

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

8.01.2024



Манічкін М.О.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ АКТ
Кафедра _____ КІТАР
Рівень вищої освіти _____ другий (магістерський)
Спеціальність _____ 151 Автоматизація та комп'ютерно-інтегровані технології
Тип програми _____ Освітньо-професійна
Освітня програма _____ Комп'ютеризовані та робототехнічні системи
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав.кафедри _____
(підпис)

«__» _____ 2024р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студентові _____ Манічкіну Максиму Олексійовичу
(шифр і назва)

1. Тема роботи: _____ Розроблення системи керування мобільним роботом
зооморфного типу

Затверджена наказом університету від _____ №1288Ст від 03.11.2023

2. Термін подання студентом роботи до екзаменаційної комісії _____ 11.01.2024р.

3. Вихідні дані до роботи: 3.1 Утримання рівноваги на базі акселерометру
GY-521; 3.2 Друк макету за допомогою PLA пластику; 3.3 Дворівнева
система керування

4. Перелік питань, що потрібно опрацювати в роботі: 4.1 Вступ; 4.2 Аналіз
конструкцій зооморфних мобільних роботів; 4.3 Аналіз методів керування
мобільними роботами; 4.4 Аналіз особливостей керування зооморфними
мобільними роботами; 4.5 Розробка моделі кінематики зооморфного робота
типу Spot; 4.6 Обґрунтування та вибір виконавчих механізмів; 4.7 Розробка
конструкції зооморфного робота; 4.8 Розробка 3D моделі деталей
конструкції мобільного робота; 4.9 Розробка 3D моделі зборки мобільного
робота типу Spot; 4.10 Зборка макета зооморфного робота типу Spot;
4.11 Аналіз та вибір апаратних модулів для реалізації системи керування;
4.12 Розробка структурної схеми системи керування; 4.13 Розробка схеми
підключення апаратних модулів; 4.14 Розробка алгоритму керування
зооморфними мобільними роботами; 4.15 Реалізація функцій керування
серводвигунами; 4.16 Постановка задач експериментів; 4.17 Аналіз
отриманих результатів; 4.18 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Графічний демонстраційний матеріал в форматі PowerPoint(*.ppt) формату А4 –13 сторінок.

6. Консультанти розділів роботи

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз сучасних систем керування мобільними роботами	2.09.2023-18.09.2023	виконано
2	Розробка моделі кінематики зооморфного робота типу Spot	19.09.2023-25.09.2023	виконано
3	Обґрунтування та вибір виконавчих механізмів	26.09.2023-2.10.2023	виконано
4	Розробка конструкції зооморфного робота	3.10.2023-21.10.2023	виконано
5	Розробка 3D моделі зборки мобільного робота типу Spot	22.10.2023-26.10.2023	виконано
6	Зборка макета зооморфного робота	27.10.2023-29.10.2023	виконано
7	Аналіз та вибір апаратних модулів	30.10.2023-5.11.2023	виконано
8	Розробка структурної схеми системи керування	6.11.2023-11.11.2023	виконано
9	Розробка схеми підключення апаратних модулів	12.11.2023-18.11.2023	виконано
10	Розробка алгоритму керування	19.11.2023-26.11.2023	виконано
11	Реалізація функцій керування серводвигунами	27.11.2023-9.12.2023	виконано
12	Експериментальні дослідження	10.12.2023-26.12.2023	виконано

Дата видачі завдання 1 вересня 2023р.

Студент _____ Манічкін М.О.
(підпис) (прізвище, ініціали)

Керівник роботи _____ Євсєєв В.В.
(підпис) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 100 с., 13 табл., 36 рис., 1 дод., 21 джерело.

РОБОТ SPOT, ЗООМОРФНИЙ РОБОТ, МОБІЛЬНИЙ РОБОТ, СИСТЕМА КЕРУВАННЯ ЗООМОРФНИМ РОБОТОМ, РОБОТ-СОБАКА, 3D МОДЕЛЬ ЗООМОРФНОГО РОБОТА, АЛГОРИТМ КЕРУВАННЯ РОБОТОМ SPOT.

Мета роботи – розроблення системи керування та експериментального макету зооморфного мобільного робота.

Об’єкт розробки – процес керування мобільним роботом.

Предмет розробки – методи керування зооморфним роботом.

В даній кваліфікаційній роботі проаналізовано конструкції зооморфних мобільних роботів, методи керування мобільними роботами та особливості керування зооморфними мобільними роботами. Розроблено модель кінематики зооморфного робота типу Spot, обґрунтовано та вибрано виконавчі механізми. Розроблено конструкцію зооморфного робота, 3D моделі деталей конструкції мобільного робота, 3D моделі зборки мобільного робота типу Spot, 3D моделі макета. Проведено збірку макета зооморфного робота типу Spot, проаналізовано та вибрано апаратні модулі для реалізації системи керування. Розроблено структурну схему системи керування, схему підключення апаратних модулів та алгоритм керування зооморфними мобільними роботами. Реалізовано функції керування серводвигунами, проведено експериментальні дослідження та проаналізовано отримані результати.

ABSTRACT

Explanatory note: 100 pages, 13 tables, 36 figures, 1 app, 21 sources.

SPOT ROBOT, ZOOMORPHIC ROBOT, MOBILE ROBOT, ZOOMORPHIC ROBOT CONTROL SYSTEM, DOG ROBOT, 3D ZOOMORPHIC ROBOT MODEL, SPOT ROBOT CONTROL ALGORITHM.

The purpose of the work is to develop a control system and an experimental model of a zoomorphic mobile robot.

The object of development is the process of controlling a mobile robot.

The subject of development is methods of controlling a zoomorphic robot.

This qualification paper analyzes the designs of zoomorphic mobile robots, methods of controlling mobile robots and features of controlling zoomorphic mobile robots. A model of the kinematics of the Spot zoomorphic robot was developed, and the executive mechanisms were justified and selected. The construction of a zoomorphic robot, 3D models of mobile robot construction details, 3D models of the Spot type mobile robot assembly, 3D layout models have been developed. The mock-up of the Spot zoomorphic robot was assembled, the hardware modules for the implementation of the control system were analyzed and selected. The structural diagram of the control system, the connection diagram of hardware modules and the control algorithm of zoomorphic mobile robots have been developed. Servo motor control functions were implemented, experimental studies were conducted and the obtained results were analyzed.

ЗМІСТ

Перелік умовних скорочень	9
Вступ.....	10
1 Аналіз сучасних систем керування мобільних роботів.....	12
1.1 Аналіз конструкцій зооморфних мобільних роботів	12
1.2 Аналіз методів керування мобільним роботами	14
1.3 Аналіз особливостей керування зооморфними мобільним роботами... ..	16
1.4 Постановка задач дослідження	18
2 Розробка макета мобільного зооморфного робота типу SPOT	19
2.1 Розробка моделі кінематики зооморфного робота типу Spot.....	19
2.2 Обґрунтування та вибір виконавчих механізмів (серводвигунів)	26
2.3 Розробка конструкції зооморфного робота	28
2.4 Розробка 3D моделі деталей конструкції мобільного робота	30
2.5 Розробка 3D моделі зборки мобільного робота типу Spot.....	36
2.6. Зборка макета зооморфного робота типу Spot.....	37
2.7 Висновки до 2 розділу	40
3 Розробка системи керування мобільним роботом	42
3.1 Аналіз та вибір апаратних модулів для реалізації системи керування .	42
3.2 Розробка структурної схеми системи керування.....	49
3.3. Розробка схеми підключення апаратних модулів	51
3.4 Розробка алгоритму керування зооморфними мобільними роботами ..	55
3.5 Реалізація функцій керування серводвигунами.....	60
3.6 Висновки до 3 розділу	65
4 Експериментальні дослідження та аналіз отриманих результатів.....	67
4.1 Постановка задач експериментів.....	67
4.2 Аналіз отриманих результатів	80
4.3. Розрахунок освітлення.....	86
4.4 Висновки до 4 розділу	87

Висновки	89
Перелік джерел посилань	90
Додаток А Апробація результатів наукових досліджень.....	93
Додаток Б Демонстраційний матеріал	99

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

FDM – fused deposition modeling;

Flash – пам'ять програм;

PLA – полілактид;

ROS – robot operating system;

SRAM – оперативна пам'ять.

ВСТУП

У сучасному світі роботизовані системи стають не лише покращенням технологій, але й важливим інструментом для вирішення різноманітних завдань у різних галузях. Однією з передових інновацій в області робототехніки є розробка зооморфних роботів, які імітують рухи та поведінку тварин для досягнення високої ефективності та адаптивності. У цьому контексті, розробка системи керування для зооморфного робота SPOT є вельми актуальною та перспективною.

SPOT може бути використаний в промисловості для автоматизації та оптимізації процесів. Здатність ефективно керувати цим роботом відкриває нові можливості для використання виробничого потенціалу та підвищення продуктивності в різних сферах. SPOT може бути ефективним інструментом у рятувальних операціях у небезпечних умовах або природних катастрофах. Керування його рухами та функціями може врятувати життя та допомогти збирати важливі дані під час медичних досліджень. SPOT має потенціал для військового використання, наприклад, для розвідувальних місій або перенесення обладнання на неприступних теренах. Система керування грає ключову роль у визначенні та координації таких завдань. Розробка системи керування для SPOT сприяє розвитку нових методів навчання та досліджень у галузі зооморфних роботів, відкриваючи можливості для розуміння природних механізмів та їхнього впливу на робототехніку.

Мета роботи – розроблення системи керування та експериментального макету зооморфного мобільного робота.

Об'єкт розробки – процес керування мобільним роботом.

Предмет розробки – методи керування зооморфним роботом.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати конструкції зооморфних мобільних роботів;

- проаналізувати методи керування мобільним роботами;
- проаналізувати особливості керування зооморфними мобільним роботами;
- розробити модель кінематики зооморфного робота типу Spot;
- обґрунтувати та вибрати виконавчі механізми;
- розробити конструкцію зооморфного робота;
- розробити 3D моделі деталей конструкції мобільного робота;
- розробити 3D моделі зборки мобільного робота типу Spot;
- розробити 3D моделі макета;
- провести збірку макета зооморфного робота типу Spot;
- проаналізувати та вибрати апаратні модулі для реалізації системи керування;
- розробити структурну схему системи керування;
- розробити схему підключення апаратних модулів;
- розробити алгоритм керування зооморфними мобільним роботами;
- реалізувати функції керування серводвигунами;
- провести постановку задач експериментів;
- проаналізувати отримані результати.

Дана кваліфікаційна робота виконана згідно ДСТУ 3008 – 15 [1], керуючись навчальним посібником з дипломного проекту [2] та методичними вказівками [3], а результати кваліфікаційної роботи описані в статті, можна знайти в гугл академії [4].

1 АНАЛІЗ СУЧАСНИХ СИСТЕМ КЕРУВАННЯ МОБІЛЬНИХ РОБОТІВ

1.1 Аналіз конструкцій зооморфних мобільних роботів

На сьогоднішній день існує значна кількість розробок у галузі зооморфних мобільних роботів. Деякі з найрозповсюдженіших конструкцій використовуються в різних дослідницьких проектах та промислових застосуваннях [5].

Роботи-орнітоптери, модель орнітоптерів використовує крила для генерації тяги, подібно до того, як це відбувається у птахів. Ці роботи можуть бути використані для аеріальних місій, таких як вивчення та моніторинг великих територій. Роботи на основі квадрупедальних конструкцій намагаються імітувати рухи чотирикінцевих тварин, зокрема теріозаврів. Такі роботи можуть бути використані для досліджень у труднодоступних або небезпечних умовах. Роботи, які моделюють рухи риб, можуть використовуватися для дослідження водних середовищ, моніторингу морських екосистем та здійснення операцій в обмеженому просторі. Інсектороботи, роботи намагаються імітувати рухи та механізми комах. Вони можуть бути використані для ефективного переміщення в складних умовах, таких як руїни або густий ліс. Роботи, імітуючі гадюк, володіють гнучкістю та здатністю пристосовуватися до умов обмеженого простору. Вони можуть бути використані для досліджень у труднодоступних місцях або для виконання операцій у вузьких просторах. Ці конструкції відкривають нові можливості в багатьох областях, включаючи науку, дослідження та промисловість. Проте, розробка зооморфних роботів постійно триває, і разом з тим появляються нові конструкції, що розширюють можливості цих технологій [6].

Для розробки та створення зооморфних мобільних роботів необхідно притримуватись вимог, представлених у таблиці 1.1.

Таблиця 1.1 – Вимоги для розробки зооморфних мобільних роботів

Вимога	Опис
Біомімікрія	Робот повинен імітувати природні рухи та механізми тварин, щоб досягти максимальної ефективності та адаптивності в різних умовах
Маневреність та адаптивність	Конструкція повинна дозволяти роботів легко пристосовуватися до змін у навколишньому середовищі та подоланню різних перешкод
Енергоефективність	Розробка повинна враховувати принципи енергозбереження, що можуть бути взяті з природи, щоб забезпечити тривалу роботу без необхідності частого заряджання
Гнучкість та герметичність	Якщо робот призначений для використання у водних середовищах, конструкція повинна бути герметичною та гнучкою для адаптації до рухів води
Технологічна оснастка	Робот повинен мати необхідне обладнання (камери, датчики, актуатори), щоб виконувати конкретні завдання, забезпечуючи точність та надійність роботи
Керованість та дистанційне управління	Робот повинен бути здатний керуватися в реальному часі та мати можливість дистанційного управління для оптимального контролю та нагляду
Забезпечення безпеки	Конструкція повинна передбачати безпеку під час роботи, уникати взаємодії з навколишнім середовищем та іншими користувачами, якщо це потрібно
Масштабованість та модульність	Розробка повинна бути модульною, щоб легко здійснювати апгрейди та заміни окремих компонентів. Конструкція повинна бути масштабованою для різних завдань
Система живлення та запаси енергії	Робот повинен мати ефективну систему живлення, а також, при необхідності, можливість зберігання енергії для подовження тривалості його роботи
Сенсорна здатність та обробка даних	Конструкція повинна бути оснащена сенсорами для збору інформації з навколишнього середовища, а також потужними системами обробки даних для аналізу та прийняття рішень
Стійкість до умов сприятливого та несприятливого середовища	Робот повинен бути стійким до впливу різних погодних умов, температурних змін, вологості та інших факторів, що можуть вплинути на його ефективність

1.2 Аналіз методів керування мобільним роботами

Існує багато методів керування мобільними роботами, і вибір конкретного методу може залежати від типу робота, його завдань, характеристик навколишнього середовища та вимог до точності. Вибір методу керування мобільним роботом є критичним етапом в розробці робототехнічних систем, оскільки цей вибір визначає ефективність та успішність виконання завдань, які передбачені для робота. Узгоджений вибір методу керування дозволяє досягати максимальної ефективності та пристосування до викликів, з якими може стикатися мобільний робот у своїй діяльності. У таблиці 1.2 представлено порівняльну характеристику основних методів керування мобільними роботами, представлено їх недоліки та переваги [7].

Таблиця 1.2 – Основні методи керування мобільними роботами

Метод	Опис	Переваги	Недоліки
1	2	3	4
Пропорційно-інтегрально-диференціальне (PID) керування	Класичний метод, який використовує пропорційну, інтегральну та диференціальну складові для підтримки стійкої траєкторії та керування рухом	Простота реалізації, ефективний для багатьох завдань	Може вимагати тюнінгу параметрів, не завжди ефективний в умовах зі значними змінами
Лінійне керування	Використовує лінійні алгоритми та математичні моделі для планування та керування рухом робота	Добре підходить для простих завдань та локального планування	Може бути неефективним для складних завдань та нелінійних систем

Продовження таблиці 1.2

1	2	3	4
Метод поля потенціалів	Моделює навколишнє середовище за допомогою потенціалів та намагається мінімізувати сумарний потенціал для досягнення мети	Ефективний для уникнення перешкод та локального планування	Може застрягати в локальних мінімумах, втрачаючи глобальний контекст
Метод генетичних алгоритмів	Використовує принципи еволюції для оптимізації руху та вибору найкращих стратегій	Ефективний для оптимізації та пошуку найкращих рішень	Вимагає значних обчислювальних ресурсів та може бути обмежений реальним часом
Метод нейромережевого керування	Використовує штучні нейромережі для навчання та адаптації до навколишнього середовища, навчається на основі даних та досвіду	Здатний адаптуватися до різних умов та необхідностей	Вимагає великої кількості даних для тренування та може бути складним у реалізації

Вибір оптимального методу керування залежить від конкретного застосування та вимог до роботи. Наприклад, PID-керування може бути ефективним для задач точного керування, тоді як метод генетичних алгоритмів може бути корисним для оптимізації стратегій у різноманітних умовах. Нейромережеве керування може виявитися ефективним у випадках, коли робот повинен навчатися в реальному часі та адаптуватися до змін у середовищі [8].

1.3 Аналіз особливостей керування зооморфними мобільними роботами

Керування зооморфними мобільними роботами має свої унікальні особливості, оскільки вони намагаються імітувати рухи та характеристики природних тварин. Основною ідеєю є використання природних рухів та механізмів тварин для досягнення оптимальної ефективності та адаптивності. Це включає імітацію ходи, політу, плавання або інших характерних рухів тварин. Керування орієнтується на точне моделювання рухів тварин, щоб робот відтворював їхні характеристики. Це може включати гнучкі та адаптивні рухи, які дозволяють роботу ефективно взаємодіяти з навколишнім середовищем. Зооморфні роботи використовують сенсори, які дозволяють їм сприймати навколишнє середовище так, як це робить природа. Наприклад, використання камер для копіювання зору тварин або акцент на сенсорах, що відтворюють ехолокацію, як у кажанів. Роботи повинні бути гнучкими та адаптивними до змін в навколишньому середовищі. Це означає, що вони повинні бути здатними швидко реагувати на перешкоди, зміни у території або інші фактори. Деякі зооморфні роботи використовують принципи енергозбереження, спостережені в природі, щоб максимізувати тривалість роботи без необхідності частого заряджання або пальби. Зооморфні роботи можуть використовувати методи навчання для автономного прийняття рішень, імітуючи інтуїцію тварин та їхніх природних реакцій на зміни в середовищі [9].

Щодо вибору найкращих методів керування для зооморфних роботів, це може залежати від конкретного завдання та вимог. Зазвичай комбінація методів, таких як PID-керування для точності та генетичні алгоритми чи нейромережі для оптимізації траєкторії, може бути ефективною. Рішення також може враховувати фізичні обмеження робота та особливості навколишнього середовища.

Керування зооморфним роботом-собакою, який імітує рухи та характеристики справжніх собак, включає в себе деякі особливості, що дозволяють роботу адаптуватися до різних умов та виконувати різноманітні завдання. Робот-собака намагається імітувати ходу, бігу, повороти та інші рухи справжньої собаки. Це включає в себе точне моделювання кінематики та динаміки рухів. Робот оснащений різними сенсорами, такими як камери, гіроскопи, акселерометри та сенсори відстані, для отримання інформації про навколишнє середовище та реагування на зміни. Робот повинен бути гнучким у використанні різних видів рухів в залежності від ситуації. Наприклад, він може використовувати більш стійкі чотирикінцеві рухи для стабільності на рівній поверхні або гнучкіші рухи для подолання нерівностей. Для точного керування рухом робота-собаки важливо мати системи, що реагують на зміни орієнтації та напрямку. Це дозволяє ефективно керувати роботом в різних умовах. Можливість зміни швидкості в залежності від умов, включаючи прискорення та гальмування, є важливою для плавного та безпечного руху. Деякі роботи-собаки можуть використовувати методи навчання для адаптації до змін у середовищі та взаємодії з користувачами для виконання різноманітних завдань. Роботи-собаки можуть використовувати принципи енергозбереження для максимізації тривалості роботи та оптимізації витрат енергії. Деякі роботи-собаки можуть мати функції автономного руху та взаємодії в групах, що робить їх ефективними для різних сценаріїв використання [10].

Роботи-собаки можуть використовуватися в різних сферах, таких як рятувальні операції, військові додатки, допоміжні та побутові завдання. Керування їх рухами та функціями розробляється з урахуванням конкретних вимог та завдань, що стоять перед роботом.

1.4 Постановка задач дослідження

В ході проведеного аналізу було виявлено, що тема даного дослідження є актуальною.

Метою даної роботи є розроблення системи керування та експериментального макету зооморфного мобільного робота. Об'єктом дослідження є процес керування мобільним роботом, предметом дослідження є методи керування зооморфним роботом, методами дослідження є метод 3D моделювання та методу аналізу та вибору апаратних модулів. Для досягнення поставленої мети потрібно вирішити наступні завдання:

- розробити модель кінематики зооморфного робота типу Spot;
- обґрунтувати та вибрати виконавчі механізми;
- розробити конструкцію зооморфного робота;
- розробити 3D моделі деталей конструкції мобільного робота;
- розробити 3D моделі зборки мобільного робота типу Spot;
- розробити 3D моделі макета;
- провести збірку макета зооморфного робота типу Spot;
- проаналізувати та вибрати апаратні модулі для реалізації системи керування;
- розробити структурну схему системи керування;
- розробити схему підключення апаратних модулів;
- розробити алгоритм керування зооморфними мобільними роботами;
- реалізувати функції керування серводвигунами;
- провести постановку задач експериментів;
- проаналізувати отримані результати.

2 РОЗРОБКА МАКЕТА МОБІЛЬНОГО ЗООМОРФНОГО РОБОТА ТИПА SPOT

2.1 Розробка моделі кінематики зооморфного робота типу Spot

Зооморфні роботи типу Spot розробляються з метою імітації рухів тварин у природі. Ці роботи відрізняються від класичних промислових роботів тим, що вони мають більш гнучку конструкцію і можуть виконувати різноманітні рухи, що набагато більше нагадують природні рухи тварин.

Однією з ключових складових зооморфних роботів є його кінематика. Моделі кінематики зооморфного робота типу Spot можуть бути представлені у різних формах, включаючи механічні моделі та математичні моделі. На рисунку 2.1 приведено загальний вид робота Spot від Boston Dynamics.

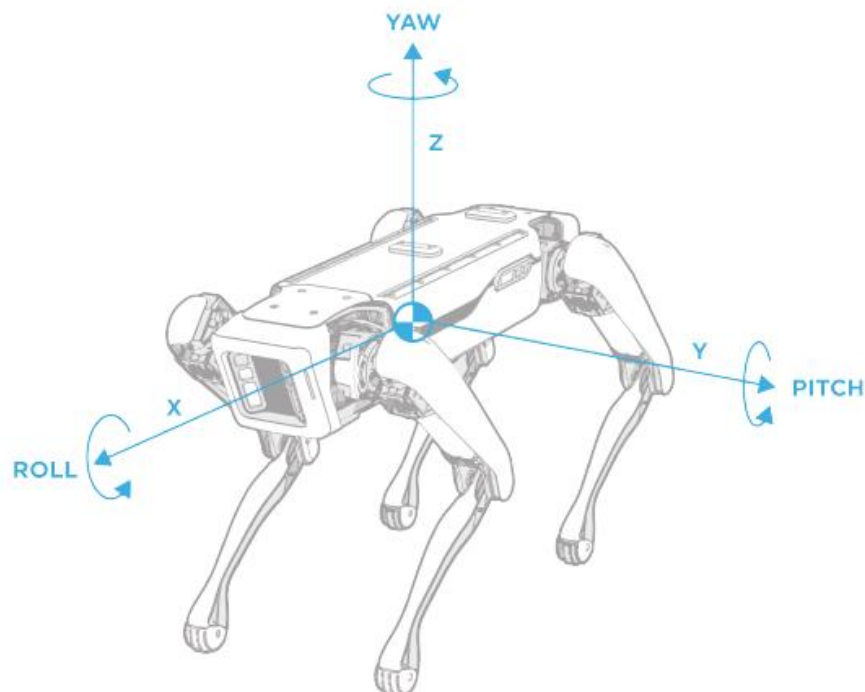


Рисунок 2.1 – Загальний вид робота Spot від Boston Dynamics у тривимірному просторі

Для стійкого керування мобільним роботом потрібно розробити кінематичну схему кінцівки та розрахувати значення положення ланки для кожного ланцюга, що в майбутньому дасть можливість обрати виконавчі механізми (серводвигуни) для реалізації переміщення мобільного робота. Але на першому етапі необхідно провести дослідження та обрати тип кінематичної схеми побудови кінцівок мобільного робота Spot. Виходячи з загального виду мобільного робота Spot, який представлено на рисунку 2.1, побудуємо схематичне представлення загальної конфігурації, кінематичної моделі (рис. 2.2).

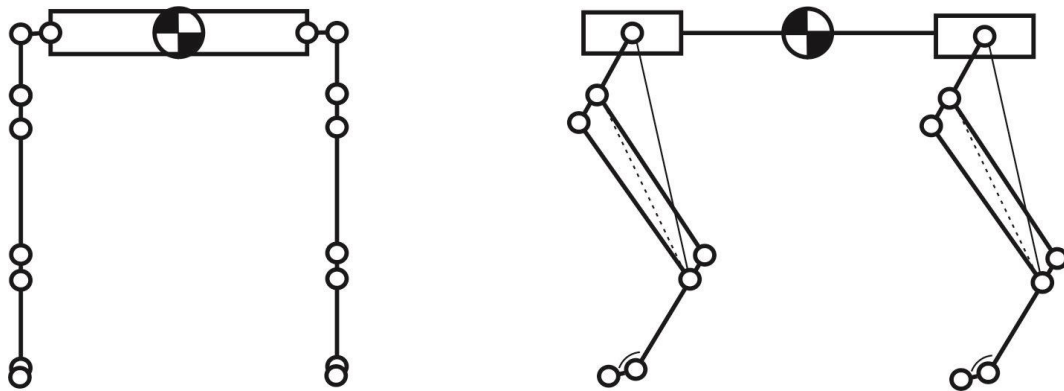


Рисунок 2.2 – Схематичне представлення загальної конфігурації мобільного робота Spot

Оберемо основні критерії, які надалі будуть являтися базами для обґрунтування вибору як матеріалів для розроблюваного макета, так і його масою габаритних параметрів, які представлені в таблиці 2.1.

Таблиця 2.1 – Основні параметри розроблюваного мобільного робота

Параметр	Значення
1	2
Загальні	
Висота (мм)	~290
Довжина (мм)	~380
Ширінка (мм)	~230
Маса без акумуляторів (гр.)	

Продовження таблиці 2.1

1		2
Корпусу		
Висота (мм)		~70
Довжина (мм)		~380
Ширінка (мм)		~110
Кінцівки		
I ланка	Висота (мм)	~150
	Довжина (мм)	~15
	Ширінка (мм)	~min 15, ~max 22
II	Висота (мм)	~140
	Довжина (мм)	~ min 30, ~max 50
	Ширінка (мм)	~ 25

На основі вище вказаних критеріїв та запропонованого схематичного представлення загальної конфігурації мобільного робота Spot (рис. 2.2), розробимо схематичне представлення кінцівки розроблюваного мобільного робота, яка представлена на рисунку 2.3.

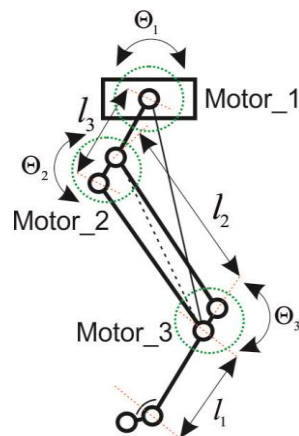


Рисунок 2.3 – Схематичне представлення кінцівки розроблюваного мобільного робота

Motor_1, ..., motor_3 – виконавчі пристрої (серводвигуни); $\Theta_1, \dots, \Theta_3$ – кути поворотів для кожного виконавчого пристрою (можуть бути від 0-360 градусів); l_1, \dots, l_2 – довжина відповідного ланцюга кінцівки мобільного робота. Для керування переміщенням необхідно розрахувати параметри

керування положенням кожної ланки зооморфного мобільного робота типу Spot для кінцівок. Для цього в рамках цих досліджень, пропонується використовувати трисегментний пантограф. Трисегментний пантограф – це механізм, що складається з трьох зчеплених між собою сегментів, що рухаються у взаємодії з двома фіксованими точками. Цей механізм використовується для збільшення діапазону руху або зменшення зусилля, необхідного для здійснення руху, у різноманітних механізмах та пристроях.

Схема трисегментного пантографа складається з двох паралельних пластин, що з'єднані між собою трьома сегментами. На кінці кожного сегмента закріплені сполучні елементи, що дозволяють зчеплювати пантограф з механізмом, який потребує змінення положення.

Коли один з кінців пантографа переміщується, наприклад, внаслідок руху елемента механізму, інші кінці пантографа починають рухатися з відповідними кутовими змінами. Це дозволяє збільшити або зменшити рух потрібної величини.

Базуючись на схемі трисегментний пантограф, розробимо кінематичну модель ноги зооморфного мобільного робота Spot, яка представлена на рисунку 2.4.

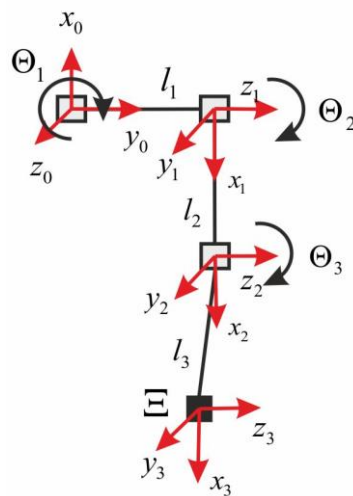


Рисунок 2.4 – Розроблена кінематична схема моделі ноги зооморфного мобільного робота

Пряма задача кінематики необхідна для обчислення координат кінцевого положення останньої ланки робота. У нашому випадку, вона буде вирішуватись для ноги крокуючого робота. Як точку відліку виберемо центр мас нашої конструкції, її початкові координати $O(x_m, y_m, z_m)$. Шуканими будуть координати положення стопи робота, які відповідають точці $\Xi(x_3, y_3, z_3)$. Для цього на базі рисунку 2.4 розробимо матрицю обертання для кожного з'єднання:

– для з'єднання 0-1:

$$R_1^0 = \begin{bmatrix} \cos\Theta_1 & -\sin\Theta_1 & 0 \\ \sin\Theta_1 & \cos\Theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} -\cos\Theta_1 & 0 & -\sin\Theta_1 \\ -\sin\Theta_1 & 0 & \cos\Theta_1 \\ 0 & 1 & 0 \end{bmatrix}; \quad (2.1)$$

– для з'єднання 1-2:

$$R_2^1 = \begin{bmatrix} \cos\Theta_2 & -\sin\Theta_2 & 0 \\ \sin\Theta_2 & \cos\Theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta_2 & -\sin\Theta_2 & 0 \\ -\sin\Theta_2 & \cos\Theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad (2.2)$$

– для з'єднання 2-3:

$$R_3^2 = \begin{bmatrix} \cos\Theta_3 & -\sin\Theta_3 & 0 \\ \sin\Theta_3 & \cos\Theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta_3 & -\sin\Theta_3 & 0 \\ -\sin\Theta_3 & \cos\Theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.3)$$

Наступним кроком розрахуємо вектори зміщень, для кожного з'єднання розробляемого зооморфного мобільного робота:

– для з'єднання 0-1:

$$d_1^0 = \begin{bmatrix} -s_{\Theta_1} \cdot l_1 \\ -c_{\Theta_1} \cdot l_1 \\ 0 \end{bmatrix}; \quad (2.4)$$

– для з'єднання 1-2:

$$d_2^1 = \begin{bmatrix} c_{\Theta_2} \cdot l_2 \\ s_{\Theta_2} \cdot l_2 \\ 0 \end{bmatrix}; \quad (2.5)$$

– для з'єднання 2-3:

$$d_3^2 = \begin{bmatrix} c_{\Theta_3} \cdot l_3 \\ s_{\Theta_3} \cdot l_3 \\ 0 \end{bmatrix}. \quad (2.6)$$

На базі отриманих матриць (2.1-2.3) та векторів зміщень (2.4-2.6), отримуємо матрицю гомогенних перетворень з використанням наступної формули:

$$H_m^n = \begin{bmatrix} R_{n+1}^n & d_{n+1}^n \\ 00 & 01 \end{bmatrix},$$

де n – порядковий номер координатної системи виконавчого механізму, в результаті було отримано матрицю гомогенних перетворень. Матриці гомогенних перетворень – це матриці 4x4, які використовуються для представлення та обчислення геометричних перетворень у тривимірному просторі. Кожна матриця гомогенних перетворень включає матрицю 3x3, що представляє лінійне перетворення, та вектор 3x1, що представляє зсув.

Матриці гомогенних перетворень дозволяють зручно виконувати різні операції з об'єктами у тривимірному просторі, такі як перенесення, обертання, масштабування, проєкцію та злиття об'єктів. Крім того, вони дозволяють об'єднувати кілька перетворень в одну матрицю для більш ефективного обчислення, яка має наступний загальний вид (2.7) та фінальний вид (2.8):

$$H_3^0 = H_1^0 \times H_2^0 \times H_3^2, \quad (2.7)$$

$$H_3^0 = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix}, \quad (2.8)$$

де h_{nm} – елементи матриці H_3^0 (n – номер рядку, m – номер стовпця).

Внаслідок чого всі елементи матриці (2.8) можна розрахувати за наступними моделями, які представлені у таблиці 2.2.

Таблиця 2.2 – Моделі розрахунків елементів матриці гомогенних перетворень

Елемент матриці	Формула розрахунку
1	2
h_{11}	$-\cos \Theta_1 \cdot \cos(\Theta_2 - \Theta_3)$
h_{12}	$\cos \Theta_1 \cdot \sin(\Theta_2 + \Theta_3)$
h_{13}	$-\sin \Theta_1$
h_{14}	$-\cos \Theta_1 \cdot (\cos(\Theta_2 - \Theta_3) \cdot l_3 + \cos \Theta_2 \cdot l_2) - \sin \Theta_1 \cdot l_1$
h_{21}	$-\sin \Theta_1 \cdot \cos(\Theta_2 - \Theta_3)$
h_{22}	$\sin \Theta_1 \cdot \sin(\Theta_2 + \Theta_3)$
h_{23}	$\cos \Theta_1$
h_{24}	$-\sin \Theta_1 \cdot (\cos(\Theta_2 - \Theta_3) \cdot l_3 + \cos \Theta_2 \cdot l_2) - \cos \Theta_1 \cdot l_1$
h_{31}	$\sin(\Theta_2 + \Theta_3)$
h_{32}	$\cos(\Theta_2 - \Theta_3)$
h_{33}	0
h_{34}	$\sin(\Theta_2 + \Theta_3) \cdot l_3 + \sin \Theta_2 \cdot l_2$

Продовження таблиці 2.2

1	2
h_{41}	0
h_{42}	0
h_{43}	0
h_{44}	1

Отримані моделі розрахунків елементів матриці гомогенних перетворень дозволить скласти таблицю параметрів керування положенням ланки зооморфного мобільного робота типа Spot для кінцівок. Для цього побудуємо таблицю 2.3 з рекомендацією параметрів для базових параметрів кінцівок мобільного робота.

Таблиця 2.3 – Значення для параметрів керування положенням ланки зооморфного мобільного робота типа Spot для кінцівок

	Θ	α	r	d
1	$\Theta_1 - 180^0$	90^0	l_1	0
2	Θ_2	0	l_2	0
3	Θ_3	0	l_3	0

Отримані результати в таблиці 2.3 дозволяють зрозуміти параметри кутів для кожного ланцюга, а це дає можливість приступити к аналізу та вибору виконавчих механізмів (серводвигунів).

2.2 Обґрунтування та вибір виконавчих механізмів (серводвигунів)

При виборі серводвигунів для мобільного зооморфного робота необхідно враховувати наступні фактори:

– динамічні характеристики: зооморфний робот повинен мати можливість швидко та плавно рухатися, повертатися та зупинятися. Для цього необхідно вибирати серводвигуни з високим коефіцієнтом прискорення та декременту, а також із низькими інерційними моментами;

– компактність та легкість: зооморфний робот повинен бути компактним та легким, щоб мати максимальну маневреність та здатність до пересування у складних умовах. Для цього необхідно вибирати серводвигуни з високою потужністю та низькою вагою;

– енергоефективність: зооморфний робот повинен мати тривалий час від однієї зарядки батареї. Для цього необхідно вибирати серводвигуни із високою енергоефективністю;

– надійність: зооморфний робот має бути надійним та мати довгий термін служби. Для цього необхідно вибирати серводвигуни з високим ступенем надійності та довговічності;

– простота управління: зооморфний робот повинен бути легко керованим і мати можливість швидкого налаштування серводвигунів для різних завдань. Для цього необхідно вибирати серводвигуни із простим та інтуїтивно зрозумілим інтерфейсом управління.

Виходячи з вище перерахованих факторів, було проведено аналіз та обрані наступні виконавчі механізми (серводвигунів), які представлені на рисунку 2.5, а порівняння їх характеристик приведено в таблиці 2.4.



а) сервопривод MG995 Tower Pro 180° [11];

б) сервопривод TowerPro MG996R [12];

с) сервопривод TD-8135MG 180° [13]

Рисунок 2.5 – Обрані виконавчі механізми для розроблюваного мобільного робота

Таблиця 2.4 – Порівняння основних характеристик сервоприводів

Характеристика	MG995 Tower Pro	TowerPro MG996R	TD-8135MG
Потужність (кг/см)	9,4	11	35
Швидкість (сек/60 градусів)	0,20	0,13	0,16
Кут повороту (градуси)	180	180	270
Напруга живлення (В)	4,8-7,2	4,8-7,2	4,8-6,0
Тип двигуна	Coreless	Coreless	Brushless
Розміри (мм)	40,7 x 19,7 x 42,9	40,7 x 19,7 x 42,9	40 x 20 x 39,5
Вага (г)	55	55	53
Ціна (грн)	135-150	185-200	~500-650

Характеристики серводвигунів MG995 Tower Pro та TowerPro MG996R дуже схожі, але у MG996R є трохи більш висока потужність та швидкість. Обидва серводвигуни можуть повертатися на 180 градусів і мають однакові розміри та вагу. Вони також використовують однаковий тип двигуна – Coreless.

Серводвигун TD-8135MG має значно більшу потужність і ширший кут повороту (270 градусів). Він також має нижчу швидкість, ніж MG996R, але досі досить швидкий. TD-8135MG використовує Brushless мотор, який зазвичай більш енергоефективний та надійний. Він також трохи менший і легший, ніж MG995 Tower Pro і TowerPro MG996R, але його ціна дуже велика, внаслідок цього в рамках даних досліджень буде використовуватись для керування переміщенням зооморфного мобільного робота сервопривод TowerPro MG996R.

2.3 Розробка конструкції зооморфного робота

Розробка конструкцій зооморфних роботів типу Spot для друку на 3D-принтері має свої особливості. Нижче наведено деякі з них:

– облік матеріалів та технології друку: при розробці конструкції зооморфного робота необхідно враховувати матеріал, з якого буде виготовлений робот, та технологію 3D-друку. Різні матеріали мають різну міцність, жорсткість, стійкість до дії температури та хімічних речовин. Також технологія друку може обмежувати можливість створення деяких деталей. Тому під час проектування необхідно враховувати ці чинники;

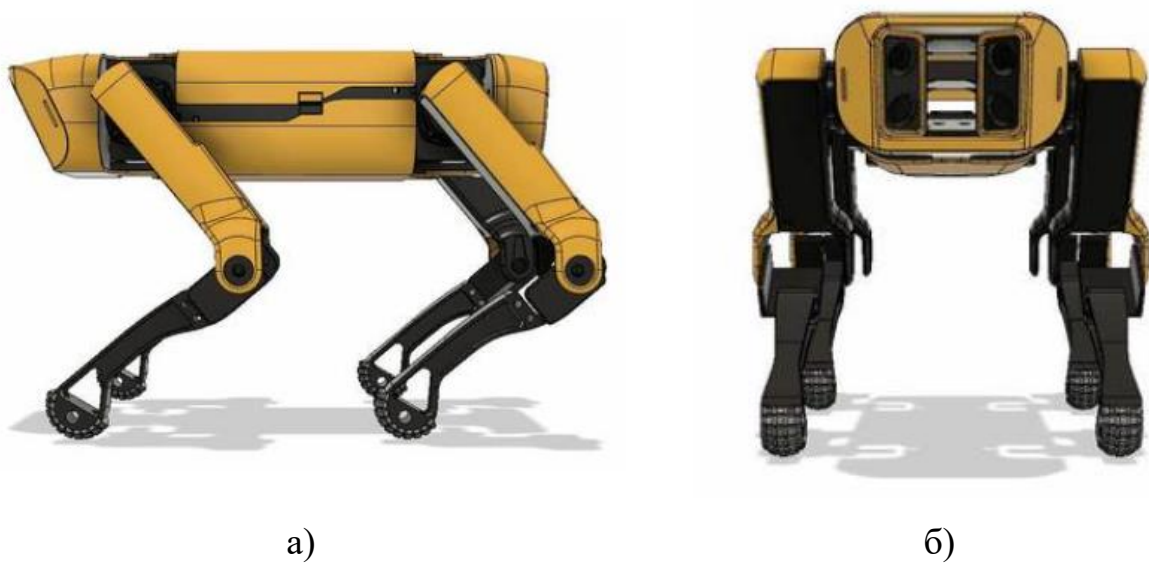
– імітація рухів тварин: зооморфний робот повинен імітувати рух тварини, яку він представляє. Для цього необхідно вивчити особливості анатомії та рухів тварини та ретельно спроектувати відповідні частини робота;

– мінімізація маси: важливим чинником розробки зооморфних роботів є мінімізація маси конструкції. Легкі роботи краще маневрують і витрачають менше енергії на пересування. Тому необхідно мінімізувати масу всіх частин робота, включаючи мотори, акумулятори та деталі корпусу;

– модульна конструкція: модульність конструкції зооморфного робота дозволяє швидко та легко замінювати та налаштовувати окремі частини робота. Це спрощує процес складання та дозволяє швидко виправляти помилки в конструкції;

– використання сенсорів: зооморфні роботи можуть використовувати різні сенсори для визначення навколишнього середовища та управління рухами. Наприклад, сенсори відстані, гіроскопи, акселерометри та камери можуть бути використані для виявлення перешкод, визначення напрямку руху та управління кутом повороту.

Внаслідок цього в «голові» розроблюваного мобільного робота планується розмістити систему комп'ютерного зору з системою відчуття у вигляді сенсорів, а саму конструкцію корпусу, розробити у вигляді окремих модулів, це дозволить отримати всі деталі за допомогою 3D друку за допомогою метода FDM. Загальний вид конструкції розробляемого мобільного робота представлено на рисунку 2.6.



а)

б)

а) вигляд збоку ; б) вигляд зпереду

Рисунок 2.6 – Загальний вид конструкції мобільного робота типа Spot

2.4 Розробка 3D моделі деталей конструкції мобільного робота

Fusion360 є популярним програмним забезпеченням для 3D-моделювання та є відмінним вибором для розробки деталей мобільного робота для подальшого 3D-друку з кількох причин:

- інтеграція: Fusion360 інтегрується з багатьма іншими програмними продуктами та сервісами Autodesk, що дозволяє легко переносити деталі між різними програмами;

- підтримка 3D-друку: Fusion360 має вбудовані функції, які полегшують підготовку моделей для 3D-друку, такі як генерація підтримки, перевірка на перетини, а також експорт у потрібний формат;

- інтуїтивний інтерфейс: Fusion360 має простий та інтуїтивно зрозумілий інтерфейс, що дозволяє швидко освоїти програму та розпочати створення 3D-моделей;

- хмарний сервіс: Fusion360 – хмарний сервіс, що дозволяє працювати з проектами з будь-якого місця та на будь-якому пристрої.

В цілому, Fusion360 є відмінним вибором для розробки деталей мобільного робота для подальшого 3D-друку, оскільки він має необхідні функції та переваги, які полегшують процес розробки та підготовки деталей для 3D-друку.

Наведемо приклад послідовності розробки моделі 3D – деталі в Fusion360. На першому кроці створимо "New Design" із головного меню Fusion360 для створення нового документа. Вибір типу деталі: виберіть тип деталі, яку потрібно створити, наприклад, тіло, поверхню, компонент або креслення, як представлено на рисунку 2.7.

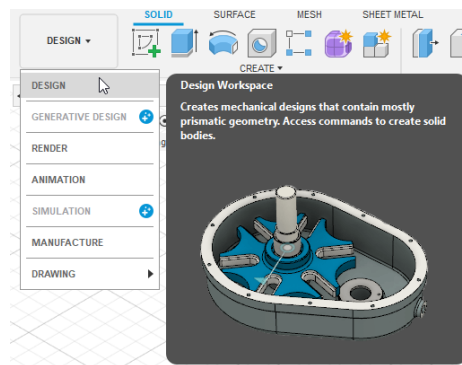


Рисунок 2.7 – Виберіть тип деталі

Далі, використовуючи команду "Sketch", оберемо площину, на якій за допомогою інструменту «2 Point Rectangle», створимо ескіз деталі, як представлено на рисунку 2.8.

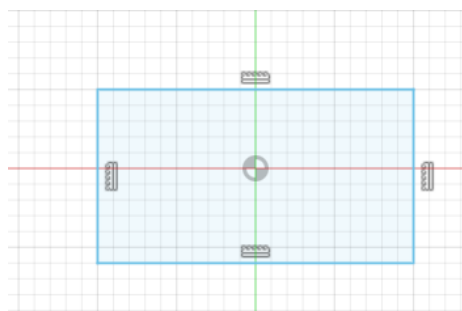


Рисунок 2.8 – Створення ескізу деталі

Наступним кроком, не виходячи з "Sketch", проведемо заокруглення країв деталі, за допомогою команди «Fillet» на $R=10\text{мм}$, як представлено на рисунку 2.9.

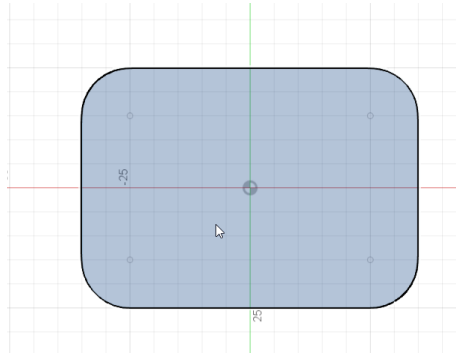
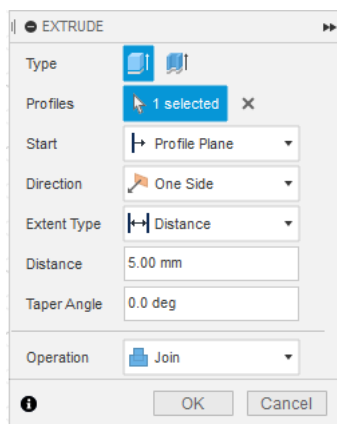


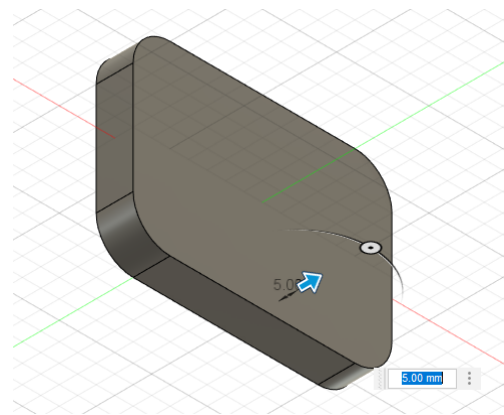
Рисунок 2.9 – Створення ескізу деталі

Далі проведемо створення 3D моделі заготовки, за допомогою команди «Extrude» на 10 мм, приклад параметрів налаштування цієї команди приведено на рисунку 2.10, а, а отриманий результат на рисунку 2.10, б.



а)

а) параметри налаштування команди «Extrude»;



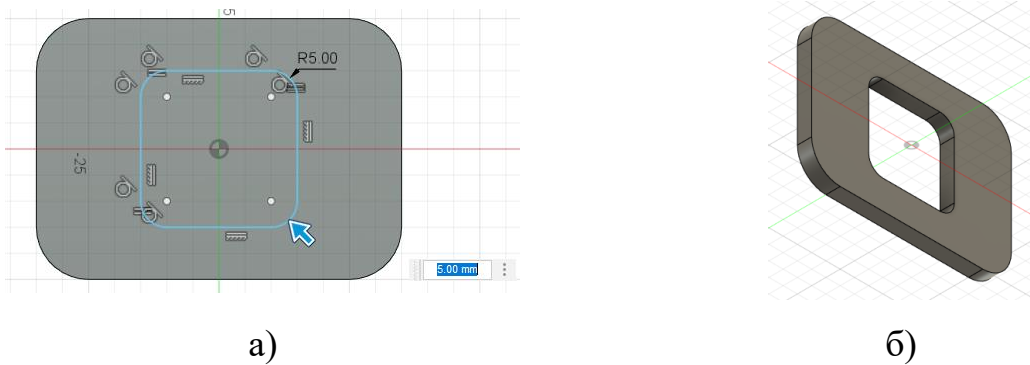
б)

б) отриманий результат

Рисунок 2.10 – Приклад параметрів налаштування команди «Extrude»

Зробимо технологічний отвір, всередині отриманої деталі, для того щоб була можливість провести кабель-менежмент при підключенні камери, датчиків. З точки зору надійності конструкцій, отвір доцільно зробити

посередині деталі. Для цього оберемо площину, та в режимі "Sketch" вкажемо параметри отвору, як представлено на рисунку 2.11, а, а результат виконання команди приведено на рисунку 2.11, б.



а)

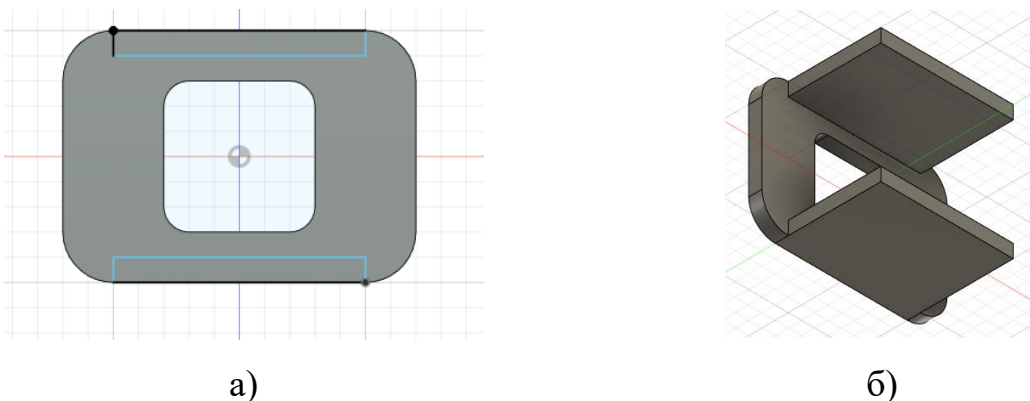
б)

а) Ескіз отвіру в режимі "Sketch";

б) отриманий отвір;

Рисунок 2.11 – Отримання технологічного отвору для кабель-менежмент

Для кріплення розробляємої деталі з елементами корпусу мобільного робота, розробимо кріпінні елементи, за допомогою команди "Sketch", нарисуємо ескізи та видаavimo їх на відстань довжини обраного серводвигуна TowerPro MG996R. Ескіз кріплення представлено на рисунку 2.12, а, а отримана деталь на рисунку 2.12, б.



а)

б)

а) ескіз кріплення;

б) деталь мобільного робота;

Рисунок 2.12 – Модель деталі мобільного робота з кріпленням

На базі отриманої 3D моделі, необхідно зробити крепіжні отвори під $M4 \times 20$ мм та отвору під фланцевий кульковий підшипник типу F625zz. Остаточний вид деталі мобільного робота спроектованого у Fusion360, представлено на рисунку 2.13.

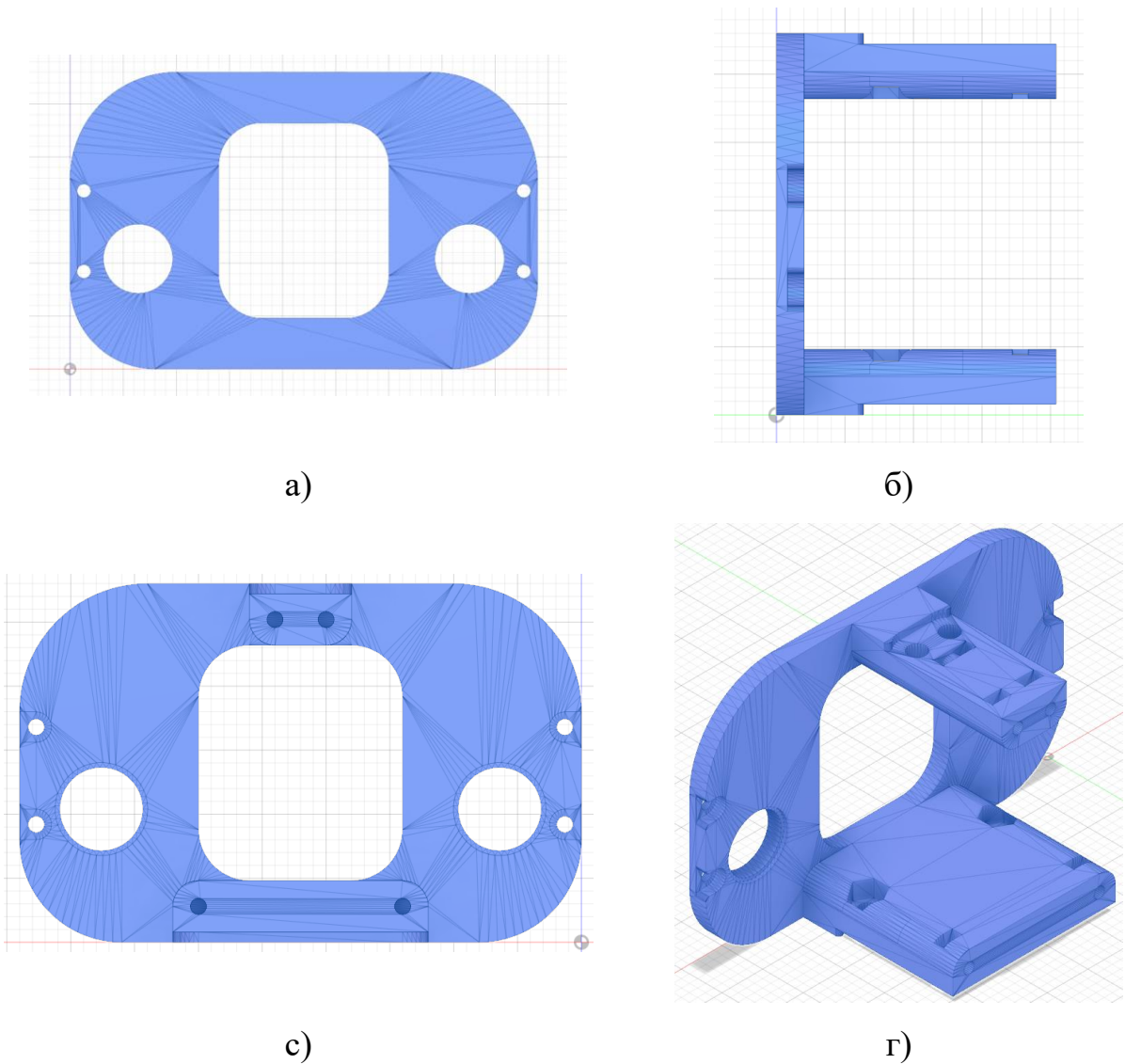
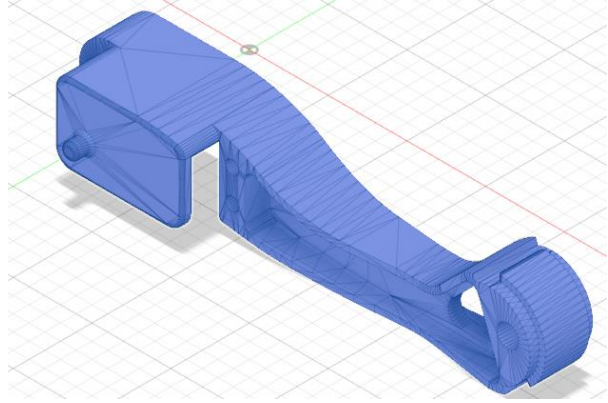


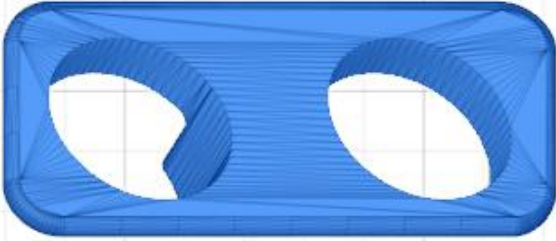
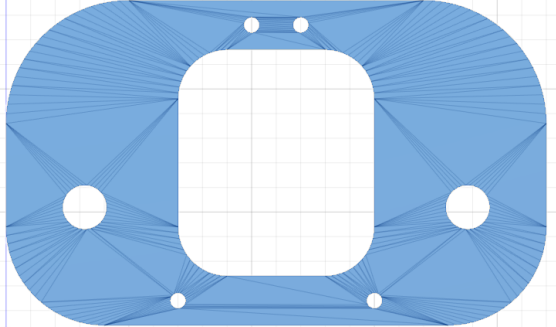
Рисунок 2.13 – П-деталь корпусу мобільного робота

Аналогічно розробленій П-деталі корпусу мобільного робота, було побудовано всі необхідні деталі, зразок и опис яких представлено в таблиці 2.5.

Таблиця 2.5 – Приклад та опис спроектованих деталей мобільного робота в Fusion360

3D – модель деталі	Описання
1	2
	<p>Наконечник для лапи зооморфного робота повинен забезпечувати добре зчеплення з поверхнею, щоб робот міг добре пересуватися та не ковзати. Для цього можна використовувати матеріали з високою адгезією, такі як гума або силікон, або створити спеціальний малюнок на поверхні наконечника, який забезпечуватиме краще зчеплення.</p>
	<p>Кріплення серводвигуна motor_1 до плеча мобільного робота має забезпечувати міцну та надійну фіксацію, а також дозволяти йому вільно повертатися для виконання заданих дій.</p>
	<p>Конструкція першої ланки кінцевої частини мобільного робота повинна забезпечувати її вільний та точний маневр у просторі. Ця ланка повинна мати достатню жорсткість та міцність, щоб підтримувати кінцеву частину робота у правильному положенні та утримувати її від деформацій при виконанні заданих дій.</p>

Продовження таблиці 2.5

1	2
	<p>Кріплення ультразвукового датчика призначене для закріплення датчика всередині "голови" мобільного робота. В конструкції таких елементів два, правий та лівий датчик від системи комп'ютерного зору.</p>
	<p>Передня кришка для П – деталі (рис.2.13), призначена для фіксації корпуса «голови» мобільного робота, а також серводвигуна moter_1.</p>
	<p>Корпус голови мобільного робота – це частина конструкції, яка розташовується в передній частині робота та містить ультразвукові датчики та камери системи комп'ютерного зору. Він виконує низку важливих функцій, таких як захист електроніки та сенсорів, а також забезпечує оптимальний кут огляду для камер та датчиків.</p>

2.5 Розробка 3D моделі зборки мобільного робота типу Spot

Для створення 3D складання мобільного робота зооморфного типу Spot у середовищі Fusion360 слід виконати таку послідовність дій: створити "New design" у діалоговому вікні "Home", та ввести назву проекту та одиниці виміру, в нашому випадку одиниці виміру обрані мм. Далі необхідно обрати «Корневий або базовий елемент збірки», для цього в лівій частині екрану вкладки "Browser" оберемо команду "Create new component". Наступним

кроком є додавання першої деталі робота з файлів вже створених деталей в розділі 2.4. Для цього натискаємо кнопку "Create new component from selection", оберемо деталь та натискаємо "ОК". Виділимо новий компонент у браузері компонентів та перетягнемо до нього деталь, щоб зробити збірку. Використовуючи функцію "Контакти" у середовищі Fusion360 для з'єднання деталей у складання необхідно зробити наступні кроки:

- обрати дві деталі, які потрібно з'єднати;
- натисніть кнопку "Contact" на панелі інструментів;
- виберіть точки контакту на кожній деталі, яка буде з'єднана;
- натисніть "ОК", щоб зберегти збірку.

Виконуючи вище вказані послідовності збірки у середовищі Fusion360, отримаємо деталізовану 3D-модель збірки мобільного зооморфного робота, яку приведено на рисунку 2.13.

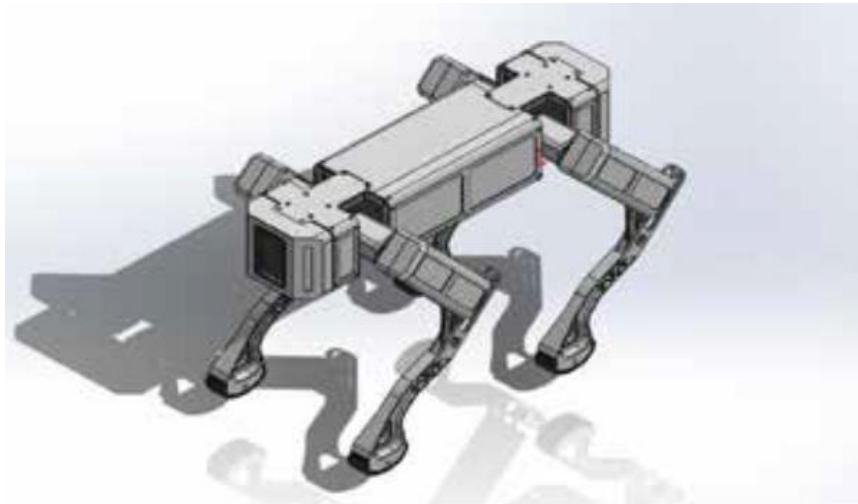


Рисунок 2.13 – 3D-модель збірки мобільного зооморфного у середовищі Fusion360

2.6. Збірка макета зооморфного робота типу Spot

Для подальших досліджень з розробки системи керування зооморфним мобільним роботом, необхідно зібрати макет згідно зі спроектованими 3D

моделями, які представлені в розділах 2.4 та 2.5. Виробництво спроектованих деталей буде робитись методом 3D друку FDM (Fused Deposition Modeling), який був обраний, через його відносну простоту та доступність. FDM – це найпоширеніший метод 3D друку, який працює шляхом нагрівання пластикового матеріалу та подальшої екструзії через сопло, створюючи тверді шари, які поступово створюють тривимірний об'єкт. В якості матеріалу обрано PLA (полілактид) – це біорозкладний пластик, який виготовляється з рослинних матеріалів, та не дуже дорогий. 3D-друк проводився на 3D принтері Anet3D Pro з наступними налаштуваннями слайсера UltiMaker Cura 5.2.2, які вказані в таблиці 2.6.

Таблиця 2.6 – Параметри налаштування 3D друку деталей мобільного робота

Параметр	Значення
1	2
Висота шару (мм)	0,2
Висота першого шару (мм)	0,2
Товщина стінки (мм)	0,8
Товщина дна кришки (мм)	0,8
Шар дна (мм)	4
Щільність заповнення (%)	20
Шаблон заповнення	лінія
Температура сопла (°C)	200
Температура столу (°C)	60
Швидкість друку (мм\с)	80
Висота підняття осі Z (мм)	0,2

Після отримання надрукованих 3D моделей, приступимо до зборки конструкції мобільного зооморфного робота типа Spot відповідно розробленої 3D моделі зборки в розділі 2.5. Фотографія базового каркасу мобільного робота зі встановленими фланцевими кульковими підшипниками типа F625zz представлена на рисунку 2.14.



Рисунок 2.14 – Базовий каркас мобільного робота зі встановленими фланцевими кульковими підшипниками типа F625zz

Наступним кроком необхідно зібрати кінцівку мобільного зооморфного робота зі встановленими в нього серводвигунами TowerPro MG996R, бажано перед встановленням серводвигунів виставити максимальний та мінімальний кут повороту, в нашому випадку це від $0^{\circ} < \Theta < 180^{\circ}$. Фотографія зібраної кінцівки зооморфного мобільного робота зі встановленими двома серводвигунами TowerPro MG996R, представлено на рисунку 2.15.

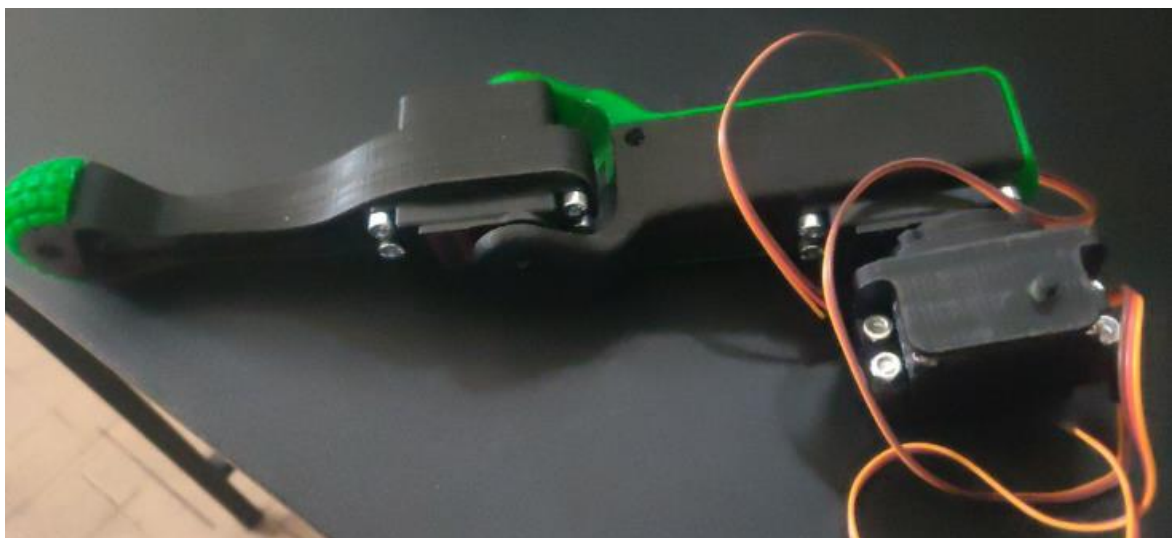
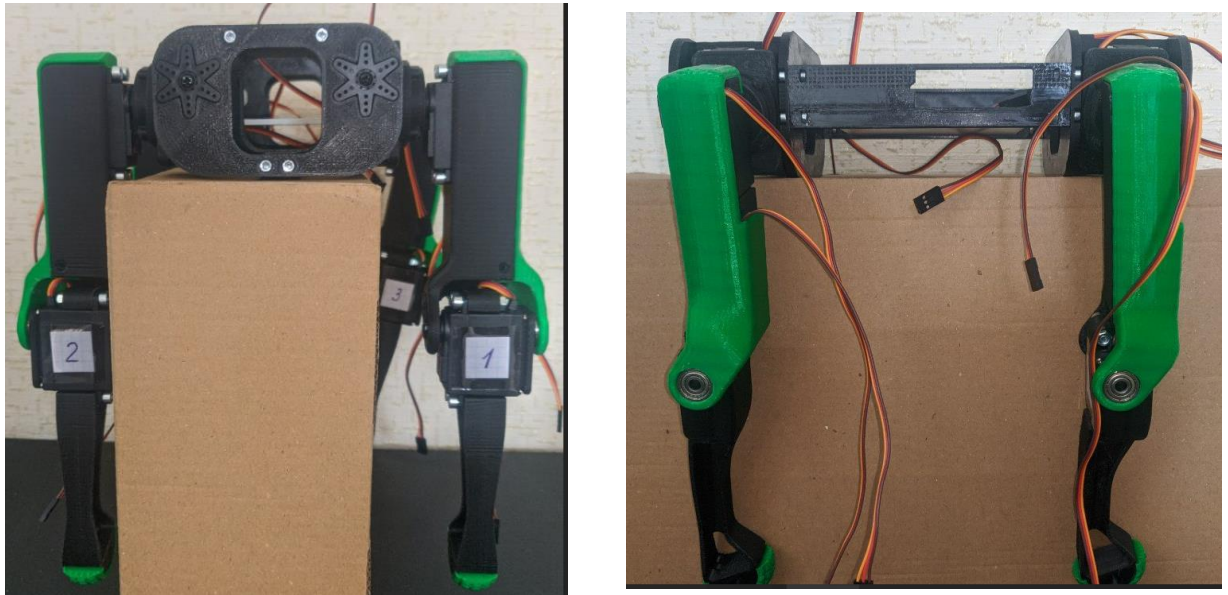


Рисунок 2.15 – Зібраної кінцівки зооморфного мобільного робота

Загальний вид зібраного макета зооморфного мобільного робота, без системи керування зі встановленими серводвигунами TowerPro MG996R представлено на рисунку 2.16.



а)

б)

а) вид спереду («голови»);

б) вид збоку;

Рисунок 2.16 – Загальний вид зібраного макету зооморфного мобільного робота типу Spot без системи керування

2.7 Висновки до 2 розділу

В ході виконання другого розділу досліджень з розробки системи керування мобільним зооморфним роботом типу Spot, було зроблено:

- розроблено схематичне представлення загальної конфігурації проєктованого мобільного робота, з вказівкам центру тяжіння всієї конструкції;
- запропоновані та обрані основні габаритні параметри мобільного робота;

- розроблено схематичне представлення кінцівки мобільного робота, з вказаними базовими параметрами;
- розроблена кінематична схема моделі ноги зооморфного мобільного робота, та вирішена пряма задача кінематики, необхідна для обчислення координат кінцевого положення останньої ланки робота;
- розроблена модель розрахунків елементів матриці гомогенних перетворень, що дозволила вичислити значення для параметрів керування положенням ланки зооморфного мобільного робота типу Spot для кінцівок;
- на базі отриманих значень, було проведено аналіз та обрані виконавчі механізми для керування переміщенням зооморфного мобільного робота, в рамках даних досліджень було обрано серводвигун TowerPro MG996R 1800;
- основується на обраних виконавчих механізмах, були спроектовані 3D моделі деталей мобільного робота в середовищі Fusion360 з урахуванням їх друку на 3D-принтері;
- розроблена 3D моделі зборки зооморфного мобільного робота типу Spot у середовищі Fusion360;
- за допомогою 3D друку було надруковано та зібрано макет зооморфного мобільного робота, без системи керування для проведення досліджень на лабораторному макеті.

3 РОЗРОБКА СИСТЕМИ КЕРУВАННЯ МОБІЛЬНИМ РОБОТОМ

3.1 Аналіз та вибір апаратних модулів для реалізації системи керування

Вибір апаратних модулів для розроблюваного макета зооморфного мобільного робота типу Spot має бути заснований на кількох критеріях, які можуть бути виділені на основі вимог до функціональності робота, його ефективності та надійності:

- потужність мікропроцесора: одним із найважливіших критеріїв вибору апаратних модулів є потужність мікропроцесора. Для зооморфного мобільного робота Spot необхідний процесор із високою продуктивністю, щоб забезпечити швидке виконання обчислювальних завдань та підтримку алгоритмів машинного навчання. Тому рекомендується використовувати процесори високого класу, або ті, що реалізовані на одноплатних мікрокомп'ютерах сімейства Raspberry Pi;

- камери та датчики: для забезпечення функціонування зооморфного мобільного робота Spot необхідні камери та датчики, які можуть забезпечити надійну та точну навігацію та виявлення перешкод. Для цього можна використовувати камери з високою роздільною здатністю та датчики, такі як ультразвуковий та акселерометр;

- батареї: важливим критерієм вибору є енергоефективність апаратних модулів, оскільки зооморфний мобільний робот Spot має працювати на батареях протягом тривалого часу. Тому слід використовувати батареї високої ємності та ефективні компоненти, які споживають мінімальну кількість енергії, з розрахунку живлення 12 серводвигунів;

- система управління: для забезпечення безперебійної роботи зооморфного мобільного робота Spot необхідна надійна система управління, яка має бути надійною та гнучкою. Рекомендується використовувати системи

управління високого класу, такі як ROS (Robot Operating System), які дозволяють керувати роботом за допомогою безлічі датчиків та пристроїв у режимі реального часу, або використовувати Raspberry Pi OS та розробити свою систему керування під розроблюваного мобільного робота;

– модулярність: зооморфний мобільний робот Spot повинен мати модульну архітектуру, щоб полегшити заміну та оновлення компонентів. Тому рекомендується використовувати апаратні модулі, які легко замінити та оновити в процесі роботи та досліджень.

Виходячи з вище перерахованих критеріїв, пропонується розробити дворівневу систему керування. У вигляді головного модулю для обробки зображень та прийняття рішень буде використовуватися потужний мікропроцесорний пристрій [14]. Виходячи з цього було проведено аналіз наступних апаратних модулів, які представлені на рисунку 3.1.



а)

б)

в)

а) Raspberry Pi 3 B+ [15];

б) NVIDIA Jetson- Jetson Nano Developer Kit [16];

в) Orange Pi 4 LTS 4Гб DDR4/16Гб EMMC [17]

Рисунок 3.1 – Модулі для системи керування вищого рівня

Порівняння технічних характеристик обраних апаратних модулів для реалізації вищого рівня керування зооморфним мобільним роботом типу Spot, приведена в таблиці 3.1

Таблиця 3.1 – Порівняння технічних характеристик обраних апаратних модулів вищого рівня керування

Параметри	Raspberry Pi 3 B+	NVIDIA Jetson Nano	Orange Pi 4 LTS
Процесор	Broadcom BCM2837B0	NVIDIA Cortex-A57	Rockchip RK3399
Частота	1,4 ГГц	1,43 ГГц	2,0 ГГц
Кількість ядер	4	4	6
Графічний процесор	Broadcom VideoCore IV	NVIDIA Maxwell GPU	ARM Mali-T860 MP4
Оперативна пам'ять	1 Гб DDR2	4 Гб LPDDR4	4 Гб DDR4
Вбудована пам'ять	МикроSD до 256 Гб	МикроSD до 128 Гб	16 Гб EMMC
Wi-Fi	802.11ac	802.11ac	802.11ac
Bluetooth	4,2	4,2	5,0
Ethernet	10/100/1000 Мбит/с	10/100/1000 Мбит/с	10/100/1000 Мбит/с
USB порти	4x USB 2.0, 2x USB 3.0	4x USB 3.0	4x USB 3.0
GPIO порти	40	40	26
Розміри	85 мм x 56 мм x 17 мм	70 мм x 45 мм	93 мм x 60 мм
Ціна	4290 грн.	9614 грн.	5187 грн.

З таблиці 3.1 порівняння параметрів Raspberry Pi 3 B+, NVIDIA Jetson Nano та Orange Pi 4 LTS видно, що Raspberry Pi 3 B+ має низьку частоту процесора та меншу оперативну пам'ять, ніж Jetson Nano та Orange Pi 4 LTS. Однак, він має ширший вибір портів USB та GPIO, що може бути важливим при розробці макета мобільного робота. Raspberry Pi 3 B+ може бути більш економічним вибором, порівняно з дорожчими та потужнішими апаратними платформами, такими як Jetson Nano або Orange Pi 4 LTS. Крім того, Raspberry Pi 3 B+ має велику спільноту розробників та підтримку в Інтернеті, що робить його легким у використанні та налагодженні.

Наступним кроком, оберемо апаратний модуль для системи керування нижнього рівня, це рішення дозволить зменшити обчислювальне навантаження та звільнити дані ресурси на реалізацію системи

комп'ютерного зору з системою ідентифікації та розпізнавання об'єктів на базі нейронних мереж. В рамках цього рекомендується використовувати модуль керування на базі мікропроцесорів сімейства AVR. Виходячи з цього, було проведено аналіз наступних апаратних модулів, які представлені на рисунку 3.2.



а) модуль Arduino Nano [18];

б) модуль Arduino Uno [19]

Рисунок 3.2 – Апаратні модулі для реалізації системи керування нижнього рівня

Порівняння технічних характеристик обраних апаратних модулів для реалізації нижнього рівня керування зооморфним мобільним роботом типу Spot, приведена в таблиці 3.2.

Таблиця 3.2 – Порівняння технічних характеристик обраних апаратних модулів нижнього рівня керування

Параметри	Arduino Nano	Arduino Uno
Мікроконтролер	ATmega328P	ATmega328P
Робоча частота	16 МГц	16 МГц
Оперативна пам'ять (SRAM)	2 Кб	2 Кб
Пам'ять програм (Flash)	32 Кб	32 Кб
Ширина слова даних	8 біт	8 біт
Число аналогових входів	8	6
Число цифрових входів/виходів	14 (в тому числі 6 з ШІМ)	14 (в тому числі 6 з ШІМ)
Розміри	18 мм x 45 мм	68,6 мм x 53,4 мм
Ціна	213 грн.	439 грн.

Обидва мікроконтролери мають однаковий мікроконтролер ATmega328P та робочу частоту 16 МГц. Вони також мають однаковий обсяг оперативної пам'яті та пам'яті програм (Flash). Однак, Arduino Nano має компактніший розмір, тому його зручніше використовувати в компактних пристроях.

Останнім кроком необхідно обрати апаратний модуль для керування 12 серводвигунами TowerPro MG996R 180° – це пристрій, який дозволяє керувати кількома серводвигунами одночасно і віддалено через шину управління. Виходячи з цього пропонуються наступні плати розширення, які представлені на рисунку 3.3., а порівняння їх характеристик приведено в таблиці 3.3.



а)

а) PCA9685 – 16-канальний PWM/Servo модуль з I2C [20];



б)

б) Arduino servo shield modul [21]

Рисунок 3.3 – Плати розширення для керування серводвигунами TowerPro MG996R

Таблиця 3.3 – Порівняння технічних характеристик плат розширення для керування серводвигунами TowerPro MG996R

Характеристика	PCA9685	Arduino Servo Shield
1	2	3
Тип	ШИМ драйвер для серводвигунів	ШИМ драйвер для серводвигунів
Кількість каналів	16	16
Роздільна здатність	12 біт	8 біт

Продовження таблиці 3.3

1	2	3
Частота ШІМ	до 1,6 кГц	50 Гц
Контроль напруги	Да	Ні
Інтерфейс зв'язку	I2C	I2C
Сумісність з Arduino	Так	Так
Ціна	~200-250 грн.	~250-300 грн.

Таким чином, якщо потрібне точне керування серводвигунами з високою роздільною здатністю та частотою ШІМ, то PCA9685 є більш підходящим вибором порівняно з Arduino Servo Shield. Крім того, PCA9685 має нижчу вартість. Якщо потрібне просте керування серводвигунами без необхідності точного позиціонування, Arduino Servo Shield може бути більш підходящим вибором.

Після вибору апаратних модулів вищого та нижнього рівня для керування зооморфним мобільним роботом типу Spot, можливо обрати апаратні модулі для отримання інформації про довкілля. Для виявлення перешкод, перед мобільним роботом буде використовуватися модуль ультразвукового датчика типа HC-SR04. Загальний вигляд датчика представлено на рисунку 3.3.

Датчик відстані HC-SR04 використовує ультразвукові хвилі для вимірювання відстані. Він широко використовується в робототехніці та інших додатках, де потрібний точний вимір відстані.

Ось кілька причин, чому HC-SR04 був обраний:

- точність: HC-SR04 забезпечує високу точність вимірювання відстані з похибкою менше 1%;
- надійність: датчик має просту конструкцію та надійний механізм вимірювання відстані, що робить його досить надійним для використання;
- легкість у використанні: HC-SR04 підключається до мікроконтролера за допомогою простого інтерфейсу, що робить його легким у використанні та програмуванні;

– доступність: HC-SR04 доступний на ринку за низькою ціною в межах 50 грн. і легко доступний для покупки.



Рисунок 3.4 – Ультразвуковий датчик типу HC-SR04

Також для контролю положення мобільного робота в просторі, та для реалізації переміщення зі збереженням рівноваги, потрібно обрати модуль, який матиме вбудований акселерометр та гіроскоп. Модуль GY-521 (MPU-6050) – це маленький модуль інерційних датчиків, який містить три осі гіроскопа (angular rate sensors) та три осі акселерометра (accelerometers). Ці датчики вимірюють прискорення та швидкості у трьох вимірах, дозволяючи визначити орієнтацію та рух об'єкта у просторі, загальний вид модулю представлено на рисунку 3.5.



Рисунок 3.5 – Загальний вид модуля GY-521 (MPU-6050)

Модуль GY-521 заснований на мікросхемі MPU-6050, яка поєднує в собі гіроскоп та акселерометр на одному кристалі. Для зручності використання модуль має інтерфейс I2C, що дозволяє підключити його до будь-якого мікроконтролера, який підтримує цей інтерфейс зв'язку. Основні характеристики модуля GY-521 приведені в таблиці 3.4.

Таблиця 3.4 – Основні технічні характеристики модуля GY-521

Параметр	Значення
1	2
Робоча напруга	3,3 або 5 В
Інтерфейс зв'язку	I2C
Діапазон вимірювання гіроскопа	$\pm 250, \pm 500, \pm 1000, \pm 2000$ град/сек
Діапазон вимірювання акселерометра	$\pm 2g, \pm 4g, \pm 8g, \pm 16g$
Частота виміру	до 1 кГц
Роздільна здатність	16 біт

Модуль GY-521 можна використовувати в різних додатках, таких як автоматична стабілізація та навігація багатьох пристроїв, наприклад, у квадрокоптерах та інших безпілотних літальних апаратах, роботах та інших пристроях, де потрібен вимір орієнтації та руху об'єкта.

3.2 Розробка структурної схеми системи керування

Структурна схема управління мобільним роботом зооморфного типу пропонується наступна:

– Raspberry Pi 3 B+ підключений до Arduino Nano через роз'єм UART. Це дозволяє Raspberry Pi 3 B+ відправляти команди Arduino Nano, використовуючи послідовний протокол зв'язку UART;

– Arduino Nano підключено до модуля PCA9685 через протокол i2c. Це дозволяє Arduino Nano керувати 12 серводвигунами TowerPro MG996R, використовуючи модуль PCA9685;

– Arduino Nano підключено до двох ультразвукових датчиків HC-SR04. Це дозволяє Arduino Nano отримувати дані від датчиків видалення об'єктів у певному напрямку;

– модуль GY-521 підключений до Raspberry Pi 3 B+ через протокол i2c та роз'єм GPIO. Це дозволяє Raspberry Pi 3 B+ отримувати дані про положення та прискорення за допомогою модуля GY-521.

Таким чином, Raspberry Pi 3 B+ виступає як система управління верхнього рівня, яка відправляє команди на Arduino Nano через UART. Arduino Nano, у свою чергу, управляє серводвигунами через PCA9685 і отримує дані від ультразвукових датчиків. Модуль GY-521 підключений безпосередньо до Raspberry Pi 3 B+, що дозволяє йому отримувати дані про положення та прискорення. Структурна схема системи керування мобільним роботом представлена на рисунку 3.6.

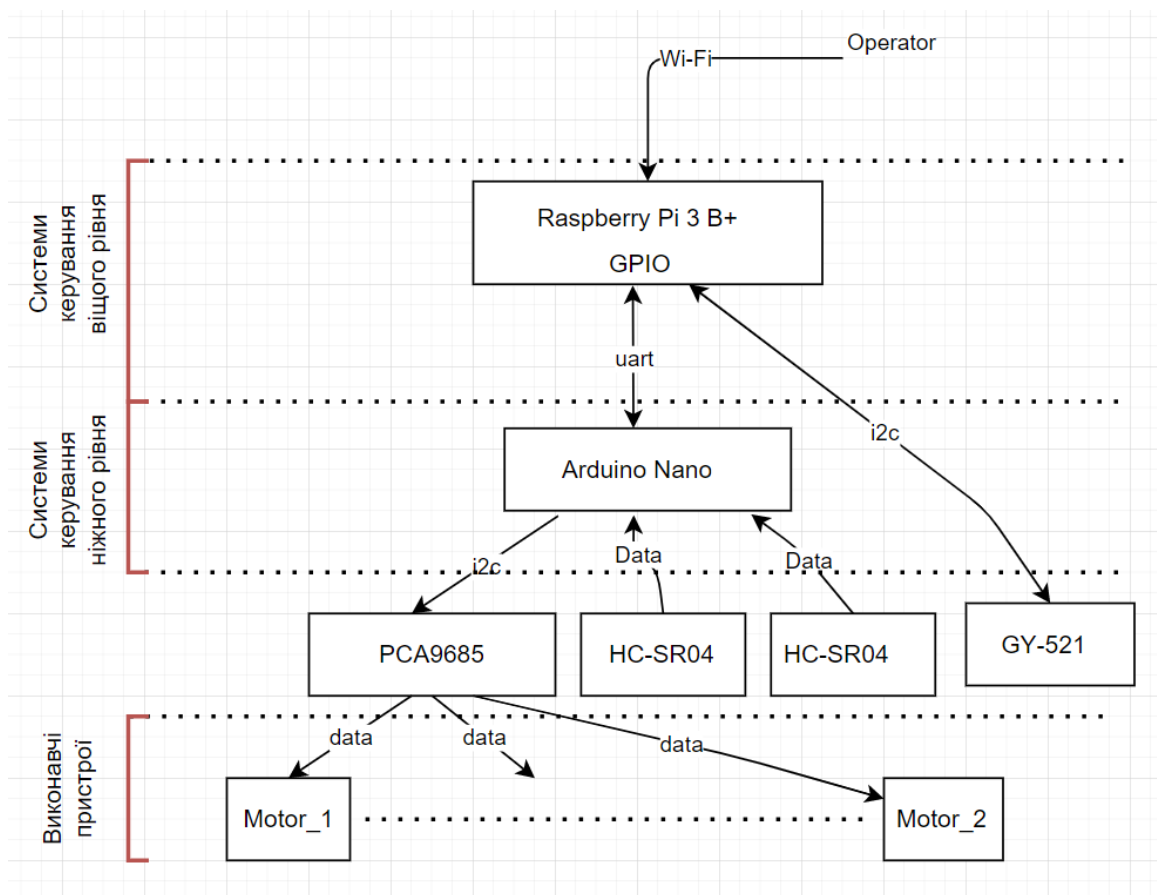


Рисунок 3.6 – Структурна схема системи керування мобільним роботом

3.3. Розробка схеми підключення апаратних модулів

В даному випадку управління мобільним роботом буде здійснюватися через Raspberry Pi 3 B+. Для цього мікроконтролер Arduino Nano буде підключений до Raspberry Pi 3 B+ через роз'єм UART. На Arduino Nano будуть підключені два ультразвукові датчики HC-SR04, які будуть використовуватися для вимірювання відстані до перешкод та виявлення перешкод на шляху робота.

Через протокол I2C Raspberry Pi 3 B+ взаємодітиме з модулем GY-521, який використовуватиметься для вимірювання орієнтації та руху робота в просторі. Модуль PCA9685 також буде підключений до Raspberry Pi 3 B+ через I2C і використовуватиметься для керування серводвигунами.

Для живлення всього робота використовуватиметься понижувальний модуль DC-DC перетворювач, який живиться від акумуляторів ємністю 1300 мАг.

В цілому, взаємодія управління мобільного робота здійснюватиметься через Raspberry Pi 3 B+, який контролюватиме всі підключені модулі та датчики, а також управлятиме рухом робота та його орієнтацією в просторі. Модулі PCA9685 та GY-521 будуть використовуватися для точного управління серводвигунами та визначення орієнтації та руху робота відповідно, а ультразвукові датчики HC-SR04 будуть використовуватися для виявлення перешкод на шляху робота та коригування його руху відповідно до цих виявлення. Схема підключення всіх апаратних модулів приведена на рисунку 3.7.

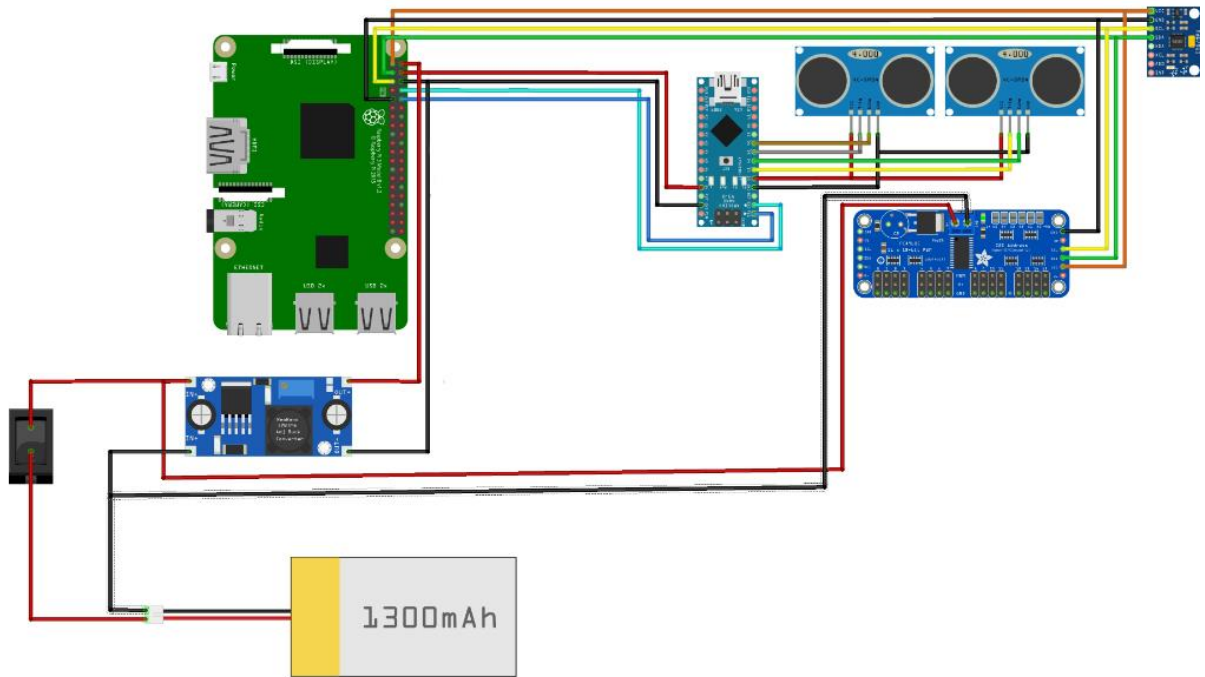


Рисунок 3.7 – Схема підключення апаратних модулів

Підключення датчика HC-SR04 безпосередньо до Raspberry Pi 3 B+ можливе і навіть простіше у реалізації, проте використання Arduino Nano для підключення датчиків має такі переваги:

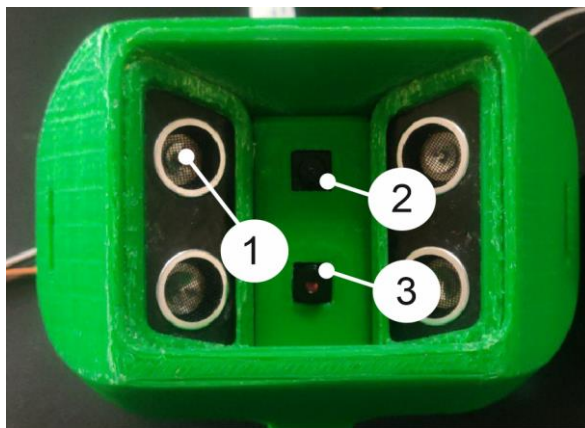
- зменшення навантаження на Raspberry Pi 3 B+: HC-SR04 працює із частотою 40 кГц і може створювати перешкоди в роботі інших підключених до Raspberry Pi 3 B+ пристроїв, таких як модуль GY-521 та PCA9685. При підключенні через Arduino Nano датчики працюватимуть незалежно від Raspberry Pi 3 B+, що дозволить зменшити навантаження на нього;

- обробка даних Arduino Nano: Arduino Nano може виконувати обробку даних від датчиків, таких як фільтрацію і перетворення даних у зручний формат, передаючи вже готові дані на Raspberry Pi 3 B+. Це може прискорити роботу системи та зменшити навантаження на Raspberry Pi 3 B+;

- розширення можливостей: використання Arduino Nano для підключення датчиків дозволяє легко додавати нові датчики та пристрої без необхідності змінювати конфігурацію підключення на Raspberry Pi 3 B+;

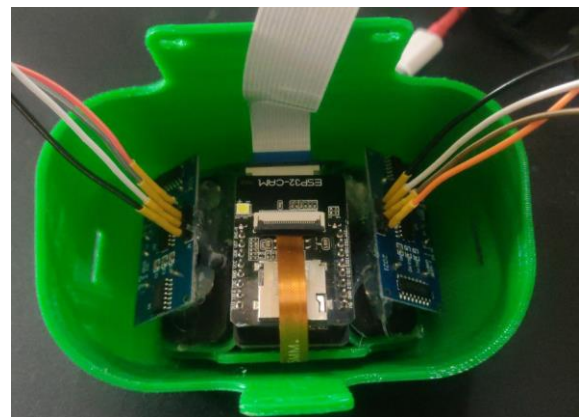
– таким чином, використання Arduino Nano для підключення датчиків HC-SR04 має низку переваг, які можуть полегшити роботу системи та зменшити навантаження на Raspberry Pi 3 B+.

Виходячи зі запропонованої схеми підключення (рис.3.7) було зібрано макет «голови» мобільного зооморфного робота, з розташованими в ньому системою комп'ютерного зору, яка складається з модулю ESP32-Cam та камери Raspberry Cam. Модуль ESP32-Cam буде використовуватися як камера візуалізації навколишнього простору на мобільний пристрій оператора. Камера Raspberry Cam використовується для отримання інформації для подальшої обробки, ідентифікації та прийняття управляючих сигналів для мобільного робота на базі Raspberry Pi 3 B+. Фотографія макет «голови» мобільного зооморфного робота представлена на рисунку 3.8.



а)

а) вид спереду (1. HC-SR04; 2. ESP32-Cam; 3. Raspberry Cam);

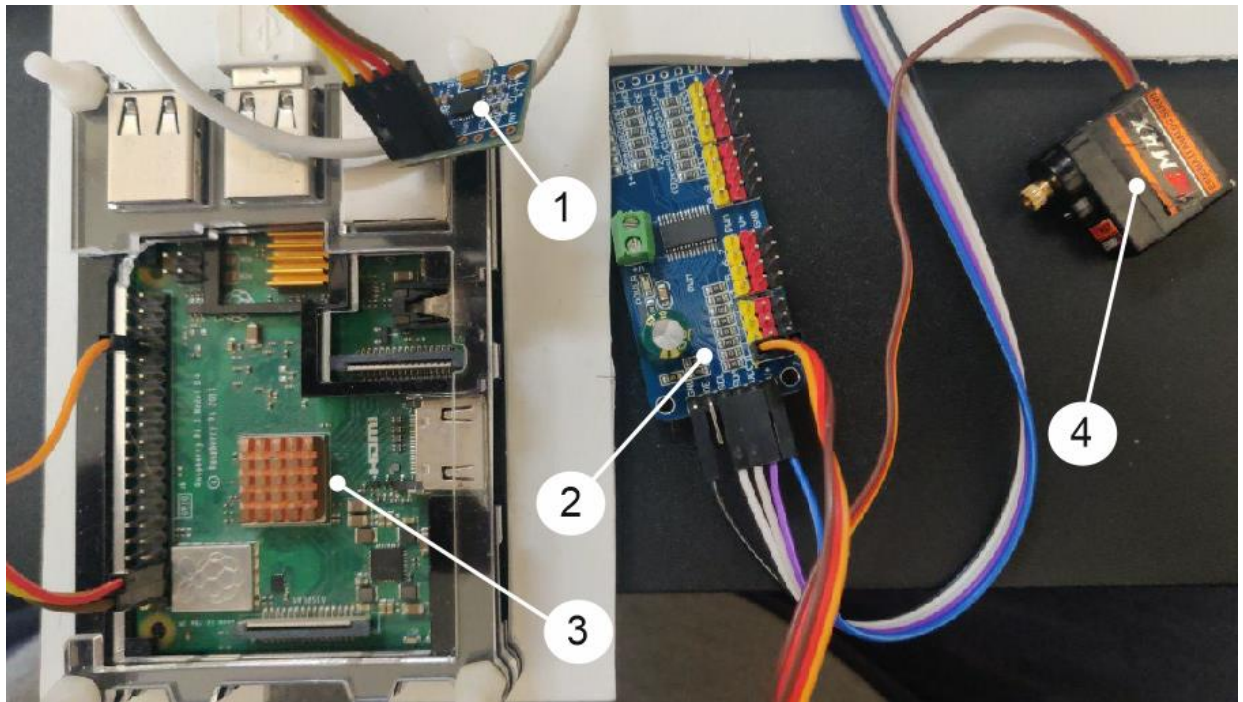


б)

б) вид усередині

Рисунок 3.8 – Апаратна реалізація «голови» мобільного зооморфного робота

Макет розробленої системи керування мобільним зооморфним роботом відповідно запропонованої схеми підключення (рис. 3.7) для тестування та відпрацювання програмного забезпечення представлено на рисунку 3.9.



1) GY-521; 2) PCA9685; 3) Raspberry Pi 3 B+; 4) тестовий серводвигун
Рисунок 3.9 – Макет розробленої системи керування мобільним зооморфним роботом

Тестова збірка системи керування зооморфним роботом на макеті, для розробки та опрацювання системи керування переведено на рисунку 3.10.



а)



б)

а) розроблена система керування;
б) система керування на макеті

Рисунок 3.10 – Тестова збірка системи керування зооморфним роботом на макеті

3.4 Розробка алгоритму керування зооморфними мобільними роботами

Загальний алгоритм керування зооморфним мобільним роботом може мати кілька етапів, які можна описати наступним чином:

Ініціалізація:

- увімкнення системи та початкова настройка всіх підключених компонентів (Raspberry Pi, Arduino Nano, модулі датчиків і серводвигуни);
- калібрування датчиків, якщо це необхідно (наприклад, ультразвукові датчики для точного вимірювання відстані).

Зчитування даних:

- Raspberry Pi зчитує дані з ультразвукових датчиків HC-SR04 для вимірювання відстані до перешкод;
- Raspberry Pi також отримує дані про орієнтацію та рух з модуля GY-521.

Аналіз даних:

- Raspberry Pi аналізує отримані дані для виявлення перешкод та визначення орієнтації та руху робота;
- на основі отриманих даних робот виробляє рішення щодо подальших кроків.

Планування руху:

- залежно від розташування та відстані до перешкод, Raspberry Pi розробляє стратегію руху для уникнення перешкод або досягнення певної цілі;
- визначається шлях руху (напрямок, кут, швидкість) та керування серводвигунами для досягнення бажаної орієнтації робота.

Виконання руху:

- Raspberry Pi передає команди керування до Arduino Nano через роз'єм UART, включаючи інструкції щодо швидкості руху та орієнтації;

– Arduino Nano керує серводвигунами з використанням модуля PCA9685, що забезпечує потрібний рух робота.

Постійна реакція та корекція:

– періодично оновлюйте дані з датчиків та перевіряйте поточний стан робота. Якщо виявлено нові перешкоди або зміни в орієнтації, перегляньте алгоритм планування шляху та внесіть необхідні корективи. Приймайте рішення про зупинку, поворот або зміну шляху залежно від ситуації. Представимо загальний алгоритм управління мобільним роботом в наступному вигляді, як представлено на рисунку 3.11.

В блоці ініціалізації операційної системи Raspberry Pi можуть виконуватися такі дії: завантаження операційної системи з підключених носіїв, налаштування мережевих параметрів (Wi-Fi або Ethernet), активація необхідних інтерфейсів та налаштування з'єднання з підключеними пристроями (наприклад, Arduino Nano) через UART або I2C.

В блоці ініціалізації сервера VNC Viewer буде виконано наступне:

– встановлення та налаштування сервера VNC: встановлення необхідного програмного забезпечення VNC Viewer на Raspberry Pi та налаштування параметрів сервера. Це може включати встановлення паролю доступу, визначення роздільної здатності екрану та вибір відповідних налаштувань безпеки;

– запуск сервера VNC: після налаштування сервера VNC Viewer запускається, щоб встановити з'єднання з VNC-клієнтом і надати можливість дистанційно управляти Raspberry Pi через графічний інтерфейс.

В блоку ініціалізації GPIO (загального призначення вводу/виводу) на Raspberry Pi будуть виконані такі дії:

– ініціалізація GPIO-пінів: здійснюється налаштування та конфігурація GPIO-пінів, які будуть використовуватися для з'єднання зі зовнішніми пристроями чи сенсорами. Це включає встановлення режиму (ввід або вивід), підтяжок (pull-up або pull-down) та інших параметрів;

– підключення до зовнішніх пристроїв: після ініціалізації GPIO-пінів здійснюється підключення до зовнішніх пристроїв чи модулів, таких як сенсори, мотори, світлодіоди тощо. Це забезпечує взаємодію Raspberry Pi з цими пристроями через GPIO-інтерфейс, дозволяючи керувати їхнім станом або отримувати дані з них.

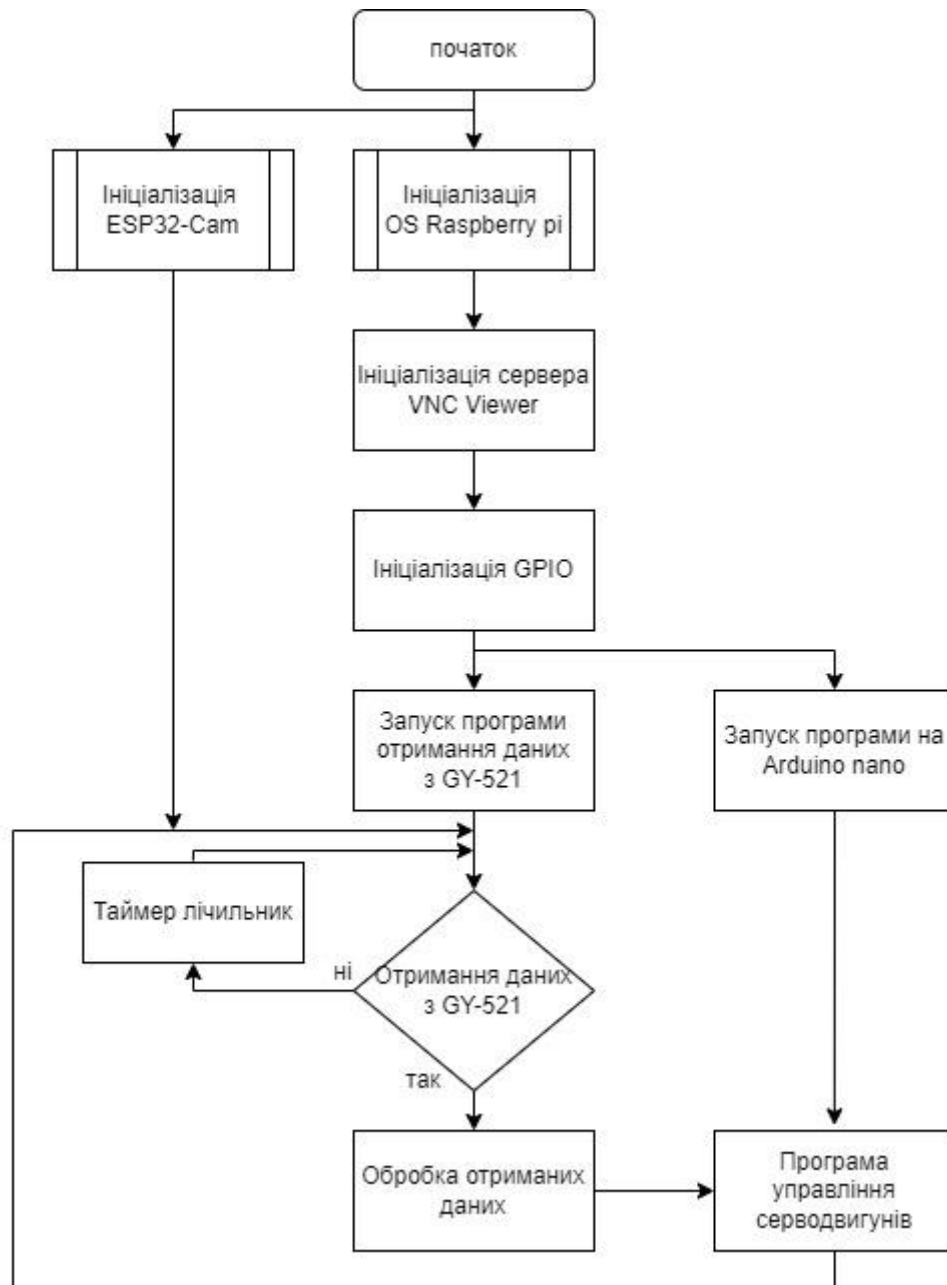


Рисунок 3.11 – Загальний алгоритм управління мобільним роботом

В блоці ініціалізації запуску програми отримання даних з модуля GY-521 (акселерометр та гіроскоп) на Raspberry Pi будуть виконані такі дії:

- підключення до модуля GY-521: здійснюється підключення Raspberry Pi до модуля GY-521 через протокол I2C. Це включає ініціалізацію I2C-інтерфейсу та встановлення необхідних параметрів комунікації;

- запуск програми отримання даних: після успішного підключення виконується запуск програми, яка зчитує дані з модуля GY-521 через I2C. Це дозволяє отримати вимірювання орієнтації та руху, такі як кут нахилу, обертання та прискорення, з модуля GY-521 для подальшого використання в контролі руху робота або в інших задачах.

В блоці запуску програми на Arduino Nano будуть виконані такі кроки:

- завантаження програми на Arduino Nano: компілюється та завантажується програмний код на Arduino Nano з використанням Arduino IDE або іншого середовища розробки. Це дозволяє передати інструкції та функціональність до мікроконтролера Arduino Nano;

- ініціалізація підключених пристроїв: в програмі на Arduino Nano виконується ініціалізація та налаштування підключених пристроїв, таких як ультразвукові датчики HC-SR04 або серводвигуни. Це включає налаштування пінів та інших параметрів, необхідних для правильної роботи цих пристроїв. Після ініціалізації програма готова до взаємодії з підключеними пристроями та виконання відповідних завдань.

В блоці отримання даних з модуля GY-521 (акселерометр та гіроскоп) на Raspberry Pi будуть виконані такі дії:

- конфігурація I2C-з'єднання: здійснюється налаштування I2C-інтерфейсу на Raspberry Pi для забезпечення зв'язку з модулем GY-521. Це включає встановлення необхідних параметрів шини I2C та адреси модуля GY-521;

- зчитування даних з GY-521: використовуючи I2C-інтерфейс, програма на Raspberry Pi зчитує дані з реєстрів модуля GY-521, які містять

вимірювання орієнтації та руху, такі як кут нахилу, обертання та прискорення. Отримані дані можуть бути подальше оброблені або використані для керування роботом чи в інших задачах.

В блоці обробки та отримання даних будуть виконані наступні дії:

- зчитування даних: зчитані дані з модулів чи сенсорів, таких як акселерометр, гіроскоп або ультразвукові датчики, будуть отримані з відповідних пристроїв чи інтерфейсів. Це може включати зчитування значень з реєстрів, аналогових чи цифрових пінів, або виконання команд для отримання потрібних даних;

- обробка даних: отримані дані можуть бути оброблені програмним шляхом для отримання корисних вимірювань, розрахунків або виконання інших алгоритмів. Це може включати фільтрацію, калібрування, розрахунки фізичних величин або аналіз даних для подальшого використання у системі або прийняття рішень.

В блоці програми управління серводвигунів будуть виконані наступні дії:

- ініціалізація серводвигунів: після підключення модуля PCA9685 до Raspberry Pi через протокол I2C, програма на Raspberry Pi виконує ініціалізацію та налаштування серводвигунів, які підключені до модуля PCA9685. Це включає налаштування каналів та параметрів, таких як частота ШІМ-сигналу та діапазон повороту серводвигунів;

- управління серводвигунами: після ініціалізації програма на Raspberry Pi відправляє відповідні сигнали та команди до модуля PCA9685 для управління кутом повороту серводвигунів. Залежно від вхідних значень або команд, програма встановлює потрібний кут повороту для кожного серводвигуна. Це дозволяє точно керувати положенням та рухом механізмів або виконувати задані дії за допомогою серводвигунів, які контролюються через модуль PCA9685.

В блоці ініціалізації ESP32-Cam будуть виконані такі дії:

- налаштування з'єднання Wi-Fi: здійснюється конфігурація модуля ESP32-Cam для підключення до бездротової мережі Wi-Fi. Це включає встановлення SSID (ідентифікатора мережі) та пароля для підключення до мережі, а також налаштування IP-адреси, якщо потрібно;

- ініціалізація камери: після підключення ESP32-Cam до Wi-Fi, виконується ініціалізація камери, що дозволяє отримувати зображення з камери та виконувати відповідні операції з обробки зображень, наприклад, зберігання, трансляцію або відправку зображень по мережі.

3.5 Реалізація функцій керування серводвигунами

Для керування кутами серводвигунів, які були розраховані в розділі 2.1, потрібно розробити програму на базі Arduino nano. Враховуючи, що управління буде здійснюватися через модуль PCA9685, необхідно використовувати бібліотеку `Adafruit_PWMServoDriver.h`. Бібліотека `"Adafruit_PWMServoDriver.h"` надає інструменти для управління серводвигунами за допомогою плати Adafruit PWM Servo Driver. Робота цієї бібліотеки включає наступні кроки:

- ініціалізація: ви починаєте роботу з бібліотекою, включаючи заголовочний файл `"Adafruit_PWMServoDriver.h"` у вашому Arduino-скетчі. Після цього ініціалізується об'єкт класу `Adafruit_PWMServoDriver`, який використовується для взаємодії з платою Adafruit PWM Servo Driver;

- налаштування: після ініціалізації об'єкту ви можете встановити різні параметри, такі як адресу пристрою, частоту PWM, діапазон повороту та інші. Ці налаштування дозволяють забезпечити сумісність з підключеними серводвигунами та налаштувати їхню роботу згідно ваших потреб;

- управління серводвигунами: бібліотека дозволяє встановлювати кути повороту для підключених серводвигунів. Ви можете використовувати метод `setPWM()` для задання кута повороту для конкретного серводвигуна. Також,

ви можете встановити швидкість руху серводвигуна за допомогою методу `setPin()`;

– додаткові функції: бібліотека також надає додаткові функції, такі як зміна частоти PWM, вимкнення всіх серводвигунів або встановлення кута повороту для всіх серводвигунів одночасно.

Також потрібно буде використовувати бібліотеку `Wire.h`. Бібліотека "`Wire.h`" надає функції для зручної роботи з протоколом I2C (TWI) на Arduino. Ця бібліотека дозволяє легко взаємодіяти з пристроями, підключеними до шини I2C, на Arduino. Ви можете передавати та отримувати дані з пристроїв, керувати протоколом I2C та виконувати різноманітні операції з даними.

Внаслідок чого, підключення бібліотек проведемо за допомогою команд `#include`:

```
#include <Wire.h>
```

```
#include <Adafruit_PWMServoDriver.h>
```

Плата Adafruit PWM Servo Driver (PCA9685) має можливість працювати з різними адресами на шині I2C. За замовчуванням, адреса 0x40 використовується для плати Adafruit PWM Servo Driver. Інші адреси, такі як 0x41, 0x42 і т.д., можуть бути використані для підключення додаткових плат до тієї самої шини I2C.

Отже, рядок коду `Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x40)` створює об'єкт `pwm`, який дозволяє взаємодіяти з платою Adafruit PWM Servo Driver, підключеною з адресою 0x40 на шині I2C. Цей об'єкт може використовуватись для керування серводвигунами, підключеними до цієї плати.

```
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x40);
```

Зазвичай серводвигуни керуються за допомогою пульсової модуляції (PWM), де ширина пульсу визначає кут повороту серводвигуна. Мінімальне значення `SERVOMIN` вказує на мінімальну ширину пульсу, яка відповідає

найменшому можливому куту повороту серводвигуна. Аналогічно, максимальне значення `SERVOMAX` вказує на максимальну ширину пульсу, яка відповідає найбільшому можливому куту повороту серводвигуна.

```
#define SERVOMIN 150
#define SERVOMAX 600
#define SERVO_1 0
```

Таке визначення символічної константи дозволяє використовувати ім'я `SERVO_1` замість числового значення `0` в усьому кодї. Це полегшує читання та редагування коду і зробить його більш зрозумілим.

Наприклад, якщо ви використовуєте `SERVO_1` для ідентифікації конкретного серводвигуна, то замість писання числа `0` в кодї, ви можете використовувати `SERVO_1`.

```
#define SERVO_2 1
#define SERVO_3 2
#define SERVO_4 3
#define SERVO_5 4
#define SERVO_6 5
#define SERVO_7 6
#define SERVO_8 7
#define SERVO_9 8
#define SERVO_10 9
#define SERVO_11 10
#define SERVO_12 11
```

Функція `setup()` є спеціальною функцією в `Arduino`, яка викликається один раз при початку виконання програми. У цій функції виконуються налаштування та ініціалізація необхідних компонентів перед основним циклом виконання програми.

У функції `setup()` ви зазвичай виконуєте наступні дії:

– налаштування пінів: встановлення режимів введення або виведення для пінів Arduino, які будуть використовуватися для підключення різних пристроїв або модулів;

– ініціалізація зовнішніх бібліотек: підключення та налаштування зовнішніх бібліотек, які можуть бути використані для спрощення роботи з певними пристроями або функціями;

– конфігурація параметрів: встановлення параметрів, які визначають роботу програми, такі як швидкість передачі даних, режими преривань, налаштування таймерів тощо;

– ініціалізація змінних: встановлення початкових значень для змінних, які будуть використовуватися в програмі.

```
void setup()
{
  pwm.begin();           // Ініціалізація
  pwm.setPWMFreq(60);   // Частота проходження імпульсів 60 Гц
}

void setServoAngle(int servo_pin, int angle)
{
  // Обмежуємо кут повороту в діапазоні 0-180 градусів
  if(angle < 0)
  {
    angle = 0;
  }
  else if(angle > 180)
  {
    angle = 180;
  }

  // Перетворимо кут на відповідне значення ширини імпульсу
```

```
int pulse_width = map(angle, 0, 180, SERVOMIN, SERVOMAX);

// Встановлюємо ширину імпульсу для вказаного серводвигуна
pwm.setPWM(servo_pin, 0, pulse_width);
}
```

Функція `loop()` є основним циклом виконання програми Arduino і викликається безперервно після виконання функції `setup()`. У цій функції виконуються дії, які повторюються відразу після їх виконання до завершення програми або до зупинки Arduino.

```
void loop()
{
    pwm.setPWM(0, 0, 160); // Встановити перший серводвигун у
положення 0 градусів
    delay(450);           // Затримка
    pwm.setPWM(0, 0, 590); // Встановити перший серводвигун у
положення 180 градусів
    delay(450);           // Затримка
}
```

Приклад реалізації команди «увімкнення» для зооморфного мобільного робота передньої правої ноги.

```
void power_on()
{
    setServoAngle(servo1_pin, 70); // серводвигуни 1;
    setServoAngle(servo2_pin, 80); // серводвигуни 2;
    setServoAngle(servo3_pin, 45); // серводвигуни 3;
```

Функція `power_on()` виконує послідовну активацію трьох серводвигунів за допомогою виклику функції `setServoAngle()` з відповідними параметрами.

– `setServoAngle(servo1_pin, 70)` встановлює кут повороту першого серводвигуна (підключеного до піна `servo1_pin`) на 70 градусів;

- `setServoAngle(servo2_pin, 80)` встановлює кут повороту другого серводвигуна (підключеного до піна `servo2_pin`) на 80 градусів;
- `setServoAngle(servo3_pin, 45)` встановлює кут повороту третього серводвигуна (підключеного до піна `servo3_pin`) на 45 градусів.

Ці дії дозволяють ініціалізувати певну початкову позицію серводвигунів під час увімкнення робота або старту програми. Залежно від потреби, ви можете змінити значення кутів повороту для відповідного налаштування серводвигунів у функції `power_on()`.

3.6 Висновки до 3 розділу

В ході виконання третього розділу досліджень з розробки системи керування мобільним зооморфним роботом типу Spot, було зроблено:

- проведено аналіз різних апаратних модулів, які можуть бути використані для реалізації системи керування мобільним роботом. Після докладного вивчення можливостей кожного модуля та їх відповідності вимогам, були зроблені висновки щодо вибору найбільш підходящих апаратних модулів для подальшої реалізації системи керування;
- розроблено структурну схему системи керування мобільним роботом. Були визначені основні компоненти системи та їх взаємозв'язки. Результатом цього розділу є структурна схема, яка відображає логічну організацію системи керування;
- розроблена схема підключення апаратних модулів до мобільного робота. Було визначено необхідні з'єднання між модулями, включаючи роз'єми, проводи та інтерфейси комунікації. Ця схема служить основою для правильного підключення та функціонування апаратури в системі керування;
- зібрана система керування та встановлена на мобільного робота, що дозволить розробити алгоритми керування та провести розробку програмного забезпечення;

– розроблено загальний алгоритм керування зооморфними мобільними роботами, та написані процедури керування кутами серводвигунів на кінцівки мобільного роботу.

Все це дозволяє приступити до розробки системи керування серводвигунами мобільного робота з урахуванням кутів похилення, які будуть виходити з модуля акселерометра.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Постановка задач експериментів

У даному експерименті використовується зооморфний мобільний робот типу Spot, який керується за допомогою системи на базі Raspberry Pi. До робота підключений модуль GY-521, який включає в себе акселерометр та гіроскоп і спілкується з Raspberry Pi за допомогою I2C-з'єднання. За допомогою написаної на мові Python програми, Raspberry Pi отримує дані від модуля GY-521. Ці дані включають значення G_x , Gyro X-axis (кутова швидкість по осі X в градусах за секунду), Gyro Y-axis (кутова швидкість по осі Y в градусах за секунду), Gyro Z-axis (кутова швидкість по осі Z в градусах за секунду), A_x , Accelerometer X-axis (прискорення по осі X в одиницях g), Accelerometer Y-axis (прискорення по осі Y в одиницях g), Accelerometer Z-axis (прискорення по осі Z в одиницях g).

Датчик GY-521 – це пристрій, який поєднує в собі гіроскоп та акселерометр і дозволяє визначати орієнтацію та рух об'єкта у просторі. Для мобільних роботів такий датчик може бути дуже корисним, оскільки він може забезпечити інформацію про поточне положення та рух робота.

Застосування датчика GY-521 може допомогти мобільному роботу в наступних задачах:

- навігація: дозволяє визначити поточне положення та орієнтацію робота в просторі, що може бути корисним при навігації в приміщенні або на вулиці;
- керування рухом: дозволяє коректно реагувати на зміни руху та орієнтації, що може бути важливим при управлінні рухом робота;

– стабілізація: дає можливість зберігати стійкість і стабільність під час руху, що може бути важливим при виконанні завдань, що потребують точності та надійності.

Ці дані можуть використовуватись для подальшого аналізу, навігації або реакції робота на зміни в орієнтації та руху. Схема підключення модуля GY-521 до Raspberry Pi приведено на рисунку 4.1.

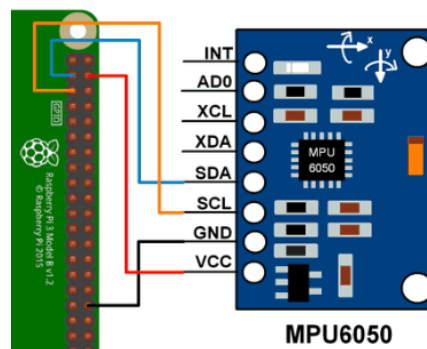


Рисунок 4.1 – Схема підключення модуля GY-521 до Raspberry Pi для проведення експерименту

Макет для отримання даних з датчика GY-521, представлено на рисунку 4.2.

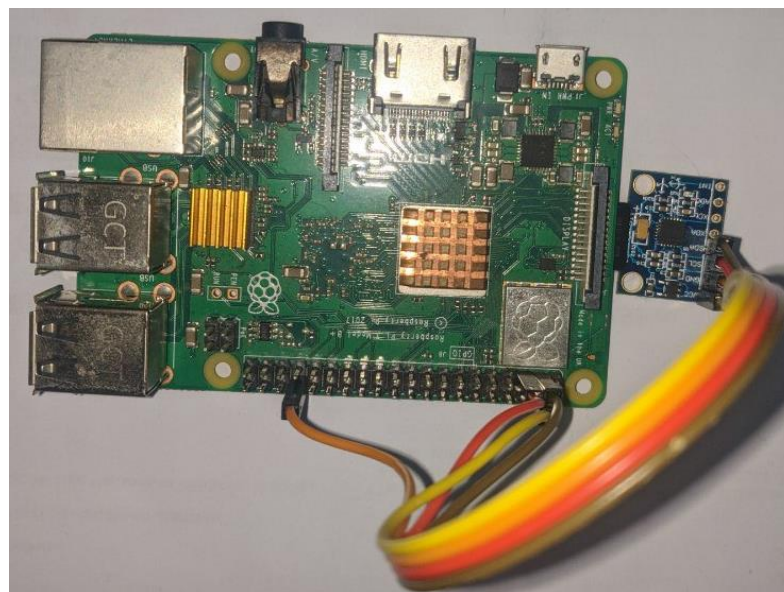


Рисунок 4.2 – Розроблений макет для отримання даних з датчика GY-521

Для отримання даних з датчика GY-521 розробимо програму на мові Python 3, в галузі середовища розробки буде використовуватись середовище розробки інтегрована в Raspberry Pi OS Tonny IDE. Листинг програми для отримання даних з датчика GY-521 приведено ніжче з описом всіх команд. На першому кроці проведемо підключення бібліотек:

```
import smbus
from time import sleep
```

Команда `import smbus` використовується для імпорту модуля `smbus`, який надає функціональність для взаємодії з пристроями через I2C-з'єднання. Цей модуль дозволяє зчитувати та записувати дані на пристрої, підключеному до Raspberry Pi за допомогою I2C-протоколу.

Команда `from time import sleep` використовується для імпорту функції `sleep` з модуля `time`. Ця функція використовується для затримки виконання програми на певний проміжок часу. У випадку отримання даних в режимі зчитування з GY-521, затримки можуть бути використані для стабілізації та отримання стабільних значень даних перед їх обробкою.

```
PWR_MGMT_1 = 0x6B
SMPLRT_DIV = 0x19
CONFIG = 0x1A
GYRO_CONFIG = 0x1B
INT_ENABLE = 0x38
ACCEL_XOUT_H = 0x3B
ACCEL_YOUT_H = 0x3D
ACCEL_ZOUT_H = 0x3F
GYRO_XOUT_H = 0x43
GYRO_YOUT_H = 0x45
GYRO_ZOUT_H = 0x47
```

В цих рядках представлені шістнадцяткові адреси реєстрів, які використовуються для зчитування даних з модуля GY-521 (акселерометр та

гіроскоп) за допомогою I2C-з'єднання. Кожна з цих адрес відповідає певному реєстру, з якого можна отримати певні дані. Ось їх опис:

- PWR_MGMT_1 (0x6B): реєстр керування живленням модуля GY-521;
- SMPLRT_DIV (0x19): реєстр, що встановлює дільник частоти дискретизації для гіроскопа та акселерометра;
- CONFIG (0x1A): реєстр конфігурації фільтрації даних;
- GYRO_CONFIG (0x1B): реєстр конфігурації гіроскопа;
- INT_ENABLE (0x38): реєстр дозволу преривань;
- ACCEL_XOUT_H (0x3B): реєстр високого байту для зчитування даних по осі X акселерометра;
- ACCEL_YOUT_H (0x3D): реєстр високого байту для зчитування даних по осі Y акселерометра;
- ACCEL_ZOUT_H (0x3F): реєстр високого байту для зчитування даних по осі Z акселерометра;
- GYRO_XOUT_H (0x43): реєстр високого байту для зчитування даних по осі X гіроскопа;
- GYRO_YOUT_H (0x45): реєстр високого байту для зчитування даних по осі Y гіроскопа;
- GYRO_ZOUT_H (0x47): реєстр високого байту для зчитування даних по осі Z гіроскопа.

Ці адреси використовуються для звернення до конкретних реєстрів у модулі GY-521 та отримання значень даних, які можуть бути використані для подальшого аналізу та обробки.

```
def MPU_Init():
    #write to sample rate register
    bus.write_byte_data(Device_Address, SMPLRT_DIV, 7)

    #Write to power management register
```

```
bus.write_byte_data(Device_Address, PWR_MGMT_1, 1)
```

```
#Write to Configuration register
```

```
bus.write_byte_data(Device_Address, CONFIG, 0)
```

```
#Write to Gyro configuration register
```

```
bus.write_byte_data(Device_Address, GYRO_CONFIG, 24)
```

```
#Write to interrupt enable register
```

```
bus.write_byte_data(Device_Address, INT_ENABLE, 1)
```

У даній функції MPU_Init() виконується ініціалізація модуля GY-521 (акселерометр та гіроскоп) шляхом запису значень до відповідних реєстрів за допомогою функції bus.write_byte_data(). Ось опис кожного рядка:

- bus.write_byte_data(Device_Address, SMPLRT_DIV, 7): встановлюється швидкість дискретизації даних з акселерометра та гіроскопа на значення 7;

- bus.write_byte_data(Device_Address, PWR_MGMT_1, 1): встановлюється режим керування живленням модуля GY-521 на значення 1;

- bus.write_byte_data(Device_Address, CONFIG, 0): встановлюється конфігурація фільтрації даних на значення 0;

- bus.write_byte_data(Device_Address, GYRO_CONFIG, 24): встановлюється конфігурація гіроскопа на значення 24;

- bus.write_byte_data(Device_Address, INT_ENABLE, 1): встановлюється дозвіл на преривання на значення 1.

Ці команди записують відповідні значення в реєстри модуля GY-521, що дозволяє налаштувати його роботу перед початком отримання даних.

```
def read_raw_data(addr):
```

```
    #Accelerometer and Gyro value are 16-bit
```

```
    high = bus.read_byte_data(Device_Address, addr)
```

```
    low = bus.read_byte_data(Device_Address, addr+1)
```

```

#concatenate higher and lower value
value = ((high << 8) | low)

#to get signed value from mpu6050
if(value > 32768):
    value = value - 65536
return value

```

У даній функції `read_raw_data(addr)` зчитуються сирі дані з вказаної адреси `addr` модуля GY-521 (акселерометр та гіроскоп) за допомогою функції `bus.read_byte_data()`. Ось опис кожного рядка:

- `high = bus.read_byte_data(Device_Address, addr)`: зчитується значення високого байта з вказаної адреси `addr`;
- `low = bus.read_byte_data(Device_Address, addr+1)`: зчитується значення низького байта з наступної адреси `addr+1`;
- `value = ((high << 8) | low)`: об'єднується високий і низький байти, утворюючи 16-бітне значення;
- `if(value > 32768): value = value - 65536`: перевіряється, чи значення перевищує 32768, якщо так, то виконується конвертація у від'ємне число;
- `return value`: повертається отримане значення.

Ці кроки дозволяють зчитати сирі значення з вказаної адреси модуля GY-521 та скоректувати їх для отримання правильних значень від акселерометра та гіроскопа.

```

bus = smbus.SMBus(1) # or bus = smbus.SMBus(0) for older version
boards

Device_Address = 0x68 # MPU6050 device address

```

У цьому коді імпортується бібліотека `smbus`, яка надає функціональність для спілкування з пристроями через I2C на Raspberry Pi. Ось опис кожного рядка:

- `bus = smbus.SMBus(1)`: створюється об'єкт `bus`, який представляє з'єднання з шиною I2C. У цьому випадку використовується номер шини I2C 1 (1), але на старіших версіях плати Raspberry Pi може використовуватись номер шини I2C 0 (0);

- `Device_Address = 0x68`: встановлюється адреса пристрою MPU6050 (GY-521) на шині I2C. В даному випадку використовується адреса 0x68.

Ці кроки дозволяють налаштувати з'єднання з шиною I2C та встановити адресу пристрою GY-521 для подальшої комунікації з ним.

MPU_Init()

Функція `MPU_Init()` виконує ініціалізацію модуля MPU6050 (GY-521) для правильної роботи і отримання даних з акселерометра та гіроскопа. Ось опис кожного рядка:

- `bus.write_byte_data(Device_Address, SMPLRT_DIV, 7)`: записується значення 7 в регістр `SMPLRT_DIV`, що відповідає за швидкість дискретизації даних;

- `bus.write_byte_data(Device_Address, PWR_MGMT_1, 1)`: записується значення 1 в регістр `PWR_MGMT_1`, що вмикає модуль GY-521 і встановлює його в режим роботи;

- `bus.write_byte_data(Device_Address, CONFIG, 0)`: записується значення 0 в регістр `CONFIG`, що налаштовує фільтрацію і режим роботи модуля;

- `bus.write_byte_data(Device_Address, GYRO_CONFIG, 24)`: записується значення 24 в регістр `GYRO_CONFIG`, що встановлює діапазон індикаторів гіроскопа;

– `bus.write_byte_data(Device_Address, INT_ENABLE, 1)`: записується значення 1 в регістр `INT_ENABLE`, що вмикає можливість виклику переривань.

Ці кроки встановлюють необхідні налаштування модуля GY-521 для отримання коректних даних з акселерометра та гіроскопа.

```
print (" Reading Data of Gyroscope and Accelerometer")
```

```
while True:
```

```
#Read Accelerometer raw value
```

```
acc_x = read_raw_data(ACCEL_XOUT_H)
```

```
acc_y = read_raw_data(ACCEL_YOUT_H)
```

```
acc_z = read_raw_data(ACCEL_ZOUT_H)
```

```
#Read Gyroscope raw value
```

```
gyro_x = read_raw_data(GYRO_XOUT_H)
```

```
gyro_y = read_raw_data(GYRO_YOUT_H)
```

```
gyro_z = read_raw_data(GYRO_ZOUT_H)
```

```
#Full scale range +/- 250 degree/C as per sensitivity scale factor
```

```
Ax = acc_x/16384.0
```

```
Ay = acc_y/16384.0
```

```
Az = acc_z/16384.0
```

```
Gx = gyro_x/131.0
```

```
Gy = gyro_y/131.0
```

```
Gz = gyro_z/131.0
```

```
print ("Gx=%.2f" %Gx, u'\u00b0'+ "/s", "\tGy=%.2f" %Gy, u'\u00b0'+ "/s",
"\tGz=%.2f" %Gz, u'\u00b0'+ "/s", "\tAx=%.2f g" %Ax, "\tAy=%.2f g" %Ay,
"\tAz=%.2f g" %Az)
```

Цей рядок виводить значення отриманих даних з акселерометра та гіроскопа. Ось опис кожного параметра:

- "Gx=%.2f" %Gx: виводить значення Gx (швидкість гіроскопа по вісі X) з двома знаками після коми, вказуючи одиницю виміру "градус/сек";

- "Gy=%.2f" %Gy: виводить значення Gy (швидкість гіроскопа по вісі Y) з двома знаками після коми, вказуючи одиницю виміру "градус/сек";

- "Gz=%.2f" %Gz: виводить значення Gz (швидкість гіроскопа по вісі Z) з двома знаками після коми, вказуючи одиницю виміру "градус/сек";

- "Ax=%.2f g" %Ax: виводить значення Ax (прискорення по вісі X) з двома знаками після коми, вказуючи одиницю виміру "g" (гравітаційне прискорення);

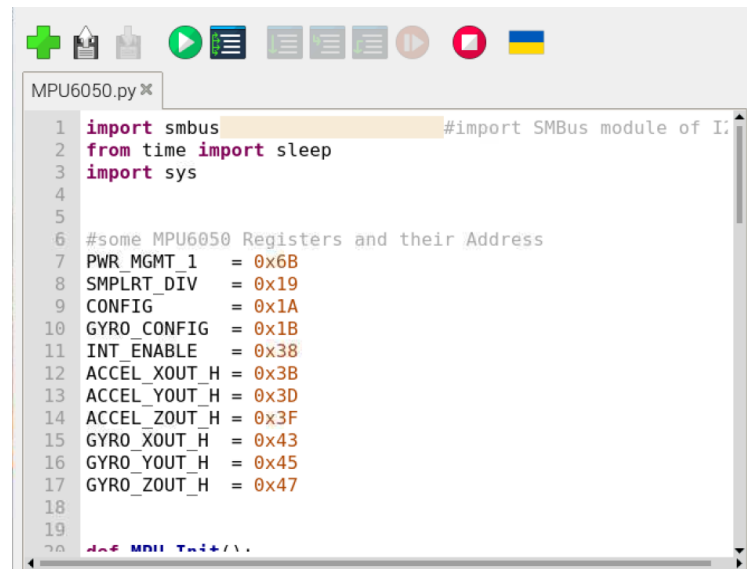
- "Ay=%.2f g" %Ay: виводить значення Ay (прискорення по вісі Y) з двома знаками після коми, вказуючи одиницю виміру "g" (гравітаційне прискорення);

- "Az=%.2f g" %Az: виводить значення Az (прискорення по вісі Z) з двома знаками після коми, вказуючи одиницю виміру "g" (гравітаційне прискорення).

Цей рядок форматує вивід, щоб відображати значення кожного параметра з відповідними одиницями виміру та з вказаною точністю до двох знаків після коми.

```
sleep(1)
```

Фрагмент програмного коду отримання даних з модуля GY-521 в середовищі розробки Tonny IDE представлено на рисунку 4.3.



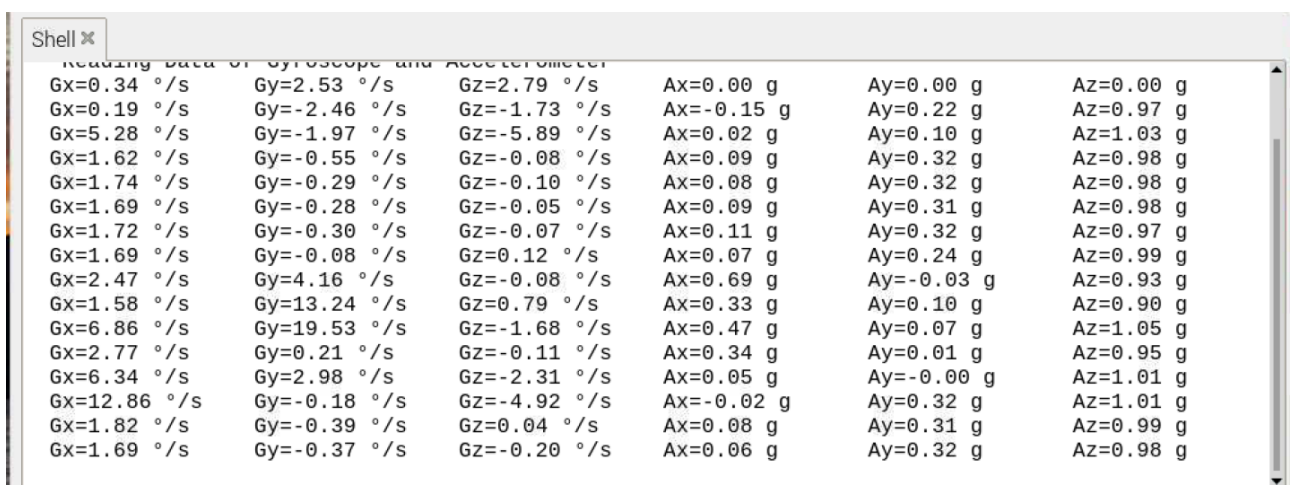
```

MPU6050.py x
1 import smbus #import SMBus module of I
2 from time import sleep
3 import sys
4
5
6 #some MPU6050 Registers and their Address
7 PWR_MGMT_1 = 0x6B
8 SMPLRT_DIV = 0x19
9 CONFIG = 0x1A
10 GYRO_CONFIG = 0x1B
11 INT_ENABLE = 0x38
12 ACCEL_XOUT_H = 0x3B
13 ACCEL_YOUT_H = 0x3D
14 ACCEL_ZOUT_H = 0x3F
15 GYRO_XOUT_H = 0x43
16 GYRO_YOUT_H = 0x45
17 GYRO_ZOUT_H = 0x47
18
19
20 def MPU_Init():

```

Рисунок 4.3 – Фрагмент програмного коду отримання даних з модуля GY-521 в середовищі розробки Tonny IDE

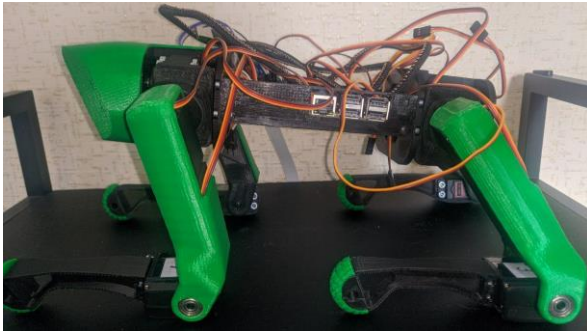
Після компіляції програми в Shell середовища Tonny IDE будуть виводитися такі дані: Gx = Gyro X-axis data in degree/seconds; Gy = Gyro Y-axis data in degree/seconds; Gz = Gyro Z-axis data in degree/seconds; Ax = Accelerometer X-axis data in g; Ay = Accelerometer Y-axis data in g; Az = Accelerometer Z-axis data in g як реального часу. Приклад отримання даних з модуля GY-521 в Shell середовища Tonny IDE представлено на рисунку 4.4.



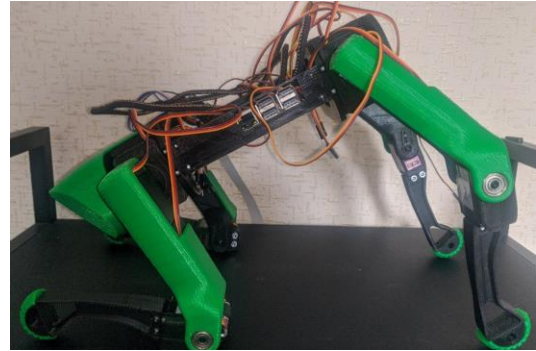
Gx	Gy	Gz	Ax	Ay	Az
Gx=0.34 °/s	Gy=2.53 °/s	Gz=2.79 °/s	Ax=0.00 g	Ay=0.00 g	Az=0.00 g
Gx=0.19 °/s	Gy=-2.46 °/s	Gz=-1.73 °/s	Ax=-0.15 g	Ay=0.22 g	Az=0.97 g
Gx=5.28 °/s	Gy=-1.97 °/s	Gz=-5.89 °/s	Ax=0.02 g	Ay=0.10 g	Az=1.03 g
Gx=1.62 °/s	Gy=-0.55 °/s	Gz=-0.08 °/s	Ax=0.09 g	Ay=0.32 g	Az=0.98 g
Gx=1.74 °/s	Gy=-0.29 °/s	Gz=-0.10 °/s	Ax=0.08 g	Ay=0.32 g	Az=0.98 g
Gx=1.69 °/s	Gy=-0.28 °/s	Gz=-0.05 °/s	Ax=0.09 g	Ay=0.31 g	Az=0.98 g
Gx=1.72 °/s	Gy=-0.30 °/s	Gz=-0.07 °/s	Ax=0.11 g	Ay=0.32 g	Az=0.97 g
Gx=1.69 °/s	Gy=-0.08 °/s	Gz=0.12 °/s	Ax=0.07 g	Ay=0.24 g	Az=0.99 g
Gx=2.47 °/s	Gy=4.16 °/s	Gz=-0.08 °/s	Ax=0.69 g	Ay=-0.03 g	Az=0.93 g
Gx=1.58 °/s	Gy=13.24 °/s	Gz=0.79 °/s	Ax=0.33 g	Ay=0.10 g	Az=0.90 g
Gx=6.86 °/s	Gy=19.53 °/s	Gz=-1.68 °/s	Ax=0.47 g	Ay=0.07 g	Az=1.05 g
Gx=2.77 °/s	Gy=0.21 °/s	Gz=-0.11 °/s	Ax=0.34 g	Ay=0.01 g	Az=0.95 g
Gx=6.34 °/s	Gy=2.98 °/s	Gz=-2.31 °/s	Ax=0.05 g	Ay=-0.00 g	Az=1.01 g
Gx=12.86 °/s	Gy=-0.18 °/s	Gz=-4.92 °/s	Ax=-0.02 g	Ay=0.32 g	Az=1.01 g
Gx=1.82 °/s	Gy=-0.39 °/s	Gz=0.04 °/s	Ax=0.08 g	Ay=0.31 g	Az=0.99 g
Gx=1.69 °/s	Gy=-0.37 °/s	Gz=-0.20 °/s	Ax=0.06 g	Ay=0.32 g	Az=0.98 g

Рисунок 4.4 – Приклад отримання даних з модуля GY-521 в Shell середовища Tonny IDE

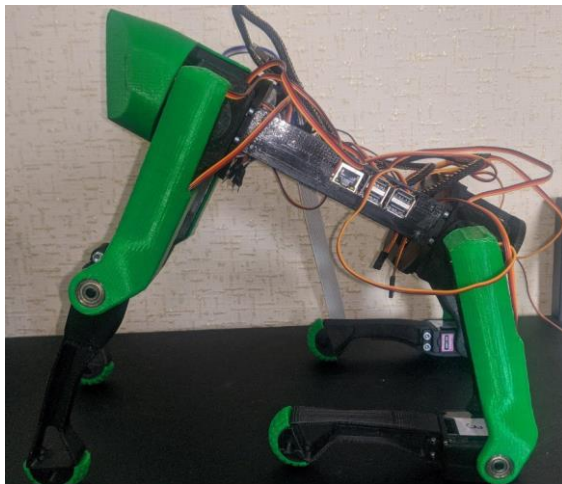
На рисунку 4.5 приведено фото змінні положення мобільного робота в просторі, для отримання експериментальних даних з модуля GY-521. Експеримент проводився в «динамічному режимі», що означає, що дані поступають безперервно.



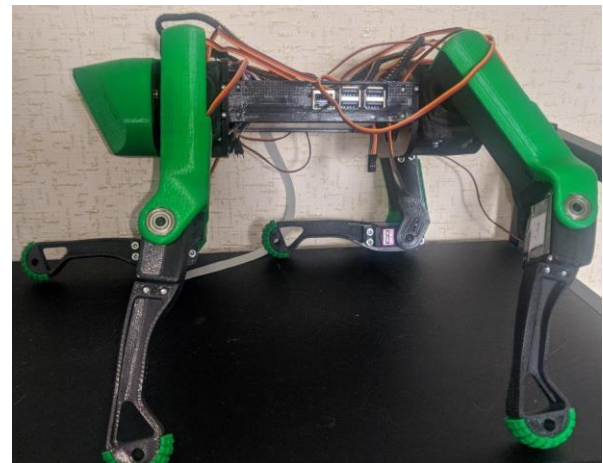
а)



б)



в)



г)

а) положення 1; б) положення 2; в) положення 3; г) положення 4

Рисунок 4.5 – Фотографії проведення експерименту з модулем GY-521

Як можна бачити з рисунку 4.5 «положення» мобільного роботу показує можливі варіанти команд, які може виконувати розроблена система керування. Проведемо ряд екскрементів для отримання даних з модуля GY-521, а отримані результати представимо у вигляді таблиці 4.1.

Таблиця 4.1 – Отримані дані з модуля GY-521 в результаті проведених експериментів

дані осі гіроскопа в градусах/секундах			дані акселерометра в G		
Gx = дані осі X	Gy = дані осі Y	Gz = дані осі Z	Ax по осі X	Ay по осі Y	Az осі Z
1	2	3	4	5	6
Gx=5.21 °/s	Gy=-5.72 °/s	Gz=-3.29 °/s	Ax=0.32 g	Ay=0.01 g	Az=1.02 g
Gx=11.05 °/s	Gy=-0.02 °/s	Gz=0.92 °/s	Ax=-0.17 g	Ay=-0.30 g	Az=1.13 g
Gx=-12.70 °/s	Gy=12.39 °/s	Gz=1.05 °/s	Ax=0.02 g	Ay=-0.12 g	Az=1.02 g
Gx=9.36 °/s	Gy=-18.98 °/s	Gz=10.02 °/s	Ax=-1.18 g	Ay=-0.33 g	Az=0.60 g
Gx=8.14 °/s	Gy=-16.79 °/s	Gz=1.40 °/s	Ax=0.13 g	Ay=0.19 g	Az=0.91 g
Gx=-20.48 °/s	Gy=13.31 °/s	Gz=-0.59 °/s	Ax=0.41 g	Ay=0.46 g	Az=0.97 g
Gx=38.64 °/s	Gy=-14.79 °/s	Gz=-14.52 °/s	Ax=-0.39 g	Ay=-0.47 g	Az=0.45 g
Gx=-3.60 °/s	Gy=21.65 °/s	Gz=-5.62 °/s	Ax=0.53 g	Ay=0.40 g	Az=0.95 g
Gx=-8.81 °/s	Gy=-11.47 °/s	Gz=17.27 °/s	Ax=-0.28 g	Ay=0.10 g	Az=1.00 g
Gx=7.34 °/s	Gy=-22.40 °/s	Gz=-32.97 °/s	Ax=-1.09 g	Ay=1.51 g	Az=0.50 g
Gx=3.63 °/s	Gy=4.11 °/s	Gz=4.02 °/s	Ax=-0.76 g	Ay=0.34 g	Az=0.79 g
Gx=-13.11 °/s	Gy=8.82 °/s	Gz=-0.94 °/s	Ax=-0.23 g	Ay=-0.01 g	Az=0.88 g
Gx=1.82 °/s	Gy=0.06 °/s	Gz=-0.96 °/s	Ax=-0.17 g	Ay=0.14 g	Az=1.05 g
Gx=-6.54 °/s	Gy=-1.23 °/s	Gz=23.98 °/s	Ax=0.17 g	Ay=-0.39 g	Az=1.11 g
Gx=1.94 °/s	Gy=-1.08 °/s	Gz=-0.11 °/s	Ax=0.68 g	Ay=-0.70 g	Az=-0.31 g
Gx=4.75 °/s	Gy=2.49 °/s	Gz=0.57 °/s	Ax=0.75 g	Ay=-0.74 g	Az=-0.22 g
Gx=2.63 °/s	Gy=16.15 °/s	Gz=-2.43 °/s	Ax=0.00 g	Ay=0.13 g	Az=1.04 g

Продовження таблиці 4.1

1	2	3	4	5	6
Gx=2.08 °/s	Gy=-8.50 °/s	Gz=0.58 °/s	Ax=-1.02 g	Ay=-0.22 g	Az=0.36 g
Gx=0.90 °/s	Gy=-0.81 °/s	Gz=0.35 °/s	Ax=-0.63 g	Ay=-0.08 g	Az=0.03 g
Gx=-5.10 °/s	Gy=-12.60 °/s	Gz=6.45 °/s	Ax=0.02 g	Ay=-0.28 g	Az=0.65 g
Gx=13.43 °/s	Gy=-10.41 °/s	Gz=9.25 °/s	Ax=0.06 g	Ay=0.30 g	Az=1.21 g
Gx=1.03 °/s	Gy=0.44 °/s	Gz=-0.42 °/s	Ax=0.64 g	Ay=0.67 g	Az=0.47 g
Gx=-12.13 °/s	Gy=-1.81 °/s	Gz=-3.21 °/s	Ax=0.22 g	Ay=0.48 g	Az=0.93 g
Gx=-4.19 °/s	Gy=-0.28 °/s	Gz=-5.37 °/s	Ax=-0.29 g	Ay=-0.81 g	Az=0.74 g
Gx=2.27 °/s	Gy=-0.26 °/s	Gz=-0.74 °/s	Ax=-0.14 g	Ay=-0.78 g	Az=0.69 g
Gx=6.52 °/s	Gy=-0.24 °/s	Gz=-3.58 °/s	Ax=-0.02 g	Ay=0.11 g	Az=1.04 g
Gx=0.86 °/s	Gy=0.22 °/s	Gz=0.21 °/s	Ax=0.53 g	Ay=0.56 g	Az=0.72 g
Gx=-4.79 °/s	Gy=10.92 °/s	Gz=0.91 °/s	Ax=0.07 g	Ay=0.26 g	Az=0.99 g
Gx=0.79 °/s	Gy=1.47 °/s	Gz=-2.27 °/s	Ax=0.59 g	Ay=0.42 g	Az=0.82 g
Gx=2.21 °/s	Gy=0.36 °/s	Gz=-0.24 °/s	Ax=0.52 g	Ay=0.34 g	Az=0.84 g
Gx=0.93 °/s	Gy=-1.08 °/s	Gz=1.73 °/s	Ax=0.04 g	Ay=0.13 g	Az=1.10 g
Gx=2.49 °/s	Gy=-0.43 °/s	Gz=-0.56 °/s	Ax=-0.30 g	Ay=-0.73 g	Az=0.65 g
Gx=11.03 °/s	Gy=2.08 °/s	Gz=-0.23 °/s	Ax=-0.21 g	Ay=0.07 g	Az=0.96 g
Gx=8.41 °/s	Gy=16.01 °/s	Gz=-5.98 °/s	Ax=-0.16 g	Ay=-0.10 g	Az=0.85 g
Gx=1.98 °/s	Gy=-1.43 °/s	Gz=0.31 °/s	Ax=-0.70 g	Ay=0.02 g	Az=0.57 g
Gx=4.60 °/s	Gy=-8.84 °/s	Gz=-5.21 °/s	Ax=-0.11 g	Ay=0.26 g	Az=1.31 g
Gx=1.44 °/s	Gy=0.79 °/s	Gz=-0.24 °/s	Ax=0.02 g	Ay=0.04 g	Az=1.07 g

Продовження таблиці 4.1

1	2	3	4	5	6
Gx=1.46 °/s	Gy=-1.67 °/s	Gz=-0.26 °/s	Ax=0.04 g	Ay=0.01 g	Az=1.04 g
Gx=-2.56 °/s	Gy=0.93 °/s	Gz=0.69 °/s	Ax=-0.05 g	Ay=0.40 g	Az=1.25 g
Gx=2.24 °/s	Gy=-1.39 °/s	Gz=-4.41 °/s	Ax=0.01 g	Ay=-0.19 g	Az=1.03 g
Gx=1.57 °/s	Gy=-1.27 °/s	Gz=0.60 °/s	Ax=-0.01 g	Ay=0.09 g	Az=1.04 g
Gx=1.91 °/s	Gy=-0.13 °/s	Gz=-0.24 °/s	Ax=-0.03 g	Ay=0.09 g	Az=1.01 g
Gx=1.69 °/s	Gy=-0.33 °/s	Gz=-0.12 °/s	Ax=-0.03 g	Ay=0.10 g	Az=1.03 g
Gx=1.82 °/s	Gy=-0.31 °/s	Gz=-0.06 °/s	Ax=-0.02 g	Ay=0.11 g	Az=1.02 g
Gx=1.81 °/s	Gy=-0.21 °/s	Gz=-0.01 °/s	Ax=-0.04 g	Ay=0.11 g	Az=1.02 g
Gx=1.69 °/s	Gy=-0.31 °/s	Gz=0.05 °/s	Ax=-0.03 g	Ay=0.11 g	Az=1.04 g
Gx=1.72 °/s	Gy=-0.31 °/s	Gz=-0.12 °/s	Ax=-0.05 g	Ay=0.11 g	Az=1.02 g
Gx=1.72 °/s	Gy=-0.39 °/s	Gz=-0.14 °/s	Ax=-0.03 g	Ay=0.10 g	Az=1.03 g

4.2 Аналіз отриманнях результатів

Для аналізу та візуалізації отриманих результатів з модуля GY-521, будемо використовувати бібліотеку plt для мови програмування Python. Бібліотека plt в мові програмування Python зазвичай використовується для візуалізації даних за допомогою графіків. Основна бібліотека для цього – matplotlib.pyplot, яка зазвичай імпортується як plt. Вона надає широкі можливості для створення різних типів графіків, діаграм, гістограм, розсіювальних графіків та інших візуалізаційних елементів. В наслідок цього обрана для обробки даних та їх візуалізації ця бібліотека. Середовищем розробки обрана PyCharm 2022.2.3. Напишемо програму для візуалізації

ориманих даних з модуля GY-521 для даних з осі гіроскопа в градусах/секундах.

Для цього будемо використовувати наступні бібліотеки:

– `matplotlib.pyplot` – це основна бібліотека для візуалізації даних у мові програмування Python. Вона надає функціонал для створення різних типів графіків та діаграм, роботи з кольорами, підписами осей та іншими елементами візуалізації;

– `numpy` – це бібліотека для наукових обчислень у мові програмування Python. Вона надає підтримку для масивів, матриць та чисельних обчислень. Використання `numpy` може спростити обробку та маніпуляцію даними перед візуалізацією.

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

Введемо отримані дані з таблиці 4.1 (дані осі гіроскопа в градусах/секундах)

```
Ax = [5.21, 11.05, 12.70, 9.36, 8.14, -20.48, 38.64, 3.60, 8.81, 7.34, 3.63,
13.11, 1.82, -6.54, 1.94, 4.75, 2.63, 2.08, 0.90, -5.10, 13.43, 1.03, -12.13, -4.19,
2.27, 6.52, 0.86, -4.79, 0.93, 2.49, 11.03, 8.41, 1.98, 4.60, 1.44, 1.46, -2.56, 2.24,
1.57, 1.91, 1.69, 1.82, 1.81, 1.69, 1.72, 1.72]
```

```
Ay = [-5.72, -0.02, 12.39, -18.98, -16.79, 13.31, -14.79, 21.65, -11.47, -
22.40, 4.11, 8.82, 0.06, -1.23, -1.08, 2.49, 16.15, -8.50, -0.81, -12.60, -10.41, 0.44,
-1.81, -0.28, -0.26, -0.24, 0.22, 10.92, 1.47, 0.36, -1.08, -0.43, 2.08, 16.01, 1.43, -
8.84, 0.79, -1.67, 0.93, 1.39, -1.27, -0.13, -0.33, -0.31, -0.21, -0.31]
```

```
Az = [-3.29, 0.92, 1.05, 10.02, 1.40, -0.59, -14.52, -5.62, 17.27, -32.97, 4.02,
-0.94, -0.96, 23.98, -0.11, 0.57, -2.43, 0.58, 0.35, 6.45, 9.25, -0.42, -3.21, -5.37, -
0.74, -3.58, 0.21, 0.91, -2.27, -0.24, 1.73, -0.56, -0.23, -5.98, 0.31, -5.21, -0.24, -
0.26, 0.69, 4.41, 0.60, -0.24, -0.12, -0.06, -0.01, 0.05]
```

```
plt.title ('MPU6050')  
plt.xlabel('The number of dimensions')  
plt.ylabel('Gyro X-axis data in degree/seconds')  
plt.plot(Ax)  
plt.grid(True)
```

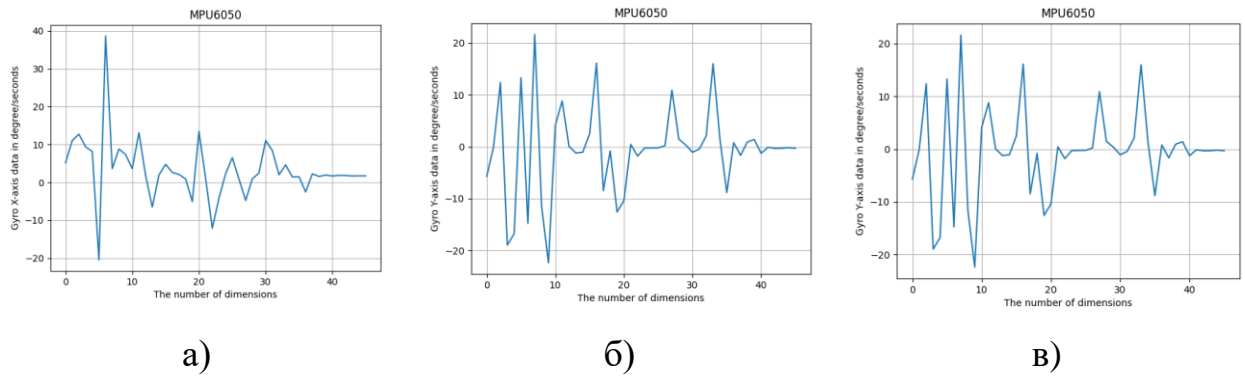
Цей код використовує функції та методи з бібліотеки `matplotlib.pyplot` для створення графіку:

- `plt.title('MPU6050')` – встановлює заголовок графіку на "MPU6050";
- `plt.xlabel('The number of dimensions')` – встановлює підпис осі x на "The number of dimensions";
- `plt.ylabel('Gyro X-axis data in degree/seconds')` – встановлює підпис осі y на "Gyro X-axis data in degree/seconds";
- `plt.plot(Ax)` – створює лінійний графік з даних Ax;
- `plt.grid(True)` – вмикає сітку на графіку.

```
plt.show()
```

Команда `plt.show()` для відображення графіку.

В наслідок чого отримуємо графіки залежності отриманих даних з модуля GY-521 ($G_x = \text{Gyro X-axis data in degree/seconds} / \text{Gyro Y-axis data in degree/seconds} / \text{Gyro Z-axis data in degree/seconds}$) від кількості вимірювань, при різних кутах нахилу. На рисунку 4.6 приведено візуалізацію отриманих результатів.



а) Gyro X-axis data in degree/seconds; б) Gyro Y-axis data in degree/seconds;
в) Gyro Z-axis data in degree/seconds

Рисунок 4.6 – Графіки залежності отриманих даних з модуля GY-521 ($G_x =$ Gyro X-axis data in degree/seconds/ Gyro Y-axis data in degree/seconds / Gyro Z-axis data in degree/seconds) від кількості вимірювань, при різних кутах нахилу

Так як зооморфний мобільний робот знаходиться в тривимірному просторі, було прийнято рішення побудувати графік відповідності знаходження отриманих даних з модуля GY-521 у тривимірному просторі, це дозволить зрозуміти на майбутнє відхилення отриманих даних від нормаль, а також побудувати систему стабілізації мобільного робота на базі нейронних мереж. Отриманий графік представлено на рисунку 4.7.

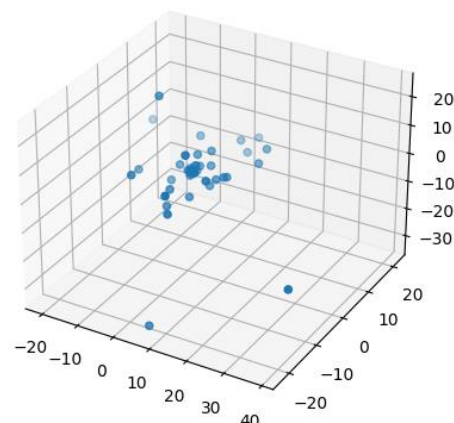
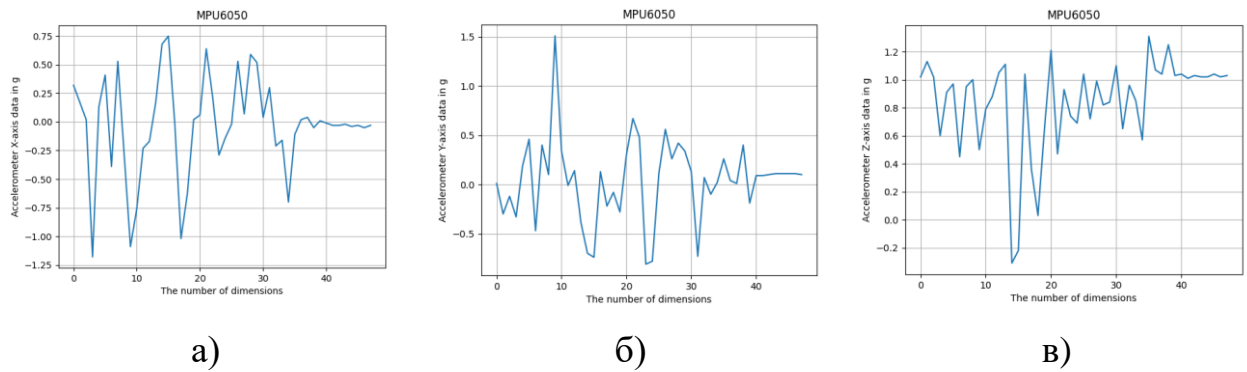


Рисунок 4.7 – Графік відповідності знаходження отриманих даних з модуля GY-521 у тривимірному просторі (Gyro X-axis, Gyro Y-axis, Gyro Z-axis data in degree/seconds)

Аналогічно проведемо аналіз отриманих даних з таблиці 4.1 для акселерометра в G.

```
import matplotlib.pyplot as plt
import numpy as np
Gx = [0.32, 0.17, 0.02, -1.18, 0.13, 0.41, -0.39, 0.53, -0.28, -1.09, -0.76, -
0.23, -0.17, 0.17, 0.68, 0.75, 0.00, -1.02, -0.63, 0.02, 0.06, 0.64, 0.22, -0.29, -0.14,
-0.02, 0.53, 0.07, 0.59, 0.52, 0.04, 0.30, -0.21, -0.16, -0.70, -0.11, 0.02, 0.04, -0.05,
0.01, -0.01, -0.03, -0.03, -0.02, -0.04, -0.03, -0.05, -0.03]
Gy = [0.01, -0.30, -0.12, -0.33, 0.19, 0.46, -0.47, 0.40, 0.10, 1.51, 0.34, -
0.01, 0.14, -0.39, -0.70, -0.74, 0.13, -0.22, -0.08, -0.28, 0.30, 0.67, 0.48, -0.81, -
0.78, 0.11, 0.56, 0.26, 0.42, 0.34, 0.13, -0.73, 0.07, -0.10, 0.02, 0.26, 0.04, 0.01,
0.40, -0.19, 0.09, 0.09, 0.10, 0.11, 0.11, 0.11, 0.11, 0.10 ]
Gz = [1.02, 1.13, 1.02, 0.60, 0.91, 0.97, 0.45, 0.95, 1.00, 0.50, 0.79, 0.88,
1.05, 1.11, -0.31, -0.22, 1.04, 0.36, 0.03, 0.65, 1.21, 0.47, 0.93, 0.74, 0.69, 1.04,
0.72, 0.99, 0.82, 0.84, 1.10, 0.65, 0.96, 0.85, 0.57, 1.31, 1.07, 1.04, 1.25, 1.03,
1.04, 1.01, 1.03, 1.02, 1.02, 1.04, 1.02, 1.03]
plt.title ('MPU6050')
plt.xlabel('The number of dimensions')
plt.ylabel('Accelerometer Z-axis data in g')
plt.plot(Az)
plt.grid(True)
plt.show()
```

В наслідок чого отримаємо графіки залежності отриманих даних з модуля GY-521 (accelerometer X-axis data in g / accelerometer Y-axis data in g / accelerometer Z-axis data in g) від кількості вимірювань, при різних кутах нахилу. На рисунку 4.8 приведено візуалізацію отриманих результатів.



а) accelerometer X-axis data in g; б) / accelerometer Y-axis data in g;
в) accelerometer Z-axis data in g

Рисунок 4.8 – Графіки залежності отриманих даних з модуля GY-521 (accelerometer X-axis data in g / accelerometer Y-axis data in g / accelerometer Z-axis data in g) від кількості вимірювань, при різних кутах нахилу

Так як зооморфний мобільний робот знаходиться в тривимірному просторі, було прийнято рішення побудувати графік відповідності знаходження отриманих даних з модуля GY-521 у тривимірному просторі, це дозволить зрозуміти на майбутнє відхилення отриманих даних від нормаль, а також побудувати систему стабілізації мобільного робота на базі нейронних мереж. Отриманий графік представлено на рисунку 4.9.

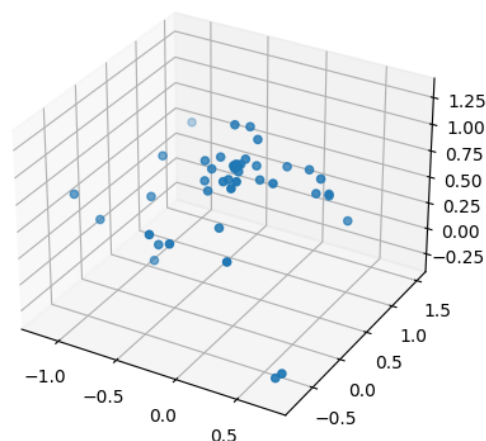


Рисунок 4.9 – Графік відповідності знаходження отриманих даних з модуля GY-521 у тривимірному просторі (accelerometer X-axis data in g / accelerometer Y-axis data in g / accelerometer Z-axis data in g)

Аналізуючи дані з акселерометра GY-521 (G_x , G_y , G_z), можна використовувати їх для реалізації управління зооморфним мобільним роботом типу Spot. Акселерометр вимірює прискорення робота по трьом осям (X , Y , Z) і дозволяє визначити його нахил у просторі. Наприклад, значення G_x можна використовувати для визначення нахилу робота вздовж горизонтальної осі (X -ось). Зміна цього значення може вказувати на нахил робота вперед або назад. Аналогічно, G_y вказує на нахил робота вздовж вертикальної осі (Y -ось), а G_z – на нахил робота вздовж осі Z .

З використанням цих даних можна реалізувати алгоритми контролю руху, стабілізації та орієнтації робота. Наприклад, якщо значення G_x змінюється в напрямку вперед, робот може виконувати рух вперед. Аналогічно, зміна значень G_y та G_z може використовуватися для керування рухом робота в інших напрямках та поворотів.

4.3. Розрахунок освітлення

Розрахунок освітлення в лабораторії включає оцінку необхідного рівня освітленості та вибір відповідного освітлювального обладнання. Основні критерії для розрахунку освітлення в лабораторії включають:

- площа лабораторії (S): 50 м^2 ;
- потрібний рівень освітленості (λ): 500 лк ;
- коефіцієнт використання приміщення (K_r): $0,8$ (80%);
- коефіцієнт запасу (K_s): $1,2$ (20%);
- коефіцієнт світлової віддачі світильників (K_v): $0,9$ (90%).

Проведемо розрахунки:

- розрахунок освітленості вираховується:

$$L_{\min} = S \times \lambda = 50 \text{ м}^2 \times 500 \text{ лк} = 25000 \text{ лк} ; \quad (4.1)$$

– розрахунок ефективної освітленості вираховується:

$$E_l = L_{\min} \times K_r \times K_s = 25000 \text{ лк} \times 0,8 \times 1,2 = 24000 \text{ лк}; \quad (4.2)$$

– розрахунок потужності світильників вираховується:

$$P_l = E_l / K_{lv} = 24000 \text{ лк} / 0,9 = 26667 \text{ лк}; \quad (4.3)$$

– вибір світильників: залежно від вимог і доступної технології, виберіть світильники, які найкраще відповідають потужності, типу та ефективності, щоб задовольнити розраховану потужність світильників;

– розміщення світильників: розмістіть світильники рівномірно по всій лабораторії, забезпечуючи рівномірний розподіл світла на робочих поверхнях.

Варто зазначити, що це лише приклад розрахунку, і фактичні значення можуть залежати від специфічних потреб вашої лабораторії та вимог нормативних документів, таких як будівельні кодекси та стандарти безпеки.

4.4 Висновки до 4 розділу

В ході виконання четвертого розділу, були решені наступні задачі:

– поставленні задачі проведення експерименту з модулем GY-521, розроблена схема підключення до Raspberry Pi;

– розроблено макет для проведення експерименту та встановлено в розроблений макет мобільного зооморфного робота типу Spot;

– розроблена програма для отримання та обробки даних з модуля GY-521 в середовищі розробки Tonny IDE на базі мови Python на базі OS Raspberry Pi;

– розроблена програма дозволяє отримати такі дані з модуля GY-521: Gy = Gyro Y-axis data in degree/seconds; Gz = Gyro Z-axis data in degree/seconds; Ax = Accelerometer X-axis data in g; Ay = Accelerometer Y-axis data in g; Az = Accelerometer Z-axis data in g як реального часу, результати експерименту приведені в таблиці 4.1;

– на базі отриманих даних було проведено аналіз та візуалізація їх відхилення від нормалі, при ранніх положеннях мобільного робота в просторі, отримані результати дають можливість реалізувати систему стабілізації мобільного робота з використанням нейронної мережі, але це дослідження не є метою даної роботи;

– побудовані графіки відповідності знаходження отриманих даних з модуля GY-521 у тривимірному просторі для даних Gyro X-axis, Gyro Y-axis, Gyro Z-axis data in degree/seconds та і для accelerometer X-axis data in g / accelerometer Y-axis data in g / accelerometer Z-axis data in g.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було проаналізовано конструкції зооморфних мобільних роботів, методи керування мобільним роботами та особливості керування зооморфними мобільним роботами. Розроблено модель кінематики зооморфного робота типу Spot, обґрунтовано та вибрано виконавчі механізми. Далі було розроблено конструкцію зооморфного робота, 3D моделі деталей конструкції мобільного робота, 3D моделі зборки мобільного робота типу Spot, 3D моделі макета. Проведено збірку макета зооморфного робота типу Spot, проаналізовано та вибрано апаратні модулі для реалізації системи керування. Розроблено структурну схему системи керування, схему підключення апаратних модулів та алгоритм керування зооморфними мобільним роботами. На останньому етапі було реалізовано функції керування серводвигунами, проведено експериментальні дослідження та проаналізовано отримані результати.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. – Введ. 2015-06-22. – К. Держстандарт України, 2017 – 29 с.

2. Невлюдов, І.Ш. Дипломне проектування для студентів усіх форм навчання спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології» [Текст]: навч. посіб. / І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарева, Г.В. Пономарьова. – Київ-58, пр. Космонавта Комарова, 1, 2016. – 320с.

3. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології, освітньо-професійних програм: «Автоматизоване управління технологічними процесами»; «Комп'ютерно-інтегровані технологічні процеси і виробництва»; «Комп'ютеризовані та робототехнічні системи» / Упоряд.: І. Ш. Невлюдов Р. В. Артюх В. В. Безкоровайний Н. П. Демська В. В. Євсєєв О. І. Филипенко О. М. Цимбал. Харків: ХНУРЕ, 2021. 55 с.

4. Манічкін М. Аналіз кінематики та розробка моделі розрахунків елементів матриці гомогенних перетворень для зооморфного мобільного робота / М. Манічкін // Автоматизація та Приладобудування («Automation and Development of Electronic Devices» ADED-2023) : збірник студентських наукових статей, 2023. – Харків : ХНУРЕ, 2023. – Вип. 2. – С. 49-54.

5. Розробка 3D-моделі зооморфного мобільного робота для вертикальних переміщень по металевим поверхням / І. Ш. Невлюдов, В. В. Євсєєв, Н. П. Демська, В. О. Руденко // Наука і техніка сьогодні. – 2022. – № 4(4). – С.163-174.

6. Долгуля А. В. Дослідження переміщення чотирилапого зооморфного робота «Робокіт» у невизначеному просторі / А. В. Долгуля // «Automation and Development of Electronic Devices» ADED-2023: Collection of Students' Scientific Paper. – Kharkiv : Kind of Kharkiv National University of Radio Electronics [electronic edition], 2023. – Part 1. –336p. P. 132-137.

7. Проектування мобільних маніпуляційних роботів: монографія / І. Ш. Невлюдов, А. О. Андрусевич, В. В. Євсєєв, С. П. Новоселов, Н. П. Демська; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків : ХНУРЕ, 2022. – 427 с.

8. Бронніков А. І. Моделі та методи адаптивного візуального керування роботами: автореф. дис. ... канд. техн. наук : 05.13.07 "Автоматизація процесів керування" / А. І. Бронніков ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2021. – 24 с.

9. Радченко Я. О. Розроблення методу ідентифікації об'єктів для мобільних роботів на базі асоціативної логіки: пояснювальна записка до атестаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 151 - Автоматизація та комп'ютерно-інтегровані технології / Я. О. Радченко; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки – Харків, 2021 – 69 с.

10. Щербина В. Розробка віддаленої системи екстреного керування мобільним роботом на базі ESP8266 / В. Щербина // «Automation and Development of Electronic Devices» ADED-2023: Collection of Students' Scientific Paper. – Kharkiv : Kind of Kharkiv National University of Radio Electronics [electronic edition], 2023. – Part 1. –310p. P. 245-247.

11. Сервопривод Tower Pro MG995 180° MG995-180 // ARDUINO.UA, 2023. URL: <https://arduino.ua/ua/prod2551-servoprivid-tower-pro-mg995-180> (дата звернення : 09.08.2023).

12. Сервопривод Tower Pro MG996R-180° MG996R-180 // ARDUINO.UA, 2023. URL: <https://arduino.ua/ua/prod5186-servodvigyn-tower-pro-mg996r-180> (дата звернення : 09.08.2023).

13. Металевий сервопривод TD-8135MG 180° 35 кг // ROBOSTORE, 2023. URL: <https://www.robostore.com.ua/metallicheskiy-servoprivod-td-8135mg-180-35-kg/> (дата звернення : 09.08.2023).

14. Невлюдов, І. Ш., Андрусевич, А. О., Євсєєв, В. В., Максимова, С. С., Стародубцев, М. Г., & Невлюдова, В. В. (2018). Автоматизована система керування технологічними процесами в SCADA системі TRACE MODE 6.

15. Микромікрокомп'ютер Raspberry Pi 3 Model B+ // ЕВОКОМ.ЮА, 2023. URL: <https://evo.net.ua/ua/raspberry-pi-3-model-b/> (дата звернення: 15.11.2023).

16. NVIDIA Jetson Nano Developer Kit // ЕВОКОМ.ЮА, 2023. URL: <https://evo.net.ua/nvidia-jetson-nano-developer-kit/> (дата звернення: 15.11.2023).

17. Orange Pi 4 LTS 4Гб DDR4/16Гб EMMC // ARDUINO.UA, 2023. URL: <https://arduino.ua/ua/prod4043-orange-pi-4-4gb-ddr4-16gb-emmc> (дата звернення: 15.11.2023).

18. Arduino Nano V3.0 AVR ATmega328P з распаяними роз'ємами // ARDUINO.UA, 2023. URL: <https://arduino.ua/ua/prod166-arduino-nano-v3-0-avr-atmega328p-s-raspayannimi-razemami> (дата звернення: 18.11.2023).

19. Плата розробника Arduino UNO+WiFi R3 // ARDUINO.UA, 2023. URL: <https://arduino.ua/ua/prod2014-plata-razrobotchika-arduino-unowifi-r3> (дата звернення: 18.11.2023).

20. 16-канальний 12-bit PWM/Servo модуль з I2C інтерфейсом на PCA9685 // ARDUINO.UA, 2023. URL: <https://arduino.ua/ua/prod1442-16-kanalnii-12-bit-pwmservo-modyl-s-i2c-interfeisom-na-pca9685> (дата звернення: 20.11.2023).

21. Arduino Motor Shield Rev3 A000079 // ARDUINO.UA, 2023. URL: <https://arduino.ua/ua/prod582-arduino-motor-shield-rev3> (дата звернення: 20.11.2023).