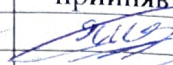
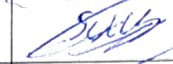
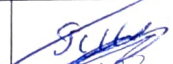
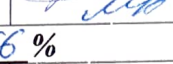
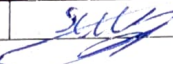
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра «10» червня 2025 р.

3. Вихідні дані до роботи: технічне та програмне забезпечення ЕОМ, вимоги до розробки програмного забезпечення, ергономічні вимоги до функціонування програмного засобу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): аналіз та огляд сучасного стану багатонагогих роботів і постановка завдань на кваліфікаційну роботу, визначення функціональних і технічних вимог до системи керування кінематикою та проєктування програмного забезпечення, обґрунтування вибору апаратних засобів, алгоритмів і методів керування, розробка прошивки й конфігураційних файлів для Arduino та складання механічної конструкції робота, тестування й налагодження програмно-апаратного комплексу, розробка бази даних шаблонів рухів і параметрів команд, впровадження заходів інформаційної безпеки та автентифікації Bluetooth-з'єднань, аналіз експериментальних результатів і рекомендації щодо подальшого розвитку системи.

5. Перелік графічного матеріалу: 16 рисунків

## 6. Консультанти розділів роботи

| Розділ                                    | Прізвище, ініціали та посада консультанта | Підпис                                                                             |                                                                                     |
|-------------------------------------------|-------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|                                           |                                           | завдання видав                                                                     | завдання прийняв                                                                    |
| Аналіз предметної області                 | Потейчук М. І.                            |  |  |
| Специфікація вимог до розробленої системи | Потейчук М. І.                            |  |  |
| Розробка об'єкта проектування             | Потейчук М. І.                            |  |  |
| Нормоконтроль                             | Вознюк А. В.                              |  |  |
| Гарант ОП                                 | Ліщина Н. М.                              |  |  |
| Показник запозичень тексту                | 0,56 %                                    |                                                                                    |                                                                                     |
| Академічна доброчесність                  | Потейчук М. І.                            |  |  |

## 7. Дата видачі завдання «20» грудня 2024 р.

| № | Назва етапів кваліфікаційної роботи бакалавра                                                   | Строк виконання етапів роботи | Примітка |
|---|-------------------------------------------------------------------------------------------------|-------------------------------|----------|
| 1 | Огляд літературних джерел по темі кваліфікаційної роботи бакалавра                              | до 04.02.2025 р.              | Виконано |
| 2 | Аналіз проблеми розробки та впровадження об'єкту проектування                                   | до 01.03.2025 р.              | Виконано |
| 3 | Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання               | до 15.03.2025 р.              | Виконано |
| 4 | Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних | до 29.03.2025 р.              | Виконано |
| 5 | Практична реалізація об'єкта проектування та розробка бази даних                                | до 26.04.2025 р.              | Виконано |
| 6 | Тестування та налагодження об'єкта проектування                                                 | до 03.05.2025 р.              | Виконано |
| 7 | Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі                            | до 10.06.2025 р.              | Виконано |

Здобувач вищої освіти



Владислав ПЕТЛЕНКО

Керівник кваліфікаційної роботи



Михайло ПОТЕЙЧУК

## АНОТАЦІЯ

Петленко В. І. Розробка програмного забезпечення для керування кінематикою багатоногого робота на платформі Arduino. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2025.

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

У першому розділі проведено комплексний аналіз сучасного стану розробок багатоногих (Hexapod) роботів, визначено переваги й обмеження існуючих платформ, а також окреслено актуальні проблеми точного керування кінематикою. Сформульовані завдання на кваліфікаційну роботу бакалавра.

У другому розділі проаналізовано функціональні та технічні вимоги до системи; розроблено програмну архітектуру та UML-діаграми, які описують обробку команд, планування ходи й реалізацію інверсної кінематики. Обґрунтовано вибір апаратних засобів (Arduino Mega 2560 PRO, сервоприводи SG90, модуль HC-05) та алгоритмів керування.

У третьому розділі описано практичну реалізацію програмно-апаратного комплексу: збирання механічної конструкції, розробку й оптимізацію прошивки для 18 сервоприводів, створення бази даних шаблонів рухів, впровадження засобів захисту й тестування системи на затримку реакції та стабільність руху. Наведено результати експериментів і рекомендовано напрями подальшого розвитку (автоматичне балансування, розширення набору ходових алгоритмів).

У результаті виконання роботи було створено програмно-апаратний комплекс для керування кінематикою багатоногого робота на платформі Arduino, що дозволило забезпечити високу точність, стабільність і гнучкість руху робота.

Ключові слова: багатонорий робот, серводвигун, мікроконтролер, ESP8266, Arduino.

## **ABSTRACT**

Petlenko V. Development of Software for Controlling the Kinematics of a Multi-legged Robot on the Arduino Platform. Manuscript. Qualification work of the bachelor's degree program "Software Engineering", specialty "Software Engineering". Lutsk National Technical University. Lutsk, 2025.

The bachelor's thesis consists of an introduction, three chapters, conclusions and suggestions, a list of references.

The first chapter provides a comprehensive analysis of the current state of multi-legged (hexapod) robot development, identifies the advantages and limitations of existing platforms, and outlines the pressing challenges of precise kinematic control. The tasks for the bachelor's thesis are formulated.

The second chapter gives a detailed analysis of the system's functional and technical requirements and develops the software architecture with UML diagrams that illustrate command processing, gait planning and inverse-kinematics implementation. The choice of hardware (Arduino Mega 2560 PRO, SG90 servomotors, HC-05 module) and control algorithms is substantiated.

The third chapter describes the practical implementation of the hardware–software complex: mechanical assembly, development and optimization of firmware for 18 servomotors, creation of a motion-pattern database, integration of protection mechanisms, and testing of the system for response latency and motion stability. Experimental results are presented, and prospects for further improvement are substantiated (automatic balancing, extension of gait algorithms).

As a result of this work, a hardware and software complex was developed for controlling the kinematics of a multi-legged robot on the Arduino platform, which ensured high accuracy, stability, and flexibility of the robot's movement.

Keywords: multi-legged robot, servo motor, microcontroller, ESP8266, Arduino.

## ЗМІСТ

|                                                                                                                        |    |
|------------------------------------------------------------------------------------------------------------------------|----|
| ВСТУП .....                                                                                                            | 7  |
| РОЗДІЛ 1 АНАЛІЗ ТЕХНОЛОГІЙ БАГАТОНОГИХ РОБОТІВ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ .....                     | 9  |
| 1.1 Аналіз сучасного стану проблеми .....                                                                              | 9  |
| 1.2 Постановка завдання на кваліфікаційну роботу бакалавра .....                                                       | 15 |
| РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ .....                                                             | 16 |
| 2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення ..... | 16 |
| 2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання .....                                          | 20 |
| РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ ДЛЯ КЕРУВАННЯ КІНЕМАТИКОЮ БАГАТОНОГОГО РОБОТА .....                   | 26 |
| 3.1 Практична реалізація об'єкта проектування .....                                                                    | 26 |
| 3.2 Тестування та налагодження програмно-апаратного комплексу .....                                                    | 30 |
| 3.3 Розробка бази даних для програмно-апаратного комплексу .....                                                       | 33 |
| 3.4 Захист інформаційно-комп'ютерної системи .....                                                                     | 35 |
| 3.5 Перспективи подальшого розвитку системи керування кінематикою багатоногого робота .....                            | 36 |
| ВИСНОВКИ .....                                                                                                         | 39 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....                                                                                       | 41 |

## ВСТУП

Актуальність теми. Розвиток робототехнічних систем вимагає вдосконалення програмного забезпечення для управління кінематикою багатонагних роботів. Важливим аспектом є створення зручного та гнучкого інтерфейсу, який дозволить користувачам ефективно керувати рухами робота, змінювати його параметри та інтегрувати систему з іншими автоматизованими платформами. Реалізація алгоритмів для координації та адаптивного руху дозволяє досягти високої точності, стабільності та плавності переміщень, що є ключовим фактором для використання багатонагних роботів у різних сферах діяльності.

Особливу актуальність має створення програмного забезпечення для контролю кінематики багатонагних роботів. Розробка зручного та ефективного інтерфейсу дозволяє операторам легко налаштовувати систему, регулювати рухи робота та забезпечувати його інтеграцію з іншими автоматизованими технологіями. Впровадження адаптивних алгоритмів управління рухом дозволяє досягти високої точності, плавності та стабільності роботи багатонагного робота на різних типах поверхонь.

Метою роботи є розробка програмного забезпечення для керування кінематикою багатонагного робота на платформі Arduino.

Об'єктом роботи є програмно-апаратний комплекс для керування кінематикою багатонагного робота, що використовує серводвигуни та мікроконтролери для точного контролю рухів.

Предметом роботи виступає розробка алгоритмів та методів автоматизованого управління рухами багатонагного робота з метою підвищення його ефективності та адаптивності до різних умов експлуатації.

Виходячи з мети роботи, було сформульовано наступні завдання:

- провести аналіз сучасного стану розробок у галузі багатонагних роботів, дослідити актуальні проблеми, технології та практичні приклади реалізацій;

- визначити функціональні та технічні вимоги до програмного забезпечення системи управління кінематикою багатоногого робота та спроектувати сценарії використання й послідовності команд;
- здійснити вибір апаратних засобів, алгоритмів і методів керування, оптимізованих для керування рухами багатоногого робота;
- виконати практичну реалізацію програмно-апаратного комплексу, в тому числі, зібрати механічну конструкцію, підключити електронні компоненти та розробити код управління;
- провести тестування та налагодження всієї системи, оцінити точність роботи, реакцію на команди, стабільність платформи та виконати ручну і програмну калібровку;
- розробити базу даних для збереження шаблонів рухів, параметрів команд і просторових координат;
- реалізувати базові заходи захисту інформаційно-комп'ютерної системи, зокрема контроль доступу, фільтрацію команд і автентифікацію підключень до Bluetooth-модуля;
- проаналізувати перспективні напрями розвитку та запропонувати можливі апаратні й програмні покращення системи.

## РОЗДІЛ 1

### АНАЛІЗ ТЕХНОЛОГІЙ БАГАТОНОГИХ РОБОТІВ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

#### 1.1 Аналіз сучасного стану проблеми

На сучасному етапі розвитку робототехніки одним із пріоритетних напрямків є створення автономних роботизованих систем, здатних ефективно функціонувати у складних умовах. Розвиток обчислювальної техніки, поява високопродуктивних енергоефективних сервоприводів і широке застосування алгоритмів машинного навчання відкрили нові можливості для створення мобільних платформ, які можуть самостійно орієнтуватися в довкіллі, адаптуватися до змін рельєфу та виконувати складні операції без участі оператора.

Особливий інтерес у цьому контексті становлять багатоногі роботи – зокрема Нехарод-роботи [1], конструкція яких забезпечує вищий рівень стійкості та прохідності порівняно з колісними чи гусеничними аналогами. Завдяки здатності одночасно взаємодіяти з кількома точками опори, такі роботи можуть працювати на нерівних, сипких і навіть слизьких поверхнях, долати перешкоди типу каміння, коренів і щілин, а також здійснювати підйоми й спуски схилами з кутом нахилу до 30-40.

Нехарод-роботи характеризуються високою маневреністю та універсальністю, адже кожна з шести кінцівок має по 3-4 ступені свободи, що дозволяє здійснювати не лише переміщення в шести напрямках простору, а й виконувати тонке коригування положення корпусу відносно опори.

Сучасні Нехарод-платформи використовують алгоритми зворотного кінематичного моделювання (Inverse Kinematics) та динамічного балансування, що дозволяє не лише розраховувати потрібні кути обертання кожного серводвигуна для одержання бажаного вектора руху, а й забезпечувати миттєве перерозподілення навантажень при зіткненні з перешкодами чи нерівностями поверхні [2].

На сьогоднішній день розроблено низку різноманітних конструкцій багатоногих роботів, які мають практичні застосування у різних сферах:

- військових (патрулювання, розмінування);
- пошуково-рятувальних (проникнення в обвалені будівлі, доставка медикаментів);
- дослідницьких (експедиції в віддалені регіони, збір наукових проб) та освітніх (лабораторні стенди для вивчення робототехніки).

Серед відкритих проєктів, що активно підтримуються спільнотами Maker та STEM-освіти, особливо вирізняється Нехарod-робот на базі Arduino, опублікований на Thingiverse і Instructables.

Ця модульна конструкція дозволяє не лише швидко зібрати платформу з 3D-друкованих деталей та стандартних сервоприводів SG90 або MG90, а й легко модернізувати її функціонал: замінити контролер на потужніший STM32, інтегрувати LIDAR-сенсори для картографування простору, додати бездротовий модуль зв'язку для керування з мобільного додатку чи підключення до ROS. Такий гнучкий підхід сприяє швидким ітераціям у розробці, обміну досвідом між інженерами та студентами.

На рисунку 1.1 наведено приклад конструкції багатоногого робота, створеного за кресленнями з Thingiverse.



Рисунок 1.1 – Конструкція багатоногого Нехарod-робота з сайту Thingiverse [3]

Основною перевагою Нехарод-роботів є їхня здатність ефективно переміщатися складними поверхнями, що пояснюється високою адаптивністю кінематичної системи. Завдяки шести кінцівкам, кожна з яких може змінювати положення у просторі незалежно від інших, платформа швидко реагує на нерівності та перешкоди. Проте для досягнення цього рівня автономності та стабільності необхідно реалізувати низку складних обчислювальних задач: від розрахунку оптимальних траєкторій руху до адаптивного коригування положення корпусу в реальному часі [4].

Важливо також враховувати зовнішні чинники, такі як ковзання поверхні, зміна центру мас при підйомі або спуску та непередбачувані зіткнення з дрібними перешкодами. Усе це висуває високі вимоги до апаратної платформи – швидкісних контролерів, потужних мікропроцесорів, систем датчиків – та до програмного забезпечення, яке має працювати з низькою затримкою й забезпечувати плавну взаємодію між підсистемами.

Створення ефективного програмного забезпечення, що дозволить реалізувати точні, плавні та стабільні рухи робота, є складним завданням через необхідність врахування великої кількості факторів, таких як балансування, синхронізація рухів ніг, розподіл ваги та стабільність на нерівних поверхнях.

У сучасних розробках зазвичай використовують модулі зворотного кінематичного моделювання (Inverse Kinematics), алгоритми генерації ходів (Gait Planning) та засоби обробки даних з інерційних та візуальних сенсорів. Крім того, для підвищення надійності часто застосовують технології захисту від аномальних ситуацій: датчики сили попереджають надмірне навантаження на сервоприводи, а системи моніторингу виконують безперервну діагностику стану моторов та з'єднань.

Історично першим промисловим роботом вважається Unimate, створений у 1961 році компанією Unimation. Ця машина плавно виконувала повторювані, рутинні операції на виробничих лініях автомобільної промисловості, що значно підвищило продуктивність та безпеку праці. Відтоді робототехніка пройшла шлях від жорстких маніпуляторів до гнучких і кооперативних систем, здатних

працювати поруч із людиною. Сучасні багатофункціональні роботи поєднують можливості адаптивного управління, машинного навчання та віддаленого моніторингу, що дозволяє їм швидко перенавчатися під нові завдання та інтегруватися в цифрові виробничі середовища Industry 4.0.

Прикладом сучасних роботів-маніпуляторів, що активно застосовуються у промисловості, є лінійка універсальних колаборативних роботів Universal Robots (UR3, UR5, UR10). Вони характеризуються не лише високою точністю виконання повторюваних завдань, але й відзначаються легкістю налаштування та перепрограмування завдяки зручному та інтуїтивно зрозумілому інтерфейсу. Завдяки відкритим API та підтримці стандартних комунікаційних протоколів, ці роботи активно використовуються для широкого спектру завдань, серед яких зварювання, пакування продукції, обробка різноманітних матеріалів, складання деталей та компонентів.

Важливою перевагою роботів серії Universal Robots є їхня здатність до безпечної роботи в безпосередньому контакті з людиною без додаткових захисних огорожень, що забезпечує їх активне застосування навіть на підприємствах з обмеженим простором та у напівскладних виробничих процесах, які вимагають частого втручання операторів.

Іншим яскравим прикладом високотехнологічних промислових роботів є серія KR QUANTEC PA, розроблена компанією KUKA Robotics. Ці роботи-маніпулятори спеціально спроектовані для роботи в складних умовах навколишнього середовища: вони здатні стабільно функціонувати при низьких температурах до  $-20^{\circ}\text{C}$ , в умовах підвищеної вологості та високого рівня запиленості. Це стає можливим завдяки застосуванню спеціального захисного покриття, герметичної конструкції та посиленої електронної компоненти, яка забезпечує стійкість до агресивних умов експлуатації.

Вбудовані системи моніторингу та управління дозволяють контролювати рівень вібрацій і температуру сервомоторів, що значно подовжує термін експлуатації роботів та знижує ймовірність виникнення аварійних ситуацій через перегрів. Завдяки цим характеристикам, маніпулятори KR QUANTEC PA є

незамінними у таких критично важливих сферах, як гірничодобувна промисловість, харчова та фармацевтична галузі, де надійність і стабільність роботи є вирішальними факторами.

На рисунку 1.2 наведено приклади сучасних промислових маніпуляторів.



Рисунок 1.2 – Промислові маніпулятори компанії Universal-robots та KUKA Robotics

Незважаючи на наявність різноманітних конструкцій роботизованих маніпуляторів та багатоногих платформ, питання ефективного програмного керування їх кінематикою залишається відкритим і надзвичайно складним. По-перше, існуючі рішення часто спираються на жорстко закладені параметри кінематичних ланцюгів, що унеможлиблює швидке переналаштування під інші типи серводвигунів чи геометрію ніг без глибокого переписування коду. По-друге, синхронізація шести й більше приводів із врахуванням реального часу та нестабільності зв'язку (наприклад, при бездротовому керуванні) накладає високі вимоги на пропускну здатність контролера та якість протоколів передачі даних.

Ще однією суттєвою проблемою є балансування під час руху: без автоматизованих алгоритмів розпізнавання опорних точок, адаптивного оновлення центру мас і коригування Gait-планування платформа може втрачати

стабільність на нерівних чи рухомих поверхнях. Багато сучасних розробок використовують статичне планування ходів, яке припускає, що середовище заздалегідь відоме або не змінюється, тоді як для реальних умов потрібні динамічні алгоритми зворотної кінематики, спроможні оперативно враховувати дані з IMU, фотореалістичних сенсорів чи LIDAR.

Крім того, складність налаштування та калібрування серводвигунів (від точності квантування кутів до компенсації гістерезису й теплових деформацій) часто вимагає залучення спеціаліста й тривалого експериментального відлагодження. Відсутність зручного інтерфейсу для інтеграції нових приводів, тестування різних ходових режимів і візуалізації результатів у серйозному обмежує можливості дослідників і дизайнерів застосовувати багатоніг платформ в універсальних завданнях [5].

Отже, актуальною залишається задача розробки універсального, модульного та гнучкого програмного забезпечення, яке поєднуватиме:

- адаптивний рушій кінематики з підтримкою різних типів сервоприводів і швидкою калібруванням в полі;
- динамічне планування ходів із використанням реальних даних сенсорів (IMU, тензодатчиків, LIDAR) та алгоритмів машинного навчання для передбачення змін рельєфу;
- масштабований інтерфейс API (з сумісністю ROS, MQTT, WebSocket), який дозволить швидко додавати нові модулі поведінки, візуалізації та діагностики;
- інструменти валідації та симуляції (Gazebo, Webots), інтегровані з конвейєром CI/CD, що забезпечать автоматичне тестування на різних конфігураціях ще до виведення на апаратний рівень;
- надійні засоби захисту й моніторингу (обмеження струму, контроль температури, швидке відключення при критичних навантаженнях) для гарантування безпеки у реальних умовах експлуатації.

Розробка такого рішення дозволить значно розширити сферу застосування багатонігих роботів на серводвигунах – від автономного дослідження складних

поверхонь і інспекції важкодоступних об'єктів до співпраці з людиною в логістиці та медицині, забезпечить високу точність рухів і стабільність роботи в умовах реального світу.

## **1.2 Постановка завдання на кваліфікаційну роботу бакалавра**

З урахуванням зазначеної мети та технічної специфіки розроблюваної системи, було визначено перелік ключових завдань, необхідних для успішного досягнення поставленої цілі та реалізації програмно-апаратного комплексу:

- провести аналіз сучасного стану розробок у галузі багатонагох роботів, дослідити актуальні проблеми, технології та практичні приклади реалізацій;
- визначити функціональні та технічні вимоги до програмного забезпечення системи управління кінематикою багатонагого робота та спроектувати сценарії використання й послідовності команд;
- здійснити вибір апаратних засобів, алгоритмів і методів керування, оптимізованих для керування рухами багатонагого робота;
- виконати практичну реалізацію програмно-апаратного комплексу, в тому числі, зібрати механічну конструкцію, підключити електронні компоненти та розробити код управління;
- провести тестування та налагодження всієї системи, оцінити точність роботи, реакцію на команди, стабільність платформи та виконати ручну і програмну калібровку;
- розробити базу даних для збереження шаблонів рухів, параметрів команд і просторових координат;
- реалізувати базові заходи захисту інформаційно-комп'ютерної системи, зокрема контроль доступу, фільтрацію команд і автентифікацію підключень до Bluetooth-модуля;
- проаналізувати перспективні напрями розвитку та запропонувати можливі апаратні й програмні покращення системи.

## РОЗДІЛ 2

### СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

#### 2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

На сучасному етапі розробки роботизованих систем питання ефективного програмного керування займає центральне місце, особливо коли мова йде про багатонігі конструкції з високою складністю кінематики. Правильно сформульовані вимоги дозволяють не лише закласти фундамент для подальшої реалізації всіх функціональних можливостей, а й провести якісну перевірку та верифікацію розробленого рішення.

Враховуючи обрану апаратну платформу та очікуваний спектр сценаріїв використання – від дослідження складних поверхонь до інтерактивної взаємодії через Bluetooth – було виділено два взаємопов'язаних блоки вимог: функціональні та технічні [6].

Функціональні вимоги до програмного забезпечення включають такі ключові напрямки, кожен із яких відіграє важливу роль у забезпеченні надійності та гнучкості системи.

Точне позиціонування кожної ноги у просторі за координатами  $X$ ,  $Y$ ,  $Z$ . Це дозволить коректно розраховувати траєкторії руху та підтримувати бажану висоту та орієнтацію корпусу без зсувів чи вібрацій.

Реалізація кількох режимів руху: прямолінійного, поворотного та зворотного. Універсальність режимів забезпечує адаптацію до різних завдань – від швидкого прямих переміщень до маневрування у обмеженому просторі.

Забезпечення синхронізації фаз руху для всіх шести ніг. Координація фаз є критичною для уникнення конфліктів між кінцівками та для підтримки стійкості під час переходу між режимами руху.

Можливість зміни параметрів руху в реальному часі (довжина кроку, висота підйому, швидкість). Така гнучкість дозволяє оператору або автономній

системі підлаштовуватися під поточні умови довкілля, наприклад, збільшувати крок для рівної поверхні або зменшувати його на слизькому схилі.

Підтримка Bluetooth-керування з можливістю передачі команд у режимі реального часу. Використання бездротового інтерфейсу відкриває можливості для дистанційного контролю та інтеграції з мобільними додатками чи промисловими контролерами.

Адаптація до нерівностей поверхні та мінімізація коливань платформи. Інтелектуальні алгоритми корекції положення та компенсації похибок руху дозволяють уникати перевантаження сервоприводів і забезпечувати плавність пересування навіть у складних умовах.

Технічні вимоги сформульовано з урахуванням обраних компонентів та обмежень апаратної частини, що визначає кадрову і матеріальну базу проєкту:

Повна сумісність із мікроконтролером Mega 2560 PRO. Це дає змогу використовувати наявні бібліотеки Arduino, реалізувати відлагодження коду в звичному середовищі та заощадити час на налагодження периферії.

Керування сервоприводами SG90 без зовнішніх PWM-контролерів (PCA9685). Виключення додаткових модулів знижує вартість і спрощує апаратну схему, але накладає підвищені вимоги на точність генерації ШІМ сигналів.

Використання вбудованих апаратних таймерів Arduino для формування ШІМ-сигналів. Апаратна підтримка гарантує стабільність частоти імпульсів та знижує затримки, необхідні для коректної роботи сервоприводів.

Інтеграція з Bluetooth-модулем HC-05 (ZS-040) через UART. Забезпечення надійного зв'язку з мінімальною затримкою й одночасна підтримка діагностичних повідомлень дозволяють реалізувати двосторонній обмін даними.

Реалізація алгоритмів зворотної кінематики для розрахунку положення ніг. Автоматизовані розрахунки кута повороту кожного приводу прискорюють процес розробки нових ходів і спрощують масштабування системи на інші моделі роботів.

Оптимізований код у середовищі Arduino IDE мовою C++. Використання професійних практик програмування та шаблонів коду сприяє підтримуваності, читаємості та подальшому розширенню проєкту.

На основі детального аналізу взаємодії користувача з роботом побудовано діаграму варіантів використання (Use Case Diagram), яка була наведена на рисунку 2.1. На діаграмі чітко виділено акторів: оператор (користувач системи), контролер Arduino Mega 2560 PRO та Bluetooth-модуль HC-05, а також їх взаємодію із гілками функціональності системи.

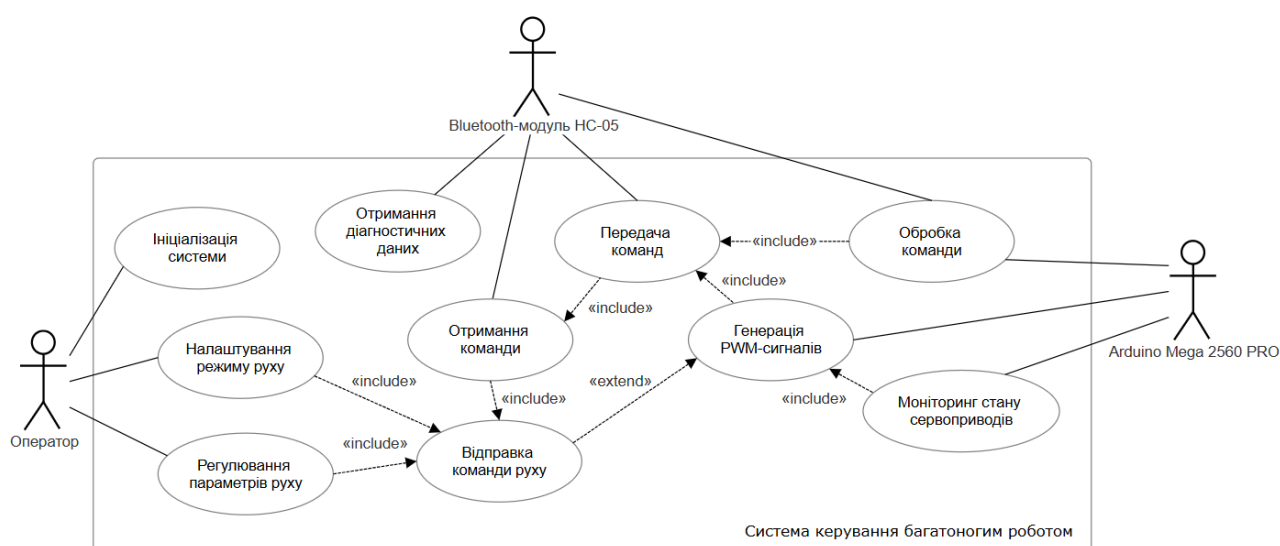


Рисунок 2.1 – UML-діаграма випадків використання системи керування багатоногом роботом

Завдяки такому відображенню користувацьких сценаріїв забезпечується наочність процесів і полегшується перевірка реалізації вимог у ході тестування та подальшого розширення функціоналу.

На рисунку 2.2 показано взаємодію всіх основних логічних модулів системи від прийому користувацької команди до безпосереднього виконання руху та моніторингу стану сервоприводів. Такий підхід дозволяє чітко визначити вузькі місця у послідовності обробки і спрощує подальшу валідацію та тестування кожного етапу окремо [7].

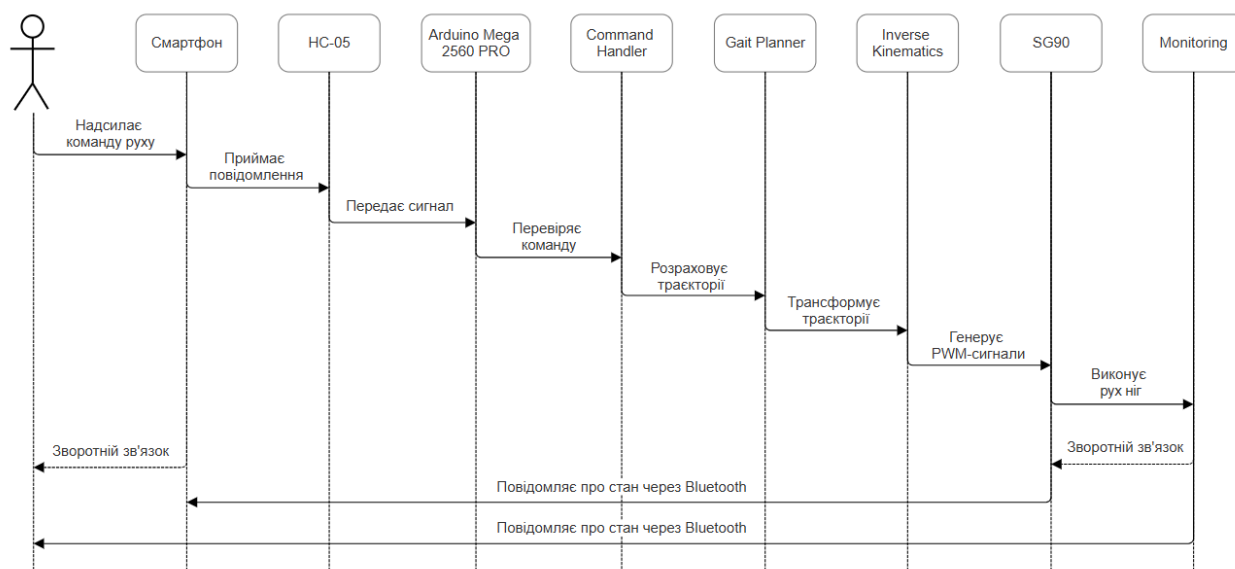


Рисунок 2.2 – Діаграма послідовностей обробки команди руху

Ця діаграма ілюструє послідовність обробки команди руху від моменту її отримання до виконання сервоприводами:

- користувач надсилає команду руху (наприклад, «Forward», «Turn Left») через мобільний додаток на смартфоні;
- Bluetooth-модуль HC-05 приймає UART-повідомлення та передає його контролеру Arduino Mega 2560 PRO;
- модуль обробки команд (Command Handler) на контролері перевіряє коректність формату та параметрів команди;
- модуль планування ходів (Gait Planner) підбирає відповідний шаблон руху та розраховує базові траєкторії для шести ніг;
- модуль зворотної кінематики (Inverse Kinematics) трансформує траєкторії у кути обертання кожного серводвигуна й передає їх на PWM-генератор;
- апаратні таймери Arduino генерують необхідні PWM-сигнали для закладених у код каналів управління SG90;
- сервоприводи SG90 отримують імпульси та виконують рухи ніг відповідно до розрахованих кутів;

– датчики позиції та сили (за наявності) відправляють зворотний зв'язок у модуль діагностики (Monitoring), який у разі перевантажень або збоїв ініціює коригувальні дії або повідомляє користувача через Bluetooth.

У процесі проектування було передбачено створення низки класів, для обчислення координат кінцівок у просторі, формування команд для кожного сервоприводу, обробки сигналів від Bluetooth-модуля, а також для реалізації базових циклів руху. Для обчислення кутів використовуються тригонометричні функції та вектори положення, а формування послідовності руху базується на відомому алгоритмі трифазного кроку.

Таким чином, чітке визначення вимог та побудова UML-діаграм забезпечують ефективне проектування програмного забезпечення, що відповідає особливостям апаратної реалізації та забезпечує стабільну роботу багатоналогового робота.

## **2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання**

У процесі реалізації програмно-апаратного комплексу для керування кінематикою багатоналогового робота особливу увагу було приділено вибору апаратних засобів, які забезпечать надійність, масштабованість та простоту інтеграції. Підбір компонентів здійснювався з урахуванням вимог до кількості керованих виконавчих механізмів, підтримки зворотного зв'язку та можливості розширення функціоналу в майбутньому.

Основною керуючою платформою обрано Mega 2560 PRO – компактну версію плати Arduino Mega 2560 з аналогічною функціональністю, яка займає менше місця на платі. Вона має 54 цифрові лінії вводу/виводу, з яких 15 можуть використовуватись як ШІМ-виходи, 16 аналогових входів, 256 кБ флеш-пам'яті, 8 кБ SRAM та 4 UART-порти. Зовнішній вигляд плати подано на рисунку 2.4.



Для забезпечення дистанційного керування роботом було обрано Bluetooth-модуль HC-05 (також відомий як ZS-040). Цей модуль працює за специфікацією Bluetooth 2.0 + EDR та забезпечує передавання даних зі швидкістю до 3 Мбіт/с. Інтеграція модуля відбувається через послідовний інтерфейс UART, що дозволяє легко підключати його до апаратних UART-портів Arduino Mega 2560 PRO [11].

При старті мікроконтролер ініціює серійний зв'язок з модулем HC-05 через апаратний UART з налаштуваннями 9600 біт/с (або 57600 біт/с в AT-режимі). Після встановлення з'єднання модуль автоматично обмінюється даними з парним пристроєм (смартфон або комп'ютер), а фрейми управління рухом ногами передаються у вигляді простих ASCII-команд чи двійкових пакетів. Це дає змогу операторам за допомогою мобільного застосунку чи комп'ютера відправляти команди на зміну режиму руху, швидкості чи траєкторії.

Модуль HC-05 (рис. 2.6) має робочий радіус до 10 м у відкритому просторі та витримує струм живлення до 30 мА у режимі очікування та до 60 мА під час передачі. Індикаторний світлодіод на платі блимає під час очікування з'єднання і горить постійно при активному зв'язку, що спрощує діагностику стану бездротового каналу [12].



Рисунок 2.6 – Bluetooth-модуль HC-05 [13]

Важливим аспектом є реалізація алгоритму керування рухом, який базується на методі зворотної кінематики. Для кожної ноги розробляється модель з трьома ступенями свободи, яка дозволяє визначити точне положення кінцівки у просторі при заданих координатах X, Y, Z. Геометричну модель та приклад розрахунку представлено на рисунку 2.7.

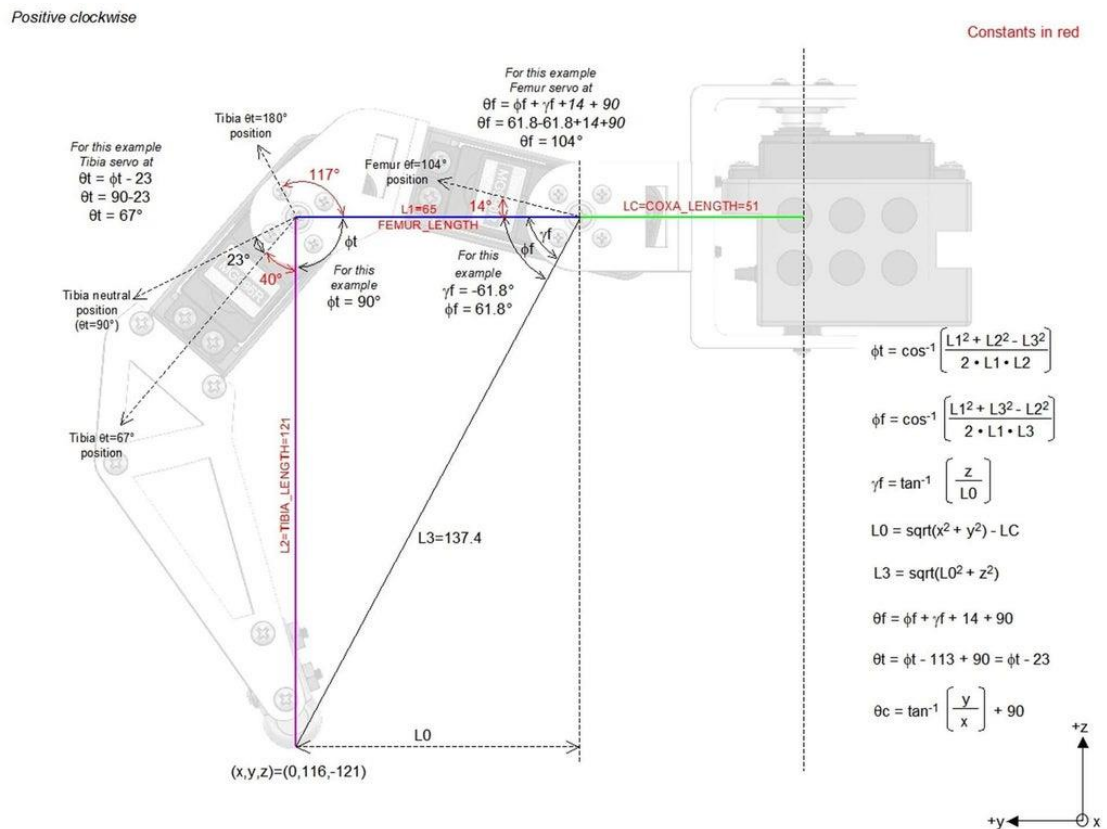


Рисунок 2.7 – Схема зворотної кінематики для ноги Нехарод-робота [14]

Розрахунок руху багатоногого робота здійснюється з урахуванням геометричних параметрів конструкції кожної ноги, таких як довжина сегментів і кут повороту сервоприводів, що відповідають за рух. Алгоритм керування рухом включає три послідовні фази для забезпечення стабільного і плавного переміщення робота:

- фаза підйому ноги;
- фаза переносу ноги;
- фаза опускання ноги.

На першому етапі робот піднімає одну з ніг, звільняючи її від контакту з поверхнею. Алгоритм враховує поточне положення сегментів та кути серводвигунів, обчислює необхідні кути повороту для плавного підйому ноги без порушення рівноваги робота.

Після підйому ноги робот здійснює горизонтальний переніс кінцівки до нової цільової позиції. На цьому етапі алгоритм визначає оптимальну траєкторію руху ноги, враховуючи необхідність уникнення перешкод та мінімізації навантаження на інші ноги, які забезпечують підтримку конструкції. Координати руху обчислюються за допомогою кінематичних рівнянь та геометричних співвідношень, закладених у програмний код.

На завершальному етапі робот плавно опускає ногу, встановлюючи її на поверхню у визначеній точці. Алгоритм керує сервоприводами для точного позиціонування сегментів ноги, що дозволяє розподілити навантаження рівномірно та стабільно підтримувати конструкцію робота на поверхні.

Ці фази виконуються циклічно і по черзі для кожної пари ніг, створюючи координований та збалансований рух робота. Алгоритм реалізується за допомогою мови програмування C++ [15], що забезпечує високий рівень продуктивності та точності обчислень. Для реалізації використовуються математичні функції (зокрема, тригонометричні розрахунки) та логічні оператори, що дозволяють динамічно адаптувати траєкторії руху до умов навколишнього середовища.

Як середовище розробки обрано Arduino IDE [16], яке є спеціалізованим інтегрованим середовищем для програмування мікроконтролерів серії Arduino. Arduino IDE забезпечує зручний і простий інтерфейс користувача, що включає текстовий редактор коду з підсвічуванням синтаксису, функції автодоповнення та автоматичної перевірки коду на помилки. Інтегроване середовище має вбудований компілятор, що дозволяє швидко та зручно компілювати програмний код, і завантажувач (bootloader), який забезпечує простий процес завантаження прошивки безпосередньо на плату мікроконтролера.

Arduino IDE підтримує мови програмування C та C++, що дозволяє ефективно використовувати ці мови для розробки прошивки. Крім того, середовище надає можливість інтеграції великої кількості спеціалізованих бібліотек, таких як Servo.h (для керування сервоприводами) та Wire.h (для роботи з інтерфейсом I2C), що значно спрощує процес розробки складних функціональних модулів та розширює можливості кінцевого пристрою.

Таким чином, використані апаратні та програмні рішення забезпечують ефективну систему керування багатоногим роботом, яка легко масштабується, дозволяє інтегрувати нові функції та модулі, а також підтримує подальшу модернізацію і розвиток конструкції.

## РОЗДІЛ 3

### РОЗРОБКА ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ ДЛЯ КЕРУВАННЯ КІНЕМАТИКОЮ БАГАТОНОГО РОБОТА

#### 3.1 Практична реалізація об'єкта проектування

Практична реалізація багатоногого робота включала поетапну збірку механічної конструкції, підключення електронних компонентів, розробку програмного коду для керування рухом та тестування системи в реальних умовах.

Першим кроком стала реалізація базового контролю серводвигунів. Було підключено три серводвигуни до мікроконтролера Mega 2560 PRO безпосередньо до цифрових PWM-виводів. На рисунку 3.1 зображено схематичне підключення трьох сервомоторів SG90 до мікроконтролера.

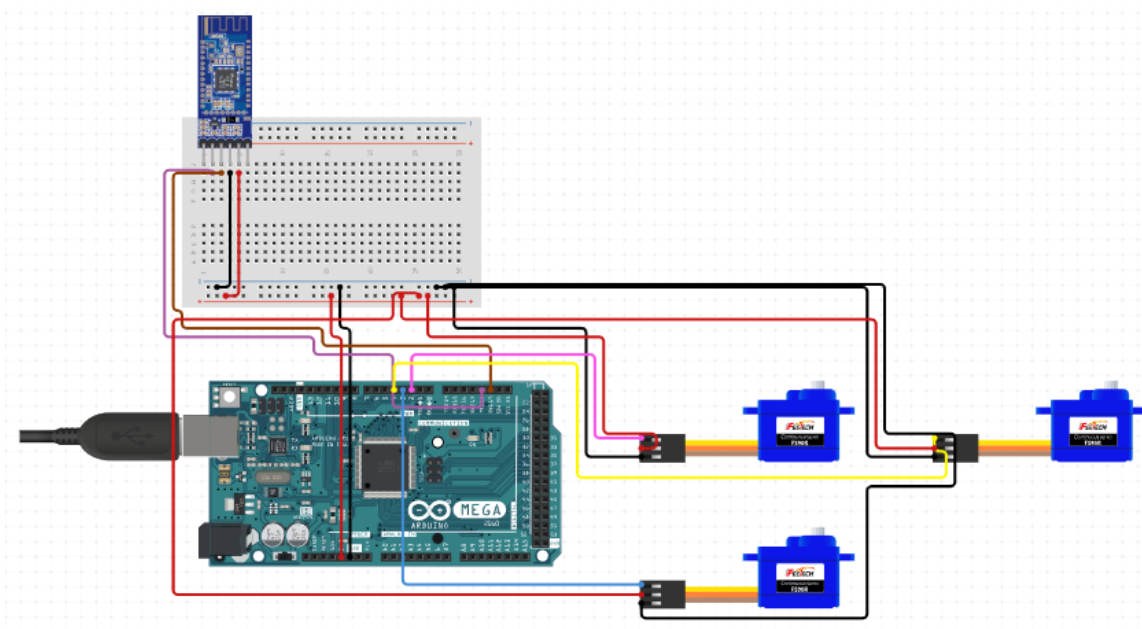


Рисунок 3.1 – Схематичне підключення трьох серводвигунів до мікроконтролера Mega 2560 PRO

Для кожної ноги реалізовано окрему функцію керування трьома серводвигунами. Приклад базового коду для ініціалізації та тестування рухів подано нижче (ліст. 3.1).

### Лістинг 3.1 – Керування серводвигунами

---

```
#include <Servo.h>

Servo servo1; // Перший сервомотор, підключений до піна D2
Servo servo2; // Другий сервомотор, підключений до піна D3
Servo servo3; // Третій сервомотор, підключений до піна D4

void setup() {
  servo1.attach(2); // Пін 2 для двигуна 1
  servo2.attach(3); // Пін 3 для двигуна 2
  servo3.attach(4); // Пін 4 для двигуна 3
}

void loop() {
  servo1.write(0);
  delay(500); // Встановлюємо кут 0° для servo1 і чекаємо 500 мс
  servo1.write(90);
  delay(500); // Повертаємо servo1 у позицію 90° і робимо паузу
  servo2.write(45);
  delay(500); // Встановлюємо кут 45° для servo2 – пауза 500 мс
  servo3.write(135);
  delay(500); // Встановлюємо кут 135° для servo3 – пауза 500 мс
}
```

---

кінець лістингу 3.1

На наступному етапі було виконано монтаж першої ноги робота. Кожен сегмент виготовлений із PLA-пластика на 3D-принтері з долею заповнення 40 %, що забезпечує достатню жорсткість при невеликій масі. З'єднання між сегментами реалізовані трьома сервоприводами SG90: базовий сервопривод відповідає за обертання стегна у горизонтальній площині (ось Z), другий – за підйом і опускання гомілки (ось Y), третій – за згин стопи (ось X), утворюючи, таким чином, три ступені свободи.

Кожен сервомеханізм закріплено в друкованих «кишенях» корпусу за допомогою гвинтів M2,5 з різьбовим фіксатором, а вихідні вали додатково підперті втулками з латуні, що зменшує люфти. Усередині сегментів прокладено сигнальні та силові проводи AWG-24. На рисунку 3.2 представлено зображення зібраної ноги робота з закріпленими сервомеханізмами.



Рисунок 3.2 – Зібрана нога багатоногого робота з трьома сервомеханізмами

Наступним кроком стало встановлення електронної начинки (мікроконтролер, модуль Bluetooth, блок живлення) у корпус. Конструкція корпусу надрукована на 3D-принтері за кресленнями з ресурсу Thingiverse. На рисунку 3.3 зображено корпус з усіма елементами, окрім ніг.

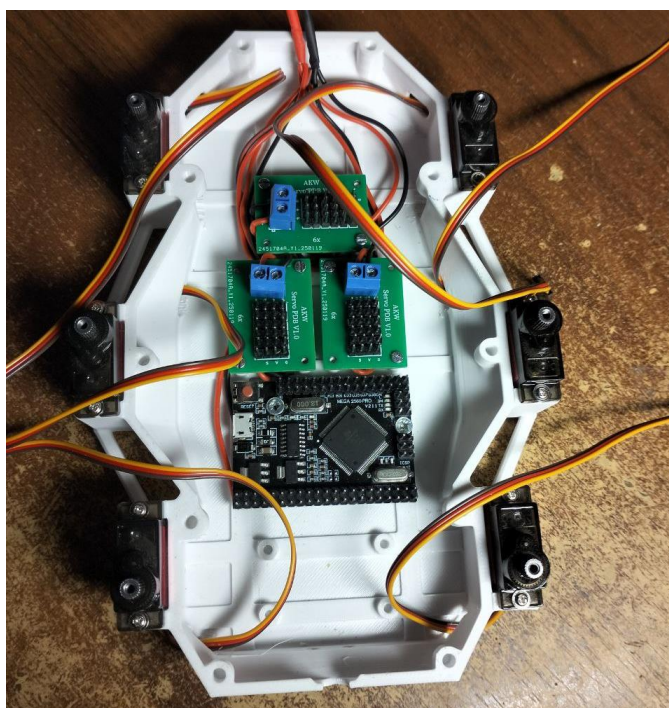


Рисунок 3.3 – Корпус робота з установленою електронікою

Після перевірки роботи однієї ноги була виконана повна збірка всіх шести ніг робота. На рисунку 3.4 показано загальний вигляд всіх зібраних кінцівок.

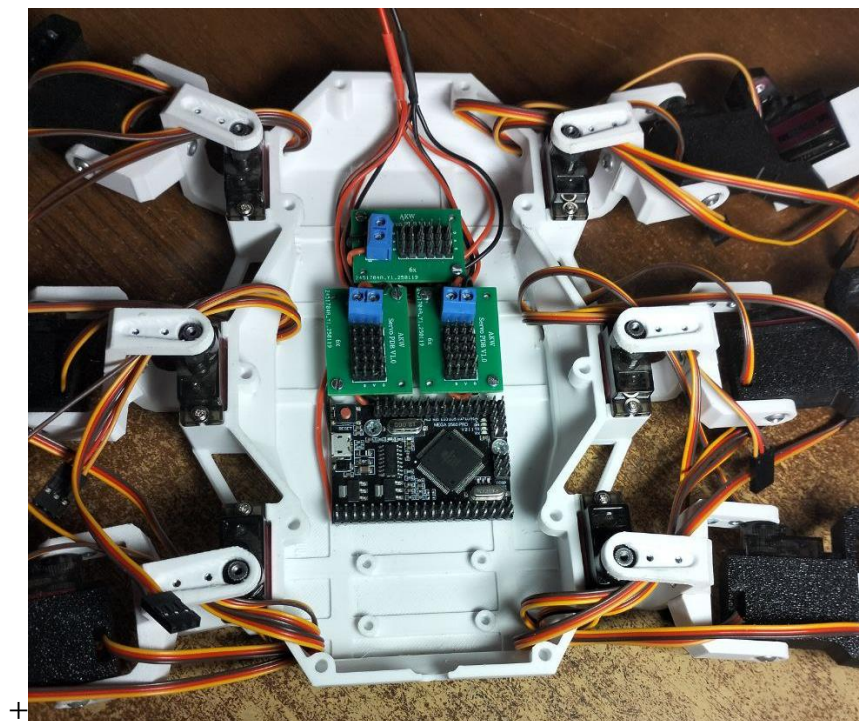


Рисунок 3.4 – Всі шість зібраних кінцівок багатоногого робота

Після фінального монтажу конструкція робота виглядала як на рисунку 3.5. На ньому видно повністю зібраний робот зі встановленими ногами та підключеним живленням.



Рисунок 3.5 – Повністю зібраний багатонорий робот

Фінальним етапом стало тестування рухів робота у реальних умовах. Після отримання команди через Bluetooth, кожна з ніг виконувала задану траєкторію. На рисунку 3.6 наведено зображення робота в процесі руху.

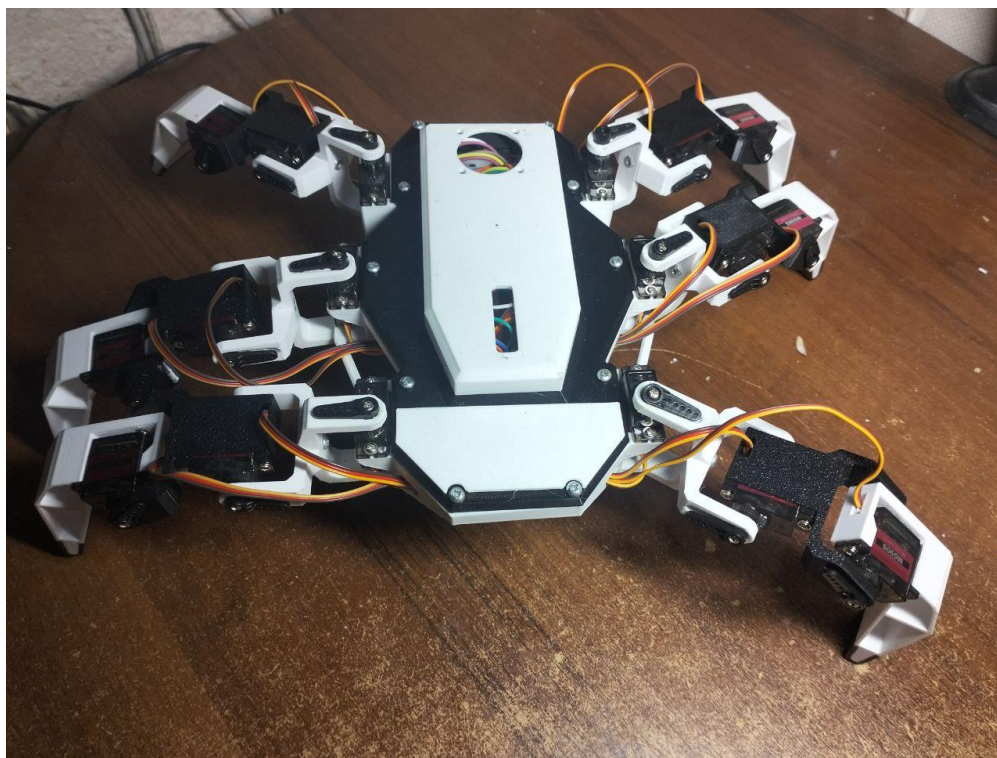


Рисунок 3.6 – Робот у процесі виконання циклу руху

Таким чином, практична реалізація охоплює повний цикл – від початкового модулювання до тестування функціонального зразка, що підтверджує працездатність усіх розроблених апаратних і програмних рішень.

### **3.2 Тестування та налагодження програмно-апаратного комплексу**

Після завершення складання всіх механічних компонентів, підключення електроніки та написання базової версії програмного забезпечення було проведено поетапне тестування та налагодження системи. Основна мета цього етапу – перевірити відповідність роботи апаратної частини розробленим алгоритмам керування та забезпечити стабільність і надійність функціонування всієї системи.

Процес тестування включав такі основні етапи:

- перевірка працездатності кожного серводвигуна окремо;
- тестування реакції сервомеханізмів на команду з програмного забезпечення;
- оцінка роботи окремої ноги в динаміці: підйом, перенесення, опускання;
- послідовне керування всіма шістьма ногами в режимі фіксованого положення;
- виконання циклів базового алгоритму ходи (3+3 чергування кінцівок);
- надсилання команд руху з мобільного пристрою через Bluetooth;
- перевірка адаптивності рухів на різних типах поверхонь (рівна, ковзаюча, м'яка).

На рисунку 3.7 подано приклад запуску тестового режиму керування ногами в ручному режимі з використанням Bluetooth-зв'язку.

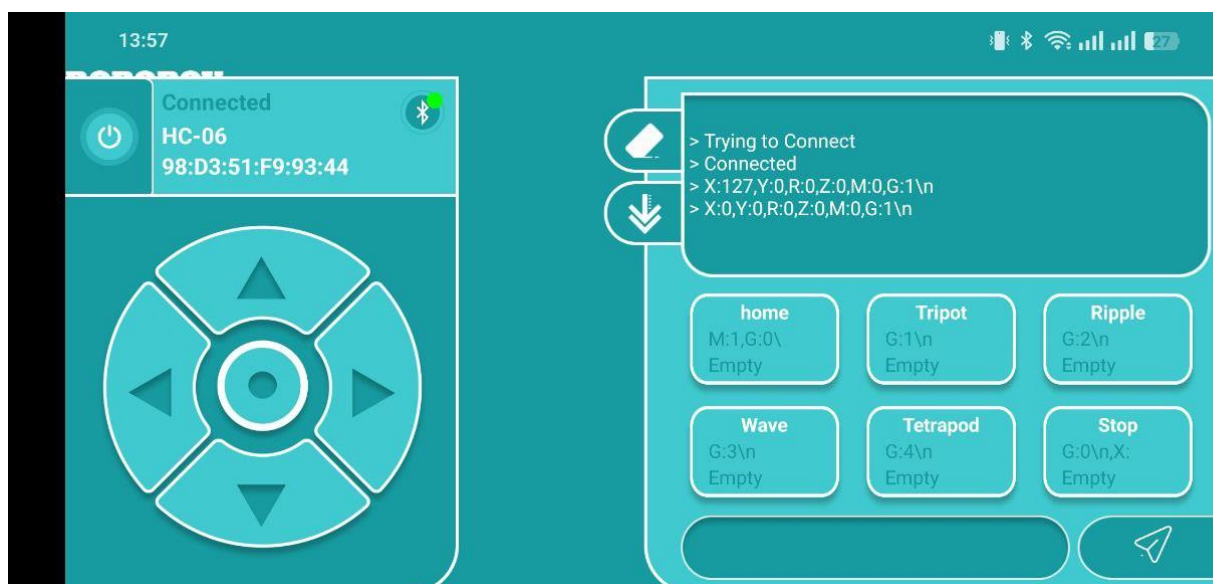


Рисунок 3.7 – Тестування системи через Bluetooth-з'єднання

Особливу увагу було приділено оцінці затримки між надсиланням команди та виконанням руху. Система показала середню затримку менше ніж 150 мс при роботі з 18 сервомеханізмами, що вважається прийнятним результатом для малопотужних вбудованих систем.

Також проводилось налагодження зворотної кінематики. У процесі тестування було зафіксовано незначні відхилення при обчисленні траєкторії для крайніх положень ніг. Після уточнення обчислень (врахування меж обертання сервомеханізмів та граничних значень довжин сегментів) помилки було усунуто.

На завершальному етапі була оцінена узгоджена робота ніг у циклі ходьби. Робот демонстрував плавні та стабільні рухи, зберігаючи центр ваги в межах опорного трикутника. Загальний вигляд робота під час виконання алгоритму руху подано на рисунку 3.8.



Рисунок 3.8 – Узгоджений рух усіх ніг у циклічному режимі ходьби

Окремим етапом у процесі налагодження стала калібровка системи, яка виконувалася як у ручному, так і у програмному режимі.

Ручна калібровка включала виставлення фізичного нульового положення для кожного серводвигуна. Кожна нога встановлювалася у початкове положення механічно за допомогою регулювання важелів і фіксування положення під час подачі команди `servo.write(90)`. Таким чином забезпечувалося, що при стандартному сигналі кожна кінцівка перебуває у нейтральному стані.

Програмна калібровка реалізовувалася через створення тестових функцій, які дозволяли змінювати положення кожного двигуна індивідуально, та вводити корекційні значення (офсети) для усунення дрібних механічних похибок. Ці значення були внесені у програмну частину системи та застосовувалися під час обчислення кутів повороту.

Таким чином, комбіноване калібрування дозволило значно покращити точність виконання рухів та симетрію роботи кінцівок у динаміці.

Таким чином, проведені тестування та налагодження підтвердили працездатність розробленого програмно-апаратного комплексу, його відповідність технічним характеристикам і готовність до подальшого вдосконалення.

### **3.3 Розробка бази даних для програмно-апаратного комплексу**

Одним із напрямків подальшого вдосконалення системи є створення бази даних, що дозволить зберігати типові шаблони рухів, позиції кінцівок у просторі, команди користувача, а також результати попередніх сеансів керування. Збереження такої інформації дозволяє оптимізувати повторні дії, реалізовувати сценарії автоматичних завдань, а також аналізувати виконані маніпуляції.

Базу даних розроблено у вигляді локальної структури на мові SQL, яку в майбутньому можливо реалізувати як частину серверного модуля або використовувати на вбудованому мікрокомп'ютері типу ESP32 чи Raspberry Pi при розширенні комплексу.

Основні таблиці бази даних:

- `commands` – зберігає унікальні команди руху;
- `servo_movements` – фіксує кут повороту кожного сервомеханізму;
- `coordinates` – містить просторові координати X, Y, Z цільових точок для ніг;
- `sessions` – реєструє серії рухів з часовими мітками.

У лістингу 3.2 наведено приклад створення таблиці команд у базі даних.

### Лістинг 3.2 – Створення таблиці

---

```
CREATE TABLE commands
(
    id INT AUTO_INCREMENT PRIMARY KEY, // унікальний ідентифікатор
    запису
    command_name VARCHAR(50) NOT NULL, // назва команди (до 50
    символів)
    command_type ENUM('move', 'grasp', 'release') NOT NULL, // тип
    команди: рух, захват або відпускання
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP // час створення
    запису за замовчуванням
);
```

---

кінець лістингу 3.2

Для взаємодії з базою даних передбачається використання веб-інтерфейсу або API-запитів (ліст. 3.3). Це дозволить у перспективі зчитувати та записувати команди до бази безпосередньо з мобільного або десктопного застосунку.

### Лістинг 3.3 – API-запит на отримання команди з бази даних

---

```
CREATE TABLE servo_movements // оголошення таблиці
(
    id INT AUTO_INCREMENT PRIMARY KEY, // унікальний ID запису
    command_id INT, // зовнішній ключ на commands
    servo_id INT, // зовнішній ключ на servos
    start_angle INT NOT NULL, // початковий кут
    end_angle INT NOT NULL, // кінцевий кут
    duration INT NOT NULL, // тривалість руху
    FOREIGN KEY (command_id) REFERENCES commands(id), // зв'язок із
    commands
    FOREIGN KEY (servo_id) REFERENCES servos(id) // зв'язок із
    servos
);
```

---

кінець лістингу 3.3

Таким чином, створення бази даних надає системі додатковий рівень функціональності, підвищує гнучкість та дозволяє зберігати та повторно використовувати наперед запрограмовані або записані сценарії рухів.

### 3.4 Захист інформаційно-комп'ютерної системи

Забезпечення захисту інформаційно-комп'ютерної системи є одним з обов'язкових аспектів при розробці сучасних роботизованих платформ. Незважаючи на відносну простоту прототипу, доцільним є впровадження базових заходів захисту для забезпечення цілісності, доступності та конфіденційності даних, що передаються в системі.

Першим і базовим рівнем захисту є фізичний контроль доступу до мікроконтролера та його елементів. Усі електронні компоненти розміщено всередині корпусу, що обмежує можливість несанкціонованого втручання у роботу пристрою.

Другим аспектом є захист бездротового з'єднання Bluetooth. У модулі HC-05 реалізовано автентифікацію за PIN-кодом, що унеможливорює підключення неавторизованого користувача. Для підвищення рівня безпеки передбачено зміну стандартного PIN-коду на індивідуальний під час конфігурації через AT-команди.

Також в Arduino-скетчі реалізовано фільтрацію команд, що надходять з Bluetooth. Команди перевіряються за шаблоном і тільки після цього виконуються. Це дозволяє запобігти надсиланню шкідливих або невалідних інструкцій, які можуть викликати некоректну поведінку системи.

У разі майбутнього переходу до використання Wi-Fi-зв'язку або мобільного додатка із віддаленим доступом, доцільним стане впровадження додаткових методів захисту, таких як шифрування даних (наприклад, через TLS/SSL), автентифікація користувачів та контроль сесій доступу.

Таким чином, навіть у компактній мобільній платформі впровадження простих, але дієвих заходів безпеки дозволяє забезпечити захищене функціонування пристрою та унеможливити доступ сторонніх осіб до керування системою.

### **3.5 Перспективи подальшого розвитку системи керування кінематикою багатоногого робота**

Подальший розвиток проєкту можна забезпечити шляхом реалізації низки удосконалень, що базуються на сучасних технологіях робототехніки. Ключовими напрямками є інтеграція з ROS, розширення сенсорного оточення, впровадження машинного навчання для оптимізації рухів, перехід на потужніші апаратні платформи, вдосконалення інтерфейсу керування та оптимізація енергоспоживання. Ознайомившись із концепціями ROS та мікро-ROS, прикладами робототехнічних платформ, роботів із глибоким навчанням та сучасними практиками енергоменеджменту, визначено наступні напрямки розвитку.

Підключення проєкту до Robot Operating System (ROS) дозволить зробити систему більш модульною та відкрити можливості для симуляції й використання розвинених інструментів. Зокрема, за допомогою micro-ROS можна перетворити Arduino-модуль на вузол ROS2, що обмінюється стандартними повідомленнями. Це дає доступ до готових драйверів для різних датчиків та контролерів, а також до симулятора Gazebo, останній є вільним 3D-симулятором, який інтегрує фізичний рушій ODE та дозволяє змодельовати датчики (лазерні далекоміри, камери тощо). Використання таких інструментів значно спрощує тестування алгоритмів і пришвидшує налагодження системи.

Додаткові датчики підвищують адаптивність робота до навколишнього середовища. Зокрема, інтеграція 2D LIDAR (лазерного далекоміра) дозволяє будувати карту місцевості і уникати перешкод. Так, комерційні шестиногі роботи (наприклад, JetHexa) оснащені LIDAR з алгоритмами картографування (Cartographer, Hector, Gmapping), що дає можливість планувати маршрут та уникати перешкод. Додавання камер (кольорових або глибинних) відкриває можливості для візуальної навігації та розпізнавання об'єктів. Інерціальний датчик (3-осьовий IMU) забезпечує інформацію про орієнтацію й прискорення; наприклад, дані з MPU6050 через Arduino можна передавати до ROS,

використовуючи `rosserial`, що допомагає стабілізувати рух і визначати положення робота.

Сучасні алгоритми штучного інтелекту можна застосувати для оптимального підбору параметрів ходьби та реакцій на зовнішні фактори. Зокрема, глибоке підсилене навчання (Deep Reinforcement Learning) вже використовується для локомоції, контролер для квадропеда Solo12, натренований за допомогою DRL, навчився слідувати заданій швидкості руху при цьому переміщуючись енергетично ефективно. Навіть на мікроконтролерах можна запускати спрощені моделі. Так, в дослідженні продемонстровано, що робот на базі Arduino UNO R4 із LIDAR може використовувати навчений класифікатор (вбудований у прошивку) для визначення напрямку руху і успішно обходити перешкоди.

Перехід з Arduino Nano чи Uno на сучасні контролери чи одноплатні комп'ютери значно розширить можливості системи. Наприклад, ESP32 (240 МГц, два ядра, вбудований Wi-Fi/Bluetooth) підтримує micro-ROS і може публікувати ROS2-повідомлення по бездротовому інтерфейсу. Одноплатний комп'ютер Raspberry Pi (процесор до 1,6 ГГц, оперативна пам'ять до 8 ГБ) забезпечує середовище Linux для виконання ROS/Gazebo і обробки даних (наприклад, відеопотоків). Для порівняння Arduino Uno використовує 8-бітний мікроконтролер (~16 МГц) з 2 КБ оперативної пам'яті, тоді як Raspberry Pi має багатоядерний процесор і гігабайти ОЗП. Такий приріст обчислювальних ресурсів дозволяє виконувати на борту значно складніші завдання.

Розробка веб-інтерфейсу чи мобільного застосунку значно спростить керування роботом. Існують готові рішення, зокрема Robot Web Tools – набір бібліотек (`rosbridge`, `roslibjs`, `ros2djs`, `ros3djs`) для створення веб-панелей, які в реальному часі обмінюються ROS-повідомленнями. Аналогічно, можна створити Android/iOS-додаток, що по Wi-Fi або Bluetooth відправляє команди контролеру (наприклад, через REST API або протокол MQTT). Це дозволить оператору керувати роботом дистанційно без потреби дротового підключення.

Для подовження часу роботи важливо впровадити енергозберігаючі режими. Зокрема, сучасні системи можуть автоматично перемикатися між станами, що дозволяє під час простою переходити у режим з мінімальним споживанням. Використання систем енергоменеджменту (схожих на тих, що застосовують у промислових роботах) дає змогу відстежувати витрати енергії та адаптувати режим роботи приводів і швидкість руху для економії. Крім того, оптимізація програмної логіки та траєкторій (наприклад, мінімізація зайвих переміщень) може суттєво знизити загальне енергоспоживання системи.

Загалом реалізація зазначених напрямів відповідає сучасним практикам робототехніки і може бути здійснена з використанням відкритих платформ та бібліотек (ROS, TensorFlow Lite, OpenCV тощо). Ці удосконалення зроблять систему управління багатонігим роботом більш універсальною, адаптивною та ефективною в реальному середовищі.

## ВИСНОВКИ

У процесі виконання кваліфікаційної роботи бакалавра було створено програмно-апаратний комплекс для керування кінематикою багатоногого робота на платформі Arduino. Розроблено механічну конструкцію, інтегровано електронні компоненти, створено програмне забезпечення та проведено комплексні випробування системи. Реалізовано базу даних для зберігання команд і рухів, впроваджено заходи інформаційної безпеки, зокрема автентифікацію через Bluetooth. В результаті тестування підтверджено високу точність, стабільність та ефективність функціонування комплексу.

У ході аналізу сучасного стану розробок у галузі багатоногих роботів було визначено актуальні проблеми, такі як точність керування рухами, балансування на нерівних поверхнях та ефективна інтеграція з навколишнім середовищем. Досліджено технології зворотної кінематики, алгоритми динамічного балансування, розглянуто практичні приклади реалізації Нехарод-платформ для військового та пошуково-рятувального використання, що підтвердило актуальність і перспективність цього напрямку розвитку робототехніки.

Під час визначення функціональних і технічних вимог до системи управління було сформульовано детальні критерії, які включають точне позиціонування кінцівок робота, підтримку різних режимів руху, можливість швидкого реагування на команди користувача, а також адаптацію до змін середовища. Проектування сценаріїв використання та послідовностей команд було реалізовано через UML-діаграми, що забезпечило чітку структуру взаємодії компонентів системи.

Здійснено вибір апаратних засобів, включаючи мікроконтролер Arduino Mega 2560 PRO, сервоприводи SG90 та Bluetooth-модуль HC-05, які забезпечують необхідну точність і стабільність рухів робота. Розроблено алгоритми керування, оптимізовані для швидкої реакції, плавності та точності рухів, з використанням зворотної кінематики. Вибрані методи дозволили досягти високих показників продуктивності та адаптивності.

Практична реалізація включала збирання механічної конструкції робота, інтеграцію всіх електронних компонентів та написання програмного коду для керування сервоприводами. Виконані тести засвідчили, що механічна конструкція ефективно взаємодіє з програмним забезпеченням, що дозволяє роботу стабільно виконувати заплановані рухи та адекватно реагувати на зовнішні команди користувача.

Проведено детальне тестування та налагодження системи, в процесі яких було оцінено точність роботи сервоприводів, реакцію на команди через Bluetooth, стабільність та плавність руху робота. В результаті ручної та програмної калібровки вдалося забезпечити високий рівень точності та стабільності платформи, що відповідає початковим вимогам до системи.

Розроблена база даних дозволяє ефективно зберігати шаблони рухів, просторові координати та параметри команд, що суттєво спрощує управління роботом.

Запровадження базових заходів захисту інформаційно-комп'ютерної системи, зокрема контроль доступу, фільтрація команд та автентифікація Bluetooth-підключень, забезпечує безпеку роботи та захист від несанкціонованого втручання.

Запропоновано перспективні напрями розвитку, серед яких інтеграція з Robot Operating System (ROS), розширення функціоналу за допомогою сенсорних систем (LIDAR, IMU), впровадження алгоритмів машинного навчання для підвищення ефективності рухів, перехід на більш потужні платформи та вдосконалення інтерфейсів користувача. Це дозволить значно розширити потенціал застосування розробленого комплексу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Lumia R. Collision Detection for Hexapod Walking Robots. Robotics and Autonomous Systems. URL: <https://doi.org/10.1016/j.robot.2022.104034> (date of access: 04.01.2025).
2. Renewable Electric Energy System Planning Considering Seasonal Electricity Imbalance Risk. IEEE Xplore. URL: <https://ieeexplore.ieee.org/document/9987654> (date of access: 08.01.2025).
3. Thingiverse. Hexapod Robot. Thingiverse. URL: <https://www.thingiverse.com/thing:4576790> (date of access: 20.01.2025).
4. Лозинський О. М. Багатоногі роботи: огляд конструкцій та методів керування. Вісник НТУ «ХПІ». URL: <https://repository.kpi.kharkov.ua/handle/KhPI-Press/58924> (дата звернення: 23.01.2025).
5. Bambanek J. Efficient Gait Generation for Low-Cost Hexapods. Journal of Field Robotics. URL: <https://onlinelibrary.wiley.com/doi/10.1002/rob.21934> (date of access: 25.01.2025).
6. ATMEGA2560 Datasheet(PDF). ALLDATASHEET.COM — Electronic Parts Datasheet Search. URL: <https://www.alldatasheet.com/datasheet-pdf/pdf/107092/ATMEL/ATMEGA2560.html> (date of access: 01.02.2025).
7. The Influence of Visible Cables and Story Content on Perceived Autonomy in Social Human–Robot Interaction. MDPI. URL: <https://www.mdpi.com/2218-6581/12/1/3> (date of access: 09.02.2025).
8. Гапонюк В. І. Arduino Mega 2560 як платформа для керування сервоприводами у навчальних роботах. Системи управління, навігації та зв'язку. URL: <http://sunc.nangu.edu.ua/article/view/214213> (дата звернення: 12.02.2025).
9. Arduino. Servo Library Reference. Arduino. URL: <https://www.arduino.cc/reference/en/libraries/servo/> (date of access: 15.02.2025).
10. Tower Pro. SG90 Micro Servo. URL: <https://towerpro.com.tw/product/sg90-9g-micro-servo> (date of access: 17.02.2025).

11. Arduino. Mega 2560 Rev3. URL: <https://docs.arduino.cc/hardware/mega-2560> (date of access: 20.02.2025).
12. Поплавський Д. Б. Використання модулів HC-05 для бездротового керування робототехнічними пристроями. Східноєвропейський журнал передових технологій. URL: <https://journals.uran.ua/eejet/article/view/270197> (дата звернення: 23.02.2025).
13. SparkFun. HC-05 Bluetooth module: Datasheet and AT Commands. SparkFun. URL: <https://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/HC-05.pdf> (date of access: 28.03.2025).
14. Instructables. Hexapod Robot. Instructables. URL: <https://www.instructables.com/Hexapod-Robot> (date of access: 01.04.2025).
15. C++ Programming Language: Power, Versatility, and Considerations — Dev community hub. Dev community hub. URL: <https://devcommunityhub.com/languages/c/c-programming-language-power-versatility-and-considerations/> (date of access: 02.04.2025).
16. Software. Arduino — Home. URL: <https://www.arduino.cc/en/software/> (date of access: 02.04.2025).